

ERASMUS+ PROJECT 2023-1-RS01-KA220-HED-000156660

EPIR | E-Procedure of Institutional Recognition of Foreign Higher Education Documents

WORK PACKAGE 3

WP3 - Development and improvement of IT systems for the recognition process of foreign students' HE documents

Report on programming phase

Project	E-Procedure of Institutional Recognition of Foreign Higher Education Documents
Work Package 3	Development and improvement of IT systems for the recognition process of foreign students' HE documents
Focus Activity 3	Programming phase of writing the code for a new IT tool or an upgraded software for the process of recognition of foreign HE documents as an integral part for international students integration into EHEA and partner HEIs
Prepared by	University of Novi Sad Faculty of Technical Sciences Igor Zečević Ljubomir Milašinović Center for Information Technologies Tatjana Zubić Tatjana Tošić Zoran Vojnović

EPIR project partners:



UNIVERSITY
OF NOVI SAD



UNIVERSITÀ
POLITECNICA
DELLE MARCHE



ROMANIA
1 DECEMBRIE 1918
UNIVERSITY OF ALBA IULIA



50 years of
University
of Split



REPUBLIC OF SERBIA
Qualifications Agency

Contents

1. Introduction	3
2. Methodological Approach and Development Strategy	3
3. Development Activities	4
3.1. System Architecture and Technology Stack.....	4
3.2. Data Model and Database Development.....	5
3.3. Backend Development and Service Implementation	7
3.4. Frontend Development and User Interface	7
3.5. System Integration and Functional Refinement	9
3.7. Monitoring, Observability and System Stabilization.....	9
4. Coordination, User Engagement and Knowledge Exchange.....	10
5. Main Achievements and Results.....	10
6. Conclusion and Next Steps.....	10
7. Appendices.....	11
APPENDIX I: Architecture and Technical Documentation of the System for Digitalization of the Recognition Process	
APPENDIX II: Conceptual and Physical Data Model of the System for Digitalization of the Recognition Process	
APPENDIX III: Graphical User Interface Documentation of the System for Digitalization of the Recognition Process	

1. Introduction

The primary objective of Work Package 3 (WP3) within the EPIR project is the development of new or the improvement of existing IT systems that support the institutional recognition of foreign higher education documents at partner institutions. This objective is pursued through a complete software development lifecycle, including system analysis and design, programming, testing, and implementation of new or upgraded digital tools.

At the University of Novi Sad (UNS) a new IT solution, designed to meet the UNS specific needs and requirements is developing in the scope of this work package. This software system should enable candidates to submit recognition requests and accompanying documents online, simplify the tracking of the recognition process for all participants, provide more efficient reporting for institutional and regulatory purposes, and reduce the number of paper documents. The introduction of this system is expected to benefit all stakeholders involved in the recognition process.

The University of Novi Sad WP3 project team consists of a design team from the Centre for Information Technologies at UNS, primarily responsible for system analysis and design and software development coordination, and a development team from the University of Novi Sad Faculty of Technical Sciences, tasked with coding and software development.

Following the completion of WP3 Activity 3.1 (System Analysis and System Design) by October 1, 2024, the resulting System Requirements and Design Specification served as the foundation for the programming phase and served as the main reference for the implementation of the system.

The main objective of the WP3 Activity 3 – Programming phase was to transform the validated requirements, system architecture, data models, and user interface concepts into an operational software solution prepared for subsequent testing and deployment phases.

2. Methodological Approach and Development Strategy

The Programming phase was conducted using an iterative and incremental development approach, enabling continuous refinement of functionalities based on feedback from users, legal experts, and institutional stakeholders. Development activities were organized in parallel streams covering system architecture, frontend user interfaces, and backend services.

Throughout the process, particular emphasis was placed on:

- alignment with the System Requirements Specification and institutional priorities defined under WP3 Activity 3.1;
- compliance with legal frameworks and administrative practices governing recognition procedures;
- performance optimization, scalability, reliability, and long-term sustainability of the system.

Coordination meetings, technical reviews, and consultations with end users supported informed decision-making and ensured that development activities remained aligned with project objectives and timelines.

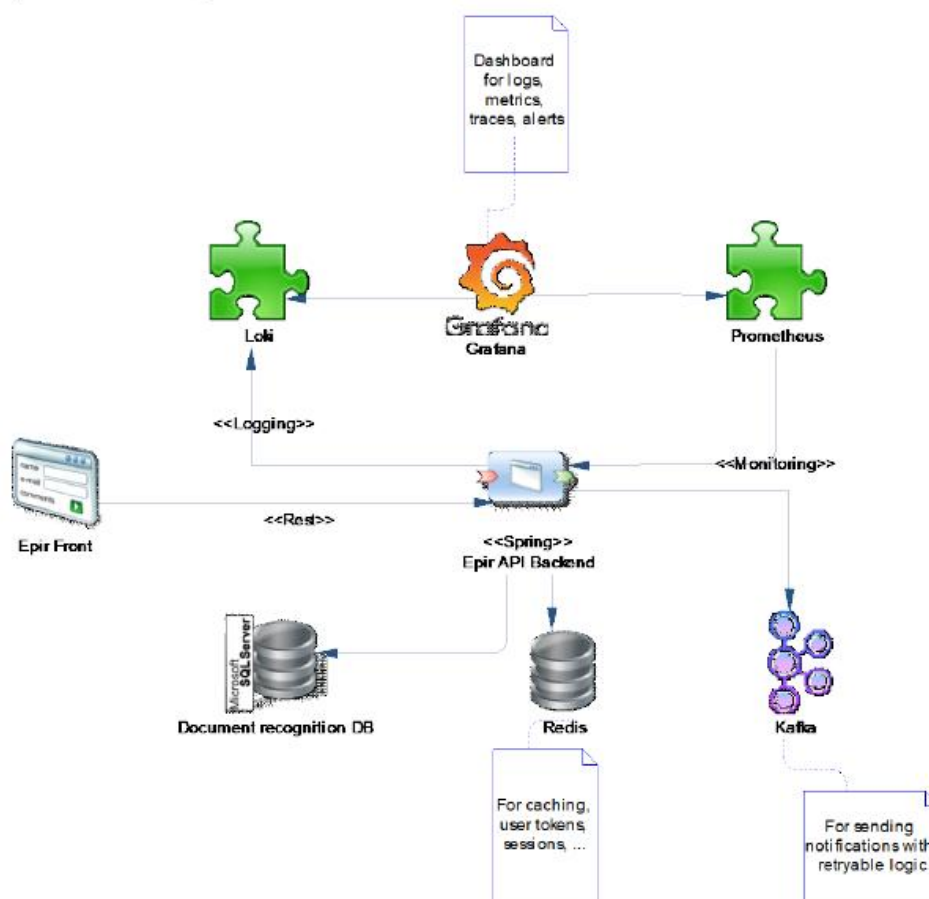
3. Development Activities

3.1. System Architecture and Technology Stack

The system architecture is based on a multi-layer approach consisting of a user interface layer, an application layer responsible for business logic and system services, and a data layer providing centralized storage and management of system data and documents.

The architecture enables applicants to submit recognition requests and supporting documentation through a web-based interface. Submitted information is processed by the application layer, where validation, workflow management, communication between system components, and other business processes are performed. The solution supports the management of recognition procedures and enables relevant users to access information according to their roles and permissions.

Details of the system architecture are in *APPENDIX I: Architecture and Technical Documentation of the System for Digitalization of the Recognition Process* (in Serbian).



EPIR system architecture

The technology stack was defined, including the selection of technologies, tools, and platforms to be used in the implementation. The technology stack includes:

- Frontend technologies – HTML, CSS, JavaScript, jQuery, Bootstrap;
- Backend technologies – Spring Boot, JPA, REST architectural style for communication;
- Database – Microsoft SQL Server relational database, MongoDB with GridFS extension for document file storage;

- Code versioning and team collaboration platform – GitLab was chosen as the platform for code management, combining version control functionality with development tools and CI/CD (Continuous Integration/Continuous Deployment) processes.

The architecture was subsequently reviewed and enhanced to address additional requirements related to performance, scalability, asynchronous communication, monitoring, and system reliability.

3.2. Data Model and Database Development

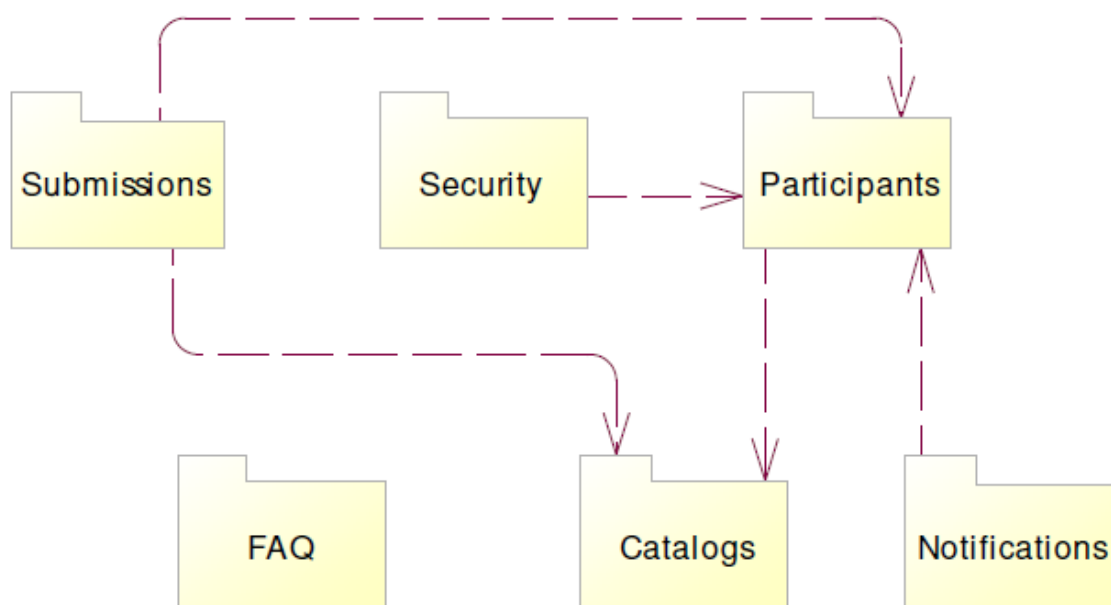
The conceptual and physical data models developed during the System Analysis and System Design phase provided the initial foundation for database implementation.

The initial database structure was designed and implemented based on the defined system requirements and architecture. The physical database design was adapted for the Microsoft SQL Server relational database management system, while MongoDB with GridFS was used for document file storage.

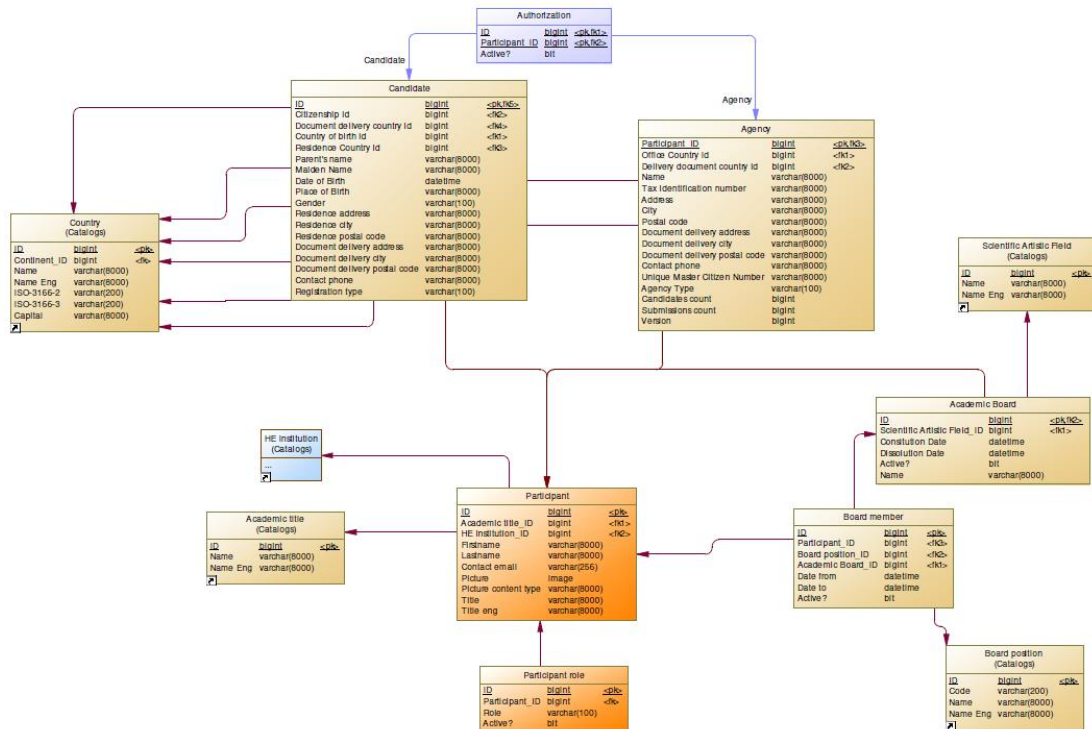
During the development process, the use of the functional prototype and further consultations with users identified the need to refine certain entities, relationships, and data structures. The conceptual and physical data models were therefore reviewed and optimized.

The revision included the redefinition of selected entities and their relationships and optimization of data structures with the aim of improving performance, scalability, maintainability, and support for the more complex business processes required by the recognition procedure.

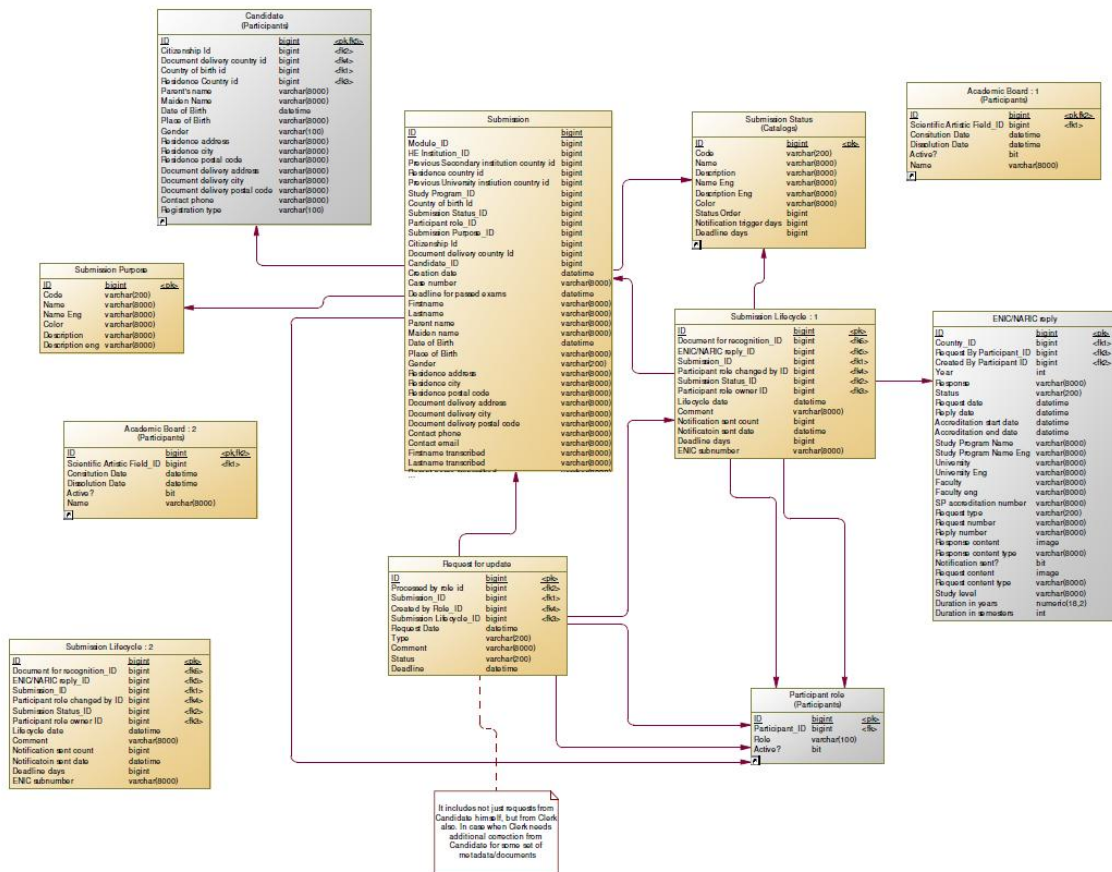
Based on the revised model, an optimized relational database structure was implemented. The resulting data model provided the foundation for the subsequent development and integration of backend services.



Conceptual data model



Physical data model (diagram Participants)



Physical data model (diagram Submission request)

For illustrative purposes, selected diagrams from the data model are presented, including the conceptual data model and the physical data model (Participants and Submission Request diagram). The complete data model and its detailed description are provided in the separate technical documentat *APPENDIX II: Conceptual and Physical Data Model of the System for Digitalization of the Recognition Process* (in Serbian).

3.3. Backend Development and Service Implementation

Backend development was carried out using the Spring Boot framework and REST-based services. The implementation followed the modular architecture defined during the system design phase.

The initial backend implementation established the framework for API services, including modules, controllers, services, data entities, and basic system configuration.

Authentication and authorization mechanisms were implemented to ensure that access to system functionalities and data is controlled according to user roles and permissions.

Core catalog functionality was implemented and subsequently extended to support the requirements of the recognition process.

The main groups of implemented backend services include:

- user authentication and authorization;
- management of core data catalogs;
- participant management;
- notification services;
- submission and recognition request services.

The backend components were continuously refined to address functional requirements, integration needs, identified issues, and changes resulting from consultations with users and institutional stakeholders.

The implementation of these services provided the core business logic required for the recognition process and enabled their subsequent integration with the graphical user interface.

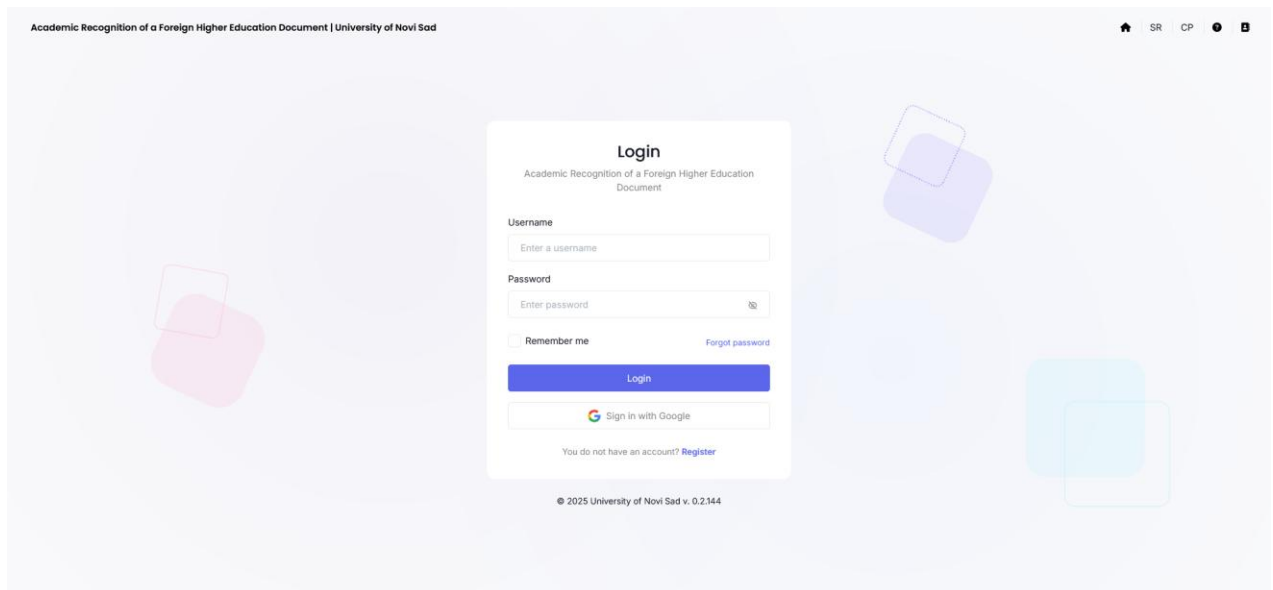
3.4. Frontend Development and User Interface

The frontend development was based on the high-fidelity user interface mockups prepared during the system analysis and design and refined through consultations with prospective users and institutional stakeholders.

The mockups covered the main user roles involved in the recognition process, including candidates, authorized representatives/agencies, officers, supervisors, vice-deans, heads of study programmes, and administrators. They provided a visual representation of the proposed system and supported validation of the planned workflows and user interactions.

An initial frontend prototype was subsequently implemented using Bootstrap, jQuery, and JavaScript libraries.

The user interface was continuously refined to ensure consistency with the approved system design, institutional workflows, user roles, and functional requirements.



Academic Recognition of
a Foreign Higher
Education Document

DASHBOARD

Calendar

CANDIDATE

My Information

APPLICATIONS

View Applications

Submission of
Application

HELP

Instructions

Documents

Contacts

Search

CTRL D

Request

Request #32450-23452345

Candidate

Viktor Boyko

Submission Date

19.08.2025

Status

Pokrenut postupak

Purpose

Upla narednog semestra

Level of Study

MAS

Institution

DF

Program/Module

Pozivka veštinama, fakultet učenosti i znanja

Edit Request

Withdraw Request

Show Timeline

PERSONAL DATA

Data on Acquired Document

Previous Education Data

Documents

Applicant's Personal Information

as stated in the personal document attached to this request

First name

Viktor

Surname

Boyko

Parent's first name

Viktor

Gender

Male

Female

Maiden Name

Enter maiden name

Date of Birth

09.10.2000.

Place of Birth

Marrakesh

Country of Birth

Maroko

Citizenship

Uzbekistan

RESIDENTIAL ADDRESS

as stated in valid personal document

Street and Number

City of Residence

Country of Residence

Page 8

Academic Recognition of a Foreign Higher Education Document

DASHBOARD

- Applications
- Calendar

APPLICATIONS

View Applications

- Submission of Application
- Agencije
- Candidates

ENIC/NARIC

- Database of answers
- Sending Inquiry

CATALOGS

- Territorial org.
- University
- Documents

REPORTS AND STATISTICS

- Reports
- Statistics

Search

Applications > View Applications

Collapse All Show All Clear All

Filters active - 0

Purpose of appl.	Degrees of Studies	Name of faculty	Status	Reviewer
Uključivanje u započeti nivo 29	DAS 4	AU 6	Dopuna zahteva 1	Empty 9
Upis istog nivoa 7	MAS 11	DIF 4	Dopunjen zahtev 1	Milica Ostojić 13
Upis narednog nivoa 6	MSS 1	EF 2	Podnet 7	Nataša Maksimović 10
	OAS 5	FF 3	Podnošenje zahteva 2	Una Carević 10

Show 10 entries

No.	Application number	Surname	First name	Purpose of application	Degrees of Studies	Name of faculty	Study programme	Status	Reviewer
1	1-12/23	Sokolov	Anna	Uključivanje u započeti nivo	MAS	FF	Komunikologija	Pokrenut postupak	Milica Ostojić
2	1-12/24	Mutiso	Esther	Upis istog nivoa	MAS	FTN	Računarstvo i automatika	Predmet upućen fakultetu	Una Carević
3	1-12/25	Kamau	John	Uključivanje u započeti nivo	OAS	MF	Sestrinstvo	Podnet	
4	1-12/26	Novikov	Elena	Uključivanje u započeti nivo	MAS	EF	Digitalni marketing	Pokrenut postupak	Milica Ostojić
5	1-12/27	Kravchenko	Oksana	Upis narednog nivoa	MAS	EF	Digitalni marketing	Podnošenje zahteva	
6	1-12/28	Moroz	Kateryna	Uključivanje u započeti nivo	DAS	PMF	Prirодно-matematički fakultet	Podnošenje zahteva	
7	1-12/29	Chibrikova	Ekaterina	Upis istog nivoa	MAS	FTN	Računarstvo i automatika	Predmet upućen fakultetu	Milica Ostojić

GUI examples

Detailed information on the graphical user interface is provided in the separate technical document *APPENDIX III: Graphical User Interface Documentation of the System for Digitalization of the Recognition Process* (in Serbian).

3.5. System Integration and Functional Refinement

The integration of frontend and backend components was carried out progressively throughout the Programming phase.

The frontend application was connected to the implemented backend services, enabling the validation of complete system workflows rather than individual components in isolation. The integration process enabled the development teams to identify technical and functional inconsistencies at an early stage and to implement the necessary adjustments.

Backend services and frontend components were refined in parallel to ensure that the implemented functionality corresponded to the defined workflows and institutional requirements.

3.7. Monitoring, Observability and System Stabilization

An observability and monitoring environment was implemented as part of the final development activities.

The implemented observability stack includes:

- *Prometheus* for system metrics collection and monitoring;
- *Loki* for centralized log aggregation and analysis;
- *Grafana* for visualization of system metrics and creation of monitoring dashboards.

The monitoring environment enables continuous observation of system performance, supports faster identification and diagnosis of potential issues, and provides a foundation for future system optimization and maintenance.

4. Coordination, User Engagement and Knowledge Exchange

The Programming phase was supported by continuous cooperation between the UNS design team, development team, legal team, prospective users, and project partners.

Coordination meetings between the design, development, and legal teams were used to clarify open technical and functional questions and to ensure a shared understanding of the system requirements.

Consultations contributed to the refinement of recognition process workflows, user interface and user experience, user roles and permissions, data requirements, reporting requirements, and functional requirements.

Knowledge exchange with project partners also supported the development process. The exchange of IT experiences and approaches enabled the partners to compare practices related to the digitalization of recognition and other higher education administrative processes and to consider relevant approaches in the development of their respective solutions.

5. Main Achievements and Results

The Programming phase resulted in substantial progress from the validated system requirements and initial prototypes towards an operational software solution.

The main achievements of the phase include:

- implementation of the defined system architecture and technology stack;
- development and optimization of the conceptual and physical data model;
- implementation of the relational database structure;
- implementation of document storage using MongoDB/GridFS;
- implementation of core backend services;
- implementation of authentication and authorization mechanisms;
- implementation and enhancement of catalog management services;
- implementation of services related to participants, notifications, and submissions;
- development of an operational graphical user interface;
- integration of frontend and backend components;
- implementation of Redis for caching and session management;
- implementation of Apache Kafka for asynchronous notification handling and event-driven communication;
- implementation of Prometheus, Loki, and Grafana for system monitoring and observability;
- continuous refinement and stabilization of the system based on technical findings and institutional feedback.

The Programming phase therefore transformed the outputs of the System Analysis and System Design phase into an operational software solution tailored to the specific institutional requirements of the University of Novi Sad.

6. Conclusion and Next Steps

The programming phase of **WP3 Activity 3** at the University of Novi Sad was successfully completed in accordance with the project plan and defined objectives.

During the software development period, the project team progressed from system requirements, architectural concepts, data models, and user interface mockups to an operational software solution prepared for systematic testing. Further refinement and development will be carried out during the testing phase, based on the results of system validation, to ensure full alignment with user needs and project requirements.

All activities were carried out with a clear focus on sustainability, legal compliance, and alignment with institutional needs for the recognition of foreign higher education documents.

7. Appendices

The following technical documents form an integral part of this report (available in Serbian):

- Architecture and Technical Documentation of the System for Digitalization of the Recognition Process (Arhitektura i tehnička dokumentacija sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava)
- Conceptual and Physical Data Model of the System for Digitalization of the Recognition Process (Konceptualni i fizički model strukture podataka sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava)
- Graphical User Interface Documentation of the System for Digitalization of the Recognition Process (Grafički korisnički interfejs sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava)

APPENDIX I: Architecture and Technical Documentation of the System for Digitalization of the Recognition Process

ERASMUS+ PROJECT 2023-1-RS01-KA220-HED-000156660

EPIR | E-Procedure of Institutional Recognition of Foreign Higher Education Documents

Dokument:	Arhitektura i tehnička dokumentacija sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava
Verzija:	1.0
Dokument pripremili:	Danijel Venus Igor Zečević Ljubomir Milašinović

EPIR project partners:



UNIVERSITY
OF NOVI SAD



UNIVERSITÀ
POLITECNICA
DELLE MARCHE



ROMANIA
1 DECEMBRIE 1918
UNIVERSITY OF ALBA IULIA



50 years of
University
of Split



REPUBLIC OF SERBIA
Qualifications Agency

APSTRAKT

Cilj digitalizacije procesa priznavanja stranih visokoškolskih isprava koju obavlja kancelarija za pravne poslove Univerziteta u Novom Sadu je omogućavanje kandidatima online podnošenje zahteva za priznavanje i pratećih dokumenata, kao i povećanje efikasnosti rada.

Na osnovu prethodno obavljene analize sistema, formirane specifikacije zahteva i kreiranja uzorka dizajna korisničkog interfejsa (mockup), kreiranog operativnog prototipa aplikacije kao i implementiranog konceptualnog, fizičkog modela strukture podataka kao i same baze podataka pristupilo se izradi arhitekture sistema kao i implementaciji API Backend servisa.

Arhitektura EPIR sistema zasniva se na višeslojnom modelu koji obuhvata korisnički interfejs za online podnošenje zahteva, aplikacioni sloj za obradu podataka i integracije, te centralizovanu bazu podataka. Kandidati putem web portala podnose zahteve i dokumenta, koja se automatski prosleđuju odgovarajućim službama. Aplikacioni sloj obavlja validacije, upravljanje tokovima i komunikaciju sa internim servisima Univerziteta. Sistem omogućava transparentno praćenje statusa, bržu obradu i pouzdano čuvanje svih podataka.

ABSTRACT

The goal of digitalizing the process of recognizing foreign higher education documents, carried out by the Legal Affairs Office of the University of Novi Sad, is to enable applicants to submit recognition requests and accompanying documents online, as well as to increase work efficiency.

Based on the previously conducted system analysis, the established requirements specification, the creation of a user interface mockup, the developed operational prototype of the application, and the implemented conceptual and physical data model including the database itself, work commenced on designing the system architecture and implementing the API backend service.

The EPIR system architecture is based on a multi-layer model that includes a user interface for online application submission, an application layer for data processing and integrations, and a centralized database. Applicants submit requests and documents through the web portal, which are automatically forwarded to the appropriate departments. The application layer performs validations, workflow management, and communication with the University's internal services. The system enables transparent status tracking, faster processing, and reliable storage of all data.

Sadržaj

Opseg sistema.....	5
Pregled sistema.....	6
Komponentne i uloge.....	7
Epir Front (Web UI)	7
Epir API Backend (Spring Boot)	7
Podkomponente:	7
Document recognition DB (MS SQL Server).....	7
Redis.....	7
Kafka.....	7
Observability stack.....	8
Primeri tokova podataka.....	9
API i primeri REST poziva	10
Paket Security	10
Paket Osnovni katalogi	13
Paket Participanti	27
Keširanje i sesije.....	31
Asinhroni mehanizmi i Kafka workflow	32
Observability (monitoring, logging, alerting)	33
Metrike (Prometheus).....	33
Logovi (Loki)	33
Grafana dashboards i alerts	33
Sigurnost i autentikacija.....	34
DevOps — deployment i operacije	35
Okruženja	35
CI/CD	35
Containerization i orkestracija	35
Backup i DR	35
Skaliranje i performanse	36
Testiranje i kvalitet.....	37
Rizični faktori i predlozi za unapređenje.....	38
Rizični faktori.....	38
Potencijalno unapređenje.....	38

Opseg sistema

Kancelarija za pravne poslove Univerziteta u Novom Sadu obavlja poslove priznavanja stranih visokoškolskih isprava za potrebe nastavka školovanja na Univerzitetu u Novom Sadu. Ova aktivnost predstavlja važan segment međunarodne saradnje i mobilnosti studenata, jer omogućava kandidatima iz drugih država da nastave svoje obrazovanje u okviru Univerziteta, u skladu sa važećim propisima i standardima.

Trenutno, proces priznavanja stranih visokoškolskih isprava nije digitalizovan, već se većina dokumenata i komunikacije odvija u papirnom obliku. Kandidati zahteve podnose lično ili putem pošte, dok zaposleni u Kancelariji za pravne poslove u upravljanju procesom koriste imejl komunikaciju i standardne kancelarijske softvere (tekstualne obrade, tabele, baze u ograničenom obliku). Ne postoji namensko softversko rešenje koje bi objedinjavalo sve aktivnosti i omogućilo integrisano praćenje toka priznavanja — od prijema zahteva do izdavanja odluke.

Digitalizacija ovog procesa ima za cilj da značajno unapredi način rada i iskustvo korisnika. Novi softverski sistem omogućiće kandidatima da elektronski podnesu zahtev za priznavanje i prate njegov status u realnom vremenu, dok će Kancelariji za pravne poslove i drugim organizacionim jedinicama Univerziteta obezbediti centralizovan, pregledan i efikasan način vođenja postupka.

Očekivani efekti uvođenja sistema su:

- Povećanje efikasnosti rada kroz automatizaciju administrativnih koraka i smanjenje manualnog unosa podataka,
- Ubrzanje razmene informacija između svih učesnika u procesu,
- Smanjenje upotrebe papira i prelazak na elektronsku dokumentaciju,
- Poboljšano izveštavanje i jednostavnije generisanje statističkih podataka za potrebe Univerziteta i nadležnih organa, kao i povećanje transparentnosti i pristupačnosti procesa priznavanja za kandidate iz inostranstva.

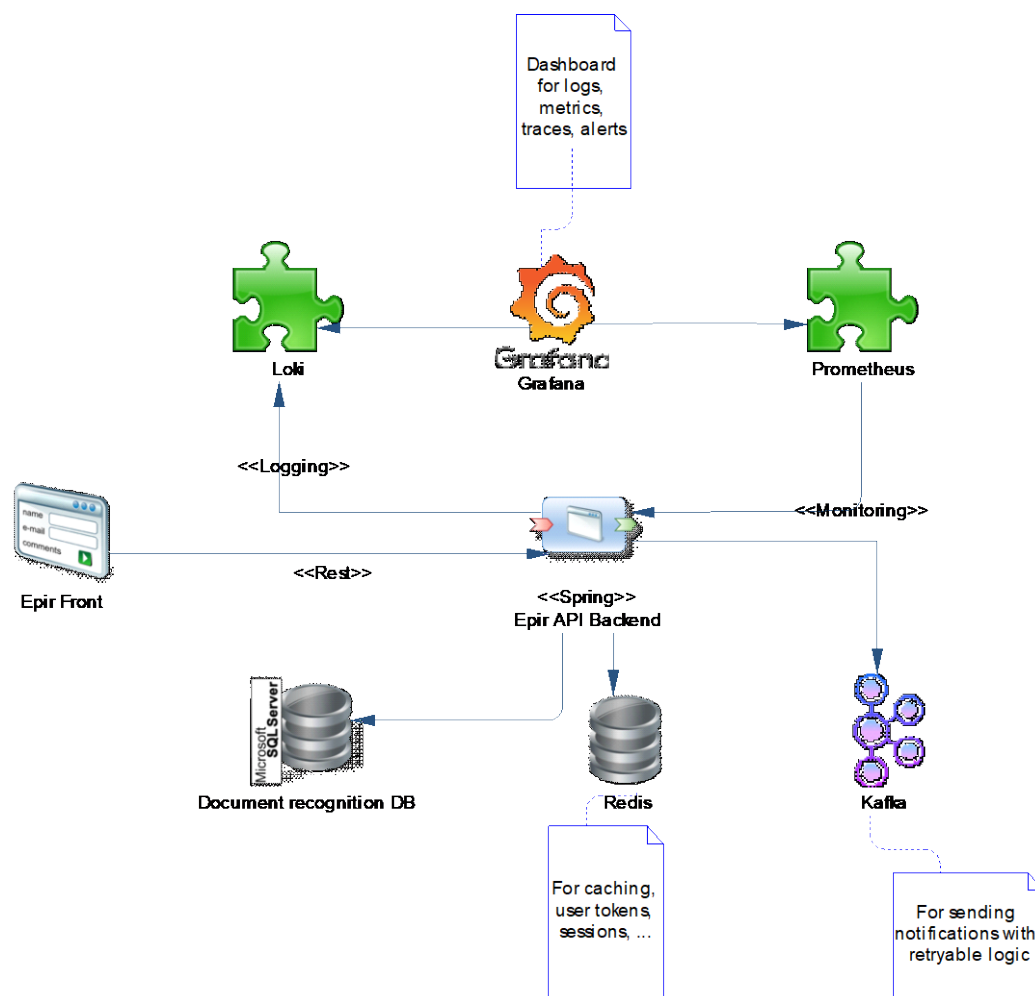
Dodatno, očekuje se da digitalizacija ovog postupka doprinese jačanju međunarodne vidljivosti Univerziteta u Novom Sadu i povećanju broja stranih studenata zainteresovanih za nastavak studija na Univerzitetu, čime će se unaprediti i proces akademske mobilnosti i međunarodne saradnje.

Pregled sistema

Sistem obezbeđuje:

- Web interfejs za kandidate (Epir Front)
- REST API backend (Spring)
- Relacionu bazu podataka za trajno čuvanje (Document recognition DB — MS SQL Server)
- Redis za keš i upravljanje sesijama
- Apache Kafka za pouzdano asinhrono slanje notifikacija i event-driven integracije
- Observability stack: Prometheus (metrike), Loki (logovi), Grafana (dashboards)

Na slici ispod ilustrovana je arhitektura EPIR sistema.



Sistem koristi klasični frontend-backend-database model sa dodatim slojem za asinhrono procesuiranje (Kafka) i keširanjem (Redis). Backend je implementiran u Spring ekosistemu, što omogućava modularnost, dependency injection i lak pristup već postojećim bibliotekama za autentikaciju, validaciju i integraciju sa bazama.

Observability se postiže slanjem metričkih podataka (Prometheus) i logova (Loki), koji se vizualizuju u Grafana dashboardima. Takva postavka omogućava brzo pronalaženje grešaka i praćenje performansi.

Komponentne i uloge

Epir Front (Web UI)

Single Page Application (SPA) — implementiran kroz nativni jQuery i Bootstrap.

Komunikacija sa backendom isključivo preko HTTPS REST API poziva.

Funkcionalnosti: prijava/registracija, popunjavanje formulara, upload dokumenata, pregled statusa zahteva, preuzimanje izveštaja.

Epir API Backend (Spring Boot)

Prima REST zahteve i orkestrira pozive ka bazi, Redis-u i Kafka-i.

Služi kao centralna tačka autorizacije i validacije podataka.

Izlaže metričke endpoint-e (Prometheus /actuator/prometheus).

Loguje u otvoreni format koji je kompatibilan sa Lokijem.

Podkomponente:

Controller layer (REST controllers)

Service layer (business logic)

Repository layer (DB access — JPA / JDBC templates)

Integration layer (Redis client, Kafka producer, notifier)

Document recognition DB (MS SQL Server)

Tabelarna struktura i transakcioni model za konzistentnost pri promenama statusa procesa.

Redis

Čuvanje session tokena, temporary upload metadata, rate limiting counters, cache za česte upite (npr. reference tabele, lista država, tipovi dokumenata).

TTL politika za ključeve (sesije npr. 30-60 minuta, cache 5-60 minuta zavisno od tipa).

Kafka

Topic primeri: `notifications`, `document-processing`, `audit-events`, `integration-outgoing`.

Producer iz backenda kreira poruke pri promeni statusa ili kada treba poslati notifikaciju.

Consumer (poseban servis ili funkcionalnost unutar backenda) obrađuje retries i pokušaje isporuke (email/SMS/poslati ka eksternom sistemu).

Observability stack

Prometheus: periodično (scrape) skuplja metrike sa backenda i drugih komponenti.

Loki: centralna agregacija logova, podržava label-based filtere (npr. ``app=epir-backend, env=prod, request_id=...``).

Grafana: definisani dashboard-i: Health overview, API performance, DB metrics, Kafka lag, Redis hits/misses, Error rates.

Primeri tokova podataka

Podnošenje zahteva (sinteza sekvence)

1. Korisnik popunjava zahtev i prilaže dokumente u Epir Front (HTTPS POST multipart/form-data).
2. Front poziva `POST /api/requests` na Epir API.
3. Backend validira payload i persistira osnovne podatke u SQL DB (tabela Request), vraća privremeni `request\_id`.
4. Backend smešta metapodatke o upload-u u Redis (dodatan ključ za brzi pristup) i vraća status uspeha frontendu.
5. Backend proizvodi poruku u Kafka topic `document-processing` sa informacijom da je novi dokument spreman.
6. Consumer koji obrađuje `document-processing` preuzima poruku, pokušava dalje korake (npr. ekstrakcija metapodataka OCR, pozivanje eksternih servisa).
7. Nakon uspešnog/neuspešnog rezultata, consumer ažurira Request status u SQL DB i šalje notifikaciju preko `notifications` topica.
8. Notifikacioni servis/consumer pokušava isporuku notifikacije (email/SMS) sa retry logikom.

Pregled i praćenje statusa

Front periodično (ili websocket/push) poziva `GET /api/requests/{id}/status`.

Backend čita iz SQL DB; za optimizaciju može prvo proveriti Redis cache.

API i primeri REST poziva

U nastavku slede primeri API i REST poziva grupisani po logičkim paketima.

Paket Security

POST [/users/renew-token](#)

Obnova tokena/promena uloge

Prava pristupa: ***admin, all***

Ukoliko se ne prosledi newRole, obnavlja se token na standardni način za trenutnu ulogu korisnika.

Ukoliko se prosledi newRole i različit je od trenutnog, takođe se obnavlja token, ali se u novi postavlja druga vrednost polja loggedInAs.

Request body

```
{  
  "accessToken": "string",  
  "refreshToken": "string",  
  "newRole": "candidate"  
}
```

POST [/users/registration](#)

Registracija korisnika

Prava pristupa: ***admin, all***

Potvrdom registracije, na imejl korisnika se šalje link za aktivaciju naloga koji je vremenski ograničen.

U pozadini se kreira participant slog sa kontakt imejl adresom i popunjenim personalnim podacima. I postavlja se odgovarajuća uloga participanta (kandidat, agencija, punomoćnik)

Sve dok korisnički nalog nije aktivan, može se izvršiti ponovni zahtev za registracijom sa istom imejl adresom. U tom slučaju se prethodni podaci o participantu brišu i kreiraju novi.

Request body

```
{  
  "email": "string",
```

```
"password": "string",
"passwordAgain": "string",
"recaptcha": "string",
"candidate": {
  "firstName": "string",
  "lastName": "string",
  "citizenshipId": 0,
  "documentDeliveryCountryId": 0,
  "countryOfBirthId": 0,
  "residenceCountryId": 0,
  "parentsName": "string",
  "maidenName": "string",
  "dateOfBirth": "2025-12-12T12:40:57.684Z",
  "placeOfBirth": "string",
  "gender": "male",
  "residenceAddress": "string",
  "residenceCity": "string",
  "residencePostalCode": "string",
  "documentDeliveryAddress": "string",
  "documentDeliveryCity": "string",
  "documentDeliveryPostalCode": "string",
  "contactPhone": "string"
},
"agency": {
  "officeCountryId": 0,
  "deliveryDocumentCountryId": 0,
  "name": "string",
  "taxIdentificationNumber": "string",
  "address": "string",
  "city": "string",
  "postalCode": "string",
  "documentDeliveryAddress": "string",
  "documentDeliveryCity": "string",
```

```
"documentDeliveryPostalCode": "string",
"contactPhone": "string"
},
"agent": {
  "firstName": "string",
  "lastName": "string",
  "officeCountryId": 0,
  "deliveryDocumentCountryId": 0,
  "address": "string",
  "city": "string",
  "postalCode": "string",
  "documentDeliveryAddress": "string",
  "documentDeliveryCity": "string",
  "documentDeliveryPostalCode": "string",
  "contactPhone": "string",
  "uniqueMasterCitizenNumber": "string"
}
}
```

POST </users/logout>

Logout korisnika

Prava pristupa: **admin, all**

Nakon uspešnog logout-a korisnika, sesija se invalidira, tako da više nije moguće koristiti zadati par tokena.

Session id se može isčitati iz refresh tokena. Potrebno je proslediti session id prilikom logout operacije. Urađeno tako da bismo mogli podržati da korisnik može da se odjavi iz neke sesije koja trenutno nije aktivan u njegovom browseru, kao što recimo gmail podržava.

Request body

```
{
  "sessionId": "string"
}
```

POST </users/login>

Login korisnik

Prava pristupa: **admin, all**

Ako je uspešan login, proveriti u response-u da li je polje loggedInAs prazno ili popunjeno.

- Ako je prazno, znači da korisnik ima više uloga u sistemu i potrebno je pozvati dodatni endpoint za dobijanje konkretnog tokena (renew-token) po prosleđenoj ulozi.
- U suprotnom access token je operativan za korišćenje servisa

Napomena: Token koji nema loggedInAs je takođe operativan za određeni broj servisa. Jedino gde nastaje problem jeste kada treba da pozove neku funkciju kao konkretna uloga, na primer u životnom ciklusu obrade zahteva za priznavanje isprave.

Request body

```
{  
  "username": "string",  
  "password": "string",  
  "recaptcha": "string"  
}
```

GET </users/activation>

Aktivacija korisnika

Prava pristupa: **admin, all**

Paket Osnovni katalozi

Statusi zahteva za priznavanje isprava

GET </catalogs/submission-statuses/{id}>

Čitanje podataka o statusu

Prava pristupa: **admin, all**

PUT [/catalogs/submission-statuses/{id}](#)

Izmena osnovnih podataka o statusu

Prava pristupa: admin

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "description": "string",  
  "nameEng": "string",  
  "descriptionEng": "string",  
  "color": "string"  
}
```

DELETE [/catalogs/submission-statuses/{id}](#)

Brisanje statusa

Prava pristupa: **admin**

GET [/catalogs/submission-statuses](#)

Čitanje svih statusa

Prava pristupa: **admin, all**

POST [/catalogs/submission-statuses](#)

Kreiranje novog statusa

Prava pristupa: **admin**

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "description": "string",  
}
```

```
"nameEng": "string",  
"descriptionEng": "string",  
"color": "string"  
}
```

GET </catalogs/submission-purposes/{id}>

Čitanje podataka o svrsi

Prava pristupa: ***admin, all***

PUT </catalogs/submission-purposes/{id}>

Izmena osnovnih podataka o svrsi

Prava pristupa: ***admin***

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "nameEng": "string",  
  "color": "string"  
}
```

DELETE </catalogs/submission-purposes/{id}>

Brisanje svrhe

Prava pristupa: ***admin***

GET </catalogs/submission-purposes>

Čitanje svih svrha

Prava pristupa: ***admin, all***

POST </catalogs/submission-purposes>

Kreiranje nove svrhe

Prava pristupa: **admin**

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "nameEng": "string",  
  "color": "string"  
}
```

GET </catalogs/study-programs/{id}>

Čitanje podataka o studijskom programu

Prava pristupa: **admin, all**

PUT </catalogs/study-programs/{id}>

Izmena osnovnih podataka o studijskom programu

Prava pristupa: **admin**

Request body

```
{  
  "helInstitutionId": 0,  
  "studyLevelId": 0,  
  "languageId": 0,  
  "scientificArtisticFieldId": 0,  
  "code": "string",  
  "name": "string",  
  "nameEng": "string",  
  "title": "string",  
  "accreditationNumber": "string",  
  "accreditationDate": "2025-12-14T10:06:19.764Z",  
  "semesterCount": 1,  
}
```

```
"totalEctsCredits": 0,  
"accreditationValidTo": "2025-12-14T10:06:19.764Z",  
"freshmanCount": 0  
}
```

DELETE </catalogs/study-programs/{id}>

Brisanje studijskog programa

Prava pristupa: **admin**

PUT /catalogs/study-programs/modules/{moduleId}

Izmena modula modula

Prava pristupa: admin

Request body

```
{  
  "name": "string",  
  "nameEng": "string",  
  "startingBlock": 0,  
  "studentCount": 0  
}
```

DELETE </catalogs/study-programs/modules/{moduleId}>

Brisanje modula

Prava pristupa: **admin**

GET </catalogs/study-programs>

Čitanje svih studijskih programa sa opcionim filterima

Prava pristupa: **admin, all**

POST </catalogs/study-programs>

Kreiranje novog studijskog programa

Prava pristupa: **admin**

Request body

```
{
  "heInstitutionId": 0,
  "studyLevelId": 0,
  "languageId": 0,
  "scientificArtisticFieldId": 0,
  "code": "string",
  "name": "string",
  "nameEng": "string",
  "title": "string",
  "accreditationNumber": "string",
  "accreditationDate": "2025-12-14T10:09:03.951Z",
  "semesterCount": 1,
  "totalEctsCredits": 0,
  "accreditationValidTo": "2025-12-14T10:09:03.951Z",
  "freshmanCount": 0
}
```

POST /catalogs/study-programs/{studyProgramId}/modules

Dodavanje novog modula studijskom programu

Prava pristupa: admin

Request body

```
{
  "name": "string",
  "nameEng": "string",
  "startingBlock": 0,
  "studentCount": 0
}
```

GET </catalogs/study-levels/{id}>

Čitanje jednog nivoa studija

Prava pristupa: **admin, all**

PUT </catalogs/study-levels/{id}>

Izmena nivoa studija

Prava pristupa: **admin**

DELETE </catalogs/study-levels/{id}>

Brisanje nivoa studija

Prava pristupa: **admin**

GET </catalogs/study-levels>

Čitanje svi nivoa studija

Prava pristupa: **admin, all**

POST </catalogs/study-levels>

Kreiranje nivoa studija

Prava pristupa: **admin**

Request body

```
{  
  "degree": 0,  
  "level": 0,  
  "name": "string",  
  "nameEng": "string",  
  "shortName": "string",  
  "color": "string",  
  "minEctsCredits": 0  
}
```

GET </catalogs/scientific-artistic-fields/{id}>

Čitanje podataka o polju

Prava pristupa: ***admin, all***

PUT </catalogs/scientific-artistic-fields/{id}>

Izmena osnovnih podataka o polju

Prava pristupa: ***admin***

Request body

```
{  
  "name": "string",  
  "nameEng": "string"  
}
```

DELETE </catalogs/scientific-artistic-fields/{id}>

Brisanje polja

Prava pristupa: ***admin***

GET </catalogs/scientific-artistic-fields>

Čitanje svih polja

Prava pristupa: ***admin, all***

POST </catalogs/scientific-artistic-fields>

Kreiranje novog polja

Prava pristupa: ***admin***

Request body

```
{  
  "name": "string",  
  "nameEng": "string"  
}
```

GET </catalogs/languages/{id}>

Čitanje podataka o jeziku

Prava pristupa: ***admin, all***

PUT </catalogs/languages/{id}>

Izmena osnovnih podataka o jeziku

Prava pristupa: **admin**

Request body

```
{  
  "name": "string",  
  "nameEng": "string",  
  "isoCode": "string"  
}
```

DELETE </catalogs/languages/{id}>

Brisanje jezika

Prava pristupa: **admi**

GET </catalogs/languages>

Čitanje svih jezika

Prava pristupa: **admin, all**

POST </catalogs/languages>

Kreiranje novog jezika

Prava pristupa: **admin**

Request body

```
{  
  "name": "string",  
  "nameEng": "string",  
  "isoCode": "string"  
}
```

GET </catalogs/institution-functions/{id}>

Čitanje podataka o funkciji

Prava pristupa: **admin, all**

PUT </catalogs/institution-functions/{id}>

Izmena osnovnih podataka o funkciji

Prava pristupa: **admin**

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "nameEng": "string"  
}
```

DELETE </catalogs/institution-functions/{id}>

Brisanje funkcije

Prava pristupa: **admin**

GET </catalogs/institution-functions>

Čitanje svih funkcija

Prava pristupa: **admin, all**

POST </catalogs/institution-functions>

Kreiranje funkcije

Prava pristupa: **admin**

Request body

```
{  
  "code": "string",  
  "name": "string",  
  "nameEng": "string"  
}
```

GET </catalogs/he-institutions/{id}>

Čitanje podataka o instituciji

Prava pristupa: **admin, all**

PUT [/catalogs/he-institutions/{id}](#)

Izmena osnovnih podataka

Prava pristupa: **admin**

DELETE [/catalogs/he-institutions/{id}](#)

Brisanje ustanove

Prava pristupa: **admin**

GET [/catalogs/he-institutions](#)

Čitanje svih ustanova

Prava pristupa: **admin, all**

POST [/catalogs/he-institutions](#)

Kreiranje ustanove

Prava pristupa: **admin**

Request body

```
{  
  "name": "string",  
  "nameEng": "string",  
  "studentAdministrationOfficeEmail": "string",  
  "shortName": "string",  
  "address": "string",  
  "email": "string",  
  "pib": "string",  
  "mb": "string",  
  "place": "string",  
  "color": "string",  
  "institutionType": "faculty"  
}
```

GET </catalogs/documents-for-download/{id}>

Čitanje podataka o jednom dokumentu

Prava pristupa: **admin, all**

PUT </catalogs/documents-for-download/{id}>

Izmena osnovnih podataka o dokumentu

Prava pristupa: **admin**

DELETE </catalogs/documents-for-download/{id}>

Brisanje dokumenta

Prava pristupa: **admin**

GET </catalogs/documents-for-download>

Čitanje svih dokumenata

Prava pristupa: **admin, all**

POST </catalogs/documents-for-download>

Kreiranje novog dokumenta

Prava pristupa: **admin**

Request body

```
{  
  "name": "string",  
  "nameEng": "string",  
  "description": "string",  
  "descriptionEng": "string",  
  "active": true  
}
```

GET </catalogs/documents-for-download/{id}/content>

Čitanje sadržaja dokumenta

Prava pristupa: **admin, all**

POST </catalogs/documents-for-download/{id}/content>

Upload sadržaja dokumenta

Prava pristupa: **admin**

DELETE </catalogs/documents-for-download/{id}/content>

Brisanje sadržaja dokumenta

Prava pristupa: **admin**

Ignoriše se ako sadržaj ne postoji u bazi podataka.

GET </catalogs/countries/{id}>

Čitanje podataka o državi

Prava pristupa: **admin, all**

PUT </catalogs/countries/{id}>

Izmena osnovnih podataka države

Prava pristupa: **admin**

Request body

```
{  
  "continentId": 0,  
  "name": "string",  
  "nameEng": "string",  
  "iso31662": "string",  
  "iso31663": "string",  
  "capital": "string"  
}
```

DELETE </catalogs/countries/{id}>

Brisanje države

Prava pristupa: **admin**

GET </catalogs/countries>

Čitanje svih država

Prava pristupa: **admin, all**

POST </catalogs/countries>

Kreiranje nove države

Prava pristupa: **admin**

GET </catalogs/academic-titles/{id}>

Čitanje podataka o zvanju

Prava pristupa: admin, all

PUT </catalogs/academic-titles/{id}>

Izmena osnovnih podataka o zvanju

Prava pristupa: **admin**

DELETE </catalogs/academic-titles/{id}>

Brisanje zvanja

Prava pristupa: **admin**

GET </catalogs/academic-titles>

Čitanje svih zvanja

Prava pristupa: **admin, all**

POST </catalogs/academic-titles>

Kreiranje novog zvanja

Prava pristupa: **admin**

GET </catalogs/board-positions/{id}>

Čitanje podataka o poziciji

Prava pristupa: **admin, all**

PUT </catalogs/board-positions/{id}>

Izmena osnovnih podataka o poziciji

Prava pristupa: **admin**

DELETE </catalogs/board-positions/{id}>

Brisanje pozicije

Prava pristupa: **admin**

GET </catalogs/board-positions>

Čitanje svih pozicija

Prava pristupa: **admin, all**

POST </catalogs/board-positions>

Kreiranje nove pozicije

Prava pristupa: **admin**

Paket Participanti

PUT </participants/{participantId}/candidate>

Izmena personalnih podataka kandidata

Prava pristupa: **candidate, clerk, supervisor, agency, agent, admin**

Izmenom imejl adrese na novu vrednost, takođe se ažurira i korisničko ime korisničkog naloga na novu imejl adresu.

Request body

```
{  
  "firstName": "string",  
  "lastName": "string",  
  "citizenshipId": 0,  
  "documentDeliveryCountryId": 0,  
  "countryOfBirthId": 0,  
  "residenceCountryId": 0,  
  "parentsName": "string",  
  "maidenName": "string",  
  "dateOfBirth": "2025-12-14T10:30:57.324Z",
```

```
"placeOfBirth": "string",
"gender": "male",
"residenceAddress": "string",
"residenceCity": "string",
"residencePostalCode": "string",
"documentDeliveryAddress": "string",
"documentDeliveryCity": "string",
"documentDeliveryPostalCode": "string",
"contactPhone": "string",
"email": "string"
}
```

PUT [/participants/{participantId}/agent](#)

Izmena personalnih podataka agenta (fizičko lice punomoćnik)

Prava pristupa: **clerk, supervisor, agent, admin**

Izmenom imejl adrese na novu vrednost, takođe se ažurira i korisničko ime korisničkog naloga na novu imejl adresu.

Request body

```
{
  "firstName": "string",
  "lastName": "string",
  "officeCountryId": 0,
  "deliveryDocumentCountryId": 0,
  "address": "string",
  "city": "string",
  "postalCode": "string",
  "documentDeliveryAddress": "string",
  "documentDeliveryCity": "string",
  "documentDeliveryPostalCode": "string",
  "contactPhone": "string",
  "uniqueMasterCitizenNumber": "string",
  "email": "string"
}
```

}

PUT [/participants/{participantId}/agency](#)

Izmena personalnih podataka agencije (pravno lice)

Prava pristupa: **clerk, supervisor, agency, admin**

Izmenom imejl adrese na novu vrednost, takođe se ažurira i korisničko ime korisničkog naloga na novu imejl adresu.

Request body

```
{
  "officeCountryId": 0,
  "deliveryDocumentCountryId": 0,
  "name": "string",
  "taxIdentificationNumber": "string",
  "address": "string",
  "city": "string",
  "postalCode": "string",
  "documentDeliveryAddress": "string",
  "documentDeliveryCity": "string",
  "documentDeliveryPostalCode": "string",
  "contactPhone": "string",
  "email": "string"
}
```

GET [/participants/{participantId}/picture](#)

Čitanje slike participanta

Prava pristupa: **admin, al**

POST [/participants/{participantId}/picture](#)

Upload slike participanta

Prava pristupa: **candidate, clerk, supervisor, agency, agent, academic_board, admin**

Naziv multipart parametra za datoteku: file

DELETE </participants/{participantId}/picture>

Brisanje slike participanta

Prava pristupa: ***candidate, clerk, supervisor, agency, agent, academic_board, admin***

Ignoriše se ako slika ne postoji u bazi podataka.

GET </participants/{participantId}>

Čitanje personalnih podataka participanta

Prava pristupa: ***candidate, clerk, supervisor, agency, agent, academic_board, admin***

Keširanje i sesije

Redis kao jedini keš mehanizam.

Cacheable sadržaji: reference tabele (države, univerziteti iz spiska), rezultati složenih upita, privremeni upload tokovi.

Session management: pošto se koristi JWT, Redis je potreban samo za blacklisting/refreshtoken store; u alternativnom session-based pristupu Redis čuva session objekte.

Asinhroni mehanizmi i Kafka workflow

Kafka topics su centralni za pouzdanu obradu i retry logiku.

Poruke treba da sadrže `request\_id`, `event\_type`, `timestamp`, `trace\_id`.

Retry mehanizam: koristite dead-letter topics (DLQ) i backoff politiku (exponential backoff); retrigger može biti ad-hoc ili automatizovan.

Potencijalne integracije: slanje email-a (SMTP/SendGrid), SMS gateway, povezivanje sa državnim registrima.

Observability (monitoring, logging, alerting)

Metrike (Prometheus)

Po endpoint-u: request_count, request_latency_seconds (histogram), error_count

DB: connection_count, slow_queries

Kafka: consumer lag, producer success/failure

Redis: hits, misses, memory_usage

Logovi (Loki)

Strukturisani logovi (JSON) da bi se lako parsirali i filtrirali.

Log labels: `app`, `env`, `instance`, `request\_id`, `user\_id`.

Grafana dashboards i alerts

Health overview: uptime, 5xx rate

Performance: p95/p99 latencije

Kafka: consumer lag thresholds alerts

DB: lock / slow queries alert

Sigurnost i autentikacija

Koristi se HTTPS sa modernim TLS konfiguracijama.

Autentikacija: JWT + Refresh token.

Autorizacija: role-based access control (RBAC) — minimalni pristup princip.

Input validation i sanitization pri upload-u fajlova (content-type, max size, antivirus scan)

Enkripcija osetljivih podataka u bazi (PII) — na nivou kolone ili disk-level E2E enkripcija.

Audit logging za sve promene statusa (ko, kada, šta).

Rate limiting na API gateway nivou (Nginx/Ingress controller/Cloud gateway).

DevOps — deployment i operacije

Okruženja

`dev`, `staging`, `prod` — odvojene baze i Kafka namespace/cluster (ili topic prefix)

CI/CD

Build pipeline: unit tests -> integration tests -> docker image build -> push registry -> deploy to staging -> e2e tests -> deploy to prod

Alati: GitLab CI/GitHub Actions/Jenkins

Containerization i orkestracija

Docker images za backend i eventualne mikroservise.

Kubernetes za orkestraciju (Deployment, Service, Ingress, ConfigMap, Secret)

Use readiness/liveness probes

Backup i DR

Redovan backup SQL Server baze, log shipping ili full backups prema RTO/RPO zahtevima

Kafka replication i retention policies

Redis persistence (RDB/AOF) i replicas

Skaliranje i performanse

Vertikalno skaliranje baze (jači instance) i horizontalno skaliranje aplikacionih podova (replica set u K8s)

Stateless backend instanci—može se horizontalno širiti bez komplikacija

Kafka i Redis su skalabilni horizontalno (sharding, partitioning)

Profilisanje uskih grla: DB indeksi, optimizacija upita, reduce N+1 problems

Testiranje i kvalitet

Unit testovi za servisne i repository slojeve

Integration testovi za DB i Redis integracije (može se koristiti testcontainers)

End-to-end testovi za kritične tokove (Cypress / Playwright)

Load testing: JMeter / Gatling za simulaciju concurrent user loads i proveru Kafka throughput

Security scans: SAST (SonarQube), Dependency vulnerability scans (Snyk/OWASP)

Rizični faktori i predlozi za unapređenje

Rizični faktori

Loša konfiguracija Kafka consumera (npr. neadekvatno handling commit-a) može dovesti do gubitka poruka ili duplih poruka.

Nedovoljno indeksa u bazi → povećane latencije.

Manjak monitoring-a oko DLQ i retry-a — poruke mogu ostati nerađene.

Neadekvatno obezbeđenje upload-ovanih fajlova (malware)

Potencijalno unapređenje

Implementirati end-to-end tracing (OpenTelemetry) i povezati trace-ove u Grafana/Loki.

Postaviti ground rules za Kafka topic retention i partitioning (npr. 3 partition-a za paralelizam, retention prema zakonskim zahtevima).

Automatizovati rotaciju i backup logova.

Obraditi GDPR i lokalne zakonske zahteve oko čuvanja ličnih podataka.

APPENDIX II: Conceptual and Physical Data Model of the System for Digitalization of the Recognition Process

ERASMUS+ PROJECT 2023-1-RS01-KA220-HED-000156660

EPIR | E-Procedure of Institutional Recognition of Foreign Higher Education Documents

Dokument:	Konceptualni i fizički model strukture podataka sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava
Verzija:	3.0
Dokument pripremili:	Danijel Venus Igor Zečević Ljubomir Milašinović Rade Lučić

EPIR project partners:



UNIVERSITY
OF NOVI SAD



UNIVERSITÀ
POLITECNICA
DELLE MARCHE



ROMANIA
1 DECEMBRIE 1918
UNIVERSITY OF ALBA IULIA



50 years of
University
of Split



REPUBLIC OF SERBIA
Qualifications Agency

APSTRAKT

Cilj digitalizacije procesa priznavanja stranih visokoškolskih isprava koju obavlja kancelarija za pravne poslove Univerziteta u Novom Sadu je omogućavanje kandidatima online podnošenja zahteva za priznavanje i pratećih dokumenata, kao i povećanje efikasnosti rada.

Na osnovu prethodno obavljene analize sistema, formiranja specifikacije zahteva i kreiranja uzorka dizajna korisničkog interfejsa (mockup) softverskog sistema, pristupilo se projektovanju konceptualnog i fizičkog modela podataka koji će obezbediti online podnošenje zahteva za priznavanje, jednostavnije praćenje procesa priznavanja za sve učesnike, efikasnije izveštavanje za potrebe institucije i nadležnih organa, i smanjenje broja dokumenata u papirnom obliku.

Treća verzija konceptualnog i fizičkog modela strukture podataka nastala je nakon uvođenja beta verzije sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava. U toku korišćenja ove verzije softvera primećen je određeni broj nedostataka sistema ali su uspostavljeni i novi korisnički zahtevi koji nisu bili obuhvaćeni inicijalnom verzijom specifikacije zahteva.

Konceptualni model predstavljao je prvi korak u projektovanju baze podataka i obuhvatio je identifikaciju svih osnovnih entiteta, njihovih atributa i međusobnih veza. Na ovom nivou definisana je poslovna logika sistema i način na koji se podaci povezuju u okviru procesa priznavanja stranih visokoškolskih isprava. Na osnovu konceptualnog modela izrađen je fizički model, koji je prilagođen konkretnom sistemu za upravljanje bazom podataka planiranom za implementaciju. U ovoj fazi su definisani tipovi podataka, primarni i strani ključevi, kao i ograničenja integriteta (constraints) koja obezbeđuju konzistentnost i tačnost podataka u sistemu.

ABSTRACT

The goal of the digitalization of the foreign higher education credential recognition process carried out by the Legal Affairs Office of the University of Novi Sad is to enable applicants to submit recognition requests and supporting documents online, while also increasing operational efficiency.

Based on the previously conducted system analysis, the development of the requirements specification, and the creation of a user interface design prototype (mockup) for the software system, the design of the conceptual and physical data models was initiated. These models provide support for online submission of recognition requests, easier monitoring of the recognition process by all stakeholders, more efficient reporting for institutional and governmental purposes, and a reduction in the use of paper-based documentation.

The third version of the conceptual and physical data model was developed following the introduction of the beta version of the system for the digitalization of the foreign higher education credential recognition process. During the use of this software version, a number of system deficiencies were identified, and new user requirements emerged that had not been included in the initial requirements specification.

The conceptual model represented the first step in the database design process and included the identification of all key entities, their attributes, and the relationships among them. At this level, the business logic of the system was defined, as well as the manner in which data are interconnected within the foreign higher education credential recognition process.

Based on the conceptual model, a physical model was developed and adapted to the specific database management system planned for implementation. In this phase, data types, primary and foreign keys, and integrity constraints were defined to ensure data consistency and accuracy throughout the system.

Sadržaj

Opseg sistema	6
Namena koceptualnog i fizičkog modela podataka	7
Konceptualni model podataka	9
Paket Catalogs.....	9
Paket Participants	10
Paket Security	12
Paket Submissions.....	12
Paket Notifications.....	18
Paket FAQ.....	19
Fizički model podataka.....	20
Paket Catalogs.....	20
Paket Participants	21
Paket Security	23
Paket Submissions.....	24
Paket Notifications.....	29
Paket FAQ.....	30
SQL Skripte	31

Opseg sistema

Kancelarija za pravne poslove Univerziteta u Novom Sadu obavlja poslove priznavanja stranih visokoškolskih isprava za potrebe nastavka školovanja na Univerzitetu u Novom Sadu. Ova aktivnost predstavlja važan segment međunarodne saradnje i mobilnosti studenata, jer omogućava kandidatima iz drugih država da nastave svoje obrazovanje u okviru Univerziteta, u skladu sa važećim propisima i standardima.

Trenutno, proces priznavanja stranih visokoškolskih isprava nije digitalizovan, već se većina dokumenata i komunikacije odvija u papirnom obliku. Kandidati zahteve podnose lično ili putem pošte, dok zaposleni u Kancelariji za pravne poslove u upravljanju procesom koriste imejl komunikaciju i standardne kancelarijske softvere (tekstualne obrade, tabele, baze u ograničenom obliku). Ne postoji namensko softversko rešenje koje bi objedinjavalo sve aktivnosti i omogućilo integrisano praćenje toka priznavanja — od prijema zahteva do izdavanja odluke.

Digitalizacija ovog procesa ima za cilj da značajno unapredi način rada i iskustvo korisnika. Novi softverski sistem omogućiće kandidatima da elektronski podnesu zahtev za priznavanje i prate njegov status u realnom vremenu, dok će Kancelariji za pravne poslove i drugim organizacionim jedinicama Univerziteta obezbediti centralizovan, pregledan i efikasan način vođenja postupka.

Očekivani efekti uvođenja sistema su:

- Povećanje efikasnosti rada kroz automatizaciju administrativnih koraka i smanjenje manuelnog unosa podataka,
- Ubrzanje razmene informacija između svih učesnika u procesu,
- Smanjenje upotrebe papira i prelazak na elektronsku dokumentaciju,
- Poboljšano izveštavanje i jednostavnije generisanje statističkih podataka za potrebe Univerziteta i nadležnih organa, kao i povećanje transparentnosti i pristupačnosti procesa priznavanja za kandidate iz inostranstva.

Dodatno, očekuje se da digitalizacija ovog postupka doprinese jačanju međunarodne vidljivosti Univerziteta u Novom Sadu i povećanju broja stranih studenata zainteresovanih za nastavak studija na Univerzitetu, čime će se unaprediti i proces akademske mobilnosti i međunarodne saradnje.

Namena koceptualnog i fizičkog modela podataka

Modelovanje podataka predstavlja jednu od ključnih faza u razvoju informacionog sistema, jer obezbeđuje pravilno razumevanje i strukturisanje informacija koje sistem obrađuje. U ovom projektu izrađeni su konceptualni i fizički model podataka, a zatim i SQL skripte za kreiranje baze podataka. Ovi modeli čine osnovu stabilne, dosledne i lako održive baze podataka koja će podržavati poslovne procese sistema.

Konceptualni model podataka (Conceptual Data Model) predstavlja apstraktni prikaz strukture informacija koje sistem koristi. Njegova glavna svrha je da prikaže šta sistem sadrži, a ne kako se to tehnički implementira.

Ovaj model obuhvata:

- identifikaciju entiteta (npr. kandidat, dokument, zahtev, odluka),
- definisanje njihovih atributa (osobina ili karakteristika svakog entiteta),
- i odnosa (relacija) između entiteta, kao što su „kandidat podnosi zahtev“ ili „zahtev sadrži dokumente“.

Konceptualni model se izrađuje na osnovu analize poslovnih procesa i potreba korisnika, najčešće u saradnji sa stručnjacima iz oblasti rada sistema. On služi kao komunikacioni most između poslovnih korisnika i tehničkog tima, jer prikazuje strukturu podataka u razumljivom i vizuelnom obliku, nezavisno od bilo koje tehnologije ili baze.

Prednosti konceptualnog modela:

- Omogućava jasno razumevanje poslovnih procesa i podataka koji su važni za funkcionisanje sistema.
- Pomaže u ranoj identifikaciji nedostataka i nelogičnosti u strukturi podataka.
- Ubrzava komunikaciju između analitičara, korisnika i programera.
- Služi kao osnova za izradu logičkog i fizičkog modela, čime se obezbeđuje kontinuitet u razvoju.

Fizički model podataka (Physical Data Model) predstavlja tehničku razradu konceptualnog modela. Dok konceptualni model opisuje šta sistem sadrži, fizički model definiše kako će ti podaci biti organizovani i sačuvani u konkretnom sistemu za upravljanje bazom podataka (npr. SQL Server, MySQL, PostgreSQL).

Ovaj model obuhvata:

- definisanje tipova podataka za svako polje,
- određivanje primarnih i stranih ključeva,
- definisanje relacija i ograničenja integriteta (constraints),
- specifikaciju indeksa, view-ova, triggera i drugih elemenata koji obezbeđuju performanse i tačnost podataka.

Fizički model je osnova za kreiranje baze i omogućava automatsko generisanje SQL skripti koje implementiraju strukturu baze u konkretnom DBMS-u.

Prednosti fizičkog modela:

- Omogućava tačnu i doslednu implementaciju baze podataka u izabranom DBMS-u.
- Smanjuje mogućnost greške prilikom ručnog unosa struktura.
- Obezbeđuje visok nivo integriteta i konzistentnosti podataka.

- Omogućava optimizaciju performansi baze kroz indeksiranje i pravilno definisane relacije.
- Pruža osnovu za kasniju migraciju ili skaliranje baze ukoliko sistem preraste početne kapacitete.

Na osnovu fizičkog modela generišu se SQL skripte (Structured Query Language scripts), koje sadrže precizne naredbe za kreiranje tabela, relacija, ograničenja i drugih objekata baze podataka. Skripte omogućavaju da se cela baza podataka automatski i reproduktivno kreira u bilo kojem okruženju, što je ključno za timski rad i kasniji razvoj sistema.

Prednosti SQL skripti:

- Omogućavaju brzu i konzistentnu implementaciju baze u različitim okruženjima (razvoj, test, produkcija).
- Pomažu u verzionisanju baze podataka, jer se svaka izmena može pratiti kroz skripte.
- Obezbeđuju lakšu replikaciju, testiranje i održavanje sistema.
- Služe kao dokaz i dokumentacioni trag o strukturi baze i njenim promenama.

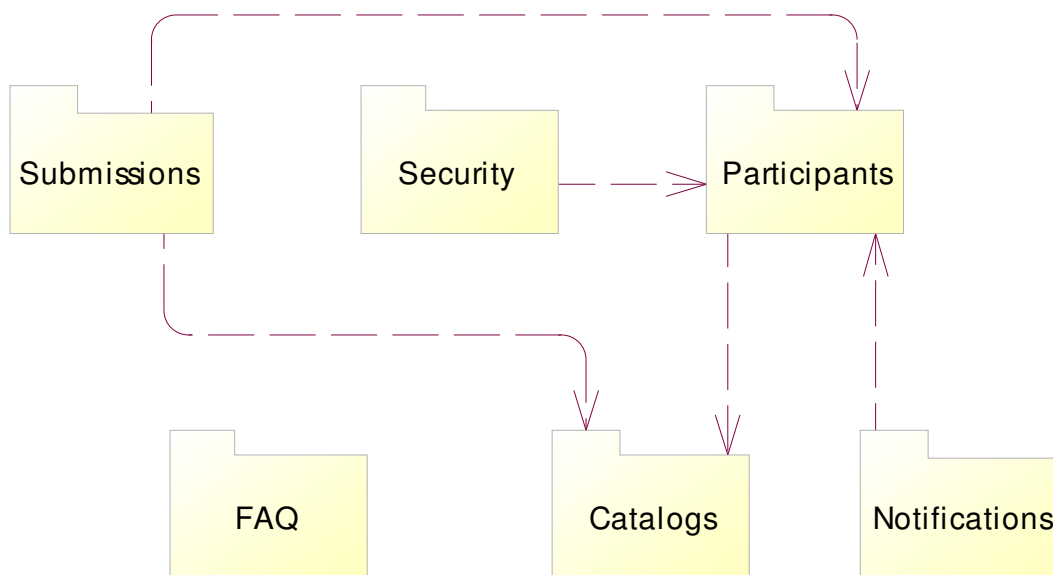
Zajedničkom upotrebom konceptualnog modela, fizičkog modela i SQL skripti obezbeđuje se kontinuiran i kontrolisan prelaz od poslovne analize do tehničke implementacije baze podataka.

Ovakav pristup donosi visok nivo preciznosti, stabilnosti i održivosti informacionog sistema, omogućava brže otklanjanje grešaka i pojednostavljuje buduće nadogradnje sistema.

Konceptualni model podataka

Konceptualni model podataka (Conceptual Data Model) predstavlja apstraktni prikaz strukture informacija koje sistem koristi. Njegova glavna svrha je da prikaže šta sistem sadrži, a ne kako se to tehnički implementira.

Konceptualni model podataka za EPIR sistem je paketiran u skladu sa pojedinačnim modulima sistema. Na slici ispod prikazan je odnos pojedinačnih paketa kompletnog sistema.



Paket **Catalogs** sadrži osnovne šifarnike sistema. U paketu **Participants** nalaze se entiteti koji predstavljaju sve učesnike sistema i njihove uloge. U paketu **Submissions** smešteni su entiteti i relacije kojima su opisani detalji zahteva, vezanim dokumentima i podrška radu stručnih veća. Paket **Notifications** sadrži entitete koji treba da obezbede podršku potrebnu za internu razmenu podataka između učesnika sistema. Paket **Security** sadrži entitete i njihove međusobne relacije koje trebaju da obezbede podršku za opis korisničkih naloga i uloga u sistemu. Paket **FAQ** je novi paket u verziji 3 i sadrži samo jedan entitet koji obezbeđuje podršku za Često postavljana pitanja.

Paket Catalogs

Paket **Catalogs** sadrži osnovne šifarnike sistema.

Na slici ispod ilustrovan je sadržaj dijagrama „Catalogs“.

App Config			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Max number of requests per candidate		Long integer	<M>
Default max number of documents		Long integer	<M>
Fixed file number		Variable characters (8000)	<M>
Enic Naric email		Variable characters (8000)	
Identifier_1	<pi>		

Study Level : 1			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Degree		Long integer	<M>
Level		Long integer	<M>
Name		Variable characters (8000)	
Name Eng		Variable characters (8000)	
Short Name		Variable characters (8000)	
Color		Variable characters (8000)	
Min ECTS credits		Integer	
Identifier_1	<pi>		

Submission Status			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Code		Variable characters (200)	<M>
Name		Variable characters (8000)	<M>
Description		Variable characters (8000)	
Name Eng		Variable characters (8000)	
Description Eng		Variable characters (8000)	
Color		Variable characters (8000)	
Status Order		Long integer	<M>
Notification trigger days		Long integer	<M>
Deadline days		Long integer	
Identifier_1	<pi>		

Language			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name Eng		Variable characters (8000)	
ISO Code		Variable characters (200)	<M>
Identifier_1	<pi>		

Scientific Artistic Field : 1			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name Eng		Variable characters (8000)	
Identifier_1	<pi>		

Continent			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name Eng		Variable characters (8000)	<M>
Identifier_1	<pi>		

In Continent

Country			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name Eng		Variable characters (8000)	
ISO-3166-2		Variable characters (200)	
ISO-3166-3		Variable characters (200)	
Capital		Variable characters (8000)	
Identifier_1	<pi>		

Metadata Field			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Code		Variable characters (200)	<M>
Name		Variable characters (8000)	<M>
Data Type		Variable characters (8000)	<M>
Description		Variable characters (8000)	
Need for verification?		Boolean	<M>
View on faculty?		Boolean	<M>
Created by		Variable characters (8000)	<M>
Name eng		Variable characters (8000)	
Description eng		Variable characters (8000)	
Level		Variable characters (200)	<M>
Deleted?		Boolean	<M>
Backend attribute		Variable characters (8000)	
Backend attribute relationship		Variable characters (8000)	
Sort order		Long integer	<M>
Max number		Long integer	<M>
Identifier_1	<pi>		

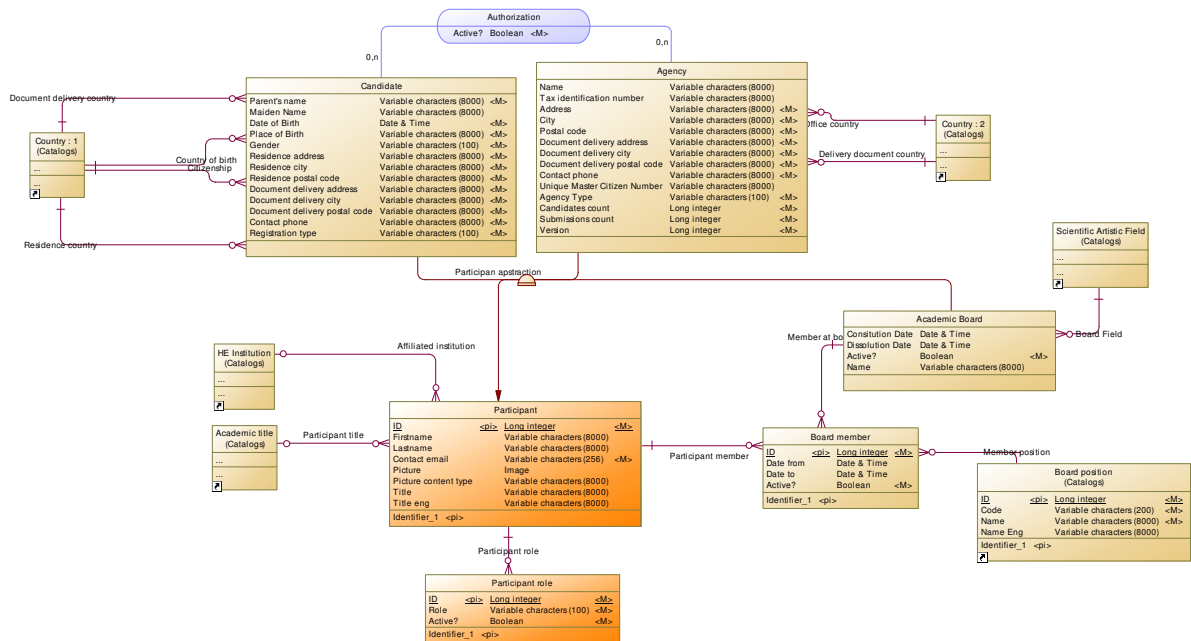
Academic title			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name Eng		Variable characters (8000)	<M>
Identifier_1	<pi>		

U odnosu na drugu verziju konceptualnog modela, u ovom paketu uveden je entitet AppConfig koji predstavlja generalne postavke aplikacije. Takođe, u ovoj verziji modela, entitet Metadata Field je iz paketa Submissions premešten u paket Catalogs.

Paket Participants

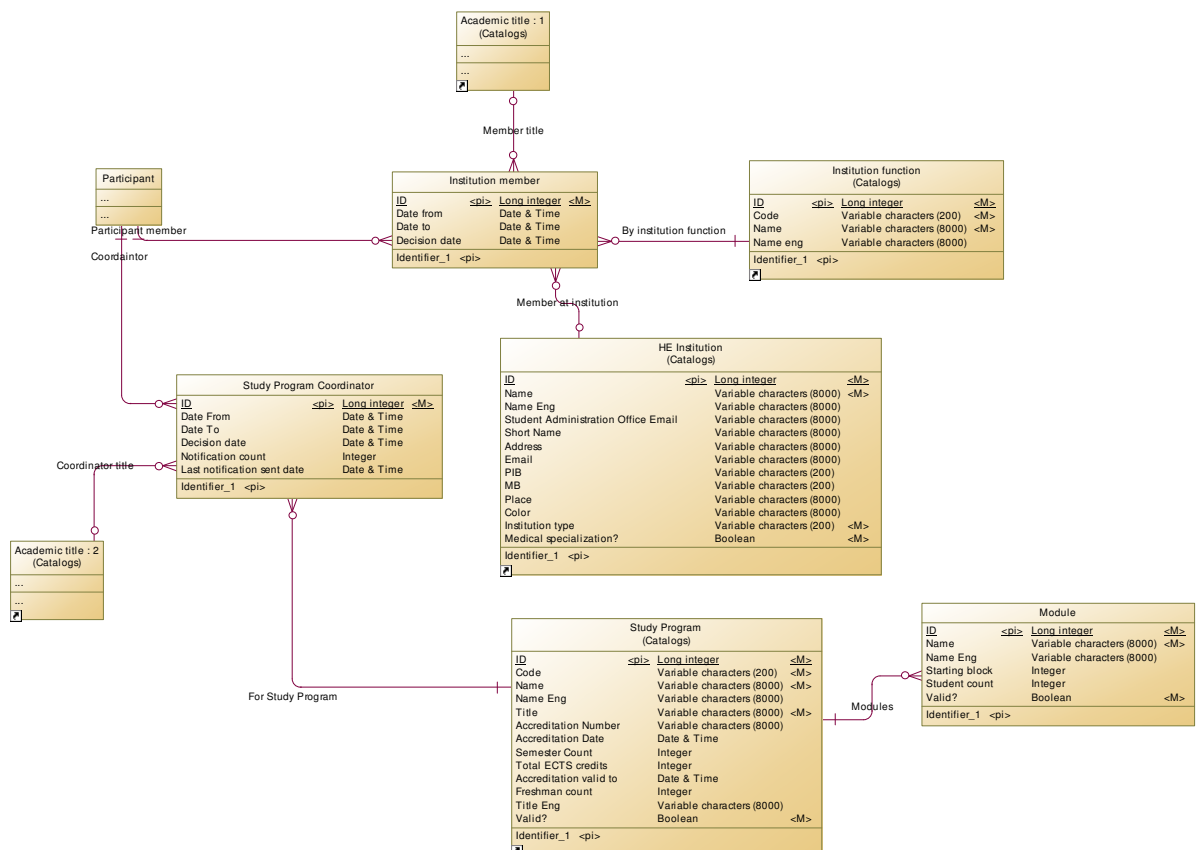
U paketu **Participants** nalaze se entiteti koji predstavljaju učesnike sistema. Paket sadrži dva dijagrama koji dodatno klasifikuju entitete ovog paketa. Dijagram „**Participants**“ sadrži entitete koji opisuju lica, odnosno attribute uloga sistema. Dijagram „**HE Institution**“ sadrži entitete koji se odnose na visokoškolske ustanove.

Na slici ispod ilustrovan je sadržaj dijagrama „Participants“.



U odnosu na drugu verziju, u ovom dijagramu nije bilo značajnijih izmena.

Na slici ispod ilustrovan je sadržaj dijagrama „HE Institution“.

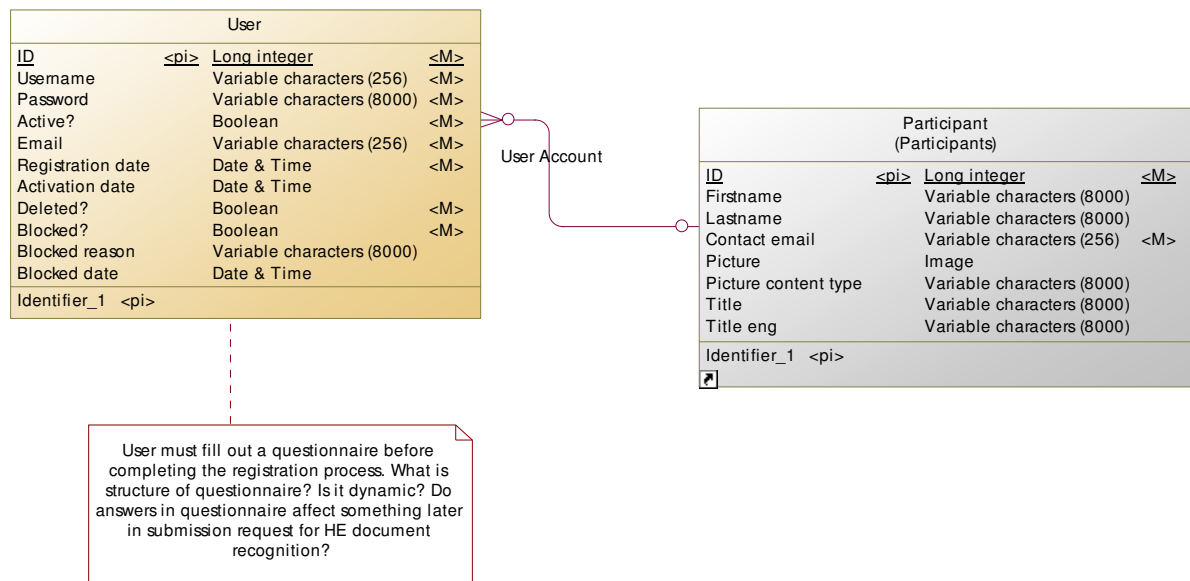


U odnosu na drugu verziju modela, u entitet „HE Institution“ uveden je indikator „Medical specialization?“ kojim se omogućava visokoškolskoj ustanovi podrška za Zdravstvene specijalizacije.

Paket Security

Paket **Security** sadrži entitete i njihove međusobne relacije koje trebaju da obezbede podršku za opis korisničkih naloga i uloga u sistemu.

Na slici ispod ilustrovan je sadržaj dijagrama „Security“.



U odnosu na drugu verziju modela, iz ovog paketa izbačen je entitet „Role“.

Paket Submissions

U paketu **Submissions** smešteni su entiteti i relacije kojima su opisani detalji zahteva. Paket sadrži osam dijagrama (tri više u odnosu na drugu verziju modela) koji dodatno klasifikuju entitete ovog paketa.

Dijagram „**ENIC/NARIC**“ sadrži entitete koji se odnose na bazu upita i dopisa ENIC/NARIC centra.

Dijagram „**Submission request**“ sadrži entitete koji opisuju životni ciklus zahteva.

Dijagram „**Submission data**“ je centralni deo modela koji sadrži entitete samog zahteva kao i podršku za validaciju zahteva.

Dijagram „**Submission snapshot data**“ sadrži entitete koji predstavljaju aktuelni presek zahteva u kome su u obzir uzete i potencijalne izmene istog.

Dijagram „**Documents**“ sadrži entitete koji predstavljaju sve vrste dokumenata koji će biti u relaciji sa zahtevima.

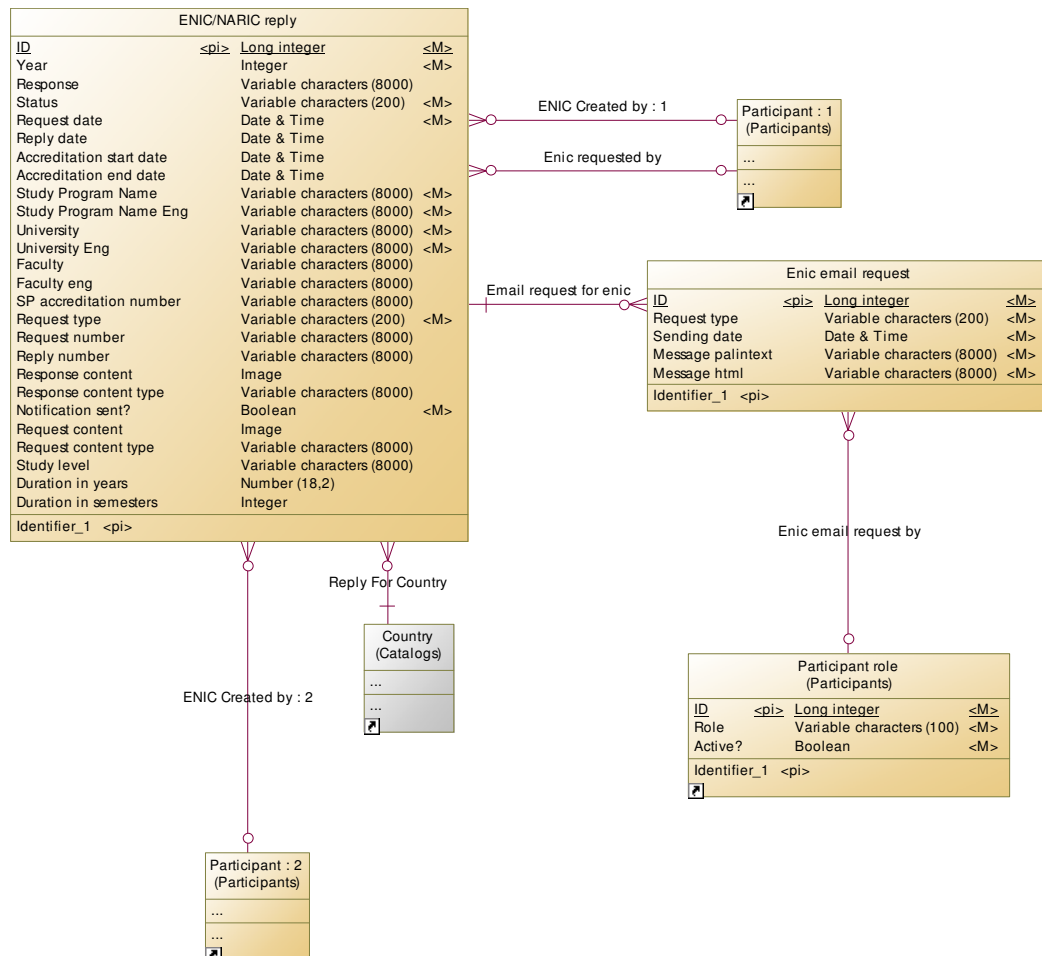
Novi dijagrami paketa Submissions u ovoj verziji modela su:

Dijagram „**Academic Board**“ sadrži entitete kojima je pružena podrška u radu Stručnih veća;

Dijagram „**Additional documents**“ sadrži entitete kojima je pružena podrška u održavanju vezanih dodatnih dokumenata zahteva od strane službenika i supervizora i

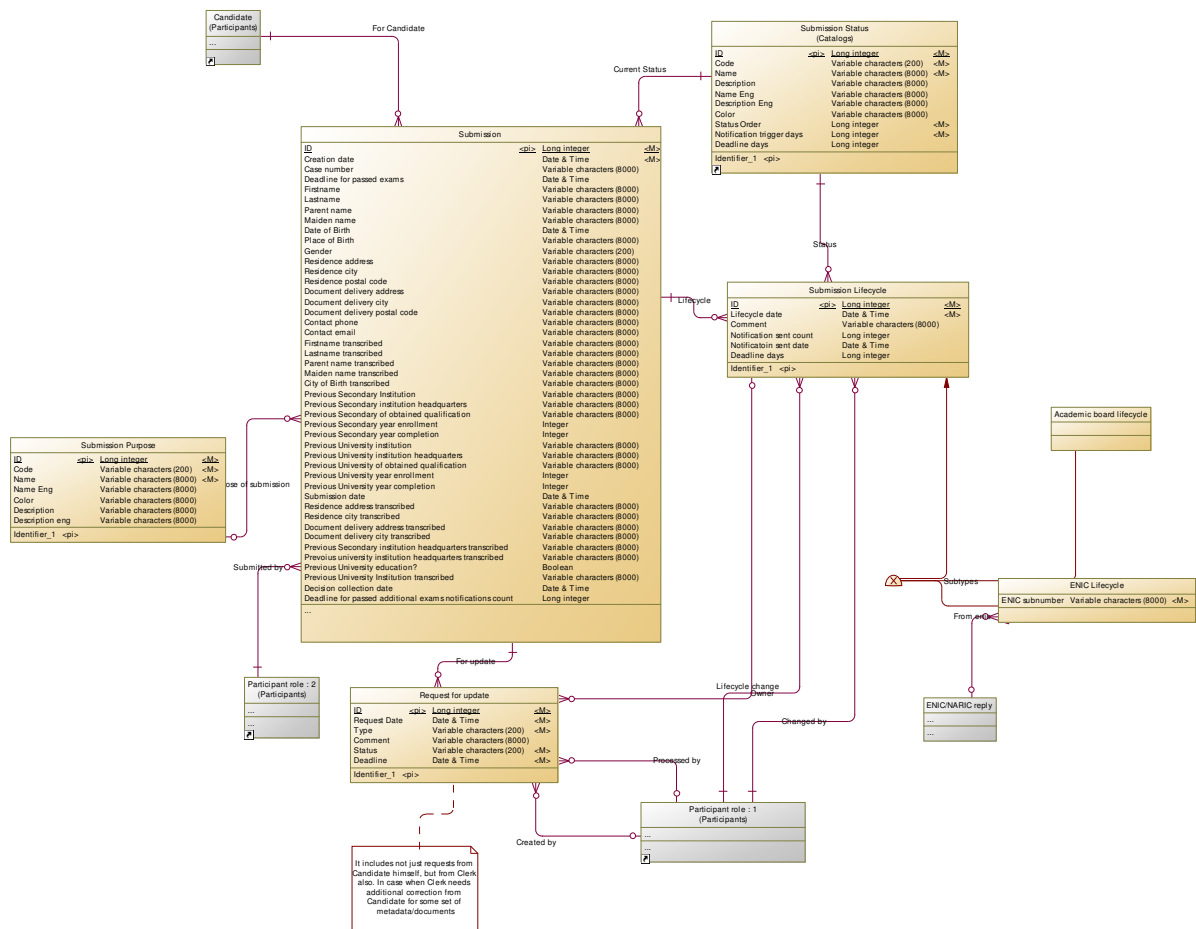
Dijagram „**Case number**“ sadrži samo jedan entitet koji služi za podršku struktuiranom broju dokumenata koji su vezani za jedan zahtev.

Na slici ispod ilustrovan je sadržaj dijagrama „**ENIC/NARIC**“.



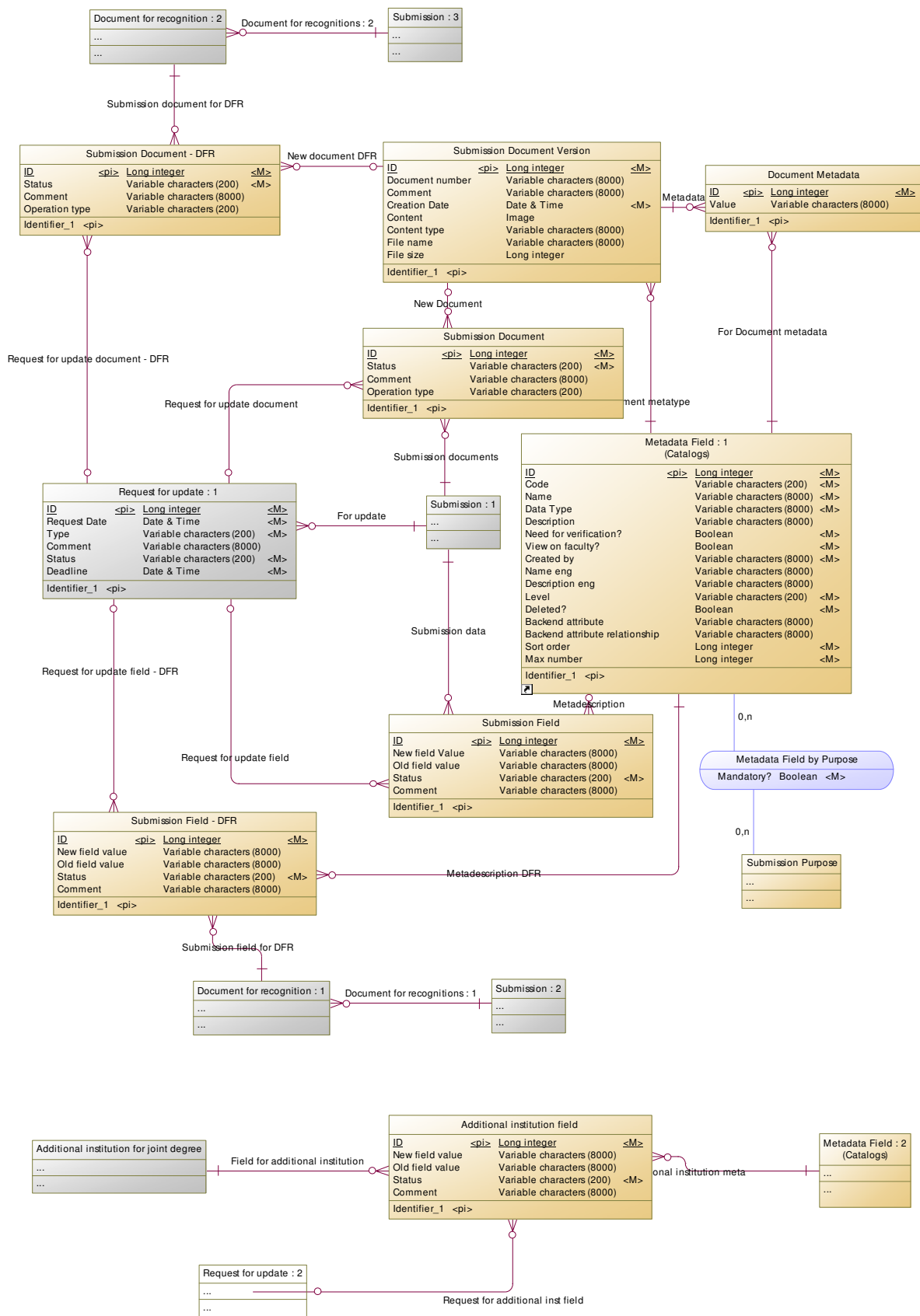
U odnosu na drugu verziju modela, dijagram je dodatno proširen sa novim entitetom koji podržava praćenje istorije slanja mejlova kao i sa dodatnim atributima.

Na slici ispod ilustrovan je sadržaj dijagrama „**Submission request**“.



U odnosu na drugu verziju ovaj dijagram je imao izmene pre svega u novim atributima entiteta Submission kao i u uvođenju relacije ka novom entitetu „Academic board lifecycle“ – odnosno podrška za stručna veća.

Na slici ispod ilustrovan je sadržaj dijagrama „Submission data“.

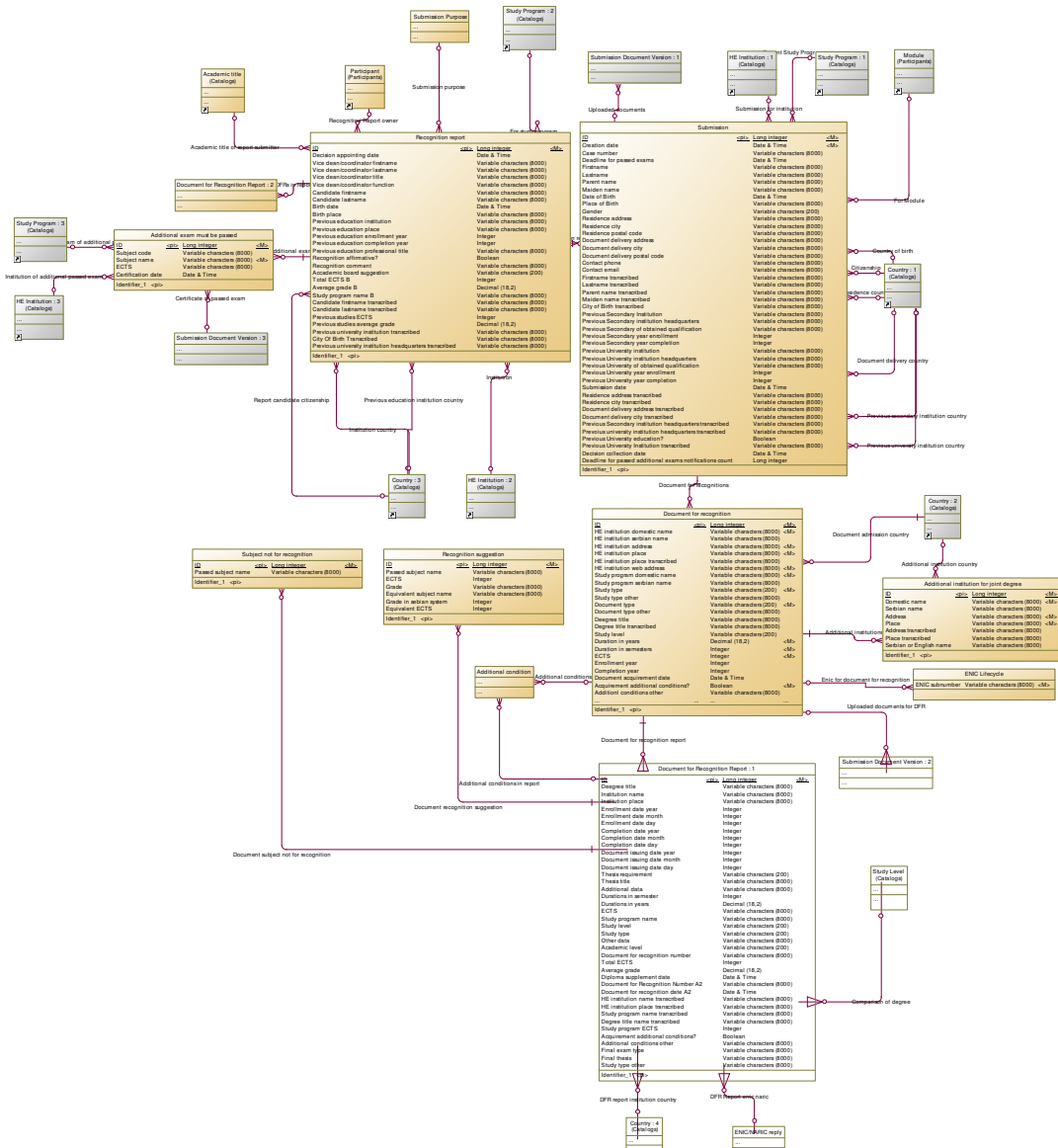


U odnosu na prethodnu verziju, u ovom dijagramu je uveden novi entitet „Submission Field - DFR“ – podrška za validaciju na nivou visokoškolske isprave koja je predmet priznavanja.

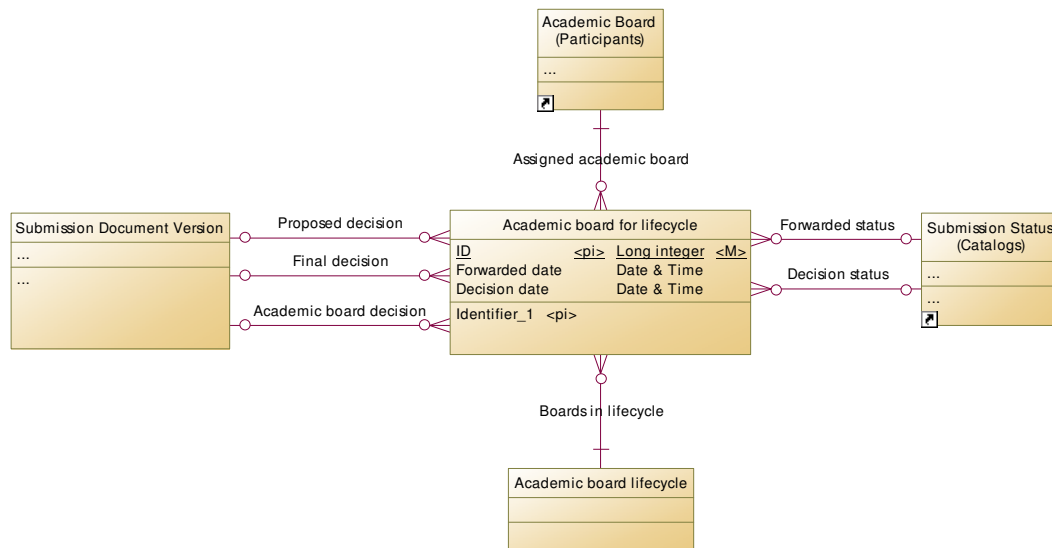
Na slici ispod ilustrovan je sadržaj paketa „Documents“. U odnosu na prethodnu verziju modela, u entitet „Document for download“ uveden je novi atribut „Content type“ čime su podržani svi tipovi dokumenta.

Document for download			
ID	<pi>	Long integer	<M>
Name		Variable characters (8000)	<M>
Name eng		Variable characters (8000)	
Description		Variable characters (8000)	
Description eng		Variable characters (8000)	
Content		Image	
Active?		Boolean	<M>
Content type		Variable characters (8000)	
Identifier_1	<pi>		

Na slici ispod ilustrovan je sadržaj paketa „Submission snapshot data“. U odnosu na prethodnu verziju modela, u entitetima ovog dijagrama postoje izmene u dodatnim atributima.

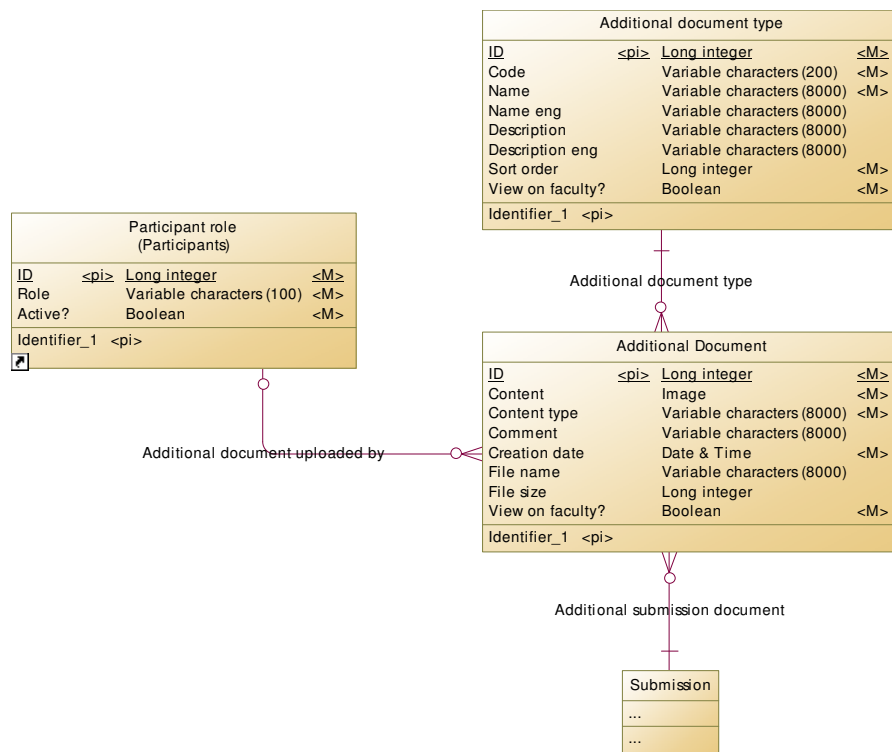


Na slici ispod ilustrovan je sadržaj paketa „**Academic board**“.



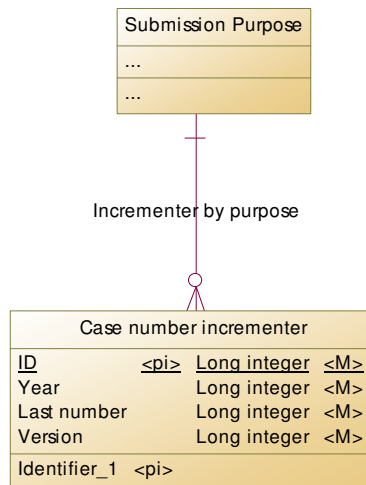
Ovaj dijagram i entiteti u njemu nije postojao u prethodnoj verziji modela.

Na slici ispod ilustrovan je sadržaj paketa „**Additional documents**“.



Ovaj dijagram i entiteti u njemu nije postojao u prethodnoj verziji modela.

Na slici ispod ilustrovan je sadržaj paketa „**Case number**“.

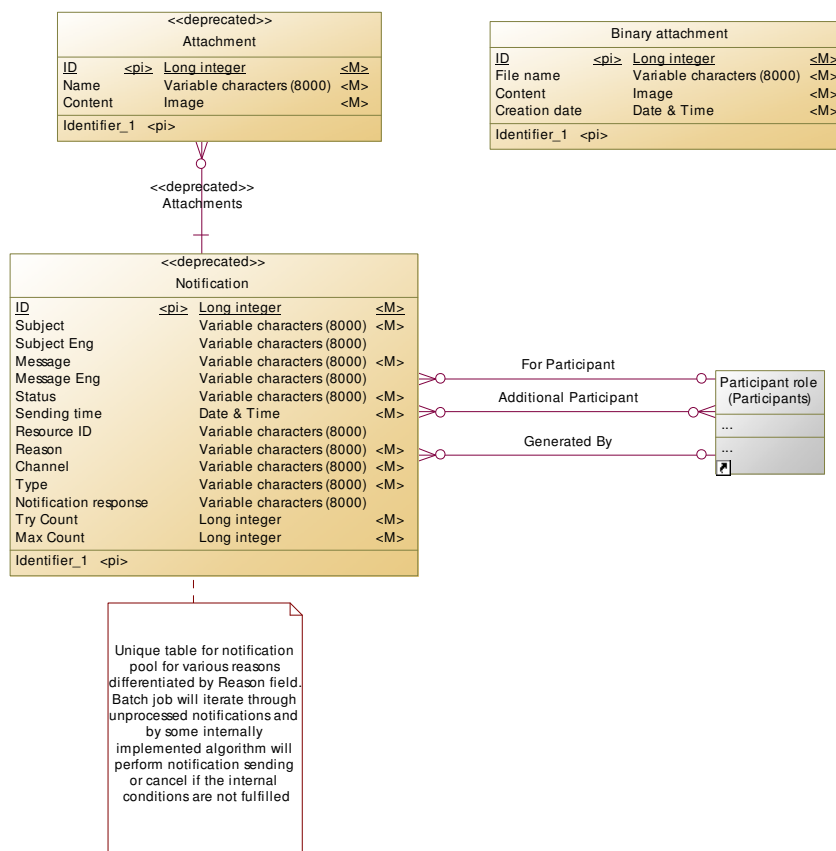


Ovaj dijagram i entitet u njemu nije postojao u prethodnoj verziji modela.

Paket Notifications

Paket „**Notifications**“ sadrži entitete koji treba da obezbede podršku potrebnu za internu razmenu podataka između učesnika sistema.

Na slici ispod ilustrovan je sadržaj dijagrama „Notifications“.



U odnosu na drugu verziju modela ovaj dijagram nije imao značajnije izmene.

Paket FAQ

Paket „**FAQ**“ sadrži entitete koji treba da obezbede podršku za održavanje i prikaz često postavljanih pitanja.

Na slici ispod ilustrovan je sadržaj dijagrama „FAQ“.

FAQ			
<u>ID</u>	<pi>	<u>Long integer</u>	<M>
Name		Variable characters (8000)	<M>
Name eng		Variable characters (8000)	<M>
Question		Variable characters (8000)	<M>
Question eng		Variable characters (8000)	<M>
Answer		Variable characters (8000)	<M>
Answer eng		Variable characters (8000)	<M>
Active?		Boolean	<M>
Identifier_1	<pi>		

U prethodnoj verziji modela ovaj paket nije postojao.

Fizički model podataka

Fizički model podataka (Physical Data Model) predstavlja tehničku razradu konceptualnog modela. Dok konceptualni model opisuje šta sistem sadrži, fizički model definiše kako će ti podaci biti organizovani i sačuvani u konkretnom sistemu za upravljanje bazom podataka.

Za projekat EPIR konkretni sistem za upravljanje bazom podataka biće MS SQL Server.

Fizički model podataka obuhvata:

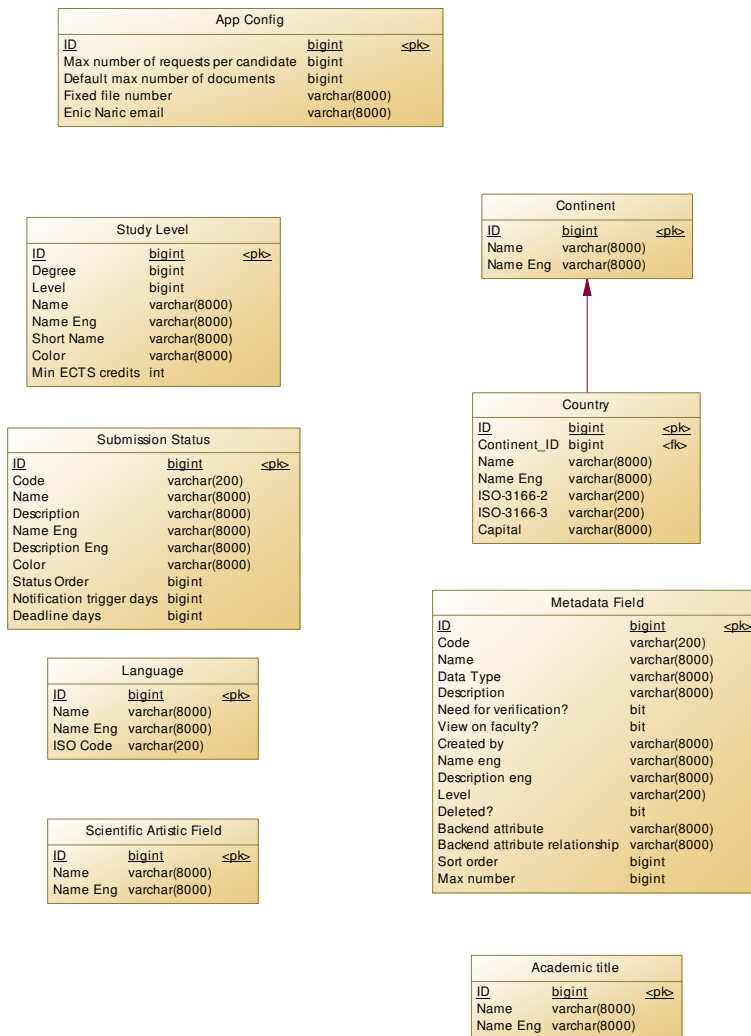
- definisanje tipova podataka za svako polje,
- određivanje primarnih i stranih ključeva,
- definisanje relacija i ograničenja integriteta (constraints),
- specifikaciju indeksa, view-ova, triggera i drugih elemenata koji obezbeđuju performanse i tačnost podataka.

U nastavku su prikazani svi pojedinačni paketi EPIR fizičkog modela podataka.

Paket Catalogs

Paket Catalogs sadrži osnovne šifarnike sistema.

Na slici ispod ilustrovan je sadržaj dijagrama „**Catalogs**“.

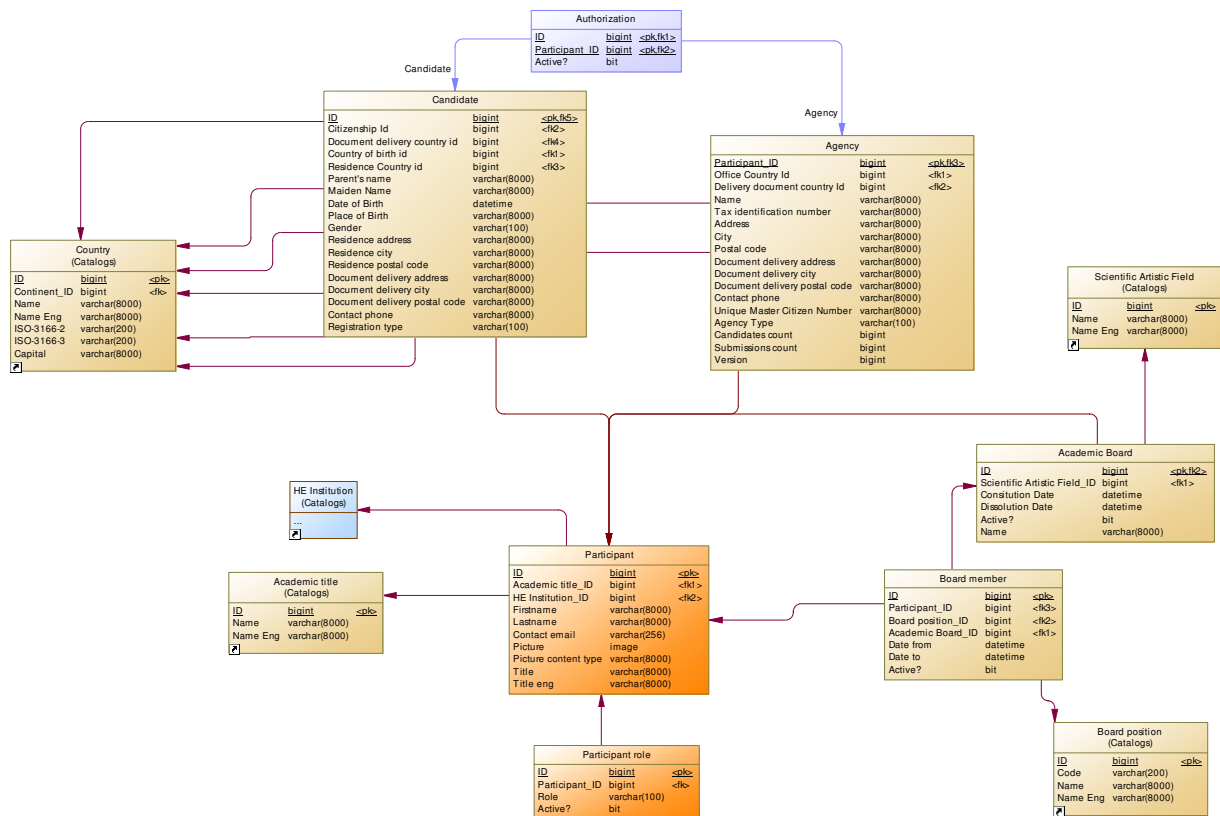


Paket Participants

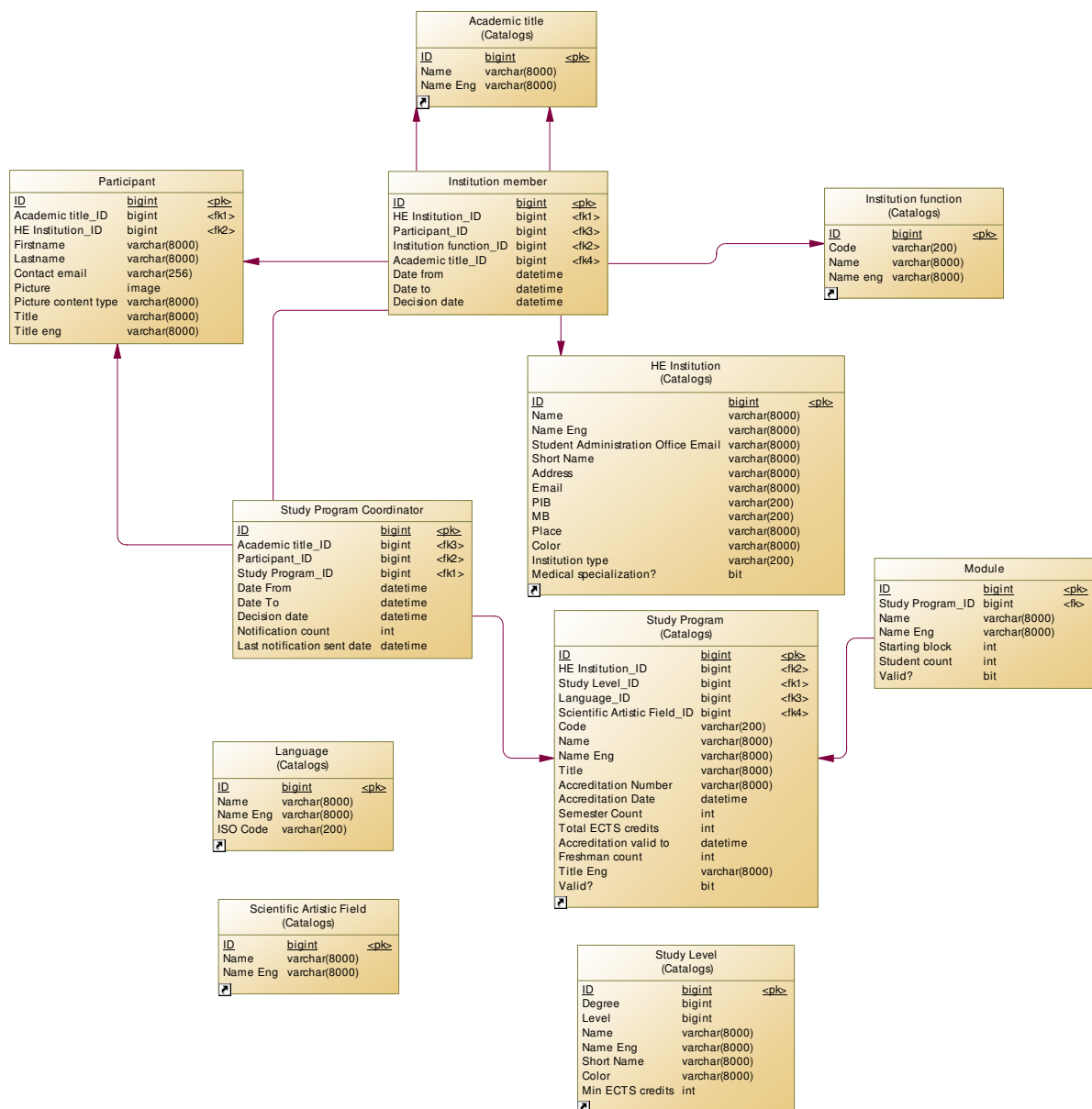
U paketu „**Participants**“ nalaze se tabele koji predstavljaju učesnike sistema.

Paket sadrži dva dijagrama koji dodatno klasifikuju tabele ovog paketa. Dijagram „**Participants**“ sadrži tabele koji opisuju lica, odnosno attribute uloga sistema. Dijagram „**HE Institution**“ sadrži tabele koji se odnose na visokoškolske ustanove.

Na slici ispod ilustrovan je sadržaj dijagrama „Participants“.



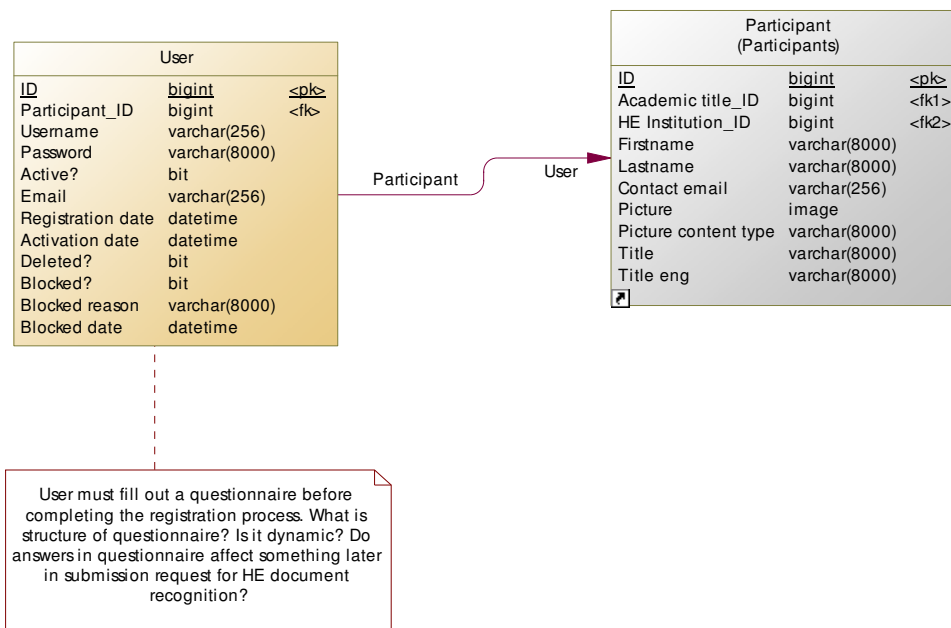
Na slici ispod ilustrovan je sadržaj dijagrama „HE Institution“.



Paket Security

Paket „**Security**“ sadrži tabele i njihove međusobne relacije koje trebaju da obezbede podršku za opis korisničkih naloga i uloga u sistemu.

Na slici ispod ilustrovan je sadržaj dijagrama „**Security**“.



Paket Submissions

U paketu „**Submissions**“ smešteni su tabele i relacije kojima su opisani detalji zahteva. Paket sadrži osam dijagrama koji dodatno klasifikuju tabele ovog paketa.

Dijagram „**Documents**“ sadrži tabelu koja opisuju dokumenta za preuzimanje.

Dijagram „**ENIC/NARIC**“ sadrži tabele koji se odnose na bazu odgovora ENIC/NARIC centra.

Dijagram „**Submission request**“ sadrži tabele koji opisuju životni ciklus zahteva.

Dijagram „**Submission data**“ je centralni deo modela koji sadrži tabele samog zahteva.

Dijagram „**Submission snapshot data**“ sadrži aktuelno stanje zahteva u jednom trenutku sa uključenim izmenama zahteva.

Dijagram „**Academic Board**“ sadrži tabele kojima je pružena podrška u radu Stručnih veća;

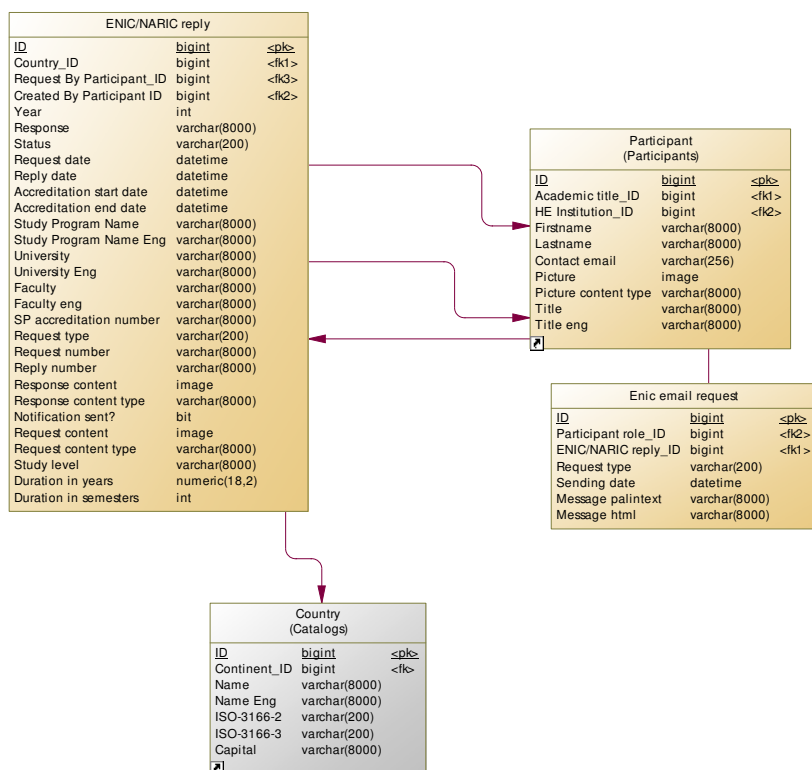
Dijagram „**Additional documents**“ sadrži tabele kojima je pružena podrška u održavanju vezanih dodatnih dokumenata od strane službenika i supervizora i

Dijagram „**Case number**“ sadrži tabelu koji služi za podršku struktuiranom broju dokumenata koji su vezani za jedan zahtev.

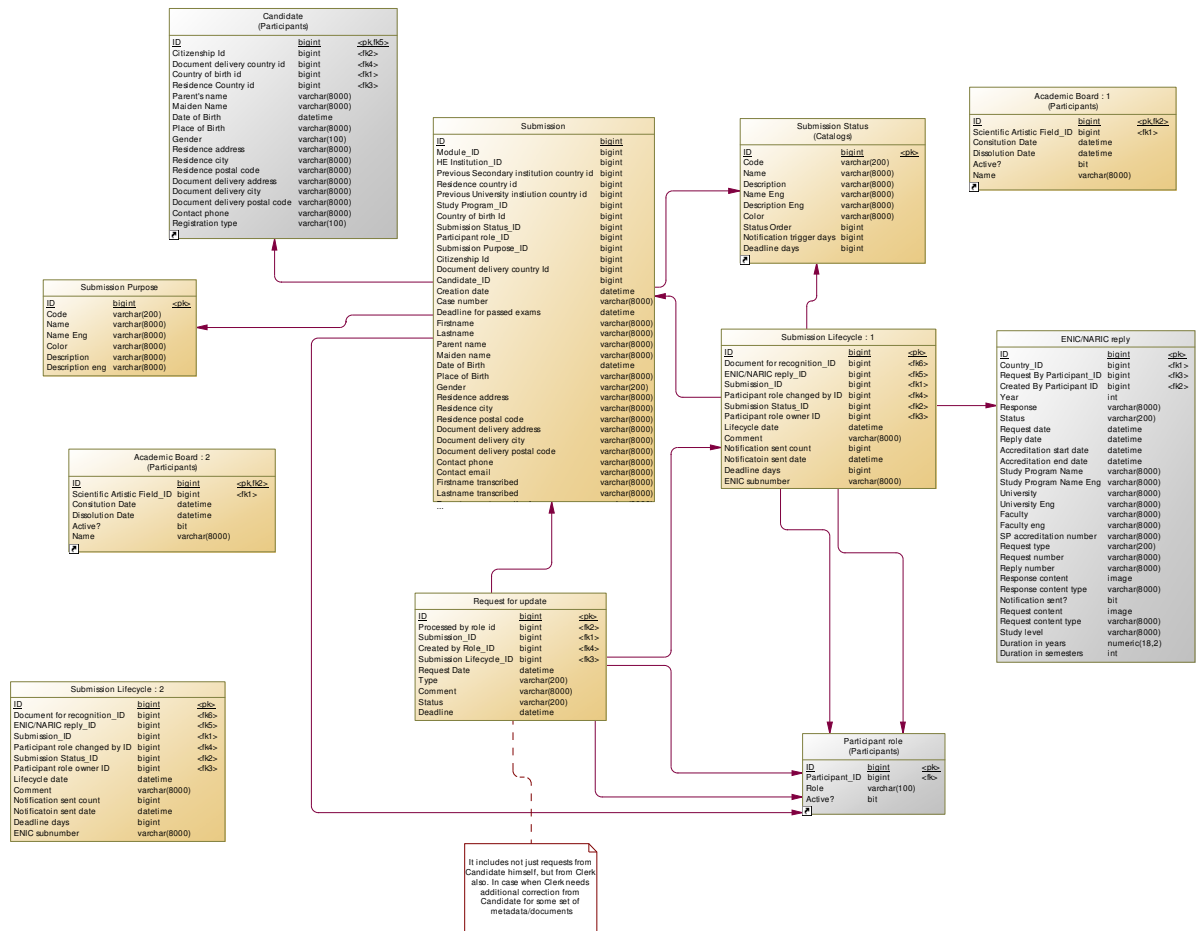
Na slici ispod ilustrovan je sadržaj dijagrama „**Documents**“.

Document for download		
ID	bigint	<pk>
Name	varchar(8000)	
Name eng	varchar(8000)	
Description	varchar(8000)	
Description eng	varchar(8000)	
Content	image	
Active?	bit	
Content type	varchar(8000)	

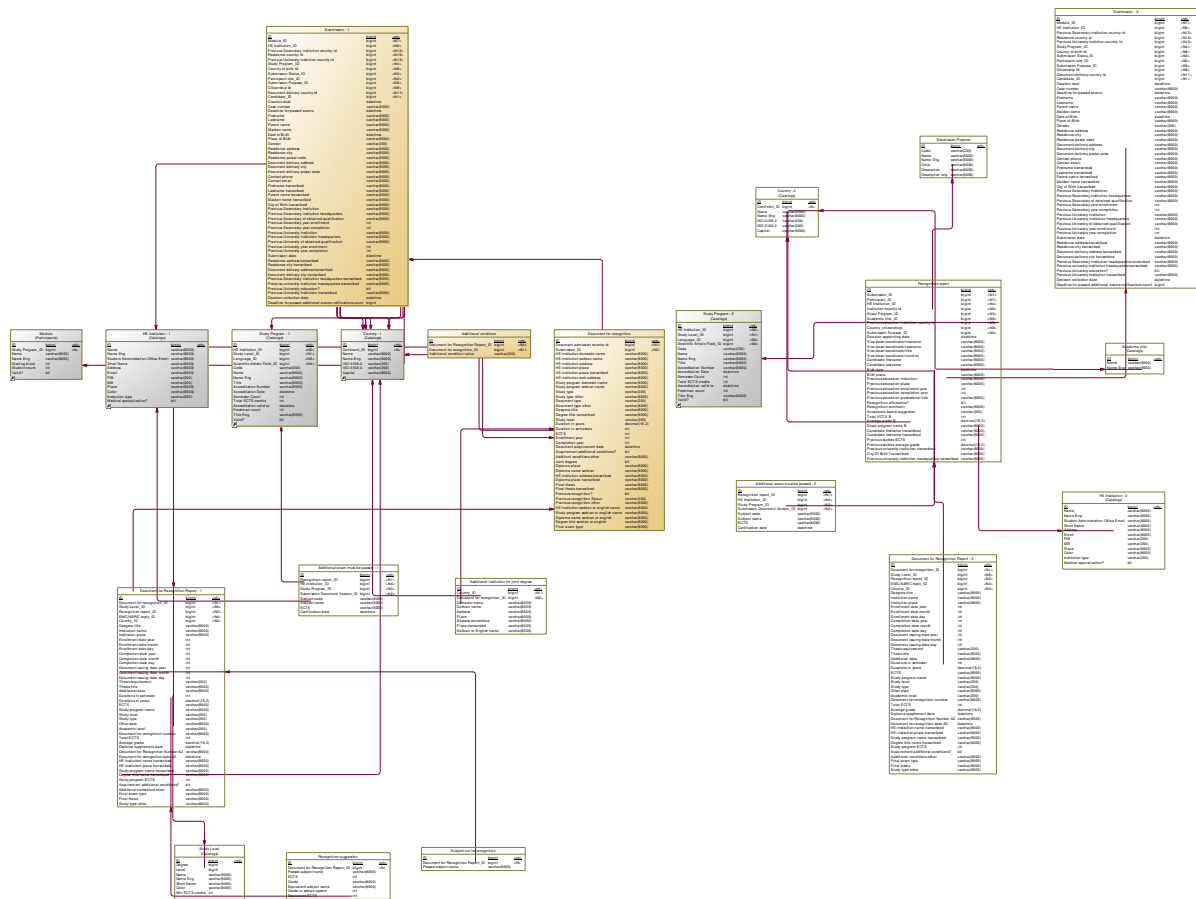
Na slici ispod ilustrovan je sadržaj dijagrama „**ENIC/NARIC**“.



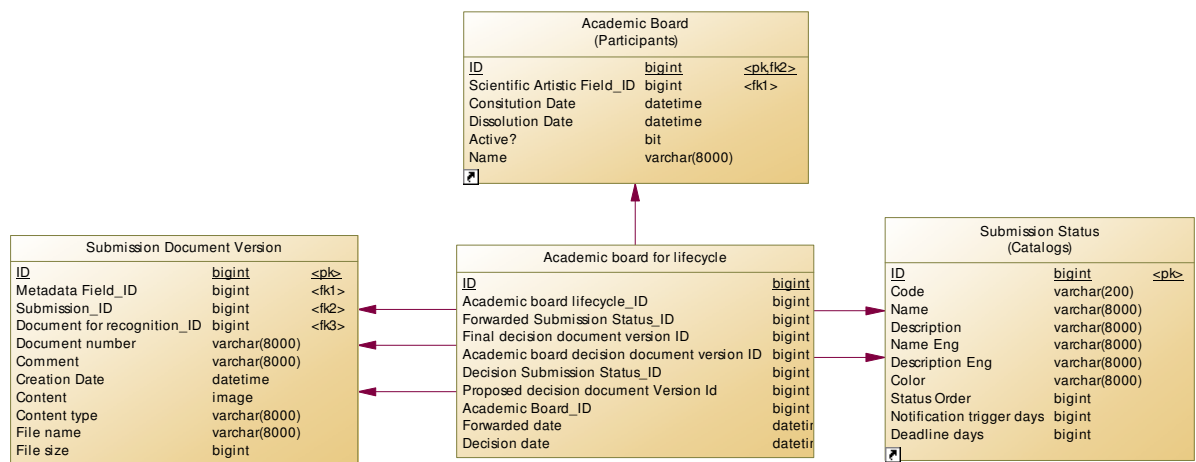
Na slici ispod ilustrovan je sadržaj dijagrama „**Submission request**“.



Na slici ispod ilustrovan je sadržaj dijagrama „Submission data“.



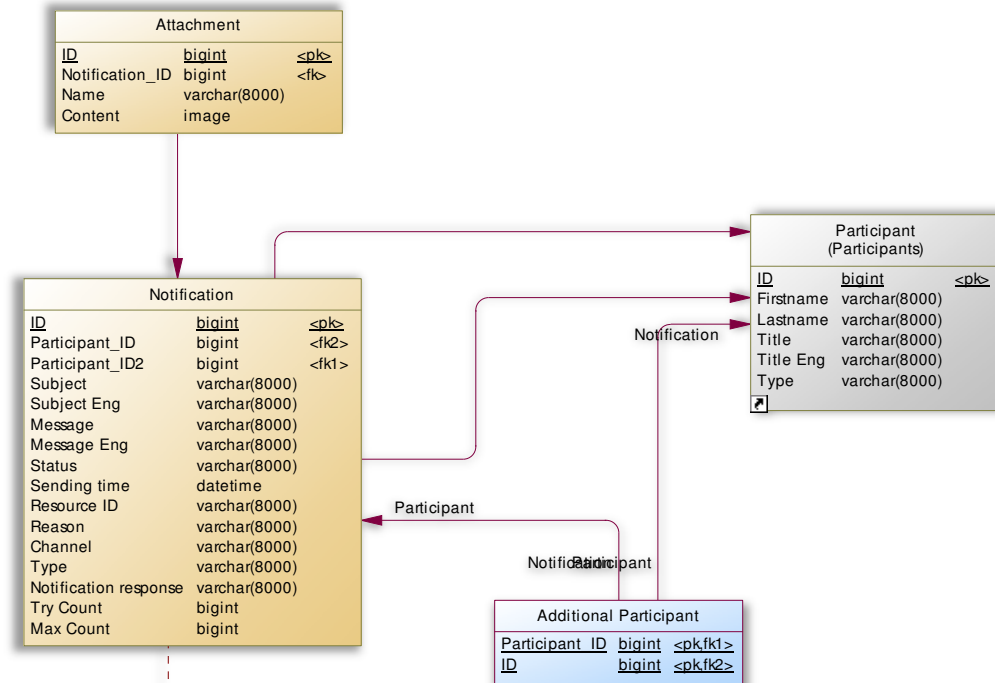
Na slici ispod ilustrovan je sadržaj dijagrama „**Academic Board**“.



Na slici ispod ilustrovan je sadržaj dijagrama „**Additional documents**“.

Paket „**Notifications**“ sadrži entitete koji treba da obezbede podršku potrebnu za internu razmenu podataka između učesnika sistema.

Na slici ispod ilustrovan je sadržaj dijagrama „Notifications“.



Paket FAQ

Paket „**FAQ**“ sadrži tabelu koja treba da obezbede podršku za održavanje i prikaz često postavljanih pitanja.

Na slici ispod ilustrovan je sadržaj dijagrama „FAQ“.

FAQ		
<u>ID</u>	bigint	<pk>
Name	varchar(8000)	
Name eng	varchar(8000)	
Question	varchar(8000)	
Question eng	varchar(8000)	
Answer	varchar(8000)	
Answer eng	varchar(8000)	
Active?	bit	

SQL Skripte

Na osnovu fizičkog modela generišu se SQL skripte (Structured Query Language scripts), koje sadrže precizne naredbe za kreiranje tabela, relacija, ograničenja i drugih objekata baze podataka. Skripte omogućavaju da se cela baza podataka automatski i reproduktivno kreira u bilo kojem okruženju, što je ključno za timski rad i kasniji razvoj sistema.

U nastavku sledi SQL Skript za EPIR bazu podataka.

```
/*=====*/
/* DBMS name:    Microsoft SQL Server 2008          */
/* Created on:    7.6.2026 10:03:25                  */
/*=====*/
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ACADEMIC_BOARD') and o.name = 'FK_ACADEMIC_BOARD_FIE_SCIENTIF')
alter table ACADEMIC_BOARD
  drop constraint FK_ACADEMIC_BOARD_FIE_SCIENTIF
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ACADEMIC_BOARD') and o.name = 'FK_ACADEMIC_PARTICIPA_PARTICIP')
alter table ACADEMIC_BOARD
  drop constraint FK_ACADEMIC_PARTICIPA_PARTICIP
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_AB_DECISION_DOC')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
  drop constraint FK_ACADEMIC_AB_DECISION_DOC
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_ASSIGNED__ACADEMIC')
```

```

alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_ASSIGNED__ACADEMIC
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_BOARDS_IN_SUBMISSI')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_BOARDS_IN_SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_DECISION_STATUS')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_DECISION_STATUS
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_FINAL_DECISION_DOC')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_FINAL_DECISION_DOC
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_FORWARDED_STATUS')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_FORWARDED_STATUS
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE') and o.name = 'FK_ACADEMIC_PROPOSED_DECISION_DOC')
alter table ACADEMIC_BOARD_FOR_LIFECYCLE
    drop constraint FK_ACADEMIC_PROPOSED_DECISION_DOC
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_CONDITION') and o.name = 'FK_ADDITION_CONDITIONS_DOCUMENT')
alter table ADDITIONAL_CONDITION
    drop constraint FK_ADDITION_CONDITIONS_DOCUMENT
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_CONDITION') and o.name = 'FK_ADDITION_ADDITIONA_DFR')
alter table ADDITIONAL_CONDITION
    drop constraint FK_ADDITION_ADDITIONA_DFR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_DOCUMENT') and o.name = 'FK_ADDITIONAL_DOCUMENT_TYPE')
alter table ADDITIONAL_DOCUMENT
    drop constraint FK_ADDITIONAL_DOCUMENT_TYPE
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_DOCUMENT') and o.name = 'FK_ADDITION_ADDITIONA_PARTICIP')
alter table ADDITIONAL_DOCUMENT
    drop constraint FK_ADDITION_ADDITIONA_PARTICIP
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_DOCUMENT') and o.name = 'FK_ADDITIONAL_DOCUMENT_DOCUMENT')
alter table ADDITIONAL_DOCUMENT
    drop constraint FK_ADDITIONAL_DOCUMENT_DOCUMENT
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED') and o.name = 'FK_ADDITION_ADDITIONA_RECOGNIT')
alter table ADDITIONAL_EXAM_MUST_BE_PASSED

```

```

drop constraint FK_ADDITION_ADDITIONA_RECOGNIT
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED') and o.name = 'FK_ADDITION_CERTIFICA_SUBMISSI')
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  drop constraint FK_ADDITION_CERTIFICA_SUBMISSI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED') and o.name = 'FK_ADDITION_INSTITUTI_HE_INSTI')
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  drop constraint FK_ADDITION_INSTITUTI_HE_INSTI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED') and o.name = 'FK_ADDITION_STUDY_PRO_STUDY_PR')
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  drop constraint FK_ADDITION_STUDY_PRO_STUDY_PR
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ADDITIONAL_INSTITUTION_FIELD') and o.name = 'FK_ADDITION_ADDITIONA_METADATA')
alter table ADDITIONAL_INSTITUTION_FIELD
  drop constraint FK_ADDITION_ADDITIONA_METADATA
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('ADDITIONAL_INSTITUTION_FIELD') and o.name = 'FK_ADDITION_FIELD_FOR_ADDITION')
alter table ADDITIONAL_INSTITUTION_FIELD
  drop constraint FK_ADDITION_FIELD_FOR_ADDITION
go

if exists (select 1

```

```

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('ADDITIONAL_INSTITUTION_FIELD') and o.name = 'FK_ADDITION_REQUEST_F_REQUEST_'
alter table ADDITIONAL_INSTITUTION_FIELD
drop constraint FK_ADDITION_REQUEST_F_REQUEST_
go

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE') and o.name =
'FK_ADDITION_ADDITIONA_DOCUMENT')
alter table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE
drop constraint FK_ADDITION_ADDITIONA_DOCUMENT
go

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE') and o.name =
'FK_ADDITION_ADDITIONA_COUNTRY')
alter table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE
drop constraint FK_ADDITION_ADDITIONA_COUNTRY
go

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('AGENCY') and o.name = 'FK_AGENCY_DELIVERY_DOCUMENT_COUNTRY')
alter table AGENCY
drop constraint FK_AGENCY_DELIVERY_DOCUMENT_COUNTRY
go

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('AGENCY') and o.name = 'FK_AGENCY_OFFICE_COUNTRY')
alter table AGENCY
drop constraint FK_AGENCY_OFFICE_COUNTRY
go

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('AGENCY') and o.name = 'FK_AGENCY_PARTICIPA_PARTICIP')
alter table AGENCY

```

```

drop constraint FK_AGENCY_PARTICIPA_PARTICIP
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('AUTHORIZATION') and o.name = 'FK_AUTHORIZ_AGENCY_AGENCY')
alter table "AUTHORIZATION"
  drop constraint FK_AUTHORIZ_AGENCY_AGENCY
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('AUTHORIZATION') and o.name = 'FK_AUTHORIZ_CANDIDATE_CANDIDAT')
alter table "AUTHORIZATION"
  drop constraint FK_AUTHORIZ_CANDIDATE_CANDIDAT
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('BOARD_MEMBER') and o.name = 'FK_BOARD_ME_MEMBER_AT_ACADEMIC')
alter table BOARD_MEMBER
  drop constraint FK_BOARD_ME_MEMBER_AT_ACADEMIC
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('BOARD_MEMBER') and o.name = 'FK_BOARD_ME_MEMBER_PO_BOARD_PO')
alter table BOARD_MEMBER
  drop constraint FK_BOARD_ME_MEMBER_PO_BOARD_PO
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('BOARD_MEMBER') and o.name = 'FK_BOARD_ME_PARTICIPA_PARTICIP')
alter table BOARD_MEMBER
  drop constraint FK_BOARD_ME_PARTICIPA_PARTICIP
go

if exists (select 1

```

```

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CANDIDATE') and o.name = 'FK_CANDIDATE_CITIZENSHIP')
alter table CANDIDATE
drop constraint FK_CANDIDATE_CITIZENSHIP
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CANDIDATE') and o.name = 'FK_CANDIDATE_COUNTRY_OF_BIRTH')
alter table CANDIDATE
drop constraint FK_CANDIDATE_COUNTRY_OF_BIRTH
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CANDIDATE') and o.name = 'FK_CANDIDATE_DOCUMENT_DELIVERY_COUNTRY')
alter table CANDIDATE
drop constraint FK_CANDIDATE_DOCUMENT_DELIVERY_COUNTRY
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CANDIDATE') and o.name = 'FK_CANDIDAT_PARTICIPA_PARTICIP')
alter table CANDIDATE
drop constraint FK_CANDIDAT_PARTICIPA_PARTICIP
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CANDIDATE') and o.name = 'FK_CANDIDATE_RESIDENCE_COUNTRY')
alter table CANDIDATE
drop constraint FK_CANDIDATE_RESIDENCE_COUNTRY
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('CASE_NUMBER_INCREMENTER') and o.name = 'FK_CASE_NUM_INCREMENT_SUBMISSI')
alter table CASE_NUMBER_INCREMENTER
drop constraint FK_CASE_NUM_INCREMENT_SUBMISSI

```

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('COUNTRY') and o.name = 'FK_COUNTRY_IN_CONTIN_CONTINEN')

alter table COUNTRY

drop constraint FK_COUNTRY_IN_CONTIN_CONTINEN

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION') and o.name = 'FK_DR_DOCUMENT_ADMISSION_COUNTRY')

alter table DOCUMENT_FOR_RECOGNITION

drop constraint FK_DR_DOCUMENT_ADMISSION_COUNTRY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION') and o.name = 'FK_DOCUMENT_DOCUMENT__SUBMISSI')

alter table DOCUMENT_FOR_RECOGNITION

drop constraint FK_DOCUMENT_DOCUMENT__SUBMISSI

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION_REPORT') and o.name = 'FK_DOCUMENT_COMPARISO_STUDY_LE')

alter table DOCUMENT_FOR_RECOGNITION_REPORT

drop constraint FK_DOCUMENT_COMPARISO_STUDY_LE

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION_REPORT') and o.name = 'FK_DOCUMENT_DFRS_IN_R_RECOGNIT')

alter table DOCUMENT_FOR_RECOGNITION_REPORT

drop constraint FK_DOCUMENT_DFRS_IN_R_RECOGNIT

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

```

    where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION_REPORT') and o.name = 'FK_DOCUMENT_DFR_REPOR_ENIC_NAR')
alter table DOCUMENT_FOR_RECOGNITION_REPORT
    drop constraint FK_DOCUMENT_DFR_REPOR_ENIC_NAR
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION_REPORT') and o.name = 'FK_DOCUMENT_DFR_REPOR_COUNTRY')
alter table DOCUMENT_FOR_RECOGNITION_REPORT
    drop constraint FK_DOCUMENT_DFR_REPOR_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('DOCUMENT_FOR_RECOGNITION_REPORT') and o.name = 'FK_DOCUMENT_DOCUMENT__DOCUMENT')
alter table DOCUMENT_FOR_RECOGNITION_REPORT
    drop constraint FK_DOCUMENT_DOCUMENT__DOCUMENT
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('DOCUMENT_METADATA') and o.name = 'FK_DOCUMENT_FOR_DOCUM_METADATA')
alter table DOCUMENT_METADATA
    drop constraint FK_DOCUMENT_FOR_DOCUM_METADATA
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('DOCUMENT_METADATA') and o.name = 'FK_DOCUMENT_METADATA_SUBMISSI')
alter table DOCUMENT_METADATA
    drop constraint FK_DOCUMENT_METADATA_SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('ENIC_EMAIL_REQUEST') and o.name = 'FK_ENIC_EMA_EMAIL_REQ_ENIC_NAR')
alter table ENIC_EMAIL_REQUEST
    drop constraint FK_ENIC_EMA_EMAIL_REQ_ENIC_NAR

```

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('ENIC_EMAIL_REQUEST') and o.name = 'FK_ENIC_EMA_ENIC_EMAI_PARTICIP')

alter table ENIC_EMAIL_REQUEST

drop constraint FK_ENIC_EMA_ENIC_EMAI_PARTICIP

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('ENIC_NARIC_REPLY') and o.name = 'FK_ENIC_NARIC_CREATED_BY')

alter table ENIC_NARIC_REPLY

drop constraint FK_ENIC_NARIC_CREATED_BY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('ENIC_NARIC_REPLY') and o.name = 'FK_ENIC_NARIC_REQUESTED_BY')

alter table ENIC_NARIC_REPLY

drop constraint FK_ENIC_NARIC_REQUESTED_BY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('ENIC_NARIC_REPLY') and o.name = 'FK_ENIC_NAR_REPLY_FOR_COUNTRY')

alter table ENIC_NARIC_REPLY

drop constraint FK_ENIC_NAR_REPLY_FOR_COUNTRY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('INSTITUTION_MEMBER') and o.name = 'FK_INSTITUT_BY_INSTIT_INSTITUT')

alter table INSTITUTION_MEMBER

drop constraint FK_INSTITUT_BY_INSTIT_INSTITUT

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

```

    where r.fkeyid = object_id('INSTITUTION_MEMBER') and o.name = 'FK_INSTITUT_MEMBER_AT_HE_INSTI')
alter table INSTITUTION_MEMBER
    drop constraint FK_INSTITUT_MEMBER_AT_HE_INSTI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('INSTITUTION_MEMBER') and o.name = 'FK_INSTITUT_MEMBER_TI_ACADEMIC')
alter table INSTITUTION_MEMBER
    drop constraint FK_INSTITUT_MEMBER_TI_ACADEMIC
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('INSTITUTION_MEMBER') and o.name = 'FK_INSTITUT_PARTICIPA_PARTICIP')
alter table INSTITUTION_MEMBER
    drop constraint FK_INSTITUT_PARTICIPA_PARTICIP
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('METADATA_FIELD_BY_PURPOSE') and o.name = 'FK_METADATA_METADATA__METADATA')
alter table METADATA_FIELD_BY_PURPOSE
    drop constraint FK_METADATA_METADATA__METADATA
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('METADATA_FIELD_BY_PURPOSE') and o.name = 'FK_METADATA_SUBMISSIO_SUBMISSI')
alter table METADATA_FIELD_BY_PURPOSE
    drop constraint FK_METADATA_SUBMISSIO_SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('MODULE') and o.name = 'FK_MODULE_MODULES_STUDY_PR')
alter table MODULE
    drop constraint FK_MODULE_MODULES_STUDY_PR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('PARTICIPANT') and o.name = 'FK_PARTICIP_AFFILIATE_HE_INSTI')
alter table PARTICIPANT
    drop constraint FK_PARTICIP_AFFILIATE_HE_INSTI
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('PARTICIPANT') and o.name = 'FK_PARTICIP_PARTICIP_ACADEMIC')
alter table PARTICIPANT
    drop constraint FK_PARTICIP_PARTICIP_ACADEMIC
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('PARTICIPANT_ROLE') and o.name = 'FK_PARTICIP_PARTICIP_PARTICIP')
alter table PARTICIPANT_ROLE
    drop constraint FK_PARTICIP_PARTICIP_PARTICIP
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_ACADEMIC__ACADEMIC')
alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_ACADEMIC__ACADEMIC
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_FOR_STUDY_STUDY_PR')
alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_FOR_STUDY_STUDY_PR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_INSTITUTI_HE_INSTI')

```

```

alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_INSTITUTI_HE_INSTI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RR_INSTITUTION_COUNTRY')
alter table RECOGNITION_REPORT
    drop constraint FK_RR_INSTITUTION_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RR_PREV_EDUCATION_INSTITUTION_COUNTRY')
alter table RECOGNITION_REPORT
    drop constraint FK_RR_PREV_EDUCATION_INSTITUTION_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_RECOGNITI_PARTICIP')
alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_RECOGNITI_PARTICIP
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_REPORT_CA_COUNTRY')
alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_REPORT_CA_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_SUBMISSIO_SUBMISSI_PURPOSE')
alter table RECOGNITION_REPORT
    drop constraint FK_RECOGNIT_SUBMISSIO_SUBMISSI_PURPOSE
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('RECOGNITION_REPORT') and o.name = 'FK_RECOGNIT_SUBMISSIO_SUBMISSI_REPORT')
alter table RECOGNITION_REPORT
  drop constraint FK_RECOGNIT_SUBMISSIO_SUBMISSI_REPORT
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('RECOGNITION_SUGGESTION') and o.name = 'FK_RECOGNIT_DOCUMENT__DOCUMENT')
alter table RECOGNITION_SUGGESTION
  drop constraint FK_RECOGNIT_DOCUMENT__DOCUMENT
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('REQUEST_FOR_UPDATE') and o.name = 'FK_REQUEST_CREATED_BY_ROLE')
alter table REQUEST_FOR_UPDATE
  drop constraint FK_REQUEST_CREATED_BY_ROLE
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('REQUEST_FOR_UPDATE') and o.name = 'FK_REQUEST__FOR_UPDAT_SUBMISSI')
alter table REQUEST_FOR_UPDATE
  drop constraint FK_REQUEST__FOR_UPDAT_SUBMISSI
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('REQUEST_FOR_UPDATE') and o.name = 'FK_REQUEST__LIFECYCLE_SUBMISSI')
alter table REQUEST_FOR_UPDATE
  drop constraint FK_REQUEST__LIFECYCLE_SUBMISSI
go

```

```

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('REQUEST_FOR_UPDATE') and o.name = 'FK_REQUEST_PROCESSED_BY_ROLE')
alter table REQUEST_FOR_UPDATE

```

```

drop constraint FK_REQUEST_PROCESSED_BY_ROLE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STUDY_PROGRAM') and o.name = 'FK_STUDY_PR_LANGUAGE__LANGUAGE')
alter table STUDY_PROGRAM
  drop constraint FK_STUDY_PR_LANGUAGE__LANGUAGE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STUDY_PROGRAM') and o.name = 'FK_STUDY_PR_STUDY_PRO_SCIENTIF')
alter table STUDY_PROGRAM
  drop constraint FK_STUDY_PR_STUDY_PRO_SCIENTIF
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STUDY_PROGRAM') and o.name = 'FK_STUDY_PR_STUDY_PRO_HE_INSTI')
alter table STUDY_PROGRAM
  drop constraint FK_STUDY_PR_STUDY_PRO_HE_INSTI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STUDY_PROGRAM') and o.name = 'FK_STUDY_PR_STUDY_PRO_STUDY_LE')
alter table STUDY_PROGRAM
  drop constraint FK_STUDY_PR_STUDY_PRO_STUDY_LE
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STUDY_PROGRAM_COORDINATOR') and o.name = 'FK_STUDY_PR_COORDAIN_T_PARTICIP')
alter table STUDY_PROGRAM_COORDINATOR
  drop constraint FK_STUDY_PR_COORDAIN_T_PARTICIP
go

if exists (select 1

```

```

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('STUDY_PROGRAM_COORDINATOR') and o.name = 'FK_STUDY_PR_COORDINAT_ACADEMIC')
alter table STUDY_PROGRAM_COORDINATOR
drop constraint FK_STUDY_PR_COORDINAT_ACADEMIC
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('STUDY_PROGRAM_COORDINATOR') and o.name = 'FK_STUDY_PR_FOR_STUDY_STUDY_PR')
alter table STUDY_PROGRAM_COORDINATOR
drop constraint FK_STUDY_PR_FOR_STUDY_STUDY_PR
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('SUBJECT_NOT_FOR_RECOGNITION') and o.name = 'FK_SUBJECT__DOCUMENT__DOCUMENT')
alter table SUBJECT_NOT_FOR_RECOGNITION
drop constraint FK_SUBJECT__DOCUMENT__DOCUMENT
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_CITIZENSHIP')
alter table SUBMISSION
drop constraint FK_SUBMISSION_CITIZENSHIP
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_COUNTRY_OF_BIRTH')
alter table SUBMISSION
drop constraint FK_SUBMISSION_COUNTRY_OF_BIRTH
go

```

```

if exists (select 1
from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_CURRENT_S_SUBMISSI')
alter table SUBMISSION
drop constraint FK_SUBMISSI_CURRENT_S_SUBMISSI

```

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_CURRENT_S_STUDY_PR')

alter table SUBMISSION

drop constraint FK_SUBMISSI_CURRENT_S_STUDY_PR

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_DOCUMENT_DELIVERY_COUNTRY')

alter table SUBMISSION

drop constraint FK_SUBMISSION_DOCUMENT_DELIVERY_COUNTRY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_FOR_CANDI_CANDIDAT')

alter table SUBMISSION

drop constraint FK_SUBMISSI_FOR_CANDI_CANDIDAT

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_FOR_MODUL_MODULE')

alter table SUBMISSION

drop constraint FK_SUBMISSI_FOR_MODUL_MODULE

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_PREVIOUS_SEC_INST_COUNTRY')

alter table SUBMISSION

drop constraint FK_SUBMISSION_PREVIOUS_SEC_INST_COUNTRY

go

if exists (select 1

from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')

```

    where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_PREVIOUS_UNI_INST_COUNTRY')
alter table SUBMISSION
    drop constraint FK_SUBMISSION_PREVIOUS_UNI_INST_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_PURPOSE__SUBMISSI')
alter table SUBMISSION
    drop constraint FK_SUBMISSI_PURPOSE__SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSION_RESIDENCE_COUNTRY')
alter table SUBMISSION
    drop constraint FK_SUBMISSION_RESIDENCE_COUNTRY
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_SUBMISSIO_HE_INSTI')
alter table SUBMISSION
    drop constraint FK_SUBMISSI_SUBMISSIO_HE_INSTI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION') and o.name = 'FK_SUBMISSI_SUBMITTED_PARTICIP')
alter table SUBMISSION
    drop constraint FK_SUBMISSI_SUBMITTED_PARTICIP
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT') and o.name = 'FK_SUBMISSION_NEW_DOCUMENT')
alter table SUBMISSION_DOCUMENT
    drop constraint FK_SUBMISSION_NEW_DOCUMENT
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT') and o.name = 'FK_SUBMISSI_REQUEST_F_REQUEST_UPD')
alter table SUBMISSION_DOCUMENT
    drop constraint FK_SUBMISSI_REQUEST_F_REQUEST_UPD
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT') and o.name = 'FK_SUBMISSI_SUBMISSIO_SUBMISSI_DOC')
alter table SUBMISSION_DOCUMENT
    drop constraint FK_SUBMISSI_SUBMISSIO_SUBMISSI_DOC
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_DFR') and o.name = 'FK_SUBMISSI_NEW_DOCUM_SUBMISSI')
alter table SUBMISSION_DOCUMENT_DFR
    drop constraint FK_SUBMISSI_NEW_DOCUM_SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_DFR') and o.name = 'FK_SUBMISSION_REQUEST_DOCUMENT_DFR')
alter table SUBMISSION_DOCUMENT_DFR
    drop constraint FK_SUBMISSION_REQUEST_DOCUMENT_DFR
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_DFR') and o.name = 'FK_SUBMISSION_DFR')
alter table SUBMISSION_DOCUMENT_DFR
    drop constraint FK_SUBMISSION_DFR
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_VERSION') and o.name = 'FK_SUBMISSI_DOCUMENT__METADATA')

```

```

alter table SUBMISSION_DOCUMENT_VERSION
    drop constraint FK_SUBMISSI_DOCUMENT__METADATA
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_VERSION') and o.name = 'FK_SUBMISSI_UPLOADED__SUBMISSI')
alter table SUBMISSION_DOCUMENT_VERSION
    drop constraint FK_SUBMISSI_UPLOADED__SUBMISSI
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_DOCUMENT_VERSION') and o.name = 'FK_SUBMISSI_UPLOADED_DOCUMENT_DFR')
alter table SUBMISSION_DOCUMENT_VERSION
    drop constraint FK_SUBMISSI_UPLOADED_DOCUMENT_DFR
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD') and o.name = 'FK_SUBMISSION_METADATA')
alter table SUBMISSION_FIELD
    drop constraint FK_SUBMISSION_METADATA
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD') and o.name = 'FK_SUBMISSI_REQUEST_F_REQUEST_UPDATE')
alter table SUBMISSION_FIELD
    drop constraint FK_SUBMISSI_REQUEST_F_REQUEST_UPDATE
go

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD') and o.name = 'FK_SUBMISSI_SUBMISSIO_SUBMISSI_FIELD')
alter table SUBMISSION_FIELD
    drop constraint FK_SUBMISSI_SUBMISSIO_SUBMISSI_FIELD
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD_DFR') and o.name = 'FK_SUBMISSION_METADATA_DFR')
alter table SUBMISSION_FIELD_DFR
    drop constraint FK_SUBMISSION_METADATA_DFR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD_DFR') and o.name = 'FK_SUBMISSION_REQUEST_FIELD_DFR')
alter table SUBMISSION_FIELD_DFR
    drop constraint FK_SUBMISSION_REQUEST_FIELD_DFR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_FIELD_DFR') and o.name = 'FK_SUBMISSION_FIELD_DFR')
alter table SUBMISSION_FIELD_DFR
    drop constraint FK_SUBMISSION_FIELD_DFR
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSION_LIFECYCLE_CHANGED_BY')
alter table SUBMISSION_LIFECYCLE
    drop constraint FK_SUBMISSION_LIFECYCLE_CHANGED_BY
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSI_ENIC_FOR__DOCUMENT')
alter table SUBMISSION_LIFECYCLE
    drop constraint FK_SUBMISSI_ENIC_FOR__DOCUMENT
go

```

```

if exists (select 1
    from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
    where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSI_FROM_ENIC_ENIC_NAR')
alter table SUBMISSION_LIFECYCLE

```

```

drop constraint FK_SUBMISSI_FROM_ENIC_ENIC_NAR
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSI_LIFECYCLE_SUBMISSI')
alter table SUBMISSION_LIFECYCLE
  drop constraint FK_SUBMISSI_LIFECYCLE_SUBMISSI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSION_LIFECYCLE_OWNER')
alter table SUBMISSION_LIFECYCLE
  drop constraint FK_SUBMISSION_LIFECYCLE_OWNER
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('SUBMISSION_LIFECYCLE') and o.name = 'FK_SUBMISSI_STATUS_SUBMISSI')
alter table SUBMISSION_LIFECYCLE
  drop constraint FK_SUBMISSI_STATUS_SUBMISSI
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('USER') and o.name = 'FK_USER_USER_ACCO_PARTICIP')
alter table "USER"
  drop constraint FK_USER_USER_ACCO_PARTICIP
go

if exists (select 1
  from sysindexes
  where id = object_id('ACADEMIC_BOARD')
  and name = 'BOARD_FIELD_FK'
  and indid > 0
  and indid < 255)
drop index ACADEMIC_BOARD.BOARD_FIELD_FK
go

```

```
if exists (select 1
          from sysobjects
          where id = object_id('ACADEMIC_BOARD')
          and type = 'U')
drop table ACADEMIC_BOARD
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
          and name = 'BOARDS_IN_LIFECYCLE_FK'
          and indid > 0
          and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.BOARDS_IN_LIFECYCLE_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
          and name = 'ACADEMIC_BOARD_DECISION_FK'
          and indid > 0
          and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.ACADEMIC_BOARD_DECISION_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
          and name = 'FINAL_DECISION_FK'
          and indid > 0
          and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.FINAL_DECISION_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
          and name = 'PROPOSED_DECISION_FK'
```

```

        and indid > 0
        and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.PROPOSED_DECISION_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
        and name = 'DECISION_STATUS_FK'
        and indid > 0
        and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.DECISION_STATUS_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
        and name = 'FORWARDED_STATUS_FK'
        and indid > 0
        and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.FORWARDED_STATUS_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
        and name = 'ASSIGNED_ACADEMIC_BOARD_FK'
        and indid > 0
        and indid < 255)
drop index ACADEMIC_BOARD_FOR_LIFECYCLE.ASSIGNED_ACADEMIC_BOARD_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('ACADEMIC_BOARD_FOR_LIFECYCLE')
        and type = 'U')
drop table ACADEMIC_BOARD_FOR_LIFECYCLE
go

```

```

if exists (select 1
            from sysobjects
            where id = object_id('ACADEMIC_TITLE')
            and type = 'U')
drop table ACADEMIC_TITLE
go

if exists (select 1
            from sysindexes
            where id = object_id('ADDITIONAL_CONDITION')
            and name = 'ADDITIONAL_CONDITIONS_IN_REPORT_FK'
            and indid > 0
            and indid < 255)
drop index ADDITIONAL_CONDITION.ADDITIONAL_CONDITIONS_IN_REPORT_FK
go

if exists (select 1
            from sysindexes
            where id = object_id('ADDITIONAL_CONDITION')
            and name = 'ADDITIONAL_CONDITIONS_FK'
            and indid > 0
            and indid < 255)
drop index ADDITIONAL_CONDITION.ADDITIONAL_CONDITIONS_FK
go

if exists (select 1
            from sysobjects
            where id = object_id('ADDITIONAL_CONDITION')
            and type = 'U')
drop table ADDITIONAL_CONDITION
go

if exists (select 1
            from sysindexes
            where id = object_id('ADDITIONAL_DOCUMENT')
            and name = 'ADDITIONAL_DOCUMENT_UPLOADED_BY_FK'
            and indid > 0
            and indid < 255)
drop index ADDITIONAL_DOCUMENT.ADDITIONAL_DOCUMENT_UPLOADED_BY_FK

```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('ADDITIONAL_DOCUMENT')
    and name = 'ADDITIONAL_DOCUMENT_TYPE_FK'
    and indid > 0
    and indid < 255)
drop index ADDITIONAL_DOCUMENT.ADDITIONAL_DOCUMENT_TYPE_FK
```

go

```
if exists (select 1
    from sysobjects
    where id = object_id('ADDITIONAL_DOCUMENT')
    and type = 'U')
drop table ADDITIONAL_DOCUMENT
```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('ADDITIONAL_DOCUMENT_TYPE')
    and name = 'ADDITIONAL_DOCUMENT_CODE'
    and indid > 0
    and indid < 255)
drop index ADDITIONAL_DOCUMENT_TYPE.ADDITIONAL_DOCUMENT_CODE
```

go

```
if exists (select 1
    from sysobjects
    where id = object_id('ADDITIONAL_DOCUMENT_TYPE')
    and type = 'U')
drop table ADDITIONAL_DOCUMENT_TYPE
```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED')
    and name = 'INSTITUTION_OF_ADDITIONAL_PASSED_EXAM_FK'
    and indid > 0
```

```

        and indid < 255)
drop index ADDITIONAL_EXAM_MUST_BE_PASSED.INSTITUTION_OF_ADDITIONAL_PASSED_EXAM_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED')
        and name = 'STUDY_PROGRAM_OF_ADDITIONAL_PASSED_EXAM_FK'
        and indid > 0
        and indid < 255)
drop index ADDITIONAL_EXAM_MUST_BE_PASSED.STUDY_PROGRAM_OF_ADDITIONAL_PASSED_EXAM_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED')
        and name = 'CERTIFICATE_OF_PASSED_EXAM_FK'
        and indid > 0
        and indid < 255)
drop index ADDITIONAL_EXAM_MUST_BE_PASSED.CERTIFICATE_OF_PASSED_EXAM_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED')
        and name = 'ADDITIONAL_EXAMS_FK'
        and indid > 0
        and indid < 255)
drop index ADDITIONAL_EXAM_MUST_BE_PASSED.ADDITIONAL_EXAMS_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('ADDITIONAL_EXAM_MUST_BE_PASSED')
        and type = 'U')
drop table ADDITIONAL_EXAM_MUST_BE_PASSED
go

if exists (select 1

```

```

        from sysindexes
        where id = object_id('ADDITIONAL_INSTITUTION_FIELD')
        and name = 'REQUEST_FOR_ADDITIONAL_INST_FIELD_FK'
        and indid > 0
        and indid < 255)

drop index ADDITIONAL_INSTITUTION_FIELD.REQUEST_FOR_ADDITIONAL_INST_FIELD_FK
go

```

```

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_INSTITUTION_FIELD')
        and name = 'FIELD_FOR_ADDITIONAL_INSTITUTION_FK'
        and indid > 0
        and indid < 255)

drop index ADDITIONAL_INSTITUTION_FIELD.FIELD_FOR_ADDITIONAL_INSTITUTION_FK
go

```

```

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_INSTITUTION_FIELD')
        and name = 'ADDITIONAL_INSTITUTION_META_FK'
        and indid > 0
        and indid < 255)

drop index ADDITIONAL_INSTITUTION_FIELD.ADDITIONAL_INSTITUTION_META_FK
go

```

```

if exists (select 1
        from sysobjects
        where id = object_id('ADDITIONAL_INSTITUTION_FIELD')
        and type = 'U')

drop table ADDITIONAL_INSTITUTION_FIELD
go

```

```

if exists (select 1
        from sysindexes
        where id = object_id('ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE')
        and name = 'ADDITIONAL_INSTITUTIONS_FK'
        and indid > 0
        and indid < 255)

```

```

drop index ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE.ADDITIONAL_INSTITUTIONS_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE')
                and name = 'ADDITIONAL_INSTITUTION_COUNTRY_FK'
                and indid > 0
                and indid < 255)
drop index ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE.ADDITIONAL_INSTITUTION_COUNTRY_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE')
                and type = 'U')
drop table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE
go

if exists (select 1
          from sysindexes
          where id = object_id('AGENCY')
                and name = 'DELIVERY_DOCUMENT_COUNTRY_FK'
                and indid > 0
                and indid < 255)
drop index AGENCY.DELIVERY_DOCUMENT_COUNTRY_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('AGENCY')
                and name = 'OFFICE_COUNTRY_FK'
                and indid > 0
                and indid < 255)
drop index AGENCY.OFFICE_COUNTRY_FK
go

if exists (select 1
          from sysobjects

```

```

        where id = object_id('AGENCY')
        and type = 'U')
drop table AGENCY
go

if exists (select 1
        from sysobjects
        where id = object_id('APP_CONFIG')
        and type = 'U')
drop table APP_CONFIG
go

if exists (select 1
        from sysindexes
        where id = object_id('AUTHORIZATION')
        and name = 'AGENCY_FK'
        and indid > 0
        and indid < 255)
drop index "AUTHORIZATION".AGENCY_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('AUTHORIZATION')
        and name = 'CANDIDATE_FK'
        and indid > 0
        and indid < 255)
drop index "AUTHORIZATION".CANDIDATE_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('AUTHORIZATION')
        and type = 'U')
drop table "AUTHORIZATION"
go

if exists (select 1
        from sysobjects

```

```

        where id = object_id('BINARY_ATTACHMENT')
        and type = 'U')
drop table BINARY_ATTACHMENT
go

if exists (select 1
        from sysindexes
        where id = object_id('BOARD_MEMBER')
        and name = 'PARTICIPANT_MEMBER_FK'
        and indid > 0
        and indid < 255)
drop index BOARD_MEMBER.PARTICIPANT_MEMBER_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('BOARD_MEMBER')
        and name = 'MEMBER_POSITION_FK'
        and indid > 0
        and indid < 255)
drop index BOARD_MEMBER.MEMBER_POSITION_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('BOARD_MEMBER')
        and name = 'MEMBER_AT_BOARD_FK'
        and indid > 0
        and indid < 255)
drop index BOARD_MEMBER.MEMBER_AT_BOARD_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('BOARD_MEMBER')
        and type = 'U')
drop table BOARD_MEMBER
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('BOARD_POSITION')
            and name = 'BOARD_POSITION_CODE_IND'
            and indid > 0
            and indid < 255)
drop index BOARD_POSITION.BOARD_POSITION_CODE_IND
go

```

```

if exists (select 1
            from sysobjects
            where id = object_id('BOARD_POSITION')
            and type = 'U')
drop table BOARD_POSITION
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('CANDIDATE')
            and name = 'DOCUMENT_DELIVERY_COUNTRY_FK'
            and indid > 0
            and indid < 255)
drop index CANDIDATE.DOCUMENT_DELIVERY_COUNTRY_FK
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('CANDIDATE')
            and name = 'RESIDENCE_COUNTRY_FK'
            and indid > 0
            and indid < 255)
drop index CANDIDATE.RESIDENCE_COUNTRY_FK
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('CANDIDATE')
            and name = 'CITIZENSHIP_FK'
            and indid > 0

```

```

        and indid < 255)
drop index CANDIDATE.CITIZENSHIP_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('CANDIDATE')
                and name = 'COUNTRY_OF_BIRTH_FK'
                and indid > 0
                and indid < 255)
drop index CANDIDATE.COUNTRY_OF_BIRTH_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('CANDIDATE')
                and type = 'U')
drop table CANDIDATE
go

if exists (select 1
          from sysindexes
          where id = object_id('CASE_NUMBER_INCREMENTER')
                and name = 'INCREMENTER_BY_PURPOSE_FK'
                and indid > 0
                and indid < 255)
drop index CASE_NUMBER_INCREMENTER.INCREMENTER_BY_PURPOSE_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('CASE_NUMBER_INCREMENTER')
                and type = 'U')
drop table CASE_NUMBER_INCREMENTER
go

if exists (select 1
          from sysobjects
          where id = object_id('CONTINENT'))

```

```

        and type = 'U')
drop table CONTINENT
go

if exists (select 1
          from sysindexes
          where id = object_id('COUNTRY')
                and name = 'COUNTRY_ISO31663_IND'
                and indid > 0
                and indid < 255)
drop index COUNTRY.COUNTRY_ISO31663_IND
go

if exists (select 1
          from sysindexes
          where id = object_id('COUNTRY')
                and name = 'COUNTRY_ISO31662_IND'
                and indid > 0
                and indid < 255)
drop index COUNTRY.COUNTRY_ISO31662_IND
go

if exists (select 1
          from sysindexes
          where id = object_id('COUNTRY')
                and name = 'IN_CONTINENT_FK'
                and indid > 0
                and indid < 255)
drop index COUNTRY.IN_CONTINENT_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('COUNTRY')
                and type = 'U')
drop table COUNTRY
go

if exists (select 1

```

```

        from sysobjects
        where id = object_id('DOCUMENT_FOR_DOWNLOAD')
        and type = 'U')
drop table DOCUMENT_FOR_DOWNLOAD
go

if exists (select 1
        from sysindexes
        where id = object_id('DOCUMENT_FOR_RECOGNITION')
        and name = 'DOCUMENT_FOR_RECOGNITIONS_FK'
        and indid > 0
        and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION.DOCUMENT_FOR_RECOGNITIONS_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('DOCUMENT_FOR_RECOGNITION')
        and name = 'DOCUMENT_ADMISSION_COUNTRY_FK'
        and indid > 0
        and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION.DOCUMENT_ADMISSION_COUNTRY_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('DOCUMENT_FOR_RECOGNITION')
        and type = 'U')
drop table DOCUMENT_FOR_RECOGNITION
go

if exists (select 1
        from sysindexes
        where id = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
        and name = 'COMPARISON_OF_DEGREE_FK'
        and indid > 0
        and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION_REPORT.COMPARISON_OF_DEGREE_FK
go

```

```

if exists (select 1
            from sysindexes
            where id  = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
                  and name = 'DFR_REPORT_ENIC_NARIC_FK'
                  and indid > 0
                  and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION_REPORT.DFR_REPORT_ENIC_NARIC_FK
go

```

```

if exists (select 1
            from sysindexes
            where id  = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
                  and name = 'DFR_REPORT_INSTITUTION_COUNTRY_FK'
                  and indid > 0
                  and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION_REPORT.DFR_REPORT_INSTITUTION_COUNTRY_FK
go

```

```

if exists (select 1
            from sysindexes
            where id  = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
                  and name = 'DFRS_IN_REPORT_FK'
                  and indid > 0
                  and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION_REPORT.DFRS_IN_REPORT_FK
go

```

```

if exists (select 1
            from sysindexes
            where id  = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
                  and name = 'DOCUMENT_FOR_RECOGNITION_REPORT_FK'
                  and indid > 0
                  and indid < 255)
drop index DOCUMENT_FOR_RECOGNITION_REPORT.DOCUMENT_FOR_RECOGNITION_REPORT_FK
go

```

```

if exists (select 1
            from sysobjects

```

```
        where id = object_id('DOCUMENT_FOR_RECOGNITION_REPORT')
        and type = 'U')
drop table DOCUMENT_FOR_RECOGNITION_REPORT
go
```

```
if exists (select 1
    from sysindexes
    where id = object_id('DOCUMENT_METADATA')
    and name = 'FOR_DOCUMENT_METADATA_FK'
    and indid > 0
    and indid < 255)
drop index DOCUMENT_METADATA.FOR_DOCUMENT_METADATA_FK
go
```

```
if exists (select 1
    from sysindexes
    where id = object_id('DOCUMENT_METADATA')
    and name = 'METADATA_FK'
    and indid > 0
    and indid < 255)
drop index DOCUMENT_METADATA.METADATA_FK
go
```

```
if exists (select 1
    from sysobjects
    where id = object_id('DOCUMENT_METADATA')
    and type = 'U')
drop table DOCUMENT_METADATA
go
```

```
if exists (select 1
    from sysindexes
    where id = object_id('ENIC_EMAIL_REQUEST')
    and name = 'ENIC_EMAIL_REQUEST_BY_FK'
    and indid > 0
    and indid < 255)
drop index ENIC_EMAIL_REQUEST.ENIC_EMAIL_REQUEST_BY_FK
go
```

```

if exists (select 1
            from sysindexes
            where id = object_id('ENIC_EMAIL_REQUEST')
            and name = 'EMAIL_REQUEST_FOR_ENIC_FK'
            and indid > 0
            and indid < 255)
drop index ENIC_EMAIL_REQUEST.EMAIL_REQUEST_FOR_ENIC_FK
go

```

```

if exists (select 1
            from sysobjects
            where id = object_id('ENIC_EMAIL_REQUEST')
            and type = 'U')
drop table ENIC_EMAIL_REQUEST
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('ENIC_NARIC_REPLY')
            and name = 'ENIC_REQUESTED_BY_FK'
            and indid > 0
            and indid < 255)
drop index ENIC_NARIC_REPLY.ENIC_REQUESTED_BY_FK
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('ENIC_NARIC_REPLY')
            and name = 'ENIC_CREATED_BY_FK'
            and indid > 0
            and indid < 255)
drop index ENIC_NARIC_REPLY.ENIC_CREATED_BY_FK
go

```

```

if exists (select 1
            from sysindexes
            where id = object_id('ENIC_NARIC_REPLY')
            and name = 'REPLY_FOR_COUNTRY_FK'
            and indid > 0

```

```

        and indid < 255)
drop index ENIC_NARIC_REPLY.REPLY_FOR_COUNTRY_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('ENIC_NARIC_REPLY')
          and type = 'U')
drop table ENIC_NARIC_REPLY
go

if exists (select 1
          from sysobjects
          where id = object_id('FAQ')
          and type = 'U')
drop table FAQ
go

if exists (select 1
          from sysobjects
          where id = object_id('HE_INSTITUTION')
          and type = 'U')
drop table HE_INSTITUTION
go

if exists (select 1
          from sysindexes
          where id = object_id('INSTITUTION_FUNCTION')
          and name = 'INSTITUTION_FUNCTION_CODE_IND'
          and indid > 0
          and indid < 255)
drop index INSTITUTION_FUNCTION.INSTITUTION_FUNCTION_CODE_IND
go

if exists (select 1
          from sysobjects
          where id = object_id('INSTITUTION_FUNCTION')
          and type = 'U')
drop table INSTITUTION_FUNCTION

```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('INSTITUTION_MEMBER')
    and name = 'MEMBER_TITLE_FK'
    and indid > 0
    and indid < 255)
drop index INSTITUTION_MEMBER.MEMBER_TITLE_FK
```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('INSTITUTION_MEMBER')
    and name = 'PARTICIPANT_MEMBER_FK'
    and indid > 0
    and indid < 255)
drop index INSTITUTION_MEMBER.PARTICIPANT_MEMBER_FK
```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('INSTITUTION_MEMBER')
    and name = 'BY_INSTITUTION_FUNCTION_FK'
    and indid > 0
    and indid < 255)
drop index INSTITUTION_MEMBER.BY_INSTITUTION_FUNCTION_FK
```

go

```
if exists (select 1
    from sysindexes
    where id = object_id('INSTITUTION_MEMBER')
    and name = 'MEMBER_AT_INSTITUTION_FK'
    and indid > 0
    and indid < 255)
drop index INSTITUTION_MEMBER.MEMBER_AT_INSTITUTION_FK
```

go

if exists (select 1

```
        from sysobjects
        where id = object_id('INSTITUTION_MEMBER')
        and type = 'U')
drop table INSTITUTION_MEMBER
go
```

```
if exists (select 1
        from sysindexes
        where id = object_id('LANGUAGE')
        and name = 'LANGUAGE_ISO_CODE_IND'
        and indid > 0
        and indid < 255)
drop index LANGUAGE.LANGUAGE_ISO_CODE_IND
go
```

```
if exists (select 1
        from sysobjects
        where id = object_id('LANGUAGE')
        and type = 'U')
drop table LANGUAGE
go
```

```
if exists (select 1
        from sysindexes
        where id = object_id('METADATA_FIELD')
        and name = 'METADATA_FIELD_CODE_IND'
        and indid > 0
        and indid < 255)
drop index METADATA_FIELD.METADATA_FIELD_CODE_IND
go
```

```
if exists (select 1
        from sysobjects
        where id = object_id('METADATA_FIELD')
        and type = 'U')
drop table METADATA_FIELD
go
```

```
if exists (select 1
```

```

        from sysindexes
    where id = object_id('METADATA_FIELD_BY_PURPOSE')
        and name = 'METADATAFIELD_FK'
        and indid > 0
        and indid < 255)
drop index METADATA_FIELD_BY_PURPOSE.METADATAFIELD_FK
go

if exists (select 1
    from sysindexes
    where id = object_id('METADATA_FIELD_BY_PURPOSE')
        and name = 'SUBMISSIONPURPOSE_FK'
        and indid > 0
        and indid < 255)
drop index METADATA_FIELD_BY_PURPOSE.SUBMISSIONPURPOSE_FK
go

if exists (select 1
    from sysobjects
    where id = object_id('METADATA_FIELD_BY_PURPOSE')
        and type = 'U')
drop table METADATA_FIELD_BY_PURPOSE
go

if exists (select 1
    from sysindexes
    where id = object_id('MODULE')
        and name = 'MODULES_FK'
        and indid > 0
        and indid < 255)
drop index MODULE.MODULES_FK
go

if exists (select 1
    from sysobjects
    where id = object_id('MODULE')
        and type = 'U')
drop table MODULE
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('PARTICIPANT')
           and name = 'PARTICIPANT_EMAIL_IND'
           and indid > 0
           and indid < 255)
drop index PARTICIPANT.PARTICIPANT_EMAIL_IND
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('PARTICIPANT')
           and name = 'AFFILIATED_INSTITUTION_FK'
           and indid > 0
           and indid < 255)
drop index PARTICIPANT.AFFILIATED_INSTITUTION_FK
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('PARTICIPANT')
           and name = 'PARTICIPANT_TITLE_FK'
           and indid > 0
           and indid < 255)
drop index PARTICIPANT.PARTICIPANT_TITLE_FK
go

```

```

if exists (select 1
           from sysobjects
           where id = object_id('PARTICIPANT')
           and type = 'U')
drop table PARTICIPANT
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('PARTICIPANT_ROLE')
           and name = 'PARTICIPANT_ROLE_FK'

```

```

        and indid > 0
        and indid < 255)
drop index PARTICIPANT_ROLE.PARTICIPANT_ROLE_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('PARTICIPANT_ROLE')
          and type = 'U')
drop table PARTICIPANT_ROLE
go

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
          and name = 'ACADEMIC_TITLE_OF_REPORT_SUBMITTER_FK'
          and indid > 0
          and indid < 255)
drop index RECOGNITION_REPORT.ACADEMIC_TITLE_OF_REPORT_SUBMITTER_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
          and name = 'REPORT_CANDIDATE_CITIZENSHIP_FK'
          and indid > 0
          and indid < 255)
drop index RECOGNITION_REPORT.REPORT_CANDIDATE_CITIZENSHIP_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
          and name = 'RECOGNITION_REPORT_OWNER_FK'
          and indid > 0
          and indid < 255)
drop index RECOGNITION_REPORT.RECOGNITION_REPORT_OWNER_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
            and name = 'INSTITUTION_COUNTRY_FK'
            and indid > 0
            and indid < 255)
drop index RECOGNITION_REPORT.INSTITUTION_COUNTRY_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
            and name = 'PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_FK'
            and indid > 0
            and indid < 255)
drop index RECOGNITION_REPORT.PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
            and name = 'INSTITUTION_FK'
            and indid > 0
            and indid < 255)
drop index RECOGNITION_REPORT.INSTITUTION_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
            and name = 'FOR_STUDY_PROGRAM_FK'
            and indid > 0
            and indid < 255)
drop index RECOGNITION_REPORT.FOR_STUDY_PROGRAM_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')

```

```

        and name = 'SUBMISSION_PURPOSE_FK'
        and indid > 0
        and indid < 255)
drop index RECOGNITION_REPORT.SUBMISSION_PURPOSE_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_REPORT')
                and name = 'SUBMISSION_RECOGNITION_FK'
                and indid > 0
                and indid < 255)
drop index RECOGNITION_REPORT.SUBMISSION_RECOGNITION_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('RECOGNITION_REPORT')
                and type = 'U')
drop table RECOGNITION_REPORT
go

if exists (select 1
          from sysindexes
          where id = object_id('RECOGNITION_SUGGESTION')
                and name = 'DOCUMENT_RECOGNITION_SUGGESTION_FK'
                and indid > 0
                and indid < 255)
drop index RECOGNITION_SUGGESTION.DOCUMENT_RECOGNITION_SUGGESTION_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('RECOGNITION_SUGGESTION')
                and type = 'U')
drop table RECOGNITION_SUGGESTION
go

if exists (select 1

```

```

        from sysindexes
        where id = object_id('REQUEST_FOR_UPDATE')
        and name = 'CREATED_BY_FK'
        and indid > 0
        and indid < 255)
drop index REQUEST_FOR_UPDATE.CREATED_BY_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('REQUEST_FOR_UPDATE')
        and name = 'LIFECYCLE_CHANGE_FK'
        and indid > 0
        and indid < 255)
drop index REQUEST_FOR_UPDATE.LIFECYCLE_CHANGE_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('REQUEST_FOR_UPDATE')
        and name = 'PROCESSED_BY_FK'
        and indid > 0
        and indid < 255)
drop index REQUEST_FOR_UPDATE.PROCESSED_BY_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('REQUEST_FOR_UPDATE')
        and name = 'FOR_UPDATE_FK'
        and indid > 0
        and indid < 255)
drop index REQUEST_FOR_UPDATE.FOR_UPDATE_FK
go

if exists (select 1
        from sysobjects
        where id = object_id('REQUEST_FOR_UPDATE')
        and type = 'U')

```

```

drop table REQUEST_FOR_UPDATE
go

if exists (select 1
          from sysobjects
          where id = object_id('SCIENTIFIC_ARTISTIC_FIELD')
          and type = 'U')
drop table SCIENTIFIC_ARTISTIC_FIELD
go

if exists (select 1
          from sysobjects
          where id = object_id('STUDY_LEVEL')
          and type = 'U')
drop table STUDY_LEVEL
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM')
          and name = 'STUDY_PROGRAM_FIELD_FK'
          and indid > 0
          and indid < 255)
drop index STUDY_PROGRAM.STUDY_PROGRAM_FIELD_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM')
          and name = 'LANGUAGE_COURSE_FK'
          and indid > 0
          and indid < 255)
drop index STUDY_PROGRAM.LANGUAGE_COURSE_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM')
          and name = 'STUDY_PROGRAM_INSTITUTION_FK')

```

```

        and indid > 0
        and indid < 255)
drop index STUDY_PROGRAM.STUDY_PROGRAM_INSTITUTION_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM')
                and name = 'STUDY_PROGRAM_LEVEL_FK'
                and indid > 0
                and indid < 255)
drop index STUDY_PROGRAM.STUDY_PROGRAM_LEVEL_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('STUDY_PROGRAM')
                and type = 'U')
drop table STUDY_PROGRAM
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM_COORDINATOR')
                and name = 'COORDINATOR_TITLE_FK'
                and indid > 0
                and indid < 255)
drop index STUDY_PROGRAM_COORDINATOR.COORDINATOR_TITLE_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('STUDY_PROGRAM_COORDINATOR')
                and name = 'COORDAINITOR_FK'
                and indid > 0
                and indid < 255)
drop index STUDY_PROGRAM_COORDINATOR.COORDAINITOR_FK
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('STUDY_PROGRAM_COORDINATOR')
              and name = 'FOR_STUDY_PROGRAM_FK'
              and indid > 0
              and indid < 255)
drop index STUDY_PROGRAM_COORDINATOR.FOR_STUDY_PROGRAM_FK
go

```

```

if exists (select 1
           from sysobjects
           where id = object_id('STUDY_PROGRAM_COORDINATOR')
              and type = 'U')
drop table STUDY_PROGRAM_COORDINATOR
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('SUBJECT_NOT_FOR_RECOGNITION')
              and name = 'DOCUMENT_SUBJECT_NOT_FOR_RECOGNITION_FK'
              and indid > 0
              and indid < 255)
drop index SUBJECT_NOT_FOR_RECOGNITION.DOCUMENT_SUBJECT_NOT_FOR_RECOGNITION_FK
go

```

```

if exists (select 1
           from sysobjects
           where id = object_id('SUBJECT_NOT_FOR_RECOGNITION')
              and type = 'U')
drop table SUBJECT_NOT_FOR_RECOGNITION
go

```

```

if exists (select 1
           from sysindexes
           where id = object_id('SUBMISSION')
              and name = 'PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_FK'
              and indid > 0
              and indid < 255)
drop index SUBMISSION.PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_FK

```

go

if exists (select 1

from sysindexes

where id = object_id('SUBMISSION')

and name = 'PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_FK'

and indid > 0

and indid < 255)

drop index SUBMISSION.PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_FK

go

if exists (select 1

from sysindexes

where id = object_id('SUBMISSION')

and name = 'DOCUMENT_DELIVERY_COUNTRY_FK'

and indid > 0

and indid < 255)

drop index SUBMISSION.DOCUMENT_DELIVERY_COUNTRY_FK

go

if exists (select 1

from sysindexes

where id = object_id('SUBMISSION')

and name = 'RESIDENCE_COUNTRY_FK'

and indid > 0

and indid < 255)

drop index SUBMISSION.RESIDENCE_COUNTRY_FK

go

if exists (select 1

from sysindexes

where id = object_id('SUBMISSION')

and name = 'CITIZENSHIP_FK'

and indid > 0

and indid < 255)

drop index SUBMISSION.CITIZENSHIP_FK

go

if exists (select 1

```

        from sysindexes
        where id = object_id('SUBMISSION')
        and name = 'COUNTRY_OF_BIRTH_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION.COUNTRY_OF_BIRTH_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION')
        and name = 'FOR_MODULE_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION.FOR_MODULE_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION')
        and name = 'SUBMISSION_FOR_INSTITUTION_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION.SUBMISSION_FOR_INSTITUTION_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION')
        and name = 'PURSPOSE_OF_SUBMISSION_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION.PURPOSE_OF_SUBMISSION_FK
go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION')
        and name = 'CURRENT_STUDY_PROGRAM_FK'

```

```
        and indid > 0
        and indid < 255)
drop index SUBMISSION.CURRENT_STUDY_PROGRAM_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION')
          and name = 'CURRENT_STATUS_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION.CURRENT_STATUS_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION')
          and name = 'SUBMITTED_BY_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION.SUBMITTED_BY_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION')
          and name = 'FOR_CANDIDATE_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION.FOR_CANDIDATE_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('SUBMISSION')
          and type = 'U')
drop table SUBMISSION
go
```

```

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT')
          and name = 'REQUEST_FOR_UPDATE_DOCUMENT_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION_DOCUMENT.REQUEST_FOR_UPDATE_DOCUMENT_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT')
          and name = 'SUBMISSION_DOCUMENTS_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION_DOCUMENT.SUBMISSION_DOCUMENTS_FK
go

```

```

if exists (select 1
          from sysobjects
          where id = object_id('SUBMISSION_DOCUMENT')
          and type = 'U')
drop table SUBMISSION_DOCUMENT
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT_DFR')
          and name = 'NEW_DOCUMENT_DFR_FK'
          and indid > 0
          and indid < 255)
drop index SUBMISSION_DOCUMENT_DFR.NEW_DOCUMENT_DFR_FK
go

```

```

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT_DFR')
          and name = 'SUBMISSION_DOCUMENT_FOR_DFR_FK'
          and indid > 0

```

```

        and indid < 255)
drop index SUBMISSION_DOCUMENT_DFR.SUBMISSION_DOCUMENT_FOR_DFR_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT_DFR')
                and name = 'REQUEST_FOR_UPDATE_DOCUMENT_DFR_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_DOCUMENT_DFR.REQUEST_FOR_UPDATE_DOCUMENT_DFR_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('SUBMISSION_DOCUMENT_DFR')
                and type = 'U')
drop table SUBMISSION_DOCUMENT_DFR
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT_VERSION')
                and name = 'UPLOADED_DOCUMENTS_FOR_DFR_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_DOCUMENT_VERSION.UPLOADED_DOCUMENTS_FOR_DFR_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_DOCUMENT_VERSION')
                and name = 'UPLOADED_DOCUMENTS_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_DOCUMENT_VERSION.UPLOADED_DOCUMENTS_FK
go

if exists (select 1

```

```

        from sysindexes
    where id = object_id('SUBMISSION_DOCUMENT_VERSION')
        and name = 'DOCUMENT_METATYPE_FK'
        and indid > 0
        and indid < 255)
    drop index SUBMISSION_DOCUMENT_VERSION.DOCUMENT_METATYPE_FK
go

```

```

if exists (select 1
    from sysobjects
    where id = object_id('SUBMISSION_DOCUMENT_VERSION')
        and type = 'U')
    drop table SUBMISSION_DOCUMENT_VERSION
go

```

```

if exists (select 1
    from sysindexes
    where id = object_id('SUBMISSION_FIELD')
        and name = 'REQUEST_FOR_UPDATE_FIELD_FK'
        and indid > 0
        and indid < 255)
    drop index SUBMISSION_FIELD.REQUEST_FOR_UPDATE_FIELD_FK
go

```

```

if exists (select 1
    from sysindexes
    where id = object_id('SUBMISSION_FIELD')
        and name = 'METADESCRIPTION_FK'
        and indid > 0
        and indid < 255)
    drop index SUBMISSION_FIELD.METADESCRIPTION_FK
go

```

```

if exists (select 1
    from sysindexes
    where id = object_id('SUBMISSION_FIELD')
        and name = 'SUBMISSION_DATA_FK'
        and indid > 0
        and indid < 255)

```

```

drop index SUBMISSION_FIELD.SUBMISSION_DATA_FK
go

if exists (select 1
          from sysobjects
          where id = object_id('SUBMISSION_FIELD')
                and type = 'U')
drop table SUBMISSION_FIELD
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_FIELD_DFR')
                and name = 'METADESCRIPTION_DFR_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_FIELD_DFR.METADESCRIPTION_DFR_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_FIELD_DFR')
                and name = 'SUBMISSION_FIELD_FOR_DFR_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_FIELD_DFR.SUBMISSION_FIELD_FOR_DFR_FK
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_FIELD_DFR')
                and name = 'REQUEST_FOR_UPDATE_FIELD_DFR_FK'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_FIELD_DFR.REQUEST_FOR_UPDATE_FIELD_DFR_FK
go

if exists (select 1
          from sysobjects

```

```

        where id = object_id('SUBMISSION_FIELD_DFR')
        and type = 'U')
drop table SUBMISSION_FIELD_DFR

go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION_LIFECYCLE')
        and name = 'ENIC_FOR_DOCUMENT_FOR_RECOGNITION_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION_LIFECYCLE.ENIC_FOR_DOCUMENT_FOR_RECOGNITION_FK

go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION_LIFECYCLE')
        and name = 'FROM_ENIC_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION_LIFECYCLE.FROM_ENIC_FK

go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION_LIFECYCLE')
        and name = 'CHANGED_BY_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION_LIFECYCLE.CHANGED_BY_FK

go

if exists (select 1
        from sysindexes
        where id = object_id('SUBMISSION_LIFECYCLE')
        and name = 'OWNER_FK'
        and indid > 0
        and indid < 255)
drop index SUBMISSION_LIFECYCLE.OWNER_FK

```

go

```
if exists (select 1
            from sysindexes
            where id = object_id('SUBMISSION_LIFECYCLE')
            and name = 'STATUS_FK'
            and indid > 0
            and indid < 255)
drop index SUBMISSION_LIFECYCLE.STATUS_FK
```

go

```
if exists (select 1
            from sysindexes
            where id = object_id('SUBMISSION_LIFECYCLE')
            and name = 'LIFECYCLE_FK'
            and indid > 0
            and indid < 255)
drop index SUBMISSION_LIFECYCLE.LIFECYCLE_FK
```

go

```
if exists (select 1
            from sysobjects
            where id = object_id('SUBMISSION_LIFECYCLE')
            and type = 'U')
drop table SUBMISSION_LIFECYCLE
```

go

```
if exists (select 1
            from sysindexes
            where id = object_id('SUBMISSION_PURPOSE')
            and name = 'SUBMISSION_PURPOSE_CODE_IND'
            and indid > 0
            and indid < 255)
drop index SUBMISSION_PURPOSE.SUBMISSION_PURPOSE_CODE_IND
```

go

```
if exists (select 1
            from sysobjects
            where id = object_id('SUBMISSION_PURPOSE'))
```

```

        and type = 'U')
drop table SUBMISSION_PURPOSE
go

if exists (select 1
          from sysindexes
          where id = object_id('SUBMISSION_STATUS')
                and name = 'SUBMISSION_STATUS_CODE_IND'
                and indid > 0
                and indid < 255)
drop index SUBMISSION_STATUS.SUBMISSION_STATUS_CODE_IND
go

if exists (select 1
          from sysobjects
          where id = object_id('SUBMISSION_STATUS')
                and type = 'U')
drop table SUBMISSION_STATUS
go

if exists (select 1
          from sysindexes
          where id = object_id('USER')
                and name = 'USER_USERNAME_IND'
                and indid > 0
                and indid < 255)
drop index "USER".USER_USERNAME_IND
go

if exists (select 1
          from sysindexes
          where id = object_id('USER')
                and name = 'USER_ACCOUNT_FK'
                and indid > 0
                and indid < 255)
drop index "USER".USER_ACCOUNT_FK
go

if exists (select 1

```

```

        from sysobjects
        where id = object_id('USER')
        and type = 'U')
drop table "USER"

go

/*=====*/
/* Table: ACADEMIC_BOARD */
/*=====*/
create table ACADEMIC_BOARD (
    ID          bigint          not null,
    SCIENTIFIC_ARTISTIC_FIELD_ID bigint          not null,
    CONSITUTION_DATE    datetime    null,
    DISSOLUTION_DATE    datetime    null,
    ACTIVE             bit          not null,
    NAME              varchar(8000)    null,
    constraint PK_ACADEMIC_BOARD primary key (ID)
)
go

/*=====*/
/* Index: BOARD_FIELD_FK */
/*=====*/
create index BOARD_FIELD_FK on ACADEMIC_BOARD (
    SCIENTIFIC_ARTISTIC_FIELD_ID ASC
)
go

/*=====*/
/* Table: ACADEMIC_BOARD_FOR_LIFECYCLE */
/*=====*/
create table ACADEMIC_BOARD_FOR_LIFECYCLE (
    ID          bigint          identity,
    ACADEMIC_BOARD_LIFECYCLE_ID bigint          not null,
    FORWARDED_SUBMISSION_STATUS_ID bigint          null,
    FINAL_DECISION_DOCUMENT_VERSION_ID bigint          null,
    ACADEMIC_BOARD_DECISION_DOCUMENT_VERSION_ID bigint          null,
    DECISION_SUBMISSION_STATUS_ID bigint          null,
    PROPOSED_DECISION_DOCUMENT_VERSION_ID bigint          null,

```

```

        ACADEMIC_BOARD_ID    bigint        not null,
        FORWARDED_DATE        datetime      null,
        DECISION_DATE          datetime      null,
        constraint PK_ACADEMIC_BOARD_FOR_LIFECYCL primary key nonclustered (ID)
    )
go

/*=====*/
/* Index: ASSIGNED_ACADEMIC_BOARD_FK          */
/*=====*/
create index ASSIGNED_ACADEMIC_BOARD_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
    ACADEMIC_BOARD_ID ASC
)
go

/*=====*/
/* Index: FORWARDED_STATUS_FK                  */
/*=====*/
create index FORWARDED_STATUS_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
    FORWARDED_SUBMISSION_STATUS_ID ASC
)
go

/*=====*/
/* Index: DECISION_STATUS_FK                   */
/*=====*/
create index DECISION_STATUS_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
    DECISION_SUBMISSION_STATUS_ID ASC
)
go

/*=====*/
/* Index: PROPOSED_DECISION_FK                 */
/*=====*/
create index PROPOSED_DECISION_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
    PROPOSED_DECISION_DOCUMENT_VERSION_ID ASC
)
go

```

```

/*=====*/
/* Index: FINAL_DECISION_FK */
/*=====*/
create index FINAL_DECISION_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
FINAL_DECISION_DOCUMENT_VERSION_ID ASC
)
go

/*=====*/
/* Index: ACADEMIC_BOARD_DECISION_FK */
/*=====*/
create index ACADEMIC_BOARD_DECISION_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
ACADEMIC_BOARD_DECISION_DOCUMENT_VERSION_ID ASC
)
go

/*=====*/
/* Index: BOARDS_IN_LIFECYCLE_FK */
/*=====*/
create index BOARDS_IN_LIFECYCLE_FK on ACADEMIC_BOARD_FOR_LIFECYCLE (
ACADEMIC_BOARD_LIFECYCLE_ID ASC
)
go

/*=====*/
/* Table: ACADEMIC_TITLE */
/*=====*/
create table ACADEMIC_TITLE (
ID          bigint          identity,
NAME        varchar(8000)   not null,
NAME_ENG    varchar(8000)   not null,
constraint PK_ACADEMIC_TITLE primary key nonclustered (ID)
)
go

/*=====*/
/* Table: ADDITIONAL_CONDITION */
/*=====*/
create table ADDITIONAL_CONDITION (

```

```

        ID          bigint          identity,
        DOCUMENT_FOR_RECOGNITION_REPORT_ID bigint          null,
        DOCUMENT_FOR_RECOGNITION_ID bigint          null,
        ADDITIONAL_CONDITION_VALUE varchar(200)          not null
        constraint CKC_ADDITIONAL_CONDIT_ADDITION check (ADDITIONAL_CONDITION_VALUE in ('thesis','final_exam','other')),
        constraint PK_ADDITIONAL_CONDITION primary key nonclustered (ID)
    )
go

/*=====*/
/* Index: ADDITIONAL_CONDITIONS_FK          */
/*=====*/
create index ADDITIONAL_CONDITIONS_FK on ADDITIONAL_CONDITION (
DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

/*=====*/
/* Index: ADDITIONAL_CONDITIONS_IN_REPORT_FK          */
/*=====*/
create index ADDITIONAL_CONDITIONS_IN_REPORT_FK on ADDITIONAL_CONDITION (
DOCUMENT_FOR_RECOGNITION_REPORT_ID ASC
)
go

/*=====*/
/* Table: ADDITIONAL_DOCUMENT          */
/*=====*/
create table ADDITIONAL_DOCUMENT (
    ID          bigint          identity,
    ADDITIONAL_DOCUMENT_TYPE_ID bigint          not null,
    PARTICIPANT_ROLE_ID bigint          null,
    SUBMISSION_ID          bigint          not null,
    CONTENT          image          not null,
    CONTENT_TYPE          varchar(8000)          not null,
    COMMENT          varchar(8000)          null,
    CREATION_DATE          datetime          not null,
    FILE_NAME          varchar(8000)          null,
    FILE_SIZE          bigint          null,

```

```

VIEW_ON_FACULTY    bit                not null default 0,
constraint PK_ADDITIONAL_DOCUMENT primary key nonclustered (ID)
)
go

/*=====*/
/* Index: ADDITIONAL_DOCUMENT_TYPE_FK                */
/*=====*/
create index ADDITIONAL_DOCUMENT_TYPE_FK on ADDITIONAL_DOCUMENT (
ADDITIONAL_DOCUMENT_TYPE_ID ASC
)
go

/*=====*/
/* Index: ADDITIONAL_DOCUMENT_UPLOADED_BY_FK          */
/*=====*/
create index ADDITIONAL_DOCUMENT_UPLOADED_BY_FK on ADDITIONAL_DOCUMENT (
PARTICIPANT_ROLE_ID ASC
)
go

/*=====*/
/* Table: ADDITIONAL_DOCUMENT_TYPE                    */
/*=====*/
create table ADDITIONAL_DOCUMENT_TYPE (
ID            bigint            identity,
CODE          varchar(200)      not null,
NAME          varchar(8000)     not null,
NAME_ENG      varchar(8000)     null,
DESCRIPTION    varchar(8000)     null,
DESCRIPTION_ENG varchar(8000)    null,
SORT_ORDER     bigint           not null default 0,
VIEW_ON_FACULTY bit            not null default 0,
constraint PK_ADDITIONAL_DOCUMENT_TYPE primary key nonclustered (ID)
)
go

/*=====*/
/* Index: ADDITIONAL_DOCUMENT_CODE                    */
/*=====*/

```

```

/*=====*/
create unique index ADDITIONAL_DOCUMENT_CODE on ADDITIONAL_DOCUMENT_TYPE (
CODE ASC
)
go

/*=====*/
/* Table: ADDITIONAL_EXAM_MUST_BE_PASSED */
/*=====*/
create table ADDITIONAL_EXAM_MUST_BE_PASSED (
ID          bigint          identity,
RECOGNITION_REPORT_ID bigint          not null,
HE_INSTITUTION_ID  bigint          null,
STUDY_PROGRAM_ID   bigint          null,
SUBMISSION_DOCUMENT_VERSION_ID bigint          null,
SUBJECT_CODE       varchar(8000)    null,
SUBJECT_NAME       varchar(8000)    not null,
ECTS               varchar(8000)    null,
CERTIFICATION_DATE datetime         null,
constraint PK_ADDITIONAL_EXAM_MUST_BE_PAS primary key nonclustered (ID)
)
go

/*=====*/
/* Index: ADDITIONAL_EXAMS_FK */
/*=====*/
create index ADDITIONAL_EXAMS_FK on ADDITIONAL_EXAM_MUST_BE_PASSED (
RECOGNITION_REPORT_ID ASC
)
go

/*=====*/
/* Index: CERTIFICATE_OF_PASSED_EXAM_FK */
/*=====*/
create index CERTIFICATE_OF_PASSED_EXAM_FK on ADDITIONAL_EXAM_MUST_BE_PASSED (
SUBMISSION_DOCUMENT_VERSION_ID ASC
)
go

```

```

/*=====*/
/* Index: STUDY_PROGRAM_OF_ADDITIONAL_PASED_EXAM_FK      */
/*=====*/
create index STUDY_PROGRAM_OF_ADDITIONAL_PASED_EXAM_FK on ADDITIONAL_EXAM_MUST_BE_PASSED (
STUDY_PROGRAM_ID ASC
)
go

```

```

/*=====*/
/* Index: INSTITUTION_OF_ADDITIONAL_PASSED_EXAM_FK      */
/*=====*/
create index INSTITUTION_OF_ADDITIONAL_PASSED_EXAM_FK on ADDITIONAL_EXAM_MUST_BE_PASSED (
HE_INSTITUTION_ID ASC
)
go

```

```

/*=====*/
/* Table: ADDITIONAL_INSTITUTION_FIELD                  */
/*=====*/
create table ADDITIONAL_INSTITUTION_FIELD (
    ID          bigint          identity,
    METADATA_FIELD_ID  bigint          not null,
    REQUEST_FOR_UPDATE_ID  bigint          null,
    ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE_ID  bigint          not null,
    NEW_FIELD_VALUE    varchar(8000)    null,
    OLD_FIELD_VALUE    varchar(8000)    null,
    STATUS             varchar(200)     not null
        constraint CKC_STATUS_ADDITION check (STATUS in ('A','T','D')),
    COMMENT            varchar(8000)    null,
        constraint PK_ADDITIONAL_INSTITUTION_FIEL primary key nonclustered (ID)
)
go

```

```

/*=====*/
/* Index: ADDITIONAL_INSTITUTION_META_FK                */
/*=====*/
create index ADDITIONAL_INSTITUTION_META_FK on ADDITIONAL_INSTITUTION_FIELD (
METADATA_FIELD_ID ASC
)

```

go

/*=====*/

/* Index: FIELD_FOR_ADDITIONAL_INSTITUTION_FK */

/*=====*/

```
create index FIELD_FOR_ADDITIONAL_INSTITUTION_FK on ADDITIONAL_INSTITUTION_FIELD (
ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE_ID ASC
)
```

go

/*=====*/

/* Index: REQUEST_FOR_ADDITIONAL_INST_FIELD_FK */

/*=====*/

```
create index REQUEST_FOR_ADDITIONAL_INST_FIELD_FK on ADDITIONAL_INSTITUTION_FIELD (
REQUEST_FOR_UPDATE_ID ASC
)
```

go

/*=====*/

/* Table: ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE */

/*=====*/

```
create table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE (
  ID          bigint          identity,
  COUNTRY_ID  bigint          null,
  DOCUMENT_FOR_RECOGNITION_ID bigint          not null,
  DOMESTIC_NAME  varchar(8000) not null,
  SERBIAN_NAME  varchar(8000) null,
  ADDRESS       varchar(8000) not null,
  PLACE         varchar(8000) not null,
  ADDRESS_TRANSCRIBED varchar(8000) null,
  PLACE_TRANSCRIBED  varchar(8000) null,
  SERBIAN_OR_ENGLISH_NAME varchar(8000) null,
  constraint PK_ADDITIONAL_INSTITUTION_FOR_ primary key nonclustered (ID)
)
```

go

/*=====*/

/* Index: ADDITIONAL_INSTITUTION_COUNTRY_FK */

/*=====*/

```

create index ADDITIONAL_INSTITUTION_COUNTRY_FK on ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE (
COUNTRY_ID ASC
)
go

```

```

/*=====*/
/* Index: ADDITIONAL_INSTITUTIONS_FK          */
/*=====*/
create index ADDITIONAL_INSTITUTIONS_FK on ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE (
DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

```

```

/*=====*/
/* Table: AGENCY                             */
/*=====*/
create table AGENCY (
PARTICIPANT_ID    bigint          not null,
OFFICE_COUNTRY_ID bigint          not null,
DELIVERY_DOCUMENT_COUNTRY_ID bigint not null,
NAME              varchar(8000)   null,
TAX_IDENTIFICATION_NUMBER varchar(8000) null,
ADDRESS           varchar(8000)   not null,
CITY              varchar(8000)   not null,
POSTAL_CODE       varchar(8000)   not null,
DOCUMENT_DELIVERY_ADDRESS varchar(8000) not null,
DOCUMENT_DELIVERY_CITY varchar(8000) not null,
DOCUMENT_DELIVERY_POSTAL_CODE varchar(8000) not null,
CONTACT_PHONE     varchar(8000)   not null,
UNIQUE_MASTER_CITIZEN_NUMBER varchar(8000) null,
AGENCY_TYPE       varchar(100)    not null
constraint CKC_AGENCY_TYPE_AGENCY check (AGENCY_TYPE in ('legal_entity','individual')),
CANDIDATES_COUNT  bigint          not null default 0,
SUBMISSIONS_COUNT bigint          not null default 0,
VERSION           bigint          not null default 0,
constraint PK_AGENCY primary key (PARTICIPANT_ID)
)
go

```

```

/*=====*/
/* Index: OFFICE_COUNTRY_FK */
/*=====*/
create index OFFICE_COUNTRY_FK on AGENCY (
OFFICE_COUNTRY_ID ASC
)
go

/*=====*/
/* Index: DELIVERY_DOCUMENT_COUNTRY_FK */
/*=====*/
create index DELIVERY_DOCUMENT_COUNTRY_FK on AGENCY (
DELIVERY_DOCUMENT_COUNTRY_ID ASC
)
go

/*=====*/
/* Table: APP_CONFIG */
/*=====*/
create table APP_CONFIG (
ID          bigint          identity,
MAX_NUMBER_OF_REQUESTS_PER_CANDIDATE bigint          not null,
DEFAULT_MAX_NUMBER_OF_DOCUMENTS bigint          not null default 10,
FIXED_FILE_NUMBER  varchar(8000)  not null default '04',
ENIC_NARIC_EMAIL   varchar(8000)   null,
constraint PK_APP_CONFIG primary key nonclustered (ID)
)
go

/*=====*/
/* Table: "AUTHORIZATION" */
/*=====*/
create table "AUTHORIZATION" (
ID          bigint          not null,
PARTICIPANT_ID  bigint          not null,
ACTIVE         bit          not null,
constraint PK_AUTHORIZATION primary key (ID, PARTICIPANT_ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Agency can submit request in behalf of a candidate',
    'user', @CurrentUser, 'table', 'AUTHORIZATION'
go

/*=====*/
/* Index: CANDIDATE_FK */
/*=====*/
create index CANDIDATE_FK on "AUTHORIZATION" (
ID ASC
)
go

/*=====*/
/* Index: AGENCY_FK */
/*=====*/
create index AGENCY_FK on "AUTHORIZATION" (
PARTICIPANT_ID ASC
)
go

/*=====*/
/* Table: BINARY_ATTACHMENT */
/*=====*/
create table BINARY_ATTACHMENT (
    ID          bigint          identity,
    FILE_NAME    varchar(8000)    not null,
    CONTENT      image           not null,
    CREATION_DATE datetime        not null,
    constraint PK_BINARY_ATTACHMENT primary key nonclustered (ID)
)
go

/*=====*/
/* Table: BOARD_MEMBER */
/*=====*/

```

```

create table BOARD_MEMBER (
    ID            bigint            identity,
    PARTICIPANT_ID  bigint            not null,
    BOARD_POSITION_ID  bigint            not null,
    ACADEMIC_BOARD_ID  bigint            not null,
    DATE_FROM       datetime         null,
    DATE_TO         datetime         null,
    ACTIVE          bit              not null,
    constraint PK_BOARD_MEMBER primary key nonclustered (ID)
)
go

/*=====*/
/* Index: MEMBER_AT_BOARD_FK                */
/*=====*/
create index MEMBER_AT_BOARD_FK on BOARD_MEMBER (
    ACADEMIC_BOARD_ID ASC
)
go

/*=====*/
/* Index: MEMBER_POSITION_FK                */
/*=====*/
create index MEMBER_POSITION_FK on BOARD_MEMBER (
    BOARD_POSITION_ID ASC
)
go

/*=====*/
/* Index: PARTICIPANT_MEMBER_FK            */
/*=====*/
create index PARTICIPANT_MEMBER_FK on BOARD_MEMBER (
    PARTICIPANT_ID ASC
)
go

/*=====*/
/* Table: BOARD_POSITION                    */
/*=====*/

```

```

create table BOARD_POSITION (
    ID            bigint            identity,
    CODE          varchar(200)      not null,
    NAME          varchar(8000)     not null,
    NAME_ENG      varchar(8000)     null,
    constraint PK_BOARD_POSITION primary key nonclustered (ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Examples:
    president
    vice president
    secretary
    ...',
    'user', @CurrentUser, 'table', 'BOARD_POSITION'
go

```

```

/*=====*/
/* Index: BOARD_POSITION_CODE_IND          */
/*=====*/
create unique index BOARD_POSITION_CODE_IND on BOARD_POSITION (
CODE ASC
)
go

```

```

/*=====*/
/* Table: CANDIDATE                        */
/*=====*/
create table CANDIDATE (
    ID            bigint            not null,
    CITIZENSHIP_ID  bigint            not null,
    DOCUMENT_DELIVERY_COUNTRY_ID bigint not null,
    COUNTRY_OF_BIRTH_ID  bigint            not null,
    RESIDENCE_COUNTRY_ID  bigint            not null,
    PARENTS_NAME      varchar(8000)     not null,
    MAIDEN_NAME        varchar(8000)     null,

```

```

DATE_OF_BIRTH      datetime      not null,
PLACE_OF_BIRTH     varchar(8000)  not null,
GENDER             varchar(100)   not null default 'not_specified'
    constraint CKC_GENDER_CANDIDAT check (GENDER in ('male','female','other','not_specified')),
RESIDENCE_ADDRESS  varchar(8000)  not null,
RESIDENCE_CITY     varchar(8000)  not null,
RESIDENCE_POSTAL_CODE varchar(8000) not null,
DOCUMENT_DELIVERY_ADDRESS varchar(8000) not null,
DOCUMENT_DELIVERY_CITY varchar(8000) not null,
DOCUMENT_DELIVERY_POSTAL_CODE varchar(8000) not null,
CONTACT_PHONE      varchar(8000)  not null,
REGISTRATION_TYPE  varchar(100)   not null
    constraint CKC_REGISTRATION_TYPE_CANDIDAT check (REGISTRATION_TYPE in ('individual','agent','clerk')),
constraint PK_CANDIDATE primary key (ID)
)
go

/*=====*/
/* Index: COUNTRY_OF_BIRTH_FK          */
/*=====*/
create index COUNTRY_OF_BIRTH_FK on CANDIDATE (
COUNTRY_OF_BIRTH_ID ASC
)
go

/*=====*/
/* Index: CITIZENSHIP_FK              */
/*=====*/
create index CITIZENSHIP_FK on CANDIDATE (
CITIZENSHIP_ID ASC
)
go

/*=====*/
/* Index: RESIDENCE_COUNTRY_FK        */
/*=====*/
create index RESIDENCE_COUNTRY_FK on CANDIDATE (
RESIDENCE_COUNTRY_ID ASC
)

```

go

```
/*=====*/
/* Index: DOCUMENT_DELIVERY_COUNTRY_FK */
/*=====*/
create index DOCUMENT_DELIVERY_COUNTRY_FK on CANDIDATE (
DOCUMENT_DELIVERY_COUNTRY_ID ASC
)
go
```

```
/*=====*/
/* Table: CASE_NUMBER_INCREMENTER */
/*=====*/
create table CASE_NUMBER_INCREMENTER (
ID          bigint          identity,
SUBMISSION_PURPOSE_ID bigint      not null,
YEAR        bigint          not null,
LAST_NUMBER  bigint          not null,
VERSION      bigint          not null default 0,
constraint PK_CASE_NUMBER_INCREMENTER primary key nonclustered (ID)
)
go
```

```
/*=====*/
/* Index: INCREMENTER_BY_PURPOSE_FK */
/*=====*/
create index INCREMENTER_BY_PURPOSE_FK on CASE_NUMBER_INCREMENTER (
SUBMISSION_PURPOSE_ID ASC
)
go
```

```
/*=====*/
/* Table: CONTINENT */
/*=====*/
create table CONTINENT (
ID          bigint          identity,
NAME        varchar(8000)    not null,
NAME_ENG    varchar(8000)    not null,
constraint PK_CONTINENT primary key nonclustered (ID)
)
```

```

)
go

/*=====*/
/* Table: COUNTRY */
/*=====*/
create table COUNTRY (
    ID          bigint          identity,
    CONTINENT_ID  bigint          not null,
    NAME         varchar(8000)    not null,
    NAME_ENG     varchar(8000)    null,
    ISO_3166_2   varchar(200)     null,
    ISO_3166_3   varchar(200)     null,
    CAPITAL      varchar(8000)    null,
    constraint PK_COUNTRY primary key nonclustered (ID)
)
go

/*=====*/
/* Index: IN_CONTINENT_FK */
/*=====*/
create index IN_CONTINENT_FK on COUNTRY (
CONTINENT_ID ASC
)
go

/*=====*/
/* Index: COUNTRY_ISO31662_IND */
/*=====*/
create unique index COUNTRY_ISO31662_IND on COUNTRY (
ISO_3166_2 ASC
)
where ISO_3166_2 IS NOT NULL
go

/*=====*/
/* Index: COUNTRY_ISO31663_IND */
/*=====*/
create unique index COUNTRY_ISO31663_IND on COUNTRY (

```

ISO_3166_3 ASC

)

where ISO_3166_3 IS NOT NULL

go

/*=====*/

/* Table: DOCUMENT_FOR_DOWNLOAD */

/*=====*/

```
create table DOCUMENT_FOR_DOWNLOAD (
  ID          bigint          identity,
  NAME        varchar(8000)   not null,
  NAME_ENG    varchar(8000)   null,
  DESCRIPTION  varchar(8000)   null,
  DESCRIPTION_ENG varchar(8000) null,
  CONTENT      image          null,
  ACTIVE       bit             not null,
  CONTENT_TYPE varchar(8000)   null,
  constraint PK_DOCUMENT_FOR_DOWNLOAD primary key nonclustered (ID)
)
```

go

/*=====*/

/* Table: DOCUMENT_FOR_RECOGNITION */

/*=====*/

```
create table DOCUMENT_FOR_RECOGNITION (
  ID          bigint          identity,
  DOCUMENT_ADMISSION_COUNTRY_ID bigint          not null,
  SUBMISSION_ID  bigint          not null,
  HE_INSTITUTION_DOMESTIC_NAME varchar(8000)     not null,
  HE_INSTITUTION_SERBIAN_NAME varchar(8000)      null,
  HE_INSTITUTION_ADDRESS varchar(8000)           not null,
  HE_INSTITUTION_PLACE varchar(8000)             not null,
  HE_INSTITUTION_PLACE_TRANSCRIBED varchar(8000) null,
  HE_INSTITUTION_WEB_ADDRESS varchar(8000)        not null,
  STUDY_PROGRAM_DOMESTIC_NAME varchar(8000)       not null,
  STUDY_PROGRAM_SERBIAN_NAME varchar(8000)        null,
  STUDY_TYPE     varchar(200)    not null
  constraint KCC_STUDY_TYPE_DOCUMENT check (STUDY_TYPE in ('regular','remote','other')),
  STUDY_TYPE_OTHER varchar(8000)    null,
```

```

DOCUMENT_TYPE      varchar(200)      not null
    constraint      CKC_DOCUMENT_TYPE_DOCUMENT      check      (DOCUMENT_TYPE      in
('diploma','certificate_sp_part','diploma_certificate','other')),
DOCUMENT_TYPE_OTHER varchar(8000)      null,
DEEGREE_TITLE      varchar(8000)      null,
DEGREE_TITLE_TRANSCRIBED varchar(8000)      null,
STUDY_LEVEL        varchar(200)      null
    constraint CKC_STUDY_LEVEL_DOCUMENT check (STUDY_LEVEL is null or (STUDY_LEVEL in ('academic','professional'))),
DURATION_IN_YEARS  decimal(18,2)      not null,
DURATION_IN_SEMESTERS int              not null,
ECTS               int                not null,
ENROLLMENT_YEAR    int                null,
COMPLETION_YEAR    int                null,
DOCUMENT_ACQUIREMENT_DATE datetime      null,
ACQUIREMENT_ADDITIONAL_CONDITIONS bit      not null,
ADDITIONL_CONDITIONS_OTHER varchar(8000)      null,
JOINT_DEGREE       bit                not null,
DIPLOMA_PLACE      varchar(8000)      null,
DIPLOMA_NAME_SERBIAN varchar(8000)      null,
HE_INSTITUTION_ADDRESS_TRANSCRIBED varchar(8000)      null,
DIPLOMA_PLACE_TRANSCRIBED varchar(8000)      null,
FINAL_THESIS       varchar(8000)      null,
FINAL_THESIS_TRANSCRIBED varchar(8000)      null,
PREVIOUS_RECOGNITION bit              not null default 0,
PREVIOUS_RECOGNITION_STATUS varchar(200)      null
    constraint      CKC_PREVIOUS_RECOGNIT_DOCUMENT      check      (PREVIOUS_RECOGNITION_STATUS      is      null      or
(PREVIOUS_RECOGNITION_STATUS in ('recognized','not-recognized','other'))),
PREVIOUS_RECOGNITION_OTHER varchar(8000)      null,
HE_INSTITUTION_SERBIAN_OR_ENGLISH_NAME varchar(8000)      null,
STUDY_PROGRAM_SERBIAN_OR_ENGLISH_NAME varchar(8000)      null,
DIPLOMA_NAME_SERBIAN_OR_ENGLISH varchar(8000)      null,
DEGREE_TITLE_SERBIAN_OR_ENGLISH varchar(8000)      null,
FINAL_EXAM_TYPE    varchar(8000)      null,
    constraint PK_DOCUMENT_FOR_RECOGNITION primary key nonclustered (ID)
)
go

/*=====*/
/* Index: DOCUMENT_ADMISSION_COUNTRY_FK      */
/*=====*/

```

```

create index DOCUMENT_ADMISSION_COUNTRY_FK on DOCUMENT_FOR_RECOGNITION (
DOCUMENT_ADMISSION_COUNTRY_ID ASC
)
go

```

```

/*=====*/
/* Index: DOCUMENT_FOR_RECOGNITIONS_FK          */
/*=====*/
create index DOCUMENT_FOR_RECOGNITIONS_FK on DOCUMENT_FOR_RECOGNITION (
SUBMISSION_ID ASC
)
go

```

```

/*=====*/
/* Table: DOCUMENT_FOR_RECOGNITION_REPORT        */
/*=====*/
create table DOCUMENT_FOR_RECOGNITION_REPORT (
    ID          bigint          identity,
    DOCUMENT_FOR_RECOGNITION_ID bigint          not null,
    STUDY_LEVEL_ID    bigint          null,
    RECOGNITION_REPORT_ID bigint          not null,
    ENIC_NARIC_REPLY_ID bigint          null,
    COUNTRY_ID        bigint          null,
    DEEGREE_TITLE     varchar(8000)  null,
    INSTITUTION_NAME   varchar(8000)  null,
    INSTITUTION_PLACE  varchar(8000)  null,
    ENROLLMENT_DATE_YEAR int          null,
    ENROLLMENT_DATE_MONTH int          null,
    ENROLLMENT_DATE_DAY  int          null,
    COMPLETION_DATE_YEAR int          null,
    COMPLETION_DATE_MONTH int          null,
    COMPLETION_DATE_DAY  int          null,
    DOCUMENT_ISSUING_DATE_YEAR int          null,
    DOCUMENT_ISSUING_DATE_MONTH int          null,
    DOCUMENT_ISSUING_DATE_DAY  int          null,
    THESIS_REQUIREMENT  varchar(200)  null
    constraint CKC_THESIS_REQUIREMEN_DOCUMENT check (THESIS_REQUIREMENT is null or (THESIS_REQUIREMENT in
('mandatory','optional'))),
    THESIS_TITLE        varchar(8000)  null,

```

```

ADDITIONAL_DATA    varchar(8000)    null,
DURATIONS_IN_SEMESTER int            null,
DURATIONS_IN_YEARS decimal(18,2)     null,
ECTS               varchar(8000)     null,
STUDY_PROGRAM_NAME varchar(8000)     null,
STUDY_LEVEL        varchar(200)      null
    constraint CKC_STUDY_LEVEL_DOCUMENT_REPORT check (STUDY_LEVEL is null or (STUDY_LEVEL in
('academic','professional'))),
STUDY_TYPE         varchar(200)      null
    constraint CKC_STUDY_TYPE_DOCUMENT_REPORT check (STUDY_TYPE is null or (STUDY_TYPE in ('regular','remote','other'))),
OTHER_DATA         varchar(8000)     null,
ACADEMIC_LEVEL     varchar(200)      null
    constraint CKC_ACADEMIC_LEVEL_DOCUMENT check (ACADEMIC_LEVEL is null or (ACADEMIC_LEVEL in
('first','second','third','integrated'))),
DOCUMENT_FOR_RECOGNITION_NUMBER varchar(8000)    null,
TOTAL_ECTS         int               null,
AVERAGE_GRADE     decimal(18,2)     null,
DIPLOMA_SUPPLEMENT_DATE datetime     null,
DOCUMENT_FOR_RECOGNITION_NUMBER_A2 varchar(8000)    null,
DOCUMENT_FOR_RECOGNITION_DATE_A2 datetime     null,
HE_INSTITUTION_NAME_TRANSCRIBED varchar(8000)    null,
HE_INSTITUTION_PLACE_TRANSCRIBED varchar(8000)    null,
STUDY_PROGRAM_NAME_TRANSCRIBED varchar(8000)    null,
DEGREE_TITLE_NAME_TRANSCRIBED varchar(8000)     null,
STUDY_PROGRAM_ECTS int               null,
ACQUIREMENT_ADDITIONAL_CONDITIONS bit         null,
ADDITIONAL_CONDITIONS_OTHER varchar(8000)     null,
FINAL_EXAM_TYPE    varchar(8000)     null,
FINAL_THESIS       varchar(8000)     null,
STUDY_TYPE_OTHER   varchar(8000)     null,
    constraint PK_DOCUMENT_FOR_RECOGNITION_RE primary key nonclustered (ID)
)
go

/*=====*/
/* Index: DOCUMENT_FOR_RECOGNITION_REPORT_FK */
/*=====*/
create index DOCUMENT_FOR_RECOGNITION_REPORT_FK on DOCUMENT_FOR_RECOGNITION_REPORT (
DOCUMENT_FOR_RECOGNITION_ID ASC
)

```

```

go

/*=====*/
/* Index: DFRS_IN_REPORT_FK */
/*=====*/
create index DFRS_IN_REPORT_FK on DOCUMENT_FOR_RECOGNITION_REPORT (
RECOGNITION_REPORT_ID ASC
)
go

/*=====*/
/* Index: DFR_REPORT_INSTITUTION_COUNTRY_FK */
/*=====*/
create index DFR_REPORT_INSTITUTION_COUNTRY_FK on DOCUMENT_FOR_RECOGNITION_REPORT (
COUNTRY_ID ASC
)
go

/*=====*/
/* Index: DFR_REPORT_ENIC_NARIC_FK */
/*=====*/
create index DFR_REPORT_ENIC_NARIC_FK on DOCUMENT_FOR_RECOGNITION_REPORT (
ENIC_NARIC_REPLY_ID ASC
)
go

/*=====*/
/* Index: COMPARISON_OF_DEGREE_FK */
/*=====*/
create index COMPARISON_OF_DEGREE_FK on DOCUMENT_FOR_RECOGNITION_REPORT (
STUDY_LEVEL_ID ASC
)
go

/*=====*/
/* Table: DOCUMENT_METADATA */
/*=====*/
create table DOCUMENT_METADATA (
ID          bigint          identity,

```

```

SUBMISSION_DOCUMENT_VERSION_ID bigint          not null,
METADATA_FIELD_ID    bigint          not null,
VALUE                varchar(8000)    null,
constraint PK_DOCUMENT_METADATA primary key (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Additional metadata for a document',
    'user', @CurrentUser, 'table', 'DOCUMENT_METADATA'
go

/*=====*/
/* Index: METADATA_FK */
/*=====*/
create index METADATA_FK on DOCUMENT_METADATA (
SUBMISSION_DOCUMENT_VERSION_ID ASC
)
go

/*=====*/
/* Index: FOR_DOCUMENT_METADATA_FK */
/*=====*/
create index FOR_DOCUMENT_METADATA_FK on DOCUMENT_METADATA (
METADATA_FIELD_ID ASC
)
go

/*=====*/
/* Table: ENIC_EMAIL_REQUEST */
/*=====*/
create table ENIC_EMAIL_REQUEST (
    ID                bigint          identity,
    PARTICIPANT_ROLE_ID bigint          null,
    ENIC_NARIC_REPLY_ID bigint          not null,
    REQUEST_TYPE      varchar(200)    not null
    constraint CKC_REQUEST_TYPE_ENIC_EMA check (REQUEST_TYPE in ('regular','reminder')),

```

```

SENDING_DATE      datetime      not null,
MESSAGE_PALINTEXT varchar(8000)  not null,
MESSAGE_HTML      varchar(8000)  not null,
constraint PK_ENIC_EMAIL_REQUEST primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Sending email request to ENIC / NARIC',
    'user', @CurrentUser, 'table', 'ENIC_EMAIL_REQUEST'
go

/*=====*/
/* Index: EMAIL_REQUEST_FOR_ENIC_FK */
/*=====*/
create index EMAIL_REQUEST_FOR_ENIC_FK on ENIC_EMAIL_REQUEST (
    ENIC_NARIC_REPLY_ID ASC
)
go

/*=====*/
/* Index: ENIC_EMAIL_REQUEST_BY_FK */
/*=====*/
create index ENIC_EMAIL_REQUEST_BY_FK on ENIC_EMAIL_REQUEST (
    PARTICIPANT_ROLE_ID ASC
)
go

/*=====*/
/* Table: ENIC_NARIC_REPLY */
/*=====*/
create table ENIC_NARIC_REPLY (
    ID          bigint          identity,
    COUNTRY_ID  bigint          not null,
    REQUEST_BY_PARTICIPANT_ID bigint          null,
    CREATED_BY_PARTICIPANT_ID bigint          null,
    YEAR        int             not null,

```

```

RESPONSE          varchar(8000)      null,
STATUS            varchar(200)        not null
        constraint KC_STATUS_ENIC_NAR check (STATUS in ('accredited','not_accredited','undefined')),
REQUEST_DATE      datetime           not null,
REPLY_DATE        datetime           null,
ACCREDITATION_START_DATE datetime     null,
ACCREDITATION_END_DATE datetime     null,
STUDY_PROGRAM_NAME varchar(8000)      not null,
STUDY_PROGRAM_NAME_ENG varchar(8000)  not null,
UNIVERSITY        varchar(8000)      not null,
UNIVERSITY_ENG    varchar(8000)      not null,
FACULTY           varchar(8000)      null,
FACULTY_ENG       varchar(8000)      null,
SP_ACCREDITATION_NUMBER varchar(8000) null,
REQUEST_TYPE      varchar(200)        not null
        constraint KC_REQUEST_TYPE_ENIC_NAR check (REQUEST_TYPE in ('study_program','institution')),
REQUEST_NUMBER    varchar(8000)      null,
REPLY_NUMBER      varchar(8000)      null,
RESPONSE_CONTENT  image              null,
RESPONSE_CONTENT_TYPE varchar(8000)  null,
NOTIFICATION_SENT bit                not null default 0,
REQUEST_CONTENT   image              null,
REQUEST_CONTENT_TYPE varchar(8000)  null,
STUDY_LEVEL       varchar(8000)      null,
DURATION_IN_YEARS numeric(18,2)      null,
DURATION_IN_SEMESTERS int             null,
        constraint PK_ENIC_NARIC_REPLY primary key nonclustered (ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
        'Reply from ENIC system for the accreditation status for a study program and/or HE institution.',
        'user', @CurrentUser, 'table', 'ENIC_NARIC_REPLY'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()

```

```

execute sp_addextendedproperty 'MS_Description',
    'Probably Study Program Name in reply from ENIC',
    'user', @CurrentUser, 'table', 'ENIC_NARIC_REPLY', 'column', 'STUDY_PROGRAM_NAME'

go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Probably University Name in reply from ENIC',
    'user', @CurrentUser, 'table', 'ENIC_NARIC_REPLY', 'column', 'UNIVERSITY'

go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Probably Faculty Name in reply from ENIC',
    'user', @CurrentUser, 'table', 'ENIC_NARIC_REPLY', 'column', 'FACULTY'

go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Email notificatoin for reminder of enic naric reply',
    'user', @CurrentUser, 'table', 'ENIC_NARIC_REPLY', 'column', 'NOTIFICATION_SENT'

go

```

```

/*=====*/
/* Index: REPLY_FOR_COUNTRY_FK */
/*=====*/
create index REPLY_FOR_COUNTRY_FK on ENIC_NARIC_REPLY (
    COUNTRY_ID ASC
)

go

```

```

/*=====*/
/* Index: ENIC_CREATED_BY_FK */
/*=====*/
create index ENIC_CREATED_BY_FK on ENIC_NARIC_REPLY (
    CREATED_BY_PARTICIPANT_ID ASC
)

go

```

```

)
go

/*=====*/
/* Index: ENIC_REQUESTED_BY_FK */
/*=====*/
create index ENIC_REQUESTED_BY_FK on ENIC_NARIC_REPLY (
REQUEST_BY_PARTICIPANT_ID ASC
)
go

/*=====*/
/* Table: FAQ */
/*=====*/
create table FAQ (
    ID          bigint          identity,
    NAME        varchar(8000)    not null,
    NAME_ENG    varchar(8000)    not null,
    QUESTION    varchar(8000)    not null,
    QUESTION_ENG varchar(8000)    not null,
    ANSWER      varchar(8000)    not null,
    ANSWER_ENG  varchar(8000)    not null,
    ACTIVE      bit              not null default 1,
    constraint PK_FAQ primary key nonclustered (ID)
)
go

/*=====*/
/* Table: HE_INSTITUTION */
/*=====*/
create table HE_INSTITUTION (
    ID          bigint          identity,
    NAME        varchar(8000)    not null,
    NAME_ENG    varchar(8000)    null,
    STUDENT_ADMINISTRATION_OFFICE_EMAIL varchar(8000)    null,
    SHORT_NAME  varchar(8000)    null,
    ADDRESS     varchar(8000)    null,
    EMAIL       varchar(8000)    null,
    PIB         varchar(200)     null,

```

```

MB          varchar(200)      null,
PLACE       varchar(8000)     null,
COLOR       varchar(8000)     null,
INSTITUTION_TYPE  varchar(200) not null
        constraint CKC_INSTITUTION_TYPE_HE_INSTI check (INSTITUTION_TYPE in ('faculty','university')),
MEDICAL_SPECIALIZATION bit          not null default 0,
constraint PK_HE_INSTITUTION primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
        'High Education Institution (Faculty, University, ...)',
        'user', @CurrentUser, 'table', 'HE_INSTITUTION'
go

/*=====*/
/* Table: INSTITUTION_FUNCTION */
/*=====*/
create table INSTITUTION_FUNCTION (
        ID          bigint          identity,
        CODE        varchar(200)    not null,
        NAME        varchar(8000)   not null,
        NAME_ENG    varchar(8000)   null,
        constraint PK_INSTITUTION_FUNCTION primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
        'Dean
        Vice dean...',
        'user', @CurrentUser, 'table', 'INSTITUTION_FUNCTION'
go

/*=====*/
/* Index: INSTITUTION_FUNCTION_CODE_IND */

```

```

/*=====*/
create unique index INSTITUTION_FUNCTION_CODE_IND on INSTITUTION_FUNCTION (
CODE ASC
)
go

/*=====*/
/* Table: INSTITUTION_MEMBER */
/*=====*/
create table INSTITUTION_MEMBER (
ID          bigint          identity,
HE_INSTITUTION_ID  bigint          null,
PARTICIPANT_ID    bigint          not null,
INSTITUTION_FUNCTION_ID bigint          not null,
ACADEMIC_TITLE_ID  bigint          null,
DATE_FROM         datetime        null,
DATE_TO           datetime        null,
DECISION_DATE     datetime        null,
constraint PK_INSTITUTION_MEMBER primary key nonclustered (ID)
)
go

/*=====*/
/* Index: MEMBER_AT_INSTITUTION_FK */
/*=====*/
create index MEMBER_AT_INSTITUTION_FK on INSTITUTION_MEMBER (
HE_INSTITUTION_ID ASC
)
go

/*=====*/
/* Index: BY_INSTITUTION_FUNCTION_FK */
/*=====*/
create index BY_INSTITUTION_FUNCTION_FK on INSTITUTION_MEMBER (
INSTITUTION_FUNCTION_ID ASC
)
go

/*=====*/

```

```

/* Index: PARTICIPANT_MEMBER_FK          */
/*=====*/
create index PARTICIPANT_MEMBER_FK on INSTITUTION_MEMBER (
PARTICIPANT_ID ASC
)
go

/*=====*/
/* Index: MEMBER_TITLE_FK                */
/*=====*/
create index MEMBER_TITLE_FK on INSTITUTION_MEMBER (
ACADEMIC_TITLE_ID ASC
)
go

/*=====*/
/* Table: LANGUAGE                        */
/*=====*/
create table LANGUAGE (
    ID          bigint          identity,
    NAME        varchar(8000)    not null,
    NAME_ENG    varchar(8000)    null,
    ISO_CODE    varchar(200)     not null,
    constraint PK_LANGUAGE primary key nonclustered (ID)
)
go

/*=====*/
/* Index: LANGUAGE_ISO_CODE_IND          */
/*=====*/
create unique index LANGUAGE_ISO_CODE_IND on LANGUAGE (
ISO_CODE ASC
)
go

/*=====*/
/* Table: METADATA_FIELD                  */
/*=====*/
create table METADATA_FIELD (

```

```

ID          bigint          identity,
CODE        varchar(200)    not null,
NAME        varchar(8000)   not null,
DATA_TYPE   varchar(8000)   not null
            constraint CKC_DATA_TYPE_METADATA check (DATA_TYPE in ('string','decimal','integer','enumeration','document','date','other')),
DESCRIPTION  varchar(8000)   null,
NEED_FOR_VERIFICATION bit    not null,
VIEW_ON_FACULTY bit         not null,
CREATED_BY   varchar(8000)   not null
            constraint CKC_CREATED_BY_METADATA check (CREATED_BY in ('CA','CL','SY','UN')),
NAME_ENG     varchar(8000)   null,
DESCRIPTION_ENG varchar(8000) null,
LEVEL        varchar(200)    not null default 'submission'
            constraint CKC_LEVEL_METADATA check (LEVEL in ('submission','document-for-recognition','additional-for-joint-degree')),
DELETED      bit            not null default 0,
BACKEND_ATTRIBUTE varchar(8000) null,
BACKEND_ATTRIBUTE_RELATIONSHIP varchar(8000) null,
SORT_ORDER   bigint         not null default 0,
MAX_NUMBER   bigint         not null default 10,
            constraint PK_METADATA_FIELD primary key nonclustered (ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Catalog of all fields that are required (or optional) to be filled by candidate',
    'user', @CurrentUser, 'table', 'METADATA_FIELD'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Metadatafield is succumbed to verification by revisor (Clerk, Supervisor, ...)'

```

```

Some metadatafield is not intended for verification, for example various document generated during the submission process.',
    'user', @CurrentUser, 'table', 'METADATA_FIELD', 'column', 'NEED_FOR_VERIFICATION'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Who creates the document for example, for a field described by this metadata.
    Candidate, Clerk, System, or some other third party...',
    'user', @CurrentUser, 'table', 'METADATA_FIELD', 'column', 'CREATED_BY'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Bean attribute in backend for this column',
    'user', @CurrentUser, 'table', 'METADATA_FIELD', 'column', 'BACKEND_ATTRIBUTE'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'The name of attribute that associates external entity identified by column backend attribute
    for example backend attribute = countryId
    backend attribute relationship = country',
    'user', @CurrentUser, 'table', 'METADATA_FIELD', 'column', 'BACKEND_ATTRIBUTE_RELATIONSHIP'
go

```

```

/*=====*/
/* Index: METADATA_FIELD_CODE_IND */
/*=====*/
create unique index METADATA_FIELD_CODE_IND on METADATA_FIELD (
CODE ASC
)
go

```

```

/*=====*/
/* Table: METADATA_FIELD_BY_PURPOSE */
/*=====*/
create table METADATA_FIELD_BY_PURPOSE (
    METADATA_FIELD_ID    bigint        not null,
    ID                   bigint        not null,
    MANDATORY            bit           not null,

```

```

        constraint PK_METADATA_FIELD_BY_PURPOSE primary key (METADATA_FIELD_ID, ID)
    )
go

/*=====*/
/* Index: SUBMISSIONPURPOSE_FK */
/*=====*/
create index SUBMISSIONPURPOSE_FK on METADATA_FIELD_BY_PURPOSE (
ID ASC
)
go

/*=====*/
/* Index: METADATAFIELD_FK */
/*=====*/
create index METADATAFIELD_FK on METADATA_FIELD_BY_PURPOSE (
METADATA_FIELD_ID ASC
)
go

/*=====*/
/* Table: MODULE */
/*=====*/
create table MODULE (
    ID          bigint          identity,
    STUDY_PROGRAM_ID  bigint          not null,
    NAME        varchar(8000)    not null,
    NAME_ENG    varchar(8000)    null,
    STARTING_BLOCK  int          null,
    STUDENT_COUNT  int          null,
    VALID         bit            not null default 1,
    constraint PK_MODULE primary key nonclustered (ID)
)
go

/*=====*/
/* Index: MODULES_FK */
/*=====*/
create index MODULES_FK on MODULE (

```

```

STUDY_PROGRAM_ID ASC
)
go

/*=====*/
/* Table: PARTICIPANT */
/*=====*/

create table PARTICIPANT (
    ID          bigint          identity,
    ACADEMIC_TITLE_ID  bigint          null,
    HE_INSTITUTION_ID  bigint          null,
    FIRSTNAME      varchar(8000)  null,
    LASTNAME       varchar(8000)  null,
    CONTACT_EMAIL   varchar(256)   not null
        constraint KC_CONTACT_EMAIL_PARTICIP check (CONTACT_EMAIL = lower(CONTACT_EMAIL)),
    PICTURE         image          null,
    PICTURE_CONTENT_TYPE varchar(8000)  null,
    TITLE           varchar(8000)  null,
    TITLE_ENG       varchar(8000)  null,
        constraint PK_PARTICIPANT primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Need abstraction because many types of participant can request submission.

    For Example, Agency, Clerk and Candidate himself can request submission for a Candidate.',
    'user', @CurrentUser, 'table', 'PARTICIPANT'
go

/*=====*/
/* Index: PARTICIPANT_TITLE_FK */
/*=====*/

create index PARTICIPANT_TITLE_FK on PARTICIPANT (
    ACADEMIC_TITLE_ID ASC
)
go

```

```

/*=====*/
/* Index: AFFILIATED_INSTITUTION_FK */
/*=====*/
create index AFFILIATED_INSTITUTION_FK on PARTICIPANT (
HE_INSTITUTION_ID ASC
)
go

/*=====*/
/* Index: PARTICIPANT_EMAIL_IND */
/*=====*/
create unique index PARTICIPANT_EMAIL_IND on PARTICIPANT (
CONTACT_EMAIL ASC
)
go

/*=====*/
/* Table: PARTICIPANT_ROLE */
/*=====*/
create table PARTICIPANT_ROLE (
ID          bigint          identity,
PARTICIPANT_ID  bigint          not null,
ROLE          varchar(100)    not null
constraint CKC_ROLE_PARTICIP check (ROLE in
('candidate','clerk','supervisor','agency','agent','faculty_member','academic_board','admin')),
ACTIVE        bit           not null default 0,
constraint PK_PARTICIPANT_ROLE primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
'Participant with specified role. Participant can have many roles like clerk, supervisor. This table defines concrete relation to one
role that participates in some business function',
'user', @CurrentUser, 'table', 'PARTICIPANT_ROLE'
go

/*=====*/

```

```

/* Index: PARTICIPANT_ROLE_FK */
/*=====*/
create index PARTICIPANT_ROLE_FK on PARTICIPANT_ROLE (
PARTICIPANT_ID ASC
)
go

/*=====*/
/* Table: RECOGNITION_REPORT */
/*=====*/
create table RECOGNITION_REPORT (
ID          bigint          identity,
SUBMISSION_ID  bigint          not null,
PARTICIPANT_ID  bigint          null,
HE_INSTITUTION_ID  bigint          null,
INSTITUTION_COUNTRY_ID bigint          null,
STUDY_PROGRAM_ID  bigint          null,
ACADEMIC_TITLE_ID  bigint          null,
PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_ID bigint          null,
COUNTRY_CITIZENSHIP_ID  bigint          null,
SUBMISSION_PURPOSE_ID  bigint          null,
DECISION_APPOINTING_DATE  datetime          null,
VICE_DEAN_COORDINATOR_FIRSTNAME  varchar(8000)          null,
VICE_DEAN_COORDINATOR_LASTNAME  varchar(8000)          null,
VICE_DEAN_COORDINATOR_TITLE  varchar(8000)          null,
VICE_DEAN_COORDINATOR_FUNCTION  varchar(8000)          null,
CANDIDATE_FIRSTNAME  varchar(8000)          null,
CANDIDATE_LASTNAME  varchar(8000)          null,
BIRTH_DATE          datetime          null,
BIRTH_PLACE          varchar(8000)          null,
PREVIOUS_EDUCATION_INSTITUTION  varchar(8000)          null,
PREVIOUS_EDUCATION_PLACE  varchar(8000)          null,
PREVIOUS_EDUCATION_ENROLLMENT_YEAR  int          null,
PREVIOUS_EDUCATION_COMPLETION_YEAR  int          null,
PREVIOUS_EDUCATION_PROFESSIONAL_TITLE  varchar(8000)          null,
RECOGNITION_AFFIRMATIVE  bit          null,
RECOGNITION_COMMENT  varchar(8000)          null,
ACCADEMIC_BOARD_SUGGESTION  varchar(200)          null

```

```

        constraint CKC_ACCADEMIC_BOARD_S_RECOGNIT check (ACCADEMIC_BOARD_SUGGESTION is null or
(ACCADEMIC_BOARD_SUGGESTION in ('A','B','C'))),
TOTAL_ECTS_B int null,
AVERAGE_GRADE_B decimal(18,2) null,
STUDY_PROGRAM_NAME_B varchar(8000) null,
CANDIDATE_FIRSTNAME_TRANSCRIBED varchar(8000) null,
CANDIDATE_LASTNAME_TRANSCRIBED varchar(8000) null,
PREVIOUS_STUDIES_ECTS int null,
PREVIOUS_STUDIES_AVERAGE_GRADE decimal(18,2) null,
PREVIOUS_UNIVERSITY_INSTITUTION_TRANSCRIBED varchar(8000) null,
CITY_OF_BIRTH_TRANSCRIBED varchar(8000) null,
PREVIOUS_UNIVERSITY_INSTITUTION_HEADQUARTERS_TRANSCRIBED varchar(8000) null,
constraint PK_RECOGNITION_REPORT primary key nonclustered (ID)
)
go

/*=====*/
/* Index: SUBMISSION_RECOGNITION_FK */
/*=====*/
create index SUBMISSION_RECOGNITION_FK on RECOGNITION_REPORT (
SUBMISSION_ID ASC
)
go

/*=====*/
/* Index: SUBMISSION_PURPOSE_FK */
/*=====*/
create index SUBMISSION_PURPOSE_FK on RECOGNITION_REPORT (
SUBMISSION_PURPOSE_ID ASC
)
go

/*=====*/
/* Index: FOR_STUDY_PROGRAM_FK */
/*=====*/
create index FOR_STUDY_PROGRAM_FK on RECOGNITION_REPORT (
STUDY_PROGRAM_ID ASC
)
go

```

```

/*=====*/
/* Index: INSTITUTION_FK */
/*=====*/
create index INSTITUTION_FK on RECOGNITION_REPORT (
HE_INSTITUTION_ID ASC
)
go

```

```

/*=====*/
/* Index: PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_FK */
/*=====*/
create index PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_FK on RECOGNITION_REPORT (
PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_ID ASC
)
go

```

```

/*=====*/
/* Index: INSTITUTION_COUNTRY_FK */
/*=====*/
create index INSTITUTION_COUNTRY_FK on RECOGNITION_REPORT (
INSTITUTION_COUNTRY_ID ASC
)
go

```

```

/*=====*/
/* Index: RECOGNITION_REPORT_OWNER_FK */
/*=====*/
create index RECOGNITION_REPORT_OWNER_FK on RECOGNITION_REPORT (
PARTICIPANT_ID ASC
)
go

```

```

/*=====*/
/* Index: REPORT_CANDIDATE_CITIZENSHIP_FK */
/*=====*/
create index REPORT_CANDIDATE_CITIZENSHIP_FK on RECOGNITION_REPORT (
COUNTRY_CITIZENSHIP ASC
)

```

go

```
/*=====*/
/* Index: ACADEMIC_TITLE_OF_REPORT_SUBMITTER_FK      */
/*=====*/
create index ACADEMIC_TITLE_OF_REPORT_SUBMITTER_FK on RECOGNITION_REPORT (
ACADEMIC_TITLE_ID ASC
)
go
```

```
/*=====*/
/* Table: RECOGNITION_SUGGESTION                      */
/*=====*/
create table RECOGNITION_SUGGESTION (
ID          bigint          identity,
DOCUMENT_FOR_RECOGNITION_REPORT_ID bigint          not null,
PASSED_SUBJECT_NAME varchar(8000) null,
ECTS        int            null,
GRADE       varchar(8000)   null,
EQUIVALENT_SUBJECT_NAME varchar(8000) null,
GRADE_IN_SEBIAN_SYSTEM int      null,
EQUIVALENT_ECTS int          null,
constraint PK_RECOGNITION_SUGGESTION primary key nonclustered (ID)
)
go
```

```
/*=====*/
/* Index: DOCUMENT_RECOGNITION_SUGGESTION_FK          */
/*=====*/
create index DOCUMENT_RECOGNITION_SUGGESTION_FK on RECOGNITION_SUGGESTION (
DOCUMENT_FOR_RECOGNITION_REPORT_ID ASC
)
go
```

```
/*=====*/
/* Table: REQUEST_FOR_UPDATE                          */
/*=====*/
create table REQUEST_FOR_UPDATE (
ID          bigint          identity,
```

```

        PROCESSED_BY_ROLE_ID bigint          null,
        SUBMISSION_ID      bigint          not null,
        CREATED_BY_ROLE_ID  bigint          null,
        SUBMISSION_LIFECYCLE_ID bigint        null,
        REQUEST_DATE        datetime        not null,
        TYPE                varchar(200)     not null
        constraint CKC_TYPE_REQUEST_ check (TYPE in ('for_data_modification','for_sp_change','for_cancel')),
        COMMENT            varchar(8000)     null,
        STATUS              varchar(200)     not null
        constraint CKC_STATUS_REQUEST_ check (STATUS in ('created','approved','declined','confirmed')),
        DEADLINE            datetime        not null,
        constraint PK_REQUEST_FOR_UPDATE primary key nonclustered (ID)
    )
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Candidate (Agency) can make a request for data modification, change Study Program or cancel submission',
    'user', @CurrentUser, 'table', 'REQUEST_FOR_UPDATE'
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Deadline for modification performed by candidate',
    'user', @CurrentUser, 'table', 'REQUEST_FOR_UPDATE', 'column', 'DEADLINE'
go

/*=====*/
/* Index: FOR_UPDATE_FK */
/*=====*/
create index FOR_UPDATE_FK on REQUEST_FOR_UPDATE (
SUBMISSION_ID ASC
)
go

/*=====*/
/* Index: PROCESSED_BY_FK */

```

```

/*=====*/
create index PROCESSED_BY_FK on REQUEST_FOR_UPDATE (
PROCESSED_BY_ROLE_ID ASC
)
go

/*=====*/
/* Index: LIFECYCLE_CHANGE_FK */
/*=====*/
create index LIFECYCLE_CHANGE_FK on REQUEST_FOR_UPDATE (
SUBMISSION_LIFECYCLE_ID ASC
)
go

/*=====*/
/* Index: CREATED_BY_FK */
/*=====*/
create index CREATED_BY_FK on REQUEST_FOR_UPDATE (
CREATED_BY_ROLE_ID ASC
)
go

/*=====*/
/* Table: SCIENTIFIC_ARTISTIC_FIELD */
/*=====*/
create table SCIENTIFIC_ARTISTIC_FIELD (
ID          bigint          identity,
NAME        varchar(8000)    not null,
NAME_ENG    varchar(8000)    null,
constraint PK_SCIENTIFIC_ARTISTIC_FIELD primary key nonclustered (ID)
)
go

/*=====*/
/* Table: STUDY_LEVEL */
/*=====*/
create table STUDY_LEVEL (
ID          bigint          identity,
DEGREE      bigint          not null,

```

```

LEVEL          bigint          not null,
NAME           varchar(8000)    null,
NAME_ENG       varchar(8000)    null,
SHORT_NAME     varchar(8000)    null,
COLOR          varchar(8000)    null,
MIN_ECTS_CREDITS int           null,
constraint PK_STUDY_LEVEL primary key nonclustered (ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Bachelor, Master, Doctoral',
    'user', @CurrentUser, 'table', 'STUDY_LEVEL', 'column', 'DEGREE'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Academic, Professional, ...',
    'user', @CurrentUser, 'table', 'STUDY_LEVEL', 'column', 'LEVEL'
go

```

```

/*=====*/
/* Table: STUDY_PROGRAM                                */
/*=====*/
create table STUDY_PROGRAM (
    ID          bigint          identity,
    HE_INSTITUTION_ID  bigint          not null,
    STUDY_LEVEL_ID    bigint          not null,
    LANGUAGE_ID      bigint          not null,
    SCIENTIFIC_ARTISTIC_FIELD_ID bigint          not null,
    CODE            varchar(200)    not null,
    NAME            varchar(8000)    not null,
    NAME_ENG        varchar(8000)    null,
    TITLE           varchar(8000)    not null,
    ACCREDITATION_NUMBER varchar(8000) null,
    ACCREDITATION_DATE datetime      null,

```

```

        SEMESTER_COUNT      int          null,
        TOTAL_ECTS_CREDITS  int          null,
        ACCREDITATION_VALID_TO datetime    null,
        FRESHMAN_COUNT      int          null,
        TITLE_ENG           varchar(8000) null,
        VALID               bit          not null default 1,
        constraint PK_STUDY_PROGRAM primary key nonclustered (ID)
    )
go

/*=====*/
/* Index: STUDY_PROGRAM_LEVEL_FK          */
/*=====*/
create index STUDY_PROGRAM_LEVEL_FK on STUDY_PROGRAM (
    STUDY_LEVEL_ID ASC
)
go

/*=====*/
/* Index: STUDY_PROGRAM_INSTITUTION_FK    */
/*=====*/
create index STUDY_PROGRAM_INSTITUTION_FK on STUDY_PROGRAM (
    HE_INSTITUTION_ID ASC
)
go

/*=====*/
/* Index: LANGUAGE_COURSE_FK              */
/*=====*/
create index LANGUAGE_COURSE_FK on STUDY_PROGRAM (
    LANGUAGE_ID ASC
)
go

/*=====*/
/* Index: STUDY_PROGRAM_FIELD_FK          */
/*=====*/
create index STUDY_PROGRAM_FIELD_FK on STUDY_PROGRAM (
    SCIENTIFIC_ARTISTIC_FIELD_ID ASC

```

```

)
go

/*=====*/
/* Table: STUDY_PROGRAM_COORDINATOR */
/*=====*/
create table STUDY_PROGRAM_COORDINATOR (
    ID          bigint          identity,
    ACADEMIC_TITLE_ID  bigint          null,
    PARTICIPANT_ID  bigint          not null,
    STUDY_PROGRAM_ID  bigint          not null,
    DATE_FROM      datetime       null,
    DATE_TO        datetime       null,
    DECISION_DATE   datetime       null,
    NOTIFICATION_EXPIRATION_COUNT int          null,
    NOTIFICATION_EXPIRATION_SENT_DATE datetime null,
    constraint PK_STUDY_PROGRAM_COORDINATOR primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Notification about expiration',
    'user', @CurrentUser, 'table', 'STUDY_PROGRAM_COORDINATOR', 'column', 'NOTIFICATION_EXPIRATION_COUNT'
go

/*=====*/
/* Index: FOR_STUDY_PROGRAM_FK */
/*=====*/
create index FOR_STUDY_PROGRAM_FK on STUDY_PROGRAM_COORDINATOR (
    STUDY_PROGRAM_ID ASC
)
go

/*=====*/
/* Index: COORDAINTOR_FK */
/*=====*/
create index COORDAINTOR_FK on STUDY_PROGRAM_COORDINATOR (

```

```

PARTICIPANT_ID ASC
)
go

/*=====*/
/* Index: COORDINATOR_TITLE_FK */
/*=====*/
create index COORDINATOR_TITLE_FK on STUDY_PROGRAM_COORDINATOR (
ACADEMIC_TITLE_ID ASC
)
go

/*=====*/
/* Table: SUBJECT_NOT_FOR_RECOGNITION */
/*=====*/
create table SUBJECT_NOT_FOR_RECOGNITION (
    ID          bigint          identity,
    DOCUMENT_FOR_RECOGNITION_REPORT_ID bigint          not null,
    PASSED_SUBJECT_NAME varchar(8000)          null,
    constraint PK_SUBJECT_NOT_FOR_RECOGNITION primary key nonclustered (ID)
)
go

/*=====*/
/* Index: DOCUMENT_SUBJECT_NOT_FOR_RECOGNITION_FK */
/*=====*/
create index DOCUMENT_SUBJECT_NOT_FOR_RECOGNITION_FK on SUBJECT_NOT_FOR_RECOGNITION (
DOCUMENT_FOR_RECOGNITION_REPORT_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION */
/*=====*/
create table SUBMISSION (
    ID          bigint          identity,
    MODULE_ID    bigint          null,
    HE_INSTITUTION_ID bigint          null,
    PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_ID bigint          null,

```

RESIDENCE_COUNTRY_ID bigint null,
 PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_ID bigint null,
 STUDY_PROGRAM_ID bigint null,
 COUNTRY_OF_BIRTH_ID bigint null,
 SUBMISSION_STATUS_ID bigint not null,
 PARTICIPANT_ROLE_ID bigint not null,
 SUBMISSION_PURPOSE_ID bigint null,
 CITIZENSHIP_ID bigint null,
 DOCUMENT_DELIVERY_COUNTRY_ID bigint null,
 CANDIDATE_ID bigint not null,
 CREATION_DATE datetime not null,
 CASE_NUMBER varchar(8000) null,
 DEADLINE_FOR_PASSED_EXAMS datetime null,
 FIRSTNAME varchar(8000) null,
 LASTNAME varchar(8000) null,
 PARENT_NAME varchar(8000) null,
 MAIDEN_NAME varchar(8000) null,
 DATE_OF_BIRTH datetime null,
 PLACE_OF_BIRTH varchar(8000) null,
 GENDER varchar(200) null default 'not_specified'
 constraint KC_GENDER_SUBMISSI check (GENDER is null or (GENDER in ('male','female','other','not_specified'))),
 RESIDENCE_ADDRESS varchar(8000) null,
 RESIDENCE_CITY varchar(8000) null,
 RESIDENCE_POSTAL_CODE varchar(8000) null,
 DOCUMENT_DELIVERY_ADDRESS varchar(8000) null,
 DOCUMENT_DELIVERY_CITY varchar(8000) null,
 DOCUMENT_DELIVERY_POSTAL_CODE varchar(8000) null,
 CONTACT_PHONE varchar(8000) null,
 CONTACT_EMAIL varchar(8000) null,
 FIRSTNAME_TRANSCRIBED varchar(8000) null,
 LASTNAME_TRANSCRIBED varchar(8000) null,
 PARENT_NAME_TRANSCRIBED varchar(8000) null,
 MAIDEN_NAME_TRANSCRIBED varchar(8000) null,
 CITY_OF_BIRTH_TRANSCRIBED varchar(8000) null,
 PREVIOUS_SECONDARY_INSTITUTION varchar(8000) null,
 PREVIOUS_SECONDARY_INSTITUTION_HEADQUARTERS varchar(8000) null,
 PREVIOUS_SECONDARY_OF_OBTAINED_QUALIFICATION varchar(8000) null,
 PREVIOUS_SECONDARY_YEAR_ENROLLMENT int null,
 PREVIOUS_SECONDARY_YEAR_COMPLETION int null,

```

PREVIOUS_UNIVERSITY_INSTITUTION varchar(8000)    null,
PREVIOUS_UNIVERSITY_INSTITUTION_HEADQUARTERS varchar(8000)    null,
PREVIOUS_UNIVERSITY_OF_OBTAINED_QUALIFICATION varchar(8000)    null,
PREVIOUS_UNIVERSITY_YEAR_ENROLLMENT int          null,
PREVIOUS_UNIVERSITY_YEAR_COMPLETION int          null,
SUBMISSION_DATE      datetime          null,
RESIDENCE_ADDRESS_TRANSCRIBED varchar(8000)    null,
RESIDENCE_CITY_TRANSCRIBED varchar(8000)    null,
DOCUMENT_DELIVERY_ADDRESS_TRANSCRIBED varchar(8000)    null,
DOCUMENT_DELIVERY_CITY_TRANSCRIBED varchar(8000)    null,
PREVIOUS_SECONDARY_INSTITUTION_HEADQUARTERS_TRANSCRIBED varchar(8000)    null,
PREVIOUS_UNIVERSITY_INSTITUTION_HEADQUARTERS_TRANSCRIBED varchar(8000)    null,
PREVIOUS_UNIVERSITY_EDUCATION bit              null,
PREVIOUS_UNIVERSITY_INSTITUTION_TRANSCRIBED varchar(8000)    null,
DECISION_COLLECTION_DATE datetime          null,
DEADLINE_FOR_PASSED_ADDITIONAL_EXAMS_NOTIFICATIONS_COUNT bigint          null,
constraint PK_SUBMISSION primary key nonclustered (ID)
)
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Submission request',
    'user', @CurrentUser, 'table', 'SUBMISSION'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Generated by the system when the submission is submitted by Candidate',
    'user', @CurrentUser, 'table', 'SUBMISSION', 'column', 'CASE_NUMBER'
go

```

```

/*=====*/
/* Index: FOR_CANDIDATE_FK */
/*=====*/
create index FOR_CANDIDATE_FK on SUBMISSION (
CANDIDATE_ID ASC

```

```

)
go

/*=====*/
/* Index: SUBMITTED_BY_FK */
/*=====*/
create index SUBMITTED_BY_FK on SUBMISSION (
PARTICIPANT_ROLE_ID ASC
)
go

/*=====*/
/* Index: CURRENT_STATUS_FK */
/*=====*/
create index CURRENT_STATUS_FK on SUBMISSION (
SUBMISSION_STATUS_ID ASC
)
go

/*=====*/
/* Index: CURRENT_STUDY_PROGRAM_FK */
/*=====*/
create index CURRENT_STUDY_PROGRAM_FK on SUBMISSION (
STUDY_PROGRAM_ID ASC
)
go

/*=====*/
/* Index: PURPOSE_OF_SUBMISSION_FK */
/*=====*/
create index PURPOSE_OF_SUBMISSION_FK on SUBMISSION (
SUBMISSION_PURPOSE_ID ASC
)
go

/*=====*/
/* Index: SUBMISSION_FOR_INSTITUTION_FK */
/*=====*/
create index SUBMISSION_FOR_INSTITUTION_FK on SUBMISSION (

```

HE_INSTITUTION_ID ASC

)

go

/*=====*/

/* Index: FOR_MODULE_FK */

/*=====*/

create index FOR_MODULE_FK on SUBMISSION (

MODULE_ID ASC

)

go

/*=====*/

/* Index: COUNTRY_OF_BIRTH_FK */

/*=====*/

create index COUNTRY_OF_BIRTH_FK on SUBMISSION (

COUNTRY_OF_BIRTH_ID ASC

)

go

/*=====*/

/* Index: CITIZENSHIP_FK */

/*=====*/

create index CITIZENSHIP_FK on SUBMISSION (

CITIZENSHIP_ID ASC

)

go

/*=====*/

/* Index: RESIDENCE_COUNTRY_FK */

/*=====*/

create index RESIDENCE_COUNTRY_FK on SUBMISSION (

RESIDENCE_COUNTRY_ID ASC

)

go

/*=====*/

/* Index: DOCUMENT_DELIVERY_COUNTRY_FK */

/*=====*/

```

create index DOCUMENT_DELIVERY_COUNTRY_FK on SUBMISSION (
DOCUMENT_DELIVERY_COUNTRY_ID ASC
)
go

/*=====*/
/* Index: PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_FK */
/*=====*/
create index PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_FK on SUBMISSION (
PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_ID ASC
)
go

/*=====*/
/* Index: PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_FK */
/*=====*/
create index PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_FK on SUBMISSION (
PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_DOCUMENT */
/*=====*/
create table SUBMISSION_DOCUMENT (
ID          bigint          identity,
SUBMISSION_ID  bigint          not null,
REQUEST_FOR_UPDATE_ID bigint          null,
SUBMISSION_NEW_DOCUMENT_ID bigint          null,
STATUS        varchar(200)    not null
constraint CKC_STATUS_SUBMISSI_DOCUMENT check (STATUS in ('A','T','D')),
COMMENT       varchar(8000)    null,
OPERATION_TYPE varchar(200)    null
constraint CKC_OPERATION_TYPE_SUBMISSI check (OPERATION_TYPE is null or (OPERATION_TYPE in ('created','deleted'))),
constraint PK_SUBMISSION_DOCUMENT primary key nonclustered (ID)
)
go

declare @CurrentUser sysname

```

```

select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Request to change a document with new version
    Both document versions must be from the same metadata field',
    'user', @CurrentUser, 'table', 'SUBMISSION_DOCUMENT'
go

/*=====*/
/* Index: SUBMISSION_DOCUMENTS_FK */
/*=====*/
create index SUBMISSION_DOCUMENTS_FK on SUBMISSION_DOCUMENT (
SUBMISSION_ID ASC
)
go

/*=====*/
/* Index: REQUEST_FOR_UPDATE_DOCUMENT_FK */
/*=====*/
create index REQUEST_FOR_UPDATE_DOCUMENT_FK on SUBMISSION_DOCUMENT (
REQUEST_FOR_UPDATE_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_DOCUMENT_DFR */
/*=====*/
create table SUBMISSION_DOCUMENT_DFR (
    ID          bigint          identity,
    DOCUMENT_FOR_RECOGNITION_ID bigint          not null,
    REQUEST_FOR_UPDATE_ID bigint          null,
    SUBMISSION_NEW_DOCUMENT_VERSION_ID bigint          null,
    STATUS      varchar(200)     not null
        constraint CKC_STATUS_SUBMISSION_DFR check (STATUS in ('A','T','D')),
    COMMENT     varchar(8000)    null,
    OPERATION_TYPE  varchar(200)  null
        constraint CKC_OPERATION_TYPE_SUBMISSI_DFR check (OPERATION_TYPE is null or (OPERATION_TYPE in
('created','deleted'))),
    constraint PK_SUBMISSION_DOCUMENT_DFR primary key nonclustered (ID)
)

```

```

go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'DFR - Document for Recognition',
    'user', @CurrentUser, 'table', 'SUBMISSION_DOCUMENT_DFR'
go

/*=====*/
/* Index: REQUEST_FOR_UPDATE_DOCUMENT_DFR_FK          */
/*=====*/
create index REQUEST_FOR_UPDATE_DOCUMENT_DFR_FK on SUBMISSION_DOCUMENT_DFR (
    REQUEST_FOR_UPDATE_ID ASC
)
go

/*=====*/
/* Index: SUBMISSION_DOCUMENT_FOR_DFR_FK              */
/*=====*/
create index SUBMISSION_DOCUMENT_FOR_DFR_FK on SUBMISSION_DOCUMENT_DFR (
    DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

/*=====*/
/* Index: NEW_DOCUMENT_DFR_FK                        */
/*=====*/
create index NEW_DOCUMENT_DFR_FK on SUBMISSION_DOCUMENT_DFR (
    SUBMISSION_NEW_DOCUMENT_VERSION_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_DOCUMENT_VERSION                */
/*=====*/
create table SUBMISSION_DOCUMENT_VERSION (
    ID          bigint          identity,
    METADATA_FIELD_ID  bigint          not null,

```

```

SUBMISSION_ID      bigint      null,
DOCUMENT_FOR_RECOGNITION_ID bigint      null,
DOCUMENT_NUMBER     varchar(8000)  null,
COMMENT            varchar(8000)  null,
CREATION_DATE       datetime      not null,
CONTENT            image          null,
CONTENT_TYPE        varchar(8000)  null,
FILE_NAME           varchar(8000)  null,
FILE_SIZE           bigint         null,
constraint PK_SUBMISSION_DOCUMENT_VERSION primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'For documents generated by the system',
    'user', @CurrentUser, 'table', 'SUBMISSION_DOCUMENT_VERSION', 'column', 'DOCUMENT_NUMBER'
go

/*=====*/
/* Index: DOCUMENT_METATYPE_FK */
/*=====*/
create index DOCUMENT_METATYPE_FK on SUBMISSION_DOCUMENT_VERSION (
METADATA_FIELD_ID ASC
)
go

/*=====*/
/* Index: UPLOADED_DOCUMENTS_FK */
/*=====*/
create index UPLOADED_DOCUMENTS_FK on SUBMISSION_DOCUMENT_VERSION (
SUBMISSION_ID ASC
)
go

/*=====*/
/* Index: UPLOADED_DOCUMENTS_FOR_DFR_FK */
/*=====*/

```

```

create index UPLOADED_DOCUMENTS_FOR_DFR_FK on SUBMISSION_DOCUMENT_VERSION (
DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_FIELD */
/*=====*/
create table SUBMISSION_FIELD (
ID          bigint          identity,
SUBMISSION_ID  bigint          not null,
REQUEST_FOR_UPDATE_ID bigint          null,
METADATA_FIELD_ID  bigint          not null,
NEW_FIELD_VALUE  varchar(8000)  null,
OLD_FIELD_VALUE  varchar(8000)  null,
STATUS         varchar(200)    not null
        constraint CKC_STATUS_SUBMISSION check (STATUS in ('A','T','D')),
COMMENT        varchar(8000)    null,
        constraint PK_SUBMISSION_FIELD primary key nonclustered (ID)
)
go

/*=====*/
/* Index: SUBMISSION_DATA_FK */
/*=====*/
create index SUBMISSION_DATA_FK on SUBMISSION_FIELD (
SUBMISSION_ID ASC
)
go

/*=====*/
/* Index: METADESCRIPTION_FK */
/*=====*/
create index METADESCRIPTION_FK on SUBMISSION_FIELD (
METADATA_FIELD_ID ASC
)
go

/*=====*/

```

```

/* Index: REQUEST_FOR_UPDATE_FIELD_FK          */
/*=====*/
create index REQUEST_FOR_UPDATE_FIELD_FK on SUBMISSION_FIELD (
REQUEST_FOR_UPDATE_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_FIELD_DFR                  */
/*=====*/
create table SUBMISSION_FIELD_DFR (
    ID          bigint          identity,
    REQUEST_FOR_UPDATE_ID bigint          null,
    DOCUMENT_FOR_RECOGNITION_ID bigint      not null,
    METADATA_FIELD_ID  bigint          not null,
    NEW_FIELD_VALUE    varchar(8000)      null,
    OLD_FIELD_VALUE    varchar(8000)      null,
    STATUS            varchar(200)        not null
    constraint CKC_STATUS_SUBMISSION_FIELD_DFR check (STATUS in ('A','T','D')),
    COMMENT            varchar(8000)      null,
    constraint PK_SUBMISSION_FIELD_DFR primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'DFR - Document for Recognition',
    'user', @CurrentUser, 'table', 'SUBMISSION_FIELD_DFR'
go

/*=====*/
/* Index: REQUEST_FOR_UPDATE_FIELD_DFR_FK      */
/*=====*/
create index REQUEST_FOR_UPDATE_FIELD_DFR_FK on SUBMISSION_FIELD_DFR (
REQUEST_FOR_UPDATE_ID ASC
)
go

```

```

/*=====*/
/* Index: SUBMISSION_FIELD_FOR_DFR_FK          */
/*=====*/
create index SUBMISSION_FIELD_FOR_DFR_FK on SUBMISSION_FIELD_DFR (
DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

/*=====*/
/* Index: METADESCRIPTION_DFR_FK              */
/*=====*/
create index METADESCRIPTION_DFR_FK on SUBMISSION_FIELD_DFR (
METADATA_FIELD_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_LIFECYCLE                 */
/*=====*/
create table SUBMISSION_LIFECYCLE (
    ID          bigint          identity,
    DOCUMENT_FOR_RECOGNITION_ID bigint          null,
    ENIC_NARIC_REPLY_ID  bigint          null,
    SUBMISSION_ID    bigint          not null,
    PARTICIPANT_ROLE_CHANGED_BY_ID bigint          not null,
    SUBMISSION_STATUS_ID bigint          not null,
    PARTICIPANT_ROLE_OWNER_ID bigint          not null,
    LIFECYCLE_DATE      datetime          not null,
    COMMENT            varchar(8000)      null,
    NOTIFICATION_SENT_COUNT bigint          null,
    NOTIFICATOIN_SENT_DATE datetime          null,
    DEADLINE_DAYS       bigint          null,
    ENIC_SUBNUMBER      varchar(8000)      null,
    constraint PK_SUBMISSION_LIFECYCLE primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()

```

```

execute sp_addextendedproperty 'MS_Description',
    'Created
    Filling data
    Processed by clerk
    Delegated to Vice Dean and then to SP Coordinator
    ...',
    'user', @CurrentUser, 'table', 'SUBMISSION_LIFECYCLE'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'Deadline days copied from deadline days from submission status by default',
    'user', @CurrentUser, 'table', 'SUBMISSION_LIFECYCLE', 'column', 'DEADLINE_DAYS'
go

```

```

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
    'ENIC number (subnumber) for the requested candidate submission',
    'user', @CurrentUser, 'table', 'SUBMISSION_LIFECYCLE', 'column', 'ENIC_SUBNUMBER'
go

```

```

/*=====*/
/* Index: LIFECYCLE_FK */
/*=====*/
create index LIFECYCLE_FK on SUBMISSION_LIFECYCLE (
SUBMISSION_ID ASC
)
go

```

```

/*=====*/
/* Index: STATUS_FK */
/*=====*/
create index STATUS_FK on SUBMISSION_LIFECYCLE (
SUBMISSION_STATUS_ID ASC
)
go

```

```

/*=====*/
/* Index: OWNER_FK */
/*=====*/
create index OWNER_FK on SUBMISSION_LIFECYCLE (
PARTICIPANT_ROLE_OWNER_ID ASC
)
go

/*=====*/
/* Index: CHANGED_BY_FK */
/*=====*/
create index CHANGED_BY_FK on SUBMISSION_LIFECYCLE (
PARTICIPANT_ROLE_CHANGED_BY_ID ASC
)
go

/*=====*/
/* Index: FROM_ENIC_FK */
/*=====*/
create index FROM_ENIC_FK on SUBMISSION_LIFECYCLE (
ENIC_NARIC_REPLY_ID ASC
)
go

/*=====*/
/* Index: ENIC_FOR_DOCUMENT_FOR_RECOGNITION_FK */
/*=====*/
create index ENIC_FOR_DOCUMENT_FOR_RECOGNITION_FK on SUBMISSION_LIFECYCLE (
DOCUMENT_FOR_RECOGNITION_ID ASC
)
go

/*=====*/
/* Table: SUBMISSION_PURPOSE */
/*=====*/
create table SUBMISSION_PURPOSE (
ID          bigint          identity,
CODE        varchar(200)    not null,
NAME        varchar(8000)   not null,

```

```

NAME_ENG      varchar(8000)    null,
COLOR         varchar(8000)    null,
DESCRIPTION    varchar(8000)    null,
DESCRIPTION_ENG varchar(8000)    null,
constraint PK_SUBMISSION_PURPOSE primary key nonclustered (ID)
)
go

/*=====*/
/* Index: SUBMISSION_PURPOSE_CODE_IND          */
/*=====*/
create unique index SUBMISSION_PURPOSE_CODE_IND on SUBMISSION_PURPOSE (
CODE ASC
)
go

/*=====*/
/* Table: SUBMISSION_STATUS                    */
/*=====*/
create table SUBMISSION_STATUS (
ID          bigint          identity,
CODE        varchar(200)    not null,
NAME        varchar(8000)    not null,
DESCRIPTION  varchar(8000)    null,
NAME_ENG    varchar(8000)    null,
DESCRIPTION_ENG varchar(8000) null,
COLOR       varchar(8000)    null,
STATUS_ORDER bigint         not null default 0,
NOTIFICATION_TRIGGER_DAYS bigint not null default 0,
DEADLINE_DAYS bigint         null,
constraint PK_SUBMISSION_STATUS primary key nonclustered (ID)
)
go

declare @CurrentUser sysname
select @CurrentUser = user_name()
execute sp_addextendedproperty 'MS_Description',
'Enumeration',
'user', @CurrentUser, 'table', 'SUBMISSION_STATUS'

```

```

go

/*=====*/
/* Index: SUBMISSION_STATUS_CODE_IND          */
/*=====*/
create unique index SUBMISSION_STATUS_CODE_IND on SUBMISSION_STATUS (
CODE ASC
)
go

/*=====*/
/* Table: "USER"                               */
/*=====*/
create table "USER" (
    ID          bigint          identity,
    PARTICIPANT_ID  bigint          null,
    USERNAME      varchar(256)    not null
        constraint KKC_USERNAME_USER check (USERNAME = lower(USERNAME)),
    PASSWORD      varchar(8000)   not null,
    ACTIVE        bit            not null,
    EMAIL         varchar(256)    not null,
    REGISTRATION_DATE  datetime    not null,
    ACTIVATION_DATE  datetime      null,
    DELETED        bit            not null default 0,
    BLOCKED        bit            not null default 0,
    BLOCKED_REASON  varchar(8000)  null,
    BLOCKED_DATE    datetime      null,
    constraint PK_USER primary key nonclustered (ID)
)
go

/*=====*/
/* Index: USER_ACCOUNT_FK                     */
/*=====*/
create index USER_ACCOUNT_FK on "USER" (
PARTICIPANT_ID ASC
)
go

```

```

/*=====*/
/* Index: USER_USERNAME_IND */
/*=====*/
create unique index USER_USERNAME_IND on "USER" (
  USERNAME ASC
)
go

alter table ACADEMIC_BOARD
  add constraint FK_ACADEMIC_BOARD_FIE_SCIENTIF foreign key (SCIENTIFIC_ARTISTIC_FIELD_ID)
    references SCIENTIFIC_ARTISTIC_FIELD (ID)
go

alter table ACADEMIC_BOARD
  add constraint FK_ACADEMIC_PARTICIPA_PARTICIP foreign key (ID)
    references PARTICIPANT (ID)
go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE
  add constraint FK_ACADEMIC_AB_DECISION_DOC foreign key (ACADEMIC_BOARD_DECISION_DOCUMENT_VERSION_ID)
    references SUBMISSION_DOCUMENT_VERSION (ID)
go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE
  add constraint FK_ACADEMIC_ASSIGNED__ACADEMIC foreign key (ACADEMIC_BOARD_ID)
    references ACADEMIC_BOARD (ID)
go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE
  add constraint FK_ACADEMIC_BOARDS_IN_SUBMISSI foreign key (ACADEMIC_BOARD_LIFECYCLE_ID)
    references SUBMISSION_LIFECYCLE (ID)
go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE
  add constraint FK_ACADEMIC_DECISION_STATUS foreign key (DECISION_SUBMISSION_STATUS_ID)
    references SUBMISSION_STATUS (ID)
go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE

```

```

add constraint FK_ACADEMIC_FINAL_DECISION_DOC foreign key (FINAL_DECISION_DOCUMENT_VERSION_ID)
references SUBMISSION_DOCUMENT_VERSION (ID)

go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE

add constraint FK_ACADEMIC_FORWARDED_STATUS foreign key (FORWARDED_SUBMISSION_STATUS_ID)
references SUBMISSION_STATUS (ID)

go

alter table ACADEMIC_BOARD_FOR_LIFECYCLE

add constraint FK_ACADEMIC_PROPOSED_DECISION_DOC foreign key (PROPOSED_DECISION_DOCUMENT_VERSION_ID)
references SUBMISSION_DOCUMENT_VERSION (ID)

go

alter table ADDITIONAL_CONDITION

add constraint FK_ADDITION_CONDITIONS_DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_ID)
references DOCUMENT_FOR_RECOGNITION (ID)

go

alter table ADDITIONAL_CONDITION

add constraint FK_ADDITION_ADDITIONA_DFR foreign key (DOCUMENT_FOR_RECOGNITION_REPORT_ID)
references DOCUMENT_FOR_RECOGNITION_REPORT (ID)

go

alter table ADDITIONAL_DOCUMENT

add constraint FK_ADDITIONAL_DOCUMENT_TYPE foreign key (ADDITIONAL_DOCUMENT_TYPE_ID)
references ADDITIONAL_DOCUMENT_TYPE (ID)

go

alter table ADDITIONAL_DOCUMENT

add constraint FK_ADDITION_ADDITIONA_PARTICIP foreign key (PARTICIPANT_ROLE_ID)
references PARTICIPANT_ROLE (ID)

go

alter table ADDITIONAL_DOCUMENT

add constraint FK_ADDITIONAL_DOCUMENT_DOCUMENT foreign key (SUBMISSION_ID)
references SUBMISSION (ID)

go

```

```
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  add constraint FK_ADDITION_ADDITIONA_RECOGNIT foreign key (RECOGNITION_REPORT_ID)
    references RECOGNITION_REPORT (ID)
go
```

```
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  add constraint FK_ADDITION_CERTIFICA_SUBMISSI foreign key (SUBMISSION_DOCUMENT_VERSION_ID)
    references SUBMISSION_DOCUMENT_VERSION (ID)
go
```

```
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  add constraint FK_ADDITION_INSTITUTI_HE_INSTI foreign key (HE_INSTITUTION_ID)
    references HE_INSTITUTION (ID)
go
```

```
alter table ADDITIONAL_EXAM_MUST_BE_PASSED
  add constraint FK_ADDITION_STUDY_PRO_STUDY_PR foreign key (STUDY_PROGRAM_ID)
    references STUDY_PROGRAM (ID)
go
```

```
alter table ADDITIONAL_INSTITUTION_FIELD
  add constraint FK_ADDITION_ADDITIONA_METADATA foreign key (METADATA_FIELD_ID)
    references METADATA_FIELD (ID)
go
```

```
alter table ADDITIONAL_INSTITUTION_FIELD
  add constraint FK_ADDITION_FIELD_FOR_ADDITION foreign key (ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE_ID)
    references ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE (ID)
go
```

```
alter table ADDITIONAL_INSTITUTION_FIELD
  add constraint FK_ADDITION_REQUEST_F_REQUEST_ foreign key (REQUEST_FOR_UPDATE_ID)
    references REQUEST_FOR_UPDATE (ID)
go
```

```
alter table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE
  add constraint FK_ADDITION_ADDITIONA_DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_ID)
    references DOCUMENT_FOR_RECOGNITION (ID)
go
```

```
alter table ADDITIONAL_INSTITUTION_FOR_JOINT_DEGREE
  add constraint FK_ADDITION_ADDITIONA_COUNTRY foreign key (COUNTRY_ID)
    references COUNTRY (ID)
go
```

```
alter table AGENCY
  add constraint FK_AGENCY_DELIVERY_DOCUMENT_COUNTRY foreign key (DELIVERY_DOCUMENT_COUNTRY_ID)
    references COUNTRY (ID)
go
```

```
alter table AGENCY
  add constraint FK_AGENCY_OFFICE_COUNTRY foreign key (OFFICE_COUNTRY_ID)
    references COUNTRY (ID)
go
```

```
alter table AGENCY
  add constraint FK_AGENCY_PARTICIPA_PARTICIP foreign key (PARTICIPANT_ID)
    references PARTICIPANT (ID)
go
```

```
alter table "AUTHORIZATION"
  add constraint FK_AUTHORIZ_AGENCY_AGENCY foreign key (PARTICIPANT_ID)
    references AGENCY (PARTICIPANT_ID)
    on update cascade on delete cascade
go
```

```
alter table "AUTHORIZATION"
  add constraint FK_AUTHORIZ_CANDIDATE_CANDIDAT foreign key (ID)
    references CANDIDATE (ID)
    on update cascade on delete cascade
go
```

```
alter table BOARD_MEMBER
  add constraint FK_BOARD_ME_MEMBER_AT_ACADEMIC foreign key (ACADEMIC_BOARD_ID)
    references ACADEMIC_BOARD (ID)
go
```

```
alter table BOARD_MEMBER
```

```

add constraint FK_BOARD_ME_MEMBER_PO_BOARD_PO foreign key (BOARD_POSITION_ID)
references BOARD_POSITION (ID)

go

alter table BOARD_MEMBER
add constraint FK_BOARD_ME_PARTICIPA_PARTICIP foreign key (PARTICIPANT_ID)
references PARTICIPANT (ID)

go

alter table CANDIDATE
add constraint FK_CANDIDATE_CITIZENSHIP foreign key (CITIZENSHIP_ID)
references COUNTRY (ID)

go

alter table CANDIDATE
add constraint FK_CANDIDATE_COUNTRY_OF_BIRTH foreign key (COUNTRY_OF_BIRTH_ID)
references COUNTRY (ID)

go

alter table CANDIDATE
add constraint FK_CANDIDATE_DOCUMENT_DELIVERY_COUNTRY foreign key (DOCUMENT_DELIVERY_COUNTRY_ID)
references COUNTRY (ID)

go

alter table CANDIDATE
add constraint FK_CANDIDAT_PARTICIPA_PARTICIP foreign key (ID)
references PARTICIPANT (ID)

go

alter table CANDIDATE
add constraint FK_CANDIDATE_RESIDENCE_COUNTRY foreign key (RESIDENCE_COUNTRY_ID)
references COUNTRY (ID)

go

alter table CASE_NUMBER_INCREMENTER
add constraint FK_CASE_NUM_INCREMENT_SUBMISSI foreign key (SUBMISSION_PURPOSE_ID)
references SUBMISSION_PURPOSE (ID)

go

```

```

alter table COUNTRY
  add constraint FK_COUNTRY_IN_CONTIN_CONTINEN foreign key (CONTINENT_ID)
    references CONTINENT (ID)
go

alter table DOCUMENT_FOR_RECOGNITION
  add constraint FK_DR_DOCUMENT_ADMISSION_COUNTRY foreign key (DOCUMENT_ADMISSION_COUNTRY_ID)
    references COUNTRY (ID)
go

alter table DOCUMENT_FOR_RECOGNITION
  add constraint FK_DOCUMENT_DOCUMENT__SUBMISSI foreign key (SUBMISSION_ID)
    references SUBMISSION (ID)
go

alter table DOCUMENT_FOR_RECOGNITION_REPORT
  add constraint FK_DOCUMENT_COMPARISO_STUDY_LE foreign key (STUDY_LEVEL_ID)
    references STUDY_LEVEL (ID)
go

alter table DOCUMENT_FOR_RECOGNITION_REPORT
  add constraint FK_DOCUMENT_DFRS_IN_R_RECOGNIT foreign key (RECOGNITION_REPORT_ID)
    references RECOGNITION_REPORT (ID)
go

alter table DOCUMENT_FOR_RECOGNITION_REPORT
  add constraint FK_DOCUMENT_DFR_REPOR_ENIC_NAR foreign key (ENIC_NARIC_REPLY_ID)
    references ENIC_NARIC_REPLY (ID)
go

alter table DOCUMENT_FOR_RECOGNITION_REPORT
  add constraint FK_DOCUMENT_DFR_REPOR_COUNTRY foreign key (COUNTRY_ID)
    references COUNTRY (ID)
go

alter table DOCUMENT_FOR_RECOGNITION_REPORT
  add constraint FK_DOCUMENT_DOCUMENT__DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_ID)
    references DOCUMENT_FOR_RECOGNITION (ID)
go

```

```
alter table DOCUMENT_METADATA
```

```
add constraint FK_DOCUMENT_FOR_DOCUM_METADATA foreign key (METADATA_FIELD_ID)
references METADATA_FIELD (ID)
```

```
go
```

```
alter table DOCUMENT_METADATA
```

```
add constraint FK_DOCUMENT_METADATA_SUBMISSI foreign key (SUBMISSION_DOCUMENT_VERSION_ID)
references SUBMISSION_DOCUMENT_VERSION (ID)
on update cascade on delete cascade
```

```
go
```

```
alter table ENIC_EMAIL_REQUEST
```

```
add constraint FK_ENIC_EMA_EMAIL_REQ_ENIC_NAR foreign key (ENIC_NARIC_REPLY_ID)
references ENIC_NARIC_REPLY (ID)
```

```
go
```

```
alter table ENIC_EMAIL_REQUEST
```

```
add constraint FK_ENIC_EMA_ENIC_EMAIL_PARTICIP foreign key (PARTICIPANT_ROLE_ID)
references PARTICIPANT_ROLE (ID)
```

```
go
```

```
alter table ENIC_NARIC_REPLY
```

```
add constraint FK_ENIC_NARIC_CREATED_BY foreign key (CREATED_BY_PARTICIPANT_ID)
references PARTICIPANT (ID)
```

```
go
```

```
alter table ENIC_NARIC_REPLY
```

```
add constraint FK_ENIC_NARIC_REQUESTED_BY foreign key (REQUEST_BY_PARTICIPANT_ID)
references PARTICIPANT (ID)
```

```
go
```

```
alter table ENIC_NARIC_REPLY
```

```
add constraint FK_ENIC_NAR_REPLY_FOR_COUNTRY foreign key (COUNTRY_ID)
references COUNTRY (ID)
```

```
go
```

```
alter table INSTITUTION_MEMBER
```

```
add constraint FK_INSTITUT_BY_INSTIT_INSTITUT foreign key (INSTITUTION_FUNCTION_ID)
```

```

        references INSTITUTION_FUNCTION (ID)

go

alter table INSTITUTION_MEMBER
    add constraint FK_INSTITUT_MEMBER_AT_HE_INSTI foreign key (HE_INSTITUTION_ID)
        references HE_INSTITUTION (ID)

go

alter table INSTITUTION_MEMBER
    add constraint FK_INSTITUT_MEMBER_TI_ACADEMIC foreign key (ACADEMIC_TITLE_ID)
        references ACADEMIC_TITLE (ID)

go

alter table INSTITUTION_MEMBER
    add constraint FK_INSTITUT_PARTICIPA_PARTICIP foreign key (PARTICIPANT_ID)
        references PARTICIPANT (ID)

go

alter table METADATA_FIELD_BY_PURPOSE
    add constraint FK_METADATA_METADATA__METADATA foreign key (METADATA_FIELD_ID)
        references METADATA_FIELD (ID)

go

alter table METADATA_FIELD_BY_PURPOSE
    add constraint FK_METADATA_SUBMISSIO_SUBMISSI foreign key (ID)
        references SUBMISSION_PURPOSE (ID)

go

alter table MODULE
    add constraint FK_MODULE_MODULES_STUDY_PR foreign key (STUDY_PROGRAM_ID)
        references STUDY_PROGRAM (ID)

go

alter table PARTICIPANT
    add constraint FK_PARTICIP_AFFILIATE_HE_INSTI foreign key (HE_INSTITUTION_ID)
        references HE_INSTITUTION (ID)

go

alter table PARTICIPANT

```

```

add constraint FK_PARTICIP_PARTICIPA_ACADEMIC foreign key (ACADEMIC_TITLE_ID)
references ACADEMIC_TITLE (ID)

go

```

```

alter table PARTICIPANT_ROLE

add constraint FK_PARTICIP_PARTICIPA_PARTICIP foreign key (PARTICIPANT_ID)
references PARTICIPANT (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RECOGNIT_ACADEMIC__ACADEMIC foreign key (ACADEMIC_TITLE_ID)
references ACADEMIC_TITLE (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RECOGNIT_FOR_STUDY_STUDY_PR foreign key (STUDY_PROGRAM_ID)
references STUDY_PROGRAM (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RECOGNIT_INSTITUTI_HE_INSTI foreign key (HE_INSTITUTION_ID)
references HE_INSTITUTION (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RR_INSTITUTION_COUNTRY foreign key (INSTITUTION_COUNTRY_ID)
references COUNTRY (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RR_PREV_EDUCATION_INSTITUTION_COUNTRY foreign key
(PREVIOUS_EDUCATION_INSTITUTION_COUNTRY_ID)
references COUNTRY (ID)

go

```

```

alter table RECOGNITION_REPORT

add constraint FK_RECOGNIT_RECOGNITI_PARTICIP foreign key (PARTICIPANT_ID)
references PARTICIPANT (ID)

go

```

```
alter table RECOGNITION_REPORT
```

```
add constraint FK_RECOGNIT_REPORT_CA_COUNTRY foreign key (COUNTRY_CITIZENSHIP)  
references COUNTRY (ID)
```

```
go
```

```
alter table RECOGNITION_REPORT
```

```
add constraint FK_RECOGNIT_SUBMISSIO_SUBMISSI_PURPOSE foreign key (SUBMISSION_PURPOSE_ID)  
references SUBMISSION_PURPOSE (ID)
```

```
go
```

```
alter table RECOGNITION_REPORT
```

```
add constraint FK_RECOGNIT_SUBMISSIO_SUBMISSI_REPORT foreign key (SUBMISSION_ID)  
references SUBMISSION (ID)
```

```
go
```

```
alter table RECOGNITION_SUGGESTION
```

```
add constraint FK_RECOGNIT_DOCUMENT__DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_REPORT_ID)  
references DOCUMENT_FOR_RECOGNITION_REPORT (ID)
```

```
go
```

```
alter table REQUEST_FOR_UPDATE
```

```
add constraint FK_REQUEST_CREATED_BY_ROLE foreign key (CREATED_BY_ROLE_ID)  
references PARTICIPANT_ROLE (ID)
```

```
go
```

```
alter table REQUEST_FOR_UPDATE
```

```
add constraint FK_REQUEST__FOR_UPDAT_SUBMISSI foreign key (SUBMISSION_ID)  
references SUBMISSION (ID)
```

```
go
```

```
alter table REQUEST_FOR_UPDATE
```

```
add constraint FK_REQUEST__LIFECYCLE_SUBMISSI foreign key (SUBMISSION_LIFECYCLE_ID)  
references SUBMISSION_LIFECYCLE (ID)
```

```
go
```

```
alter table REQUEST_FOR_UPDATE
```

```
add constraint FK_REQUEST_PROCESSED_BY_ROLE foreign key (PROCESSED_BY_ROLE_ID)  
references PARTICIPANT_ROLE (ID)
```

go

alter table STUDY_PROGRAM

add constraint FK_STUDY_PR_LANGUAGE__LANGUAGE foreign key (LANGUAGE_ID)
references LANGUAGE (ID)

go

alter table STUDY_PROGRAM

add constraint FK_STUDY_PR_STUDY_PRO_SCIENTIF foreign key (SCIENTIFIC_ARTISTIC_FIELD_ID)
references SCIENTIFIC_ARTISTIC_FIELD (ID)

go

alter table STUDY_PROGRAM

add constraint FK_STUDY_PR_STUDY_PRO_HE_INSTI foreign key (HE_INSTITUTION_ID)
references HE_INSTITUTION (ID)

go

alter table STUDY_PROGRAM

add constraint FK_STUDY_PR_STUDY_PRO_STUDY_LE foreign key (STUDY_LEVEL_ID)
references STUDY_LEVEL (ID)

go

alter table STUDY_PROGRAM_COORDINATOR

add constraint FK_STUDY_PR_COORDAINT_PARTICIP foreign key (PARTICIPANT_ID)
references PARTICIPANT (ID)

go

alter table STUDY_PROGRAM_COORDINATOR

add constraint FK_STUDY_PR_COORDINAT_ACADEMIC foreign key (ACADEMIC_TITLE_ID)
references ACADEMIC_TITLE (ID)

go

alter table STUDY_PROGRAM_COORDINATOR

add constraint FK_STUDY_PR_FOR_STUDY_STUDY_PR foreign key (STUDY_PROGRAM_ID)
references STUDY_PROGRAM (ID)

go

alter table SUBJECT_NOT_FOR_RECOGNITION

add constraint FK_SUBJECT__DOCUMENT__DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_REPORT_ID)

```

        references DOCUMENT_FOR_RECOGNITION_REPORT (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSION_CITIZENSHIP foreign key (CITIZENSHIP_ID)
        references COUNTRY (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSION_COUNTRY_OF_BIRTH foreign key (COUNTRY_OF_BIRTH_ID)
        references COUNTRY (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSI_CURRENT_S_SUBMISSI foreign key (SUBMISSION_STATUS_ID)
        references SUBMISSION_STATUS (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSI_CURRENT_S_STUDY_PR foreign key (STUDY_PROGRAM_ID)
        references STUDY_PROGRAM (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSION_DOCUMENT_DELIVERY_COUNTRY foreign key (DOCUMENT_DELIVERY_COUNTRY_ID)
        references COUNTRY (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSI_FOR_CANDI_CANDIDAT foreign key (CANDIDATE_ID)
        references CANDIDATE (ID)

go

alter table SUBMISSION
    add constraint FK_SUBMISSI_FOR_MODUL_MODULE foreign key (MODULE_ID)
        references MODULE (ID)

go

alter table SUBMISSION

```

```

add constraint FK_SUBMISSION_PREVIOUS_SEC_INST_COUNTRY foreign key
(PREVIOUS_SECONDARY_INSTITUTION_COUNTRY_ID)
references COUNTRY (ID)

go

```

```

alter table SUBMISSION

add constraint FK_SUBMISSION_PREVIOUS_UNI_INST_COUNTRY foreign key
(PREVIOUS_UNIVERSITY_INSTITUTION_COUNTRY_ID)
references COUNTRY (ID)

go

```

```

alter table SUBMISSION

add constraint FK_SUBMISSI_PURPOSE__SUBMISSI foreign key (SUBMISSION_PURPOSE_ID)
references SUBMISSION_PURPOSE (ID)

go

```

```

alter table SUBMISSION

add constraint FK_SUBMISSION_RESIDENCE_COUNTRY foreign key (RESIDENCE_COUNTRY_ID)
references COUNTRY (ID)

go

```

```

alter table SUBMISSION

add constraint FK_SUBMISSI_SUBMISSIO_HE_INSTI foreign key (HE_INSTITUTION_ID)
references HE_INSTITUTION (ID)

go

```

```

alter table SUBMISSION

add constraint FK_SUBMISSI_SUBMITTED_PARTICIP foreign key (PARTICIPANT_ROLE_ID)
references PARTICIPANT_ROLE (ID)

go

```

```

alter table SUBMISSION_DOCUMENT

add constraint FK_SUBMISSION_NEW_DOCUMENT foreign key (SUBMISSION_NEW_DOCUMENT_ID)
references SUBMISSION_DOCUMENT_VERSION (ID)

go

```

```

alter table SUBMISSION_DOCUMENT

add constraint FK_SUBMISSI_REQUEST_F_REQUEST_UPD foreign key (REQUEST_FOR_UPDATE_ID)
references REQUEST_FOR_UPDATE (ID)

go

```

```
alter table SUBMISSION_DOCUMENT
```

```
add constraint FK_SUBMISSI_SUBMISSIO_SUBMISSI_DOC foreign key (SUBMISSION_ID)
```

```
references SUBMISSION (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_DFR
```

```
add constraint FK_SUBMISSI_NEW_DOCUM_SUBMISSI foreign key (SUBMISSION_NEW_DOCUMENT_VERSION_ID)
```

```
references SUBMISSION_DOCUMENT_VERSION (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_DFR
```

```
add constraint FK_SUBMISSION_REQUEST_DOCUMENT_DFR foreign key (REQUEST_FOR_UPDATE_ID)
```

```
references REQUEST_FOR_UPDATE (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_DFR
```

```
add constraint FK_SUBMISSION_DFR foreign key (DOCUMENT_FOR_RECOGNITION_ID)
```

```
references DOCUMENT_FOR_RECOGNITION (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_VERSION
```

```
add constraint FK_SUBMISSI_DOCUMENT__METADATA foreign key (METADATA_FIELD_ID)
```

```
references METADATA_FIELD (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_VERSION
```

```
add constraint FK_SUBMISSI_UPLOADED__SUBMISSI foreign key (SUBMISSION_ID)
```

```
references SUBMISSION (ID)
```

```
go
```

```
alter table SUBMISSION_DOCUMENT_VERSION
```

```
add constraint FK_SUBMISSI_UPLOADED_DOCUMENT_DFR foreign key (DOCUMENT_FOR_RECOGNITION_ID)
```

```
references DOCUMENT_FOR_RECOGNITION (ID)
```

```
go
```

```
alter table SUBMISSION_FIELD
```

```
add constraint FK_SUBMISSION_METADATA foreign key (METADATA_FIELD_ID)
```

```
references METADATA_FIELD (ID)
```

go

alter table SUBMISSION_FIELD

add constraint FK_SUBMISSI_REQUEST_F_REQUEST_UPDATE foreign key (REQUEST_FOR_UPDATE_ID)
references REQUEST_FOR_UPDATE (ID)

go

alter table SUBMISSION_FIELD

add constraint FK_SUBMISSI_SUBMISSIO_SUBMISSI_FIELD foreign key (SUBMISSION_ID)
references SUBMISSION (ID)

go

alter table SUBMISSION_FIELD_DFR

add constraint FK_SUBMISSION_METADATA_DFR foreign key (METADATA_FIELD_ID)
references METADATA_FIELD (ID)

go

alter table SUBMISSION_FIELD_DFR

add constraint FK_SUBMISSION_REQUEST_FIELD_DFR foreign key (REQUEST_FOR_UPDATE_ID)
references REQUEST_FOR_UPDATE (ID)

go

alter table SUBMISSION_FIELD_DFR

add constraint FK_SUBMISSION_FIELD_DFR foreign key (DOCUMENT_FOR_RECOGNITION_ID)
references DOCUMENT_FOR_RECOGNITION (ID)

go

alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSION_LIFECYCLE_CHANGED_BY foreign key (PARTICIPANT_ROLE_CHANGED_BY_ID)
references PARTICIPANT_ROLE (ID)

go

alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSI_ENIC_FOR__DOCUMENT foreign key (DOCUMENT_FOR_RECOGNITION_ID)
references DOCUMENT_FOR_RECOGNITION (ID)

go

alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSI_FROM_ENIC_ENIC_NAR foreign key (ENIC_NARIC_REPLY_ID)

```
references ENIC_NARIC_REPLY (ID)

go
```

```
alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSI_LIFECYCLE_SUBMISSI foreign key (SUBMISSION_ID)

references SUBMISSION (ID)

go
```

```
alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSION_LIFECYCLE_OWNER foreign key (PARTICIPANT_ROLE_OWNER_ID)

references PARTICIPANT_ROLE (ID)

go
```

```
alter table SUBMISSION_LIFECYCLE

add constraint FK_SUBMISSI_STATUS_SUBMISSI foreign key (SUBMISSION_STATUS_ID)

references SUBMISSION_STATUS (ID)

go
```

```
alter table "USER"

add constraint FK_USER_USER_ACCO_PARTICIP foreign key (PARTICIPANT_ID)

references PARTICIPANT (ID)

go
```

APPENDIX III: Graphical User Interface Documentation of the System for Digitalization of the Recognition Process

ERASMUS+ PROJECT 2023-1-RS01-KA220-HED-000156660

EPIR | E-Procedure of Institutional Recognition of Foreign Higher Education Documents

Dokument:	Grafički korisnički interfejs sistema za digitalizaciju procesa priznavanja stranih visokoškolskih isprava
Verzija:	1.0
Dokument pripremili:	Igor Zečević Ljubomir Milašinović Danijel Venus

EPIR project partners:



UNIVERSITY
OF NOVI SAD



UNIVERSITÀ
POLITECNICA
DELLE MARCHE



ROMANIA
1 DECEMBRIE 1918
UNIVERSITY OF ALBA IULIA



50 years of
University
of Split



REPUBLIC OF SERBIA
Qualifications Agency

APSTRAKT

Cilj digitalizacije procesa priznavanja stranih visokoškolskih isprava koju obavlja kancelarija za pravne poslove Univerziteta u Novom Sadu je omogućavanje kandidatima online podnošenje zahteva za priznavanje i pratećih dokumenata, kao i povećanje efikasnosti rada.

Na osnovu prethodno izrađenog operativnog prototipa aplikacije pristupilo se izradi konačne verzije grafičkog korisničkog interfejsa sistema koji će obezbediti online podnošenje zahteva za priznavanje, jednostavnije praćenje procesa priznavanja za sve učesnike, efikasnije izveštavanje za potrebe institucije i nadležnih organa, i smanjenje broja dokumenata u papirnom obliku.

U ovoj fazi razvoja, prethodno izrađen operativni prototip veb aplikacije zasnovan na tehnologijama jQuery i Bootstrap preveden je na konačnu verziju grafičkog korisničkog interfejsa. Prevođenje je realizovano zamenom dummy podataka koji su korišćeni u prototipu interfejsa sa konkretnim pozivima REST servisa koji su u međuvremenu implementirani.

ABSTRACT

The goal of digitalizing the process of recognizing foreign higher education documents, carried out by the Legal Affairs Office of the University of Novi Sad, is to enable applicants to submit recognition requests and accompanying documents online, as well as to increase work efficiency.

Based on the previously developed operational prototype of the application, work continued on creating the final version of the system's graphical user interface, which will provide online submission of recognition requests, simplify monitoring of the recognition process for all participants, improve reporting for the needs of the institution and relevant authorities, and reduce the number of paper documents.

In this phase of development, the previously created operational prototype of the web application—built using jQuery and Bootstrap—was translated into the final version of the graphical user interface. The translation was carried out by replacing the dummy data used in the prototype interface with actual calls to the REST services that were implemented in the meantime.

Sadržaj

Opseg sistema	5
Grafički korisnički interfejs - osnovne funkcionalnosti	6
Grafički korisnički interfejs iz perspektive kandidata	10
Grafički korisnički interfejs iz perspektive punomoćnika	16
Grafički korisnički interfejsa iz perspektive službenika	21
Grafički korisnički interfejs šifarnika sistema	26
Grafički korisnički interfejs iz perspektive prodekana za nastavu	33
Grafički korisnički interfejs iz perspektive rukovodioca studijskog programa	34

Opseg sistema

Kancelarija za pravne poslove Univerziteta u Novom Sadu obavlja poslove priznavanja stranih visokoškolskih isprava za potrebe nastavka školovanja na Univerzitetu u Novom Sadu. Ova aktivnost predstavlja važan segment međunarodne saradnje i mobilnosti studenata, jer omogućava kandidatima iz drugih država da nastave svoje obrazovanje u okviru Univerziteta, u skladu sa važećim propisima i standardima.

Trenutno, proces priznavanja stranih visokoškolskih isprava nije digitalizovan, već se većina dokumenata i komunikacije odvija u papirnom obliku. Kandidati zahteve podnose lično ili putem pošte, dok zaposleni u Kancelariji za pravne poslove u upravljanju procesom koriste imejl komunikaciju i standardne kancelarijske softvere (tekstualne obrade, tabele, baze u ograničenom obliku). Ne postoji namensko softversko rešenje koje bi objedinjavalo sve aktivnosti i omogućilo integrisano praćenje toka priznavanja — od prijema zahteva do izdavanja odluke.

Digitalizacija ovog procesa ima za cilj da značajno unapredi način rada i iskustvo korisnika. Novi softverski sistem omogućiće kandidatima da elektronski podnesu zahtev za priznavanje i prate njegov status u realnom vremenu, dok će Kancelariji za pravne poslove i drugim organizacionim jedinicama Univerziteta obezbediti centralizovan, pregledan i efikasan način vođenja postupka.

Očekivani efekti uvođenja sistema su:

- Povećanje efikasnosti rada kroz automatizaciju administrativnih koraka i smanjenje manuelnog unosa podataka,
- Ubrzanje razmene informacija između svih učesnika u procesu,
- Smanjenje upotrebe papira i prelazak na elektronsku dokumentaciju,
- Poboljšano izveštavanje i jednostavnije generisanje statističkih podataka za potrebe Univerziteta i nadležnih organa, kao i povećanje transparentnosti i pristupačnosti procesa priznavanja za kandidate iz inostranstva.

Dodatno, očekuje se da digitalizacija ovog postupka doprinese jačanju međunarodne vidljivosti Univerziteta u Novom Sadu i povećanju broja stranih studenata zainteresovanih za nastavak studija na Univerzitetu, čime će se unaprediti i proces akademske mobilnosti i međunarodne saradnje.

Grafički korisnički interfejs - osnovne funkcionalnosti

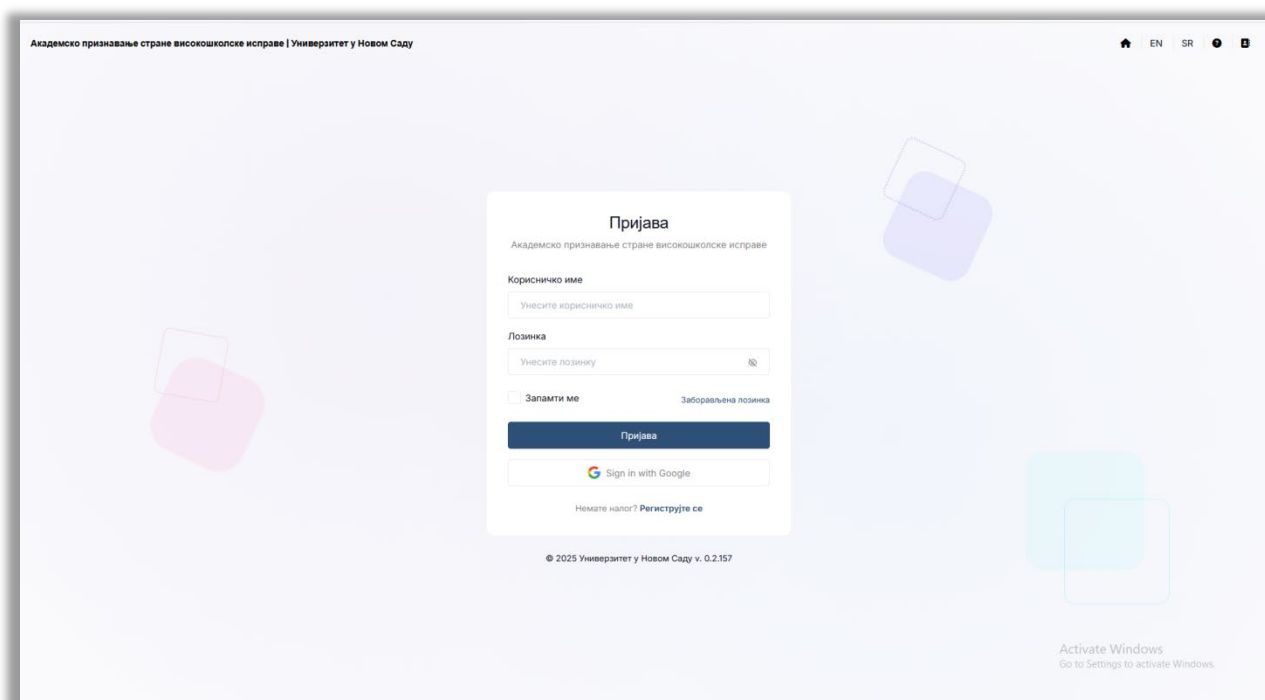
U ovoj fazi implementacije projekta, poslednja verzija aplikacije stalno je dostupna na adresi: <https://epir-test.ftn.uns.ac.rs>.

Verzija koja je dostupna kompletnom razvojnom tipu automatski se prenosi na ovu lokaciju kroz implementirani „pipeline“ koji je razvijen u okviru GitLab platforme na kojoj je nalazi i razvojna verzija programskog koda. Nakon faze testiranja sistem će biti premešten na produkcione servere i biće promenjen domen.

Na početnoj stranici aplikacije, korisnici imaju mogućnost izbora jezika i pisma (srpski ćirilica, srpski latinica, engleski jezik), linkove ka stranicama za pomoć i kontakt informacijama.

Na početnoj stranici korisnicima su dostupni i linkovi ka stranicama za registraciju i zaboravljenu lozinku kao i mogućnost prijave, ukoliko su prethodno registrovani na sistem.

Na slici ispod prikazan je izgled početne stranice aplikacije.



Na glavnom meniju nalaze se ikonice za promenu jezika i pisma, kao i linkovi ka stranicama za pomoć i kontakt informacije.

Registracija korisnika se obavlja u više koraka, što je predstavljeno na slikama ispod (odabran je slučaj kada je korisnik izabrao način registracije kao kandidat).

U prvom koraku, korisnik bira način registracije (kandidat ili punomoćnik).

Академско признавање стране високошколске исправе | Универзитет у Новом Саду

ENG LAT

Регистрација корисника

Да бисте поднели захтев за признавање стране високошколске исправе морате бити регистровани корисник.

1 2 3 4

Региструјете се као

Молим вас изаберите начин регистрације:

☐ Кандидат *

☐ Пуномоћник - Физичко лице *

☐ Пуномоћник - Правно лице *

Напред >>

Имате креиран налог? [Пријавите се овде](#)

© 2025 Универзитет у Новом Саду v. 0.2.157

U drugom koraku registracije, kandidat unosi lične podatke kao na slici ispod.

Академско признавање стране високошколске исправе | Универзитет у Новом Саду

ENG LAT

Регистрација корисника

Да бисте поднели захтев за признавање стране високошколске исправе морате бити регистровани корисник.

1 2 3 4

Лични подаци

Унесите Ваше личне податке.

Име * ✓ Презиме * ✓

Име једног од родитеља * ✓ Девотијачко презиме

Датум рођења * ✓ Место рођења * ✓

Држава рођења * x Држављанство * x

Пол ☒ Мушки * ☐ Женски *

<< Назад Напред >>

Имате креиран налог? [Пријавите се овде](#)

© 2025 Универзитет у Новом Саду v. 0.2.157

U trećem koraku registracije, kandidat unosi adresne podatke. Adresni podaci obuhvataju podatke prebivališta i podatke za prijem dokumenata što je ilustrovano na slici ispod.

Академско признавање стране високошколске исправе | Универзитет у Новом Саду

ENG LAT

Регистрација корисника

Да бисте поднели захтев за признавање стране високошколске исправе морате бити регистровани корисник.

1 2 3 4

Адресни подаци

АДРЕСА ПРЕБИВАЛИШТА

Улица и број * Место пребивалишта *

Усинсар 23 Вржота

Поштански број * Држава пребивалишта *

348765 Русија

АДРЕСА ЗА ПРИЈЕМ ДОКУМЕНАТА

Улица и број * Место *

Усинсар 23 Вржота

Поштански број * Држава *

348765 Русија

<< Назад Напред >>

Имате креиран налог? [Пријавите се овде](#)

© 2025 Универзитет у Новом Саду v. 0.2.157

U последњем koraku registracije, kandidati moraju da unesu podatke vezane za korisnički nalog: imejl adresu, broj telefona i lozinku.

Академско признавање стране високошколске исправе | Универзитет у Новом Саду

ENG LAT

Регистрација корисника

Да бисте поднели захтев за признавање стране високошколске исправе морате бити регистровани корисник.

1 2 3 4

Кориснички налог

Унесите податке корисничког налога

Имејл адреса * Телефон *

Унесите имејл адресу (912) 345-67-89

Лозинка * Потврдите лозинку *

Унесите лозинку Унесите потврду лозинке

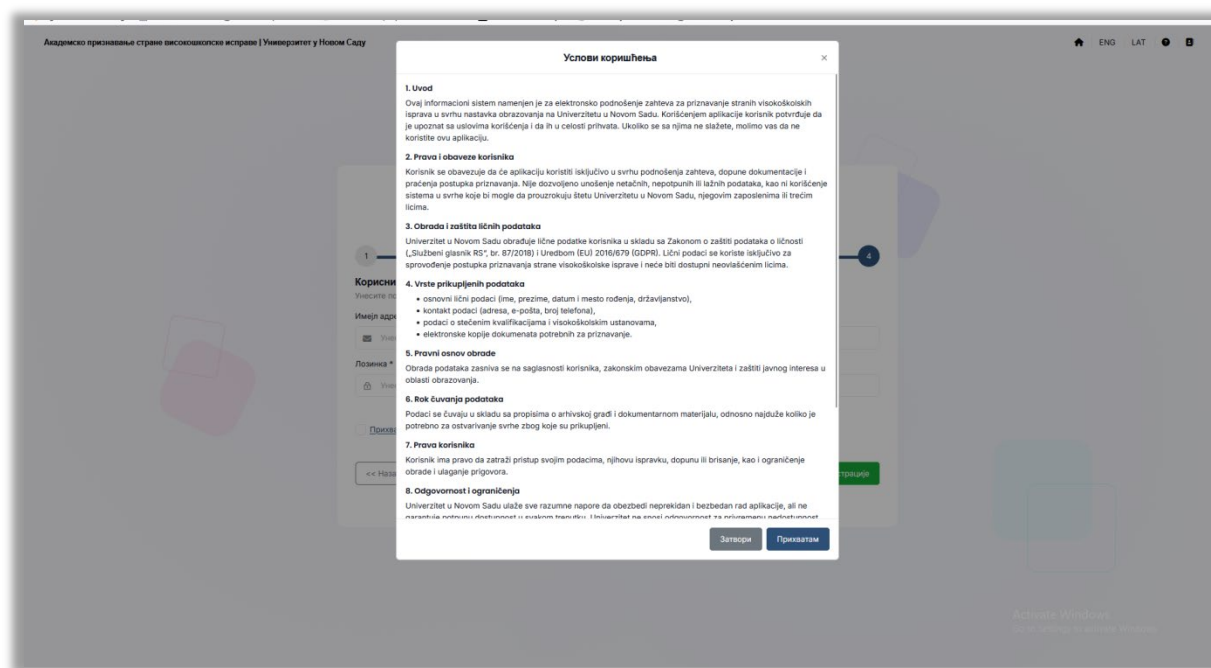
☐ Прихватам услове коришћења

<< Назад Потврда регистрације

Имате креиран налог? [Пријавите се овде](#)

© 2025 Универзитет у Новом Саду v. 0.2.157

U последњем koraku procesa registracije, nalaze se i uslovi korišćenja sistema sa kojima kandidat mora biti eksplicitno saglasan da bi završio proces registracije.

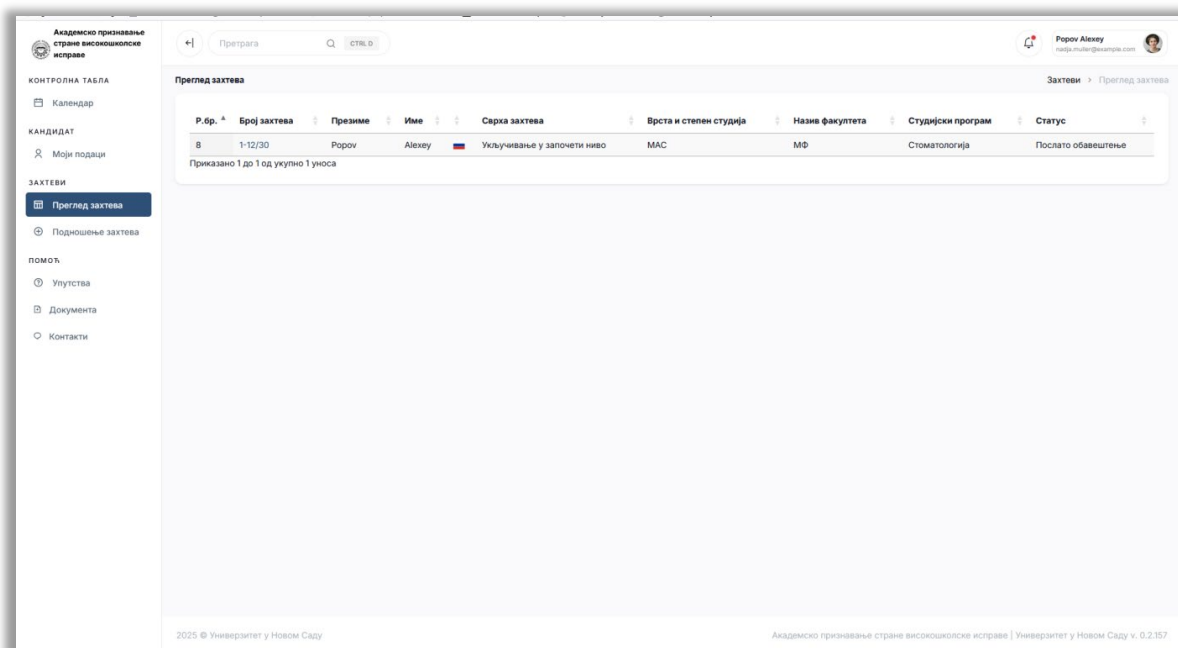


Po uspešnoj registraciji, kandidatima na imejl adresu stiže mejl koji sadrži aktivacioni link. Klikom na aktivacioni link, kandidati postaju registrovani korisnici i mogu da izvrše prijavu na sistem.

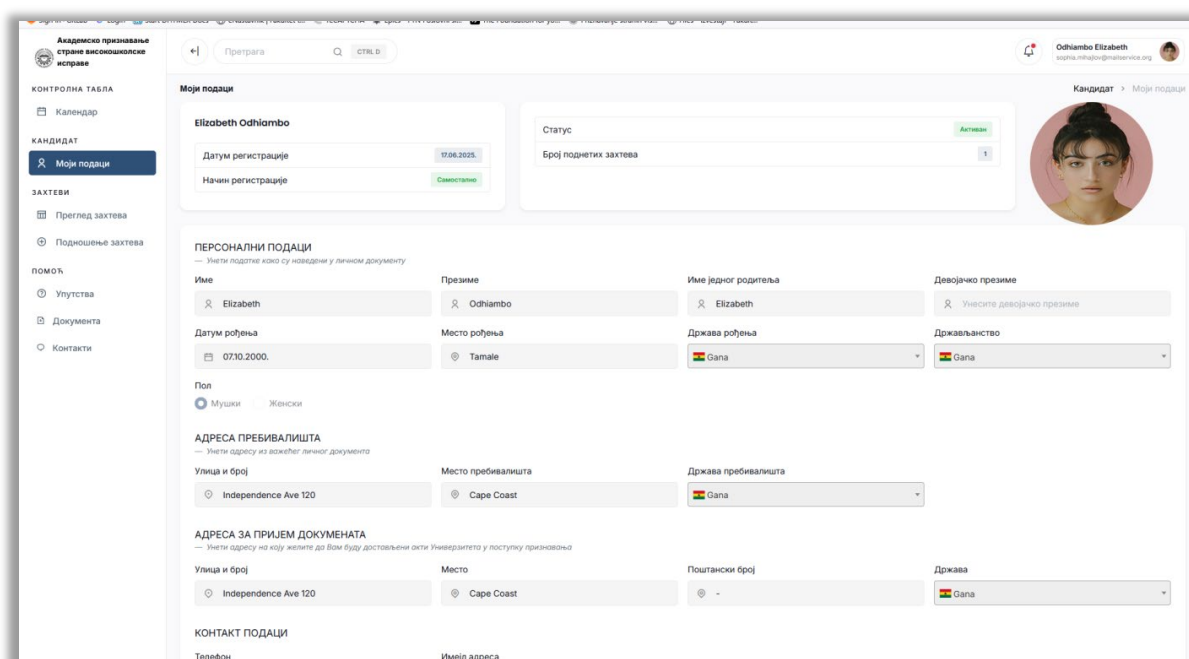
Grafički korisnički interfejs iz perspektive kandidata

Kandidat je fizičko lice koje podnosi zahtev za priznavanje strane visokoškolske isprave. Jedan kandidat može podneti više zahteva za priznavanje.

Nakon prijave na sistem kandidat ima uvid u listu svih svojih zahteva i njihovih statusa kao na slici ispod. Bitno je napomenuti da u listi zahteva mogu da se nađu i zahtevi koji su u pripremi za podnošenje od strane kandidata, odnosno zahtevi koje je kandidat započeo i ima mogućnost da nastavi njihovu pripremu.



Putem stavke menija „Moji podaci“, kandidatu je dostupan uvid u njegove lične i adresne podatke koje je inicijalno uneo prilikom registracije na sistem.



Kandidatu je takođe na raspolaganju i mogućnost izmene personalnih i adresnih podataka, što je ilustrovano na slici ispod. U zavisnosti od postojanja i statusa zahteva za priznavanje, akcija promene ličnih podataka može dovesti do obaveštavanja službenika o promenama.

The screenshot shows the 'Personal Data' form for a candidate named Elizabeth Odiambo. The form is divided into several sections: 'Personal Data', 'Address of Residence', 'Address for Document Submission', and 'Contact Data'. Each section contains input fields for various details, including name, date of birth, place of birth, and contact information. A sidebar on the left contains navigation links for 'Control Panel', 'Calendar', 'Candidate', 'My Data', 'Requests', 'View Request', 'Submit Request', 'Help', 'Instructions', 'Documents', and 'Contacts'. The top right corner shows the candidate's profile picture and name.

Podnošenje novog zahteva kandidat obavlja u više koraka. U prvom koraku, predviđen je unos personalnih i adresnih podataka. Podrazumevane su vrednosti koje je kandidat uneo prilikom registracije na sistem. U ovom koraku, kandidat ima mogućnost da inicijalne podatke po potrebi i izmeni.

The screenshot shows the 'Submit Request' form, which is a multi-step process. The first step, 'Personal Data', is highlighted. The form contains the same input fields as the 'Personal Data' form in the previous screenshot, but with additional validation markers (asterisks) indicating required fields. A progress bar at the top shows the sequence of steps from 1 to 6. The bottom right corner features a 'Next >>' button.

U drugom koraku procesa podnošenja zahteva, kandidat bira svrhu podnošenja zahteva kao i visokoškolsku ustanovu i studijski program/modul za koje se zahtev odnosi. Ovo je ilustrovano na slici ispod.

U trećem koraku procesa podnošenja zahteva, kandidat unosi skup podataka koji se odnose na stečenu visokoškolsku ispravu za koju se traži priznavanje. Kandidat ima mogućnost unosa podataka za više visokoškolskih isprava.

U četvrtom koraku podnošenja zahteva, od kandidata se očekuje da unese podatke o prethodnom obrazovanju.

U ovom koraku, kandidat unosi podatke o stečenom srednjoškolskom obrazovanju i opciono (u zavisnosti od prethodno odabrane svrhe podnošenja zahteva) podatke o stečenom visokom obrazovanju. Skup polja koja su predviđena za unos su prikazana na slici iznad.

U petom koraku procesa podnošenja zahteva, kandidat mora da otpremi skup dokumenata koji su uslovljeni prethodno odabranom svrhom podnošenja zahteva.

Poslednji korak podnošenja zahteva podrazumeva pregled i kontrolu prethodno popunjenih polja i otpremljenih dokumenata kao i potvrdu podnošenja zahteva.

Nakon uspešnog podnošenja zahteva, kandidat ima mogućnost uvida u podnete zahteve. Za svaki podneti zahtev korisnik ima mogućnost uvida u njegove detalje kao i vremensku liniju obrade zahteva. U zavisnosti od trenutnog statusa pojedinačnog zahteva, kandidatu su dostupne i odgovarajuće akcije nad zahtevom, poput dopune zahteva, delimične izmene zahteva i odustanka od zahteva.

Академско признавање стране високошколске исправе

ПРЕГЛЕД ЗАХТЕВА

Захтев #32450-23452345

Кандидат: Alexey Popov
Датум подношења: 18.06.2025
Статус: Prijeto obavezno

Сврха: Укључивање у запоселни лист
Ниво студија: MAS
Установа: MS
Програм/Модул: International

ИЗМЕНА ЗАХТЕВА
ОДУСТАНИ ОД ЗАХТЕВА
Прикажи временску линију

ПЕРСОНАЛНИ ПОДАЦИ

Име: Alexey
Презиме: Popov
Име једног од родитеља: Alexey
Пол: Мушки
Девојачко презиме: Унесите девојачко презиме
Датум рођења: 08.10.2000.
Место рођења: Kazan
Држава рођења: Rusija
Држављанство: Gana

АДРЕСА ПРЕБИВАЛИШТА

Улица и број: Tverskaya St 129
Место пребивалишта: Kazan
Држава пребивалишта: Rusija

Адреса за пријем докумената

Registrovanim kandidatima na raspolaganju su i stranice sa korisničkim uputstvima kao i različitim vrstama dokumenata koje kandidati po potrebi mogu preuzeti.

Stavka menija „Uputstva“ kandidate vodi do stranice na kojoj se nalaze odgovori na najčešće postavljana pitanja.

Академско признавање стране високошколске исправе

FAQ's

Search for ...

Podnošenje zahteva:

Ko može podneti zahtev za priznavanje strane visokoškolske isprave?

Gde se podnosi zahtev za priznavanje?

Da li se zahtev može podneti elektronski?

Koja dokumenta je obavezno priložiti uz zahtev?

Da li je potrebno dostaviti originalna dokumenta?

Koliko traje postupak nakon podnošenja zahteva?

Šta ako dokumentacija nije potpuna?

Browse Help Categories

- Account Management**
Create, manage, and secure your account easily.
- Settings & Preferences**
Customize your experience and app behavior.
- Privacy & Security**
Understand how we keep your data safe account.
- Team & Collaboration**
Manage users, permissions, assign access level and groups.

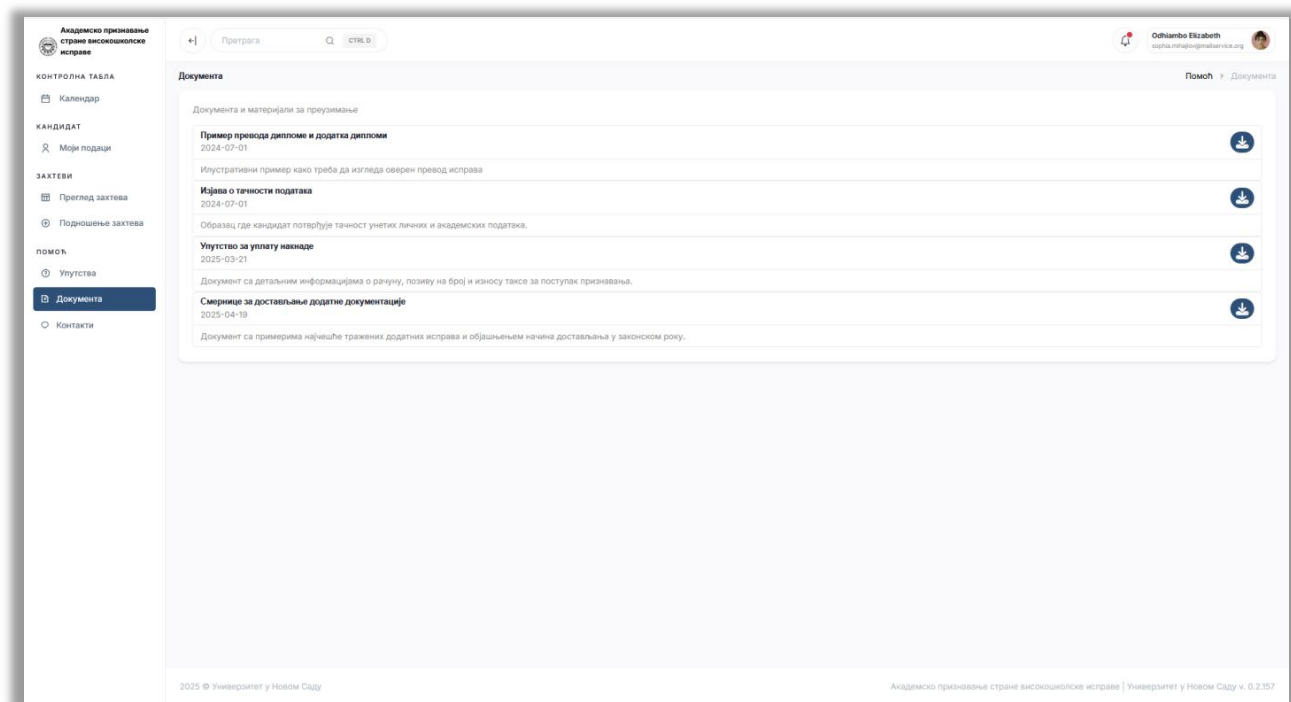
Still need support?
Reach out to us anytime for assistance, we're here to help!

support@uns.ac.rs
+4151 9874 5621

2025 © Универзитет у Новом Саду

Академско признавање стране високошколске исправе | Универзитет у Новом Саду v. 0.2.157

Stavka menija „Dokumenta“ kandidate vodi do stranice na kojoj se nalaze dokumenti i šabloni koje kandidati mogu preuzeti, a koji im mogu pomoći u procesu podnošenja zahteva.



Grafički korisnički interfejs iz perspektive punomoćnika

Punomoćnik predstavlja pravno ili fizičko lice koje vrši podnošenje zahteva za priznavanje stranih visokoškolskih isprava za više drugih fizičkih lica (kandidata). Punomoćnik može imati proizvoljan broj zahteva, ali za jedno fizičko lice može imati maksimalno 3 zahteva koja su u postupku.

Korisnici u ulozi punomoćnika imaju sva prava i dostupne funkcionalnosti kao i korisnici u ulozi Kandidata sa razlikom da korisnici u ulozi Punomoćnika mogu da izvršavaju funkcionalnosti za veći broj kandidata. Nakon prijave na sistem, punomoćnika dočekuje početna stranica na kojoj se nalazi pregled svih njegovih zahteva kao na slici ispod.

Академско признавање страних високошколских исправа

Контролна табела

Календар

Кандидати

Пregled кандидата

Нови кандидат

Захтеви

Пregled захтева

Подношење захтева

Помоћ

Упутства

Документи

Контакти

Пregled захтева

Активни филтери - 0

Врста и степен студија

ДАС

МАС

ОАС

Статус

Поднет

Покренут поступак

Примљен извештај факултета

Скупље све Прикажи све Обриши све

Р.бр.	Број захтева	Презиме	Име	Српска захтева	Врста и степен студија	Назив факултета	Студијски програм	Статус
1	1-12/23	Sokolov	Anna	Усључивање у започети ниво	МАС	ФФ	Комуникологија	Покренут поступак
2	1-12/38	Otieno	Peter	Усључивање у започети ниво	ОАС	ФТН	Ра-чунарство и аутоматика	Поднет
3	1-12/46	Tkachenko	Petro	Усључивање у започети ниво	ОАС	ПМФ	Геоинформатика	Примљен извештај факултета
4	1-12/56	Tkachenko	Petro	Усључивање у започети ниво	МАС	МФ	Сестринство	Примљен извештај факултета
5	1-12/61	Tkachenko	Petro	Усључивање у започети ниво	МАС	ФТН	Архитектура	Примљен извештај факултета
6	1-12/64	Kiptoo	Michael	Усључивање у започети ниво	ДАС	ФТН	Ра-чунарство и аутоматика	Примљен извештај факултета

Приказано 1 до 6 од укупно 6 уноса

2025 © Универзитет у Новом Саду

Академско признавање страних високошколских исправа | Универзитет у Новом Саду v. 0.2.157

Punomoćniku je dostupan uvid u detalje svakog podnetog zahteva. U zavisnosti od trenutnog statusa pojedinačnog zahteva, punomoćniku su dostupne i odgovarajuće akcije nad zahtevom, poput dopune zahteva, delimične izmene zahteva i odustanka od zahteva.

Академско признавање стране високошколске исправе

КОНТРОЛНА ТАБЛА

КАНДИДАТИ

ПРЕГЛЕД КАНДИДАТА

НОВИ КАНДИДАТ

ЗАХТЕВИ

ПРЕГЛЕД ЗАХТЕВА

ПОДНОШЕЊЕ ЗАХТЕВА

ПОМОЋ

УПУТСТВА

ДОКУМЕНТА

КОНТАКТИ

Подношење захтева

1 2 3 4 5 6

ПЕРСОНАЛНИ ПОДАЦИ

Подаци о подносиоцу захтева

Унесите податке како су наведени у личном документу који се прилаже овом захтеву

Или одaberите већ постојећег кандидата

Име *

Презиме *

Датум рођења *

Место рођења *

Пол

Држављанство *

Девичанско презиме

Адреса пребивалишта

Улица и број *

Место пребивалишта *

Држава пребивалишта *

Адреса за пријем докумената

Улица и број *

Место *

Поштански број *

Држава *

Контакт подаци

Телефон *

Имејл адреса *

Напред >>

Na prvom koraku podnošenja novog zahteva za priznavanje, punomoćnik može da odabere prethodno evidentirane kandidate odnosno kandidate za koje je prethodno već podnosio zahtev. Odabirom kandidata (ilustrovano na slici iznad), svi personalni i adresni podaci kandidata automatski se popunjavaju.

Svi ostali koraci u podnošenju novog zahteva od strane punomoćnika se ne razlikuju od podnošenja zahteva od strane kandidata.

Registrovanim punomoćnicima na raspolaganju su i stranice sa korisničkim uputstvima kao i različitim vrstama dokumenata koje po potrebi mogu preuzeti.

Grafički korisnički interfejsa iz perspektive službenika

Službenik predstavlja lice zaduženo za administriranje zahteva za priznavanje stranih visokoškolskih isprava kao i neophodnih šifarnika sistema.

Službenik ima uvid u sve zahteve koji postoje u sistemu. Takođe, za zahtev koji je uzeo u obradu, ima maksimalne privilegije koje mu omogućavaju da menja metapodatke u okviru zahteva, dodaje transkribovane metapodatke, dodaje potrebna dokumenta u zahtev, upućuje upit ENIC/NARIC centru, prosleđuje predmet fakultetu (prodekanu za nastavu i rukovodiocu studijskog programa), prosleđuje predmet stručnom veću, kreira predlog rešenja, šalje obaveštenja kandidatu. Ima mogućnost generisanja raznih statističkih i drugih izveštaja.

Nakon prijave na sistem, službenika dočekuje stranica sa pregledom svih aktivnih zahteva. Pored osnovnih informacija o zahtevima, službeniku na raspolaganju stoji i mogućnost sortiranja zahteva, brza i napredna pretraga zahteva kao i skup kaskadnih filtera po svrsi zahteva, vrsti i stepenu studija, fakultetu, statusu i obrađivaču zahteva.

Акademsko признавање стране високошколске исправе

КОНТРОЛНА ТАБЛА

Закрени

Календар

ЗАХТЕВИ

Поглед захтева

Подношење захтева

Пуномоћници

Кандидати

ENIC/NARIC

База одговора

Слање упита

КАТАЛОЗИ

Територијална орг.

Универзитет

Документа

ИЗВЕШТАЈИ И СТАТИСТИКЕ

Извештаји

Статистике

Поглед захтева

Активни филтери - 0

Скуп све Прикажи све Обриши све

Сврха захтева	Врста и степен студија	Назив факултета	Статус	Обрађивач
Укључивање у започети ниво	ДАС	ЛУ	Верификован извештај	Нема података
Упис истог нивоа	МАС	ДИФ	Допуна захтева	Милица Остојић
Упис наредног нивоа	МСС	ЕФ	Допумен захтев	Наташа Максимовић
	ОАС	МФ	Поднет	Уна Царевић

Прикажи 10 уноса

Бр.з	Број захтева	Презиме	Име	Сврха захтева	Врста и степен студија	Назив факултета	Студијски програм	Статус	Обрађивач
1	1-12/23	Sokolov	Anna	Укључивање у започети ниво	МАС	ФФ	Компјутерологија	Покренут поступак	Милица Остојић
2	1-12/24	Mulliso	Esther	Упис истог нивоа	МАС	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
3	1-12/25	Kamsau	John	Укључивање у започети ниво	ОАС	МФ	Сестринство	Поднет	
4	1-12/26	Novikov	Elena	Укључивање у започети ниво	МАС	ЕФ	Дигитални маркетинг	Покренут поступак	Милица Остојић
5	1-12/27	Kravchenko	Oksana	Упис наредног нивоа	МАС	ЕФ	Дигитални маркетинг	Подношење захтева	
6	1-12/28	Moroz	Kateryna	Укључивање у започети ниво	ДАС	ПМФ	Природно-математички факултет	Подношење захтева	
7	1-12/29	Odhiambo	Elizabeth	Упис истог нивоа	МАС	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
8	1-12/30	Popov	Alexey	Укључивање у започети ниво	МАС	МФ	Стоматологија	Послато обавештење	Уна Царевић
9	1-12/31	Boyko	Viktor	Упис наредног нивоа	МАС	ДИФ	Физичко васпитање, физичка активност и здравље	Покренут поступак	Наташа Максимовић
10	1-12/32	Kovalenko	Mykola	Укључивање у започети ниво	МАС	ФТН	Мехатроника	Преузет	Милица Остојић

Приказано 1 до 10 од укупно 42 уноса

Прва Претходна 1 2 3 4 5 Следећа Последња

Activate Windows
Go to Settings to activate Windows.

2025 © Универзитет у Новом Саду

Академско признавање стране високошколске исправе | Универзитет у Новом Саду v. 0.2.159

Одабиром захтева из табеле, службеник добија детаљан преглед захтева са свкупом доступних акција у зависности од тренутног статуса захтева, што је илустровано на слици испод.

Zahtev
1-12/23

Kandidat: Alma Sokolov
Datum podnošenja: 10.08.2025
Status: Poverljivost postupak
Обрађивач: Милош Острошћ

Сврха: Удруженица у златном злату
Ниво студија: МАС
Установа:
Програм/Модул:
Подаци о стеченој исправи
Подаци о претходном образовању
Документа

ПЕРСОНАЛНИ ПОДАЦИ
Подаци о подносиоцу захтева
— издати у личном документу који се прилагођава овом захтеву

Име: Алма Презиме: Sokolov Име једног од родитеља: Алма Пол: Мушки Женски
Девичанско презиме: Датум рођења: 01.10.2000. Место рођења: Cape Coast Држава рођења: Gana
Држављанство: Gana

АДРЕСА ПРЕБИВАЛИШТА
— из важећег личног документа

Улица и број: Болгатага Ст 33 Место пребивалишта: Kumas Држава пребивалишта: Gana

Адреса за пријем документа
— На коју желите да Вам буду достављени акти Универзитета у поступку признавања

Улица и број: Болгатага Ст 33 Место: Kumas Поштански број: - Држава: Gana

Službenicima je na raspolaganju i mogućnost evidencije novog zahteva. Ova funkcionalnost je dostupna iz stavke menija „Podnošenje zahteva“.

Podnošenje zahteva

1 2 3 4 5 6

ПЕРСОНАЛНИ ПОДАЦИ
Подаци о подносиоцу захтева
— Унесите податке како су наведени у личном документу који се прилагођава овом захтеву

Име * Презиме * или одаберите већ постојећег кандидата
Датум рођења * Место рођења * или одаберите већ постојећег кандидата
Пол * Девичанско презиме *
Држављанство * Одаберите држављанство

АДРЕСА ПРЕБИВАЛИШТА
— Унесите адресу из важећег личног документа

Улица и број * Место пребивалишта * Држава пребивалишта *
Одаберите државу становања

АДРЕСА ЗА ПРИЈЕМ ДОКУМЕНАТА
— Унесите адресу на коју желите да Вам буду достављени акти Универзитета у поступку признавања

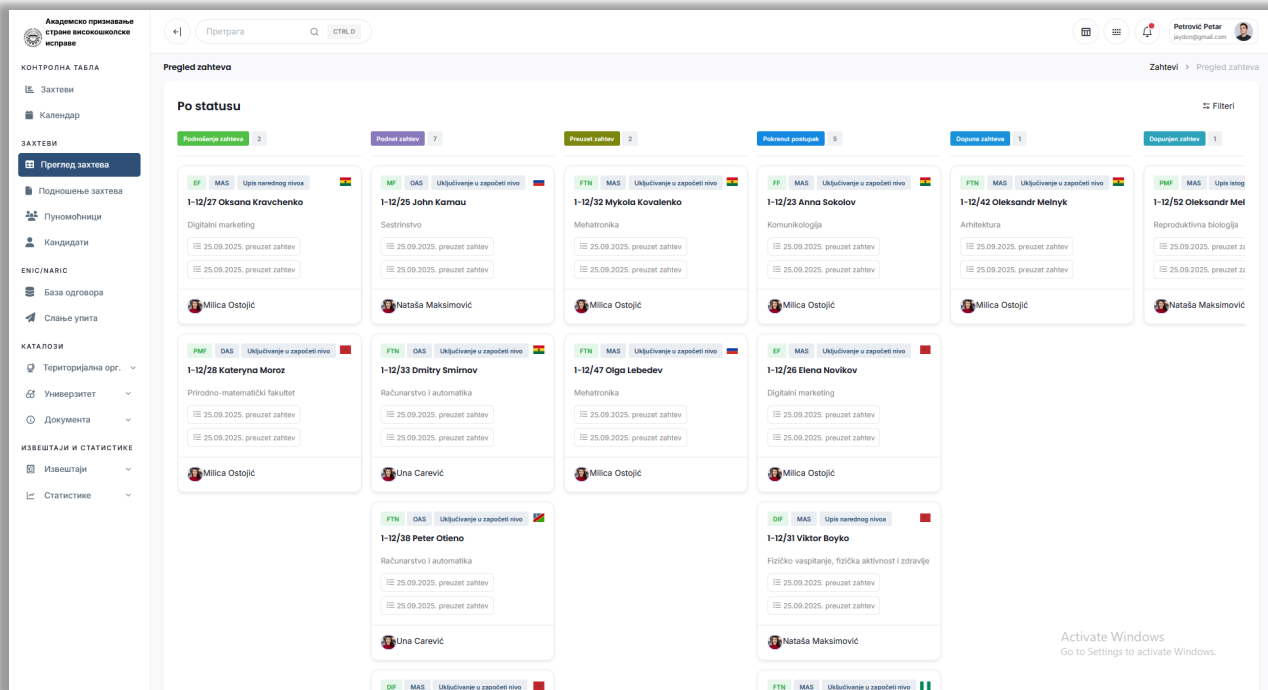
Улица и број * Место * Поштански број * Држава *
Одаберите државу за пријем документа

КОНТАКТ ПОДАЦИ

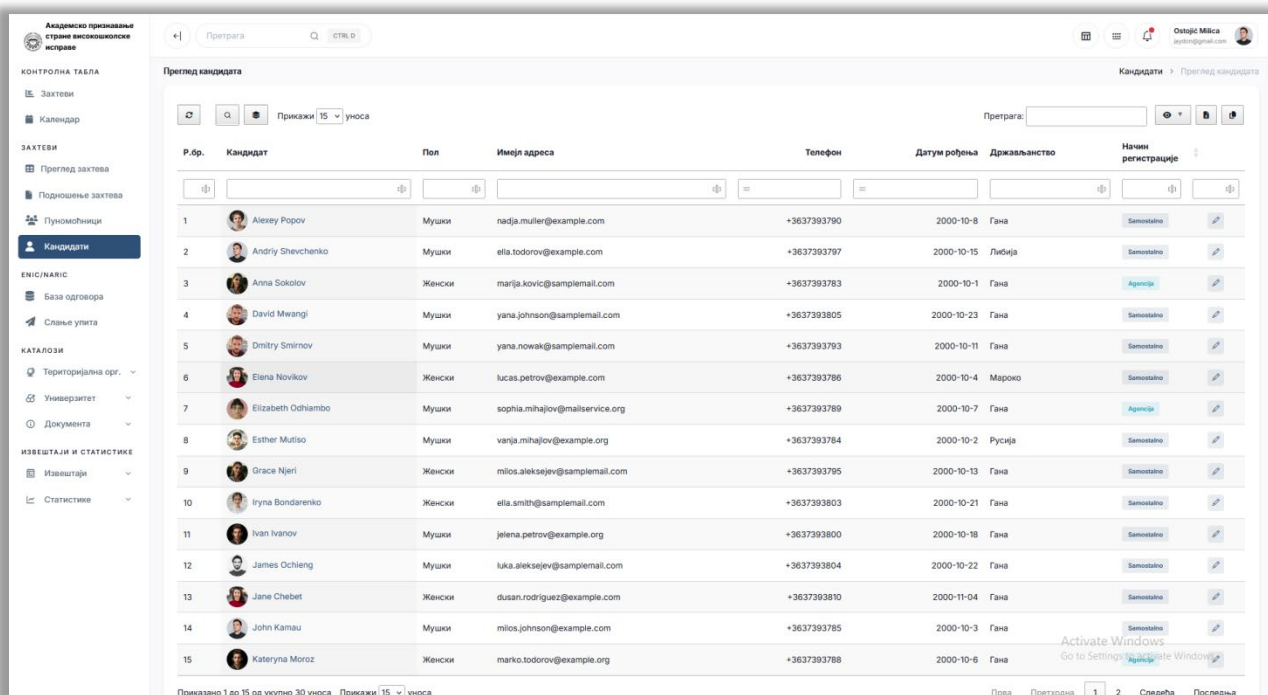
Телефон * Имејл адреса *

Prilikom podnošenja novog zahteva, službeniku je kao i kod punomoćnika dostupan izbor kandidata za koga se podnosi zahtev.

Pored tabelarnog prikaza aktivnih zahteva, službenicima je dostupan i „Kanban“ prikaz svih zahteva. U ovom prikazu koji je ilustrovan na slici ispod, zahtevi su grupisani po statusu ili obrađivaču.



Korisnicima u ulozi službenika na raspolaganju stoji i pregled svih kandidata i njihovih detalja što je prikazano na slikama ispod.



Кандидат

Dmitry Smirnov

Статус: Активен

Број поднетих захтева: 1

Датум регистрације: 21.06.2025

Начин регистрације: Самостално

ПЕРСОНАЛНИ ПОДАЦИ

Име: Dmitry Презиме: Smirnov Име једног родитеља: Dmitry Девеојачко презиме: Унесите девеојачко презиме

Датум рођења: 11.10.2000. Место рођења: Takoradi Држава рођења: Ghana Држављанство: Ghana

Пол: ☒ Мушки ☐ Женски

АДРЕСА ПРЕБИВАЛИШТА

Улица и број: Bolgatanga St 154 Место пребивалишта: Kumasi Држава пребивалишта: Ghana

АДРЕСА ЗА ПРИЈЕМ ДОКУМЕНАТА

Улица и број: Bolgatanga St 154 Место: Kumasi Поштански број: - Држава: Ghana

КОНТАКТ ПОДАЦИ

Телефон: +3637393793 Емајл адреса: yana.nowak@samplemail.com

Activate Windows
Go to Settings to activate Windows.

Takođe, korisnicima u ulozi službenika, na raspolaganju je i uvid u listu registrovanih punomoćnika i njihovih detalja što je ilustrovano na slici ispod.

Пунomoћници

Прикажи 15 уноса

Претрага:

Р.бр.	Назив	Држава	Контакт	Статус	Број кандидата	Број пријава
1	Emily Rodriguez ID: TCH-1003	Meksiko	e.rodriguez@example.com (345) 678-9012	активан	44	65
2	Michael Chen ID: TCH-1002	Francuska	m.chen@example.com (234) 567-8901	активан	78	86
3	Sarah Johnson ID: TCH-1001	Velika Britanija	sarah.j@example.com (123) 456-7890	активан	65	72

Приказано 1 до 3 од укупно 3 уноса Прикажи 15 уноса

Права Претходна 1 Следећа Последња

Activate Windows
Go to Settings to activate Windows.

Службеници имају могућност слања упита ERIC/NARIC центру као и евидентирање и претрагу одговора. На слици испод приказан је изглед екрана са базом одговора ERIC/NARIC центра.

Grafički korisnički interfejs šifarnika sistema

Značajan deo rada iz uloge službenika obuhvata pregled i održavanje različitih šifarnika u koje spadaju: države, visokoškolske ustanove, nivoi studija, studijski programi, moduli, rukovodioci programa, prodekani za nastavu, šabloni poruka, statusi zahteva, dokumenti za preuzimanje, prilozi zahteva.

Sve stranice koje prikazuju šifarnike imaju identičnu strukturu i identičan skup funkcionalnosti koje su predviđene za uvid i održavanje pojedinačnih šifarnika. Za svaki šifarnik, korisnicima u ulogu službenika dostupne su opcije brze i napredne pretrage, sortiranja, paginacije, kaskadnih filtera, eksporta u excel i kopiranja u clipboard. Takođe, za svaki šifarnik korisnicima u ulogu službenika dostupne su mogućnosti dodavanja, izmene i brisanja slogova.

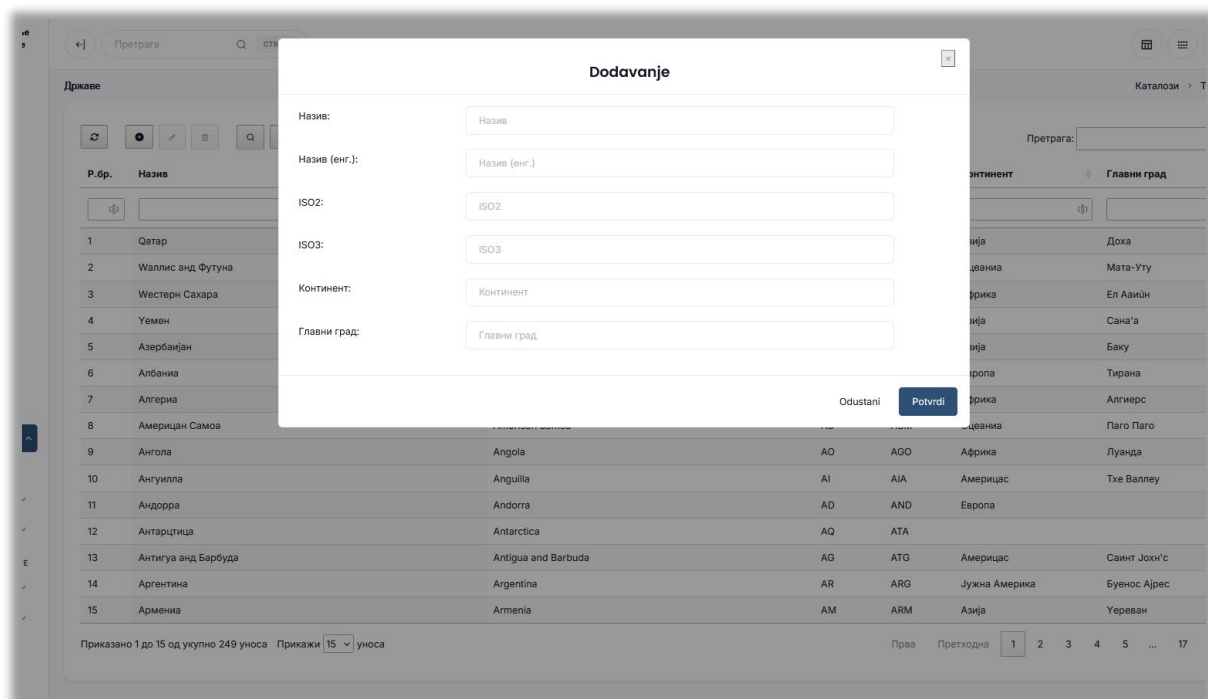
Ono što je takođe bitno napomenuti je i to da je za sve šifarnike neophodno obezbediti i održavanje naziva i opisa na engleskom jeziku i to je ono po čemu su ovi podaci dodatno karakteristični.

Na slici ispod prikazan je šifarnik država.

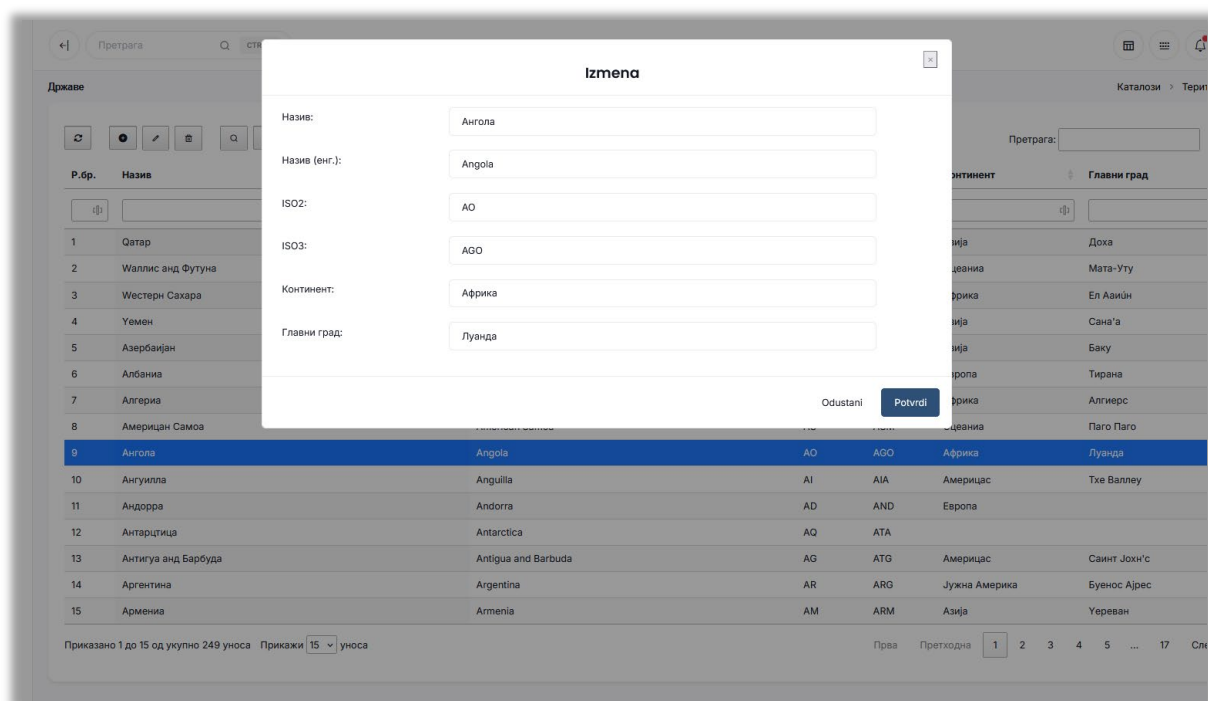
R.br.	Naziv	Naziv (eng.)	ISO2	ISO3	Kontinent	Glavni grad
1	Qatar	Qatar	QA	QAT	Azija	Doha
2	Wallis and Futuna	Wallis and Futuna	WF	WLF	Oceania	Mata-Utu
3	Western Sahara	Western Sahara	EH	ESH	Afrika	El Aaiun
4	Yemen	Yemen	YE	YEM	Azija	Sana'a
5	Azerbajdžan	Azerbaijan	AZ	AZE	Azija	Baku
6	Albanija	Albania	AL	ALB	Evropa	Tirana
7	Algerija	Algeria	DZ	DZA	Afrika	Alžir
8	Američan Samoa	American Samoa	AS	ASM	Oceania	Pago Pago
9	Angola	Angola	AO	AGO	Afrika	Luanda
10	Angvila	Anguilla	AI	AIA	Američas	Twe Valley
11	Andorra	Andorra	AD	AND	Evropa	Andorra la Vella
12	Antarctička	Antarctica	AQ	ATA		
13	Antigva i Barbuda	Antigua and Barbuda	AG	ATG	Američas	Saint John's
14	Argentina	Argentina	AR	ARG	Juzna Amerika	Buenos Aires
15	Armenija	Armenia	AM	ARM	Azija	Yerevan

Na svakoj stranici šifarnika, iznad same tabele sa podacima dostupna je komandna linija sa skupom akcija: dodavanje, izmena i brisanje sloga.

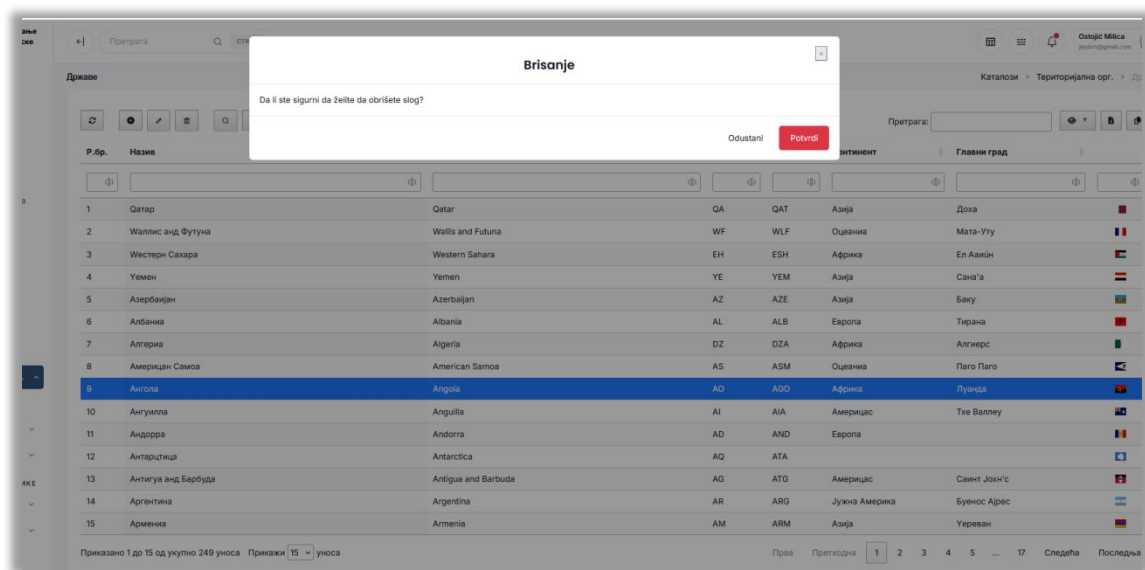
Na slici ispod ilustrovan je izgled dijaloga za dodavanje nove države.



Na slici ispod ilustrovan je izgled dijaloga za izmenu odabrane државе.

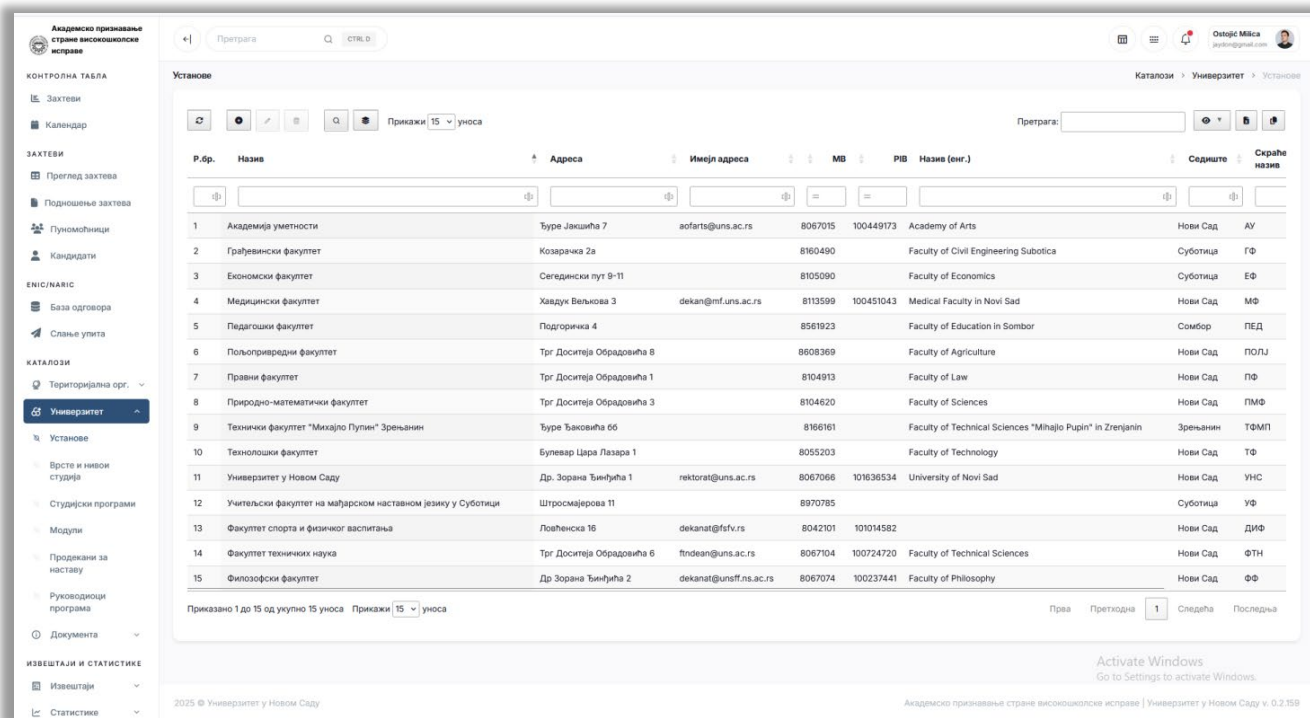


Na slici ispod prikazan je dijalog за поврду брисања селектоване државе.



Ilustrovani dijalozi za dodavanje, izmenu i brisanje slogova identični su za sve šifarnike. U nastavku sledi prikaz dela šifarnika koji su dostupni u aplikaciji.

Na slici ispod prikazan je šifarnik visokoškolskih ustanova.



Šifarnik ustanova trenutno sadrži sve članice Univerziteta u Novom Sadu kao i Univerzitet koji i sam izvodi studijske programe.

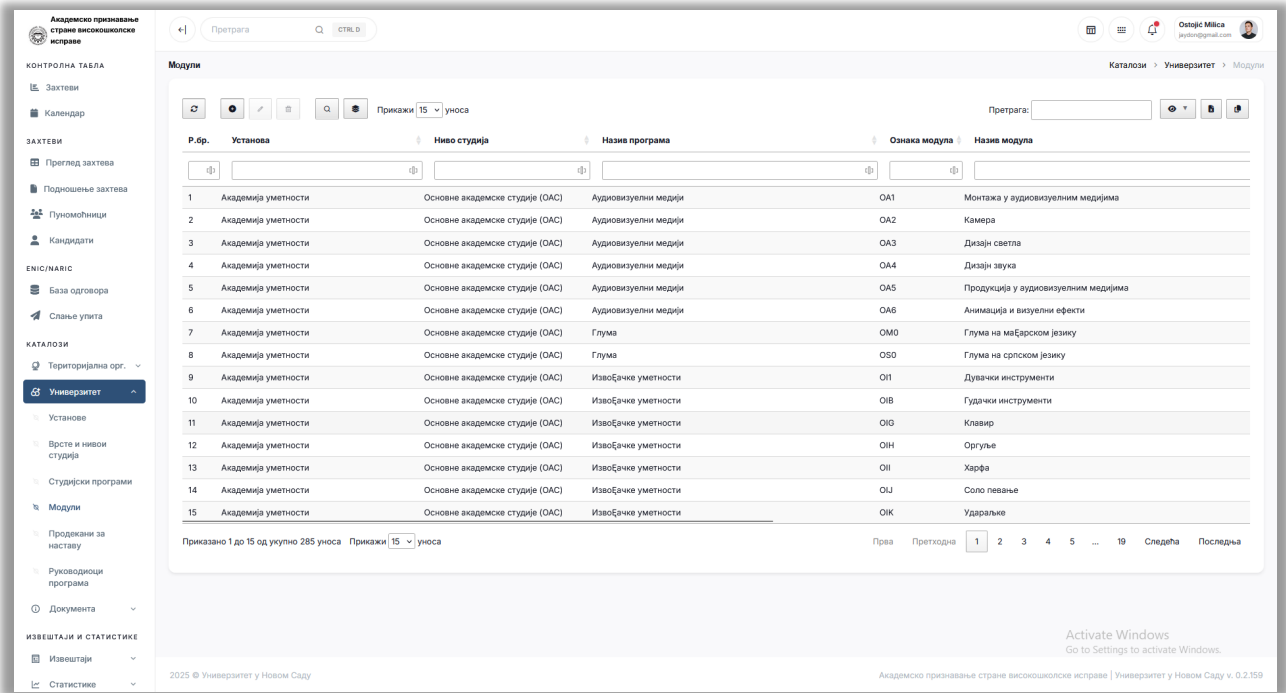
Na slici ispod prikazan je šifarnik vrsta i nivoa studija.

R.br.	Назив	Назив (eng.)	Скраћени назив	Боја
1	Основне академске студије (OAC)	Undergraduate Academic Studies	OAS	Blue
2	Основне струковне студије (OCC)	Undergraduate Professional Studies	OSS	Green
3	Специјалистичке струковне студије (CCC)	Undergraduate Professional Studies	SSS	Yellow
4	Мастер академске студије (MAC)	Master Academic Studies	MAS	Orange
5	Специјалистичке академске студије (CAC)	Specialised Academic Studies	SAS	Pink
6	Специјалистичке струковне студије (Старо)	Specialised Professional Studies	SSSS	Black
7	Интегрисане академске студије (IAC)	Integrated Academic Studies	IAS	Grey
8	Мастер струковне студије (MCC)	Professional Master Studies	MSS	Cyan
9	Интегрисане академске студије из медицинских наука (IAMS)	Professional Master Studies	IAMS	Red
10	Докторске студије (DAC)	Doctoral Academic Studies	DAS	Black

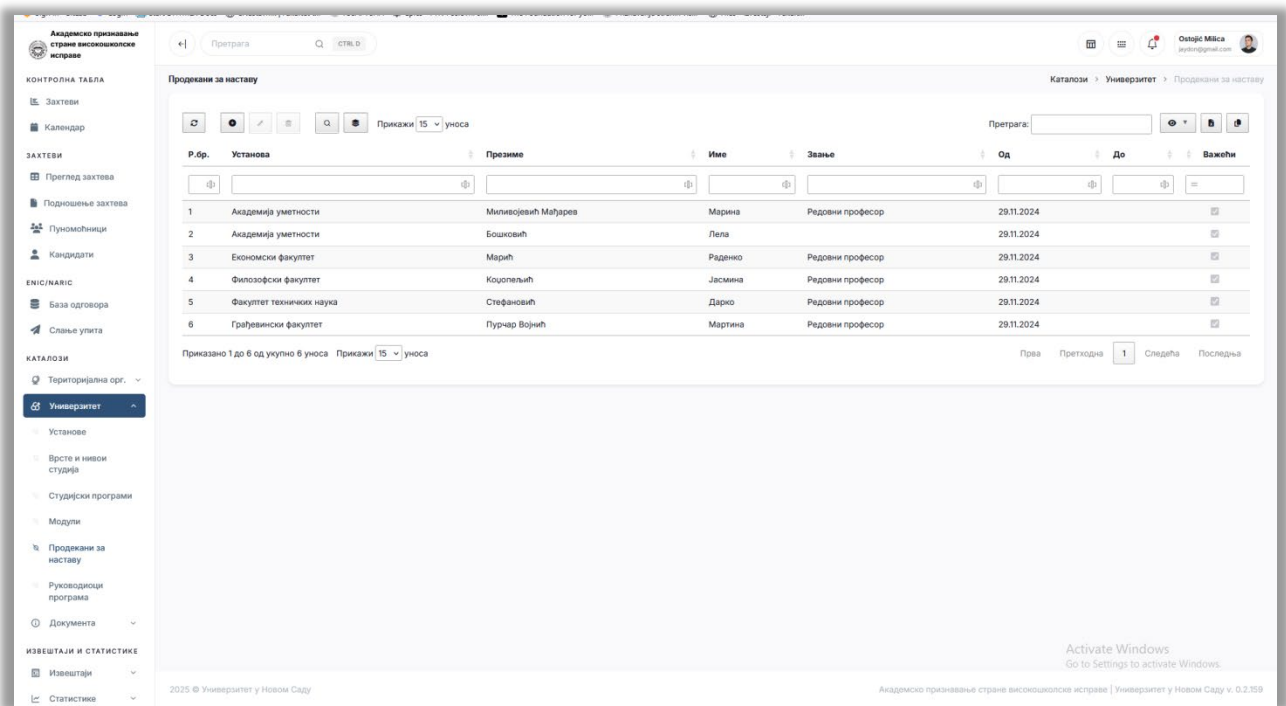
Na slici ispod nalazi se šifarnik studijskih programa koji sadrži sve studijske programe koji su akreditovani na svim članicama Univerziteta u Novom Sadu.

R.br.	Установа	Ниво студија	Назив програма	Поље
1	Академија уметности	Основне академске студије (OAC)	Аудиовизуелни медији	Уметност
2	Академија уметности	Основне академске студије (OAC)	Драматургија	Уметност
3	Академија уметности	Основне академске студије (OAC)	Етномузикологија	Друштвено-хум
4	Академија уметности	Основне академске студије (OAC)	Музикологија	Друштвено-хум
5	Академија уметности	Основне академске студије (OAC)	Глума	Уметност
6	Академија уметности	Основне академске студије (OAC)	Извођачке уметности	Уметност
7	Академија уметности	Основне академске студије (OAC)	Композција	Уметност
8	Академија уметности	Основне академске студије (OAC)	Ликовне уметности	Уметност
9	Академија уметности	Основне академске студије (OAC)	Музичка продукција	ИМТ
10	Академија уметности	Основне академске студије (OAC)	Мултимедијална режија	Уметност
11	Академија уметности	Основне академске студије (OAC)	Примењене уметности и дизајн	Уметност
12	Академија уметности	Основне академске студије (OAC)	Дизајн видео игара	ИМТ
13	Академија уметности	Мастер академске студије (MAC)	Заједничка мастер академске студије Гитара / Извођачке уметности	Уметност
14	Академија уметности	Мастер академске студије (MAC)	Аудиовизуелни медији	Уметност
15	Академија уметности	Мастер академске студије (MAC)	Мултимедијална режија	Уметност

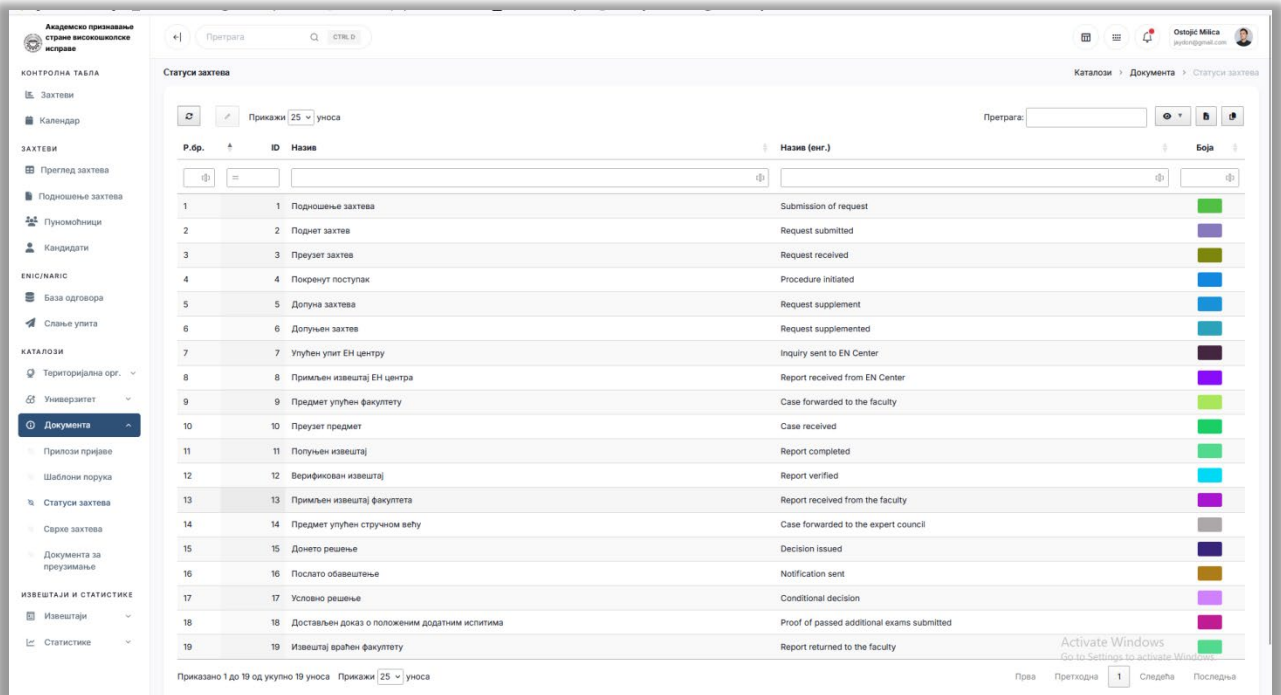
Na slici ispod nalazi se šifarnik modula studijskih programa koji su „napunjeni“ sa realnim podacima, odnosno već sadrže sve module svih studijskih programa koji su akreditovani na svim članicama Univerziteta u Novom Sadu.



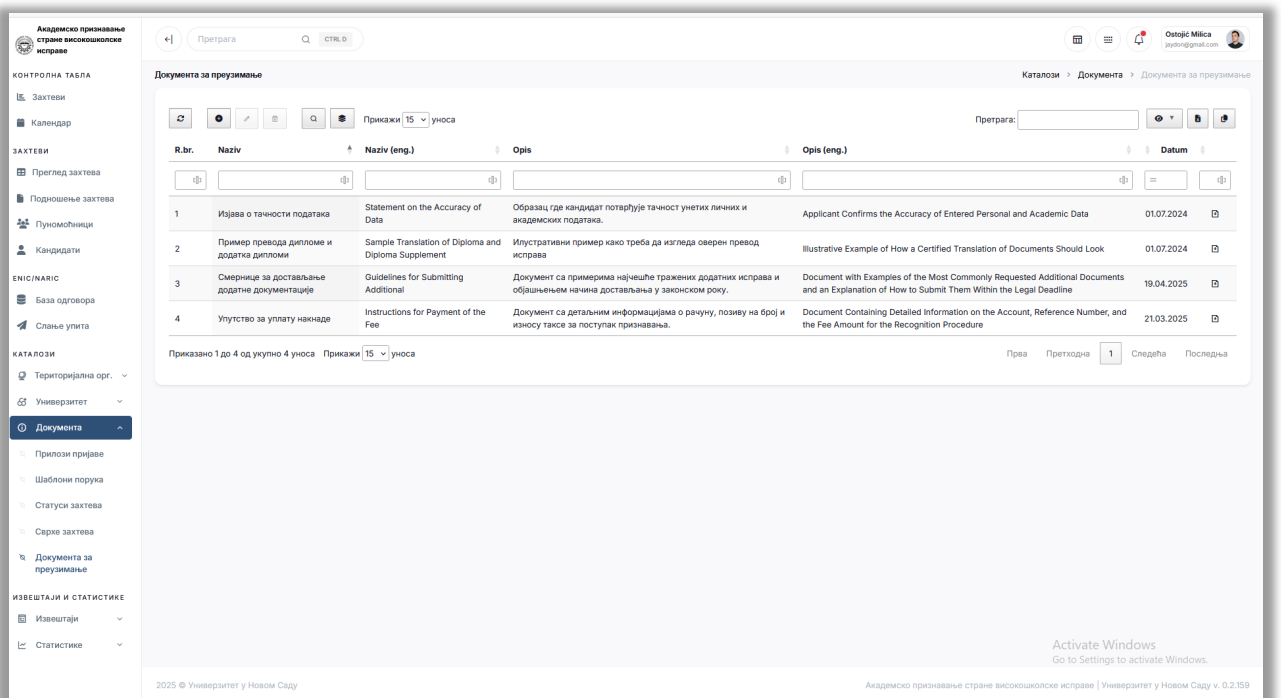
Na slici ispod nalazi se stranica predviđena za pregled i održavanje podataka o prodekanima za nastavu na pojedinačnim fakultetima.



Na slici ispod nalazi se stranica predviđena za pregled i održavanje podataka o rukovodiocima studijskih programa. Predviđeno je da ove podatke primarno održavaju prodekanima za nastavu na pojedinačnim ustanovama.



Na slici ispod nalazi se pregled šifarnika Dokumenta za preuzimanje.



Grafički korisnički interfejs iz perspektive prodekana za nastavu

Prodekan za nastavu je odgovorno lice fakulteta za proces priznavanja. Ima uvid u zahteve za priznavanje koje službenik univerziteta dodeli fakultetu. Nadgleda rukovodioce studijskih programa koji vrše postupak priznavanja, a ima mogućnost i sam da vrši priznavanje ukoliko za tim postoji potreba.

Nakon prijave na sistem, prodekan za nastavu dobija uvid u sve zahteve koji su u procesu priznavanja podneti na njegov matični fakultet.

Академско признавање стране високошколске исправе

Преглед захтева

Активни филтери - 0

Сврха захтева: Уписивање у започети ниво (14), Упис истог нивоа (3)

Врста и степен студија: ДАС (2), МАС (10), МСС (1), ОАС (3)

Студијски програм: Архитектура (4), Мехатроника (2), Рачунарство и аутоматика (11)

Обрађивач: Нема података (3), Милица Остојић (6), Наташа Максимовић (4), Уна Царевић (4)

Р.бр.	Број захтева	Презиме	Име	Сврха захтева	Врста и степен студија	Назив факултета	Студијски програм	Статус	Обрађивач
1	1-12/24	Mufso	Esther	Упис истог нивоа	МАС	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
2	1-12/29	Odhiambo	Elizabeth	Упис истог нивоа	МАС	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
3	1-12/32	Kovalenko	Mykola	Укључивање у започети ниво	МАС	ФТН	Мехатроника	Преузет	Милица Остојић
4	1-12/33	Sminov	Dmitry	Укључивање у започети ниво	ОАС	ФТН	Рачунарство и аутоматика	Поднет	
5	1-12/34	Wanlinu	Mary	Укључивање у започети ниво	МАС	ФТН	Рачунарство и аутоматика	Покренут поступак	Наташа Максимовић
6	1-12/35	Njeri	Grace	Укључивање у започети ниво	МСС	ФТН	Архитектура	Верификован извештај	Наташа Максимовић
7	1-12/38	Otieno	Peter	Укључивање у започети ниво	ОАС	ФТН	Рачунарство и аутоматика	Поднет	
8	1-12/42	Melnyk	Oleksandr	Укључивање у започети ниво	МАС	ФТН	Архитектура	Допуна захтева	Милица Остојић
9	1-12/45	Mwangi	David	Упис истог нивоа	ДАС	ФТН	Рачунарство и аутоматика	Преузет предмет	Наташа Максимовић
10	1-12/47	Lebedev	Olga	Укључивање у започети ниво	МАС	ФТН	Мехатроника	Преузет	Милица Остојић

Приказано 1 до 10 од укупно 17 уноса

Прва Претходна 1 2 Следећа Последња

Activate Windows
Go to Settings to activate Windows.

2025 © Универзитет у Новом Саду Академско признавање стране високошколске исправе | Универзитет у Новом Саду v. 0.2.159

Pored pregleda zahteva i praćenja njihovih pojedinačnih statusa, prodekan ima mogućnost da pojedinačne zahteve preuzme na obradu.

Prodekani za nastavu takođe imaju i mogućnost da održavaju listu studijskih programa i modula visokoškolske ustanove kao i listu rukovodioca studijskih programa u okviru ustanove.

Registrovanim prodekanima za nastavu na raspolaganju su i stranice sa korisničkim uputstvima kao i različitim vrstama dokumenata koje po potrebi mogu preuzeti.

Grafički korisnički interfejs iz perspektive rukovodioca studijskog programa

Rukovodilac studijskog programa je odgovorno lice studijskog programa. Vršiti priznavanje strane visokoškolske isprave i sastavlja izveštaj o priznavanju. Ima uvid samo u one predmete koje ustanovi dodeli službenik i to samo na onim programima na kojima je evidentiran kao rukovodilac.

Nakon prijave na sistem, rukovodilac studijskog programa dobija uvid u sve zahteve koji su u procesu priznavanja podneti na njegov matični fakultet i na studijske programe na kojima je on rukovodilac.

Na slici ispod ilustrovan je primer u kome Rukovodilac studijskog programa MAS Računarstvo i automatski sa Fakulteta tehničkih nauka u Novom Sadu ima uvid u sve zahteve.

Академско признавање стране високошколске исправе

КОНТРОЛНА ТАБЛА

Захтеви

Календар

ЗАХТЕВИ

Пregлед захтева

Пregлед кандидата

ИЗВЕШТАЈИ И СТАТИСТИКЕ

Извештаји

Статистике

ПОМОЋ

Упутства

Документа

Контакти

Пregлед захтева

Активни филтри - 0

Сврха захтева

Укључивање у започети ниво

Упис истог нивоа

Обрађивач

Милица Остојић

Наташа Максимовић

Уна Царевић

Прикажи 10 уноса

Р.бр.	Број захтева	Презиме	Име	Сврха захтева	Врста и степен студија	Назив факултета	Студијски програм	Статус	Обрађивач
1	1-12/24	Mutiso	Esther	Упис истог нивоа	MAC	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
2	1-12/29	Odhiambo	Elizabeth	Упис истог нивоа	MAC	ФТН	Рачунарство и аутоматика	Предмет упућен факултету	Уна Царевић
3	1-12/34	Wanjiru	Mary	Укључивање у започети ниво	MAC	ФТН	Рачунарство и аутоматика	Покренут поступак	Наташа Максимовић
4	1-12/57	Lebedev	Olga	Укључивање у започети ниво	MAC	ФТН	Рачунарство и аутоматика	Попуњен извештај	Уна Царевић
5	1-12/59	Kiptoo	Michael	Укључивање у започети ниво	MAC	ФТН	Рачунарство и аутоматика	Примљен извештај факултета	Милица Остојић

Приказано 1 до 5 од укупно 5 уноса

Прва Претходна 1 Следећа Последња

Activate Windows
Go to Settings to activate Windows.

2025 © Универзитет у Новом Саду

Академско признавање стране високошколске исправе | Универзитет у Новом Саду v. 0.2.159

Одабиrom појединачног захтева, у зависности од тренутног статуса конкретног захтева, руководиоцу је на располагању увид у детаље захтева као и одговарајући skup акција што је илустровано на слици испод.

Registrovanim rukovodiocima studijskih programa na raspolaganju su i stranice sa korisničkim uputstvima kao i različitim vrstama dokumenata koje po potrebi mogu preuzeti.