SW REQUIREMENTS AND DESIGN
E-Student Services System

LIST OF ACRONYMS

EMIS	Education Management Information System
EC	Enrolment Committee
SAO	Student Administration Office
MoES	Ministry of Education and Science
AAI	Authentication and Authorization Infrastructure

CONTENT

1 List of functions

The e-Student Services System consists of the following modules:

(SCH) Module for study programs and schedules

- Defining student programs, courses, prerequisites and rules for studies
- Mapping of faculty staff to courses
- Equivalence of courses, modules and programs
- Schedule – mapping groups, rooms and teachers

(SAM) Student activities module

- Enrolment in a semester and selection of courses
- Forming groups

(ADM) Module for the administration

- Administration of faculties and accredited study programs
- Administration of members of the faculty
- Administration of classrooms, rooms and laboratories

(ACR) Module for administration of academic results

- Administration of courses taken
- Completing semesters
- Administration of exams
- Administration of earned ECTS credits and grades from exams passed

(PID) Module for personal identification and access control

- Identification (RFID or similar card)
- Authentication system
- Access control

(REC) Module for personal records of students

- Photographing and issuing cards
- Personal records for students

(PRS) Module for presence monitoring and student activities

- Schedules for each student
- Presence monitoring/ Attendance recording

(EPY) Module for electronic payment and use of resources

- Administration of payments by the students
- Administration of the use of resources (library, Internet, photocopying)
- Administration of the use of learning systems (LMS)

(REP) Module for Reporting

- Reporting about successfully completed student activities
- Quality of completed courses (GPA)
- Electronic access to university record system
- Issuing of diploma supplement and other certificates
- Updating alumni data
- Reporting about skills and knowledge
- Issuing documents
- Issuing other papers
- Reports for the University management

(EXS) Reports for the Ministry of Education and exchange of information with other systems Interface to exiting systems

- Exchange of information with the Ministry of education

- Integration with the e-system of Student Administration Office and the e-system for Student personal file overview
- Exchange od data and reports with the Statistical Office of Republic of Macedonia

(SAC) Module for Other Student Activity

- Administration of completed student mobility cases
- Change of study program by students
- Submitting for a master thesis or a PhD
- Submitting for a diploma thesis

(CER) Module for Certificates, Europass CV, Erasmus, ECTS, Diploma Supplement

- Implementation of ECTS (European Credit Transfer System)
- Administration of ECTS
- Issuing documents and assisting mobility
- Issuing other diplomas
- Assisting employment
- Updating Europass CV

(QUA) Module for Quality of the Education

- Administration of polls
- Implementation of electronic log of completed classes
- Implementation of a system for complaints and compliments

(ALU) Module for Alumni Students Data Collection

- Administration of alumni students
- Management of CV, record of employment and career progress
- Possibility for the alumni student to maintain their profile

(XWS) XML Web Services to enable the Ministry of Education and Statistical Office access to the data

- Defined XML files to be response to the call of the web services instead of ready paper reports

The system should cover all the features arising from the unique basis of the organization and implementation of study programs and models according to the European Credit Transfer System. The system should also cover the functionalities arising from the established bases for organizing and implementation of mobility of the students from one university to another or from one to another high education institution in the country and abroad.

The system should allow flexible definition of the content and rules of studying. This means that the universities should be able independently (without intervention in the source code) to define data for the exams, the rules and criteria of studying, ways of financing and so on.

The system should cover all the cycles of study at university level. There are three cycles:

- The first cycle of study – Undergraduate Studies
- Second cycle of study – Master Studies
- Third Cycle studies – PhD Studies.

1.1 Referenced documents

Ref.	Title	Document Id.
[1]	Закон за високото образование, ,,Службен весник на Република Македонија", број 35, 14 март 2008 Измена и дополнување 19.08.2008 Одлука на Уставен суд 03.12.2008 Измена и дополнување 24.02.2009 Измена и дополнување 03.07.2009 Измена и дополнување 05.08.2009 Измена и дополнување 31.08.2010 Закон за високо образование, ,,Службен весник на Република Македонија", број 17 од 11.02.2011	http://www.pravo.org.mk/documentDetail.php?id=799
[2]	КОНКУРС за запишување студенти на прв циклус студии на студиските програми на Универзитетот „Св. Кирил и Методиј" во учебната 2010/11 година	
[3]	КОНКУРС за запишување студенти на втор циклус студии на студиските програми на Универзитетот „Св. Кирил и Методиј" во учебната 2010/2011 година	Microsoft Office Word Document
[4]	ДОПОЛНИТЕЛЕН КОНКУРС за запишување студенти на додипломски студии на студиските програми на Универзитетот „Св. Кирил и Методиј" во учебната 2010/11 година	
[5]	Best practice: Croatian National AAI service for education: http://www.aaiedu.hr/	
[6]	GEANT Identity federations: https://rnd.feide.no/identity_federations/	
[7]	Europass CV web site	
[8]	European Credit Transfer and Accumulation System (ECTS) Site	
[9]	Правилник за студирање по КС на ПМФ	
[10]	Diploma Supplement	
[11]	Statute of UKIM, UKLO, UGD, SEEU.	
[12]	ECST users guide, Brussels, 6 February 2009	
[13]	Правилник за студирање на ФЕИТ	
[14]	Microsoft, "Design and Implementation Guidelines for Web Clients", http://www.microsoft.com/downloads/en/details.aspx?FamilyId=BB0B680E-6CC9-49B8-A872-8B6DD8D9B8E6&displaylang=en	

1.2 Introduction and definitions

1.2.1 European Credit Transfer System (ECTS) and Diploma Supplement

ECTS is a system that is used by all beneficiaries, establishing academic program in all current, first second and third, cycle of studies.

Diploma supplement can be issued to the students on their request with all relevant information about educational system, courses and credits.

The existing Student Information System deals with these issues through personal contact with the Student Administration Office. There is no electronic system to support the students' requests for this kind of documents and reports.

1.2.2 Graduate Percentage Average (GPA)

Average GPA is calculated on the scale 6-10 (6 being lowest – 10 being highest).

Average GPA is included in:

- Exam certificate
- Graduate degree
- Diploma Supplement

Average GPA is not available for student and faculty staff overview through the system. Cumulative GPA is not implemented. It takes into account the credits assigned to the course.

1.2.3 Europass CV

Students have personal file accessible through the system of the Student Administration Office. However, students do not have access to their personal file, and cannot make any direct modification. The only changes available are the ones listed below which can be altered exclusively by the staff at the Student Administration Office. The following information is available:

- Name and Surname
- Address
- Contact information (telephone and e-mail)
- Previous education
- Foreign language
- Faculty + Group + Index no.

No skills, additional knowledge and extracurricular certificate information (National and International Olympiads) is included.

Students are not able to update their student information.

Faculty staff does not have access to this information; the only way to obtain this information is through direct contact with an employee of the Student Administration Office only upon request.

1.3 Objective

The e-Student Services System should automate several very important business processes of the university. The system should organize student data, employees, study programs, various data about the educational process and the faculty's research and scientific work. The system should automate the work of the student services, manage study programs and active courses offered, follow the entire course of studies for each student, the student's semesters, courses, exams, human resources management at the university, schedule of lectures, management of student payments on various grounds, distribution of resources (professors and classrooms), organization of online services for the students and various other activities.

The integrated information system for the university ought to achieve numerous enhancements:

- Fast and efficient administration
- Quality of the services provided to the students
- Engagement of the faculty staff in administering the data and information
- Accurate and updated data for the University and the faculties
- High quality, continuous and instant access to the faculty and university data.
- To fit the new system into the already present (or the future developed) infrastructure. Integrating functionalities via interfaces that will cover the whole set of university daily operations.

The objective of this project is to design a web-based system that will support the students and university management as well as to make it possible to integrate and communicate with other existing systems essential for contemporary University management.

One of the features of this system will be the reporting system. Its intent is to generate

- Student access to reports for successfully completed activities
- Student access to their Europass CV, GPA and personal file
- Student requests for issuing diploma and other existing certificates.
- Professor access to student Europass CV, GPA and personal file
- Professor access to system to upload reports concerning his courses
- Reports to university management and Ministry of Education
- Reports to the Statistical Office
- Alumni site for graduated students

1.4 General Overview

The system is intended to be used by existing and prospective students; academic and administrative staff at all of the levels (University, faculty, department) and all the resource providers outside of the University (libraries, etc), and other government institutions like the State Statistical Office and the Ministry of Education.

All rules related to the business logic should be easily defined without changes to the application (for instance min/max number of ECTS credits).

Every change made in the system by the users should be recorded – logged. The log can later be used to see who, when and where from has made the changes in the database. The system should have an intuitive user interface where a simple procedure with the fewest steps possible will enable completing the required tasks.

The user interface – the use of all components of the system by the: faculty members, associates, students and prospective students should be available only through a WEB interface. The WEB should use modern technologies that will bring fast response with small burden on the server by doing most of the validation process on the client side. All user forms should have fast access to on-line help, context dependent. The components used by the system administrator or office users can be implemented as standalone desktop applications or WEB applications. Additional possibilities for an additional web interfaces for small devices (mobile phones) should be foreseen

1.5 Scope

The aim of the proposed system is to automate the entire system of student programs, courses, the relationships among courses, courses with programs, mapping of teachers with courses, equivalence of courses. The process of studying, enrolment of students in semesters, enlisting in courses, exams passed, ECTS credits earned should all be automatically managed online. Members of faculty, classrooms, laboratories and all resources should be administered with this module. The data used by the system is to be stored in a database. A web-based application is to be provided for the various users that will enable them to administer their small portion of the entire system, with high regard of hierarchy of decision making, data access control and flow of data and processes. Each user can access its portion of the system allowing for distributed, but accurately located responsibility, quick update of data and timely access to completed results. This will enable the process to be simplified and considerably quickened, making the jobs of the people involved, especially the university staff and admission officers easier and faster. The ultimate acceleration will be at the students end, enabling online access and management of their activities, obligations and usual steps a student makes during studies. The system should be able to support the current process, but is meant to centralize it and make possible for the processes to go faster, easier and more accurate. The system should enable the future extension to almost fully paperless management as soon as laws allow it.

As a part of the system, the authentication and authorization Infrastructure is proposed (AAI). This infrastructure should cover many levels of user authentication and authorization. It will begin from the level of presence identification using RFID or other means of presence identification. The infrastructure will also enable secure authentication of users toward multiple heterogeneous resources (networks, networking services, web applications, libraries, etc), while distributing the responsibilities for the management of the users/resources databases to the home institutions of the users (or institutions owning the resources).

The integration with the other internal systems gives a complete view to the user's data from many different perspectives. The internal systems that are taken into account are: the financial system, e-mail, learning management system and the library system. The connection to the financial system will provide fast interaction with the student financial status and the Core functionalities. The e-mail system will provide fast communication with the students. The connection with the used LMS will give the opportunity to easily choose courses from different

faculties or to easily see the student's achievements within enrolled course. The connection to the library system (on University or faculty / division level) should manage to connect the courses within one university with the course materials – the library resources. It will provide automation in the process of getting information about the available material, managing the library resources within a semester or provide priority in issuing the material to the end users.

Another aim of the proposed system is to automate the reporting system as well as to present an interface to the existing systems. The data used by the system are to be stored in a database. A web-based application is to be provided for five different users; Student, Professor, Teaching assistant, Administration officer, Administrator. The Student user should have all the activities described above, from viewing reports to editing his/her personal file skills and additionally obtained certificates. The Professor should be able to have review on the student's progress only on the courses taught by the professor, as well as to create and upload reports concerning exams, courses, statements and novelties. The Teaching assistant should have more limited access than the Professor account; however their roles should be quite similar. The Administration officer should be able to report to existing and potential students whether it is their enrolment procedure or certificate request. These specifications will be further analyzed and defined. This will enable the process to be simplified and considerably quickened, making the jobs of the people involved, especially the university staff and admission officers easier and faster. The system can supports the current process but is meant to centralize it and make possible for easier communication and information distribution for the students, faculty staff and the organization concerned with these data.

The system should enable electronic workflow realization fo all student activities, including fully paperless admission, enrolment and all relevant university management processes for provision of student services, diploma/certificate/report requests and exam/course reports.

1.6 Existing System(s)

No university has an integrated web-based system that can support the process of ECTS. There are isolated software applications that are not an integrated IS.

At the moment, there is no AA infrastructure established in Republic of Macedonia and no unified system for student services and communication with the Ministry of Education. Also there is no complete realization of electronic Student services.

The current infrastructures have no international connectivity and visibility. The only example of international visibility is the eduroam infrastructure (which is in the process of establishment at UKIM – on 2 locations so far).

There is no connection in the Universities in Macedonia from a University library to courses developed in the curricula of the faculties. There is central University library, but also libraries exist within every faculty, even within the divisions of the faculties. The interested students borrow, use and return the library materials, but not in a timely fashion. The proposed Library interface will manage the Library resources more efficiently.

Most universities already have in place LMS, finance management systems and e-mail student's accounts. Some of them have different forms of presence and/or access control systems.

1.6.1 Existing systems in EURM

Registration of a semester

Students first complete paper application and deliver it to an administrative officer. After that the administrative officer checks the application and verifies it in the electronic system, but this part is linked with the financial department. If the student hasn't paid all fees related to semester registration, the system will automatically inform the administrative officer that the student is not eligible to register. If all the fees are covered by the student, then he/she will be qualified to register the semester.

Disadvantages:

- No electronic registration of semester.
- Deans and other responsible personal (professors, assistants etc.) cannot see any reports with the registered students.

Registration of courses

Current system doesn't provide any services related with registration of courses.

Registration of elective subjects

The registration of electives goes in old-fashioned way. The student needs to fill out a form, and chose one or more of the electives. After that administrative officer collects all forms and prepares a list of students for all electives. Then the list is delivered to the dean.

Disadvantages:

- No electronic registration of electives
- The system has no mechanism to identify what each student has chosen.
- Professors cannot see any electronic list of students who chose their subjects.

Class schedule

Current system doesn't provide any services related with class scheduling. Class schedule is attached to separated systems like: Electronic Notice Board and Moodle.

Pro-active customized notification

Current system doesn't provide any pro-active customized notifications.

Administration of completed courses and ECTS

After completing the process of registration exam, administrative officer collect applications and submit them to the professor. After finishing the exam the professor is obligated to enter all grades in the application forms and then return them in student office, in a period of 7 days. Administrative officer verifies the application forms and grades into the system. If any mistakes have happened the administrative officer can make changes.

Disadvantages:

- Professors have no access to the electronic system, so he cannot be sure that the administrative officer correctly put the grades into the system. The only way to check is through personal contact with the Student Administration Office, and asks them to provide him with the list of grades for the particular course.
- Current electronic system is not compatible with ECTS.

Reporting

Current system provides reach set of reports, but the main problem is that professors don't have any access to the system.

1.6.2 Existing systems in FON University

FON University has the following systems:

- System for student affairs
- ERP system
- System for class attendance
- Enrollment system
- Learning Management System

All of these systems are working independently because all of them are from various vendors. In this moment their integration is based on exports from one system to another.

Student Services

The system for student affairs is web-based system with a lot of bugs and lacks of functionalities. The main issue is its incompatibility with a financial system; so many issues are solved manually. Decision of the management is definitive replacement of this system. A new system is in a development phase, so many business rules in this project, proposed by FON University are appropriate to the new system.

ERP

The ERP system that is installed at FON University is complete solution provided from one of the leading software vendors in the region, so it is bug-free system and the new system for student affairs is built up on top of this system.

Attendance

A system for class attendance is Soyal based solution with pair of card readers set on each classroom. On top of this solution there is a web based solution for reporting about attendance of teachers and number of students. This should be upgraded to be integrated with class schedules and student files for reporting needed for ECTS implementation.

The enrollment system is basic web-based system for collecting information about the candidates, and later to provide a complete ranking list for each enrollment cycle which is edited manually. The export from this system is used to fill the student files in the system for student affairs.

Learning Management System

The platform for Learning Management System is Moodle. It is system which will be kept as LMS system on the university, and it is expected that this project will bring integration of the system for student affairs with this LMS.

1.6.3 Existing systems in UGD

UGD realizes smaller AA infrastructure. At UGD, there is already a platform for E-Learning (Moodle) and independently of it is developed service called E - index. E-Learning system (Moodle) exists several years and now is a standard component of the university. E – index although is at an early stage of implementation works flawlessly.

1.6.4 Existing systems in UKIM

UKIM is the oldest and most heterogeneous university in the Republic of Macedonia. That is the source of most of the problems in the processes of its centralization and the aim of building an integrated information system for it. For a very long time faculties that are members of UKIM have developed different process rules for most of the student and faculty activities. Interfacing to these systems is of great importance and is crucial for the success of the proposed system.

Furthermore, fundamental differences exist among the areas taught staring from technical sciences, social science, art, architecture, music, medicine, physical education... just to name a few. However, uniting factors like the implementation of ECTS and equal legal obligations should be the starting point when building an integrated information system. Certain adaptations to common business processes should be undertaken by members of the university for the system to be applicable everywhere. The system should be versatile and robust to accommodate the needs and processes existing in every faculty member of UKIM. In the process of deployment of the Information system, members from each faculty should test, configure and accept the final common software solution. The process of importing legacy data from all faculties in the system should also be undertaken and the integrated information system should provide the means to import legacy data from older existing systems.

1.6.5 Existing systems in UKLO

First, students fill different paper applications and submit it to the Administrative office (Student's affairs). The officer checks the applications submitted and verifies in the electronic system. It's the same for the student's scholarship payments, exam payments, semester payments and etc… The Students affairs are not linked with the financial department. It means that this department has not the real-time information about student payments. It is important these two departments to be linked and student can access through web site to have an overview of all necessary payments, calculation of amounts and view all necessary bank accounts and examples of completed orders.

Also, there is no electronic registrations of semester, applications for exam, electronic registrations of courses, and etc. There is no automatic calculation for the acquired credits and checking accomplishment of the conditions for taking a particular exam. The system should provide an opportunity for updating data items (type of subject, title, number of classes, credits, professor, etc.). . In this way base is formed from which educational programs for years and guidance will be created.

When the process of registration exam is completed, the administrative officer collects applications and submits them to the professors. When the test ends, the professors enter all grades into the applications form end return them to the Administrative office, and the administrative officer verify them into the system. There is a possibility for mistakes, and the professors have no access for that kind of imported data and cannot be sure that the grades are correctly stored in the system. The only way to see how the grades are entered in the existed system is to ask the officer to print a list of entered grades.

Current existing system is not compatible with ECTS.

1.7 Benefits

The aim of the proposed system is to address the limitations of the current systems existing in different universities or faculties members of universities. The requirements for the system have been defined according the defects recorded in the past and are also based on the feedback from the users. Following are the objectives of the proposed system:

- Online access to system modules and distribution of work and responsibilities to students, faculty staff, officers and university management.
- Reducing time in activities. Reduce the time to update courses and programs, reduce the time and effort it takes for a student to enrol in a semester or enlist in a course, reduce the time for application to an exam or the time for the exam results to be updated reduce the time taken for the students to apply, process exam/GPA/course/individual reports, enrich student personal file.
- Centralized data handling. Transfer data smoothly to all departments involved and handle the data in a centralized way. Organize reliable backups. Enable fast and timely access to accurate and updated data.
- Paperless management of processes with severely reduced manpower. Reduce the manpower needed to perform all the student studies tasks by reducing the paper works needed and by distributing responsibilities.
- Cost cutting. Reduce the cost involved in the business process included.
- Operational efficiency. Improve the operational efficiency by improving the quality of the process.
- Simple and efficient mechanism for their authentication and authorization for multiple, heterogeneous systems.
- Integration of all necessary data from different sources. From the point of view of the resource providers, it will simplify the management and the allocation of the resources to the users. It will also enable user mobility within and between the universities. From the point of view of the educational authorities, there will be a central collection of the shared resources committed to the higher education institutions (identified by the member institutions and/or services).
- Connection to the University Library System (ULS) and to the course material provider. The idea is to enable direct automatic connection with the courses developed in the different curricula in the University.
- Reach to geographically scattered students. One of the important objectives of the reporting system is easier communication with all geographically scattered students.

1.8 Goals

The main goal of the system is to automate the process carried out in the organization with improved performance and realize the vision of paperless university management and electronic workflow of student activities. Some of the goals of the system are listed below:

- Manage large number of student details.
- Create student accounts and maintain student's data in an effective way
- View all the details of the students.
- Create statistical reports.
- Enable the students and faculty staff members to take proactive role in the process by entering their portion of data in the system
- Provide the students and faculty staff with updates of the data they are interested in and entitled to access.
- Provide interfaces to all existing and future systems in the university, complementing the iKnow infrastructure to an even more powerful system.
- Establish AA infrastructure as simple, yet robust and secure mechanism for user authentication and authorization toward various shared resources with possibility to enable unique identification; Support RFID and biometric identification; Enable administration of personal files; Provide authentication of each user by its home institution; Provide authorization for university resources and services; and Send accounting reports to university management.
- Interfacing the financial, e-mail, library systems and the LMS will achieve the following goals: faster and timely fashion communication with the students, immediate information about their financial status, control over the library resources; Better communication and knowledge-transfer; etc.

2 User and functional requirements

2.1 Users

The software should enable creation of users and user groups through its user interface. For every new user created, the administrator of the software should be able to specify the forms and modules within forms that the created user can access and work with, with various levels of user permissions. All activities are tracked and stored. The following users and groups are to be initially created:

ID	User groups	Roles
IAD	IT Administrator	• Performs regular maintenance of software applications, servers. • Manages backups of databases, • Upgrades hardware / software. • Create and manage user accounts for other user groups • Manages all setup data, configurations • Manage the student and staff accounts using standard LDAP schemas • Enable and revoke privileges • Maintain and monitor the infrastructure • Provide hosting for the AAI services for institution with smaller members • Maintain the users/resource database (LDAP and associated servicer) • Maintain the identities of the users (identity providers)
ADM	Administrator	• Create and manage user accounts for other user groups • Create and manage courses, study programs, connections among courses and programs, revisions of study programs • Update and manage global information stored in the system (for instance changes in faculty staff, cities, countries, municipalities) • Manage the student and staff accounts using standard LDAP schemas • Enable and revoke privileges • Manages all Faculty/study program data specifics of the above mentioned processes • Create and manage user accounts for other user groups • Manages all setup data
TAD	Technical Administrator	• Manages the reports parameters. • Create and manage user accounts for other user groups of reports.
RAD	Resource administrators	• Define access to their reports using standard LDAP schemas
UCC	University computing centre	• Maintain and monitor the infrastructure • Provide hosting for the AAI services for institution with smaller members • Maintain the users/resource database (LDAP and associated servicer) • Maintain the identities of the users (identity providers)
FCC	Faculty computing centre	• Maintain the users/resource database (LDAP and associated servicer) • Maintain the identities of the users (identity providers)
HRD	HR Department	• Require data on the past and current engagements of the teaching staff.
HSS	Head of the student service	• Create and manage courses, study programs, connections among courses and programs, revisions of study programs • Overview reporting (subset) • Receive notifications for special requests made by students online and confirm/deny requests. • Manually modify data for a student in special circumstances (late enrolment, special quota, various special cases) • Analyses statuses and response times • Analyses quality of service provided • Joins faculty with courses

ID	User groups	Roles
		• Creates semesters exam terms • Views and analyze reports and data • Analyses statuses and response times • Analyses quality of service provided • Issues reports to the University management • Issues reports to media / Ministry of Education • Monitor and supervise the process carried out by Faculty / Study program • Work in the office for communications with students • Provide administrative services for students • Manage students data • Coordinate work with faculty members and provide services
SSS	Administrative staff of the student service	• Be able to manually perform all activities that students can do through the online system. Manually modify data for a student in special circumstances (late enrolment, special quota, various special cases) • Be able to perform some of the activities that faculty staff users can do through the system (enter grades for students exams, export lists with exam candidates) • Generate reports (subset) • Generate notifications to students, faculty staff etc. • Generate and print documents upon students requests • Joins faculty with courses • Creates semesters, exams terms • Views and analyze reports and data • Analyses statuses and response times • Analyses quality of service provided • Issues reports to the University management • Issues reports to media / Ministry of Education • Monitor and supervise the process carried out by Faculty / Study program • Work in the office for communications with students • Provide administrative services for students • Manage students data • Coordinate work with faculty members and provide services • Receives and reviews diploma/certificate requests • Sends answers/reports • Edits FAQ
FIN	Administrative officer of the department of financing	• Generate reports with financial implications (for both students and faculty – visiting lecturers, honorary payments of additional activities) • Monitor student payments
ECTS	ECTS coordinator	• Create and manage courses, study programs, connections among courses and programs, revisions of study programs • Generate reports (subset)
R	Rector	• Generate reports (subset)
UM	University Manager	• Generate reports (subset)
AVR	Academic Vice-Rector	• Generate reports (subset) • Analyses statuses and response times • Analyses quality of service provided
AVD	Academic Vice-Dean	• Generate reports (subset) • Make decisions on high level student requests for special cases submitted online • Analyses statuses and response times • Analyses quality of service provided
DEAN	Dean	• Generate reports (subset)

ID	User groups	Roles
FFO	Faculty Financial Officer	• Generate reports with financial implications • Make decisions on high level student requests for special cases submitted online • Provide data about payments
PROF	Professor	• Access resources using single credentials • Authenticate and authorize in home and visiting institutions (mobility) • Teaches • Updates results from exams in the system • Provides data in the system about exams • Views reports for student enrolment in courses and attendance • Update other data (diploma thesis results etc.) • Create and manage courses • Create and organize sections in every course • Create and manage assessment questions and possible answers • Views and analyze results • Analyses learning statuses and response times • Analyses quality of service provided • Issues reports to the University management • Views reports on student personal file, skills and additional extracurricular certificates • Issues reports to the University management
TA	Teaching Assistant	• Access resources using single credentials • Authenticate and authorize in home and visiting institutions (mobility) • Provides data in the system about exams • Views reports for student enrolment in courses and attendance • Update other data (diploma thesis results etc.) • Manage courses (subset of parameters)
STT	Student Tutor	• Access resources using single credentials • Provides data in the system about exams (Grades for laboratory activities of students, etc)
STU	Students	• Access various resources using single credentials • Be identified for presence at lectures using RFID • Sign up semesters • Authenticate and authorize in home and visiting institutions (mobility) • Apply for exams in the system • Enlist for courses • View their current status • Submit seminar thesis, diploma thesis etc. • Registers to the system • Inputs personal details, fills application • Fills details from his/her certificates (academic scores) • Submits / prints application/ reports/diploma/ certificates • Views FQA • View results and progress • Take assessment, • Check results, • Change personal information, • Communicate with other students • Update personal details, skills and additional extracurricular certificates and activities
MOE	Ministry of Education	• View various statistical reports for applied / admitted candidates or students
SSO	State Statistical	• View various statistical reports for applied / admitted candidates or students

ID	User groups	Roles
	Office	
LB	Library Officer	• View reports of active students to import to library information systems • Import data for students owing books to the university • Access the Library resources and student's library accounts
PG	Parent/Guardian	• View reports for a student and his progress in the studies
SP	Scholarship Provider	• View reports for a student and his progress in the studies
EVA	Evaluator	• View reports (subset)
ACM	Accreditation committee member	• View reports (subset)

2.2 Functionalities

2.2.1 Tabular overview

Basic functionalities of the system are specified as follows:

ID	Functionality	Description	Objective	Module	Users
[1]	Administration of students data	Enable administration or only preview of the students personal data. Data needed for reports, statistics, contacts and guidance during studies.	Insert/edit students personal and background data. Students can preview part of their data and update some of the information themselves.	Core	SSS STU
[2]	Administration of students enrolment details	Enable administration of the parameters of a student at the time of enrolment (initial program, tuition size etc.)	Manage and maintain information for the details for the beginning of the studies of the student	Core	SSS
[3]	Administration of the student semesters	Sign in new semesters for the student in question, change programs, tuition group – amount. Close completed semesters	Keep track of the progress the student makes during the years and manage his/hers current status/program/tuition.	Core	SSS STU
[4]	Administration of students courses	Enter the student's choices for elective and mandatory courses for the signed semester in question. Calculate credits for the semester based on the courses chosen.	Manage the courses the student ought to attend in the semester. Control of overall credits earned during the semester. Implement control mechanisms for repetition of failed exams. Change of elective students.	Core	SSS STU
[5]	Administration of student exam applications	Enter student exam applications or manage student exam applications entered online.	Manage student's applications to pass exams. Students can apply for exams by themselves. Officers can make changes in the applications if needed	Core	SSS STU
[6]	Administration of student's exams	Insert or update grades achieved by the student on a particular exam. Review list of exams passed.	Manage student's exams.	Core	SSS PROF
[7]	Administration of seminar thesis	Insert or update details of the seminar thesis undertaken by the student	Manage student's seminar thesis.	Core	SSS STU PROF
[8]	Administration of diploma thesis	Update details for the diploma thesis that the student works on	Manage diploma thesis	Core	SSS STU PROF
[9]	Administration of exam terms	Add new exam terms when exams are scheduled	Manage exam terms	Core	SSS
[10]	Administration of semesters	Create new semesters with appropriate list of activated courses eligible for taking. Join courses with professors for the	Manage semester details	Core	SSS

ID	Functionality	Description	Objective	Module	Users
		semester in question			
[11]	Administration of quotas and tuition prices	Inserting and updating tuition details for various quotas that students can enroll in, costs for different programs and quotas.	Manage costs for studying		SSS
[12]	Administration of courses in programs	Make relations between programs and courses that can be taken in the program in question	Manage distribution of courses in program	Core	SSS
[13]	Administration of courses	Insert or update details for each course, ECTS credits, fall/spring semester, number of classes	Manage details for the courses	Core	SSS
[14]	Administration of programs	Insert/update details for the study program that a student can enroll in (start year, bachelor/master/PhD level etc).	Manage program	Core	SSS
[15]	Administration of program revisions	Insert revisions of program	Manage revisions of program	Core	SSS
[16]	Administration of members of the faculty	Insert new employees, update details for the current employees / faculty members	Manage faculty members	Core	SSS
[17]	Diploma thesis report	Present a list of completed ongoing diploma thesis work	Review diploma thesis	Reporting	SSS
[18]	Exams report	Present results from exams with various filtering options	Review exam results	Reporting	SSS STU
[19]	Graduated report	Present list of graduated students	Review graduated students and various statistics	Reporting	SSS
[20]	Elective courses report	Present a list of students per course per semester	Review students taking a particular course	Reporting	SSS
[21]	Master book	Prepare the report for printing the master book requested by the ministry of education	Exact columns and details as requested by the ministry for all active and graduated students	Core	SSS
[22]	List students per program	Present list of students with various filtering options	Review students per program, per semester, per year etc	Reporting	SSS
[23]	Seminar thesis report	Present list of seminar thesis	Review seminar thesis	Reporting	SSS
[24]	Administration of student mobility	Insert / update details for student exchange programs the student has participated in.	Manage student mobility	Students	SSS STU
[25]	Administration of Master thesis	Update details on the Master thesis the student is working on	Manage the student's master thesis	Students	SSS STU
[26]	Issuing documents	Issuing various documents for the student (list of courses and grades, proof of enrolment, etc)	Easy generation of requested documents	Students	SSS STU
[27]	Issuing news and announcements	Post news and announcement or mail bulk mails to students	Manage communication with students	Students	SSS
[28]	Administration of financial services	Create and manage various services with financial implications	Set prices for services	General data Core	ADM
[29]	Administration of lookup data	Administration of cities, countries, municipalities and other general data used in various modules in the system	Manage lookup tables	General data Core	ADM SSS

ID	Functionality	Description	Objective	Module	Users
[30]	Administration of class schedule	Management of classes, rooms and groups of students	Overview of taken rooms	Core	STU PROF SSS
[31]	Administration of faculties	Administration of the faculties in the university	Manage faculties	General data	ADM
[32]	Administration of student sign outs	Administration of the students leaving the university	Manage the students that left the studies	Students	STU SSS
[33]	Retrieval of enrolment data	Retrieve data from the enrolment module when accepting a new student at the university	When the student finally applies for the university after being accepted, the data provided when enrollment should be transferred to the core database	Students	SSS
[34]	Establish AAI infrastructure	Establish the necessary infrastructure for supporting the AAI	Make the initial infrastructure	AAI	UCC
[35]	Maintain AAI infrastructure	Maintain the central point of the infrastructure, LDAP schemas, and registry of member institutions, software repository.	Keep the infrastructure operational	AAI	UCC
[36]	Support the member institutions	Provide support to FCC to establish the identity providers and resource providers	Establish identity and resource providers	AAI	UCC FCC
[37]	Manage users identities	Open accounts for students and faculty staff, manage levels of authorization, revoke accounts	Define the single user identity	AAI	ADM
[38]	Manage resource access	Define access rights for own resources with respect to user roles and responsibilities	Define the access policies	AAI	RES
[39]	Synchronize AA data with Core module data	Maintain synchronized student records between LDAP directories and core modules	Maintain student records	AAI	ADM
[40]	Access student resources	Using single credential, access all the needed resources.	Simplify the access model to resources	AAI	STU
[41]	Access faculty resources	Using single credential, access all the needed resources.	Simplify the access model to resources	AAI	FAC
[42]	Staff and students mobility	Access resources in visiting institutions using home identities	Enable roaming access	AAI	STU FAC
[43]	Presence identification	Using RFID or other means, identify physical presence of students	Presence identification	AAI	STU
[44]	Issuing of personal ID cards – student ID	Take pictures, compose layout and print student e-ID cards	Establish e-ID for the students	AAI	ADM
[45]	Maintain student records	Manage the students' records.	Student records.	AAI	ADM
[46]	Track classes attendance	Using the schedule and the presence records, track the attendance to lectures	Attendance logs	AAI	ADM
[47]	Establish mailing lists	Using the data about the course enrolment, establish appropriate mailing lists	Email communication infrastructure	AAI	ADM
[48]	Get financial info for students	Check the payments from students for their invoices for services	Check financial records	AAI Core	FIN

ID	Functionality	Description	Objective	Module	Users
[49]	Generate invoice for service	Generate invoices based on the fees for each type of service (semester, exam, penalties, …)	Generate financial documents	AAI	ADM
[50]	Exchange data with LMS	Populate LMS courses, extract grades	Data exchange with LMS	AAI Core	ADM
[51]	Information about books recommended by a course	Borrow a book recommended by a course or other books from the same author that ULS possess	Simplify the students access to ULS resources	AAI Core	STU
[52]	Manage ULS resources	Define rules for returning the borrowed books in different administrative situations.	Control the ULS resources	AAI Core	ADM
[53]	Acquire HR information	Get information for the needs of the HR systems, on the engagements of the staff	Support HR	AAI Core	HRD
[54]	Customizing the user types, layout of the reports and the module accessibility	The administrator should have opportunity to customize the layout of the reports. Based on the university preferences, every university should be able to fully customize which reports will be accessible from the module	The system must be fully customizable by user types, reports types and reports layout. To create and manage the reports templates and the accessibility to specific reports	AAI Core	ADM SSS
[55]	Manage look-up tables	The administrator should have opportunity to manage the look-up tables used in the report process (exam report form, exam application form, …)	To manage the date that can appear in reports in an organized way	AAI Core	ADM
[56]	Create/edit/delete exam reports and notifications	Faculty staff should be able to create/edit/delete exam reports that he/she has created. Reports should be signed with digital signature.	These exam reports should be viewed by the students and accessible by the Administration office staff	AAI Core Reporting	SSS
[57]	Fill exam applications	Faculty staff should fill in the exam applications of the students who applied for his/her course exam with the appropriate details	These electronic exam applications are to eliminate the need of the Administration office to fill them by hand in the system.	Core	SSS STU
[58]	View student's data/ reports on student attendance	Faculty staff should be able to view student's personal information, additional skills, certificates, course enrolment	Faculty staff should have information on how to contact the student, should be aware of his abilities and interests in order to define a proper practical project and internship	Core Reporting	STU PROF SSS
[59]	Student registration	Students register on the system by entering their university e-mail. After the account activation the student can use the system	Students can register on the system after they have been accepted in the university.	Core Reporting	STU SSS
[60]	Viewing completed activities	Students should be able to have overview of his/her activities status.		Core Reporting	STU SSS
[61]	Filling/updating personal data	Students can fill their personal data. The data involve basic information and additional skills, activities and obtained	The student can recheck and correct the entered data anytime during his studies at the university.	Core Reporting	STU SSS

ID	Functionality	Description	Objective	Module	Users
		certificates	This could be implemented through filling template form of EuropassCV.		
[62]	Registering/ Viewing course information	Students should be able to register to a course with password given by the specified professor.	Students should be able to register to any course where they have permission, and to be able to follow the materials uploaded and the given assignments. Registered students should be able to involve themselves in the discussion forum for every course.	Core Reporting	STU SSS
[63]	Filling complaint/ question	Students should be able to rise complain about an error in his/her data in the reports. Students should be able to address administration staff about unclear questions.	To allow the student to request correction of mistakes or revision of his academic/exam results or to obtain an answer by the university staff.	Core Reporting	STU SSS
[64]	Request diploma/ certificate	Students should be able to request issuing of diploma or certificate by the Administration Office.	The Administration Office will have data of all the students requesting diploma/certificate issuing. The system should automatically check and return result whether the student fulfills the requirements to be issued a diploma/certificate.	Core Reporting	STU SSS
[65]	Ranking the students	The system should be able to automatically rank all the students for every study program based on their average and cumulative GPA	To produce list of students in descending order, that will allow information on the top students for a specific study program.	Core Reporting	STU SSS
[66]	Produce various statistical reports for existing students	The system should produce various statistical reports for all existing students based on their average and cumulative GPA	Statistics reports	Core Reporting	STU SSS
[67]	Generate various statistical reports for applied / admitted candidates	The system should generate various statistical reports for all candidates that have applied, for the accepted and rejected ones	Statistics reports	Core Reporting	STU SSS
[68]	Generate various statistical reports for students	The system should generate various statistical reports for all students	Statistics reports	Core Reporting	STU SSS
[69]	Generate various statistical reports for faculty staff	The system should generate various statistical reports for faculty staff	Statistics reports	Core Reporting	PROF SSS
[70]	Generate various statistical reports for university	The system should generate various statistical reports for office member of university	Statistics reports	Core Reporting	PROF SSS

ID	Functionality	Description	Objective	Module	Users
	administration (SSS)	administration			
[71]	Generate various statistical reports for supporting personnel	The system should generate various statistical reports for supporting personnel	Statistics reports	Core Reporting	SSS
[72]	Audit log	The system should keep audit log with time stamp and user ID of all changes taking place in the Reporting module	The Administrator will have full access to the audit logs in order to easily track certain misuse or faults in the system	AAI Core Reporting	ALL
[73]	Reports for the Ministry of Education	Generate reports that are requested by the Ministry of Education	To automate the process of data query from the faculties and creation of reports from the universities to the Ministry	AAI Core Reporting	SSS
[74]	Reports for the Statistical Office	Generate reports that are requested by the Statistical Office	To automate the process of data query from the faculties and universities to the Statistical Office	AAI Core Reporting	SSS
[75]	XML Web services to MOE and Statistical Office	The system must enable web services for exchange of info	The ministry of Education and the Statistical office to be able to customize their internal reports using the data from the universities	AAI Core Reporting	SSS
[76]	Alumni Site	Alumni site with CMS capabilities to follow the career of the alumni students	Create and follow a network of alumni students	AAI Core Reporting	STU SSS

2.2.2 Detailed description

Administration of students data

This functionality is accessible to the backend officers in the student services as well as the student himself/herself. The office staff can change/update all the parameters, while the student can only update contact information, and read the remaining information. Most of the data is automatically imported when the student is enrolled at the university from the enrolment module, and changes are rare later during studies.

If the system entered a temporary identification number for foreigners, the system automatically displays fields for entering the passport number and date of expiry of passport and address of temporary residence.

The system should provide sequential generation and allocation of index numbers respectively within the university. The system will not allow regeneration and allocation of existing index number for two students.

Administration of students enrolment details

The parameters for the student at the time of enrolment should automatically be imported after they have officially enrolled, from the enrolment module. However certain corrections and updates might be necessary, so after the data is imported, a form should be accessible for corrections for these parameters for each student. The form should be accessible for editing by the officer at the student services department, and in read only form to the students themselves as well as other interested parties (vice-dean or enrolment commission members).

Administration of the student semesters

Whenever a semester starts, the students should sign in for the following semester. That is the point when changes in programs are made or when the student can change tuition group (for example move from state sponsored to a private quota due to multiple repetitions of major courses). Finished semesters are also marked in this module. The form is accessible to students or to officers in the student services department. Students can update most of the data, but confirmation for program or quota changes are necessary from officers or vice-deans. When students demand irregular sign ups (changes in study program or quota), an automatic request is posted to the officers with a notification to confirm or deny the request. Also, the students are served predefined parameters when signing up according to their enrolment choice and ranks, as well as previous semesters. If changes are imposed due to the student's style of studies (repetition of courses over the defined maximum), the system serves the implied changes in the quota to the student when he/she accesses the module to sign in the following semester. When the student applies to start a new semester, the system retrieves information from the library module about the books borrowed by the student. If the student holds books borrowed for longer than a year, the student is not allowed to start the new semester until he/she returns those books.

A student can request for a semester off from the Faculty Administration if he/she has eligible reasons (medical reasons, travelling, etc) that has to be approved by the Faculty authority (Vice Dean …). There are no fees for that semester. In the students e-index there is mark that he/she is inactive for that semester. During the period, the student is not allowed to take any exams. If the student holds books borrowed for longer than a year, the student is not allowed to take the semester off until he/she returns those books.

The system should notify the Head of Student Affairs for students who have not regulated the status 15 days (or other configurable number) after the deadline for regulating the academic year. The system will notify about students who have not verified semester after the deadline for validation of the semester.

The system should allow for defining different modes of study (part time, full time).

The module calculates the cost of the semester, and requests from a web service from the financial department if the student has completed payment or has money on his/hers account. The module updates information to the financial department via a web service that the semester has been signed, and money should be deduced from the student's account. The semester enrolment is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Administration of students courses

When the students signs in a new semester, he/she should also enlist in courses for the semester in question. This module serves for the student to enter choices for elective and mandatory courses for the semester. The module calculates credits for the semester based on the courses chosen. The module should implement control mechanisms for repetition of failed mandatory exams. Also the module serves the student only courses that he/she is eligible for (part of the student's study program; beware of connections with previous prerequisite courses). The module implies repetition of previously failed mandatory courses. Control of overall credits earned during the semester is to be implemented. Confirmation from a officer after being automatically notified is necessary for irregular choice of courses made by the student (exceeding maximum number of credits per semester, changes in elective courses that the student has failed the previous semester, or other irregularities). The module can be accessed by officers for special cases when the student is unable to complete the job or when changes after deadlines are necessary. According to the courses enlisted, and credits earned, the appropriate sum for payment for the semester is calculated based on the quota group of the student. The system will not allow taking courses unless semester is enrolled. The module updates information to the financial department via a web service that courses have been chosen, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The courses selection is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Administration of student exam applications

In a limited predefined time interval before the exam term the students can access the module and enlist for exams. The list of exams is limited to the courses taken by the student in the current and previous semester. Officers can change exam applications in special cases.

If the student is not satisfied with the resulting grade for a course, it is possible to reverse and the test can be re-taken. For cancellation of an already passed exam, the student must submit an online application for annulment of the examination through the system. The system will not allow taking exams or partial exams unless a semester is enrolled. The system stores data for both partial and complete exams. The module updates information to the financial department via a web service that exams have been applied, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The exam application is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Administration of student's exams

After the exam has passed, the results from the exams, the grade the student has received can be updated through this module by the teacher of the course in question, or by the officer in student services for all exams. A module for bulk insert of grades for all students per exam per exam term will also be implemented. This module is also accessible by officers or by the teachers covering the exam in question. Except for entering grades in the list of students applied for the exam presented, an option for uploading a predefined Excel list template of students and grades should also be implemented.

Administration of seminar thesis

When seminar theses are due for some courses, the student can apply for a thesis in this module. The notified selected teacher can confirm the thesis and after the student submits the final version, the teacher can update the grades and the status of the thesis. Officers in the student services can also perform the same tasks if necessary. The module updates information to the financial department via a web service that a seminar thesis has been applied for, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The seminar thesis application is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Administration of diploma thesis

The student can apply for a diploma thesis in this module. The notified selected teacher can confirm the thesis and after the student submits the final version, the teacher can update the grades and the status of the diploma thesis. Officers in the student services can also perform the same tasks if necessary. Before the diploma thesis is completed and the student finishes his/hers studies, the system retrieves information from the library module for the books borrowed by the student. The student must not have any books borrowed at the time of diploma thesis presentation. The module updates information to the financial department via a web service that a diploma thesis has been applied for, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The diploma thesis application is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Administration of exam terms

Regular and irregular exam terms can be scheduled along with particular exams schedules in this module. Officers at the student services in coordination with the vice-deans should update data in the module.

Administration of semesters

Each new semester could be created as a copy of the previous fall/spring semester. However certain changes are always possible (activation/deactivation of a course, change of the teacher for a course). This module serves to create new semesters with appropriate list of activated courses eligible for taking. It allows joining courses with professors for the semester in question. Copying previous semesters makes for quicker work, but changes should also be easy to make.

Administration of quotas and tuition prices

This module serves for inserting and updating tuition details for various quotas that students can enroll in. Costs for different study programs and quotas can vary from year to year, so new values for each year can be updated. The student pays the costs effective the year he/she started studying.

Administration of courses in programs

Each course can be mandatory or elective in different study programs. Also, each course can be taken in different year of study in different programs, and have different prerequisites. The relations between programs and courses that can be taken in the program in question are managed in this module.

Administration of courses

This module serves for inserting or update details for each course that is taught in the faculty / university. Parameters like ECTS credits, fall/spring semester for the course, number of classes per week can be updated here. Officers update data, as well as teachers for the particular course.

Administration of programs

Sometimes new study programs are introduced, so they are managed through this module. Parameters like name, start year, bachelor/master/PhD level, revision of the program, etc are updated.

Administration of programs revisions

Manage revisions of study programs. Each program belongs to a revision in the programs.

Administration of members of the faculty

The module serves for management of the employees (members of faculty). New employees are inserted, and details for current employees / faculty members can be updated. Parameters like status, education can be updated.

Diploma thesis report

This module provides listings of diploma thesis by various filters (by professor, by course, by year, time interval, diploma thesis status etc). With this listing the teachers can overview their diploma thesis.

Exams report

This module provides listings of exam applications by students by various filters (by professor, by course, by year, time interval etc). With this listing the teachers can overview the students have applied for an exam.

Graduated report

This module provides listings of graduated students by various filters.

Elective courses report

This module provides listings of students by various filters (by professor, by course). With this listing the teachers can overview the students have applied to follow his/hers course.

Master book

This module provides printing the master book of enrolled students requested by the ministry of education. The system should generate and update main registers for each faculty. Types of main registers for each faculty are the following:

- Registry of enrolled students at the first cycle of studies
- Registry of enrolled students of second cycle studies
- Registry of enrolled students at the third cycle
- Registry for graduates of first cycle studies
- Registry for graduates of second cycle studies
- Registry for graduates of the third cycle

List students per program

This module provides listings of students by various filters, mainly the study program they have enrolled in.

Seminar thesis report

This module provides listings of seminar thesis by various filters (by professor, by course, by year, time interval, seminar thesis status etc). With this listing the teachers can overview their seminar thesis.

Administration of student mobility

The system should allow easy updating of data for the student who wants to be switched from one study program to another, in the frame of unit at the University. The student should apply for the implementation of mobility and appropriate documentation to transfer from one to another study program. This means that the student can complete the form on the Internet and can immediately print. The document stands and ID. If the student needs to pay, is paid to the respective ID. The application must be submitted in the office of Student Affairs. Approval of the request or notice due to incomplete documentation or failure must be transferred via e-mail notification.

Student logs through the Web site of the faculty. Also, there is and other options of submitting an application for transfer from one to another study program.

Equivalence of subject - will be used for migration from in the faculty from one to another program. In the future it would be good to can afford this on the university level.

Equivalence of subjects covers which subject of which organizational unit or study year covers wholly or partially any other subject. This equivalence provides students transferring from one program to another study program in the same or other faculty in the organizational units of the faculty.

For students who are enrolled from other faculties, the equivalence should be created with recognition of subjects or the appropriate number of credits from some subject of a specific form in which all subjects are shown for the respective study program.

The other subjects that will not find equivalence are treated as optional.

All documentation for the student mobility is an integral part of the student's file.

According to the Law on Higher Education is made possible mobility of students from one unit of a University or another higher education institution in the country and abroad.

The officer of Administrative office shall publish international exchange programs for students, or on the basis of bilateral agreements between universities. There is a need of opportunity to announce and compatibly between academic programs and courses.

The system should allow easy updating of data for the student who wants and is transferred from one to another higher education institution in the country and abroad. The student should submit a request and appropriate documentation to transfer from one to another higher education institution.

Administration of Master thesis

The student can apply for a master thesis in this module. The notified selected teacher can confirm the thesis and after the student submits the final version, the teacher can update the status of the master thesis. Other authorities (faculty secretary) must confirm the status update after completion. Officers in the student services can also perform the same tasks if necessary. Before the master thesis is completed and the student finishes his/hers studies, the system retrieves information from the library module for the books borrowed by the student. The student must not have any books borrowed at the time of master thesis presentation. The module updates information to the financial department via a web service that a master thesis has been applied for, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The master thesis application is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Issuing documents

In the course of study, students need documents issued to confirm their status. The system will automatically provide the issue on the basis of data on students entered in the system. The form and content of the certificates should to comply with the prescribed form. Templates for the certificates should be integrated in the system. The templates of the documents should be changeable by system administrators without interventions in the source code by the supplier of the software. The student can login and request one of the certificates online, for the certificate to be prepared in advance. The student can later pick up the certificate from the student services department, or have it delivered by mail. All services are paid for by the student. The system should print the certificates for passed exams and graduate students under the templates provided by the Ministry of Education and given in the official Gazette. 35/08, 103/08 and 26/09. Templates should be editable. The module updates information to the financial department via a web service that the student has applied for a certificate, and money should be deduced from the student's account. The web service response from the Finance department should confirm the activity. The document is issued only if the system receives a web service notification from the financial module that financial commitments have been completed.

Certificate for passed exams

In the course of study, students need this kind of certificates to compete for scholarships, credits, student homes and so on. The certificate lists all the exams passed by the student, and some small amount of personal details for the student, among other things. The detailed template must be changeable and customizable per faculty.

Certificate for completion of studies

After passing all the exams, and the presentation of diploma thesis (if applicable), at the request of the student, the Office of Student Affairs issues transcripts and certificate that the student graduated. The certificate lists all the exams passed by the student, and some small amount of personal details for the student, among other things. The detailed template must be changeable and customizable per faculty.

Sign out document

For the student who wants to transfer to another educational institution, at his/hers request, a sign out document is issued. The certificate lists all the exams passed by the student, and some small amount of personal details for the student, among other things. The detailed template must be changeable and customizable per faculty.

Issuing news and announcements

This module serves for sending notifications, news and announcements to students and members of faculty. The authorized user can select a subset of students and/or members of faculty, write a message and send it. Each authorized user can only select a subset according to his/hers authorization (teachers can send messages to students taking their courses, vice

deans to students at their faculty etc). The filters for subsets of students are various (by course, by start year, by study program, by name, by schedule group etc.). The messages will primarily be e-mails, but an SMS module should also be developed.

The system should also support notifications and alerts defined for various events that are automatically triggered and sent.

Administration of financial services

This module serves for defining new and updating old services that imply financial consequences for the students. The financial services can have fixed cost for all students or can be different for different study programs or for different start years of study. Also costs can be different for different quotas for students. The module servers to calculate the amount the student should pay when requesting a service from the university. Also the module should retrieve data from a web service from the financial department to check whether the student has paid the wanted amount. Only if the student has completed the financial commitments expected, this module can allow for the service to be provided for the student. Examples of services include, but are not limited to:

- issuing certificates,
- fixed fee for enrolment in a semester,
- applying for a diploma thesis,
- applying for a master thesis,
- applying for a PhD thesis,
- applying for an exam outside the predefined time period,
- applying for a seminar thesis,
- library membership,
- Tuition
- Costs per year
- Uniforms
- Kimono
- Seminars
- Project assignments
- Colloquia (with penalty or regular)
- Exams (In regular term or In penalty term)
- Portable computer
- Differential exams
- Certificates
- Specialist labor etc.

The system should allow definition of new services and related services. Newly created financial services can be associated as additional fees to various predefined functionalities in the system (semester enrolment, courses selection, exam application etc). The system should allow the definition of models of payments (packages). The system should enable the allocation of services and models of paying to students. The system must allow the granting of defined services for a student. The system must send information to the financial system for each service performed for one student. The system must receive information from the financial system of any change of the balance of a student.

When the system is using a standard post-paid way of managing finances, whenever a student demands a service, the appropriate fee is calculated according to his/hers parameters and the pricelist defined. The amount is showed to the student along with a unique code for the payment that he/she should fill in when paying in the bank (online or directly). When the information for the bank payment and the code of payment is received in the financial module with the bank report, the code of the payment is matched to the code of the pending service requested by the student. Since the pending service's code is matched to the code of a real bank payment, the service status is then transformed from "pending" to "executed" (if the software can complete it) or status "allowed to be executed" (if a person needs to execute it).

In the pre paid mode, the student can deposit certain amount of money via a bank transaction to the financial module. Whenever a service is requested by the student, the fee is calculated

along with a payment code. The amount is then deduced from the stored money of the student, marked with the unique code of the particular service for the particular student. The service status is then quickly transformed to “executed” or “allowed to be executed”, not waiting for additional payments or a bank report.

In both pre and post paid modes, web services are used to communicate among the core module and the financial services module. The core module calculates the amount that should be paid by the student and a unique code for the payment in question. This information is submitted to the financial module via a web service. In the prepaid mode a reply is instantly received with the web service if the amount requested is available in the student’s account, and the amount is deduced from the account. In the post paid mode, the financial service module triggers a web service in the core module sending information that a payment has been received with the particular code and amount. The code and amount are matched to an existing combination in the core module and the service is granted to the student.

Administration of lookup data

Master database tables containing cities, countries, high schools, municipalities, types of employment, types of education etc should be managed through this module. There should be forms for inserting new records and editing previous. Control for preventing insertion of existing records should be implemented.

Administration of class schedule

This module should enable inserting the already generated classes schedule in the system. The module serves as source of information for faculty members as to what classroom is available or taken, when in need of organizing additional meetings with students. Also students and teachers will get informed of their schedule during the semester through this module. There should be views of the schedule by class group, by classroom and by teacher.

Administration of faculties

The module serves for management of the faculties within the university. New faculties are inserted, and details for current faculties can be updated. Parameters like name, address etc.

Administration of sign outs

The module serves for the students to request for complete leaving of the university, withdrawal of all original documents (diplomas of previous education) etc. A document for the exams passed and status at the time of sign-out is issued. The student’s status is updated to “signed out”. Students cannot sign out if they have borrowed books from the library. This module requests confirmation via web services from the library module for any remaining books borrowed by the student. The module updates information to the financial department via a web service that the student has applied for a sign out, and money should be deduced from the student’s account. The web service response from the Finance department should confirm the activity. The sign out is completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Import of candidate data from enrolment module

This module imports candidates’ data from the enrolment module. When the student candidate is properly ranked and allowed to enroll in the university within the enrolment module, he/she comes to actually sign up. Some final documents can be submitted, payments completed and the office staff should retrieve the data for the student contained in the enrolment module and insert the data into the core students database. The office staff can search for the candidate using his unique identifier and migrate his/hers data into the student database. The system uses a web service provided by the enrolment module to retrieve the candidate’s data, and to confirm that the student has been enrolled in which case the enrolment module sets an appropriate marker for the candidate. The parameters transferred from the enrolment to the core module include all personal information in the Student profile data model and information for the initial faculty, study program, quota etc. The enrolment is

completed only if the system receives a web service notification from the financial module that financial commitments have been completed.

Other reports

The system will generate the following reports:

- Report for a list of candidates and students by high schools where they come from
- Report for a list of candidates and students by cities where they come from
- Report on the age structure of candidates and students
- Report for teachers who do not return the reports back to the Student Affairs
- Report for insured and uninsured students in a particular academic year
- Report on average passed exams in term, the faculty, university
- Report applicants and students by ethnicity
- Report the number of students with particular status
- Reports per time dimensions
- How much and how many passed the exam
- Report students who did not passed some exam
- Report on the number of students per grade received
- Report on the number of students who applied for exam
- Report the number of students who attend the exam previously applied by subject, by faculty, by generation
- Report the number of students who attend the colloquium previously applied by subject, by faculty, by generation
- Report for students with their average grade in some average range at certain moment

The system should calculate an average:

- By passed exams in term
- By subject, for certain academic year
- By faculty, by professor
- By subject, by professor
- By professor

The system should provide an export of all views in excel or pdf format, with feature for further editing.

The system should generate history for a student which will cover information about a student in given time period by the following categories:

Exam

- Subject
- Term
- Academic year
- Order number of exam
- Date of application for exam
- Date of attendance of exam
- Date of passing
- Date of cancelation
- Date of repetitions of an exam
- Professor

Colloquium

- Subject
- Order number of colloquium
- Semester
- Academic year
- Date of application for colloquium

- Date of attendance of colloquium
- Date of passing
- Professor

Semester

- Which semester
- Order number of semester
- Academic year
- Date of accomplishment
- Date of verification

Diploma

- Title of diploma
- Date of application
- Date of delivery
- Professor
- Commission members

Master thesis

- Title of the thesis
- Date of application
- Date of delivery
- Professor
- Commission

2.3 General Business Rules

This delivery aims to design a system that satisfies the following requirements

- Uses one centralised database to keep data on applicants and the enrolment procedure.
- Allows online access to all actors.
- Uses role-based authorisation.
- Implements customisable notification functionality to all interested parties e.g. exams schedule is published, change is made in class schedule, remainder to students to choose elective subjects etc.
- Supports multiple notification mechanisms: SMS, email, social networking sites.
- Implements web based application.
- Implements the “single point of data entry” principle.
- Administrator should be able to configure the system.

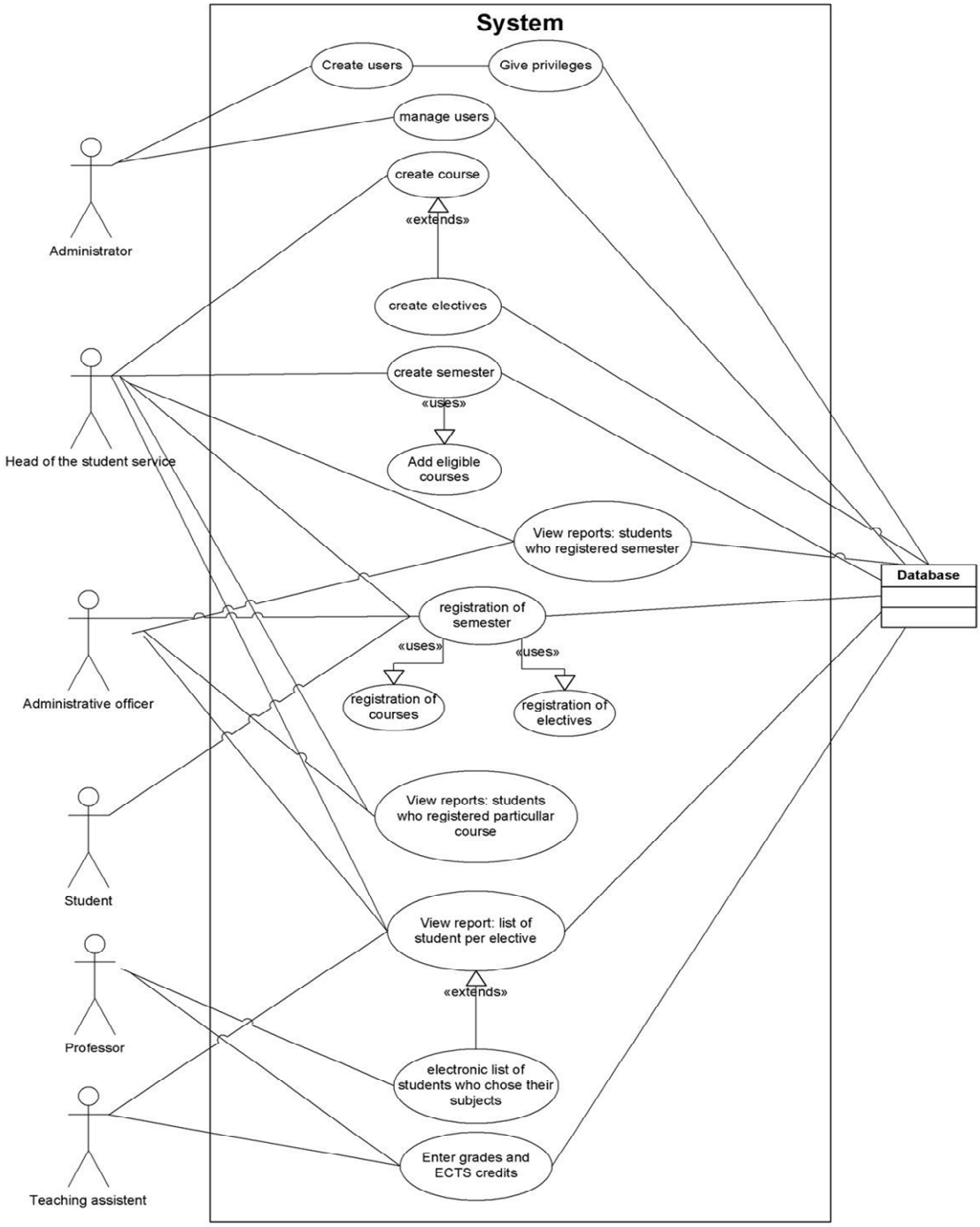
2.4 Data archiving

Data for graduated students and students who left the university is required to be made available to application users for reporting purposes. After a student ceases his student status at the University e.g. graduation or leaves the University, Head of the student service (HSS) is allowed and required to lock the data from the last enrolment term. Further changes to data are not allowed after the locking action.

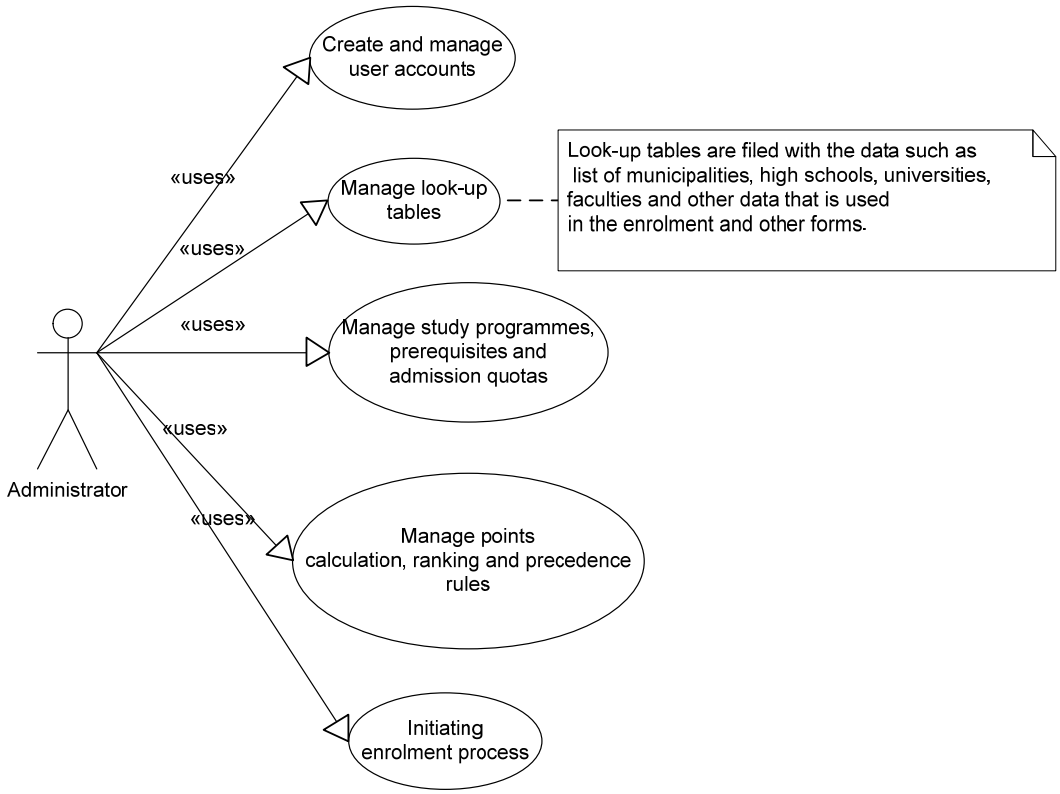
3 UML Diagrams

3.1 Use case diagrams

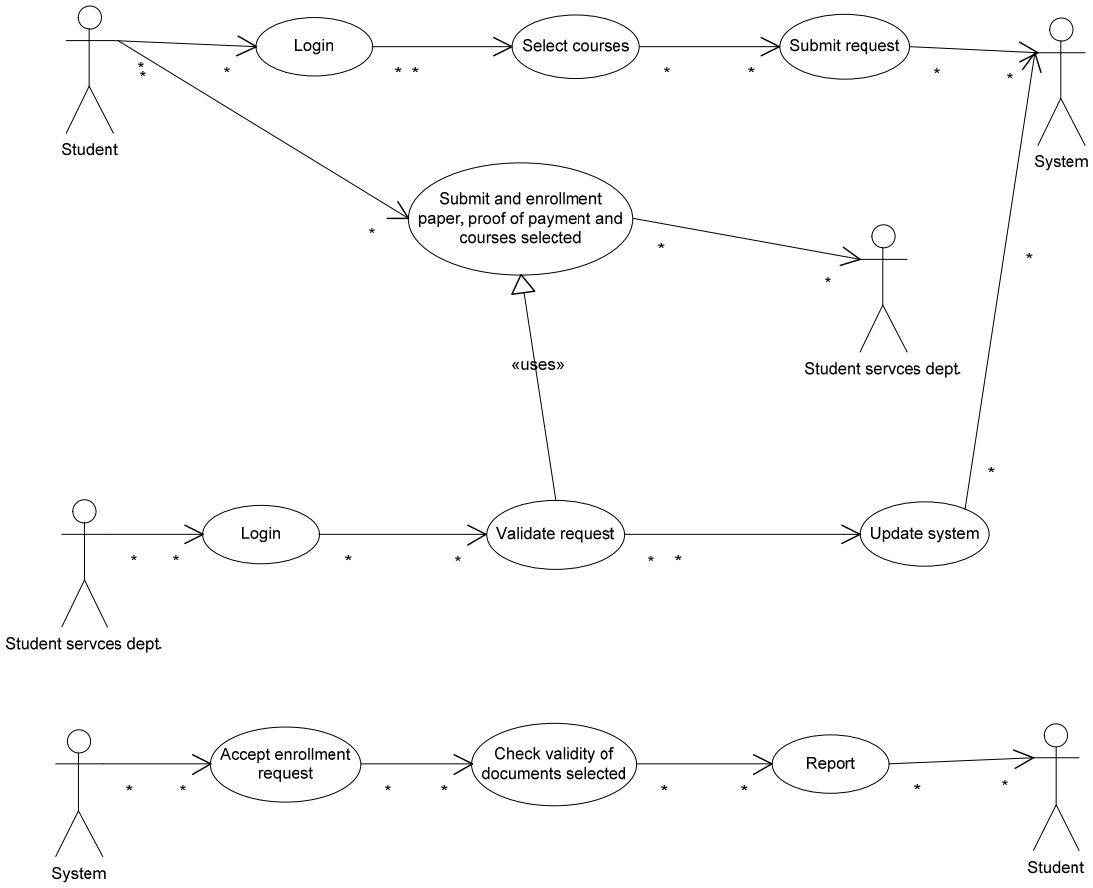
3.1.1 General use case diagram



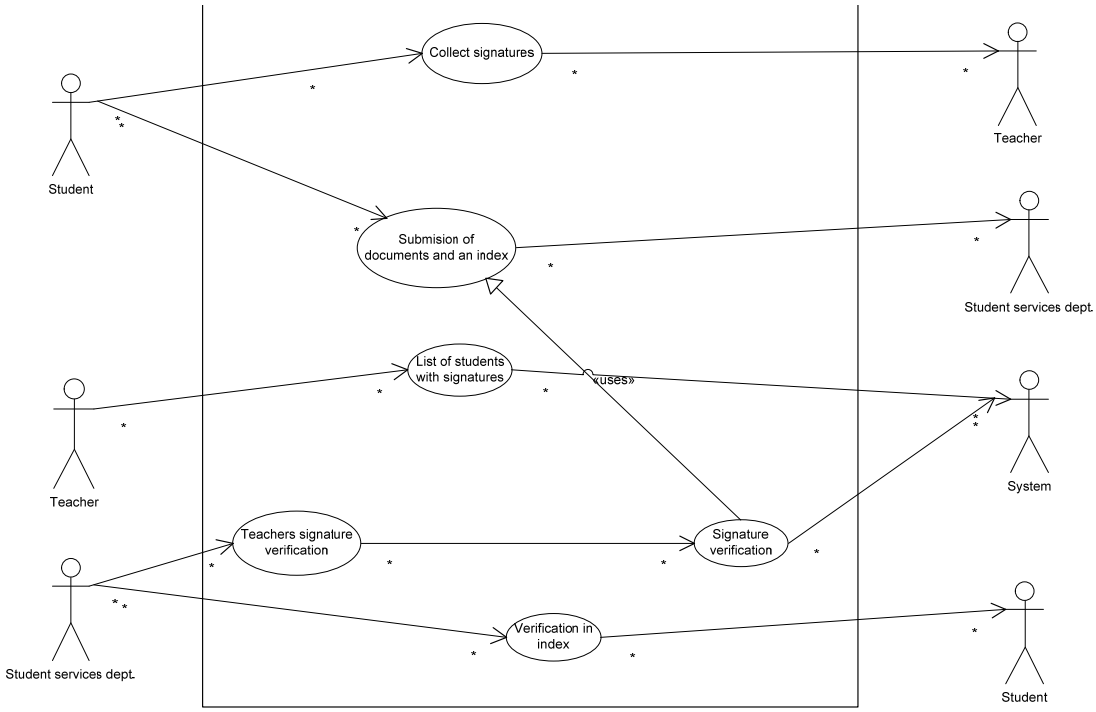
3.1.2 System administrator



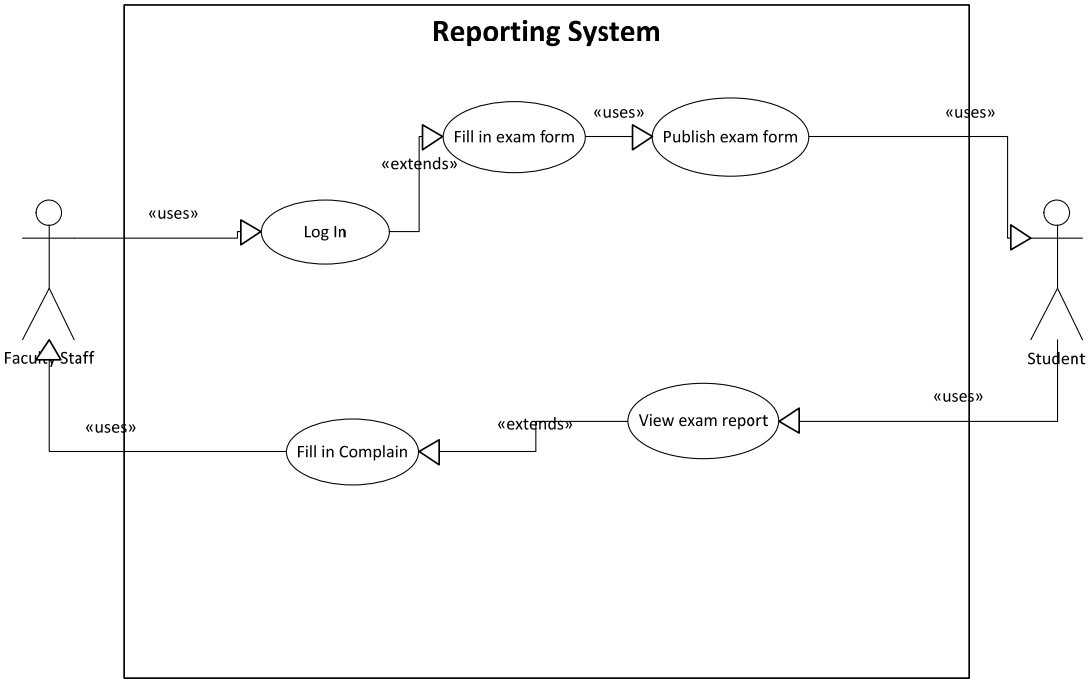
3.1.3 Enrolment in a semester



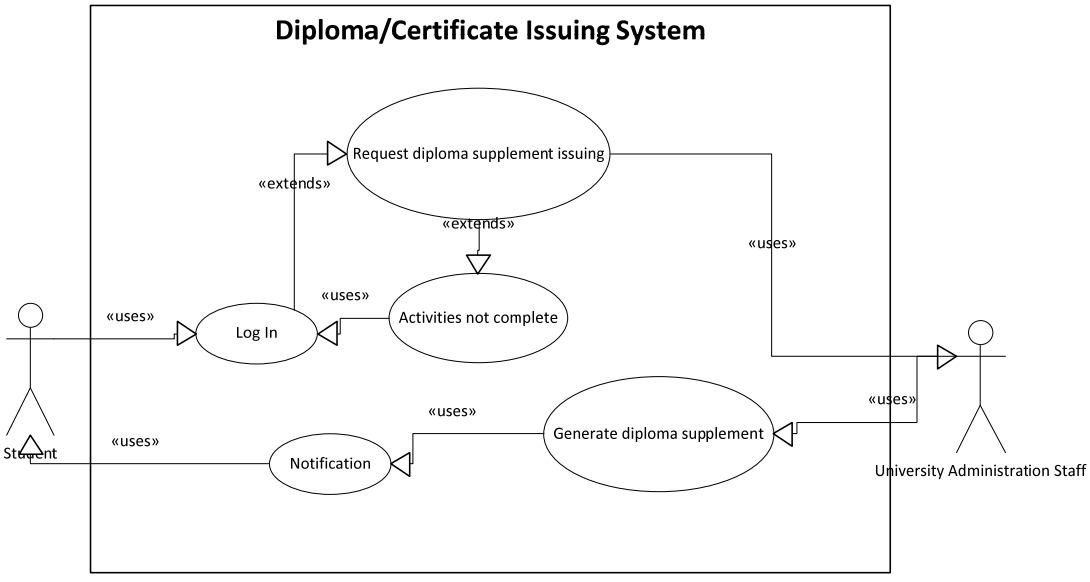
3.1.4 Certifying a finished semester



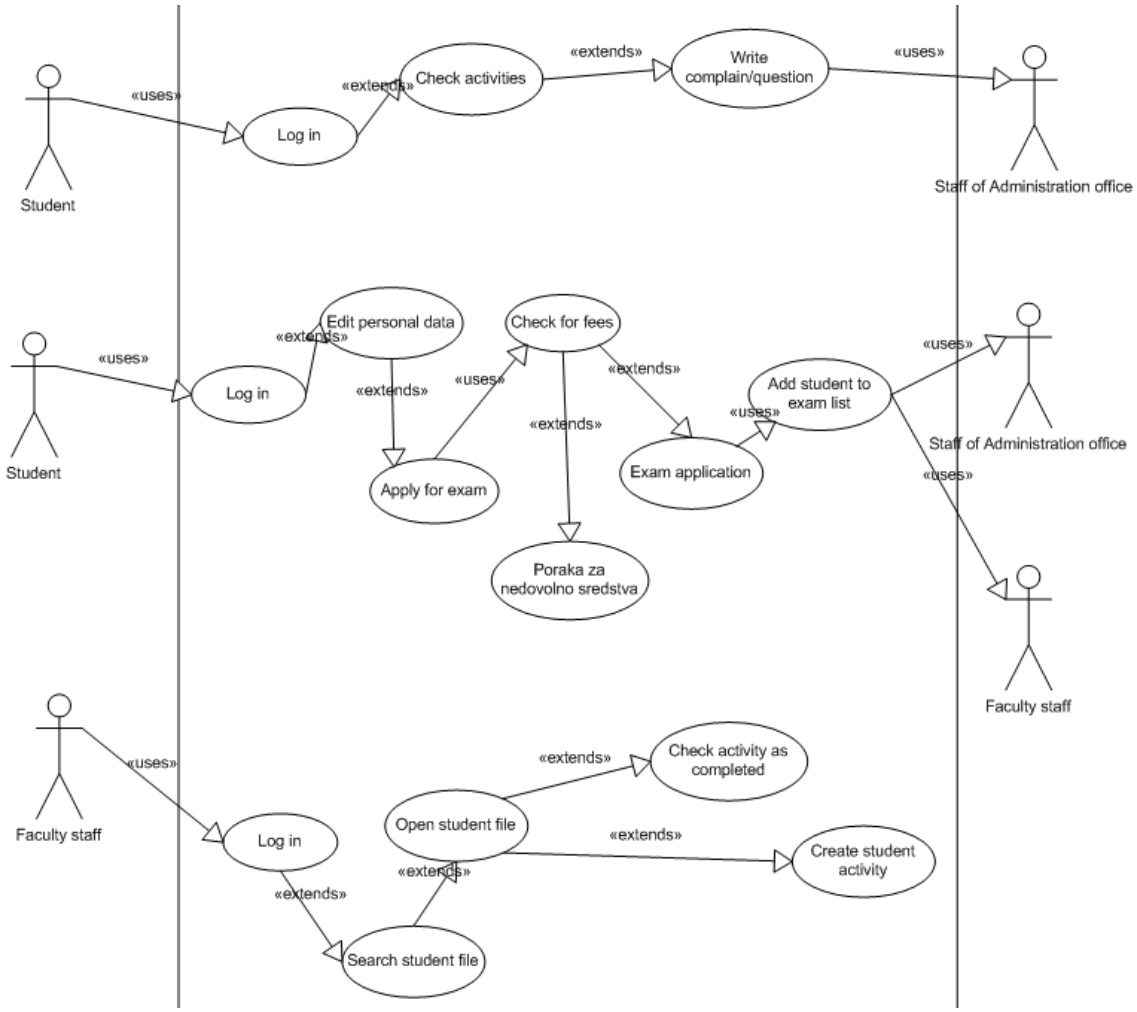
3.1.5 Exam report



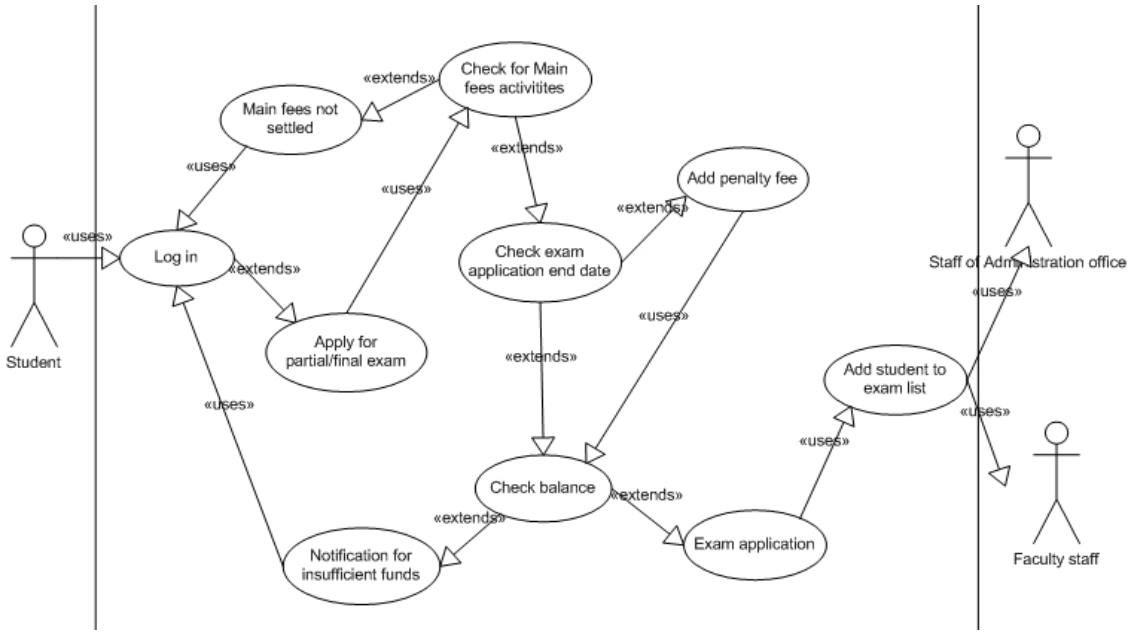
3.1.6 Diploma/certificate issuing



3.1.7 Overview of student file



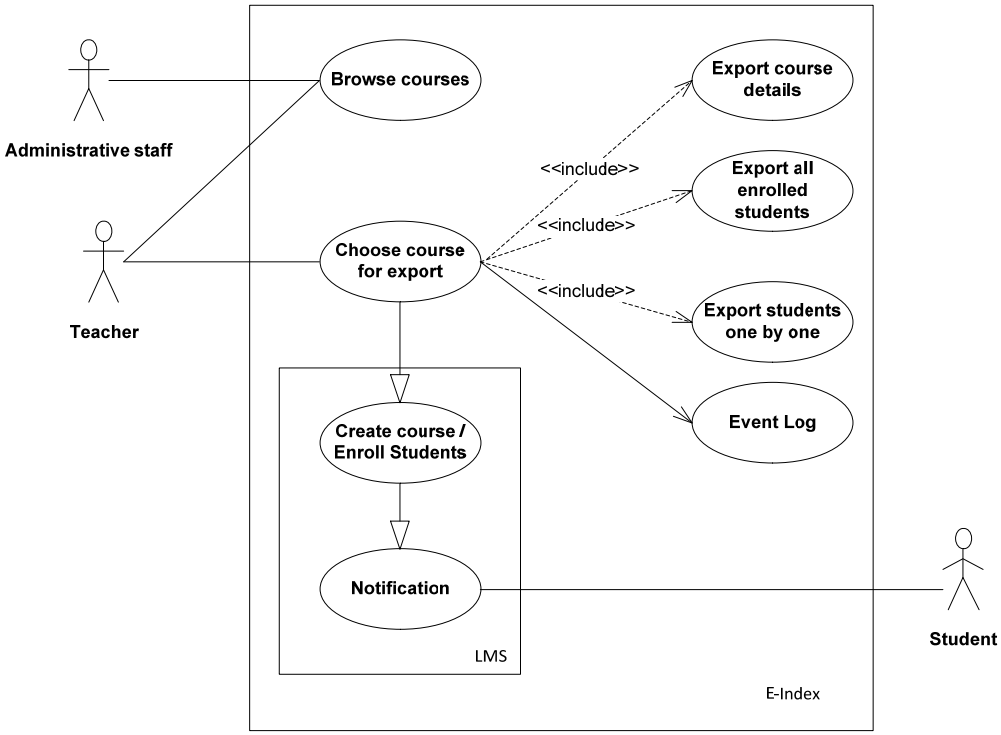
3.1.8 Apply for partial/final exam



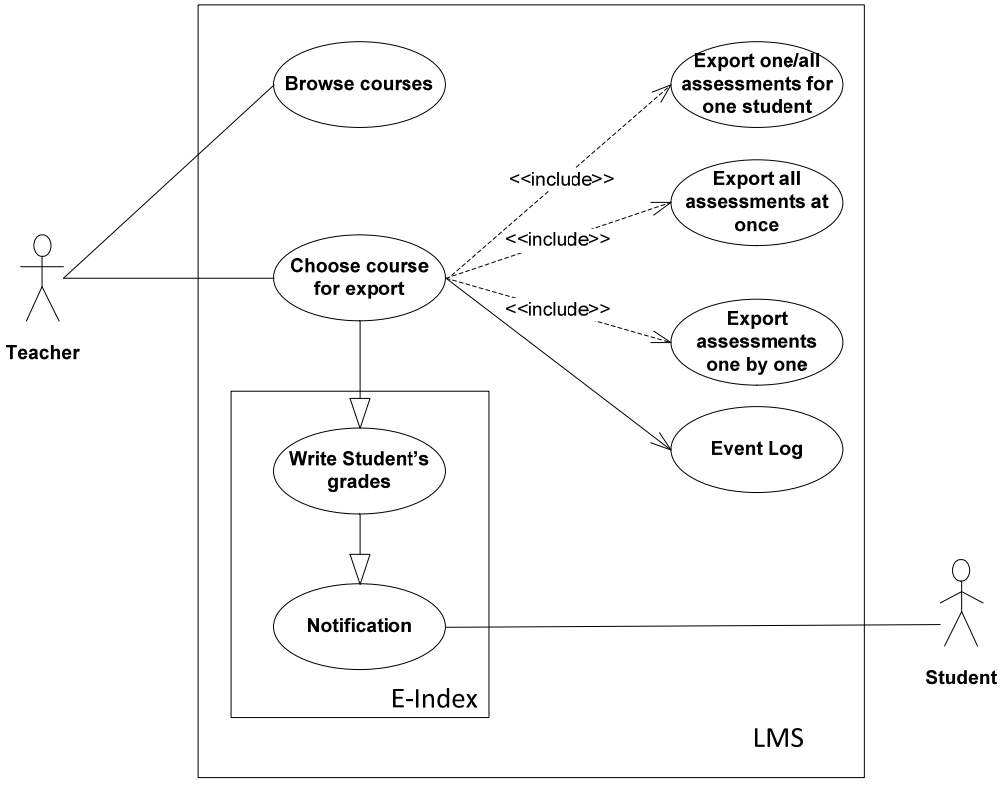
- The system checks if the student has settled the Main fees (Tuition, books, etc.).
- A notification is returned with the type of the fees not paid.
- If all the Main fees are covered the system checks whether the current date has passed the exam application date. If the date has passed, then a fine/penalty is added to the exam fee.
- System checks to see if there are available funds for the acquired payment. If the student has enough funds he/she is allowed to make an exam application and to be added to the exam applicants list.

3.1.9 Connection to LMS:

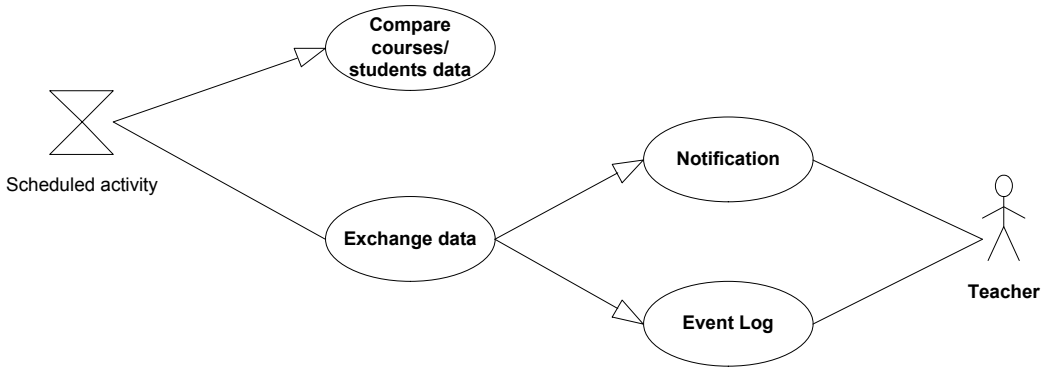
e-Index to LMS



LMS to e-Index

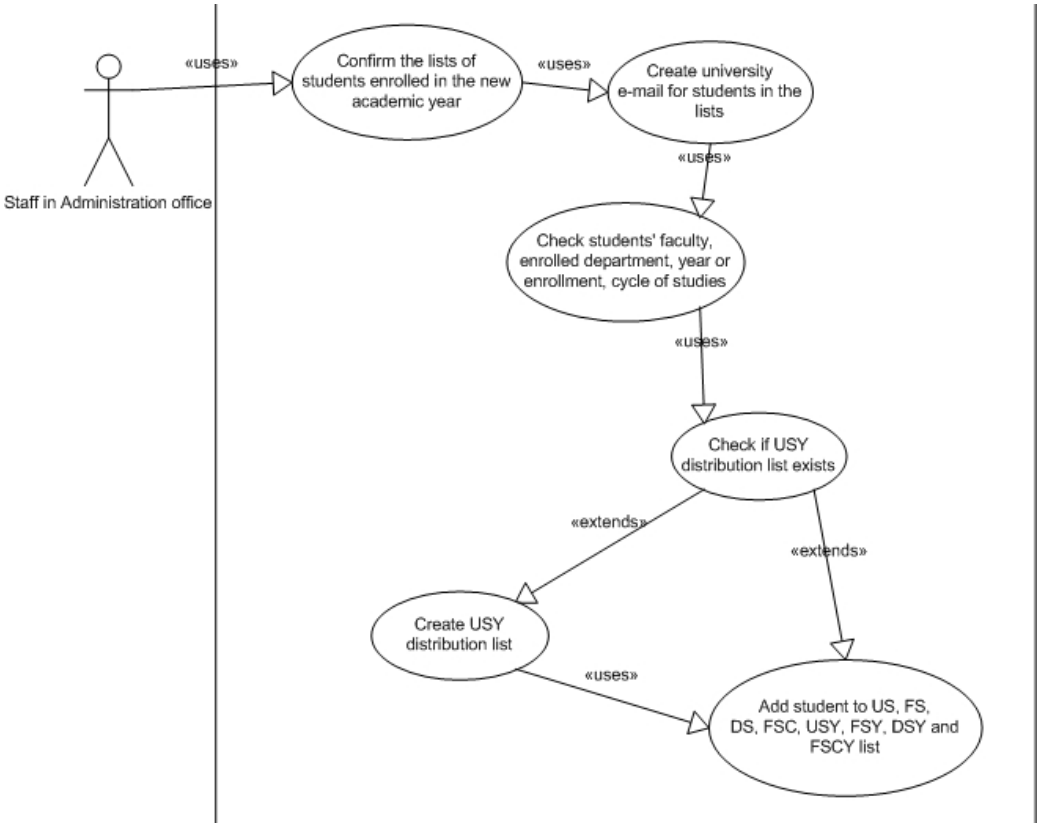


Synchronization e-Index and LMS



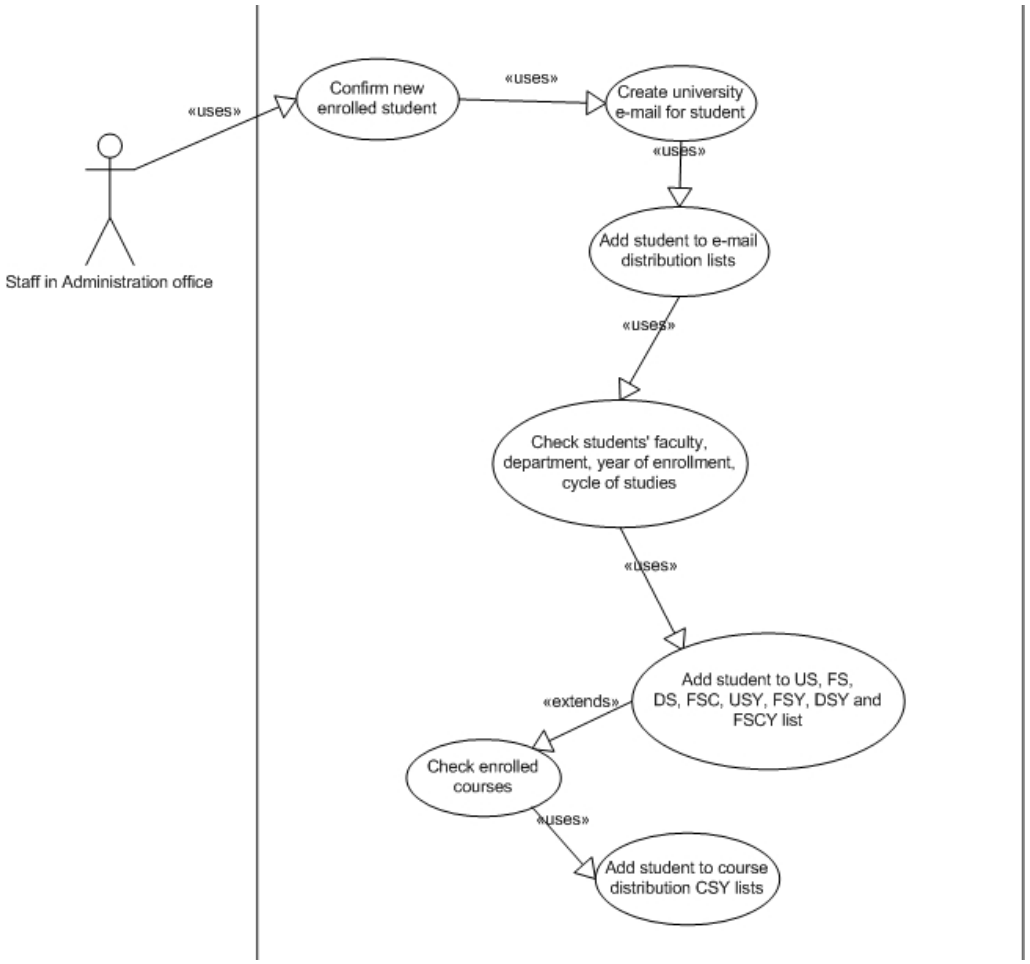
3.1.10 Connection to Email

Annual distribution list update



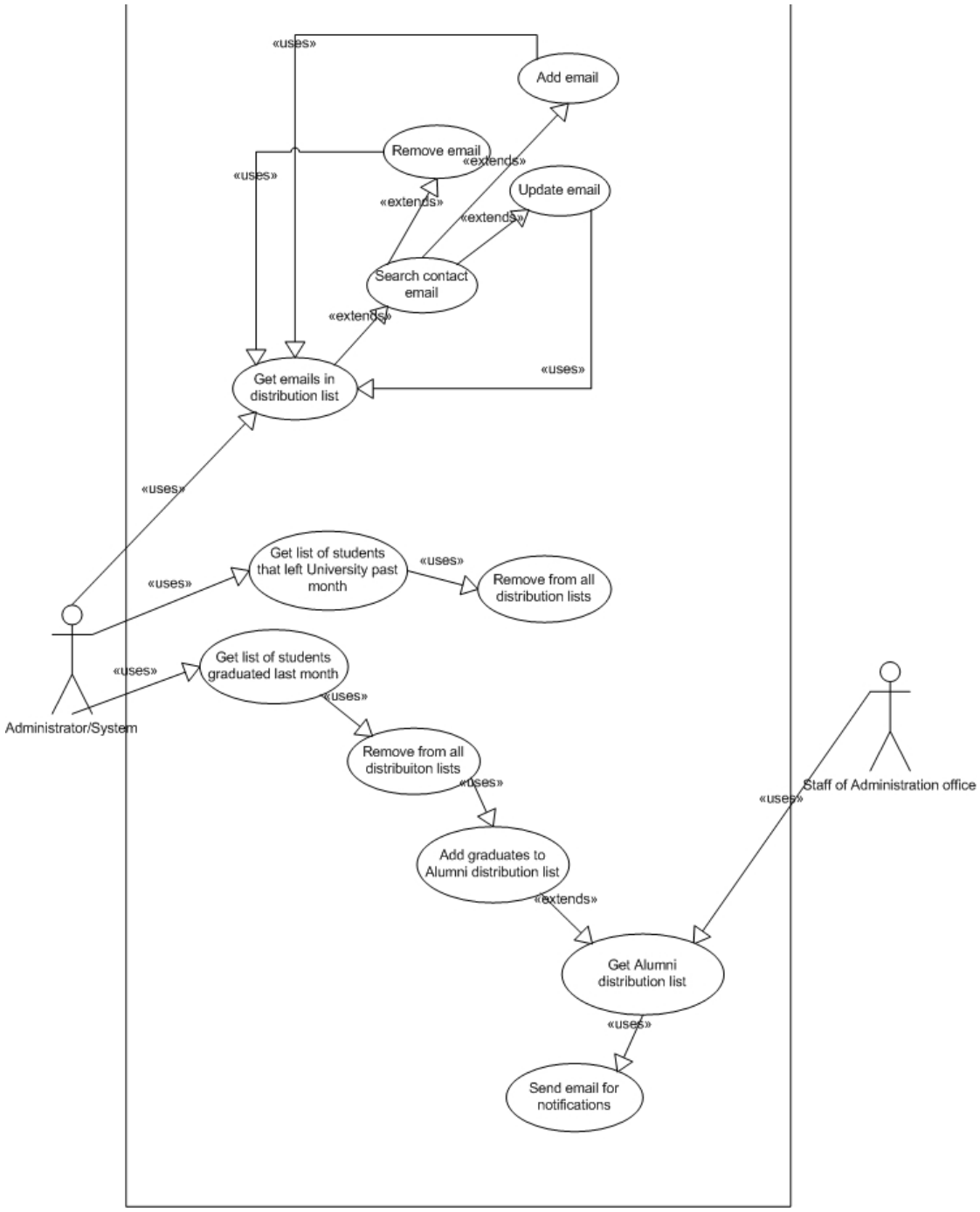
- Staff from the Administration Office confirms the list of students enrolled in the current academic year. The system uses this info to create each new student an individual University email.
- New students are to be added to distribution lists and the system checks student info. The system should create USY distribution list; List of students for a given school year. The system checks whether the list exists, if not it creates it.
- Students are added to all existing Distribution lists.

Update of distribution lists for late enrolment



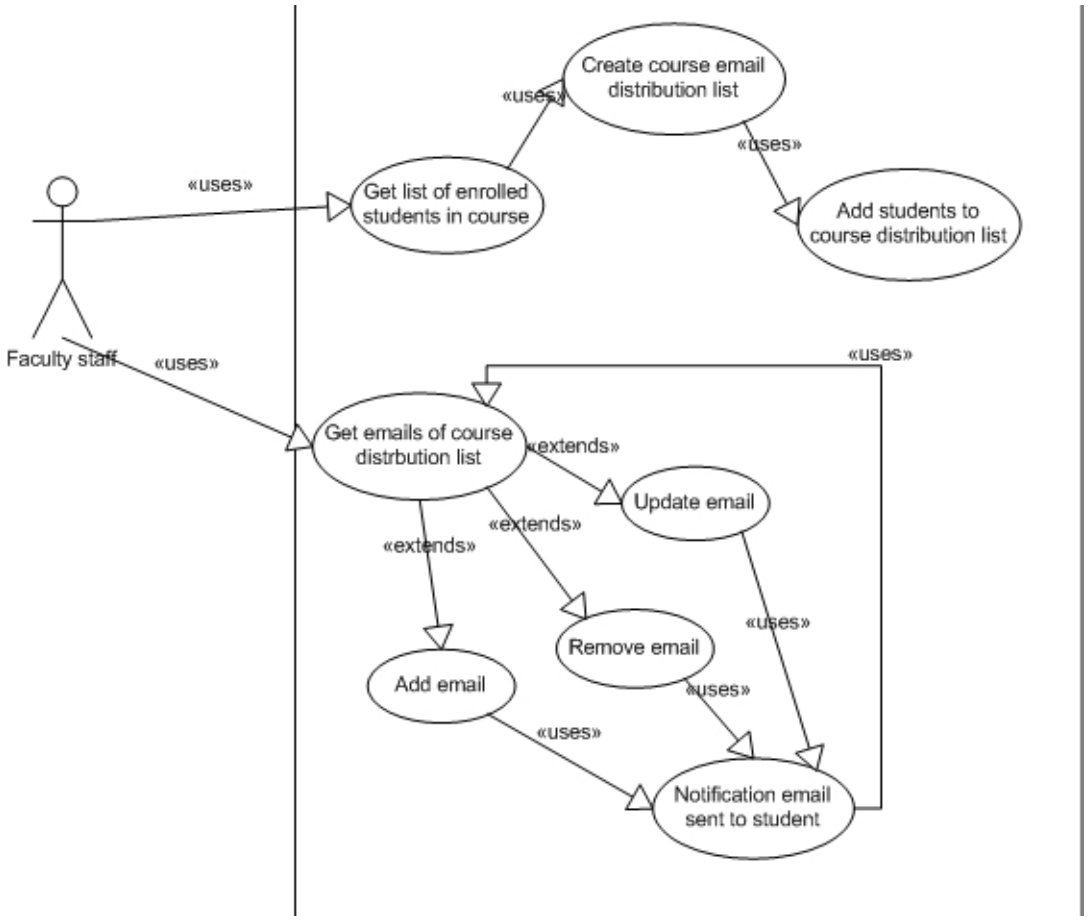
- Staff from the Administration Office confirms the list of students enrolled late (after deadline) in the current academic year.
- The system uses this info to create each new student an individual University email.
- Late admission students are added to all relevant distribution lists.
- Late admission students are added to their course distribution lists.

Administrator access to distribution lists



- Administrator should be able to manually add/update/remove emails from distribution lists
- Administrator should be able to remove students that have left the University/graduated from all distribution lists
- Administrator should be able to create/update the Alumni distribution list with graduated students. This distribution list should be used for university or individual notifications.

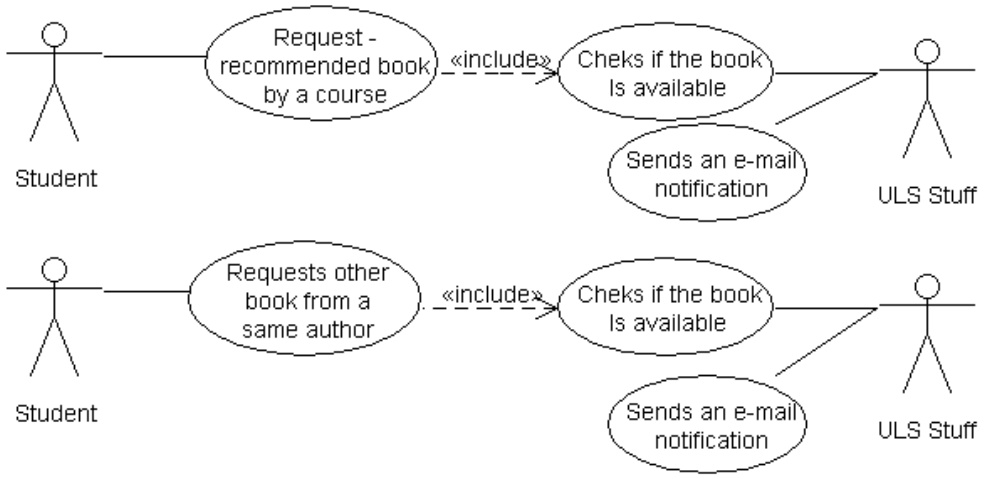
Course distribution lists



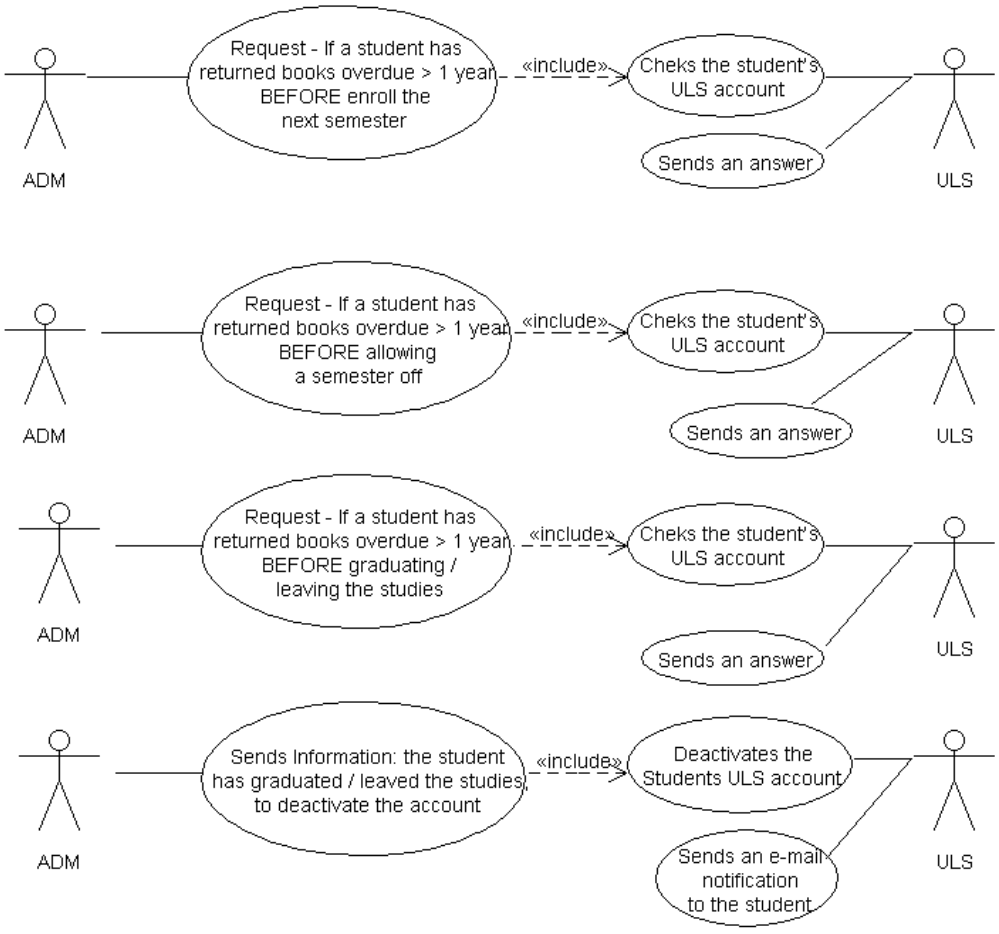
- Faculty staff should be able to create course distribution lists for the courses he/she teaches.
- Faculty staff should be able to create/update/remove individual emails from course list.

3.1.11 Connection to LMS and ULS (library)

LMS → ULS



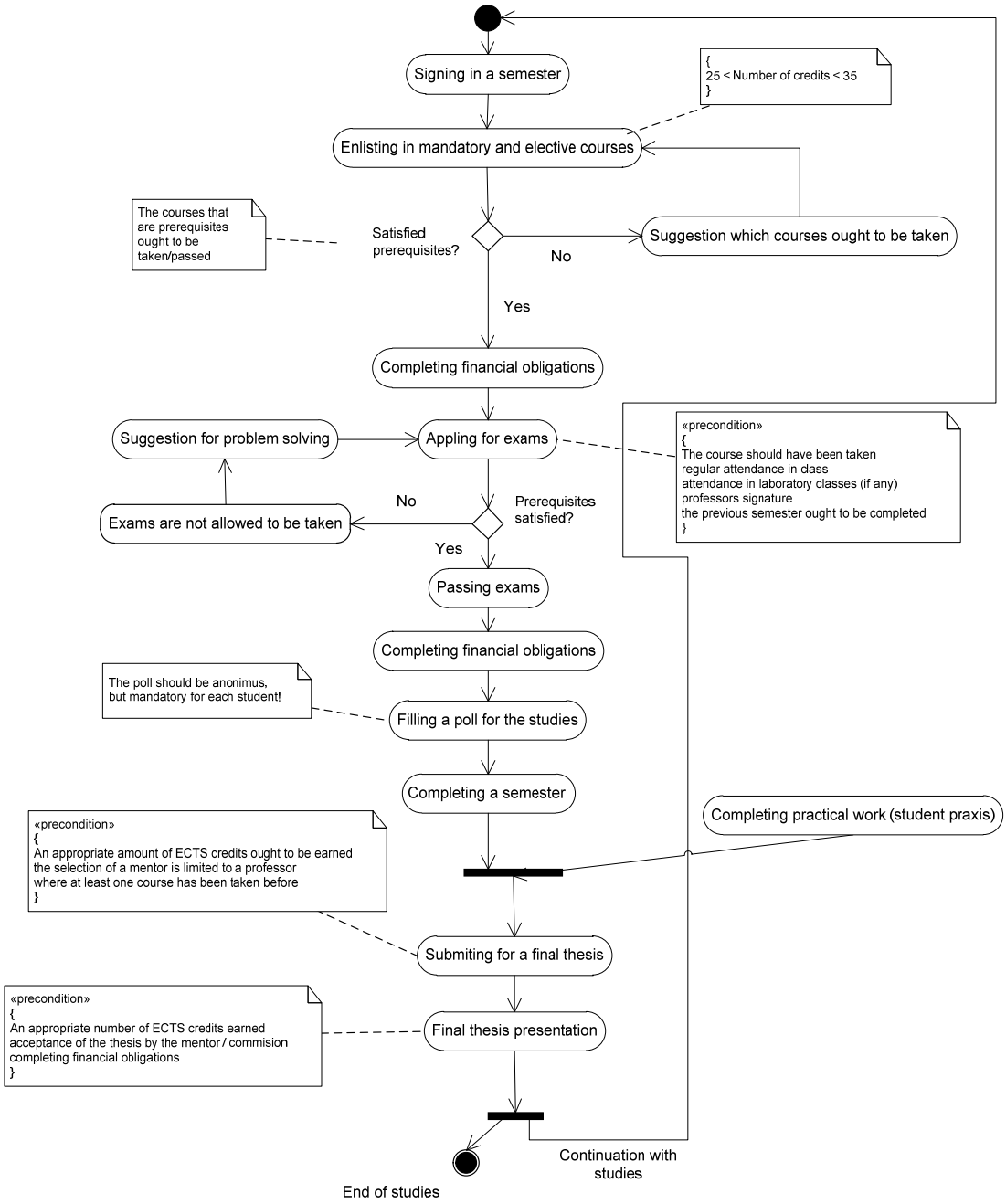
The CORE → ULS



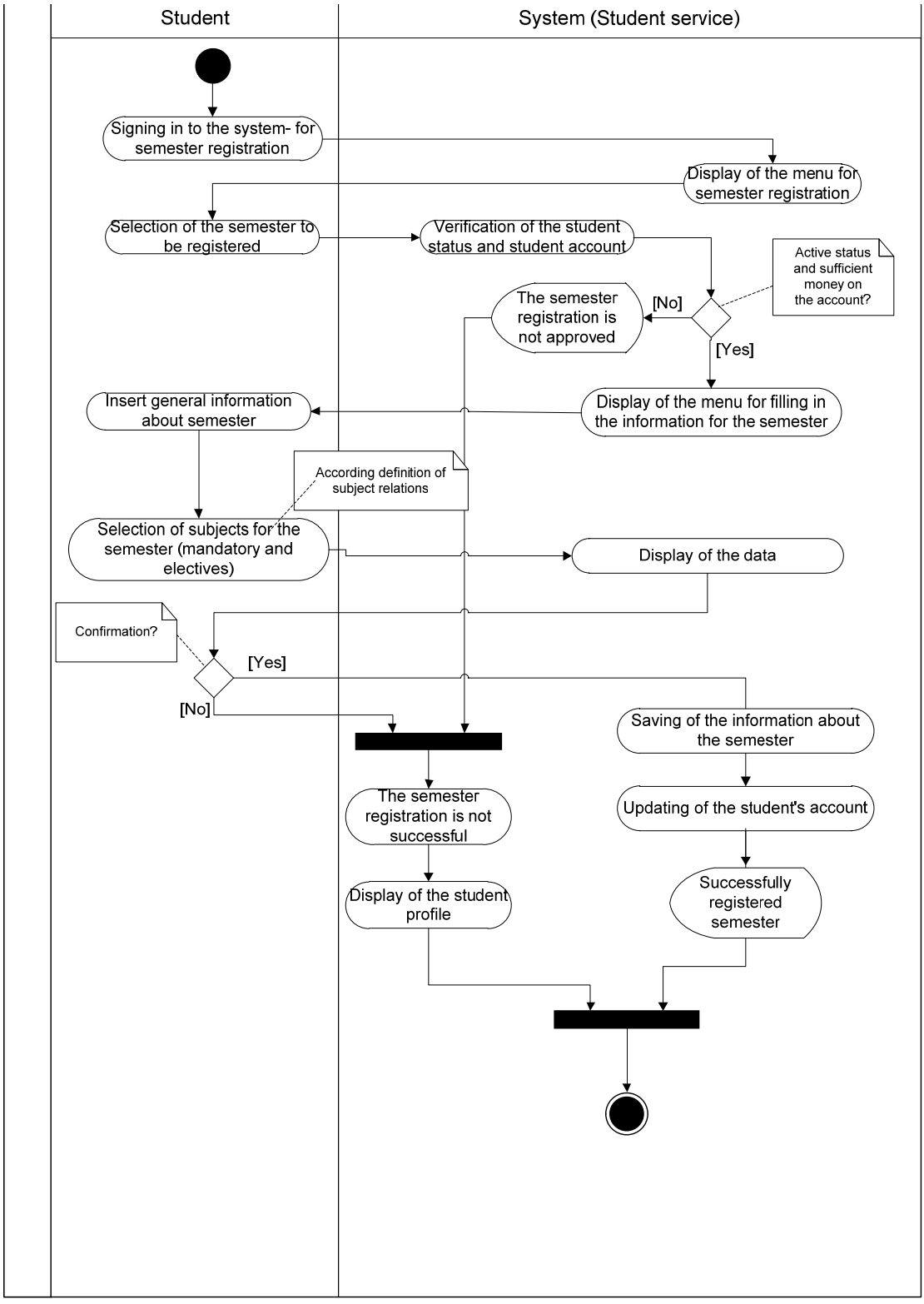
3.2 Activity diagrams

3.2.1 Process of studies

Following activity diagrams represents the business logic flow, and shows the activities and the events that make the process of studies.

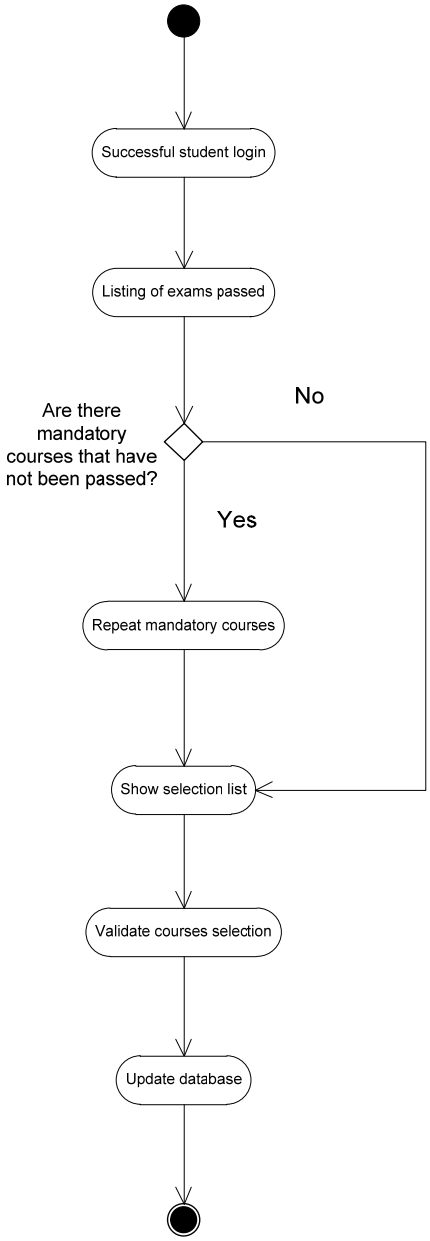


3.2.2 Semester enrolment – register a semester

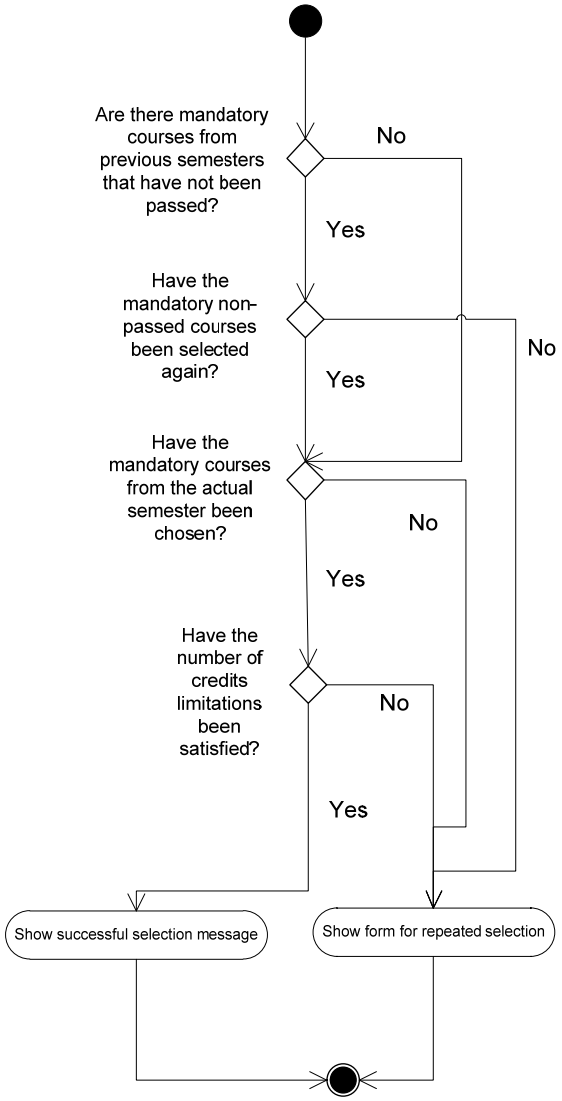


3.2.3 Choosing courses during semester enrolment (system)

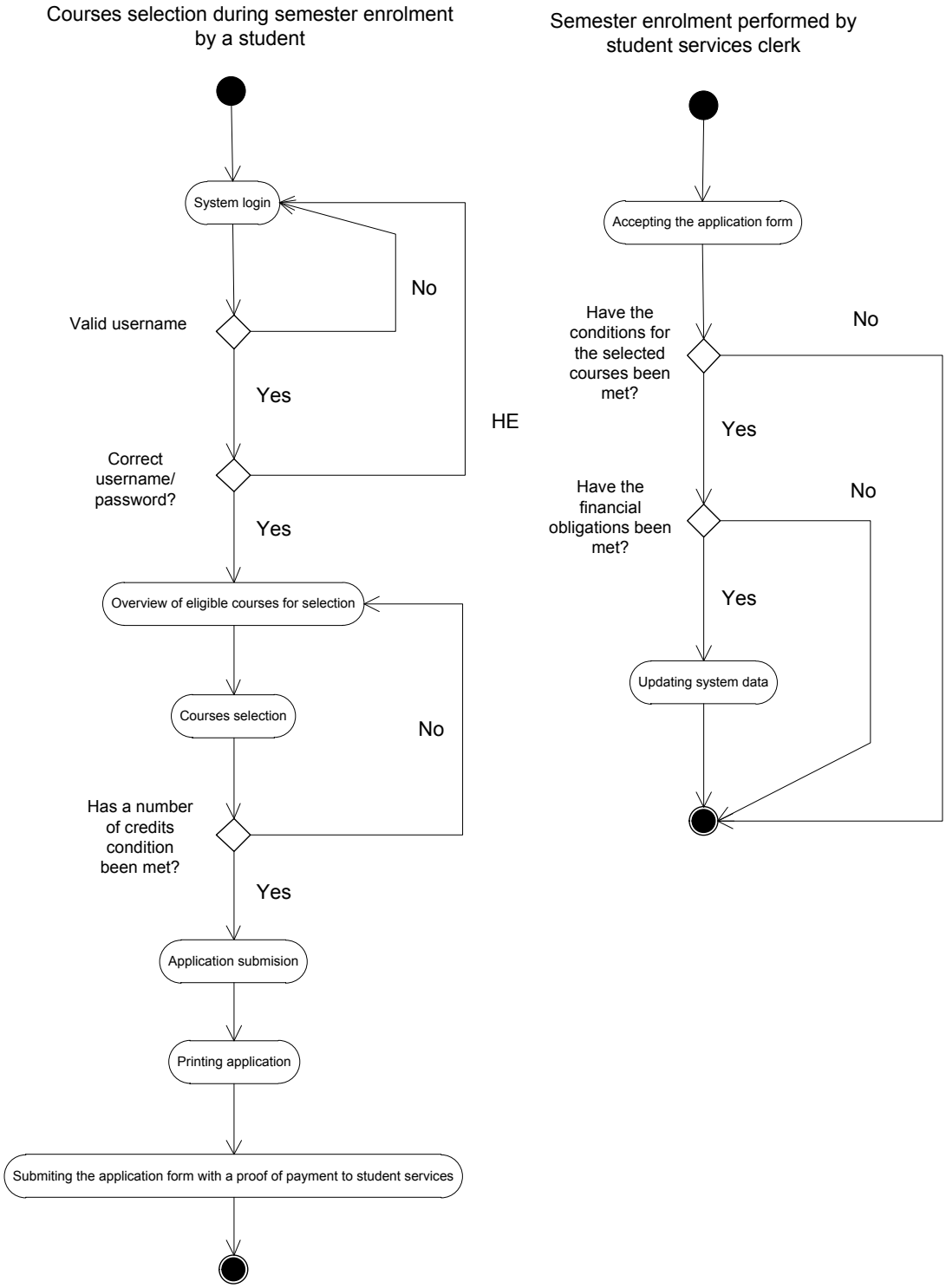
Semester enrolment from a system aspect



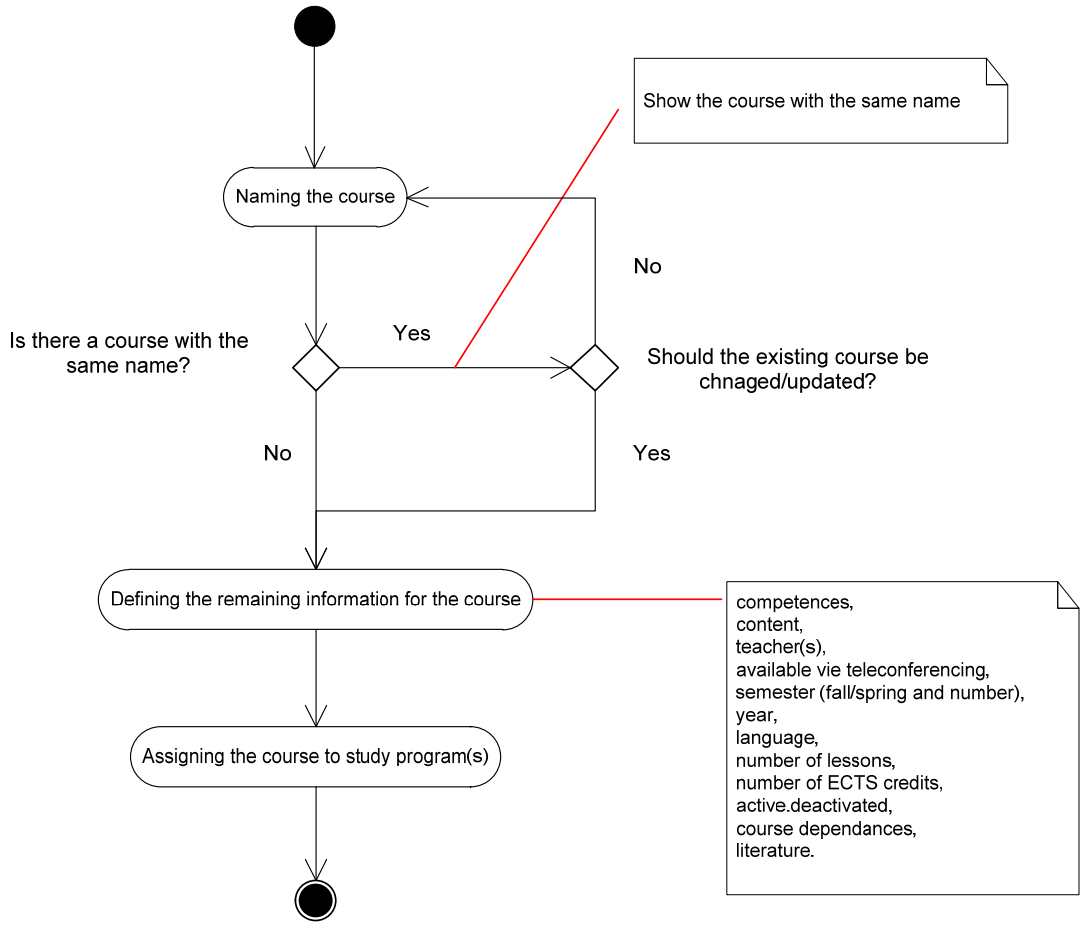
Detailed overview of system validation of the selected courses



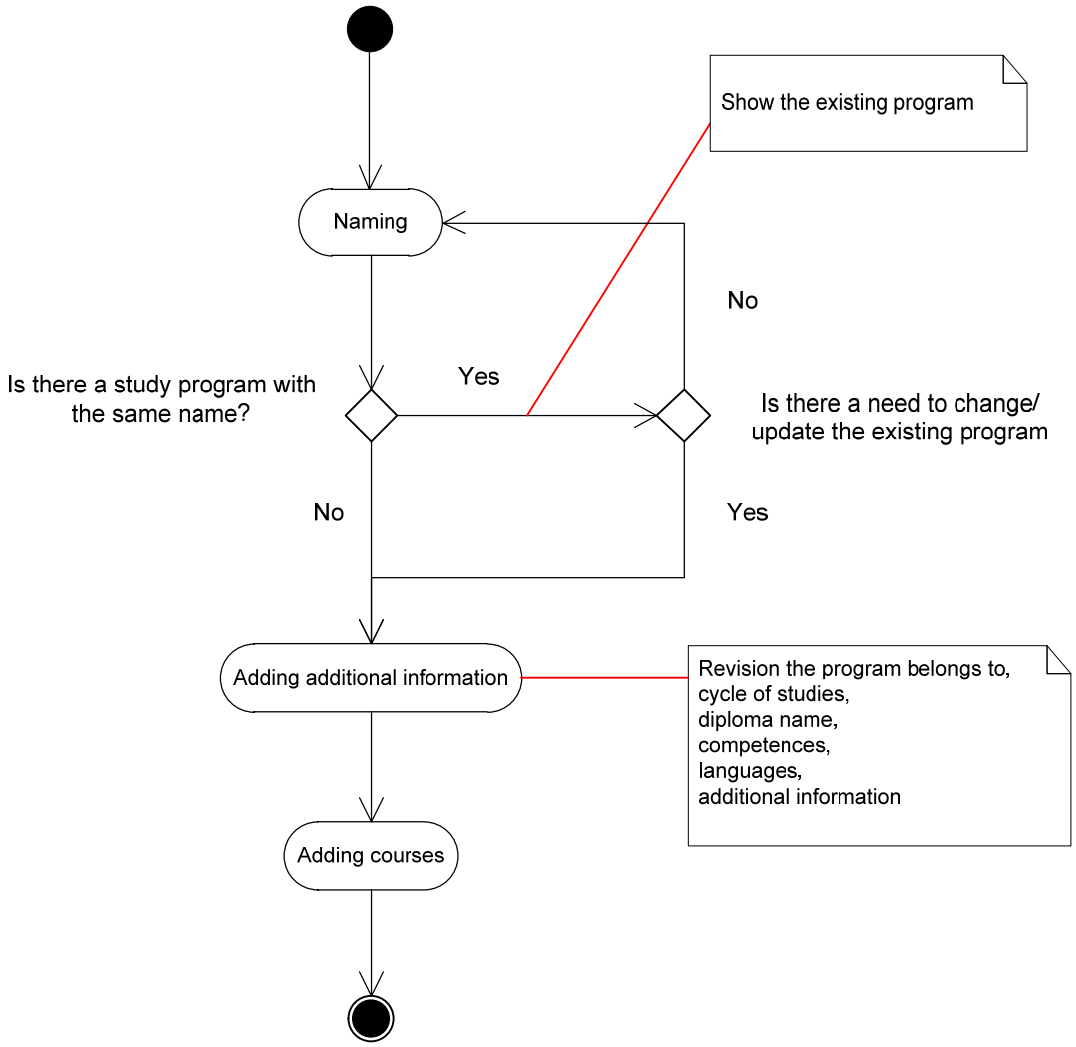
3.2.4 Choosing courses and enrolment in a semester



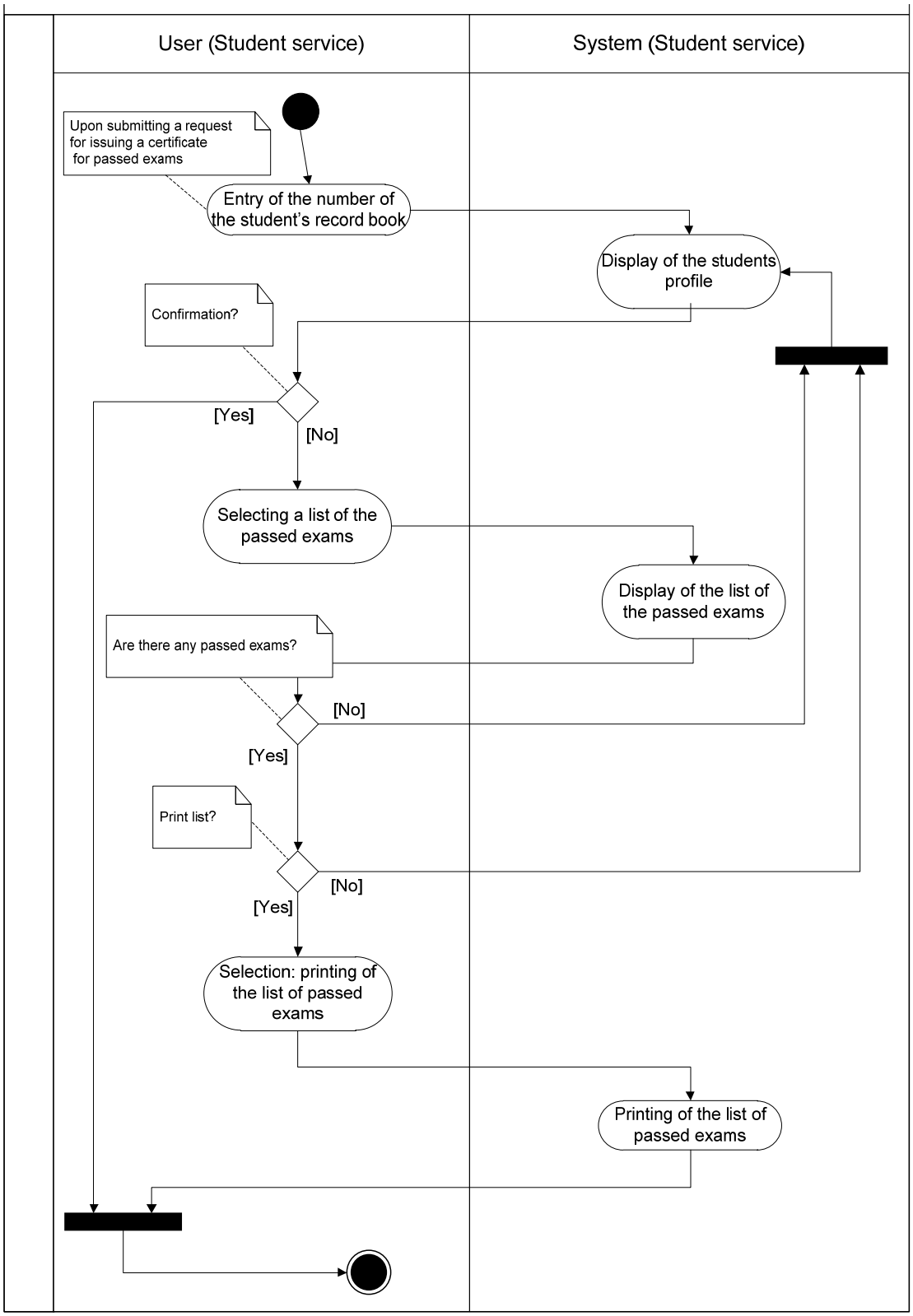
3.2.5 Creation of a course



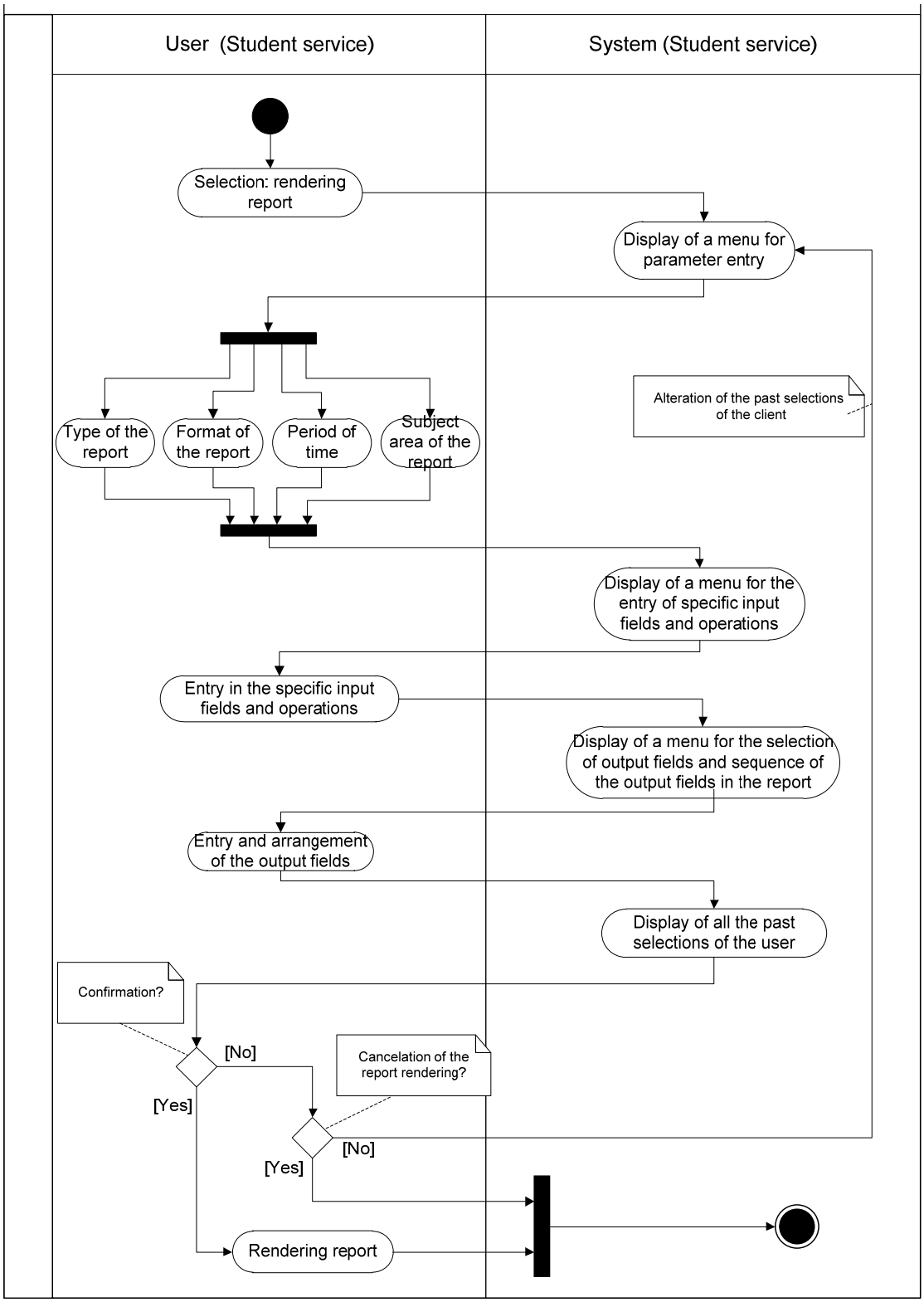
3.2.6 Creation of a study program



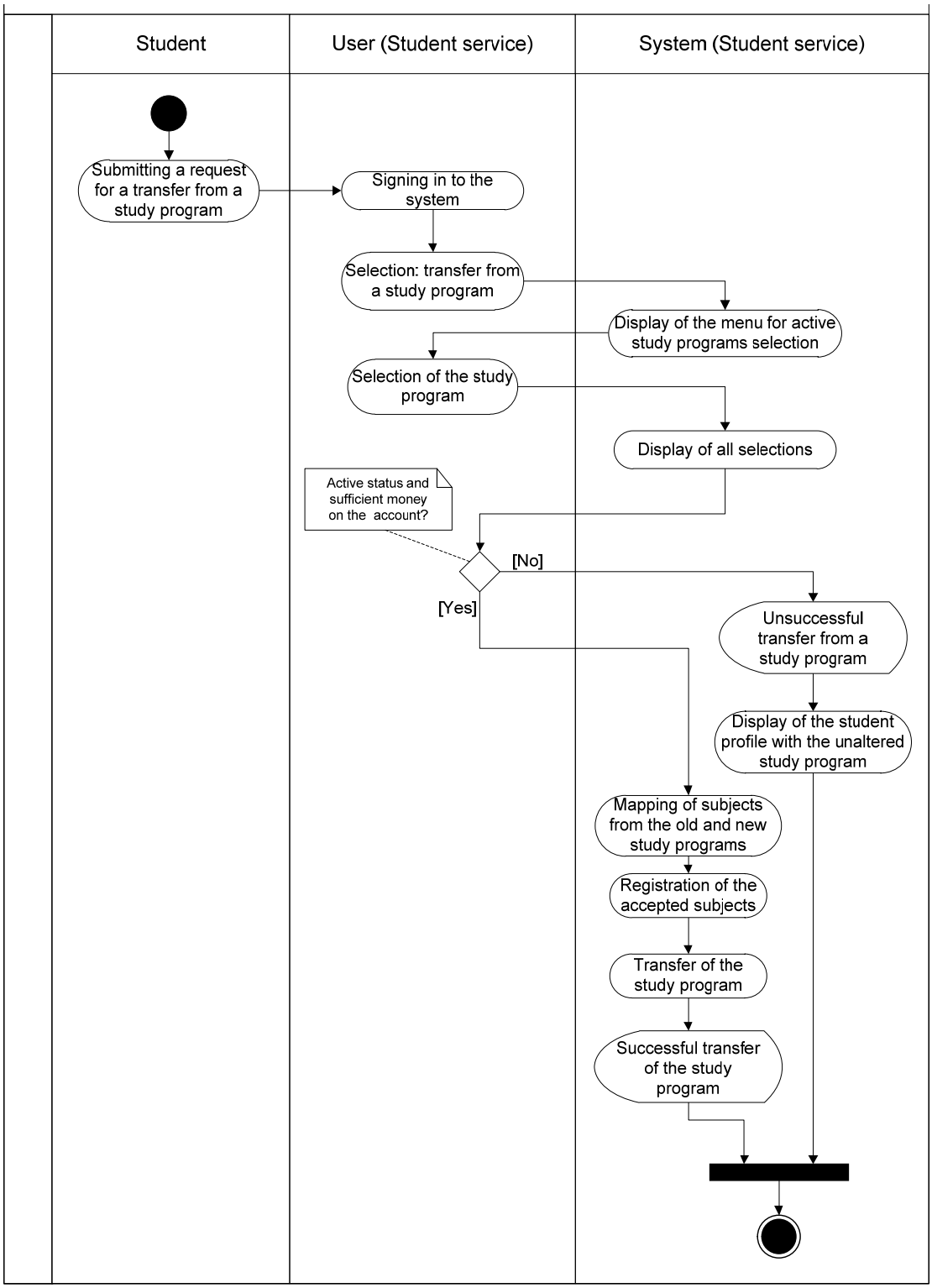
3.2.7 Issuing a certificate of exams passed



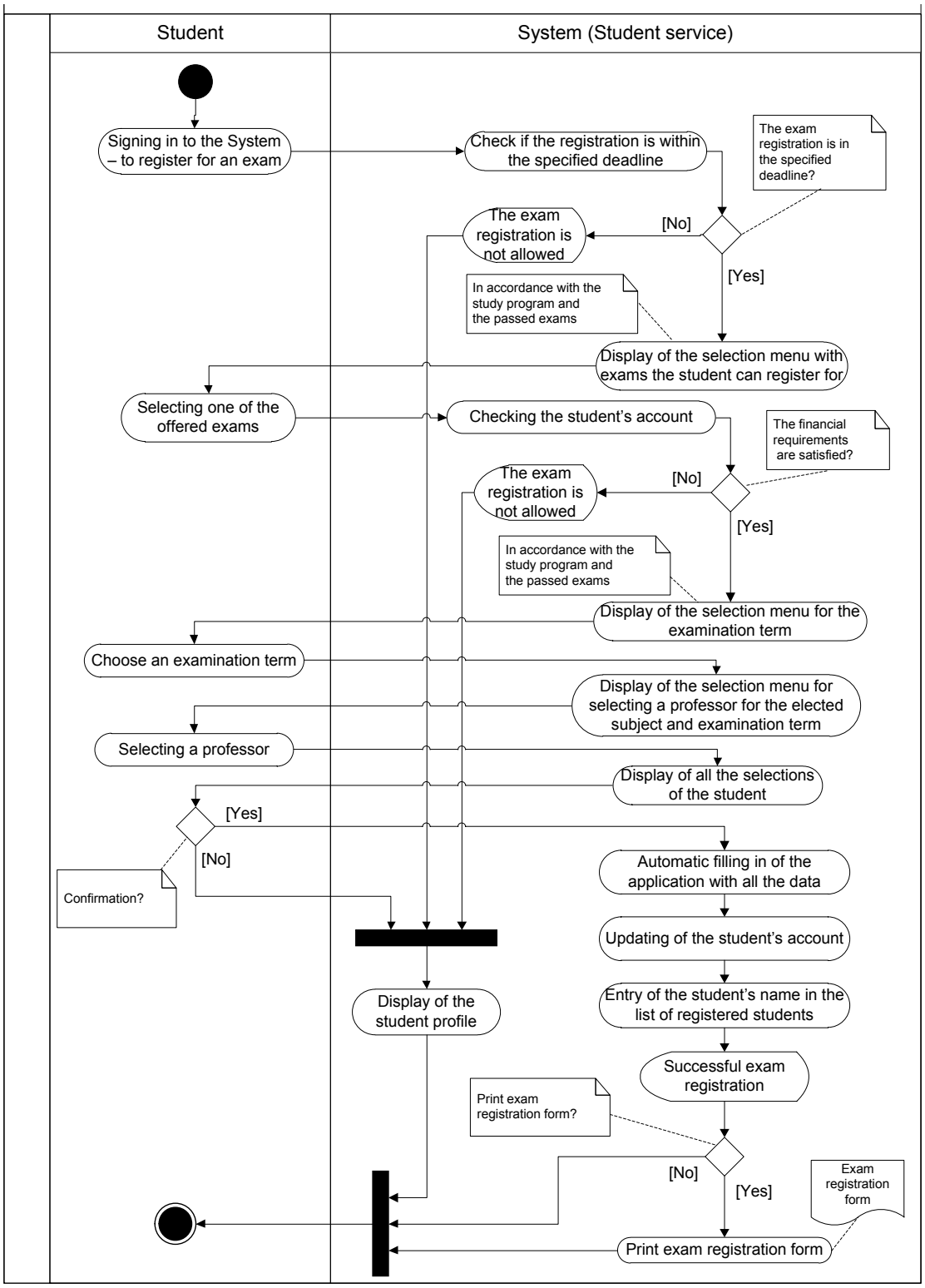
3.2.8 Rendering a report



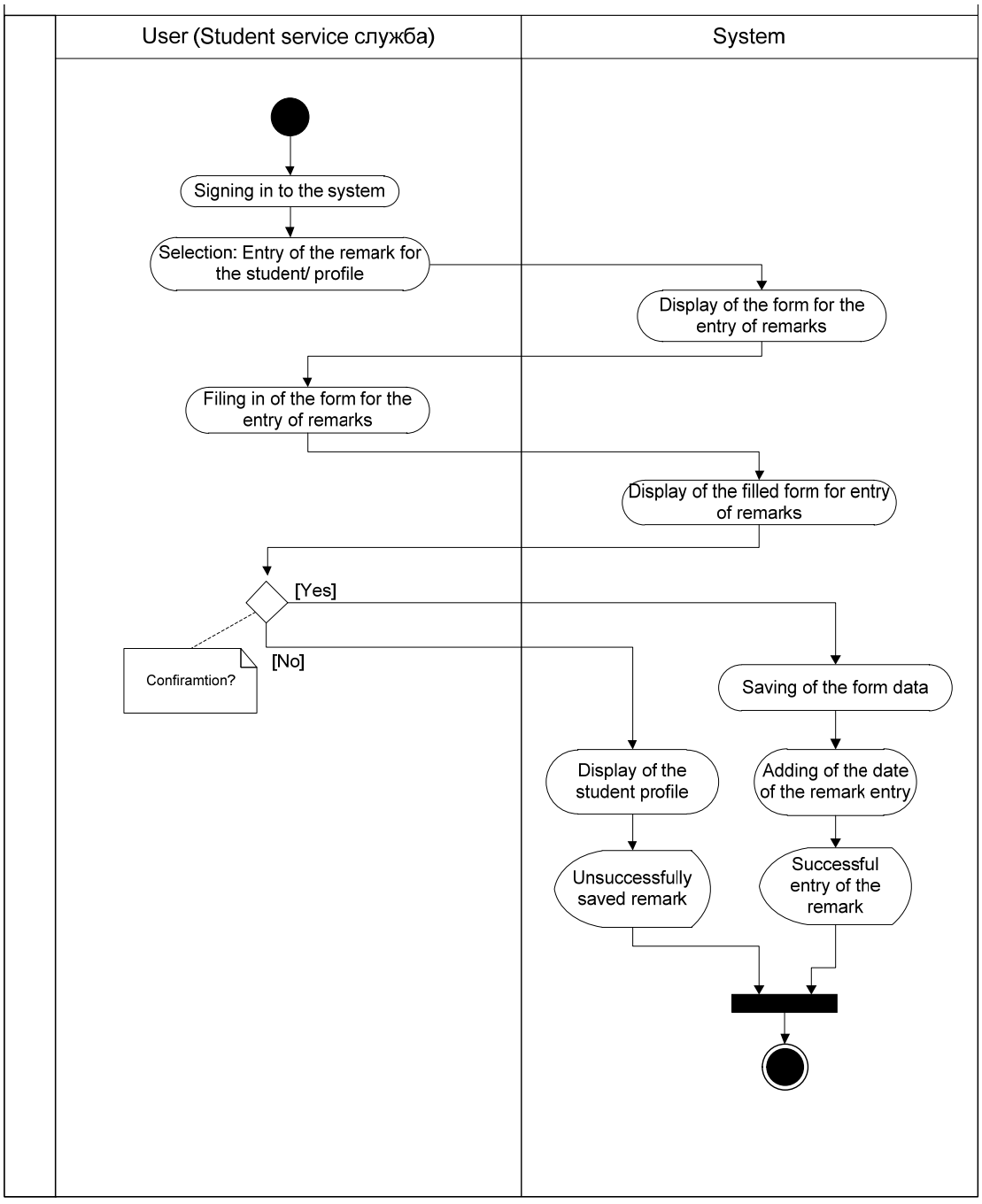
3.2.9 Changing a study program



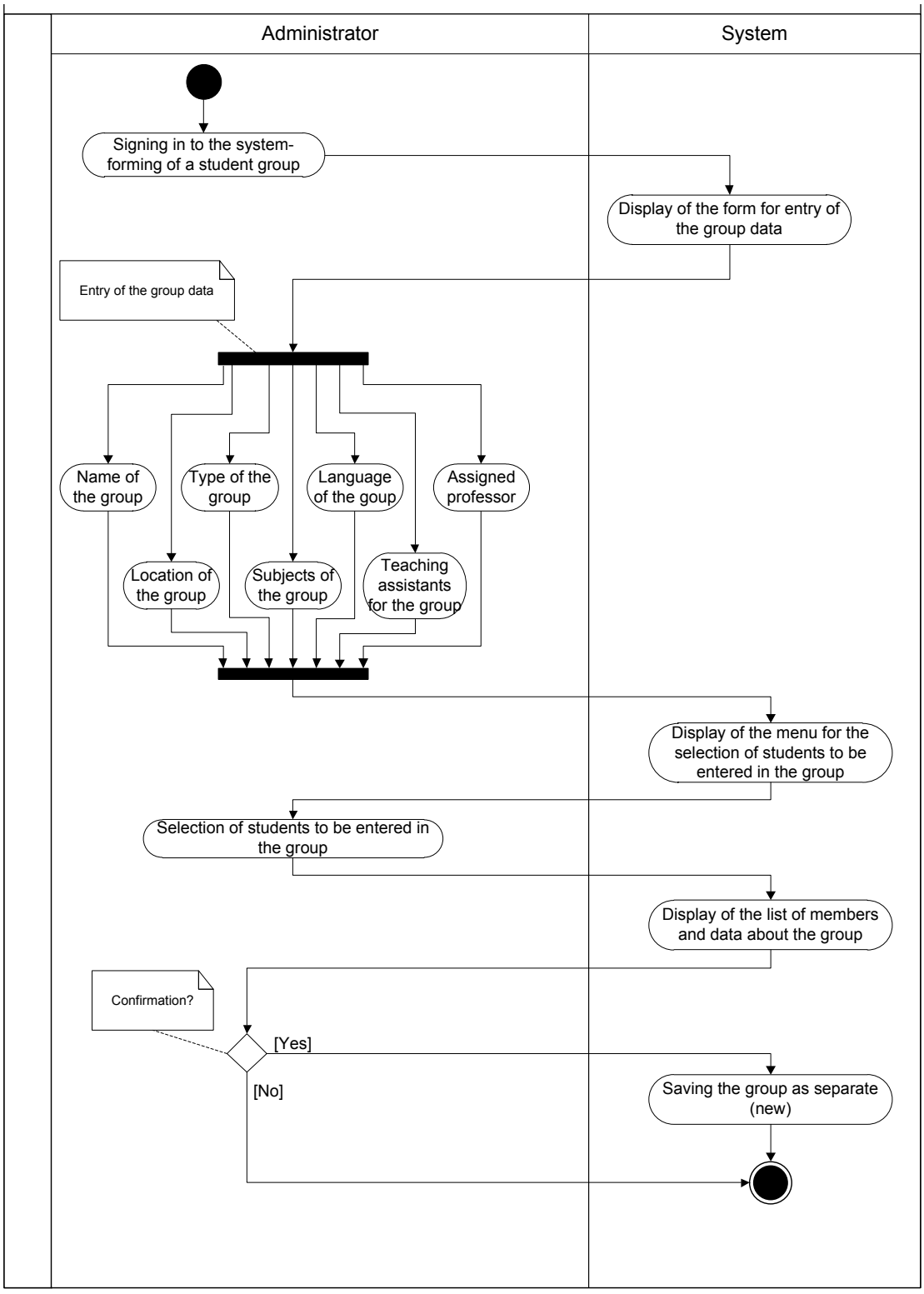
3.2.10 Applying for an exam



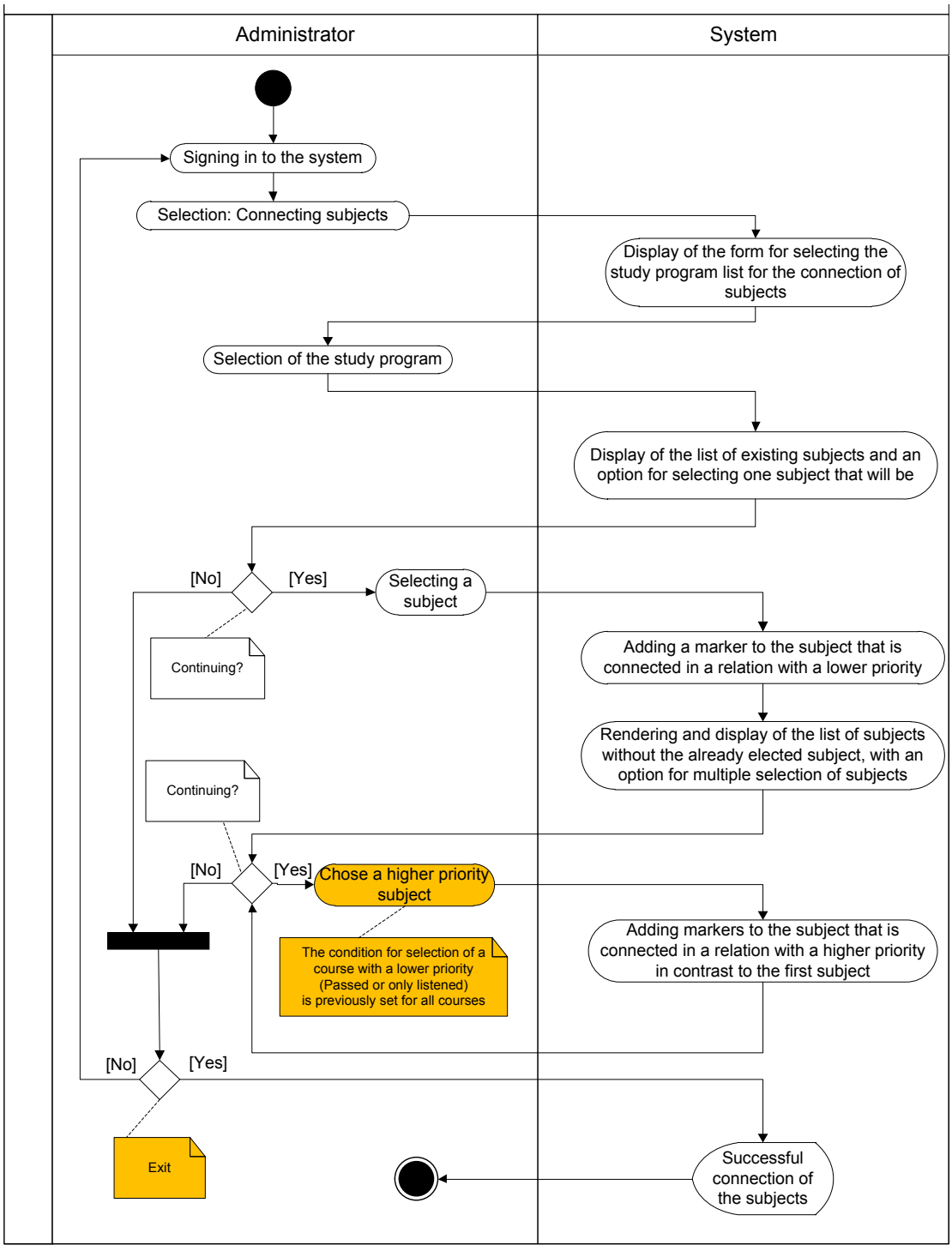
3.2.11 Entry of remarks for the student



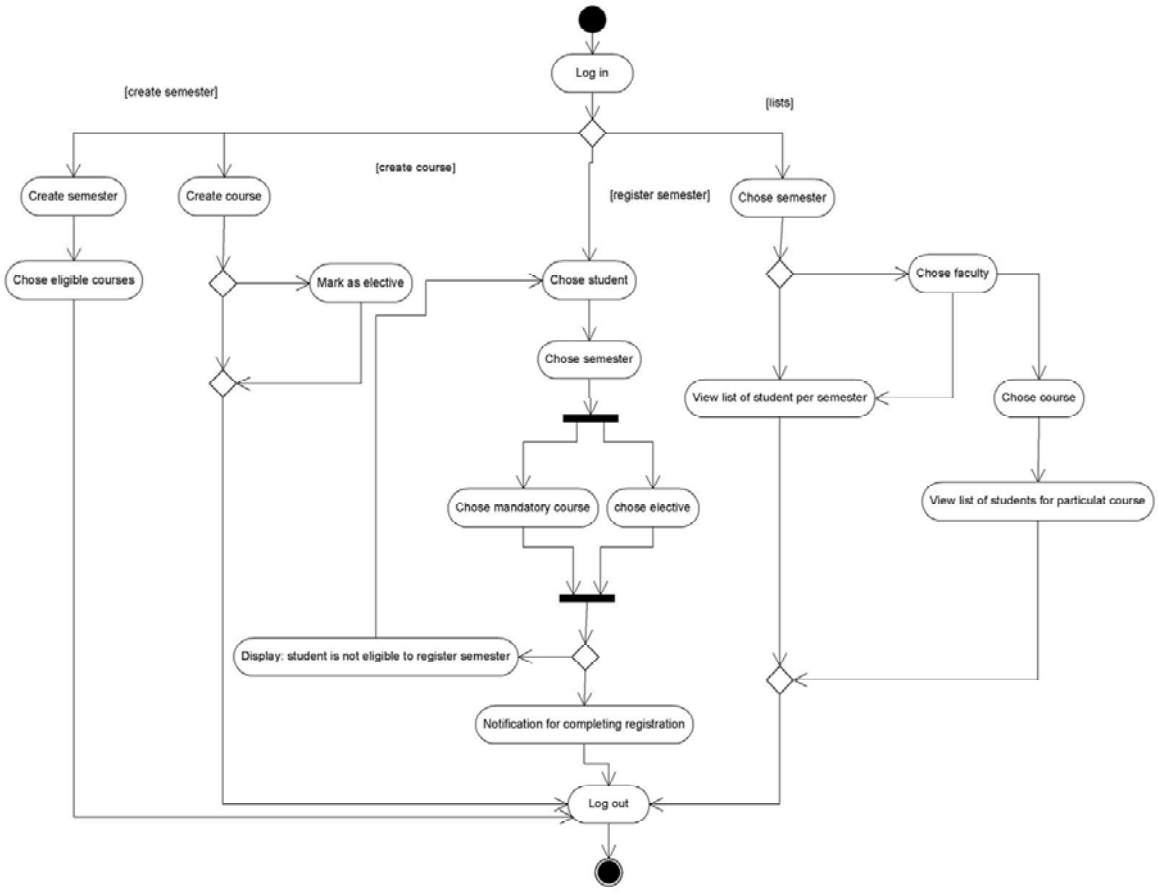
3.2.12 Forming a student group



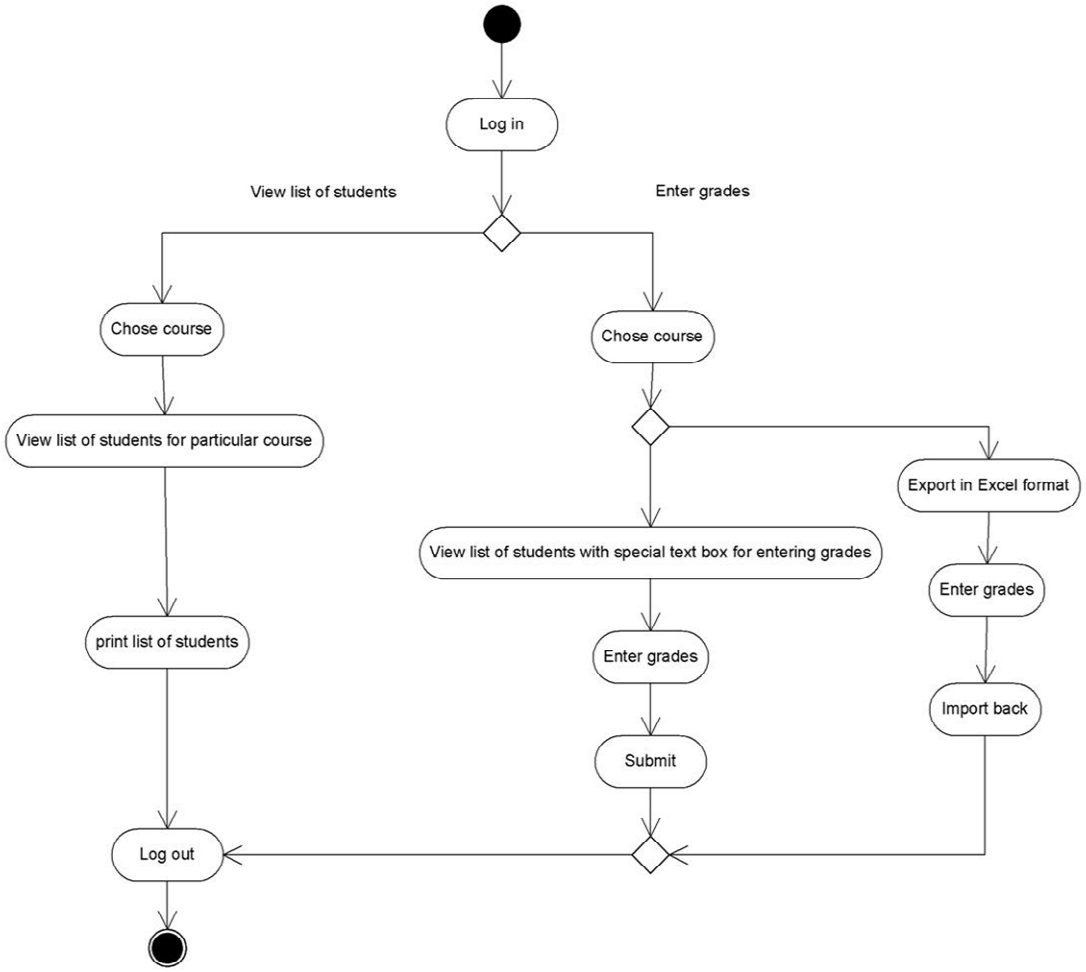
3.2.13 Defining connected subjects



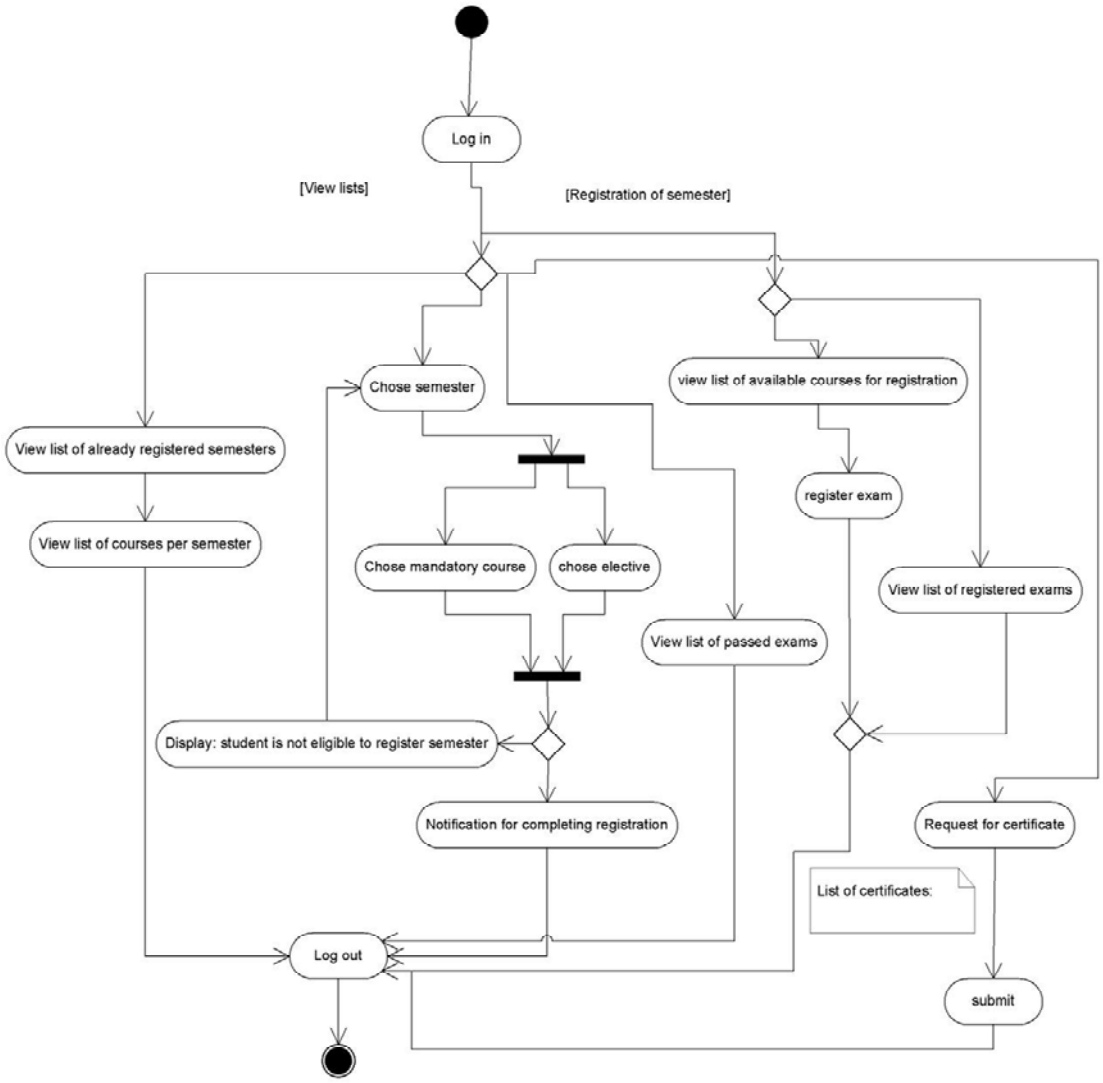
3.2.14 Officer in student services activities



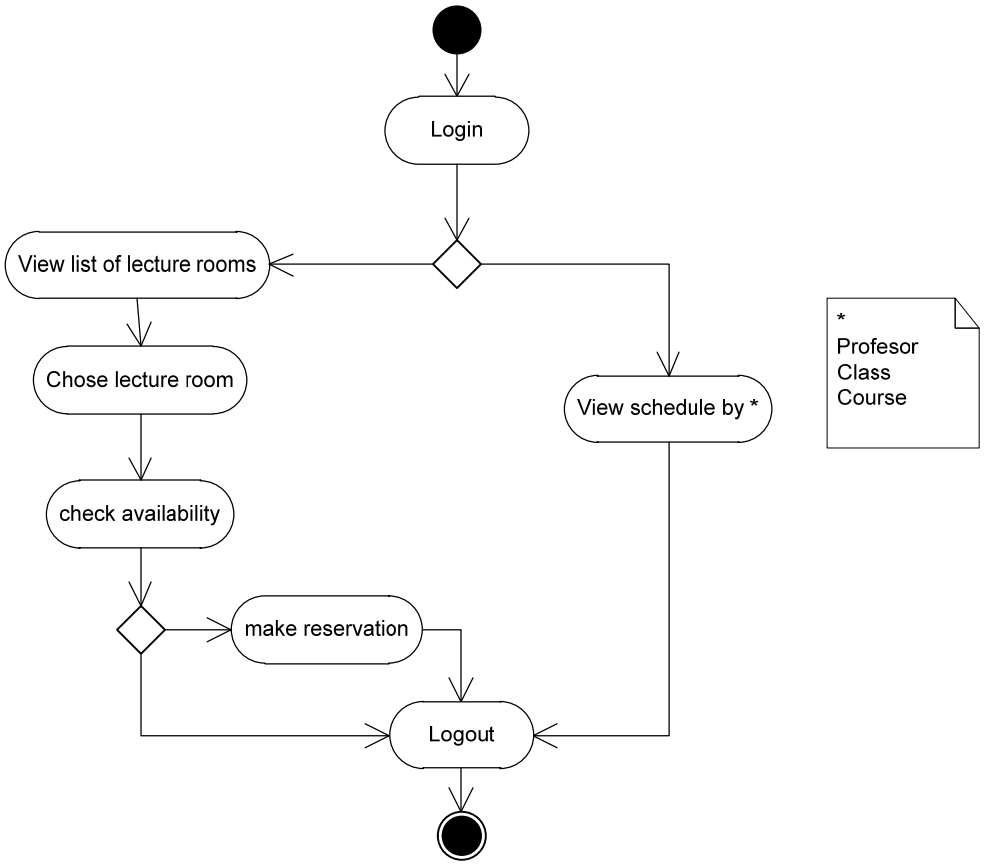
3.2.15 Teachers activities



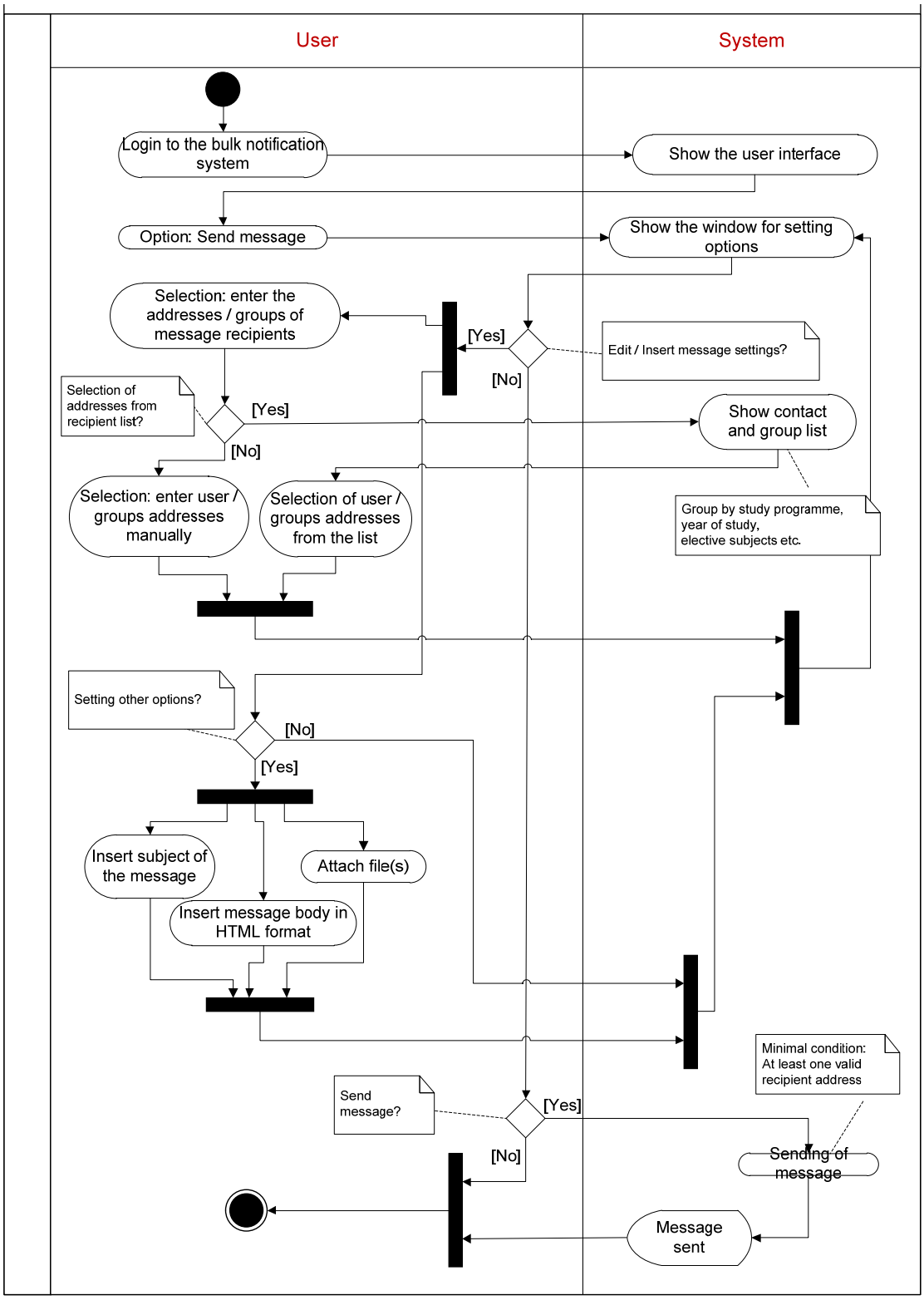
3.2.16 Student's activities



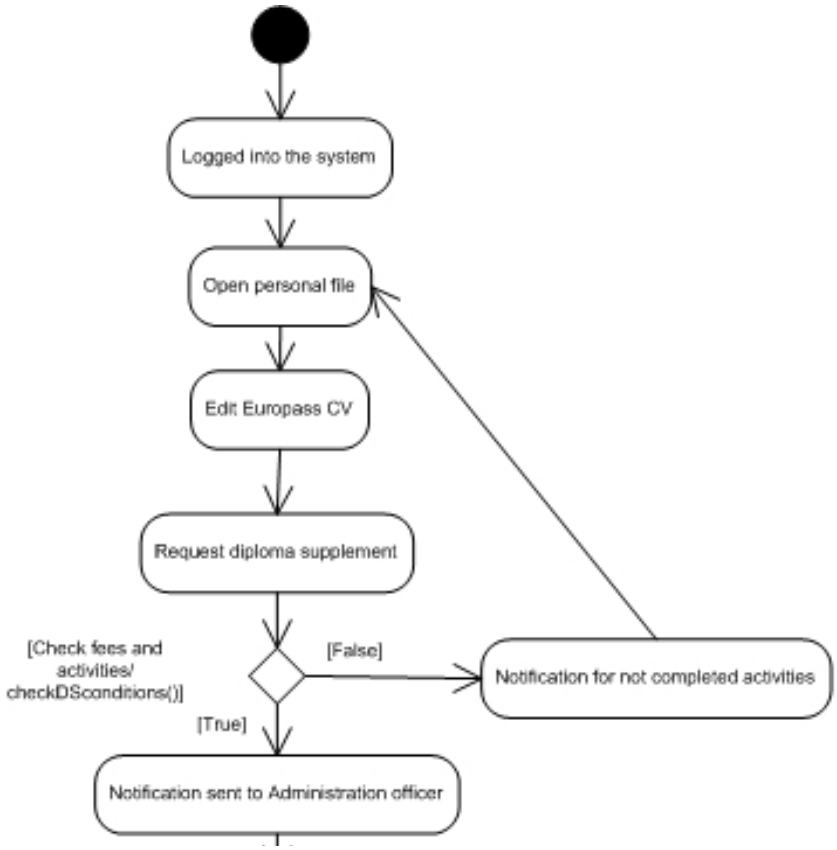
3.2.17 Class schedules



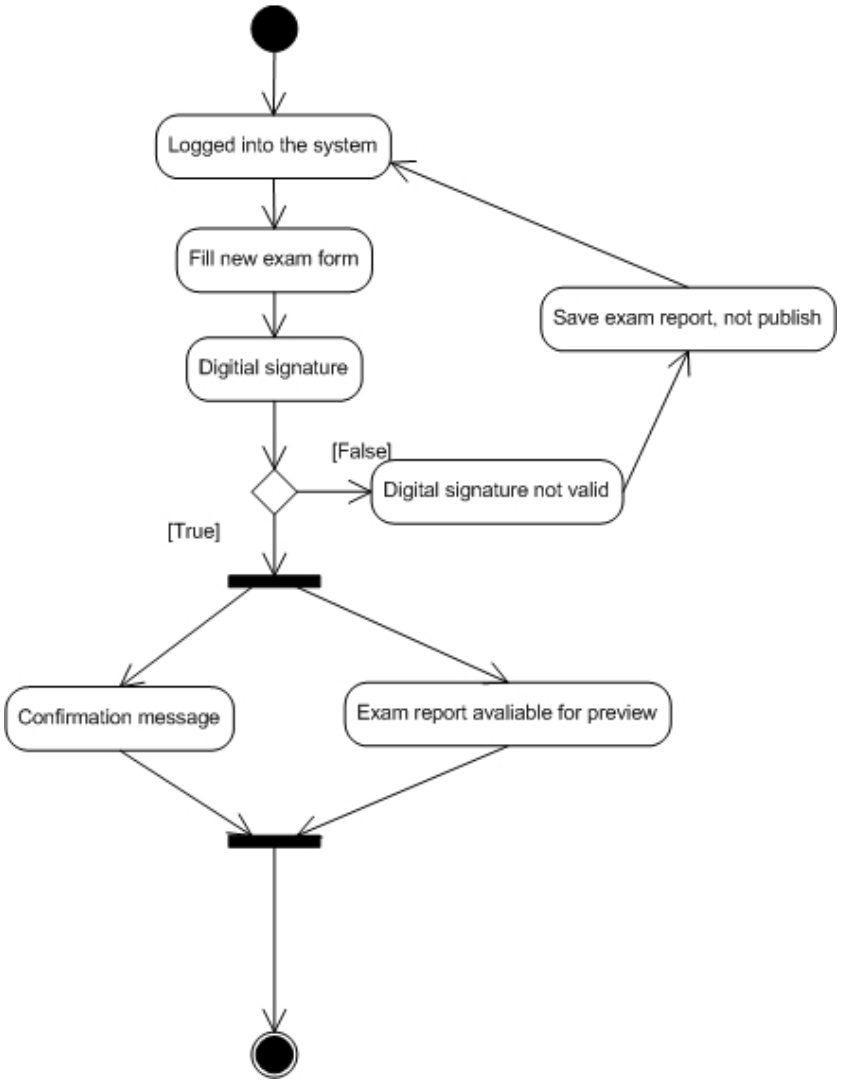
3.2.18 Bulk notifications to students



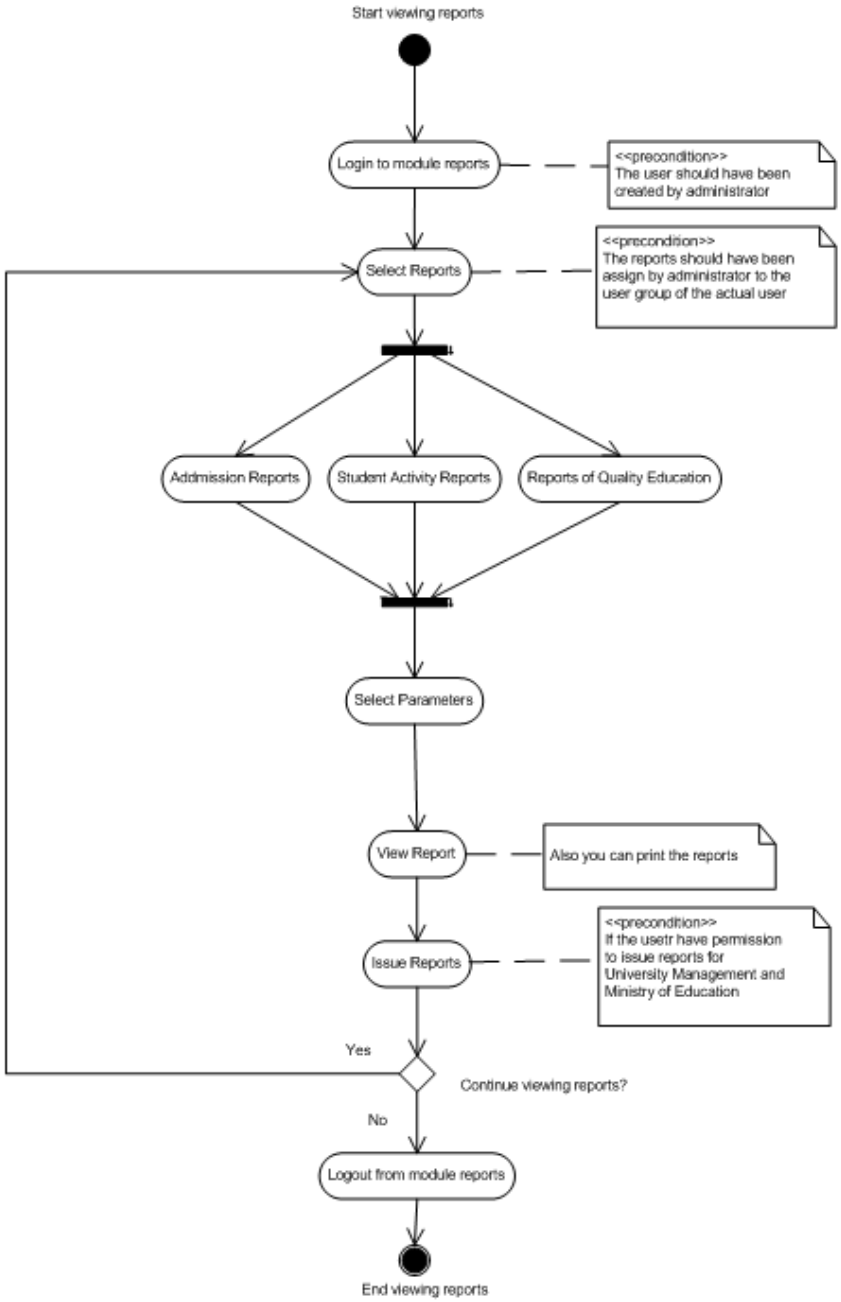
3.2.19 Europass CV issuing



3.2.20 Exam report



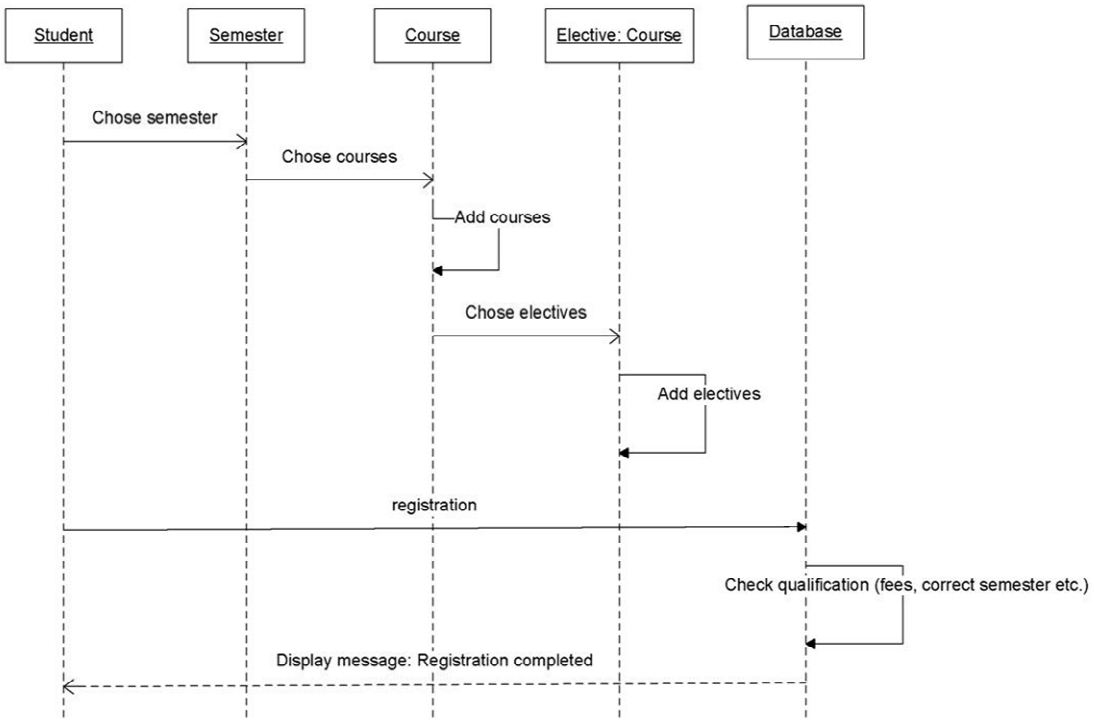
3.2.21 Request a report



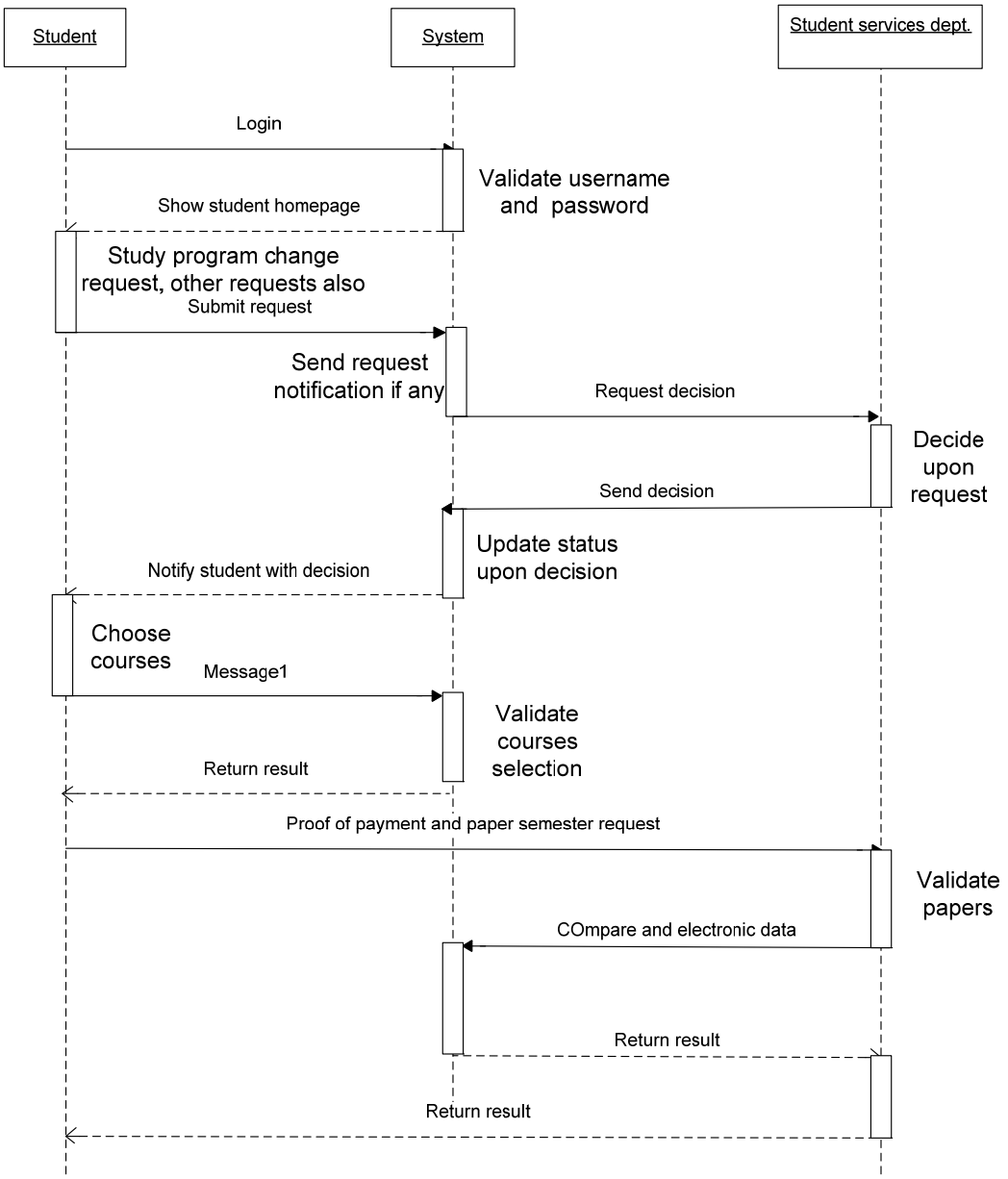
3.3 Sequence diagrams

Following UML sequence diagram shows the order of steps during the enrolment procedure, and the actors responsible for their execution.

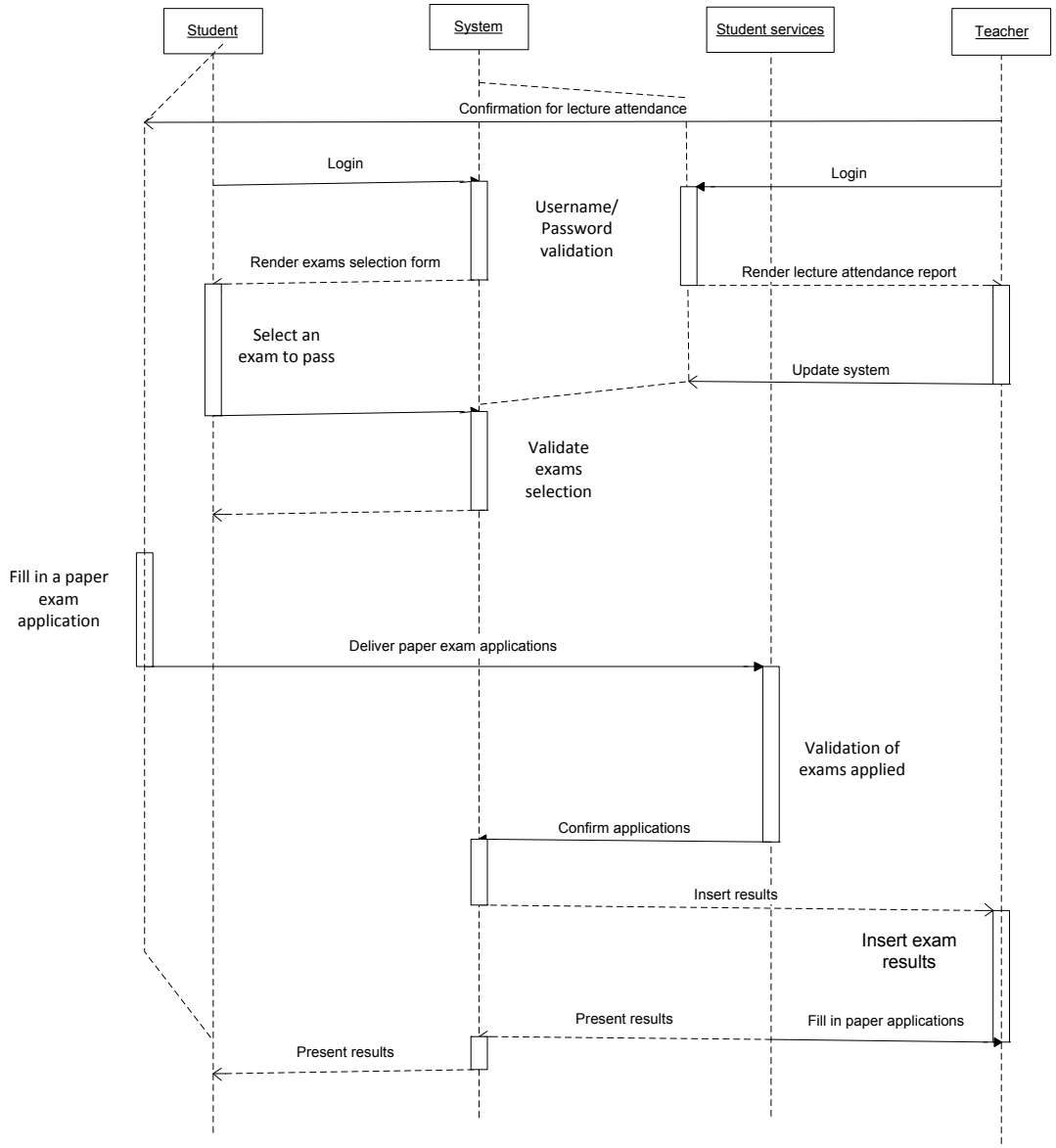
3.3.1 Enrolment process



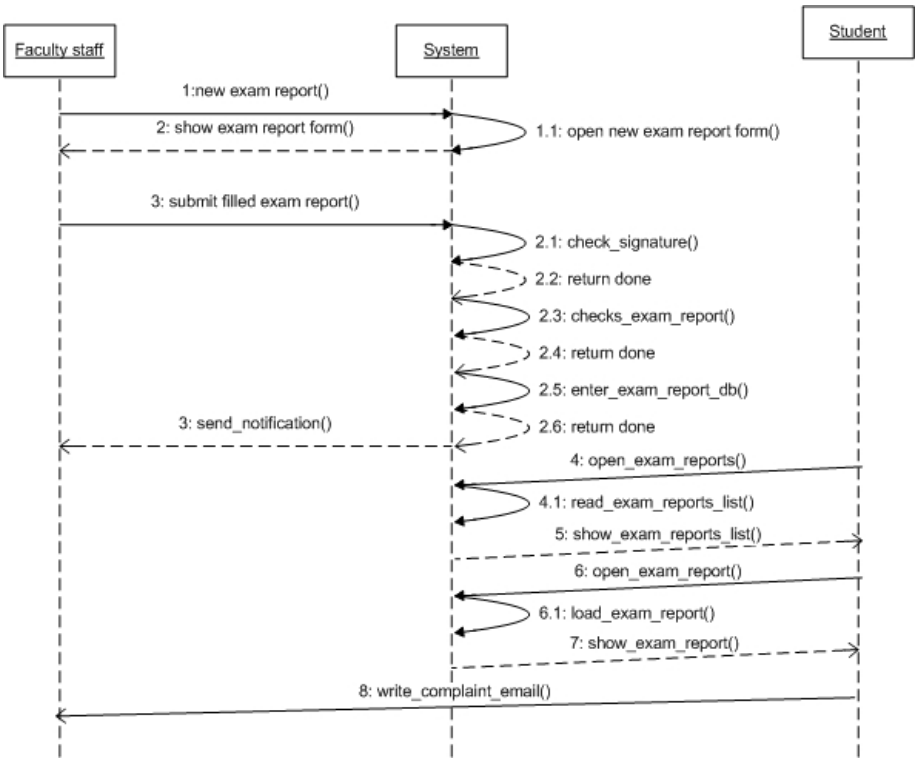
3.3.2 Semester verification



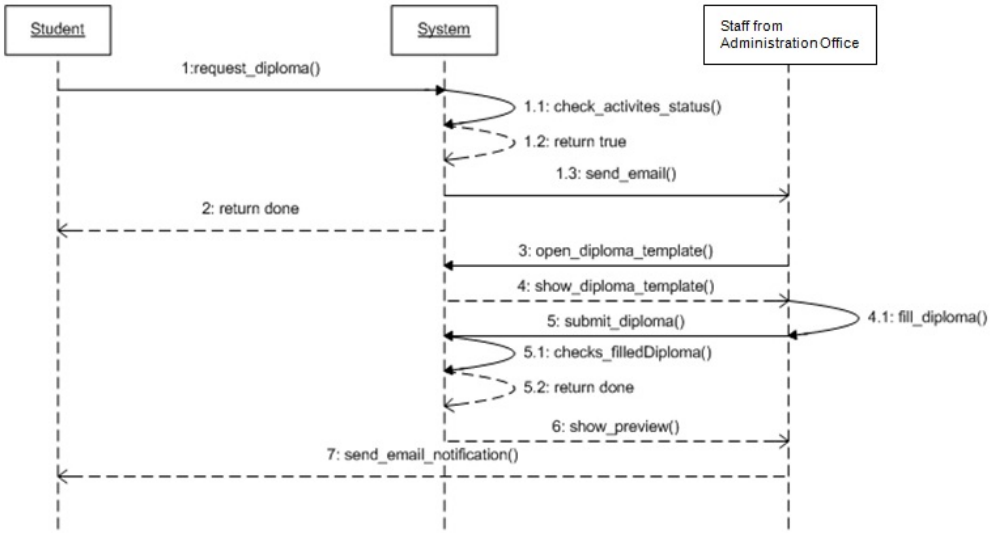
3.3.3 Applying for an exam



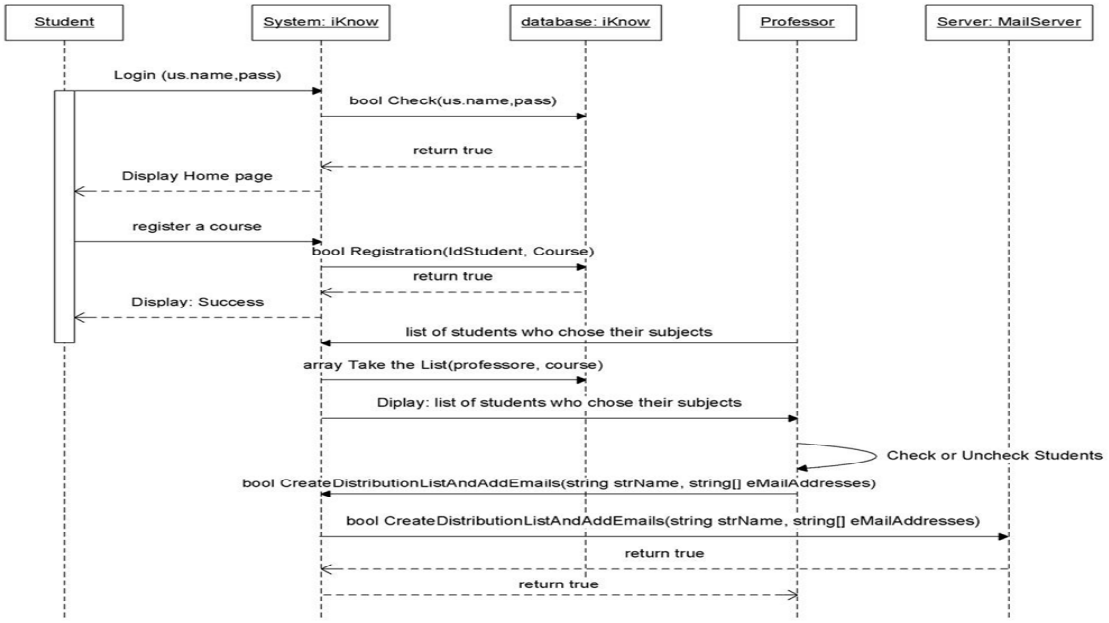
3.3.4 Exam reporting



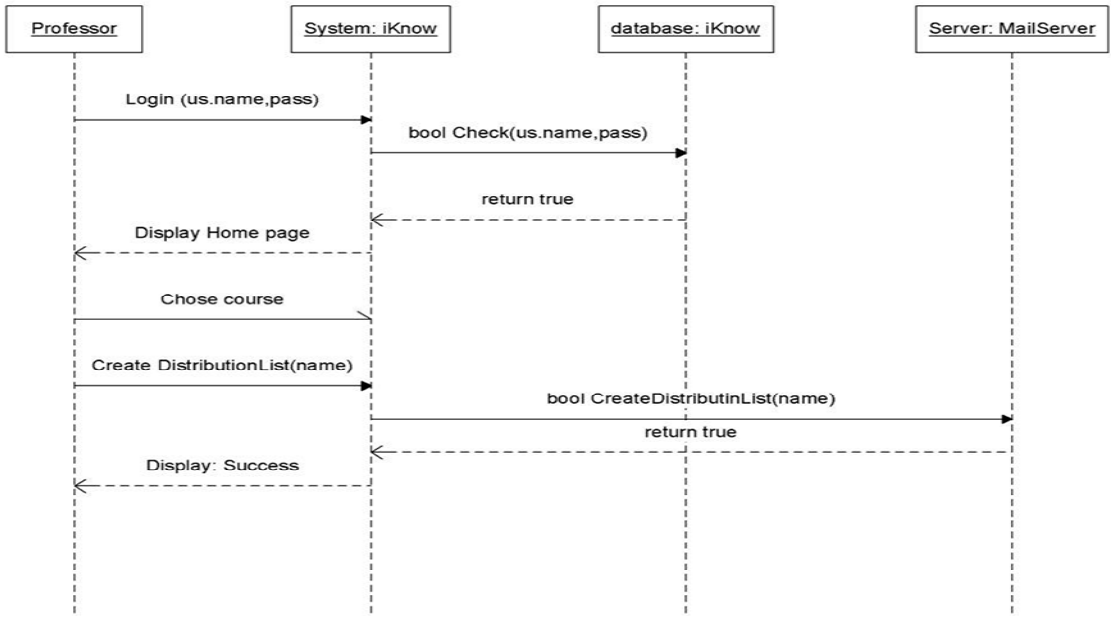
3.3.5 Diploma requesting



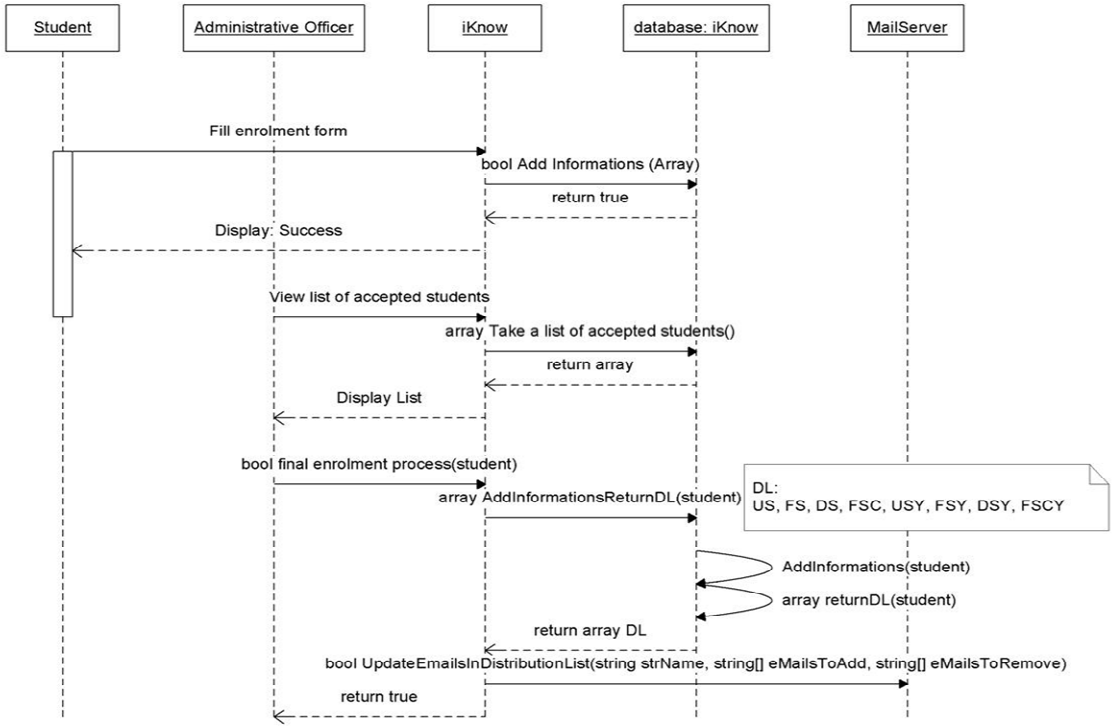
3.3.6 Course enrolment



3.3.7 Creating new e-mail Distribution List



3.3.8 Enrolment at University – email distribution list



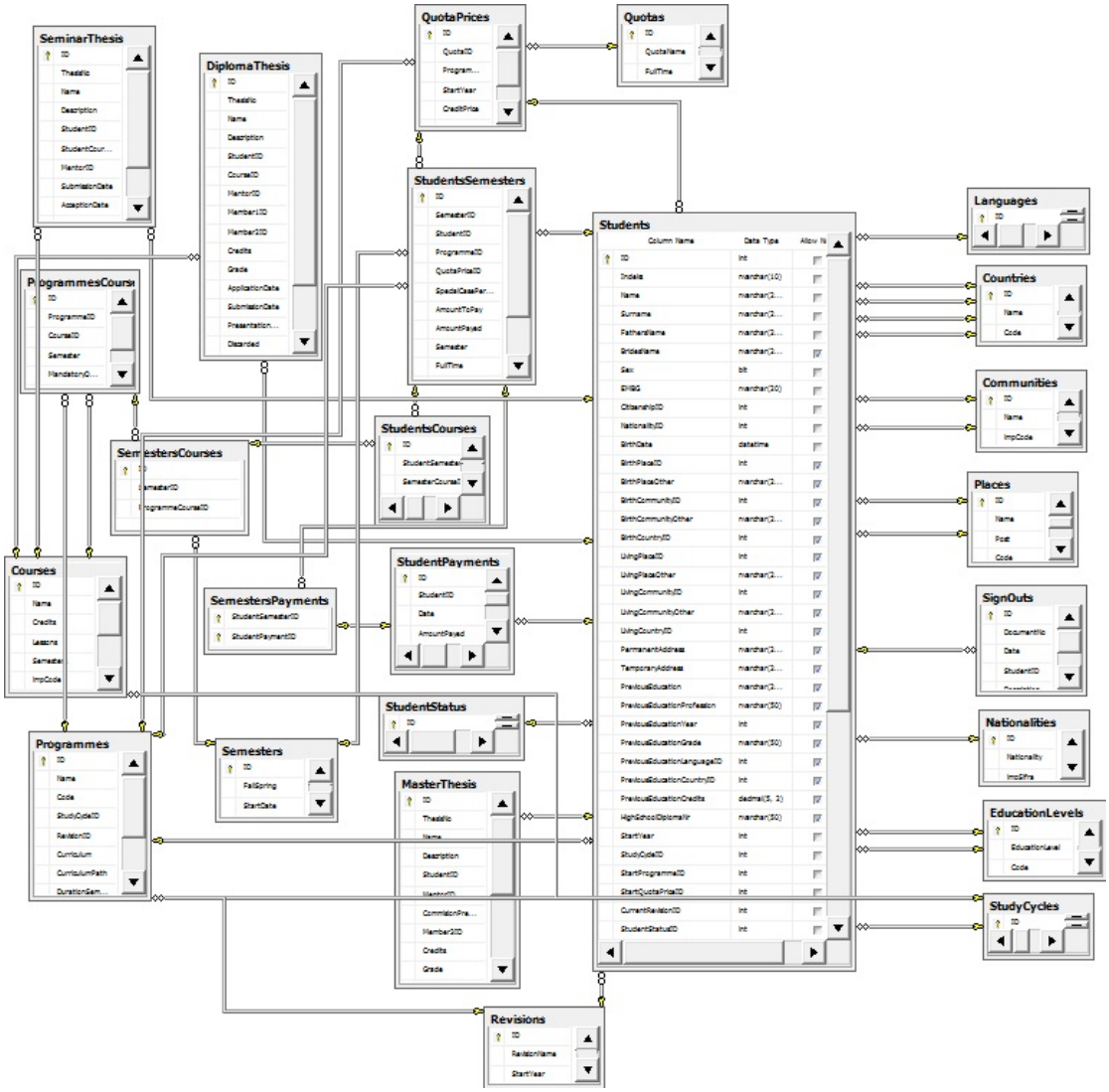
4 Data model

This section describes the data entities, and the business rules related to them.

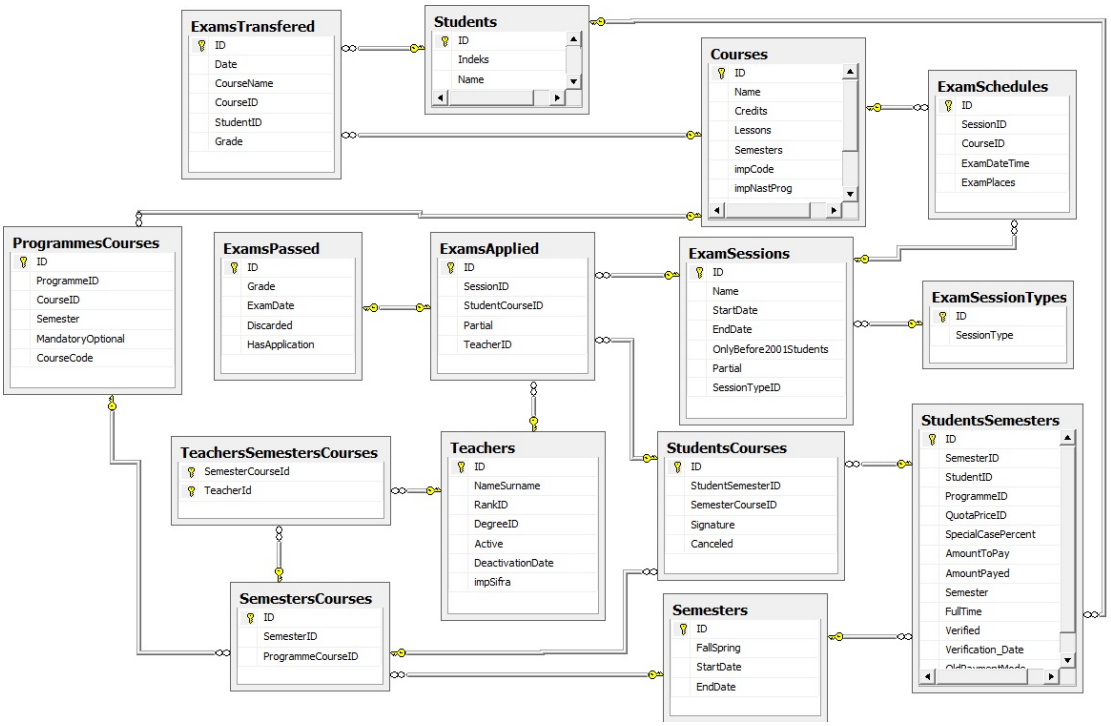
4.1 ER diagrams

This section provides info about relations among tables in database. However, the bidder may add and modify tables and relations in order to optimize the software solution and this is just basic overview of possible organization.

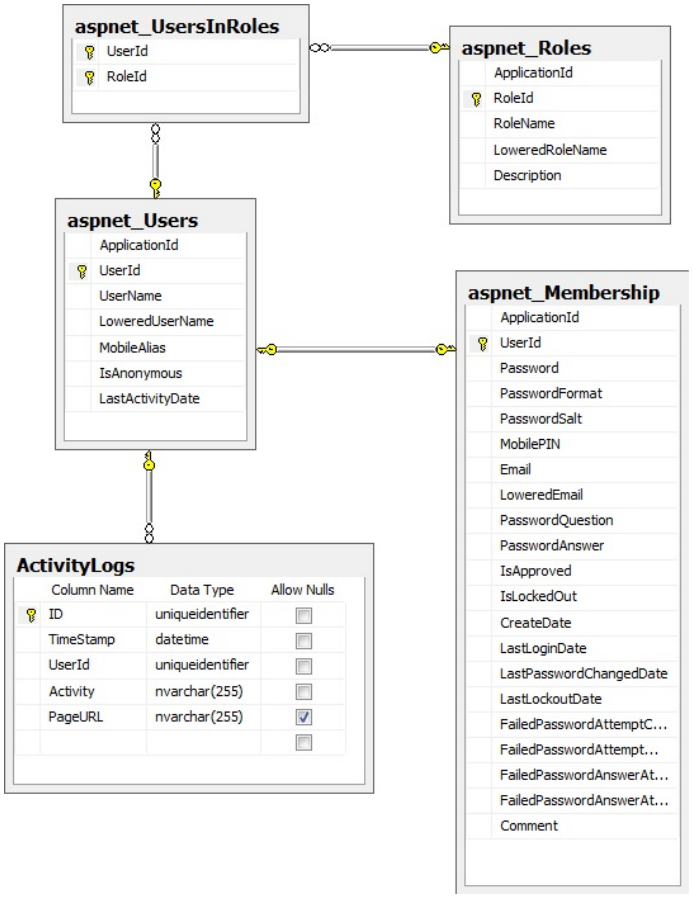
4.1.1 General relations for student information system



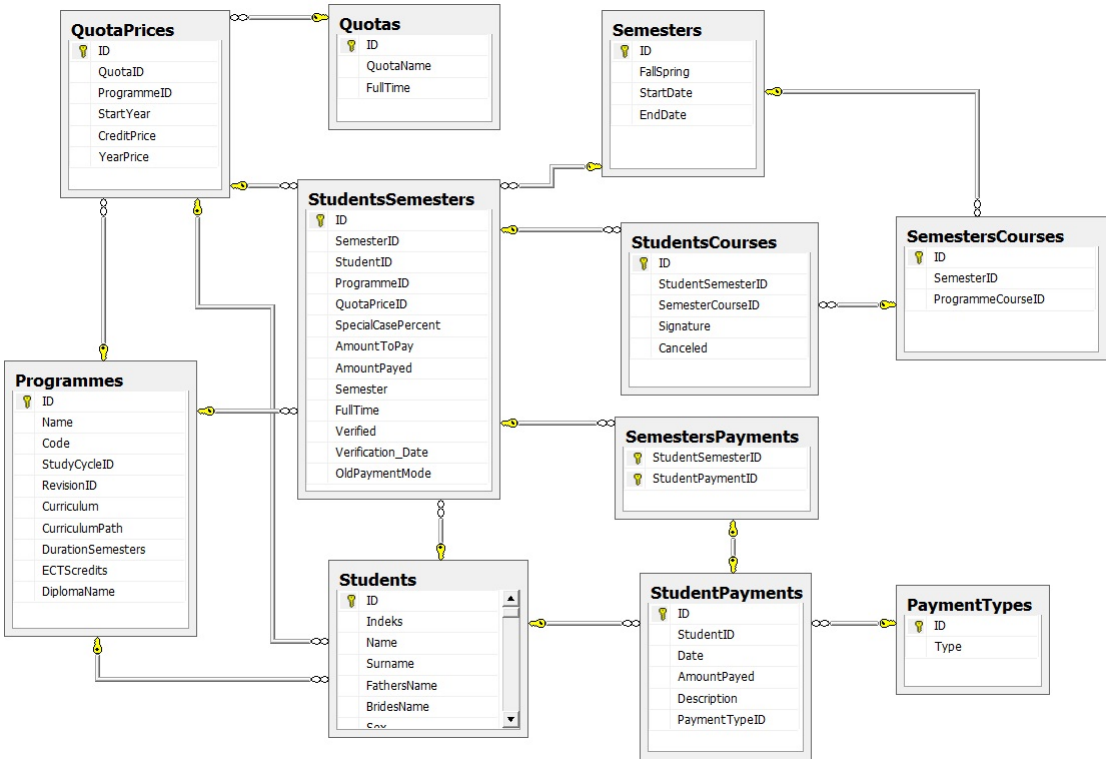
4.1.2 Relations for courses and exams



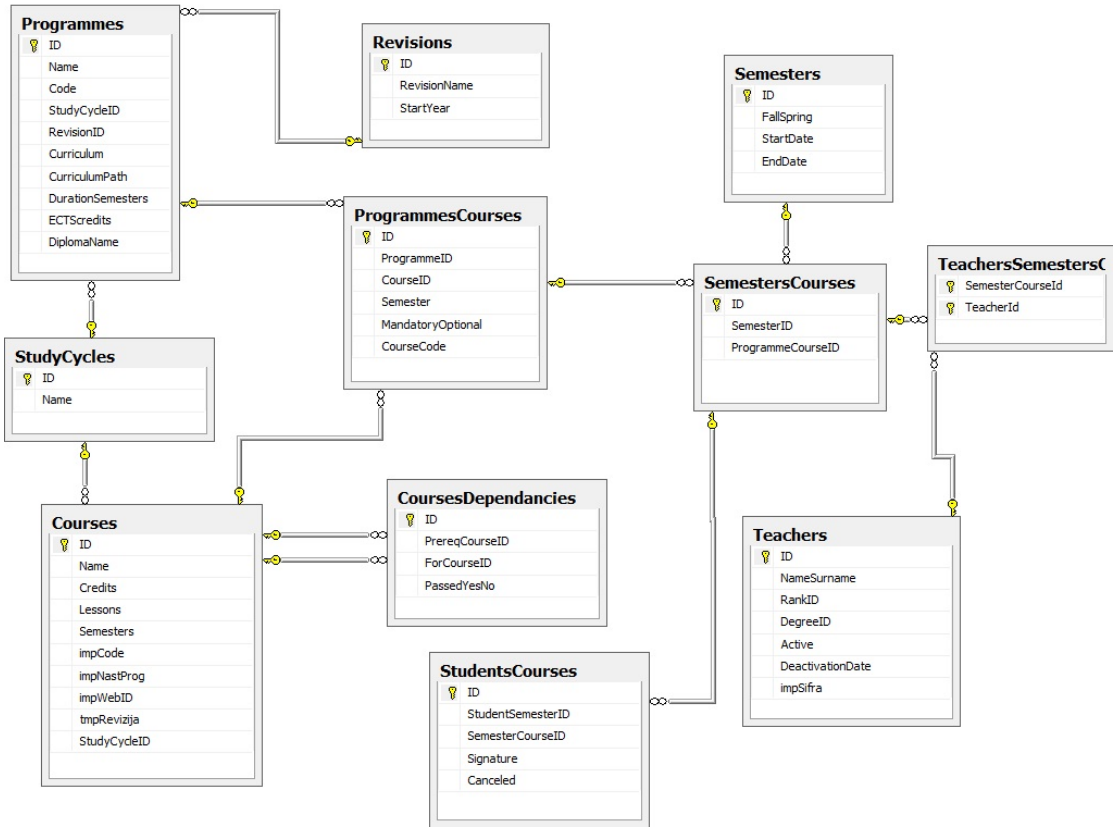
4.1.3 Membership and activity model



4.1.4 Financial model and other interfaces



4.1.5 Study programmes and courses



4.2 Description of data entities

4.2.1 Student profile

Functionality

- **List**: Provide a list of all students enrolled in the university/faculty.
- **Search** for students by various criteria (index, name, surname, program, start year, etc).
- **Maintain** the personal details stored for each student.
- **Auditing**: All changes to student's data are logged.

Fields

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Field	**Description**	**Type**	**Possible Values**
Index	Unique identifier of the student	text	
Master book number	Unique number in the master book of students	number	
Faculty	Name of the Faculty	Selection, Single	Stored list in a database
Study program	Name of the program	Selection, Single	Stored list in a database
Location	Location of the studies	Selection, Single	Stored list in a database
Status	Student status	Selection, Single	Full time / Part time
Language	Study language	Selection, Single	Macedonian, Albanian, English
Study cycle	Cycle of studies	Selection, Single	1,2,3
Start year	Year of enrolment	Number	
Previous credits	Credits from previous education	Decimal	
Previous diploma number	Previous diploma number	text	
Initial quota	Payment quota in the time of enrolment	Selection	Stored list in the database

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Personal Data			
Surname	Surname	text	
Maiden surname	Maiden surname	text	
Father's name	Father's name	text	
Name	Name	text	
EMBG	Unique number of the citizen	text	
Date of birth	Date of birth	Date	
Place of birth	Place of birth	text	
Municipality of birth	Municipality of birth	text	
Birth Country	Country of birth	text	
Sex	Sex	Selection, Single	Male / Female
Citizenship	Citizenship	text	List stored in the database
Etnicity	Etnicity	text	

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Address			
Town	Place of living	text	
Street	Street	text	
Number	Street Numbers	text	
Municipality	Municipality	text	
Country	Country	text	

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Contact/Phones			
Home	Home phone number	text	
Cellular	Cellular Phone number	text	
Office	Office phone number	text	
E-Mail	e-mail	text	

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Previous education			
Name of previous school	Name of previous school	text	
Town	Place of previous education	text	
Profession	Profession, educational profile of the previous education	text	
Average grade of the previous education	Average grade of the previous education	Decimal number	
Foreign languages studied	Foreign languages studied in the previous education	Selection, Multiple	Stored list in the database
Tuition payer	Person paying the tuition	Selection, Multiple	Parent/Guardian; personally, Other person; completely free of charge; partial discount
Year of completion	Year of completion	Number	

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Personal data for the parent / guardian			
Name and Surname	Name and Surname	text	
Profession	Profession	text	
Employment	Employment	text	

Data stored for each student during studies and after			
Initial details acquired from the enrolment process			
Other			
Scholarship	Whether the student is receiving a scholarship	Selection, Single	Yes/No
Scholarship provider	Provider of the scholarship	text	

Business rules and access rights

- Fields are grouped in blocks according to related topic.
- Some of the fields are connected (changing the start year affects the available list of study programs and quota prices; change in the start study program changes the available quota prices; change in study cycle changes the available list of programs, etc).
- Students can access for reading most of the fields (their own record only), while can only change a small subset (contact information etc.). Officers at the student services can read and change all data for all students (limitations for officers at one faculty for students at that faculty only).
- There is a different rule for generating index numbers in each faculty.

4.2.2 Staff and members of faculty profile

Functionality

- **List**: Provide a list of all staff members.
- **Search** for staff members by name, surname, education, title.
- **Maintain** the list of staff members.
- **Auditing**: All changes to staff members are logged.

Fields

File form for academic staff			
Field	**Description**	**Type**	**Possible Values**
Name	Name	Text	
Surname	Surname	Text	
EMBG	Citizen ID of employee	Text	
Residence	Residence	Text	
Municipality	Municipality of residence	Selection, Single	Stored list in a database
Profession	Profession	Selection, Single	Stored list in a database
Undergraduate title	Diploma degree title	Selection, Single	Stored list in a database
Undergraduate field	Field of earned degree	Selection, Single	Stored list in a database
Undergraduate date	Date of accomplishment of undergraduate studies	Date	
Undergraduate institution	Institution of undergraduate studies	Selection, Single	Stored list in a database
Doctorate title	Doctorate degree title	Selection, Single	Stored list in a database
Doctorate field	Field of earned degree	Selection, Single	Stored list in a database
Dissertation	Doctor dissertation title and mentor	Text	
Doctorate date	Date of accomplishment of doctorate studies	Date	
Doctorate institution	Institution of doctorate studies	Selection, Single	Stored list in a database
Election	Election of title for subject or field	Text	
Announcement Date	Announcement date for election of title	Date	
Election Date	Election date	Date	
Expiration date	Date of expiration of election	Date	
Entrusted subjects	Entrusted subjects	Selection, Multiple	All courses in the database
Releases	Books released, or other teaching materials	Text	
Function	Function	Text	
Faculty	Assigned to faculty	Selection, Single	
Employment Date	Employment Date	Date	
Termination Date	Date of termination of employment	Date	

Business rules and access rights

- Fields are grouped in blocks according to related topic.

- Staff members (teachers) can access for reading most of the fields (their own record only), while can only change a subset (contact information etc.). Administrators can read and change all data.
- Administrator confirms and accepts changes made by teachers in their profile.

4.2.3 Faculty staff's studies

Functionality

- **List**: Provide a list of all academic achievements of faculty members (diplomas).
- **Search** for diplomas staff member, type of education, title.
- **Maintain** the list of diplomas.
- **Auditing**: All changes to diplomas are logged.

Fields

File form for academic achievements of accademic staff			
Field	**Description**	**Type**	**Possible Values**
Staff member	The staff member in question	Text / ID	
Postgraduate title	Masters degree title	Selection, Single	Stored list in a database
Postgraduate field	Field of earned degree	Selection, Single	Stored list in a database
Thesis	Master thesis title and mentor	Text	
Postgraduate date	Date of accomplishment of postgraduate studies	Date	
Postgraduate institution	Institution of postgraduate studies	Selection, Single	Stored list in a database

Business rules and access rights

- Staff members (teachers) can access all fields (their own record only) and make changes that the administrator should confirm later. Administrators can read and change all data.
- Administrator confirms and accepts changes made by teachers in their profile.

4.2.4 Revisions of study programs

Functionality

- **List**: Provide a chronological list of all revisions of study programs.
- **Search** for revisions of study programs.
- **Maintain** the list of revisions.
- **Auditing**: All changes to revisions are logged.

Fields

Revisions of study programs			
Field	**Description**	**Type**	**Possible Values**
Name	Name	Text	
Start year	The beginning year of the new study programs	number	
Faculty	The faculty that the revision is associated to	Text / ID	Stored in the database

Business rules and access rights

- Administrators, vice-deans (for their faculty only), student services officers (for their faculty only) can read and change all data.

4.2.5 Study programs

Study program is the officially acknowledged and accredited comprehensive set of compulsory and elective course units that define the principal field of study

Functionality

- **List**: Provide a list of all study programs.
- **Search** for study programs according to revisions, study cycles, names, and faculty.
- **Maintain** the list of study programs.
- **Auditing**: All changes to study programs are logged.

Fields

Study programs			
Field	**Description**	**Type**	**Possible Values**
Name	The name of the program	Text	
Code	Short name	Text	
Study Cycle	Cycle of studies for the program	Selection, Single	1, 2, 3
Revision	The revision of study programs that the program belongs to.	Number	List stored in a database
URL	Web page for the program	Text	
Duration semesters	The number of semester the program lasts	Number	
ECTS credits	Credits necessary for finishing	Number	
Diploma Name	The name of the diploma received	Text	
Faculty	The faculty that the program is associated to	Text / ID	List stored in the database

Business rules and access rights

- Administrators, vice-deans (for their faculty only), student services officers (for their faculty only) can read and change all data.

4.2.6 Quotas

Functionality

- **List**: Provide a list of all quotas that the students can be enrolled in.
- **Search** for quotas according to study cycles, names, faculty.
- **Maintain** the list of quotas.
- **Auditing**: All changes to quotas are logged.

Fields

Quotas			
Field	**Description**	**Type**	**Possible Values**
Name	Name	Text	
Full Time	Full time / part time	Selection, Single	1,0

Business rules and access rights

- Administrators, student services officers (for their faculty only) can read and change all data.

4.2.7 Quota prices

Functionality

- **List**: Provide a list of all quota prices per quota per year per faculty that the students can be enrolled in.
- **Search** for quota prices according to quota, study cycles, faculty.
- **Maintain** the list of quota prices.
- **Auditing**: All changes to quota prices are logged.

Fields

Quota prices			
Field	**Description**	**Type**	**Possible Values**
Quota	Name	Selection, Single	Stored list in the database
Study Program	The study program that the quota price is applicable to	Selection, Single	Stored list in the database
Start year	Starting year of validity of the price	Number	
Price per credit	Price per ECTS credit	Number	
Price per year	Price for whole year	Number	
Faculty	The faculty that the quota price is associated to	Text / ID	Stored in the database

Business rules and access rights

- Administrators, student services officers (for their faculty only), financial officer can read and change all data.

4.2.8 Financial services

Functionality

- **List**: Provide a list of all prices for financial services per year per faculty per study program.
- **Search** for prices for financial services according to various criteria.
- **Maintain** the list of all prices for financial services.
- **Auditing**: All changes to prices for financial services are logged.

Fields

Financial services			
Field	**Description**	**Type**	**Possible Values**
Service	Name (type) of the service provided	Selection, Single (ID)	Stored list in the database (there should be a editable table defined of services in the database)
Study Program	The study program that the service price is applicable to	Selection, Single	Stored list in the database
Start year	Starting year of validity of the price	Number	
Amount	Amount of the price - cost	Number	
Faculty	The faculty that the price is related to	Selection, Single	Stored list in the database
Student Quota	The quota of students that the price is associated to	Text / ID	Stored list in the database

Business rules and access rights

- Administrators, student services officers (for their faculty only), financial officer can read and change all data.
- The table is used to calculate the amount a student should pay when being provided with a service by the university.

4.2.9 Semesters

Functionality

- **List**: Provide a chronological list of semesters.
- **Search** for semesters.
- **Maintain** the list of semesters.
- **Auditing**: All changes to semesters are logged.

Fields

Semesters			
Field	**Description**	**Type**	**Possible Values**
Start date	Starting date of the semester	Date	
End date	Ending date of the semester	Date	
Fall/Spring	A flag differentiating fall and spring semesters	Selection, Single	Fall, Spring

Business rules and access rights

- Administrators, student services officers (for their faculty only) can read and change all data.
- Each semester keeps a different constellation of teachers covering courses at different study programs.

4.2.10 Courses

Functionality

- **List**: Provide a list of all courses that the students can take.
- **Search** for courses according to names, study program, cycle, semester, credits, etc.
- **Maintain** the list of courses.
- **Auditing**: All changes to courses are logged.

Fields

Courses			
Field	**Description**	**Type**	**Possible Values**
Name	Name of the course	Selection, Single	Stored list in the database
Study Program	The study programs that the course is applicable to	Selection, Multiple	Stored list in the database
Mandatory	Is the course mandatory or optional in each study program that the course is listed	List	True/false
Semester	The semester that the course is taught in each of the study programs where it is listed	List	Numbers
Semesters	The number of semesters the course is taught	Number	
Credits	Number of ECTS credits	Number	
Faculty	The faculty that the course is associated to	Text / ID	Stored in the database
Study cycle	In which study cycle the course is taught	Selection, Single	1, 2, 3
Lessons	Weekly number of classes	Text	

Business rules and access rights

- Administrators, student services officers (for their faculty only) can read and change all data.
- The system should allow definition of equivalence (compatibility for recognition) of courses from previous programs of study subjects with the current study programs.
- In the system, multiple teachers can teach one course
- In the system, one course belongs to one or more study programs.

4.2.11 Student's semesters

Functionality

- **List**: Provide a list of all semesters that a particular student has enrolled in.
- **Maintain** the list of semesters for the student.
- **Auditing**: All changes to student semesters are logged.

Fields

Student's semesters			
Field	**Description**	**Type**	**Possible Values**
Student	The student that signed up a semester	Text / ID	All the students
Semester Code	The semester that the student signed up	Text / ID	All semesters
Study Program	The study program that the student is enrolled in	Text / ID	All programs
Quota Price	The quota price for the student for this semester	Text / ID	All quota prices
Fulltime	Fulltime/Part-time	Boolean	0,1
Verified	Whether the semester is verified or not	Boolean	0,1
Verification Date	The date of verification of the semester	Date	
Semester	The order number of semester for the student	Number	1,2,3,4…

Business rules and access rights

- Students can sign in for a new semester
- Students can apply for changes in study program or style of studies (fulltime) when signing in a semester

- Student services officers and/or vice deans (for their faculty only) can accept or deny student requests after being notified.
- Student services officers (for their faculty only) can read and change all data.
- It is assumed that each student upon enrollment gets an account in the Library system. When the student applies for signing up a semester, the system requires information from the library module if the student has returned the books borrowed more than a year before the date of semester sign up.

4.2.12 Exam terms

Functionality

- **List**: Provide a list of all exams terms.
- **Search** for exam terms according to names, dates, cycle.
- **Maintain** the list of terms.
- **Auditing**: All changes to exam terms are logged.

Fields

Exam terms			
Field	**Description**	**Type**	**Possible Values**
Name	Name of the term	Text	
Start Date	Starting date of the term	Date	
End Date	Ending date of the term	Date	
Faculty	The faculty that the term is associated to	Text / ID	Stored list in the database

Business rules and access rights

- Student services officers and/or vice deans (for their faculty only) create and change exam terms.

4.2.13 Student's courses

Functionality

- **List**: Provide a list of all courses that a student has chosen to follow.
- **Search** for courses according to names, semester, etc.
- **Maintain** the list of courses.
- **Auditing**: All changes to courses are logged.

Fields

Student's courses			
Field	**Description**	**Type**	**Possible Values**
Student	The student that signed up for a course	Text / ID	All the students
Course	The course that the student signed up	Text / ID	All semesters
School year		Integer	2010, 2011, …
Study year		Integer $\in$ {0, 1, 2, 3, 4, 5}	

Business rules and access rights

- The student selects and chooses the course he/she is going to follow in the next semester
- Constraints in the selection of courses per student include: (Courses are limited to the ones tied to the study program the student is enrolled in; There is a 35 (40) credits sum per semester limitation for students. Breach of credits sum must be confirmed by vice deans; Connections and dependences among courses must be respected; The

courses are offered ordered by semesters; Failed courses must be signed up again. Change in failed elective courses must be accepted by student services officers ; Payments for the selected courses and course changes must be in place; Periodical limitations for the edit-ability of the selections must be maintained; and For taking a course without prerequisites, the teacher and vice dean must authorize)
- Student services officers (for their faculty only) can change data at any time according to student requests.

4.2.14 Student's exam applications

Functionality

- **List**: Provide a list of all exam applications that a student has made.
- **Search** for exam applications according to semester, course, student, teacher etc.
- **Maintain** the list of exam applications.
- **Auditing**: All changes to exam applications are logged.

Fields

Student's exam applications			
Field	**Description**	**Type**	**Possible Values**
Student	The student that applied for an exam	Text / ID	All students
Course	The course that the student applied to pass	Text / ID	All courses of the student
Grade	The grade the student received	Number	5,6,7,8,9,10
Teacher	The quota price for the student for this semester	Text / ID	All teachers
Exam date	Date of the exam	Date	
Partial	Partial exam, or full	Boolean	0,1

Business rules and access rights

- The student applies for an exams he/she is going to take in the particular exam term
- Constraints in the exam applications include: Courses are limited to the ones the student has taken in the previous two semesters; Periodical limitations for the edit-ability of the exam applications must be maintained; and Students cannot enter grades for their exam applications.
- Teachers can update grades for their courses and their students after the exam has taken place
- Student services officers (for their faculty only) can change data at any time according to student requests or teachers reports.
- Fast user interface must obtained for functionalities like this one, where massive amount of data should be processed
- Bulk insert of exam results vie excel import with a standardized template must be provided.
- The system should allow the cancellation of exams for the head of student affairs.
- The system can print reports completely filled with data, so teachers will sign only the application to make it valid.

4.2.15 Exam schedules

Functionality

- **List**: Provide a list of all exam schedules in an exam term.
- **Search** for exam schedules according to course, teacher etc.
- **Maintain** the list of exam schedules.
- **Auditing**: All changes to exam schedules are logged.

Fields

Exam Schedules			
Field	**Description**	**Type**	**Possible Values**
Exam term	The exam term that the exam takes place in	Text / ID	All terms
Course	The course that the student applied to pass	Text / ID	All active courses
Exam date	Date of the exam	Date	
Places	The rooms the exam takes place	List	List stored in the database

Business rules and access rights

- Student services officers (for their faculty only) can change data at any time.

4.2.16 Course prerequisites

Functionality

- **List**: Provide a list of all course connections and prerequisites.
- **Search** for course prerequisites according to course name.
- **Maintain** the list of course connections and prerequisites.
- **Auditing**: All changes to course connections and prerequisites are logged.

Fields

Course dependancies			
Field	**Description**	**Type**	**Possible Values**
Prerequisite course	The course that is a prerequisite for the course in question	Text / ID	All active courses
For course	The course whose prerequisite is stated	Text / ID	All active courses
Passed/listened	Whether the dependency demands for passed or only listened prerequisite course	List	

Business rules and access rights

- Student services officers and/or vice deans (for each faculty) can insert and update course dependencies.

4.2.17 Master thesis

Functionality

- **List**: Provide a list of all master thesis submitted.
- **Search** for master thesis according to students, mentors, courses, names.
- **Maintain** the list of master thesis.
- **Auditing**: All changes to master thesis are logged.

Fields

Master thesis			
Field	**Description**	**Type**	**Possible Values**
Thesis Number	Code of the thesis	Text	
Name	Name of the thesis	Text	
Description	Description of the thesis	List	
Student	The student that applied with the master thesis	Text / ID	All students
Mentor	The mentor of the thesis	Text / ID	All teachers
Commission President	The president of the committee for thesis presentation	Text / ID	All teachers
Member	Member of the committee for thesis presentation	Text / ID	All teachers
Credits	ECTS credits earned with the thesis	Text	
Grade	Grade awarded	Number	
Application Date	Date of thesis application	Date	
Submission Date	Date of thesis submission	Date	
Presentation Date	Date of thesis presentation	Date	
Supplement	Code of diploma supplement	Text	
Supplement Date	Date of diploma supplement	Date	
Faculty	The faculty that the thesis is associated to	Text / ID	Stored list in the database

Business rules and access rights

- Student services officers can insert and update data.
- Students insert data for their master thesis
- The mentor selected must confirm and accept the thesis.

4.2.18 Diploma thesis

Functionality

- **List**: Provide a list of all diploma theses submitted.
- **Search** for diploma thesis according to students, mentors, courses, names.
- **Maintain** the list of diploma theses.
- **Auditing**: All changes to diploma theses are logged.

Fields

Diploma thesis			
Field	**Description**	**Type**	**Possible Values**
Thesis Number	Code of the thesis	Text	
Name	Name of the thesis	Text	
Description	Description of the thesis	List	
Student	The student that applied with the master thesis	Text / ID	All students
Course	The course that the thesis is associated with	Text / ID	
Mentor	The mentor of the thesis	Text / ID	All teachers
Commission member 1	Member of the committee for thesis presentation	Text / ID	All teachers
Commission member 2	Member of the committee for thesis presentation	Text / ID	All teachers
Credits	ECTS credits earned with the thesis	Text	
Grade	Grade awarded	Number	
Application Date	Date of thesis application	Date	
Submission Date	Date of thesis submission	Date	
Presentation Date	Date of thesis presentation	Date	
Supplement	Code of diploma supplement	Text	
Supplement Date	Date of diploma supplement	Date	
Faculty	The faculty that the thesis is associated to	Text / ID	Stored list in the database

Business rules and access rights

- Student services officers can insert and update data.
- Students insert data for their diploma thesis
- Constraints exist when applying for a diploma thesis (Selection of courses/mentors is limited to the ones taken by the student during studies; and The credits sum must be sufficiently big for a diploma thesis to be applied (the number of credits is configurable per faculty / study program)
- The mentor selected must confirm and accept the thesis.
- Bulk import should be provided for deployment purposes.
- When the student applies for defending the diploma thesis, the system requires information from the library module if the student has returned all the books borrowed.
- The system sends information to the Library module that the student has graduated/ left the studies in order to deactivate its account.

4.2.19 Seminar thesis

Functionality

- **List**: Provide a list of all seminar theses submitted.
- **Search** for seminar thesis according to students, mentors, courses, names.
- **Maintain** the list of seminar theses.
- **Auditing**: All changes to seminar theses are logged.

Fields

Seminar thesis			
Field	**Description**	**Type**	**Possible Values**
ThesisNo	Code of the thesis	Text	
Name	Name of the thesis	Text	
Description	Description of the thesis	List	
Student	The student that applied with the master thesis	Text / ID	All students
Course	The course that the thesis is associated with	Text / ID	Student's courses
Mentor	The mentor of the thesis	Text / ID	All teachers
Credits	ECTS credits earned with the thesis	Text	
Grade	Grade awarded	Number	
Application Date	Date of thesis application	Date	
Submission Date	Date of thesis submission	Date	
Presentation Date	Date of thesis presentation	Date	
Faculty	The faculty that the term is associated to	Text / ID	Stored list in the database

Business rules and access rights

- Student services officers can insert and update data.
- Students insert data for their seminar thesis
- Constraints exist when applying for a seminar thesis (Selection of courses/mentors is limited to the ones taken by the student during studies)
- The mentor selected must confirm and accept the thesis.

4.2.20 Class schedule

This entity contains data about class schedules.

Functionality

- **List**: Display the schedule in various formats: (Schedule for professor; Schedule for student; Schedule for study program; and Schedule for student group).
- **Collisions**: Display a list of all collisions for a schedule e.g. two classes are scheduled at the same time in the same classroom, a professor is scheduled to have two classes at the same time with two different student groups.
- **Search**: Search for classes by multiple criteria.
- **Add/Modify**: SSS, professors or other users with access rights can modify.
- **Auditing**: All changes to schedule are logged.

Fields

Field	Description	Type	Possible Values
School year		Integer	2010, 2011, …
Study year		Integer ∈ {0, 1, 2, 3, 4, 5}	
Semester		Selection	Summer/Winter
Campus	Campus where lecture is held	ID	Master list of campuses
Faculty		ID	Master list of faculties
Study program		ID	Master list of study groups
Course		ID	Master list of courses
Professor		ID	Master list of teachers
Student Group	Students can be grouped in multiple groups.	Text	
Day	Day of the week	Int ∈ {0, 1, 2, 3, 4, 5, 6, 7}	
StartTime	Start time of class	Time	
Classroom	Classroom where class is held.	ID	Master list of classrooms
Status	Status of schedule	{InProgress, Locked}	

Business rules and access rights

- A student can view his schedule.
- SSS or other users with access rights can modify the schedule.
- Professor can modify the schedule for his courses.
- If a schedule modification results in a collision (e.g. two classes at the same time in the same classroom), then a corresponding alarm is shown to the user.
- SSS or other users with access rights can lock the schedule.
- A schedule can be modified if schedule's status is *InProgress*. If its status is *Locked*, then it cannot be modified.
- Upon locking a schedule, a notification containing the locked schedule is sent to the professor, all students registered in the student group, and SAO staff.

4.2.21 Student groups

This entity contains data about student groups. Students attend classes in groups. According to the law, the maximum size of a student group is predetermined. It depends on the type of classes: lectures, lab exercises, tutoring etc. Additionally, the group size can be limited by the capacity of the classrooms. For elective subjects, the number and size of groups can also greatly vary depending on the popularity of courses. The system should provide automatic and manual distribution of students in predefined study groups.

Functionality

- **List**: Display the student group with all its attributes and students
- **Search**: Search for student groups by multiple criteria: campus, faculty, course, etc.
- **Add/Modify**: Professors can modify student groups.
- **Auditing**: All changes to student groups are logged.

Fields

Field	Description	Type	Possible Values
School year		Integer	2010, 2011, …
Study year		Integer ∈ {0, 1, 2, 3, 4, 5}	
Semester		Selection	Summer/Winter
Campus	Campus	ID	Master list of campuses
Faculty		ID	Master list of faculties
Study program		ID	Master list of study groups
Course		ID	Master list of courses
{Student}	Students that make up the group.	ID	Master list of students

Business rules and access rights

- A student can view the groups he belongs to.
- Professor can view and modify the student groups for his courses.

4.2.22 Faculties

Functionality

- **List**: Provide a list of all faculties in the university.
- **Search** for faculties according to name.
- **Maintain** the list of faculties.
- **Auditing**: All changes to faculties are logged.

Fields

Faculties			
Field	**Description**	**Type**	**Possible Values**
Faculty name	The name of the faculty	Text	
Short name	Short name of the faculty	Text	
Address	Address of the faculty	Text	
Web URL	Website address	Text	

Business rules and access rights

- Administrator can insert and update all data.

4.2.23 Students sign-outs

Functionality

- **List**: Provide a list of all student sign-outs.
- **Search** for signed out students.
- **Maintain** the list of signed out students.
- **Auditing**: All changes to students are logged.

Fields

Student sign-outs			
Field	**Description**	**Type**	**Possible Values**
Student	The student that has signed out	Text / ID	All students
Date	The date the student has left the university	Date	
Document number	Code of the document	Text	

Business rules and access rights

- Students initiate the procedure for leaving the university
- Student services officers can insert and update data.
- The system checks the library module for any books borrowed by the student to prevent leaving the university if the student has not returned all books.

4.2.24 Student remarks

Functionality

- **List**: Provide a list of all remarks for students.
- **Search** for remarks for students.
- **Maintain** the list of remarks for students.
- **Auditing**: All changes to remarks for students are logged.

Fields

Remarks for Students			
Field	**Description**	**Type**	**Possible Values**
Student	The student that the remark is for	Text / ID	All students
Date	Timestamp	Date	
Description	Description of the remark for the student	Text	

Business rules and access rights

Remarks are entered for students in special cases (when caught cheating, when special discounts are approved etc).

4.2.25 Student europass CV info

Functionality

- **View/Update** personal file information in Europass CV
- **Access** reports and notifications published by university staff
- **Pay fees** of diverse type
- Request diploma/diploma supplement/confirmation letters/certificate to be issued

Field	Description	Type
Student ID	Student index assigned by the University. It contains number + string which shows the faculty he/she is enrolled.	String [80]
{Paid Fee}	Fees paid by the student	Structure of type Paid Fee
Year of studies	Shows the year of current studies, or graduated if the student has graduated (Alumni)	String[30]
Average GPA	The current GPA, till the last semester	Int
Cumulative GPA	The current GPA, till the last semester	Int
Skills	(Editable) Students can add numerous skills they posses, depending on the Faculty. This information is intended to facilitate the process of internship assignment by the Faculty staff as well as potential job offers.	String
Certificates	(Editable) Students state the certificates they hold. It also enables them to input new certificates gained during their studies.	String

Business rules and access rights

- A student can view/edit/delete data of his/her personal file. Not all information is available to be edited. The editable sections are marked “editable” in the above table.
- A student can apply on-line for partial/final exam.
- A student can view reports published/created by university staff.
- A student can request a diploma/diploma supplement/confirmation letters/certificate issuing by the Staff of the Administration Office.

4.2.26 Faculty staff

Functionality

- **View** student’s Europass CV
- **Assign** activities to student. The system should send confirmation e-mail to faculty staff and notification email to the student.
- **Mark** activities as completed.
- Create/edit/delete reports.
- **Course-notifications** should be able to be sent by faculty staff. These notifications concern only a group of students enrolled/registered to the course taught by that faculty staff.

Field	Type	Description
ID	String [80]	
Name	String [80]	
Surname	String [80]	
Faculty ID	String [80] FK	FK to the table of Faculty

Business rules and access rights

- Faculty staff should be able to create exam reports by filling in the exam report template. Faculty staff should update and delete only the reports created by his/her user account.

- Faculty staff can publish exam reports, class schedules, exam schedules, and any kind of notification
- Faculty staff should manage notifications in the same manner as exam reports.
- Faculty staff should only have privileges to edit activities assigned by them.
- Faculty staff should be able to view students' file only of the students from the same faculty or to students that take some of his/her courses. Faculty staff should be able to enter Student ID or to search through the students in the faculties he/she teaches courses. Search criterions are the following: Index no., First name, Surname, Faculty. Faculty staff should only view curriculum-related activities. No fees or additional activities should be visible.
- Faculty staff should be able to see Average and Cumulative GPA rankings of students in his/her Faculty.
- Faculty should be able to fill in the electronic exam application of the student with the appropriate details and sign it with his/her digital signature.

4.2.27 Staff in the Administration Office

Functionality

- **View/edit** student personal file.
- **Send notifications** to individual students.
- Create/edit/delete notifications.
- **Diploma/confirmation letters/certificate issuing** should be handled by the Administration Office staff.

Business rules and access rights

- Staff from the Administration Office should be able to enter Student ID info or to search through the students in the faculties. Search criterions are the following: Index no., First name, Surname, Faculty. The Administration Office should have the privilege to edit all the personal info of the student including his skills and certificates.
- Administration Office should have access to the reports and notifications published by the University Staff. However they should only be able to create notifications concerning the students or the University.
- Administration Office should be able to edit the notifications created by them and delete the same. They can publish only notifications**.**
- Administration Office should fill in the diploma/ diploma supplement/ certificate/ confirmation letters templates upon student's request through the system. After a student has successfully requested a diploma/diploma supplement/confirmation letters/certificate issuing, the Administration Office staff should start the procedure of the issuing. After the documents are ready, the Administration Staff should contact the student through email/phone when come and collect them.
- Administration Office should be able to send notifications to individual students when accessing their personal student file.

4.2.28 Reports

Functionality

- Show reports
- **Search** should be enabled for all users of the reports module.

Field	Type	Description
ID	Int, Identity	
Title	String [80]	
Type	String [80]	Exam report, exam schedule, class schedule, notification
faculty ID	String [80], FK	FK to table Faculty. Faculty which it concerns. Null if it's relevant for whole university
staff ID	String [80], FK	FK to table Staff and Administration Staff. University staff creating the report/notification
date valid	String [200]	Date of validity, after this the report is no longer visible in the reports module.

Business rules and access rights

- Show all the reports created by the University Staff. Reports that have passed their validity date should no longer be available. They should be kept in the database for one year.
- Edit/delete operations, before the validity period has ended, on reports should be enabled only to the user who has created the report.
- Search should be allowed by type of report: exam report, exam schedules, class schedules, notifications; or by faculty.

4.2.29 Activities and fees (Activities)

Functionality

- **Online completion** of the student activities during the studies
- Activity assignment through the system.
- **Restrictions** based on the activities' completion

Field	Type	Description
Name of activity	String [30]	
completed	Boolean	Status of the activity

Field	Type	Description
Name of activity	String [30]	Fees is a subclass of Activity
Completed	Boolean	Status of the activity
Value	Int	Value in denars of the fee
faculty ID	String [20], FK	FK to the table Faculty. Faculty for which the fee applies. Additional fees are diverse for different faculties at the university

Business rules and access rights

- Students should complete all their activities to be able to require a diploma/diploma supplement issuing.
- Students should not be able to use the other modules of the system if they have not completed the basic activities (fees, register a semester) for the following semester.
- Activity assignment should be enabled only to university staff with the business rules described in the previous sections.
- Available fees should always be checked before a student is allowed to make an exam application.
- The activity status should only be changed by the system or the faculty staff that has created the activity.

4.2.30 Activities and Fees (Fees)

Functionality

- **Online completion** of the student activities.
- Activity assignment through the system.
- **Restrictions** based on the activities' completion

Field	Type	Description
Name of activity	String [30]	
Completed	Boolean	Status of the activity

Field	Type	Description
Name of activity	String [30]	Fees is a subclass of Activity
Completed	Boolean	Status of the activity
Value	Int	Value in denars of the fee
faculty ID	String [20], FK	FK to table Faculty. Faculty for which the fee applies. Additional fees are diverse for different faculties at the university

Business rules and access rights

- Students should complete all their activities to be able to require a diploma/diploma supplement issuing.
- Students should not be able to use the other modules of the system if they have not completed the basic activities (fees, register a semester) for the following semester.
- Activity assignment should be enabled only to university staff with the business rules described in the previous sections.
- Available fees should always be checked before a student is allowed to make an exam application.
- The activity status should only be changed by the system or the faculty staff that has created the activity.

4.3 Migration from legacy systems

There are legacy applications functioning in all universities participating in the project. Furthermore, there are different legacy applications operating on various faculties at some of the universities. Certain portion of existing data must be migrated to the new application. Previous records for students, student's semesters, exams passed, semester payments, diploma thesis, seminar thesis, courses, study programs, master theses, quota prices per year per study program, faculty staff, student sign outs, student groups and other data must be migrated. The migration should be performed with bulk inserting functionalities implemented in every form of the software application.

For bulk inserting of more records for migration purposes or similar activities, an Excel file and CSV (comma separated file) import should be provided for every data model described in the document or defined in the database. Excel and CSV templates should be defined with columns matching the ones in the data models described in the document and implemented in the database. The data can be extracted from legacy applications, prepared in the Excel or CSV format of the template provided, with column data matching the data defined in the database. Then the officers could go to forms where the Excel or CSV files can be browsed to, selected and uploaded to the server, then their contents imported directly in the students database in the appropriate database tables. The excel templates should strictly define allowed values per columns, matching existing data in the database tables (available cities of birth, available courses for exams passed matched by code and/or name etc.). Migrated records should be flagged differently in the database, and a log should be kept on the user that performed the migration and the date of migration.

The forms for browsing and upload/import of Excel or CSV files should provide filters for faculties or other parameters that could / should be set before migrating the data in question, if the data definition of the template itself does not contain sufficient information.

4.4 Master data

4.4.1 General overview

Master data consists of the following lists:

- List of faculties
- List of classrooms
- List of laboratories
- List of contract types
- List of document types
- List of fees payable by applicants
- List of High schools
- List of places
- List of cities
- List of municipalities
- List of ethnicities
- List of countries
- List of subject types (mandatory, elective, facultative …)
- List of scientific areas
- List of exam types
- List of system tasks (includes menu options, functions etc.)

This data changes very rarely. Still, there is a need for the data to be maintainable.

The list is maintainable by the System Administrator, as depicted in the corresponding use case diagram (section 3.1.2):

4.4.2 Code tables

Ethnicity:	
1.	Македонец
2.	Албанец
3.	Турчин
4.	Ром
5.	Србин
6.	Бошњак
7.	Влав
8.	Друго

Type of high-school / occupational profile	
01	гимназиско образование
02	електротехничка струка
03	машинска струка
04	сообраќајна струка
05	хемиско-технолошка струка
06	графичка струка
07	геолошко-рударска и металуршка струка
08	градежно-геодетска струка
09	економско-правна и трговска струка
10	здравствена струка
11	земјоделско-ветеринарна струка
12	текстилно-кожарска струка
13	угостителско-туристичка струка
14	уметничко образование
15	лични услуги
16	шумарско-дрвопреработувачка струка

List od municipalities in Macedonia	
01	Аеродром
02	Арачиново
03	Берово
04	Битола
05	Богданци
06	Боговиње
07	Босилово
08	Брвеница
09	Бутел
10	Валандово
11	Василево
12	Вевчани
13	Велес
14	Виница
15	Вранештица
16	Врапчиште
17	Гази Баба
18	Гевгелија
19	Ѓорче Петров
20	Гостивар
21	Град Скопје
22	Градско
23	Дебар
24	Дебарца
25	Делчево
26	Демир Капија
27	Демир Хисар
28	Дојран
29	Долнени
30	Другово
31	Желино
32	Зајас
33	Зелениково
34	Зрновци
35	Илинден
36	Јегуновце
37	Кавадарци
38	Карбинци
39	Карпош
40	Кисела Вода
41	Кичево
42	Конче
43	Кочани
44	Кратово
45	Крива Паланка
46	Кривогаштани
47	Крушево
48	Куманово
49	Липково
50	Лозово

51	Маврово и Ростуша
52	Македонска Каменица
56	Македонски Брод
54	Могила
55	Неготино
56	Новаци
57	Ново Село
58	Осломеј
59	Охрид
60	Петровец
61	Пехчево
62	Пласница
63	Прилеп
64	Пробиштип
65	Радовиш
66	Ранковце
67	Ресен
68	Росоман
69	Сарај
70	Свети Николе
71	Сопиште
72	Старо Нагоричане
73	Струга
74	Струмица
75	Студеничани
76	Теарце
77	Тетово
78	Центар
79	Центар Жупа
80	Чаир
81	Чашка
82	Чешиново
83	Чучер-Сандево
84	Штип
85	Шуто Оризаре

List of high schools in Macedonia	
БЕРОВО	
01	ОСУ „Ацо Русковски“
БИТОЛА	
02	СОУ „Јосип Броз Тито“
03	СОТУ „Ѓорѓи Наумов“
04	СОУ „Таки Даскало“
05	СОЕУ „Јане Сандански“
06	ДСЗУ „Кузман Шапкарев“
07	ОСМУ „Д-р Јован Калаузи“
08	Државно музичко училиште - Битола
БОГДАНЦИ	
93	СОУ „Богданци“
БРОД МАКЕДОНСКИ	
09	СОУ „Св. Наум Охридски“
ВАЛАНДОВО	
10	ОСУ „Гоце Делчев“
ВЕЛЕС	
11	УСО „Кочо Рацин“
12	УСО „Јовче Тесличков“
13	ССОУ „Коле Неделковски“
14	УСО „Димитрија Чуповски“
ВИНИЦА	
15	СОУ „Ванчо Прке“
ГОСТИВАР	
16	СОУ “Гостивар“ (“Панче Попоски”)
17	ОСТУ „Гостивар“ (“Злате Малакоски”)
18	СЕОУ „Гостивар" (“Чеде Филипоски”)
ГЕВГЕЛИЈА	
19	СОУ „Јосиф Јосифовски“
ДЕБАР	
20	ЦСНО „Здравко Чочковски“
ДЕЛЧЕВО	
21	СОУ „Методија М. Брицо“
ДЕМИР ХИСАР	
22	СОУ „Крсте П. Мисирков“
КАВАДАРЦИ	
23	ОСУ „Добри Даскалов“
24	СОУ „Киро Спанџов - Брко“
25	СОЗШУ „Ѓорче Петров“
КИЧЕВО	
26	ОСУ „Мирко Милески“
КОЧАНИ	
27	ОСУ „Љупчо Сантов“
28	СОУ „Гошо Викентиев“
КРАТОВО	
29	СОУ „Митко Пенџуклиски“
КРИВА ПАЛАНКА	
30	СОУ „Ѓорче Петров“

КРУШЕВО	
31	ОСУ „Наум Наумовски - Борче“
КУМАНОВО	
32	СОУГ „Гоце Делчев“
33	ССОУ „Киро Бурназ“
34	СОУ „Перо Наков“
35	ОСТУ „Наце Буѓони“
НЕГОТИНО	
36	СОУ „Св. Кирил и Методиј“
ОХРИД	
37	ОСУ „Св. Климент Охридски“
38	ОЕМУЦ „Св. Наум Охридски“
39	ОУТУ „Ванчо Питошески“
ПРИЛЕП	
40	СОУГ „Мирче Ацев“
41	СОУ „Ѓорче Петров“
42	СОУ „Орде Чопела“
43	СОЕПТУ „Кузман Јосифовски-Питу“
44	СОУ „Ристе Ристески-Ричко“
ПРОБИШТИП	
45	СОУ „Наум Наумовски-Борче“
РАДОВИШ	
46	СОУ „Коста Сусинов“
РЕСЕН	
47	СОУ „Цар Самуил“
СВЕТИ НИКОЛЕ	
48	СОУ „Кочо Рацин“
СКОПЈЕ	
49	СУГСГ „Раде Јовчевски-Корчагин“
50	СУГС „Никола Карев“
51	СОУ „Зеф Љуш Марку“
52	СУГСГ „Јосип Броз Тито“
53	СУГСГ „Орце Николов“
54	СУГС „Кочо Рацин“
55	СУГС „Панче Арсовски“
56	СУГС „Браќа Миладиновци“
57	СУГС „Георги Димитров“
58	СМУГС „Д-р Панче Караѓозов“
59	СЕПУГС „Васил Антевски - Дрен“
60	СЕПУГС „Арсени Јовков“
61	СУГС „Цветан Димов“
62	СУГС „Лазар Танев“
63	СУГС „Марија Кири Склодовска“
64	СГГУ „Здравко Цветковски“
65	СУГС „8-ми Септември“
66	СУГС „Владо Тасевски“
67	СУГС „Димитар Влахов“
68	АСУЦ „Боро Петрушевски“
69	СЕТУГС „Михајло Пупин“
70	ДМБУЦ „Илија Николовски-Луј“

71	ДСУПУ „Лазар Личеноски“
72	ДУФК „Методи Митевски Брицо“
73	ЦОР “Партенија Зографски”
74	ДУРДМОВ “Димитар Влахов”
75	ДСУРО “Св. Наум Охридски”
СТРУГА	
76	СОУ „Нико Нестор“
92	СОУ „Д-р Ибрахим Темо“
СТРУМИЦА	
77	СОУ „Јане Сандански“
78	СОУ „Димитар Влахов“
79	СОУ „Никола Карев“
ТЕТОВО	
80	СОУГ „Кирил Пејчиновиќ“
81	СОМУ „Никола Штејн“
82	СОЕУ „8 Септември“
83	ССОУ „Моша Пијаде“
84	СТОУ „Гоце Стојчески“
85	СМУ „Тодор Скалоски-Тетоец“
ШТИП	
86	СОУ „Славчо Стојменски“
87	СОУ „Коле Нехтенин“
88	СОУ „Димитар Мирасчиев“
89	СОУ „Јане Сандански“
90	ДСМУ „Сергеј Михајлов“
91	ЦССНО „Искра“
ПРИВАТНИ УЧИЛИШТА	
95	ПСУ „Јахја Кемал“
96	ПСУ „Петта приватна гимназија“
97	ПУСО „САБА“
99	**ДРУГИ приватни и странски училишта**

Special circumstances	
0	Normal application (no special circumstances)
1	деца без родители
2	лица со прв степен на инвалидност
3	воени инвалиди
4	лица од дом за напуштени деца
5	припадници на безбедносни сили
6	деца на починати учесници во мировни операции

4.5 Audit Log

4.5.1 Functionality

This software can hold a lot of information, so it requires a proper audit logging method. Most of the updates in the database include several table updates as well as several field updates. The audit log should be accessed via web by several viewpoints:

- **User.** A system may be accessible by many users, so it is important to be able to identify updates by user.
- **Session.** A session covers the time period between a user logging on and logging off the system. This system should not allow a user to have more than one session active at the same time.
- **Transaction.** Within a session, a user processes transactions. This is the same as database transactions which covers all updates between a 'start' and a 'commit' commands.
- **Database Table.** Within a transaction any number of database tables may be modified.
- **Database Field.** Within a database table any number of fields may be modified.

As well as being able to store this information, the design should allow for any details to be viewed in a single screen. These details should make it easy to see exactly what information has changed, preferably showing both the original and newest values.

This method can be implemented with five tables, as described below. The first four tables are concerning the data logging, and the last one concerns the system logon fails.

Log_Session		
Field	**Description**	**Type**
SessionID (PK)	A unique number given to each session as the first set of details are logged	Unique identifier
UserID	User Identity. Reference to a user in the User table	Unique identifier
Date_Time	The date and the time when the record in this table is created	DateTime

Log_Transaction		
Field	**Description**	**Type**
SessionID (PK)	Reference to Log_Session(SessionID)	Unique identifier
TransactionSeqNo (PK)	Transaction Sequence Number. This starts at 1 for each Session. Each time the database is updated - when the user presses the SUBMIT button which initiates a start transaction and ends with a commit - this is treated as a separate database transaction. This may include any number of database inserts, deletions and updates.	Integer
Date_Time	The date and the time when the record in this table is created	DateTime
TaskID	Reference to the task in the system task table (menu option, module function, stored procedure etc…)	Unique identifier

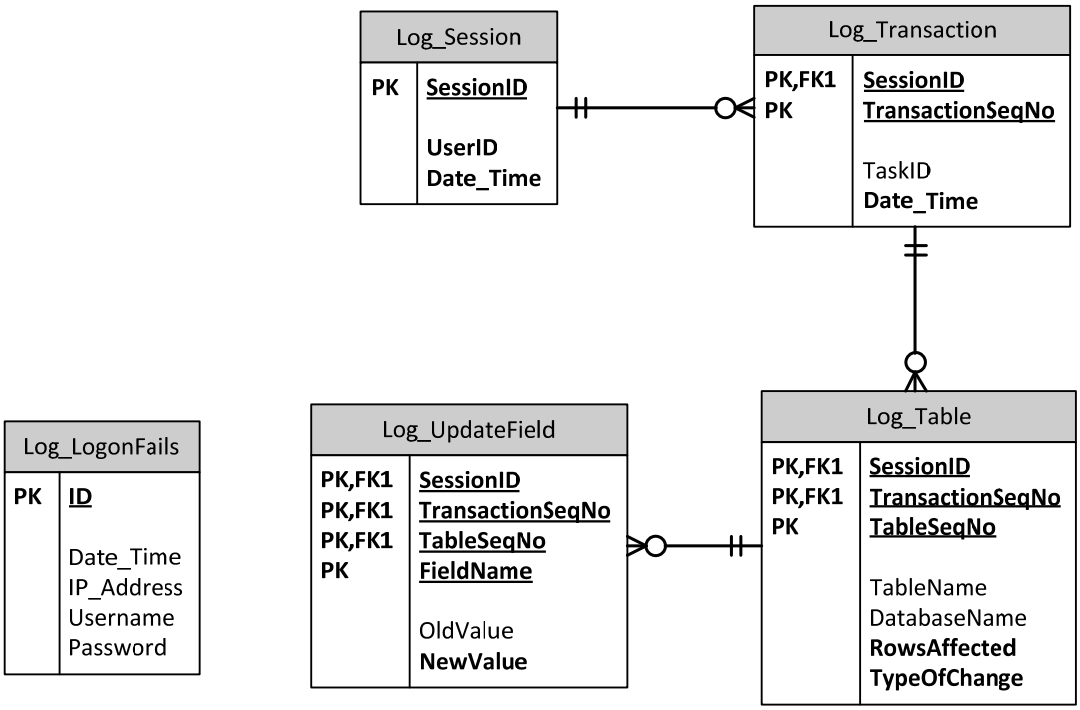
Log_Table		
Field	**Description**	**Type**
SessionID (PK)	Reference to Log_Session(SessionID)	Unique identifier
TransactionSeqNo (PK)	Reference to Log_Transaction(TransactionSeqNo)	Integer
TableSeqNo (PK)	Table Sequence Number. This starts at 1 for each Transaction. There may be changes to several occurrences of the same table, so each occurrence is given its own sequence number.	Integer
TableName	Table Name. The name of the database table being updated.	String
DatabaseName (Optional)	Database name, if the system lies on multiple databases	String
RowsAffected	Identifier of the rows which are affected with this transaction on this table. It is shown in the format of the WHERE clause of an sql SELECT statement (for example: field1='value1' OR field2='value2')	String
TypeOfChange	Type of change (INSERT, UPDATE or DELETE)	String

Log_UpdateField*		
Field	**Description**	**Type**
SessionID (PK)	Reference to Log_Session(SessionID)	Unique identifier
TransactionSeqNo (PK)	Reference to Log_Transaction(TransactionSeqNo)	Integer
TableSeqNo (PK)	Reference to Log_Table(TableSeqNo)	Integer
FieldName (PK)	Field Name which is updated	String
OldValue	The value in this field before the database update.	Text
NewValue	The value in this field after the database update.	Text

* this table (Log_UpdateField) is used only if the type of change is UPDATE. This table keeps trails about the old data.

Log_LogonFails		
Field	**Description**	**Type**
ID (PK)	Identifier.	Unique identifier
Date_Time	Date and time of the fail.	DateTime
IP_Address	IP address of the request which generated the error.	String
UserName	Username part of the input which generated the error.	String
Password	Password part of the input which generated the error.	String

ER diagram



Business rules

- Only four extra log tables are required for any number of application tables in any number of databases.
- Logging for individual tables can be turned on or off
- For each change both the 'old' and 'new' values are recorded.
- Fields which have not changed are not shown.
- There is no potential confusion with null values.
- As there are only four audit tables in a strict hierarchy their contents can easily be viewed online via a single screen, and filtered via a variety of selection criteria.
- As well as being a useful tool in a production environment in order to see who changed what, and when, it is also a useful tool in a development environment as it can instantly show how the database was changed after each user transaction.

4.5.2 Viewing the Log

As there are only a small number of tables to hold the audit log details the contents can be viewed with a few simple screens:

- **Search Log Details** – The purpose of this function is to allow the user to enter selection criteria before retrieving entries from the Log_Field table
- **List Log Details** – The purpose of this form is to allow the user to browse through the contents of the Log_Field table. This screen is divided into 2 parts: the top part shows who made the change and when it was made, and the bottom half shows which fields/items/columns were actually changed.
- **List Log Details for an Object** – The purpose of this form is to allow the user to browse through the contents of the Log_Field table. This screen is divided into 3 parts: the top part shows which object was selected, the middle part shows who made the change and when it was made and the bottom half shows which fields/items/columns were actually changed.
- **List Logon Errors** – The purpose of this form is to allow the user to browse through the contents of the Log_LogonFails table. This will contain a record of every logon attempt which failed.

5 User interface

5.1 General requirements

The user interface should be user friendly and intuitive. It should be AJAX based providing for fast response and always updated data. AJAX should be used for filtering lists whose content is dependent from other selections on the same form. (For instance, when a faculty from the university is chosen in a list, the next list containing study programs should immediately filter to only show programs for the faculty selected. Furthermore, when a program is chosen in the second list, a third list showing courses should filter out only courses related to the program chosen). AJAX should be used for real-time updates of data (For instance, as courses are chosen for signing up, the sum of credits and amount to be paid for the semester should be recalculated in real-time).

The layout of the contents in the form should be optimized for fast and intuitive work. The layout should be stable and consistent in all popular browsers (Firefox, Internet Explorer, Safari, Chrome). The layout of contents on the forms should be optimized for 17'' monitors or higher. Efforts should be made to enable most of the work to be done without scrolling the forms.

The size of the lists should ne predetermined according to expected number of records, and a scroll bar should appear if the average number of records is exceeded. Javascript and tab indexes should be used for supporting keyboard work and form submission by pressing "Enter".

Client and server side validators should be implemented in order to keep data consistent and avoid errors. The validator's messages should be noticeable, clearly visible and descriptive, providing for fast correction. Submission of incorrect forms should be disabled. Progress bars and modal pop-ups should be visible when form submission is delayed or prolonged due to slower network connections or server side delays.

The selection between dropdown lists, list boxes or check box lists is context based, providing for most optimal, fast and robust work.

Application is expected to follow the good design practices for web user interface such as [14]. Following is a suggested list of guidelines:

5.1.1 Validation

There are several options for handling validation errors:

- Raise an exception. This might not be a useful action in the presentation layer. However, application might rise (or log) an exception if one of the validator controls detects some sort of attack.
- Display individual error messages, indicating the reason for the validation error and allowing the user to re-enter data accordingly.
- Use the ValidationSummary control to display a summary of all validation errors in a single location.
- Whatever action application performs when validation fails, application must make sure that the user is clearly notified about what is wrong and is given an opportunity to correct the data entry.

When the user clicks to "Save" data, any field that is bound to validation logic, for which the validation logic fails (for example: no data has been provided for a "Required" field) will show a red "Validation" message at the right side of that control. Once data are entered and the focus is changed to other controls that red "Validation" message will disappear. Here are the common validation types and messages that will be used in the application:

- Required: "Required"
- Email address format: "Valid email address required"
- Date format: "Not a valid date"
- Number: "Not a number"

- Fractional value: “Enter a valid number or value”

The following messages should be included “*” for required fields, and also a description message text “* indicates required fields” at the top right of the UI panel.

When the user exceeds the recommended length of text, a message box will be shown automatically and no characters can be entered.

A validation for the EMBG field should be provided (for details on checksum calculation and EMBG structure see: http://en.wikipedia.org/wiki/Unique_Master_Citizen_Number). After entering the EMBG the date and place of birth and sex can be automatically extracted and the corresponding fields in the form should be populated. The system should perform this without post-back to the server.

5.1.2 Control focus

The following rules apply for control focus:

- When a form is loaded, the focus will be on the first control.
- There will be a default submit button in each web form.

5.1.3 Grid

The lists will be generated on a grid. The following rules should apply:

- Include “Delete Confirmation Box”
- Include Paging.
- Include Sorting. When the application retrieves a large amount of data, data paging techniques should be used to avoid scalability problems. Application should not retrieve more data than required at any one time. For example, if a screen displays 1,000 rows of data in a grid with 20 rows per page, then the application should implement data retrieval logic that retrieves 20 rows at a time.
- Enable filtering for all displayed columns.
- For action buttons use text/ image.

5.1.4 Control to access a list of predefined values

The fields that have a limited set of predetermined values should be presented as:

- A drop down list if the number of values is greater than 5.
- A set of radio buttons if the number of values is less than 4.

5.1.5 Preview of details page

The details info of a record of tabular data can be viewed and updated in any of the following ways:

- **Grid**: Data can be updated on the grid. Include save, cancel, and delete button
- **Preview details panel**: Details content will be placed on the same page and can be updated.
- **Pop-up page**: Content for the selected record will be shown in a pop-up page.
- **Multiple pages**: Include “Show” button to go to another page showing details for the selected record

5.1.6 Error page

If an error occurs at the web server while processing user requests, an error page will be shown with user friendly and informative text.

5.1.7 File attachments

Maximum file size should be mentioned in the file upload page.

5.1.8 Consistency

Consistency of UI should be achieved in all web pages. It is suggested to use CSS at the very minimum, and also other mechanisms specific to the technology used e.g. Themes, Master pages, Skins, Navigation controls in ASP.NET.

5.1.9 Data formatting

Formatting of all data types (number, currency, date, time etc.) must follow the regional settings in Macedonia.

5.1.10 Help system and tips assistance for filling forms

On every screen a clearly visible link to the help subsystem that explains the procedure and what is expected by the user. Next to every field the applicant is expected to fill, an image (with question mark for example) that displays as a floating window (or in a dedicated part of the screen) explanation of the field and what is expected by the applicant.

Tips (text messages to appear when mouse is positioned on data items) should be provided for data field identifications in forms and grids.

5.2 Authentication and identification

The following functionalities are specified in this system:

- Registration of new users
- Logging into the system
- Retrieving forgotten password
- Changing a password
- Reporting a problem without logging into the system

5.2.1 Overall functional requirements

This section describes the requirements for the following five functionalities:

- Register on the system – The user registers on the system by providing valid e-mail address, choosing a password (entered twice for verification) and entering a security code (captcha). Links to the “Terms of Use” and “Privacy Policy” documents should be present on this screen. The system checks if the e-mail address is already registered on the system and if it is not sends an email to the new user’s address with activation code. The university user account is activated after the user enters the activation code (send as a hyperlink) in the e-mail message.
- The start page of the system is the login screen. A registered user can log in on the system by entering his e-mail and password. A link to the registration form and the password recovery form should also be present on the initial screen.
- For retrieving forgotten password the user should be presented with a form to enter his e-mail address. By submitting the e-mail the system will reset the password for the corresponding user to a random password and send it to the user’s e-mail.
- The user should be able to change its present password when logged on to the system by editing his profile
- The system should provide a way for a user to report a problem with the system without logging to the system by sending an e-mail or filling and submitting an on-line form describing the problem (verified by captcha to avoid robots).

5.2.2 Registration

Description and priority

Students and university staff will carry out their own registration using the features of the module Student Identification and Accounting System. This will enable the system to display personalized information when the user logs in and certain information. Giving each user a

unique ID will allow the system to associate the user with the data entered in the system associated to their personal file and privileges.

On the initial visit to the system the students will be offered to create a more detailed profile with their preferences within the system. After identifying with their e-mail address and choosing a password (and verifying by captcha) an account activation e-mail message will be sent to the student.

Stimulus/Response Sequences

The following information is to be obtained in order to register an account:

- Name
- Surname
- University e-mail address

Functional Requirements

Activation – First time users should first activate their account by logging in with their university e-mail address and input a password.

Password generation – The system should provide automatic password generation for registered account.

Registration – The system should give access to students to their personal file, reports, requests module and e-learning centre when provided correct account information, while university staff to wider set of operations.

Forgotten password - The system should provide a mechanism for resetting / retrieving a forgotten password.

5.2.3 Profile configuration

Description and priority

The candidate should be able to configure:

- the user profile on the system regarding the interface language,
- the channel(s) used for notification (e-mail, SMS …)
- the notifications to be sent by the system on different evens (submission has started, list of candidates has been published …).

5.3 Administrator functions

5.3.1 Administration of students data

Description and priority

The form will consist of text fields and drop down lists showing all values in the data model of this functionality. The fields will be grouped by topic (previous education, birth information, initial enrollment data, parents etc). AJAX should be used to update dependent related fields/lists when one of the fields is changed. Insert/Update buttons should be provided.

Stimulus/Response Sequences

Students should be searched through with filters by index, name, surname, social security number, cycle of studies, study program, start year. The form should present all personal data of the student grouped by topic, texts in text boxes and selections in drop down lists.

Functional Requirements

- Create/edit/delete students
- Send notifications/e-mails to a single student

5.3.2 Administration of students enrolment details

Description and priority

The form will consist of text fields and drop down lists showing all values in the data model of this functionality. The fields will be grouped by topic (previous education, birth information, initial enrolment data, parents etc). AJAX should be used to update dependent related fields/lists when one of the fields is changed. Insert/Update buttons should be provided.

Stimulus/Response Sequences

Students should be searched through with filters by index, name, surname, social security number, cycle of studies, study program, start year. The form should present all personal data of the student retrieved through enrolment grouped by topic, texts in text boxes and selections in drop down lists

Functional Requirements

- Create/edit/delete students
- Upload/Download data for students at enrolment
- Send notifications/e-mails to a student

5.3.3 Administration of the student semesters

Description and priority

The form should provide a table (grid) with previous semesters signed in by the student, along with details for each semester (quota, prices, study program, status, financial obligations uncompleted etc). There should be an option to sign up a new semester, or correct parameters in previous semesters (depending on the user privileges). When students sign up, if they change parameters that are not automatically allowed, then a request for confirmation should be sent to clerks and/or vice deans.

Stimulus/Response Sequences

After the student is searched and found through filters, or is preselected, parameters for the semester to be signed in should be chosen (semester year – fall/spring, quota, price, study program, status – regular/part-time). Eventually the semester is signed up for the student and conditions are created for courses to be selected for the signed in semester.

Functional Requirements

- Create/edit/delete semesters for a student

5.3.4 Administration of students courses

Description and priority

The form should provide a list (checkboxes or / and multi-select list) of available allowed courses (both mandatory and optional) that the student can enlist in. The student or clerk should be able to select from the given courses. Constraints should be implemented. Failed previous courses should be preselected. Credits and semester costs should be automatically calculated during courses selection. Insert / update functionality should be provided. The access to the form by students is periodical and limited only during semester signup periods. Clerks in student services department have no limitations. Courses are presented in chronological order according to the way they are planned to be taught in the study program.

Stimulus/Response Sequences

After the student is selected/filtered, a set of courses should be chosen for him/by him for his/hers last enrolled semester. After the list of mandatory/optional courses is created the submission of the form will enlist the courses for the student for the semester in question. The sum of credits is calculated in real-time as courses are selected, along with the required amount to be paid for the semester. Constraints for the maximum number of credits in a semester should be implemented strictly when students access the form, and only as a notification for clerks, because they can override the constraint is special cases for special students under defined circumstances.

Functional Requirements

- Create/edit/delete enlisted courses for a student
- Print prepared invoice for payment of a semester

5.3.5 Administration of student exam applications

Description and priority

The form should provide a list of exams that the student can apply to. The list can be in form of a list of checkboxes or a multi-select list box. The student can, in a limited time window, change the selection of exams that he/she is applying for. Clerks have no time limitations. In some cases, the student can choose from a list of teachers for each exam, if there are more teachers for that exam (if that information is not connected to the group the student is part of and therefore extracted from there).

Stimulus/Response Sequences

The student/or a clerk in the name of the student is given a list of courses available for exam application (a list of courses enlisted in the past two semesters). The user selects the courses and by submitting the form applies for the exam. Before the exam application deadline, the user can log in and change previous exam applications. After the deadline has passed, only clerks can change exam applications. Students applying for exams after the deadline are ought to pay a fine.

Functional Requirements

- Create/edit/delete exam applications

5.3.6 Administration of student's exams

Description and priority

In the form for exam applications per student, when viewed by clerks in student services department, an option for inserting a grade for the exam should be visible, for special occasions.

Multiple grades for more students can be inserted by the teacher or a clerk, in a form that enables listing of all students that have applied for the exam for that exam term. The form presents a list of names with dropdown lists with grades beside each name.

There should also be a form where an excel template filled with students and grades can be uploaded for a particular exam in an exam term, for bulk insert.

Also a form should be enabled where an exam can be selected, an exam term, and a date, and the teacher can insert an index for a student, press Enter to see the name of the student in question (via AJAX technology), then enter the grade in the next field, then press Enter again to store the grade, and move back to the field for entering a new index. All previously inserted students are shown on a list below.

A form should be provided with a list of all passed exams by the student, and all courses taken so far. Beside the name of every non-passed courses taken, there should be an option to insert a grade (a pop up window should enable selection of an exam term and a date when inserting a grade for the non-passed taken course).

Stimulus/Response Sequences

The interface is different for the different forms supporting this functionality.

In case of insertion of a grade for one student only, the inputs consist of the student's index number, selection of a course and insertion of a grade.

In case of bulk insertion of more grades for more students per course, the input is the selected course, the inserted indexes and then grades for every index.

Functional Requirements

- Create/edit/delete grades per exam (course) per student
- Upload/Download data for bulk results for an exam for more students.
- Send notifications/e-mails to students for their exam grades.

5.3.7 Administration of seminar thesis

Description and priority

The form should provide a list of all courses that the student can choose for a seminar thesis and a list of teachers. Text fields for the data necessary to be inserted. The teacher should have a form listing all seminar thesis applied by students choosing him/her. The teacher can choose one of the theses and update his/hers portion of data for it. Clerks can access all three forms.

Stimulus/Response Sequences

The inputs in the form are the selected student, a selected course and parameters for the seminar thesis to be inserted according to the data model of a seminar thesis. The submitted form should create a record for a new seminar thesis for the student

Functional Requirements

- Create/edit/delete seminar thesis for a student
- Upload/Download data for the seminar thesis (PDF document containing the seminar thesis description itself).

5.3.8 Administration of diploma thesis

Description and priority

The form should provide a list of all courses that the student can choose for a diploma thesis and a list of teachers. Text fields for the data necessary to be inserted. The teacher (mentor) should have a form listing all diploma thesis applied by students choosing him/her. The teacher can choose one of the theses and update his/hers portion of data for it. Clerks can access all three forms.

Stimulus/Response Sequences

The inputs in the form are the selected student, a selected course and parameters for the diploma thesis to be inserted according to the data model of a diploma thesis. The submitted form should create a record for a new diploma thesis for the student

Functional Requirements

- Create/edit/delete diploma thesis for a student
- Upload/Download data for the diploma thesis (PDF document containing the diploma thesis description itself)

5.3.9 Administration of exam terms

Description and priority

The form should provide a list of exam terms and fields to update a selected term, or to insert a new one.

Stimulus/Response Sequences

The clerks should be able to create new exam terms by selecting start and end dates for the exams applications and start and end dates for the exam term itself. The name of the term should also be entered. The created exam terms are later used for exam applications etc.

Functional Requirements

- Create/edit/delete exam terms
- Send notifications/e-mails to students and faculty for the created exam term

5.3.10 Administration of semesters

Description and priority

The form should provide a list of semesters and fields to update a selected semester, or to insert a new one. The semester is a complex structure. The semester itself is only a name and a start date and end date for the semester. However, additional data considering the distribution of courses and teachers per course must be maintained for each semester. For instance, in some semesters, certain course can be activated, while in other semesters certain courses can be deactivated. Also teachers per course per semester can change. Furthermore, inserting this dat for each semester can be a time consuming job. Therefore the system should enable creation a new semester as a copy of a previous semester, while some little changes can be performed in the new semester.

Stimulus/Response Sequences

The form should accept as inputs a start date and an end date for the semester. In the process of creation of the semester, the structure of courses and teachers per course should be copied from the previous appropriate fall/spring semester.

Functional Requirements

- Create/edit/delete semesters
- Create/edit courses per study program per semester
- Create/edit teachers per course per semester

5.3.11 Administration of quotas and tuition prices

Description and priority

The form should provide a list of quotas and fields to update a selected quota, or to insert a new one. Various prices for various quotas for various study programs and various start years can be inserted / updated.

Stimulus/Response Sequences

The user should insert a name for the quota (state funded, private funded, children of war veterans etc), For the available quotas, different quota prices per study program, per start year should be inserted. Also prices can be different for full time / part time students.

Functional Requirements

- Create/edit/delete quotas
- Create/edit/delete quota prices

5.3.12 Administration of courses in programs

Description and priority

The form should provide a list of study programs, and a list of courses for the program in question. Courses can be subtracted or associated to the program. When associating a course to a program, the semester number and other parameters are set for that course for the program selected.

Stimulus/Response Sequences

The form should enable selection of a study program and provide a list of courses linked to the study program. Also a list of the remaining courses should be provided so that another course could be linked to the study program.

Functional Requirements

- Create/edit/delete links among study programs and courses

5.3.13 Administration of courses

Description and priority

The form should provide a list of all courses and fields to show / update / insert details for each course.

Stimulus/Response Sequences

A list of available courses should be provided. Details for each course should be changeable. A new course can be added.

The list of courses can be filtered and sorted by faculty, study program, study year, and school year.

Functional Requirements

- Create/edit/delete courses
- Upload/Download data for courses (bulk insert)

5.3.14 Administration of programs

Description and priority

The form should provide a list of all programs and fields to show / update / insert details for each program.

Stimulus/Response Sequences

Filter through programs should be provided by their names. Details for the programs based on the data model should be shown on the form and updateble.

Functional Requirements

- Create/edit/delete study programs
- Upload/Download data for study programs (bulk insert)

5.3.15 Administration of programs revisions

Description and priority

The form should provide a list of all revisions and fields to show / update / insert details for each revision.

Stimulus/Response Sequences

The revisions of study programs occur once in several years and this form should provide fields for inserting new revision or change details of old ones. The name of the revision and the start/end year of validity are the only parameters.

Functional Requirements

- Create/edit/delete revisions

5.3.16 Administration of members of the faculty

Description and priority

The form should provide a list of all members of faculty and fields to show / update / insert details for each faculty member.

Stimulus/Response Sequences

The list of faculty members should be given as well as filters through the list. The filters should be by name, education level. Fields with details for each faculty member should be provided showing the data for the selected faculty member. New faculty members should be inserted and retired or members that have left should be deactivated (cannot be deleted since they are used for archiving, but they should be deactivated/hidden from the list and only show if special filter for them is activated.) The fields in the form are mapped to the data model.

Functional Requirements

Create/edit/delete faculty members

Upload/Download data for faculty members (bulk insert)

Send notifications/e-mails to faculty members, a subset of them according to filters

5.4 User interface to request reports and documents

5.4.1 Diploma thesis report

Description and priority

The form should provide reports of diploma thesis by various filters (by professor, by course, by year, time interval, diploma thesis status etc). With this reports the professors can overview their diploma thesis. Students can choose a diploma thesis from a list of all courses and a list of professors. The form should provide a list of numbered students that chose a certain diploma thesis to appropriate professor.

Stimulus/Response Sequences

The filter on the form for a diploma report should be based on a student, a professor, a time interval, a course etc. The report should result in a list / grid of details for the diploma thesis. The grid should be exportable in MS Excel format.

Functional Requirements

- Generate reports
- Download data for diploma thesis

5.4.2 Exams report

Introduction

This form should provide reports of exam applications by students by various filters (by professor, by course, by year, time interval etc). With this reports the professors can overview the students that have applied for an exam.

Once the examination is completed, recording of grades, deployment of applications to the personal files of students and publication of lists of passed candidates can be done. This service can prepare various reports by subjects, by study program, after the test terms or colloquium week:

- How many students applied for the exam and how many have passed it;
- Overview by test terms, how many applications are recorded for each term, how many of applied students attended the exam and how many have passed it;
- Average positive grade on the exam: the ratio between the total received positive grades with the number of candidates who have passed - According to examination terms and the chance to compare the average positive grades of each examination term.
- Average grade on the exam (ratio between total scores on the exam - positive and negative and the number of exam candidates)
- Review - out of passed students, how many passed with each grade;
- Review of candidates who passed the enrolled subject for the first time and summary by scored grades and average received positive grade of those who enrolled the subject for the first time.
- Review of candidates who passed the enrolled subject more than once and summary by scored grades and average positive grade of those who enrolled the subject for more than once.

Description and priority

Exam reports should be created solely by the Faculty staff (Professors). A professor should give out/edit/delete reports only for his/her courses. The report should follow a predefined structure (Word Template document) with editable fields.

Stimulus/Response Sequences

The system is supposed to enable entering all data that are required for the exam reports. This includes:

- Faculty / study program(s) selection
- Partial exam/exam
- Name of the course
- Date when the exam was held
- Date when the report is published
- Table of exam results (Index number and Exam mark)
- Name of professor
- Other relevant notes
- Digital signature of the professor

The exam report should be accessible by the Administration office staff.

Functional Requirements

Create/Edit/Delete an exam report form – Faculty staff should be able to fill in the template document for the exam report, upload/publish the exam report to be accessible to the students logging in the system. Faculty staff should have access only to the reports published by him/her and editing and deletion operations should include only his/her created reports.

Fill exam application – Faculty staff should fill the exam applications of the students applying for exam for his/her courses. The exam application should contain the date the exam took place, the name of the course, the grade and the ECTS credits for the course. In the end he/she should sign this application with his/her digital signature.

5.4.3 Graduated report

Description and priority

This form should provide the report of all graduated student by various filters. How many students has graduated in each academic year, report for average grade of graduated students, report of graduated students according the nationality, sex, age, average time of studying, study program, reports for part-time graduated student and full-time graduated students. This form also should provide the report of graduated students sorted by the professor who was a mentor of the diploma thesis.

Stimulus/Response Sequences

The filters for the report should be based on start year of study, quotas, enrolled semester, and study programs. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports
- Download data for graduated students

5.4.4 Elective courses report

Description and priority

This form should provide reports of students by various filters (by professor, by course, by study program). The report lists how many students chose a certain elective course. With this report the professors can overview which students have applied to follow his/her course.

Stimulus/Response Sequences

The filters for the report should be based on professor, course, study program. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports

- Download data for students

5.4.5 Master book

Description and priority

This form should provide printing the master book of enrolled students requested by the ministry of education.

Stimulus/Response Sequences

The filters for the report should be based on study program, start year of studies, enrolled semester, sex. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports
- Download data for students

5.4.6 List students per program

Description and priority

The form should provide reports of students per study program, how many students enrolled to any study program by various filters (sex, nationality etc...).

Stimulus/Response Sequences

The filters for the report should be based on study program, start year of studies, enrolled semester, sex. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports
- Download data for students

5.4.7 Seminar thesis report

Description and priority

The form should provide reports of seminar thesis by various filters (by professor, by course, by year, time interval,). With this report the professors can overview their seminar thesis. Students can choose a seminar thesis from a list of all courses and a list of professors. The form should provide a list of numbered students which choose a certain seminar thesis to appropriate professor.

Stimulus/Response Sequences

The filters for the report should be based on professors, courses, study programs. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports
- Download data for students

5.4.8 Administration of Master thesis

Description and priority

The form should provide reports of master thesis by various filters (by professor, by course, by year, time interval, master thesis status etc). With this reports the professors can overview their master thesis. Students can choose a master thesis from a list of all courses and a list of professors. The form should provide a list of numbered students which choose a certain master thesis to appropriate professor.

Stimulus/Response Sequences

The filters for the report should be based on professor, course, year, time interval, master thesis status. The resulting list should be exportable in MS Excel.

Functional Requirements

- Generate reports
- Download data for students

5.4.9 Issuing documents

Description and priority

The form should provide various types of documents - certificates, confirmation documents, forms, and announcements and other required documentation, reviews. It prepares various statistical reports in collaboration with students, teachers and the needs of the University management.

Some requirements for certificates, confirmation documents and so on, can be completed online, but will have to be submitted in person at Administrative office, to protect against abuse of the system.

In the course of study, students need confirmation documents of their status. The system will automatically provide the same issue on the basis of data on students who entered in the system.

In the course of study, students need certificates for passed exam to compete for scholarships, credits, student homes and so on.

After passing all the exams, at the request of the student, the Administrative Office issues transcripts and certificate that the student graduated – certificate for the acquisition of experts knowledge and passed exams.

The student who wants to transfer to another educational institution, at their request, a sign out document is issued.

Stimulus/Response Sequences

Users (students) are signed in so their identity is known, and their identity is used for issuing documents. When clerks use the form, the student is chosen by a filter.

Functional Requirements

Generate documents

5.4.10 Diploma/Diploma supplement form

Description and priority

The diploma/diploma supplement form should allow the Administration Office to fill in and issue diploma/diploma supplement information of a student requested through the system. After the documents are ready, the Administration Office should contact the student to collect the documents.

Stimulus/Response Sequences

The diploma/diploma supplement form should maintain a structure defined in the Law on Higher Education of RM.

Functional Requirements

Fill form – The Administration Office should fill in the form for the requested documents.

Check form – the system should check the data entered by the user in order to validate the fields.

5.4.11 Confirmation letter form

Description and priority

The confirmation letter form should allow the Administration Office to fill in and issue confirmation letters that a student is enrolled through the system. After the documents are ready, the Administration Office should contact the student to collect the documents.

Stimulus/Response Sequences

The confirmation letter form should have a static part which is the same, saying that the student is part of the University. The following fields should be editable in the form:

- Name
- Surname
- Index no.
- Address
- Faculty
- Year of enrollment

Functional Requirements

Fill form – The Administration Office should fill in the form for the requested documents.

Check form – the system should check the data entered by the user in order to validate the fields.

5.4.12 Notifications report form

Description and priority

Notifications should be enabled for the university staff. The university staff should be able to create/edit/delete notifications applicable for the students and for the rest of the university staff.

Notifications include:

- Class/exam schedules
- Events
- Activities
- Other

Stimulus/Response Sequences

The system is supposed to enable university staff access to notifications form. Template form should be filled in order for the report to be created and available. University staff should be able to edit/delete notification created solely by their user account.

The system should enable overview of all notification reports by students and university staff. University staff notifications should be addressed through the university e-mail system.

The system should enable creating notification with the following data:

- Title of notification
- Category (Class/exam schedule, important, holiday, exam application, etc.)
- Notification text
- Date of publishing
- Duration of notification

Functional Requirements

Filling notification form – the system should enable university staff to fill in the notifications form and edit/delete the notifications created by that user account. The notification should stay published during the period entered by the university staff that created the notification, and should be kept as hidden after the assigned duration in case of review process.

5.4.13 E-learning center

Description and priority

E-learning center should involve the functionalities described in Moodle system. It should enable course access and course creation/editing/deletion. Faculty staff should be able to create course structure, edit/delete his/her existing course, involve in forum discussions. Faculty staff should be able to easily contact course student group for notifications.

Course structure should have the following elements:

- Upload of lecture material
- Upload of additional material
- Create assignments and homework activities
- Grading

Students should be able to register to courses for which they have the enrolment key. They should be able to view course structure, download uploaded materials and lectures, upload assignments and homework, involve in discussion forums.

Stimulus/Response Sequences

The system is supposed to enable faculty staff log in into the center and associate and show their created course accounts.

The system is supposed to enable courses to be created with the following elements:

- Category (Faculty)
- Course name
- Course ID number
- Description
- Structure format
- Course start date
- Enable/disable view access to activity and grading reports by students
- Maximum upload size
- Enrolment start date – end date (duration)
- Enrolment notifications
- Availability (Available, Enrolment key)

The system should keep separate logs for the e-learning center for security reports in case of center failure or misuse report.

Functional Requirements

Create/edit/delete course – Faculty staff should be able to create courses, later edit end delete the courses created with his/her user account.

Create/edit course structure – Faculty staff should be able to upload lecture and additional materials, create assignments and homework, and perform student grading.

Upload/Download data – Students should be able to access registered courses and upload projects and homework and download/view uploaded materials in the course structure.

Send notifications/e-mails – The system should enable the faculty staff to easily contact a course group of students. The system should use the e-mail addresses saved in the students' user accounts to contact the students registered to the course by e-mail, or to display e-learning center notification on the next student log-in.

5.4.14 Student file center

Description and priority

The system should enable students and faculty staff to log into the Student file center.

Students should be able to check their activities status, write complaint for activities review, apply for exams, check passed exam marks, current average and cumulative GPA and edit personal information. Personal information should be filled in template document of EuropasssCV.

Faculty staff should be able to overview students' personal file by entering any of the later described search criterion. Faculty staff should also be enabled to assign internships to specific students as a student activity, and to mark a completed activity as finished.

Stimulus/Response Sequences

The system should enable faculty staff to view students' personal file by entering any of the following search criterion:

- Index no.
- First name
- Surname
- Faculty

The system should enable faculty staff to assign compulsory internships/other to students and to mark the same as completed when finished by the student. The faculty staff should enter the following information to assign a student activity:

- Title of activity
- Short description
- Date to be finished
- Completed/Not completed

The system should enable students to view/edit/delete information from their personal file. The following information is only to be viewed by the student:

- First name
- Surname
- Index no.
- Faculty
- Group
- Address
- Secondary educational institution
- Average and Cumulative GPA
- Courses completed
- Activities (Completed)

The following information is to be edited by the student:

- Exam application
- Diploma/diploma supplement/confirmation letter/certificate request
- Activities (Not completed, the student can just mark it as finished, and the same is later checked by the responsible faculty staff and marked completed)
- EuropassCV
- Additional certificates

Functional Requirements

Overview of Europass CV– The system should enable faculty staff an overview of student file of students in his/her faculty. Faculty staff should be able to confirm a finished student activity (ex. Internship, Seminar/Project assignment, etc.). Faculty staff should be able to assign activity to student (Internship, Project assignment) to be completed. The system should enable each student an overview of his/her personal file.

Create activity – The system should enable faculty staff to create activity assignments to students.

Edit personal information – Students should be able to edit their personal information included in the EuropassCV, while not the one's entered by the Administration Office.

Request diploma/diploma supplement/confirmation letters/certificate – Students should be able to request the above mentioned documents to be issued by the University through the system. The Administration Office should be responsible for implementing the procedure, fulfilling the requests and notify the students of the date to collect the documents.

5.4.15 List of documents issued to students

Students can require the following list of documents:

- Enrollment contract (by Template)
- Student's file (The file includes a list of all the payments made by the student)
- Student's personal information (This report does not include payments, solely the semesters validated by the student)
- Warning for not completed final course exams (by Template)
- List of all payments for all academic years of the student
- Certificate for completed exams (by Template)
- Confirmation letter for regular student (by Template)
- Resolution for leaving the University (by Template)
- Graduation certificate/Diploma supplement (by Template)

5.4.16 Alumni Site

Description and priority

The system should enable alumni students to be able to create and maintain a public user profile online.

Faculty staff should be able to overview students' personal file by entering any of the later described search criterion. Faculty staff should also be enabled to assign internships to specific students as a student activity, and to mark a completed activity as finished.

Stimulus/Response Sequences

The system should enable faculty staff to view students' personal file

The system should enable alumni students to view/edit/delete information from their personal profiles. The following information is only to be viewed by the student:

- First name
- Surname
- Faculty
- Group
- Address
- Current educational
- Average and Cumulative GPA
- Career progress
- Activities

The above information is to be edited by the student or the faculty staff:

Functional Requirements

Overview of Europass CV– The system should enable faculty staff an overview and edit of student history and profile.

Edit personal information – Students should be able to edit their personal information, while not the one's entered by the Administration Office.

5.5 Reports for MOE, SSO, internal and other external institutions

5.5.1 General overview

Description and priority

This section defines several statistical and analytical reports. In addition, the system should allow generation of reports which show all the records for a given data entity in tabular format.

The reporting subsystem should allow producing lists and reports. The lists should be configurable in a form of an advanced search allowing multiple criteria connected with logical expressions, also allowing additional sorting and grouping criteria to be defined. The system should enable for the user to choose which columns (attributes) to be shown on the list and in what order. Each report should be producible in paged (for printing) and continuous list (for web) form. The number of rows per page should be configurable. When producing table reports it is expected the header row to appear on each page of a multipage report.

For every report the user should be able to define certain text (title) that will appear on the beginning and on the end of the report, as well as the text that will appear in the header and the footer of each page. The user should be able to place certain predefined fields (like date, time, page number, enrolment period) in the report. The font to be used, the size, column width, and colors (including background color) should be configurable independently for the different parts (title, table, footer, …) of the report.

The system should also enable once configured report to be saved for future reuse.

Every generated list/report should be exportable in: csv, xls, xml, pdf, html and txt format.

Every list produced by the advanced search should allow on-screen manipulation of the result like filtering by column value, changing the column for sorting and the sort order before exporting.

At the end of every enrolment or annually or requested by MOE or SSO, the system allows report generation displaying various statistics on the enrolled candidates.

Summary reports (demos are provided in appendices 1-17) can be have the following forms:

- Summary reports by study group (app.1)
- Summary report for first year student enrolment (app.2)
- Summary report by ethnical characteristics (app.3)
- Summary reports for graduated PhD and M.SC students (app.4)
- Summary report for second cycle of studies (app.5)
- Summary reports for final year and graduated first cycle students (app.6)
- Summary report for enrolled first year students by quota (app. 7)
- Summary reports for student mobility (app. 8)
- Comparative analysis of enrolled students in two consecutive years (app.9)
- Statistical reports for each enrolled and graduated student for SSO (app.10-14)
- ECTS, diploma supplement and transcript of records template (app.15 and 16)

5.5.2 Advanced search and graphs

Description and priority

Advanced search should be realized according to setting logical AND function of several search criteria. Each criteria is logical OR combination of setting values to a given data field. For example, advanced search can be find all applications of prospective students that are living in municipalities Неготино OR Кавадарци and have finished type of high school гимназиско образование OR електротехничка струка.

Supported graphs to all search results are pie, bar and XY line graphs. All parameters for the graph header and selection of graph type can be entered by a form. Each graph should be exported in (MS Office recognizable) object form or picture format like jpg or png.

The system should allow a content of an e-mail message to be specified and an circular e-mail to be send to all the applicants of a produced list (by advanced search/filtering options) on each applicant's registered e-mail address.

5.5.3 Notification to students

Description and priority

The system should allow a content of an e-mail message to be specified and an circular e-mail to be send to all the students of a produced list (by advanced search/filtering options) on each applicant's registered e-mail address (or an SMS to be sent to their registered phone).

Similar functionality should be enabled also for mobile technologies using text messages (SMS) to be sent to their registered phone).

5.5.4 Summary reports by study group

Description and priority

For each faculty within the university and for each study programme show the number of enrolled candidates (by sex and total) in each of study years.

A sample presented on next figure shows only the first page of total report (excel version can be found as appendix.1)

P R E G L E D

na вкупниот broj studenti po studiski godini vou~ebnata 2010/2011

FAKULTET	Status		prva godina			vtora godina			treta godina			~etvrta godina			peta godina			{esta godina			vkupno		
			M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk
PRIRODNOMATEMATI^KI	Prvpat zapi{ani	R	244	169	413	131	152	283	142	206	348	152	224	376	111	110	221	47	67	114	827	928	1 755
		Vk	244	169	413	131	152	283	142	206	348	152	224	376	111	110	221	47	67	114	827	928	1 755
	So kofinansirawe	R	27	32	59	4	17	21	13	30	43	2	5	7	0	0	0	0	0	0	46	84	130
		V	47	22	69	16	13	29	13	7	20	8	25	33	2	5	7	0	0	0	86	72	158
		Vk	74	54	128	20	30	50	26	37	63	10	30	40	2	5	7	0	0	0	132	156	288
	Povtorno zapi{ani	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		V	0	0	0	0	0	0	0	0	0	8	39	47	0	0	0	0	0	0	8	39	47
		Vk	0	0	0	0	0	0	0	0	0	8	39	47	0	0	0	0	0	0	8	39	47
	Vkupno	R	271	201	472	135	169	304	155	236	391	154	229	383	111	110	221	47	67	114	873	1 012	1 885
		V	47	22	69	16	13	29	13	7	20	16	64	80	2	5	7	0	0	0	94	111	205
		Vk	318	223	541	151	182	333	168	243	411	170	293	463	113	115	228	47	67	114	967	1 123	2 090
Matematika	Prvpat zapi{ani	R	5	6	11	6	7	13	3	20	23	6	18	24	2	10	12	0	3	3	22	64	86
		Vk	5	6	11	6	7	13	3	20	23	6	18	24	2	10	12	0	3	3	22	64	86
	So kofinansirawe	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		V	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	2	2
		Vk	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	2	2
	Povtorno zapi{ani	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		V	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	4	4
		Vk	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	4	4
	Vkupno	R	5	6	11	6	7	13	3	20	23	6	18	24	2	10	12	0	3	3	22	64	86
		V	0	0	0	0	1	1	0	1	1	0	4	4	0	0	0	0	0	0	0	6	6
		Vk	5	6	11	6	8	14	3	21	24	6	22	28	2	10	12	0	3	3	22	70	92

5.5.5 Summary report for first year student enrolment

Description and priority

Reports need to display the following data in tabular format: For each study programme show the number of enrolled candidates (by ethnicity and total) that:

- have applied
- were tested (if testing was required)
- have passed the test
- are accepted for enrolment in state quota
- are accepted for enrolment in private quota (full-time + part-time + total)
- are accepted for enrolment in ethnicity quota
- are accepted for enrolment under special circumstances (for every type)
- are accepted for enrolment (total – sum of previous)

The report also displays total number of enrolled candidates for faculty, group of faculties (natural sciences, engineering, humanities, arts, …), and the number of available places for the next enrolment term (separately for the state quota, private quota and ethnic quotas, full-

time and part-time). The MS Excel document in appendix 2 gives further details about this report. A sample is presented on the next figure:

П Р Е Г Л Е Д

Реализација на Конкурсот за запишување студенти државјани на РМ во прва година според националната

ФАКУЛТЕТ	Нацио-налност	ПРИЈАВЕНИ КАНДИДАТИ																
		пријавени			проверка на знаење и умеења			дополнителн о тестирање			положиле			во државна квота	со кофинанси-рање		
		р	в	вк	р	в	вк	р	в	вк	р	в	вк	р	р	в	вк.
Факултет за електротехника и информациски технологии	Македонци	475	0	475	0	0	0	0	0	0	0	0	0	317	111	0	111
	Албанци	8	0	8	0	0	0	0	0	0	0	0	0	4	4	0	4
	Турци	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	Срби	13	0	13	0	0	0	0	0	0	0	0	0	2	4	0	4
	Власи	9	0	9	0	0	0	0	0	0	0	0	0	2	5	0	5
	Роми	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Бошњаци	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Други	9	0	9	0	0	0	0	0	0	0	0	0	4	3	0	3
	Вкупно	518	0	518	0	0	0	0	0	0	0	0	0	329	127	0	127
☐ телекомуникации	Македонци	73	0	73	0	0	0	0	0	0	0	0	0	73	0	0	0
	Албанци	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Турци	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Срби	2	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0
	Власи	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
	Роми	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Бошњаци	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Други	4	0	4	0	0	0	0	0	0	0	0	0	4	0	0	0
	Вкупно	80	0	80	0	0	0	0	0	0	0	0	0	80	0	0	0
☐ компјутерско системско инженерство и автоматика	Македонци	83	0	83	0	0	0	0	0	0	0	0	0	76	7	0	7
	Албанци	4	0	4	0	0	0	0	0	0	0	0	0	4	0	0	0
	Турци	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Срби	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	Власи	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Роми	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Бошњаци	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Други	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Вкупно	90	0	90	0	0	0	0	0	0	0	0	0	80	7	0	7

га припадност во учебната 2010/2011 - ПРВ УПИСЕН РОК

ЗАПИШАНИ КАНДИДАТИ											СЛОБОДНИ МЕСТА					
вон квота по националност	деца без родители	лица со прв степен на инвалидност	воени инвалиди	лица од дом за напуште-ни деца	припадници на безбед-носни сили	деца на починати учесници во мир-овни операции	ВКУПНО запишани			државна квота	кофинан-сирање		ВКУПНО		
р	р	р	р	р	р	р	р	в	вк	р	р	в	р	в	вк
0	0	0	0	0	2	0	430	0	430						
0	0	0	0	0	0	0	8	0	8						
0	0	0	0	0	0	0	0	0	0						
2	0	0	0	0	0	0	8	0	8	0	0	0	0	0	0
1	0	0	0	0	0	0	8	0	8						
0	0	0	0	0	0	0	0	0	0						
1	0	0	0	0	0	0	1	0	1						
0	0	0	0	0	0	0	7	0	7						
4	0	0	0	0	2	0	462	0	462	0	0	0	0	0	0
0	0	0	0	0	0	0	73	0	73						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	4	0	4						
0	0	0	0	0	0	0	80	0	80	0	0	0	0	0	0
0	0	0	0	0	0	0	83	0	83						
0	0	0	0	0	0	0	4	0	4						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	87	0	87	0	0	0	0	0	0

5.5.6 Summary report by ethnical characteristics

Description and priority

For each faculty within the university and for each study programme show the number of enrolled students (by ethnical group and total).

A sample presented on next figure shows only the first page of total report (excel version can be found as appendix.3)

P R E G L E D

n a s t u d e n t i t e - d r ` a v j a n i n a R M s p o r e d n a c i o n a l n a t a p r i p a d n o s t v o u ~ e b n a t a 20102011

FAKULTET	Status	Makedonci			Albanci			Turci			Vlasi			Romi			Srbi			Bo{waci			Drugi			Vkupno		
		M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk	M	@	Vk
PRIRODNOMATEMATI^KI	R	783	920	1 703	29	32	61	17	14	31	5	1	6	7	0	7	22	16	38	5	5	10	5	24	29	873	1 012	1 885
	V	63	89	152	4	4	8	0	0	0	0	0	0	0	0	0	2	0	2	1	1	2	24	17	41	94	111	205
	Vk	846	1 009	1855	33	36	69	17	14	31	5	1	6	7	0	7	24	16	40	6	6	12	29	41	70	967	1 123	2 090
Matematika	R	17	60	77	4	1	5	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	2	22	64	86
	V	0	5	5	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6
	Vk	17	65	82	4	2	6	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	2	22	70	92
Fizika	R	21	10	31	4	0	4	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	28	10	38
	V	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2
	Vk	22	11	33	4	0	4	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	29	11	40
Hemija	R	36	180	216	2	3	5	0	2	2	0	0	0	0	0	0	1	5	6	0	1	1	0	5	5	39	196	235
	V	5	23	28	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	5	26	31
	Vk	41	203	244	2	5	7	0	2	2	0	0	0	0	0	0	1	5	6	0	2	2	0	5	5	44	222	266
Biologija	R	81	309	390	3	23	26	3	10	13	2	0	2	0	0	0	4	6	10	0	3	3	0	1	1	93	352	445
	V	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8
	Vk	81	317	398	3	23	26	3	10	13	2	0	2	0	0	0	4	6	10	0	3	3	0	1	1	93	360	453
geografija	R	150	98	248	3	1	4	6	0	6	0	0	0	3	0	3	5	3	8	4	0	4	0	1	1	171	103	274
	V	32	31	63	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	39	31	70
	Vk	182	129	311	7	1	8	6	0	6	0	0	0	3	0	3	5	3	8	4	0	4	3	1	4	210	134	344
etnologija i antropologija	R	31	45	76	0	0	0	0	0	0	2	0	2	3	0	3	1	0	1	0	0	0	0	1	1	37	46	83
	V	16	10	26	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	1	0	1	0	0	0	19	10	29
	Vk	47	55	102	0	0	0	0	0	0	2	0	2	3	0	3	3	0	3	1	0	1	0	1	1	56	56	112
informatika	R	279	151	430	7	4	11	5	1	6	0	1	1	0	0	0	5	1	6	1	1	2	0	0	0	297	159	456
	V	7	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	8
	Vk	286	152	438	7	4	11	5	1	6	0	1	1	0	0	0	5	1	6	1	1	2	0	0	0	304	160	464
dispezirani studii po informatika vo Kriva Palanka	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Vk	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
informati-kitehnologii(6 semestri)	R	164	58	222	5	0	5	2	0	2	1	0	1	0	0	0	6	0	6	0	0	0	3	0	3	181	58	239
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	2	23	21	2	23
	Vk	164	58	222	5	0	5	2	0	2	1	0	1	0	0	0	6	0	6	0	0	0	24	2	26	202	60	262

5.5.7 Summary reports for graduated PhD and M.SC students

These reports show the number of graduated students for third (PhD) and second (M.Sc.) cycle of studies. The following figure is a sample report and appendix 4 contains details (Appendix 4 presents corresponding excel file).

P R E G L E D

na brojot na magistriranite i doktoriranite kadri za u~ebnata 2009/2010

FAKULTET / INSTITUT	MAGISTRI magistrirale na:				DOKTORI NA NAUKITE doktorirale na:			
	Univerzitetot "Sv. K i M"		Drugi univerziteti		Univerzitetot "Sv. K i M"		Drugi univerziteti	
	do 2009/2010	vo 2009/2010	do 2009/2010	vo 2009/2010	do 2009/2010	vo 2009/2010	do 2009/09	vo 2009/2010
PRIRODNO - MATEMATI^KI	499	44	0	0	342	26	0	0
VKUPNO PRIRODNO-MATEMATI^KI	499	44	0	0	342	26	0	0
ARHITEKTONSKI	27	2	0	0	23	2	0	0
GRADE@EN	75	29	0	0	57	6	0	0
FAKULTET ZA ELEKTRO-TEHNIKA I INFORMA-CISKI TEHNOLOGII	242	50	0	0	76	5	25	0
MA[INSKI	236	17	0	0	100	0	0	0
TEHNOLO[KO - METALUR[KI	73	13	7	0	58	4	12	0
VKUPNO TEHNI^KO-TEHNOLO[KI NAUKI	653	111	7	0	314	17	37	0

5.5.8 Summary report for second cycle of studies

Description and priority

For each faculty within the university and for each study programme show the number of enrolled and graduated students for second cycle of studying.

A sample presented on next figure shows only the first page of total report (excel version can be found as appendix.5)

P R E G L E D

na brojot na zapi{ani i diplomirani magistri i specijalisti vo u~ebnata 2009/2010

FAKULTET	pol	Broj na zapi{ani postdiplomci									Broj na diplomirani postdiplomci								
		Magistri			Specijalisti			Vkupno			Magistri			Specijalisti			Vkupno		
		do 2009/2010	vo 2009/2010	Vk.	do 2009/2010	vo 2009/2010	Vk.	do 2009/2010	vo 2009/2010	Vk.	do 2009/2010	vo 2009/2010	Vk.	do 2009/2010	vo 2009/2010	Vk.	do 2009/2010	vo 2009/2010	Vk.
PRIRODNO-MATEMATI^KI	M	399	83	482	110	0	110	509	83	592	363	17	380	102	2	104	465	19	484
	@	500	49	549	59	0	59	559	49	608	399	21	420	43	4	47	442	25	467
	Vk.	899	132	1 031	169	0	169	1 068	132	1200	762	38	800	145	6	151	907	44	951
VKUPNO PRIRODNO-MATEMATI^KI	M	399	83	482	110	0	110	509	83	592	363	17	380	102	2	104	465	19	484
	@	500	49	549	59	0	59	559	49	608	399	21	420	43	4	47	442	25	467
	Vk.	899	132	1 031	169	0	169	1 068	132	1200	762	38	800	145	6	151	907	44	951
ARHITEKTONSKI	M	66	0	66	0	0	0	66	0	66	18	1	19	0	0	0	18	1	19
	@	50	0	50	0	0	0	50	0	50	13	1	14	0	0	0	13	1	14
	Vk.	116	0	116	0	0	0	116	0	116	31	2	33	0	0	0	31	2	33
GRADE@EN	M	440	53	493	0	0	0	440	53	493	60	21	81	0	0	0	60	21	81
	@	102	24	126	0	0	0	102	24	126	15	8	23	0	0	0	15	8	23
	Vk.	542	77	619	0	0	0	542	77	619	75	29	104	0	0	0	75	29	104
FAKULTET ZA ELEKTRO-TEHNIKA I INFORMA-CISKI TEHNOLOGII	M	1 127	221	1 348	0	0	0	1 127	221	1348	200	37	237	0	0	0	200	37	237
	@	255	112	367	0	0	0	255	112	367	42	13	55	0	0	0	42	13	55
	Vk.	1 382	333	1 715	0	0	0	1 382	333	1715	242	50	292	0	0	0	242	50	292
MA[INSKI	M	812	137	949	0	0	0	812	137	949	207	14	221	0	0	0	207	14	221
	@	94	33	127	0	0	0	94	33	127	29	3	32	0	0	0	29	3	32
	Vk.	906	170	1 076	0	0	0	906	170	1076	236	17	253	0	0	0	236	17	253
TEHNOLO[KO-METALUR[KI	M	296	12	308	14	0	14	310	12	322	84	2	86	0	0	0	84	2	86
	@	285	21	306	22	0	22	307	21	328	122	11	133	0	0	0	122	11	133
	Vk.	581	33	614	36	0	36	617	33	650	206	13	219	0	0	0	206	13	219

5.5.9 Summary reports for final year and graduated first cycle students

These reports show the number of final year and graduated students for first cycle of studies (undergraduate studies). The following figure is a sample report and appendix 6 contains details.

P R E G L E D

na brojot na apsolventi i diplomirani studenti vo u~ebnata 2009/2010

FAKULTET	Na~in na studii	Apsolventi vo u~ebna 2009/2010	Broj na diplomirani studenti do u~ebna 2009/2010	Broj na diplomirani studenti vo u~ebna 2009/2010	VKUPNO
PRIRODNO-MATEMATI^KI	redovni	2933	8291	335	8626
	vonredni	115	1068	29	1097
	vkupno	3048	9359	364	9723
VKUPNO PRIRODNO-MATEMATI^KI	**redovni**	**2.933**	**8.291**	**335**	**8.626**
	vonredni	**115**	**1.068**	**29**	**1.097**
	vkupno	**3.048**	**9.359**	**364**	**9.723**
ARHITEKTONSKI	redovni	305	3879	113	3992
	vonredni	0	0	0	0
	vkupno	305	3879	113	3992
GRADE@EN	redovni	20	3228	120	3348
	vonredni	0	0	25	25
	vkupno	20	3228	145	3373
FAKULTET ZA ELEKTRO-TEHNIKA I INFORMA-CISKI TEHNOLOGII	redovni	0	6213	0	6213
	vonredni	0	0	0	0
	vkupno	0	6213	0	6213
MA[INSKI	redovni	497	5634	0	5634
	vonredni	0	0	0	0
	vkupno	497	5634	0	5634
TEHNOLO[KO-METALUR[KI	redovni	147	4697	100	4797
	vonredni	0	0	0	0
	vkupno	147	4697	100	4797
VKUPNO TEHNI^KO-TEHNOLO[KI NAUKI	**redovni**	**969**	**23.651**	**333**	**23.984**
	vonredni	**0**	**0**	**25**	**25**
	vkupno	**969**	**23.651**	**358**	**24.009**
MEDICINSKI	redovni	214	9634	186	9820
	vonredni	48	0	25	25
	vkupno	262	9634	211	9845
STOMATOLO[KI	redovni	0	3172	0	3172
	vonredni	0	0	0	0
	vkupno	0	3172	0	3172
FARMACEVTSKI	redovni	17	1739	126	1865
	vonredni	0	0	0	0
	vkupno	17	1739	126	1865
VKUPNO MEDICINSKI NAUKI I ZDRAVJE	**redovni**	**231**	**14.545**	**312**	**14.857**
	vonredni	**48**	**0**	**25**	**25**
	vkupno	**279**	**14.545**	**337**	**14.882**

5.5.10 Summary report for enrolled first year students by quota

Description and priority

For each faculty within the university and for each study programme show the number of enrolled students for first cycle of studying, and group by quota.

A sample presented on next figure shows only the first page of total report (excel version can be found as appendix.7)

TABELA 1

P R E G L E D

na prvpat zapi{anite studienti vo prva godina vo akademskata 2010/2011 godina, dr`avjani na Republika Makedonija

FAKULTET	pol	vo ramkite na dr`avna kvota	So kofinansirawe na studiraweto			Redovni nadvor od kvota po osnov na nacionalnost			Nadvor od kvota so zavr{en del od srednoto obrazovanie vo stranstvo, ili so me|unarodna matura			Nadvor od kvota kako pripadnici na bezbednosnite sili na RM	Nadvor od kvota - deca bez roditeli	Nadvor od kvota -lica so 1 stepen na invaliditet	Nadvor od kvota - voeni invalidi	Nadvor od kvota - lica rasteni vo domovi za napu{teni lica	deca na po~inati u~esnici vo mirovni operacii	vkupno zapi{ani		
		r	r	v	Vk.	Dr`a	Kofina	vk	Dr`a.	Kofina.	vk	r	r	r	r	r	r	r	v	vk
PRIRODNO-MATEMATI^KI	ma{ki	**257**	**13**	**52**	**65**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**271**	**52**	**323**
	`enski	**181**	**28**	**29**	**57**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**209**	**29**	**238**
	vkupno	**438**	**41**	**81**	**122**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**480**	**81**	**561**
studii po matematika	ma{ki	6	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**6**	**1**	**7**
	`enski	6	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**6**	**1**	**7**
	vkupno	12	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	**12**	**2**	**14**
studii po fizika	ma{ki	9	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**9**	**1**	**10**
	`enski	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**2**	**0**	**2**
	vkupno	11	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**11**	**1**	**12**
studii po hemija	ma{ki	10	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**11**	**0**	**11**
	`enski	35	0	7	7	0	0	0	0	0	0	0	0	0	0	0	0	**35**	7	**42**
	vkupno	45	1	7	8	0	0	0	0	0	0	0	0	0	0	0	0	**46**	7	**53**
studii po biologija	ma{ki	20	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	**24**	**0**	**24**
	`enski	41	24	0	24	0	0	0	0	0	0	0	0	0	0	0	0	**65**	**0**	**65**
	vkupno	61	28	0	28	0	0	0	0	0	0	0	0	0	0	0	0	**89**	**0**	**89**
studii po geografija	ma{ki	84	8	22	30	0	0	0	0	0	0	1	0	0	0	0	0	**93**	**22**	**115**
	`enski	43	4	11	15	0	0	0	0	0	0	0	0	0	0	0	0	**47**	**11**	**58**
	vkupno	127	12	33	45	0	0	0	0	0	0	1	0	0	0	0	0	**140**	**33**	**173**
etnologija i antropologija	ma{ki	6	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	**6**	**5**	**11**
	`enski	9	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	**9**	**5**	**14**
	vkupno	15	0	10	10	0	0	0	0	0	0	0	0	0	0	0	0	**15**	**10**	**25**
studii po informatika	ma{ki	68	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	**68**	**4**	**72**
	`enski	29	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**29**	**1**	**30**
	vkupno	97	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	**97**	**5**	**102**

5.5.11 Summary reports for student mobility

These reports show the number of student mobility. The following figure is a sample report and appendix 8 contains details.

P R E G L E D

		Преглед на остварени мобилности на студенти ОД ДРУГИ УНИВЕРЗИТЕТ НА УКИМ во учебната 2008/09 година во рамките на наставно-образовниот процес				Преглед на остварени мобилности на студенти ОД ДРУГИ УНИВЕРЗИТЕТ НА УКИМ во учебната 2008/09 година во рамките на наставно-образовниот процес			
FAKULTETI		број на студенти кои оствариле мобилност	вид на мобилност (во склоп на студиската програма, во рамки на проект, практична	вид на мобилност (во склоп на студиската програма, во рамки на проект, практична	освоени кредити	број на студенти кои оствариле мобилност	вид на мобилност (во склоп на студиската програма, во рамки на проект, практична	вид на мобилност (во склоп на студиската програма, во рамки на проект, практична	освоени кредити
ARHITEKTONSKI	додипломски студии (прв циклус)	0	0	0	0	0	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	0	0	0	0	0	0	0	0
GRADE@EN	додипломски студии (прв циклус)	0	0	0	0	0	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	3 (2007/08)	SEEFORM	0	po zavr{uvawe na	5	SEEFORM	0	po zavr{uvawe na studiite
FAKULTET ZA ELEKTRO-TEHNIKA I INFORMA-CISKI TEHNOLOGII	додипломски студии (прв циклус)	0	0	0	0	2	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	0	0	0	0	0	0	0	0
MA[INSKI	додипломски студии (прв циклус)	0	0	0	0	0	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	0	0	0	0	0	0	0	0
TEHNOLO[KO-METALUR[KI	додипломски студии (прв циклус)	0	0	0	0	0	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	0	0	0	0	0	0	0	0
PRIRODNO-MATEMATI^KI	додипломски студии (прв циклус)	0	0	0	0	0	0	0	0
	постдипломски студии (втор циклус)	0	0	0	0	0	0	0	0
	докторски студии (трет циклус)	0	0	0	0	0	0	0	0

5.5.12 Comparative analysis of enrolled students in two consecutive years

Description and priority

For each faculty within the university and for each study programme show the comparison of the number of enrolled students for first cycle of studying in two consecutive years.

A sample presented on next figure shows only the first page of total report (excel version can be found as appendix.9)

SPOREDBENA TABELA
za brojot na zapi{anite studenti vo prva godina
vo u~ebnite 2005/06 i 2006/07 godina

FAKULTET	pol	vkupno zapi{ani vo u~ebnata 2006/07			vkupno zapi{ani vo u~ebnata 2005/06			procent na zgolemuvawe na brojot na studenti		
		r	*v*	*vk*	*r*	*v*	*vk*	*r*	*v*	*vk*
ARHI-TEKTONSKI	ma{ki	43	0	43	53	0	53	-23,26		-23,26
	`enski	66	0	66	50	0	50	24,24		24,24
	vkupno	109	0	109	103	0	103	5,50		5,50
GRADE@EN	ma{ki	113	0	113	79	0	79	30,09		30,09
	`enski	27	0	27	23	0	23	14,81		14,81
	vkupno	140	0	140	102	0	102	27,14		27,14
FAKULTET ZA ELEKTROTEHNIKA I INFORM TEHNOLOGII	ma{ki	467	0	467	312	0	312	33,19		33,19
	`enski	148	0	148	107	0	107	27,70		27,70
	vkupno	615	0	615	419	0	419	31,87		31,87
MA[INSKI	ma{ki	400	0	400	341	0	341	14,75		14,75
	`enski	55	0	55	50	0	50	9,09		9,09
	vkupno	455	0	455	391	0	391	14,07		14,07
TEHNOLO[KO-METALUR[KI	ma{ki	61	0	61	43	0	43	29,51		29,51
	`enski	81	0	81	72	0	72	11,11		11,11
	vkupno	142	0	142	115	0	115	19,01		19,01
RUDARSKO-GEOLO[KI	ma{ki	61	6	67	80	9	89	-31,15	-50,00	-32,84
	`enski	17	2	19	23	2	25	-35,29	0,00	-31,58
	vkupno	78	8	86	103	11	114	-32,05	-37,50	-32,56

5.5.13 Statistical reports for each graduated student for SSO

These reports are sent to SSO with details about study program and personal data for each graduated student.

- Appendix 10 is the report for each graduated student in first cycle (undergraduate)
- Appendix 11 is the report for each graduated student in second cycle (M.Sc)
- Appendix 12 is the report for each graduated student in third cycle (Ph.D)
- Appendix 13 is the report for each enrolled student in second cycle (M.Sc)
- Appendix 14 is the report for each enrolled student in third cycle (Ph.D)

5.5.14 ECTS related student guides and documents (app.15 and 16)

The system should be able to generate essential ECTS documents as attached in app.15 and 16.

5.6 Internal reports

The following reports are used by the management, administrative and academic staff.

5.6.1 Final exam reports

The next figure presents the list of students registered for course's final exam.

Факултет: Факултет за економски науки — Насока: Маркетинг менаџмент

СПИСОК

На студенти кои го полагале испитот по предметот

Планирање на маркетинг

кај Проф. д-р Петреска Ленче

Октомвриски испитен рок во 2009/2010

Ред. бр.	Индекс	Име и Презиме	Оценка	со букви
1	0214 ФЕН/ММ	Николов Зоран		
2	0332 ФЕН/ММ	Колчкоски Пецо		
3	0729/08 ФЕН	Ѓаковски Иван		

The next figure presents results form for final exam.

Факултет: Факултет за информатика — Насока: Софтверско инженерство

СПИСОК

На студенти кои го полагале испитот по предметот

Компјутерски алатки

кај Доцент Д-р. Гелев Сашо

Октомвриски испитен рок во 2009/2010

Датум: 13.10.2010

Ред. бр.	Индекс	Име и Презиме	Оценка	
1	0026/10 ФИ	Бојковски Влатко	7	седум
2	02526-колеџ	Николовски Дарко	8	осум

Заклучно со реден број 2

The next figure presents list and number of students registered for final exam (All courses for which students registered for course's final exam per exam session)

ЛИСТА НА ПРИЈАВЕНИ ПРЕДМЕТИ

Во Ноемвриски испитен рок во 2004/2005

Факултет за економски науки

Банкарски менаџмент

Предмет	Број на пријавени
Основи на економија	44

Факултет за економски науки

Финансиско-банкарски и даночен менаџмент

Предмет	Број на пријавени

5.6.2 Partial exams

The next figure presents list of students registered for course's partial exam

Факултет	Насока
Факултет за економски науки	Маркетинг менаџмент

СПИСОК

На студенти кои го полагале колоквимот по предметот

Управување со производ

кај Проф. д-р Конеска Љиљана

Прв колоквиум во Зимски семестар во 2010/2011

Ред. бр.	Индекс	Име и Презиме	Поени
1	0183/09 ФЕН	Ѓорѓиоска Ивана	
2	0461/08 ФЕН	Митев Игор	
3	0489/08 ФЕН	Сламкова Милена	
4	0491/08 ФЕН	Тошевски Благоја	

Lists that can be generated by advanced search functions include:

- List of professors at the University (Contac information included)
- List of student who left the University (Students who during their studies have left the University. Contact information is included)
- List of courses by Faculty department – a sample is presented on next figure

ЛИСТА НА ПРЕДМЕТИ

По насоката Софтверско инженерство од Факултет за информатика

Ред. Број	Предмет
Прв семестар	
1	Алгебарски структури
1	Алгебарски структури
1	Алгебарски структури
1	Англиски јазик 1
1	Англиски јазик 1

- List of all students at the University (Index and contact information included)
- List of all students sorted by Surname, Department, Faculty, Address, Elective language, Semester
- List of students with unfulfilled activities
- List of students by Faculty showing number of passed exams
- List of students who have not signed the Enrollment contract.
- List of all existing courses at the University
- List of students who have not validated the semester (by Semester) – a sample is presented on next figure

ЛИСТА НА СТУДЕНТИ КОИ НЕ ЗАВЕРИЛЕ

Летен семестар во 2009/2010

Р.бр.	Индекс	Презиме и Име	Телефон дом.	Телефон моб.
1	00004	Манева Маја	043 222-652	
2	00006-Охрид	Чутра Кренар	046 832-521	070/325-254

5.6.3 Seminar and project work

The menu options and advanced search functions can also be used for obtaining information about seminar and projects by the following lists:

- Students who have not attended a seminar and have not registered for project assignment
- Students who presented their project assignment for not attending the seminar
- Students who attended the seminar
- Students who registered but have not attended the seminar
- Students who registered for project assignments listed by course – a sample is presented on next figure

ЛИСТА НА СТУДЕНТИ

- проектни задачи -

Предмет Објектно ориентирано програмирање

Р.бр.	Индекс	Презиме и Име	Одбранета
1	0172/08 ФИ	Мишева Елена	Одбранета
Тема:	Контрола на тек на програма и условни искази		
2	0178/08 ФИ	Ценик Горди	Одбранета
Тема:	Структури и енумерации		

- Students who are supposed to present a project assignment for not attending the seminar

5.6.4 Statistics about success rate

The menu options and advanced search functions can also be used for business intelligence about success rate:

- List of courses with number of students that passed and failed the exam – a sample is presented on next figure

СПИСОК НА ПРЕДМЕТИ СО БРОЈ НА ПОЛОЖЕНИ СТУДЕНТИ

за Втора година година во 2009/2010 учебна година по насокатаСофтверско инженерство

Р.бр.	Предмет	Положиле	Не положиле
0	Англиски јазик 1	4	74
Р.бр.	Предмет	Положиле	Не положиле
0	Вовед во организација на компјутери	75	3

- List of professors and their average mark for all course
- List of professors and their average mark per course
- Number of enrolled students per year, Faculty, Department. (Report shows the number of students sorted by year of enrollment and Faculty who have successfully graduated, ones that have not graduated and the total number)
- The number of students enrolled by semester. (The report includes the number of students enrolled by Faculty as well as the semester in which they enrolled).

5.6.5 Financial and fee reports

The menu options and advanced search functions can also be used for obtaining financial info. The following are example of lists obtained in internal reporting about financials:

- Students – in debt by academic year
- Students who have not submitted tuition and fees for the academic year and other obligatory activities.

ЛИСТА НА СТУДЕНТИ КОИ ПЛАТИЛЕ

Учебна година _____

Школска година _____

Уплата _____

Р.бр.	Индекс	Презиме и Име	Обврска	Година
1	00004	Манева Маја	Студентска организација	1
2	00004	Манева Маја	Осигурување	1
3	00006-Охрид	Чутра Кренар	Студентска организација	1

- Students who have submitted tuition and fees (The report can be generated by Academic year, School year and type of fee (Tuition, Uniform, etc.)

ЛИСТА НА СТУДЕНТИ КОИ ПЛАТИЛЕ

Учебна година 2009/2010

Школска година Прва година

Уплата Школарина

За истата година

Р.бр.	Индекс	Презиме и Име
1	0008/09 ФИ	Миовска Ирена
2	0015/09 ФИ	Томиќ Денис
3	0017/09 ФС	Додевски Марио

- Students who have not submitted tuition and fees. (The report can be generated by Academic year, School year and type of fee (Tuition, Uniform, etc.)
- Fees submitted by students generated by date.

УПЛАТИ

Уплати книжени на: 10.09.2010

Досие	Име и Презиме	Година	Обврска	Сума	Извод
0009/10 ФАД	Кристијан Ниновски	Општа уплата	Униформа	5200	КБ-
0010/08 ФДК	Сабина Бакиќ	Општа уплата	Испит	3720	КБ-
0013/10 ПФ	Ненад Танески	Општа уплата	Униформа	880	КБ-
0013/10 ФЕН	Кристина Николова	Прва година	Школарина	61700	КБ-

- Payments made by type of fee and date (date from-to)
- Payments made by student
- List of students for payments made for Tuition and fees.

5.7 Other administrative roles

Description and priority

The form should provide timely and accurate information for students.

Stimulus/Response Sequences

It is necessary to provide bulleting to:

- Reports which will be generated by the system (report of examination held, a list of exams per year and guidance, etc.);
- Other appropriate announcements (for example - a announcement for approved additional test term, an announcement of deferred exam or colloquium, announcement for delayed classes, announcements for seminars, training etc..);
- Schedule of classes;
- Schedule of exams;
- Terms and so on.

This kind of reports should be available on the Web.

The form, also, should provide the timely information and continuity in curriculum, the teachers could highlight different announcements, students statements and other notices in connection with teaching units, classes, classes delay, holding a variety of extracurricular activities and more. Also, there is a need to define a determined period of validity of the bulletin announcement on-line "bulletin board".

Functional Requirements

- Generate reports

5.7.1 Administration of class schedule

Description and priority

The form should provide each student with the opportunity to examine currently enrolled subjects, professors and assistants, who will teach these subjects, the whole schedule with the total number of class duration.

Stimulus/Response Sequences

The student can select a course and retrieve the schedule for the course. Also a schedule for the study program/study year can be received after selecting the appropriate input.

Functional Requirements

- Download schedules

5.7.2 Administration of faculties

Description and priority

The form should provide a list of all faculties and fields to show / update / insert details for each faculty.

Stimulus/Response Sequences

The form should enable the change of details for the selected faculty. The data fields should be mapped from the data model.

Functional Requirements

- Create/edit/delete faculties

5.7.3 Import of candidate data from enrolment module

Description and priority

The form should enable retrieval of data from the enrolment module into the core database.

Stimulus/Response Sequences

This form provides fields to search for a student candidate already registered in the enrolment module. Candidates can be searched by the unique number given when enrolling or by first name or by last name or by EMBG. A web service returns the results of the search from the enrolment module, and then the results are listed. User can then select a candidate for actual enrolment. A web service returns all data for the selected candidate to the core module. Core module's interface takes the user to the student personal data form, where the clerk can update any additional information and finally save the data for the new student.

Functional Requirements

- Create/edit/delete student from data retrieved from the enrolment module

6 Interface with other systems

6.1 Interface to MoES

6.1.1 Functionalities of the web service

Description and priority

The system should enable web services to query data from the system and enable the Ministry of Education and Statistical Office to have up to date information.

Furthermore, the web service should enable exchange of information with external public agencies regarding students' data. One of these processes is the check for students' scholarship.

Functional Requirements

The functions and the structure of the response will be defined during the elicitation phase of the project in working groups from universities, Ministry of Education and Statistical office.

6.1.2 XML Web Services to external systems

The web services will have to return data in agreed XML format. The structure presented below is just for estimation purposes and will need to be defined in details during the elicitation phase.

6.1.3 Descrtiption of the Web service

Web service named *ExchangeData_MoES* is consumed by the MoES system from each university to get data for the enroled students. This web service is used by the Statsitical office to query the data for the statistical reports that the Faculties are obliged to provide at certain times. Using the methods from this web service, the external systems can obtain data for enroled students from the iKnow system at the universities. The external system qill have to implement a client for this web service.

Following class diagram gives the list of web methods in the *ExchangeData_MoES* web service.

There are several methods to query data from the WebService. If there is an exception , the web service returns *ExchangeDataError*, which explains the reasons for exception while transferring data from iKnow system.

ExchangeDataError is a class containing two fields: descrition and status.

The webservice *ExchangeData_MoES* implements the following methods:

Method Signature	**Description**
ExchangeData_MoES.ExchangeDataError[] GetStudentData(string StudenttID, string StudyProgram, ref string surname, ref string maidenSurname, ref string fatherName, ref string name, ref string embg, ref DateTime birthDate, ref string birthPlace, ref string birthManicipality, ref string birthState, ref string sex, ref string citizenship, ref string nationality, ref string livingPlace, ref string livingAddress, ref string livingManicipality, ref string livingState, ref string sPhoneNumber, ref string mPhoneNumber, ref string email, ref string nameSurname, ref string profession, ref string employment, ref string schoolName, ref string schoolProfile, ref string schoolManicipality, ref string schoolAverage, ref string fSchoolYear, ref string studyingLanguage, ref ExchangeData_MoES.GraduationType graduationType, ref string scholarship, ref string scholarshipProvider, ref int,)	The method returns the personal data, address information, contact information, information related to the previous education, data for the parent/guardian, other data and data for state/school/international graduation type (depending of the type of graduation taken by the candidate) for the enrolled student with StudentID. Otherwise the method's return value of type *ExchangeDataError[]* gives a list of reasons why the method can't transfer the requested data.
ExchangeEnrolledData. ExchangeDataError[] GetEnrolmentData*(the structure of the data should correspond to the structure of the reports that the MoES will be requesting from the universities at the moment of system design phase. Please refer to the annexed reports templates for estimation of work load)*	The method returns the study program, the number of enrolled students by the study program, the types of student enrolled and other fields as requested by M0ES at the moment of system design.
ExchangeStatisticalData. ExchangeDataError[] GetStatisticalData(*the structure of the data should correspond to the structure of the reports that the Statistical office will be requesting from the universities at the moment of system design phase. Please refer to the annexed statistical reports templates SV50, SV70,SV70a,SV80,SV80a and the web site http://www.stat.gov.mk/glavna.asp?br=96#образование for latest versions of the templates for estimation of work load*)	The method returns the statistical data as requested by Statistical office. at the moment of system design.
ExchangeDataExternal ExchangeDataError[] GetStudentData(string Name, string Surname, string embg, ref bool IsActiveStudent, ref string StudyProgram, ref int EarnedECTSCredits, ref double CurrentGPA,)	The method returns the basic data for an enrolled student. The data is queried either by Name, StudentID or embg and basic data for the student is returned.

6.2 Module for personal identification and access control

6.2.1 General overview

The main elements of the PID module are infrastructure for authentication and authorization AAI, along with the presence control infrastructure based on RFID or similar technology.

The AAI infrastructure should consist of 3 main elements:

- core AAI elements
- home institution components
- resource providers components

The cores AAI elements are:

- Central RADIUS proxy
- Central Web Service
- Central Single Sign On – SSO service
- Central Meta Database

Home institution components include:

- LDAP directory
- RADIUS server
- Web service for LDAP access and management (LDAPWS)

Resource provider components depend of the type of the resources. Since most of the resources are web applications and web services, the components include modules for different systems and applications performing the authentication and authorization process using this AA infrastructure.

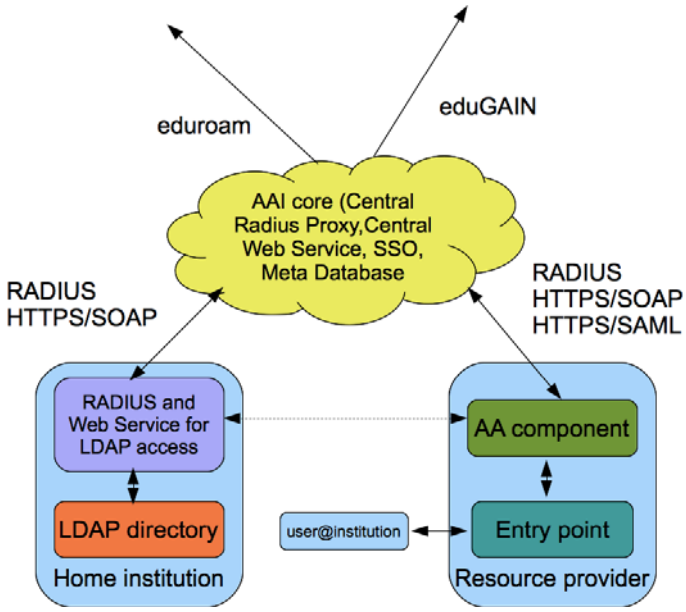
6.2.2 Core AAI elements and architecture

The role of core AAI elements is to provide simple, secure and reliable operation of the authentication and authorization process. The process is initiated by the user requiring access to some resource that is conformant with the underlying AAI standard. The resource could be network, web applications, other network based applications, etc. The core AA components will then contact the AA components at the home institutions that perform the authentication, but also provide other information from the electronic identity of the user that will be used by the resource providers in the authorization process.

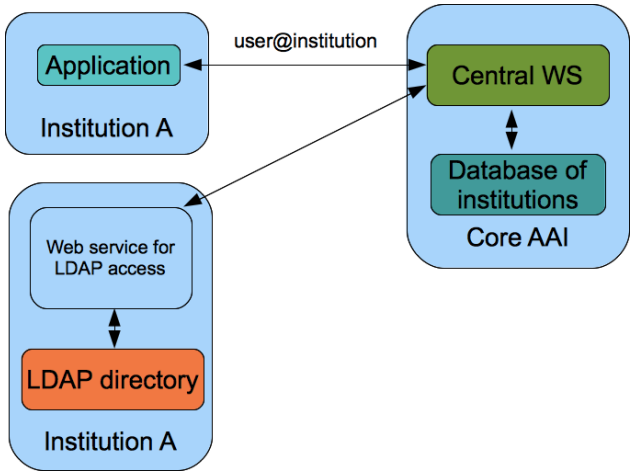
The elements of the AAI core and their functions are given in the table below

Element	Function
Central RADIUS proxy	Used in the authentication process as a relay between the resource providers and home institutions (can also be used for international authentication needs, such as eduroam)
Central Web Service	Enables authentication and authorization of the users using the HTTPS/SOAP protocols, useful in network applications that use HTTP and HTTPS protocols
Central SSO service	Enables single authentication using HTTPS/SAML protocols, based on the central WS. Can be used as interface or bridging element to external systems using Shibboleth standard.
Central Meta Database	Contains data for the elements comprising the system.

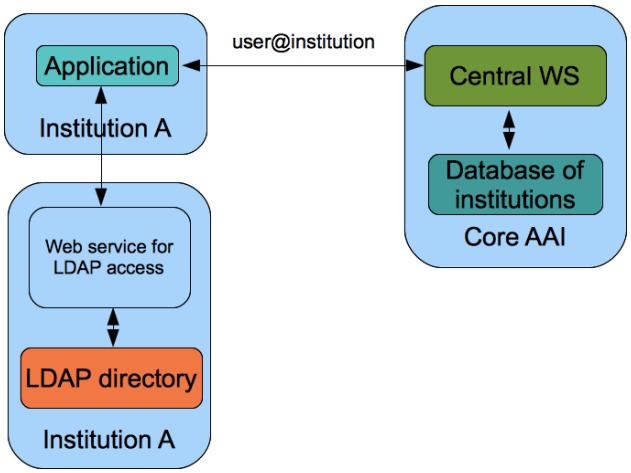
The architecture of the system is given in the figure below.



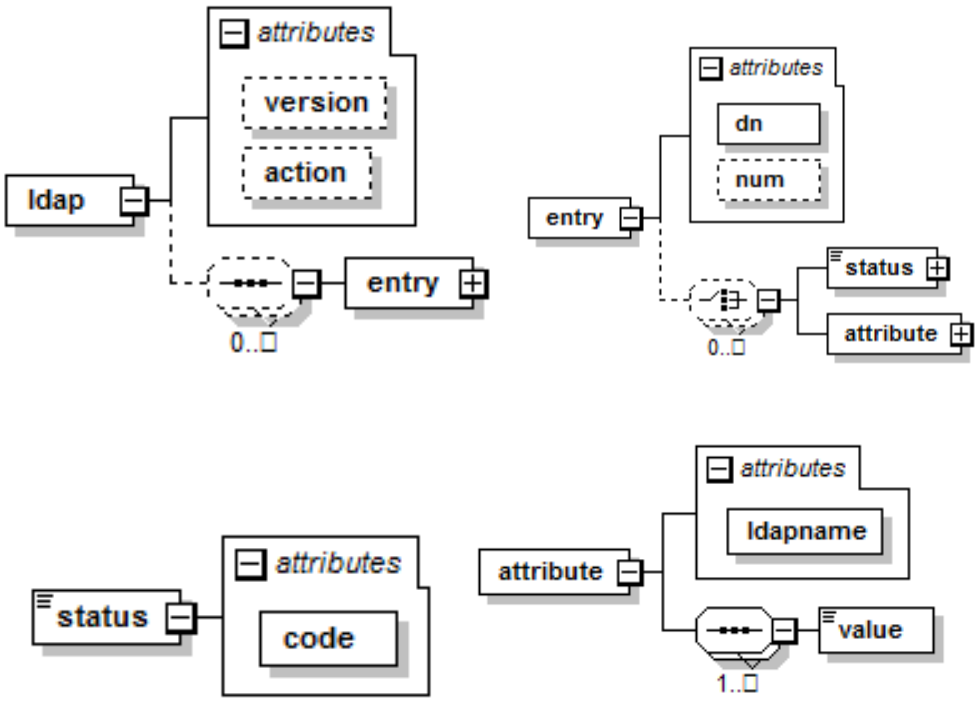
The Central Web Service provides the brokering and discovery element to the AAI. It is contacted by applications requiring user authentication. It then finds the home institution of the users (based on the format of the user credentials username@institution.edu.mk) in the Central Meta Database and forwards the request to the web service for LDAP access. The results of the equerry are then returned to the application that initiated the request. It can work in two modes: as a proxy and as a locator service. When in proxy mode, forwards the request for authentication to the home institution and backward to the initiating application. This mode is illustrated in the figure below.



When working as locator service, it returns the initiating application only the data about the location of the home institution, and the authentication process is done between the initiating application and the home institutions. This mode is illustrated in the figure below.

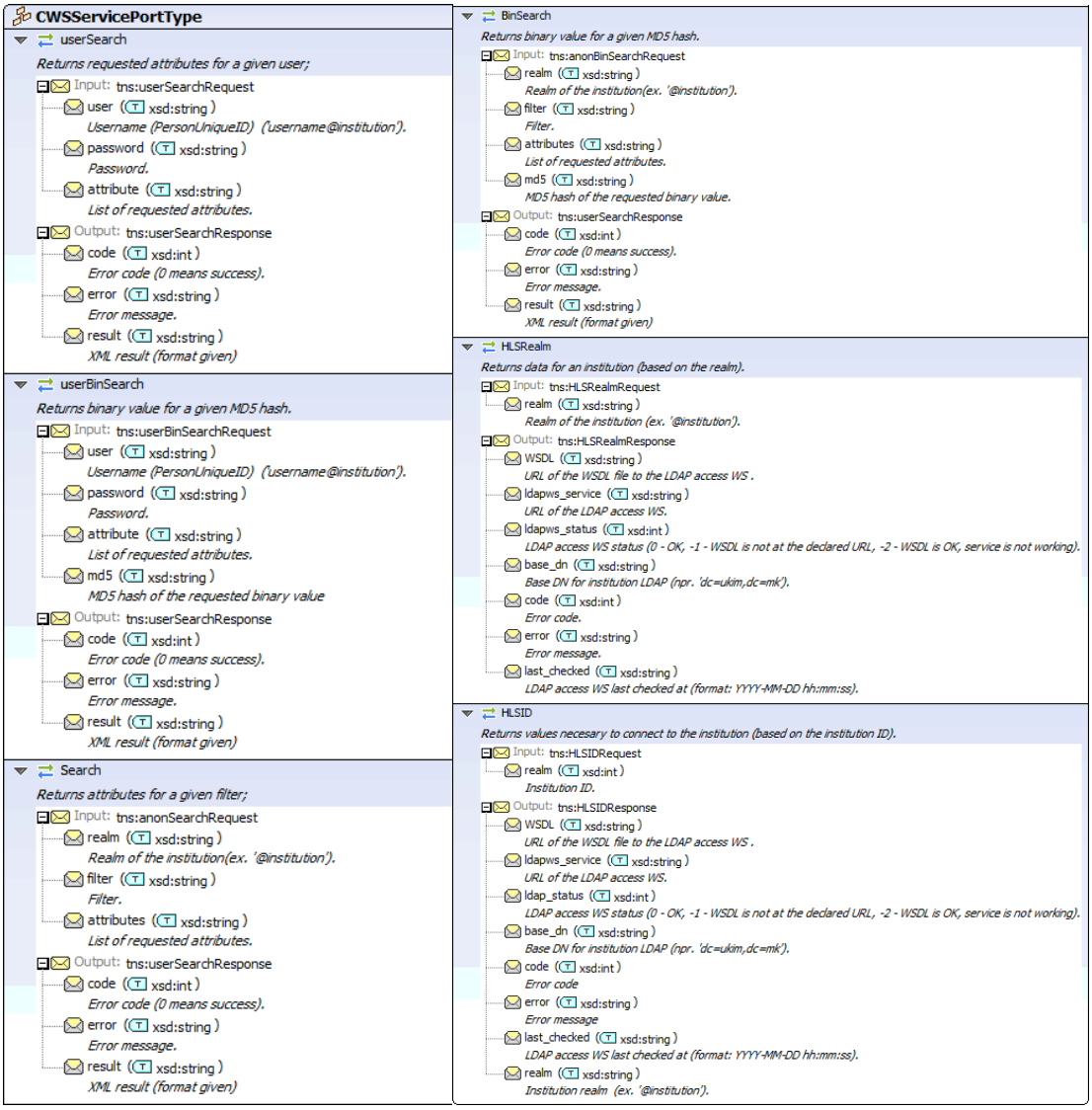


In both modes the response is in the following XML format.



6.2.3 Web service description

The description of the web service is given below.



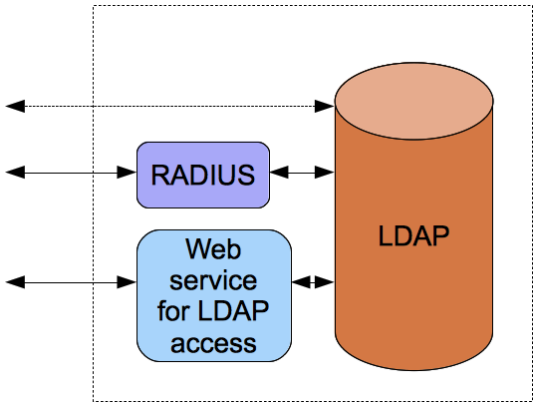
6.2.4 Home Institutions components

The home institution is always responsible for performing the authentication itself. This is the strongest point of such systems, providing single point of storing user identities. Moreover, using single sign on feature means that the users passwords will be entered only on a single, secure place, regardless of the resource/application that is being accessed. The elements of the infrastructure that are at the home institutions are given in the table below. Data for all the participants (students, faculty, administration) is kept in the same LDAP directory.

The important point of such distribution of roles and responsibilities is that the whole responsibility and ownership of the personal information is at the home institutions of the users. They are responsible to keep their user database accurate and up-to-date. Also, the resources providers have the possibility to authorize users based on well-defined criteria, applied to the necessary electronic identity data.

Element	Function
LDAP directory	Contains all the data describing the electronic identity of each user.
RADIUS server	Used in the process of authentication, making decisions based on the LDAP data.
Web service for LDAP access and management - LDAPWS	Used for maintenance of the users LDAP database. Primarily intended to be used from other applications for user management.
LDAP maintenance client	Client application that uses the web service for LDAP, used to manage the database, used if no other application is used (through the web service).

The architecture of the elements at the home institution is shown in the figure below.



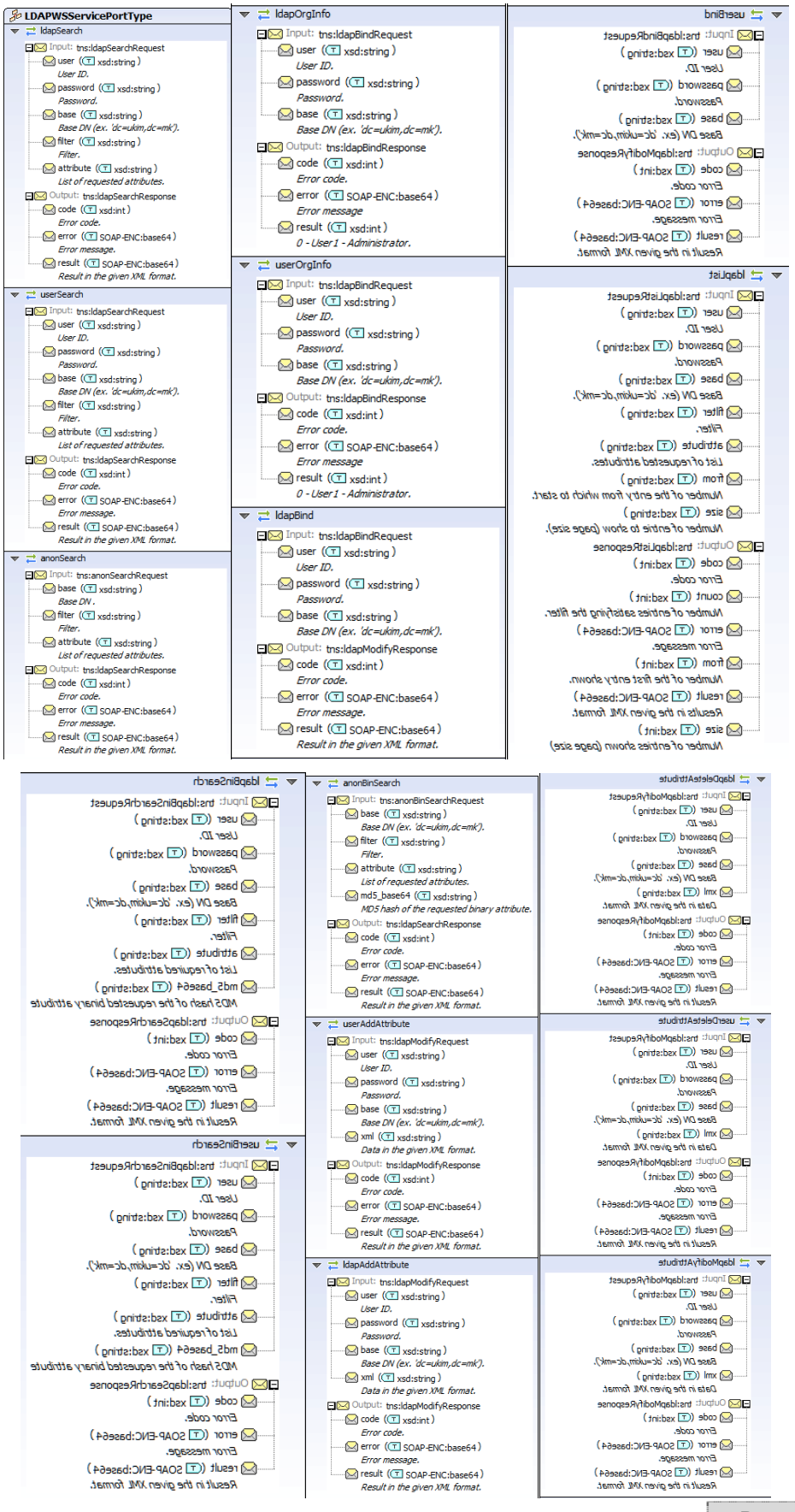
The web service for LDAP access and management is intended as an interface to the LDAP directory, simplifying the access to the data, due to:

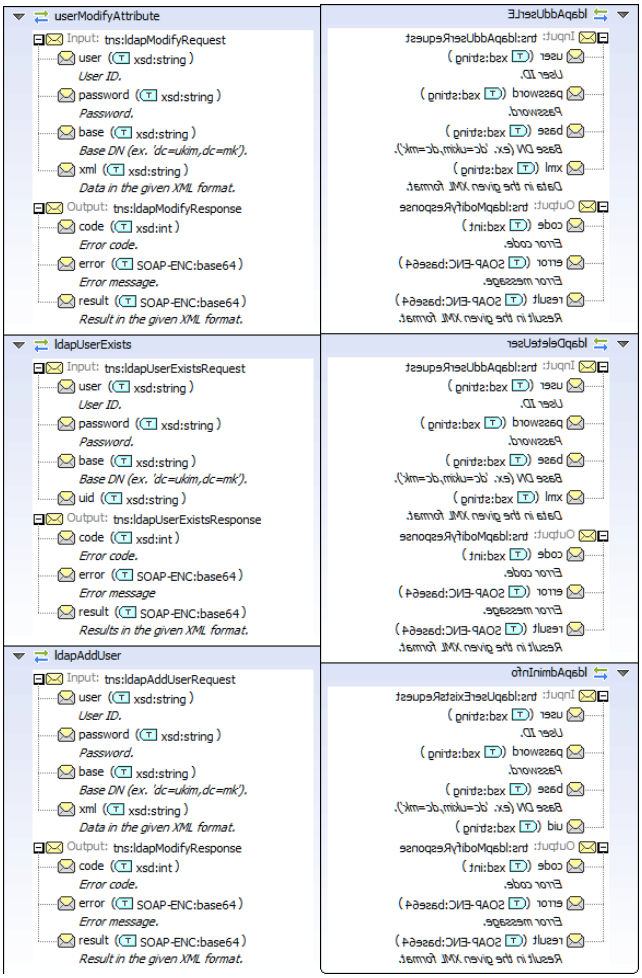
- SOAP is commonly supported in many languages and platforms, enabling application developed using them to have access to the LDAP data for authentication and authorization
- All the communication is done using XML, the de-facto standard for electronic data interchange
- Accessing LDAP through web service simplifies otherwise complicated structure

The communication is secure

- Enables multiple levels of security

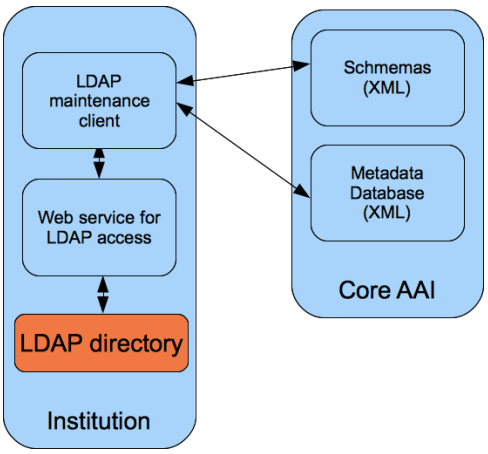
The description of this web service is given below.





This web service should also be used to synchronize the students records from the core modules and the LDAP databases, enabling only small subset of data to be kept in the LDAP (parameter needed for authentication and authorization), while the rest of the student record is part of the core modules.

The access to the web service could be directly from any application that implements the required WSDL, or using the front-end application – LDAP maintenance client. Its architecture is given below.



The LDAP structure should follow the recommended schemas for educational institutions. The schemas should cover both individuals and organizations. Elements of these schemas are given in the tables below.

Elements for individuals	Description
PersonUniqueID	Unique id in the form user@institution.mk
userPassword	Password (preferably base64 encoded)
lastName	Last name
givenName	First name
mail	Email
PersonUniqueNumber	Unique Number of the Citizen
PersonDateOfBirth	Date of birth
PersonGender	Gender
jpegPhoto	Photo (from the presence control system)
PersonProfessionalStatus	Status
PersonAcademicStatus	Academic status
PersonStudentCategory	Student category
PersonTitle	Title
PersonRole	Role
orgName	Organization name
PersonHomeOrg	Organization ID
PersonCardNum	e-ID card number

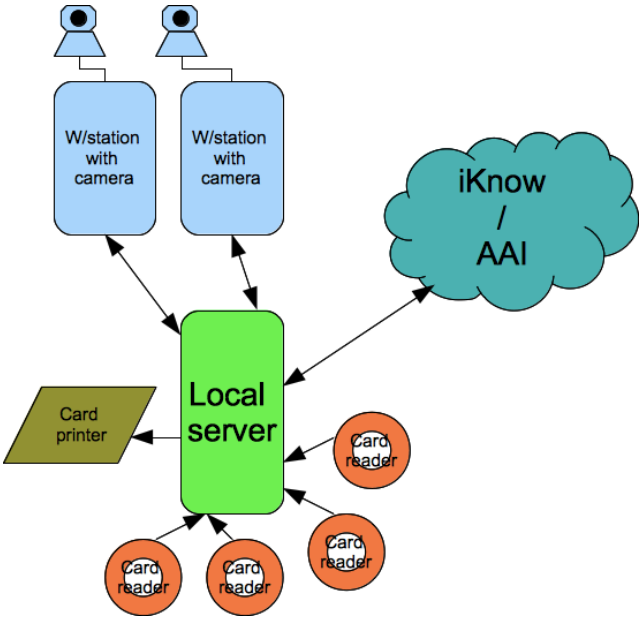
Elements for institutions	Description
orgName	Organization name
OrgUniqueNumber	Unique ID
postalAddress	Address
city	City
postalCode	Postal code
street	Street
telephoneNumber	Phone number
facsimileTelephoneNumber	Fax number
OrgMail	e-mail
OrgType	Type of organization
OrgURL	URL of the org. web site

The AA infrastructure should be supplemented with presence and/or access control infrastructure, based on the RFID or similar technology, presented in the next chapter.

6.2.5 Module for personal records of students

Part of the presence control system should include a module for production of student and staff e-ID cards and the maintenance of the student and staff records.

The architecture of the presence/access control system is given below.



The module for the production of e-IDs consists of

- a computer equipped with digital camera,
- software for data entry into the LDAP database of the home institution (or data exchange from the already entered student records through the enrolment module),
- software for e-ID visual editing and
- a web service to send the data to the printing facility.

The users are photographed; the photo is attached to the user data in the LDAP database, along with the record from the enrolment/core module. Next, the data is sent via web service to the printing facility (to optimize the cost, one printing facility can service multiple locations – faculties). The printed and initialized IDs are returned to the home institution of the user (student services for students, personnel services for the staff).

The data in the LDAP database is a subset of the data on the core module, containing the attributes necessary for the AA process. The data in both directories (LDAP and iKnow student records) should be synchronized periodically to avoid inconsistency.

Module for presence monitoring and student activities

To support the learning process, but also to comply with the laws, each university should keep track of the attendance of the students and teaching staff. The iKnow systems, using the data for the students and curriculum, should merge it with the data acquired from the presence and access control systems and create detailed attendance logs. These logs can be analyzed compared to the average marks, reviews and questionnaires to evaluate the learning process itself.

The network of readers will track student e-IDs on the entrance of classrooms. This data will then be sent over web service to the iKnow system, where it will be cross-referenced with the schedule to produce attendance logs. Also, if appropriate infrastructure for access control is in place, it will check the authorization through web service from the iKnow system.

The data that is sent contains student IDs, resource IDs (room number, lab number) and timestamp.

6.3 Module for electronic payment and use of resources

6.3.1 General overview

This module has 2 main roles:

- Financial calculations and interface to financial systems and
- Interface to other supporting systems and infrastructures in the daily operations of the university

Coordination of access and usage of other resources should be integrated with the iKnow systems through several interfaces. The first point of integration of all the internal systems should be the unified AAI. This means that all the systems will use the AAI for authentication and authorization of the access to the resources, whether through the single sing on or through web services.

The other points of integration depend on the resource in question. So far, we have identified four main groups of internal systems that can be integrated with the iKnow infrastructure.

6.3.2 Financials

The interface to financial systems provides the connection between the iKnow generated financial obligations and their actual realizations in the financial department. The iKnow systems can calculate the all the fees based on the pricelist and the students current course enrolment, year of study, number of credits, exams to be taken etc. The generated financial obligation is checked via web service with the financial department. It they give clearance, the student can proceed with the activity in question. If not, he/she will be notified to cover the expenses.

The iKnow financial elements are described in section 1.12.8 of the Core module.

We foresee 2 models for the clearance of the payments:

- pre-paid, where the students deposits some amount up-front, and then each new obligation is subtracted from that amount
- post-paid, where for each activity, new financial obligation is generated (invoice), and the financial department is queried if this specific obligation is paid.

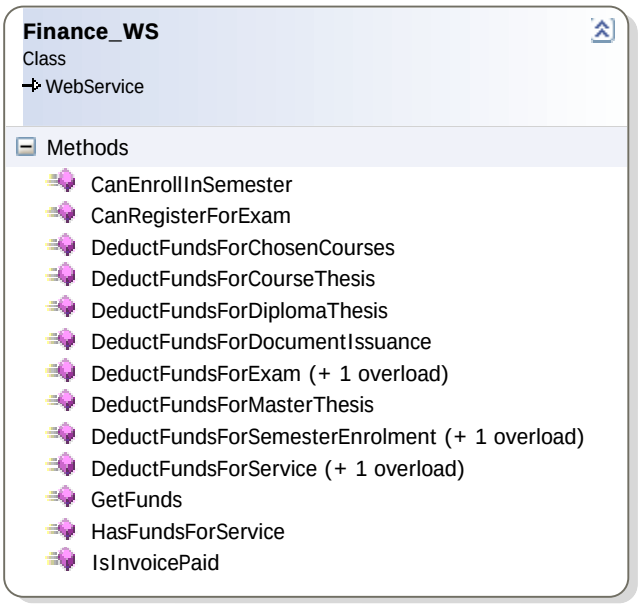
The description of the web service is given below.

6.3.3 Web service description

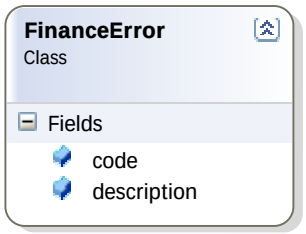
Web service Finance_WS is consumed by iKnow to get data on payments made by students. Using the methods from this web service, iKnow can enquire if a student has enough funds for a service e.g. to register for an exam, to enroll in a semester etc. , and whether the service should be rejected or accepted, and to inform the finance system to deduct funds from student's account. When funds are deducted, the amount is either passed as an argument to the finance system, or left to the finance system to calculate it. Each University can choose to continue using its existing finance system and build a web service as an addition to its finance system, or start using the finance module built as part fo the iKnow system and its corresponding web service.

More information on the functional description of the web service can be found in the section 2.2.2 Post-paid method of payment for stduent services is exaplined in that section. Support for the post-paid method is provided by the web method *IsInvoicePaid()*.

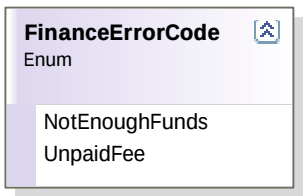
Following class diagram gives the list of web methods in the Finance_WS web service.



All web methods return an array of values from the type *FinanceError*, which explains the reasons to reject a service to the student. *FinanceError* is a structure containing two fields as shown in the following diagram:



Field *description* is of type *string*, while field *code* gets values from the enumeration *FinanceErrorCode*:



Web service *Finance_WS* implements the following methods:

Method Signature	Description
`double GetFunds(string StudentID, ref FinanceError[] finError)`	Get the current balance for the student *StudentID*.
`bool CanRegisterForExam(string StudentID, string examID, string examType, DateTime examRegistrationStartDate, DateTime examRegistrationEndDate, ref FinanceError[] finError)`	Check if student *StudentID* can register for exam *examID*. *examType* can be partial exam, final exam, etc. Student can register for exams between *examRegistrationStartDate* and *examRegistrationEndDate*. Returns *true* if the student can register for the exam. Returns *false* if the student can not register for the exam, and value *finError* of type *FinanceError[]* returns a list of reasons.
`bool CanEnrollInSemester(string StudentID, ref FinanceError[] finError)`	Check if student *StudentID* can enroll in semester. Returns *true* if the student enroll in semester. Returns *false* if the student can not enroll in semester, and value *finError* of type *FinanceError[]* returns a list of reasons.
`bool HasFundsForService(string StudentID, string serviceType, ref FinanceError[] finError)`	Check if student *StudentID* has enough fund to register for the service *serviceType*: inauguration, final thesis defence, etc. *serviceType* is an enumeration. Returns *true* if the student has enough funds to register for the service. Otherwise it returns *false*, and the value *finError* of type *FinanceError[]* returns a list of reasons.
`double DeductFundsForExam(string StudentID, string examID, string examType, DateTime examRegistrationStartDate,DateTime examRegistrationEndDate, ref FinanceError[] finError)`	Deduct funds from student account for exam registration. Deduction amount is calculated by finance system and returned.
`void DeductFundsForExam(string StudentID, string examID, string examType, double amount, ref FinanceError[] finError)`	Deduct funds from student account for exam registration. Deduction amount is passed to finance system in the parameter *amount*.
`double DeductFundsForSemesterEnrolment(string StudentID, ref FinanceError[] finError)`	Deduct funds from student account for semester enrolment. Deduction amount is calculated by finance system and returned.
`void DeductFundsForSemesterEnrolment(string StudentID, double amount, ref FinanceError[] finError)`	Deduct funds from student account for semester enrolment. Deduction amount is passed to finance system in the parameter *amount*.
`double DeductFundsForService(string StudentID, string serviceType, ref FinanceError[] finError)`	Deduct funds from the account for student *StudentID* for service *serviceType*. A comprehensive list of service types is given in section 1.9.28 in the CMF specification. Deduction amount is calculated by finance system and returned.
`void DeductFundsForService(string StudentID, string serviceType, double amount, ref FinanceError[] finError)`	Deduct amount *amount* from the account for student *StudentID* for service *serviceType*. Deduction amount is passed to finance system in the parameter *amount*.
`bool IsInvoicePaid(string StudentID, string InvoiceID, double amount, ref FinanceError[] finError)`	Check if invoice with ID *InvoiceID* has been paid by student *StudentID*. Check if the amount paid is *amount*. Returns *true* if the invoice is paid fully. Otherwise returns *false*.
`void DeductFundsForChosenCourses(string`	Deduct amount *amount* from the account for student *StudentID* for registration to attend

`StudentID, string[] chosenCourses, double amount, ref FinanceError[] finError)`	courses *chosenCourses*. Deduction amount is passed to finance system in the parameter *amount*.
`void DeductFundsForCourseThesis(string StudentID, string courseID, double amount, ref FinanceError[] finError)`	Deduct amount *amount* from the account for student *StudentID* for registration to prepare course thesis. Deduction amount is passed to finance system in the parameter *amount*.
`void DeductFundsForDiplomaThesis(string StudentID, double amount, ref FinanceError[] finError)`	Deduct amount *amount* from the account for student *StudentID* for registration to prepare diploma thesis. Deduction amount is passed to finance system in the parameter *amount*.
`void DeductFundsForMasterThesis(string StudentID, double amount, ref FinanceError[] finError)`	Deduct amount *amount* from the account for student *StudentID* for registration to prepare master thesis. Deduction amount is passed to finance system in the parameter *amount*.
`void DeductFundsForDocumentIssuance(string StudentID, string DocumentType, double amount, ref FinanceError[] finError)`	Deduct amount *amount* from the account for student *StudentID* for issuance of document. Deduction amount is passed to finance system in the parameter *amount*.

6.4 Email interface

6.4.1 General overview

This section describes the functionality between iKnow and the internal student email system. It is assumed that

- Upon enrollment a student will be given an email address from a university subdomain e.g. student.university.edu.mk, or
- A student will register a private email for all email notifications from the University

Most of the required functionality is about managing student membership in the distribution lists for University, faculties, cycles of studies, departments, school years, courses, etc.

We envisage the existence of the following distribution lists:

- All University students (US) e.g. Students
- All faculty students (FS) e.g. ComputerScience
- All department students (DS) e.g. e.g. SoftwareEng
- All students in a cycle of studies in a faculty (FSC) e.g. ComputerScienceFirstCycle
- All University students enrolled in a given school year (USY) e.g. Students2010
- All faculty students enrolled in a given school year (FSY) e.g. ComputerScience 2010
- All department students enrolled in a given school year (DSY) e.g. SoftwareEng2010
- All students in a cycle of studies in a faculty in a given school year (FSCY) e.g. ComputerScienceFirstCycle

All students enrolled in a course in a given school year (CSY) e.g. Cryptography2010

It is assumed that the distribution lists will be maintained by the email system, and not by iKnow. The only exception is that the CSY distribution lists are created by iKnow.

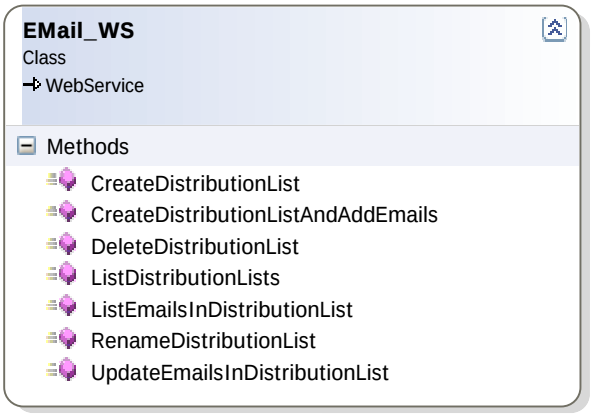
Following Web services will be implemented by the other internal systems and will be consumed by iKnow:

Web Service name	Description	Implemented At
EMail_WS	Implements functionality to add, modify and delete email distribution lists, and to add, modify or remove email addresses to distribution lists.	University implements this web service as an addition to its email system.

6.4.2 Web service description

Web service Email_WS implements functionality to add, modify and delete email distribution lists, and to add, modify or remove email addresses to distribution lists. iKnow consumes this web service. It is the responsibility of each University to implement this web service as an addition to its internal email system.

Following class diagram gives the list of web methods in the Email_WS web service:



Web service Email_WS implements the following methods:

Method Signature	Description
`bool CreateDistributionList(string strName)`	Create a distribution list with name *strName*.
`bool CreateDistributionListAndAddEmails(string strName, string[] eMailAddresses)`	Create a distribution list with name *strName* and add email addresses from the list *eMailAddresses* to the distribution list.
`bool DeleteDistributionList(string strName)`	Delete the distribution list with name *strName*.
`bool UpdateEmailsInDistributionList(string strName, string[] eMailsToAdd, string[] eMailsToRemove)`	For distribution list with name *strName*, add the email addresses from the list *eMailAddresses* and remove the email addresses from the list *eMailsToRemove*.
`string[] ListEmailsInDistributionList(string strName)`	Return a list of the email addresses in the distribution list with name *strName*.
`string[] ListDistributionLists()`	Return a list of distribution list.

The authentication for the email infrastructure is expected to be performed via the proposed AA infrastructure, leveraging the user management process.

6.5 Library interface

6.5.1 General overview

The interaction with University library systems can be viewed as a two-way communication, as described below.

The LMS → ULS

- The student asks for an information if a book is recommended by a course is available in the Library, uses the link from LMS to the ULS
- The student looks upon a recommended book by a course and requires information about other books from the same author that ULS possess, uses the link from LMS to the ULS

The CORE → ULS

- Faculty administration requires information if a student has returned the books overdue more than a year BEFORE he/she is allowed to enroll the new semester.
- Faculty administration requires information if a student has returned the books overdue more than a year BEFORE he/she is allowed to take a semester off
- Faculty administration requires information if a student has returned all the books BEFORE he/she is allowed to graduate/ leave the studies
- Faculty administration sends information to the Library System that the student has graduated/ leaved the studies in order to deactivate its account.

The description of the Library web service is given below

6.5.2 Web service description

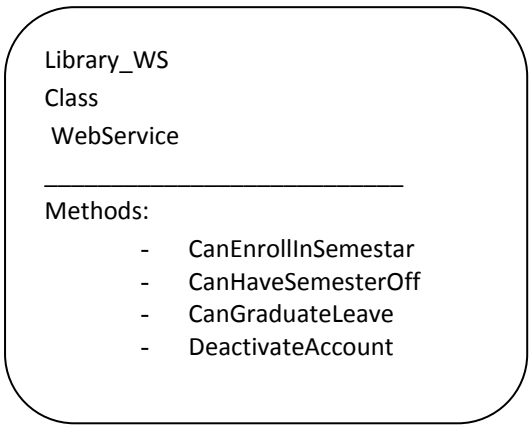
This section describes the technical interface between iKnow and other internal systems run at University.

Following Web service will be implemented by the other internal systems and will be consumed by iKnow:

Web Service name	Description	Implemented At
ULS_WS	Enquires information on borrowed books by the students in order to finish the course or stydy and whether the service should be rejected or accepted depending on the book status within a course	University implements this web service as an addition to its library system.

Web service Library_WS is consumed by iKnow to get data if the students return the borrowed books on time from the Library system. Using the methods from this web service, iKnow can enquire if a student has returned books that he/she owes more than a year in order to enable the student to enroll in a semester or take a semester off. Also, a method controlls if a student has returned all the books in order to allow her/his graduation or leaving the University. Each University is expected to implement this web service as an addition to its Library system (ULS).

List of web methods:



Web service Library_WS implements the following methods:

Method Signature	Description
`bool CanEnrollInSemestar (string StudentID, integer DuePeriod)`	Returns True if a student can enroll in a semester depending if the student's borrowed books aren't due more than a year, i.e. DuePeriod<365
`bool Can CanHaveSemesterOff (string StudentID, integer DuePeriod)`	Returns True if a student can take a semester off depending if students borrowed books aren't due more than a year, i.e. DuePeriod<365
`bool CanGraduateLeave (string StudentID)`	Returns True if all the borrowed book are returned to the Library
`bool DeactivateAccount (string StudentID)`	Returns true if the Library system has succesfuly deactivated the student account after her/his graduation or deserting the University.

6.6 Other interfaces

6.6.1 Human resources

Many elements in the iKnow infrastructure can be very useful to the HR department. The process of election and nomination of professors and other teaching staff depends and is evaluated based on the teaching engagements, among other criteria. Also, especially at the state universities, part of the salary is calculated based solely on the current engagement parameters like number of subjects, classes, students etc. Providing a web service that could deliver past and present engagements data to HR department will add additional value to the iKnow infrastructure, having in mind that the data is already there and up-to-date.

6.6.2 LMS

The purpose of these systems is to:

- Import the enrolled students from e - Index to LMS per course
- Export data from the LMS in e - Index for the realized activities and student's grades.
- Prepare reports required of the teachers and the University's management

The implementation of this interface should be done based on web services and according to the LMS platform in place at each university.