

SW REQUIREMENTS AND SYSTEM DESIGN

Security and interface to other internal systems

Author	Checked by	Description
Marjan Gusev		Initial idea and drafting of e-Students information system
Anastas Misev	Marjan Gusev	AAI and interface to presence and access control systems
Toni Stojanovski	Anastas Misev	Interface to financial systems and email
Ana Madevska Bogdanova	Anastas Misev	Interface to library systems
Zoran Zdravev	Anastas Misev	Interface to LMS
Toni Stojanovski	Anastas Misev	New version of the Finances interface
Ana Madevska	Anastas Misev	New version of Library interface

CIRCULATION LIST

This document is intended only for participants of i-Know Project.

REVISIONS SUMMARY

Revision	Date	Author	Modified paragraphs and kind of modification
Draft.1.0	16/10/2010	Marjan Gusev	Initial idea and elaboration of the project
Draft 2.5	27/01/2011	Anastas Misev	Version for the Bitola w/shop
Draft 3.0	06/02/2011	Anastas Misev	Final draft according to agreement
Draft 3.1	09/02/2011	Anastas Misev	Final draft after the Skopje workshop

LEGAL NOTICE

This document is prepared in the scope of the Innovation and Knowledge Management towards eStudent Information System TEMPUS Project JPGR 511342 – iKnow.

CONTENT

1 (SI) Security and interfaces to other internal systems	4
2 Introduction	4
2.1 Objective	5
2.2 General Overview	5
2.3 Scope	5
2.4 Existing System(s)	5
2.5 Benefits	6
2.6 Goals	6
2.7 Users	7
2.8 Basic functionalities	7
2.9 Detailed description	8
2.9.1 (PID) Module for personal identification and access control	8
2.9.2 (REC) Module for personal records of students	17
2.9.3 (PRS) Module for presence monitoring and student activities	17
2.9.4 (EPY) Module for electronic payment and use of resources	18
Financial	18
LMS	21
Email	21
Library	22
Human resources	24
2.10 UML	24
Financial	24
LMS	25
Email	26
Library systems	31

1 (SI) Security and interfaces to other internal systems

(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)

2 Introduction

Establishment of sound AAI infrastructure is a key element for supporting any heterogeneous system with mass usage. The AAI infrastructure solves the problem of the users having multiple identities (user/password or other forms of identification) for each of the resources they would like to access. This problem is even greater when the users and the resources belong to different administration domains (institutions).

The basic AAI infrastructure consists of 3 basic elements: users, identity providers and resource providers. The main purpose of such infrastructure is to provide unique identification of all the involved users (students, faculty, administration) and their authorization to use the available resources. The infrastructure should be based on the following qualifications:

- The management of the user databases used for authentication will be done at the institutes responsible for the particular users (their home institutions)
- The management of the resource databases used for authorization will be done at the institutions owning the particular resources
- There will be no central repository, neither for users nor for resources (meaning that there will be no central responsibility for the users and resources, only for the operation of the infrastructure)

The AAI system will enable easy user mobility (both student and staff), which is in accordance to the current trends in education (Bologna declaration). The infrastructure enables authentication and authorization for usage of various resources, such as networks (fixed or wireless), network services (e-mail, ftp), web based applications (LMS, CMS), reference resources (libraries). It also enables integration with pan-European systems such as edu roam (wireless roaming), GEANT federated identities, etc.

The infrastructure should be built on open standards and protocols, enabling easy maintenance and expanding.

Combining the AAI with the appropriate presence and access control technologies extends the reach of this infrastructure.

When properly deployed, this infrastructure will enable detailed evidence on the students and teachers presence at the lectures, eliminating subjectivity and supporting the teacher-student interaction. The data provided by the presence control system can easily be cross-referenced with each student schedule, making the systems also applicable as resource access control system.

Given the complexity of the operations of University IT systems, the iKnow system does not intend to solely cover every aspect of these operations. The general idea is to connect to the

other present and future internal and external systems and exchange the necessary information, making a complex network of IT services supporting the Universities daily operations.

Interfacing financial systems is a very important element of such integrated system, since it gives the sustainability perspective to the whole education process.

Interfaces to other internal systems, like communication infrastructure (email, IM) extends the communication capabilities of the iKnow infrastructure.

Other internal systems, like Library and Learning Management Systems are very important parts of the whole University ICT infrastructure, so the need to be integrated and interfaced to the iKnow systems also.

2.1 Objective

The general objective of these modules is to fit the new system into the already present (or the future developed) infrastructure. Integrating these functionalities will add value to the proposed system, covering the whole set of university daily operations.

2.2 General Overview

The system is intended to be used by the students; academic and administrative staff at all the levels (University, faculty, department) and all the resource providers outside of the University (libraries, etc).

2.3 Scope

The proposed 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).

Integration other internal systems gives a complete view to the users data from many different perspectives.

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.

2.4 Existing System(s)

At the moment, there is no AA infrastructure established in Republic of Macedonia. There are smaller infrastructures at some of the Universities (for example UGD Stip), but they are local infrastructures. 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 and finance management systems. Some of them have different forms of presence and/or access control systems.

Comment [11]: Every participant of the project can fill a block here

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 integration 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 systems. Interfacing to these systems is of great importance and is crucial for the success of the proposed system.

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) exist several years and now is a standard component of the university. E – index although is at an early stage of implementation works flawlessly.

2.5 Benefits

The benefits of such system are multiple. Regarding the users, it will enable simple and efficient mechanism for their authentication and authorization for multiple, heterogeneous systems. It will also integrate all the 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 educations (identified by the member institutions and/or services).

The University Library System (ULS) is a standalone resource, but its connection with the students through their courses as a course material provider is essential for the knowledge transfer in every educational system. The idea is to enable direct automatic connection with the courses developed in the different curricula in the University. If the library interface manages to enable faster information about the required course material, the student's academic success will rise. On the other hand, it is very important that the course materials are returned on time, so more students will have the opportunity to extend their knowledge on the given subject. The Library interface will provide this benefit.

2.6 Goals

The main goal of the proposed modules is to provide interfaces to all existing and future systems in the University, complementing the iKnow infrastructure to an even more powerful system.

The proposed AA infrastructure is simple, yet robust and secure mechanism for user authentication and authorization toward various shared resources. Some of the goals of the system are listed below.

- 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
- Send accounting reports to university management

The general goal of such interfaces is to integrate the existing and future build systems into a single infrastructure, providing an integrated view of all the data needed to perform universities daily activities.

Interfacing the library systems could achieve the following goals

- Control the library resources
- Better communication and knowledge-transfer
- Enhance the accessibility to the teaching materials

- Provide the priority in using library resources

The general goal of such interfaces is to integrate the existing and future build systems into a single infrastructure, providing an integrated view of all the data needed to perform universities daily activities.

2.7 Users

The following users and groups are to be realized:

ID	User groups	Roles
[1]	Student users (STU)	Access various resources using single credentials Be identified for presence at lectures using RFID Authenticate and authorize in home and visiting institutions (mobility)
[2]	Faculty staff (FAC)	Access resources using single credentials Authenticate and authorize in home and visiting institutions (mobility)
[3]	Faculty administrations (ADM)	Manage the student and staff accounts using standard LDAP schemas Enable and revoke privileges
[4]	Resource administrators (RES)	Define access to their resources using standard LDAP schemas
[5]	University computing center (UCC)	Maintain and monitor the infrastructure Provide hosting for the AAI services for smaller membering institutions.
[6]	Faculty computing center (FCC)	Maintain the users/resource database (LDAP and associated servicer) Maintain the identities of the users (identity providers)
[7]	University Library System (ULS)	Access the Library resources and student's library accounts
[8]	Finance Department (FIN)	Provide data about payments
[9]	HR Department (HRD)	Require data on the past and current engagements of the teaching staff.

2.8 Basic functionalities

Basic functionalities of the AAI infrastructure are specified as follows:

ID	Functionality	Description	Objective	User
[1]	Establish AAI infrastructure	Establish the necessary infrastructure for supporting the AAI	Make the initial infrastructure	UCC
[2]	Maintain AAI infrastructure	Maintain the central point of the infrastructure, LDAP schemas, and registry of member institutions, software repository.	Keep the infrastructure operational	UCC
[3]	Support the member institutions	Provide support to FCC to establish the identity providers and resource providers	Establish identity and resource providers	UCC, FCC
[4]	Manage users identities	Open accounts for students and faculty staff, manage levels of authorization, revoke accounts	Define the single user identity	ADM
[5]	Manage resource access	Define access rights for own resources with respect to user roles and	Define the access policies	RES

ID	Functionality	Description	Objective	User
		responsibilities		
[6]	Synchronize AA data with Core module data	Maintain synchronized student records between LDAP directories and core modules	Maintain student records	ADM
[7]	Access student resources	Using single credential, access all the needed resources.	Simplify the access model to resources	STU
[8]	Access faculty resources	Using single credential, access all the needed resources.	Simplify the access model to resources	FAC
[9]	Staff and students mobility	Access resources in visiting institutions using home identities	Enable roaming access	STU, FAC
[10]	Presence identification	Using RFID or other means, identify physical presence of students	Presence identification	STU
[11]	Issuing of personal ID cards – student ID	Take pictures, compose layout and print student e-ID cards	Establish e-ID for the students	ADM
[12]	Maintain student records	Manage the students' records.	Student records.	ADM
[13]	Track classes attendance	Using the schedule and the presence records, track the attendance to lectures	Attendance logs	ADM
[14]	Establish mailing lists	Using the data about the course enrolment, establish appropriate mailing lists	Email communication infrastructure	ADM
[15]	Get financial info for students	Check the payments from students for their invoices for services	Check financial records	FIN
[16]	Generate invoice for service	Generate invoices based on the fees for each type of service (semester, exam, penalties, ...)	Generate financial documents	ADM
[17]	Exchange data with LMS	Populate LMS courses, extract grades	Data exchange with LMS	ADM
[18]	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	STU
[19]	Manage ULS resources	Define rules for returning the borrowed books in different administrative situations.	Control the ULS resources	ADM
[20]	Acquire HR information	Get information for the needs of the HR systems, on the engagements of the staff	Support HR	HRD

2.9 Detailed description

2.9.1 (PID) Module for personal identification and access control

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.

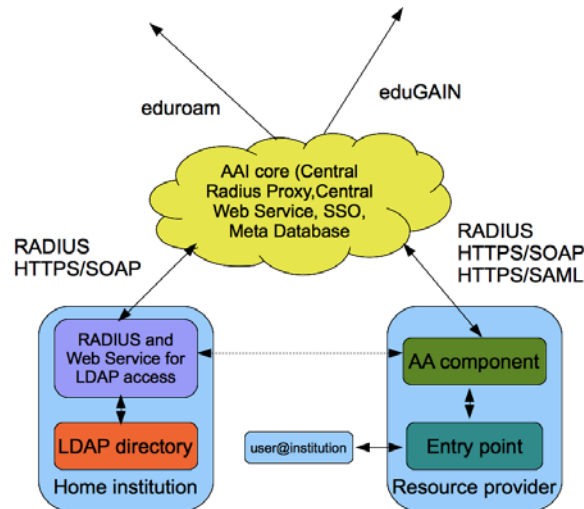
Core AAI elements and architecture

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.

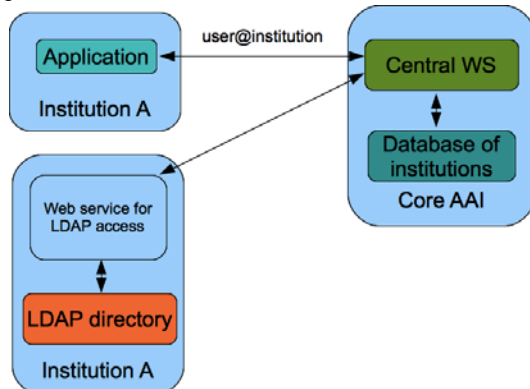
The elements of the AAI core and their functions are given in the table below

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.

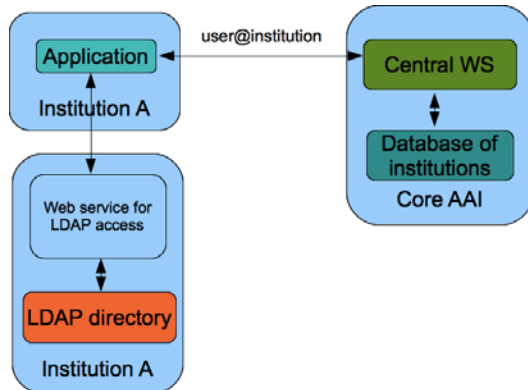
The architecture of the system is given in the figure below.



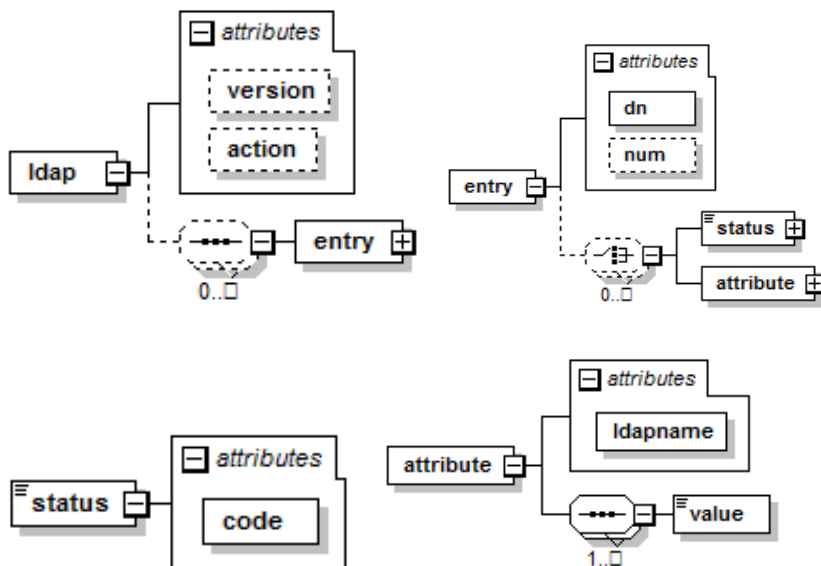
The Central Web Service provides the brokering and discovery element to the AAI. It is contacted by applications requiring user authentication. It then finds the home institution of the users (based on the format of the user credentials username@institution.edu.mk) in the Central Meta Database and forwards the request to the web service for LDAP access. The results of the query 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.



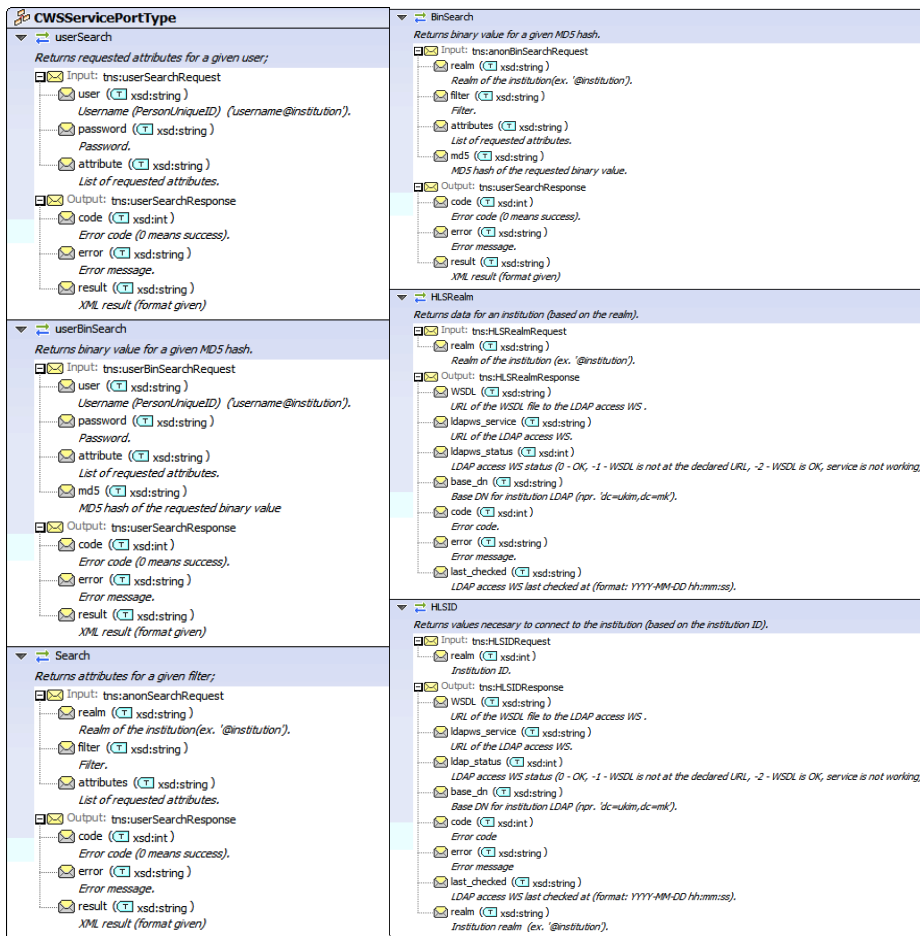
When working as locator service, it returns the initiating application only the data about the location of the home institution, and the authentication process is done between the initiating application and the home institutions. This mode is illustrated in the figure below.



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



The description of the web service is given below.



Home Institutions components

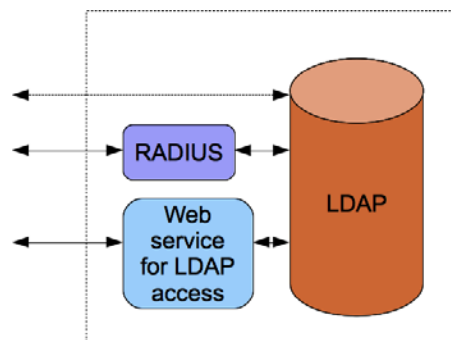
The home institution is always responsible for performing the authentication itself. This is the strongest point of such systems, providing single point of storing user identities. Moreover, using single sign on feature means that the users passwords will be entered only on a single, secure place, regardless 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,

Element	Function
	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

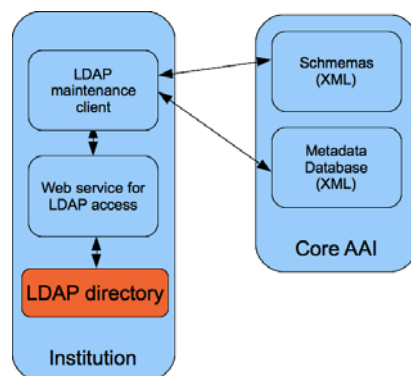
The description of this web service is given below.





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.

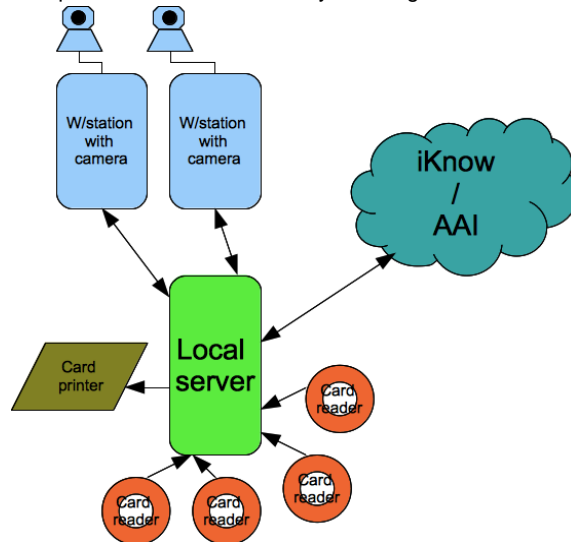
Elements for individuals	Description
PersonUniqueID	Unique id in the form user@institution.mk
userPassword	Password (preferably base64 encoded)
lastName	Last name
givenName	First name
mail	Email
PersonUniqueNumber	Unique Number of the Citizen
PersonDateOfBirth	Date of birth
PersonGender	Gender
jpegPhoto	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.

2.9.2 (REC) 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.

2.9.3 (PRS) 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.

2.9.4 (EPY) Module for electronic payment and use of resources

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 sign 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.

Financial

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. If 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.

Finance_WS

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 1.9.28 Administration of financial services of the CMF specification. Post-paid method of payment for student services is explained 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.



□

<p><code>bool CanEnrollInSemester(string StudentID, ref FinanceError[] finError)</code></p>	<p>Check if student <i>StudentID</i> can enroll in semester. Returns <i>true</i> if the student enroll in semester. Returns <i>false</i> if the student can not enroll in semester, and value <i>finError</i> of type <i>FinanceError[]</i> returns a list of reasons.</p>
<p><code>bool HasFundsForService(string StudentID, string serviceType, ref FinanceError[] finError)</code></p>	<p>Check if student <i>StudentID</i> has enough fund to register for the service <i>serviceType</i>: inauguration, final thesis defence, etc. <i>serviceType</i> is an enumeration. Returns <i>true</i> if the student has enough funds to register for the service. Otherwise it returns <i>false</i>, and the value <i>finError</i> of type <i>FinanceError[]</i> returns a list of reasons.</p>
<p><code>double DeductFundsForExam(string StudentID, string examID, string examType, DateTime examRegistrationStartDate, DateTime examRegistrationEndDate, ref FinanceError[] finError)</code></p>	<p>Deduct funds from student account for exam registration. Deduction amount is calculated by finance system and returned.</p>
<p><code>void DeductFundsForExam(string StudentID, string examID, string examType, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct funds from student account for exam registration. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>
<p><code>double DeductFundsForSemesterEnrolment(string StudentID, ref FinanceError[] finError)</code></p>	<p>Deduct funds from student account for semester enrolment. Deduction amount is calculated by finance system and returned.</p>
<p><code>void DeductFundsForSemesterEnrolment(string StudentID, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct funds from student account for semester enrolment. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>
<p><code>double DeductFundsForService(string StudentID, string serviceType, ref FinanceError[] finError)</code></p>	<p>Deduct funds from the account for student <i>StudentID</i> for service <i>serviceType</i>. 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.</p>
<p><code>void DeductFundsForService(string StudentID, string serviceType, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for service <i>serviceType</i>. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>
<p><code>bool IsInvoicePaid(string StudentID, string InvoiceID, double amount, ref FinanceError[] finError)</code></p>	<p>Check if invoice with ID <i>InvoiceID</i> has been paid by student <i>StudentID</i>. Check if the amount paid is <i>amount</i>. Returns <i>true</i> if the invoice is paid fully. Otherwise returns <i>false</i>.</p>
<p><code>void DeductFundsForChosenCourses(string StudentID, string[] chosenCourses, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for registration to attend courses <i>chosenCourses</i>. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>
<p><code>void DeductFundsForCourseThesis(string StudentID, string courseID, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for registration to prepare course thesis. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>
<p><code>void DeductFundsForDiplomaThesis(string StudentID, double amount, ref FinanceError[] finError)</code></p>	<p>Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for registration to prepare diploma thesis. Deduction amount is passed to finance system in the parameter <i>amount</i>.</p>

<code>void DeductFundsForMasterThesis(string StudentID, double amount, ref FinanceError[] finError)</code>	Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for registration to prepare master thesis. Deduction amount is passed to finance system in the parameter <i>amount</i> .
<code>void DeductFundsForDocumentIssuance(string StudentID, string DocumentType, double amount, ref FinanceError[] finError)</code>	Deduct amount <i>amount</i> from the account for student <i>StudentID</i> for issuance of document. Deduction amount is passed to finance system in the parameter <i>amount</i> .

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.

Email

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. 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.

Email_WS

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:

4. 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
5. Faculty administration requires information if a student has returned all the books BEFORE he/she is allowed to graduate/ leave the studies
6. 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

Library Web service

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 controls 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:

Library_WS
Class
WebService

Methods:

- CanEnrollInSemestar
- CanHaveSemesterOff
- CanGraduateLeave
- DeactivateAccount

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 succesfully deactivated the student account after her/his

	graduation or deserting the University.
--	--

Human resources

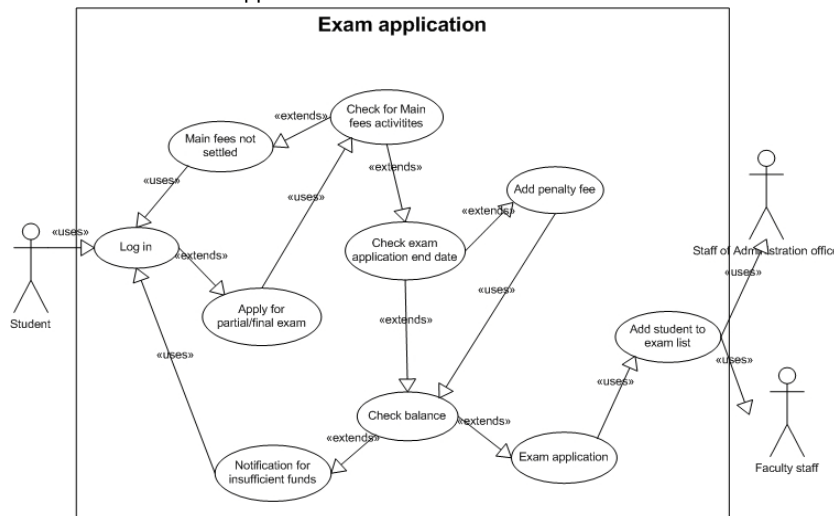
Many elements in the iKnow infrastructure can be very useful to the HR department. The process of election and nomination of professors and other teaching staff depends and is evaluated based on the teaching engagements, among other criteria. Also, especially at the state universities, part of the salary 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.

2.10 UML

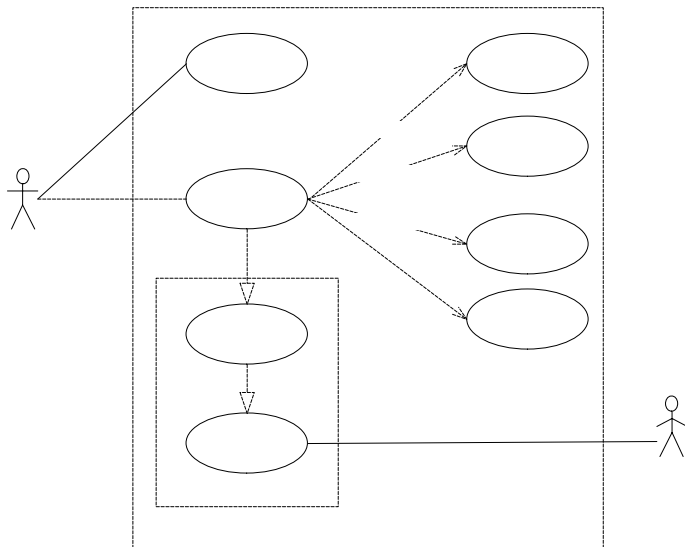
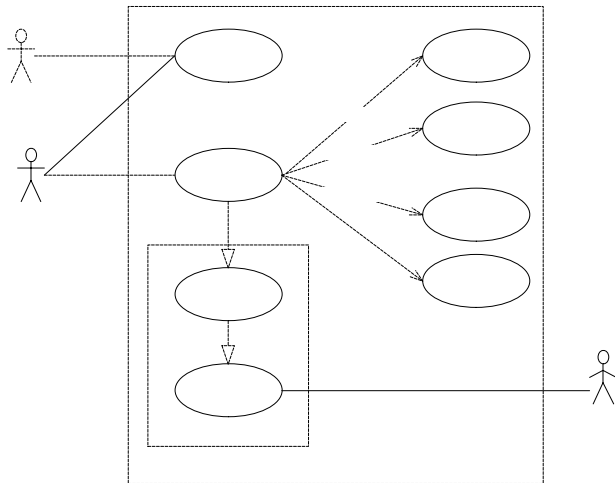
Financial

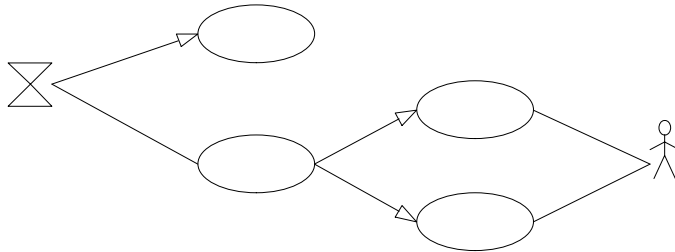
Use Case 1. The student uses the system to apply for partial/final exam.

- 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.



LMS

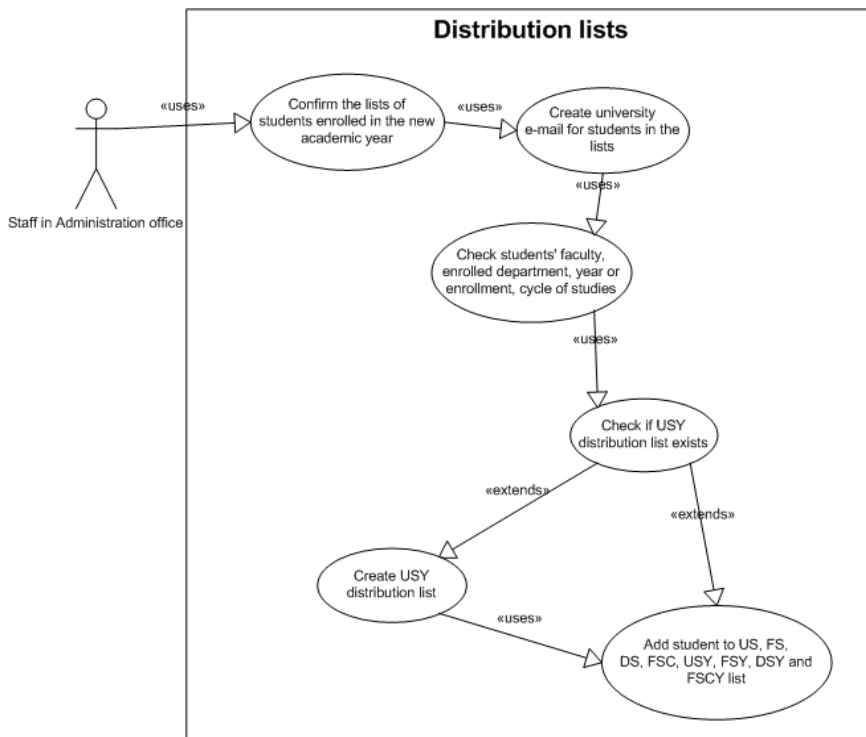




Email

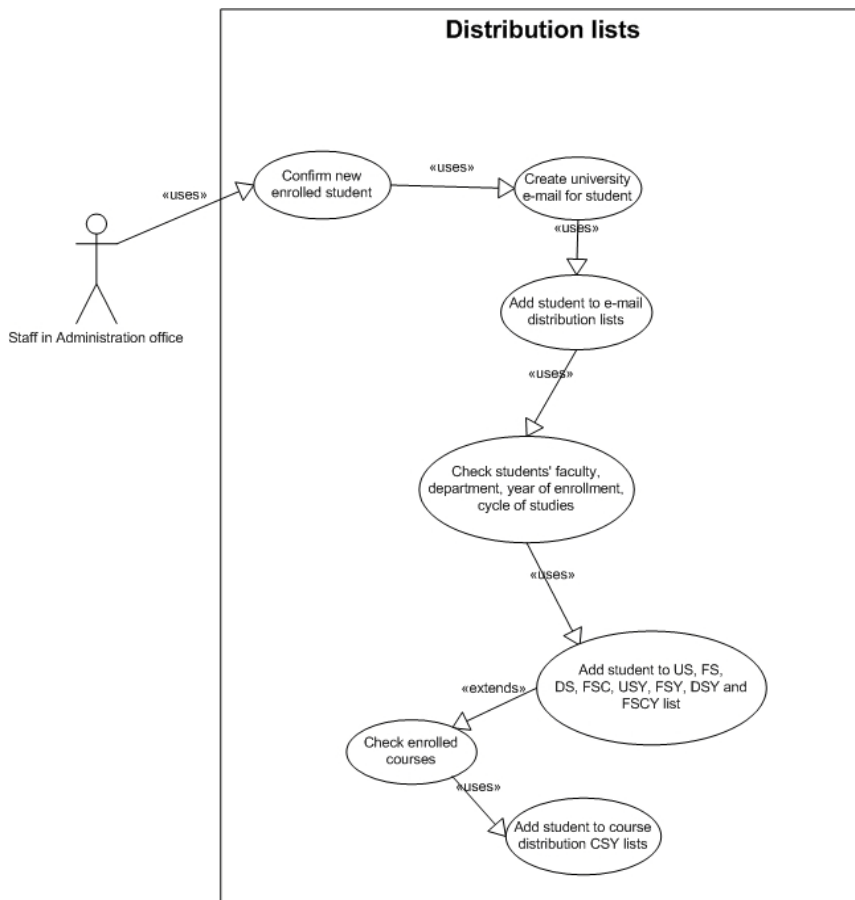
Use Case 1. Annual distribution list update

- 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.



Use Case 2. Update of distribution lists for late enrolment

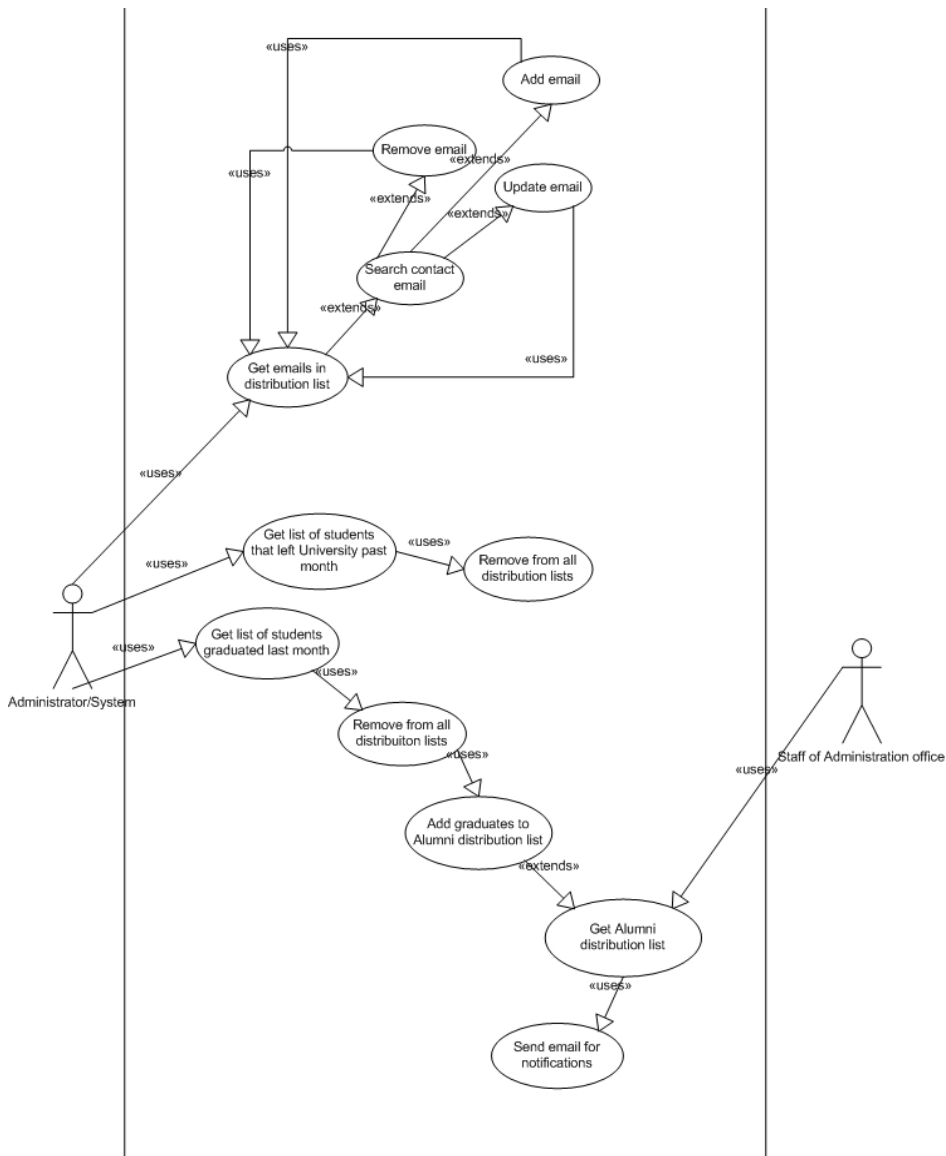
- Staff from the Administration Office confirms the list of students enrolled late (after deadline) in the current academic year.
- The system uses this info to create each new student an individual University email.
- Late admission students are added to all relevant distribution lists.
- Late admission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



Use Case 3. Administrator access to distribution lists

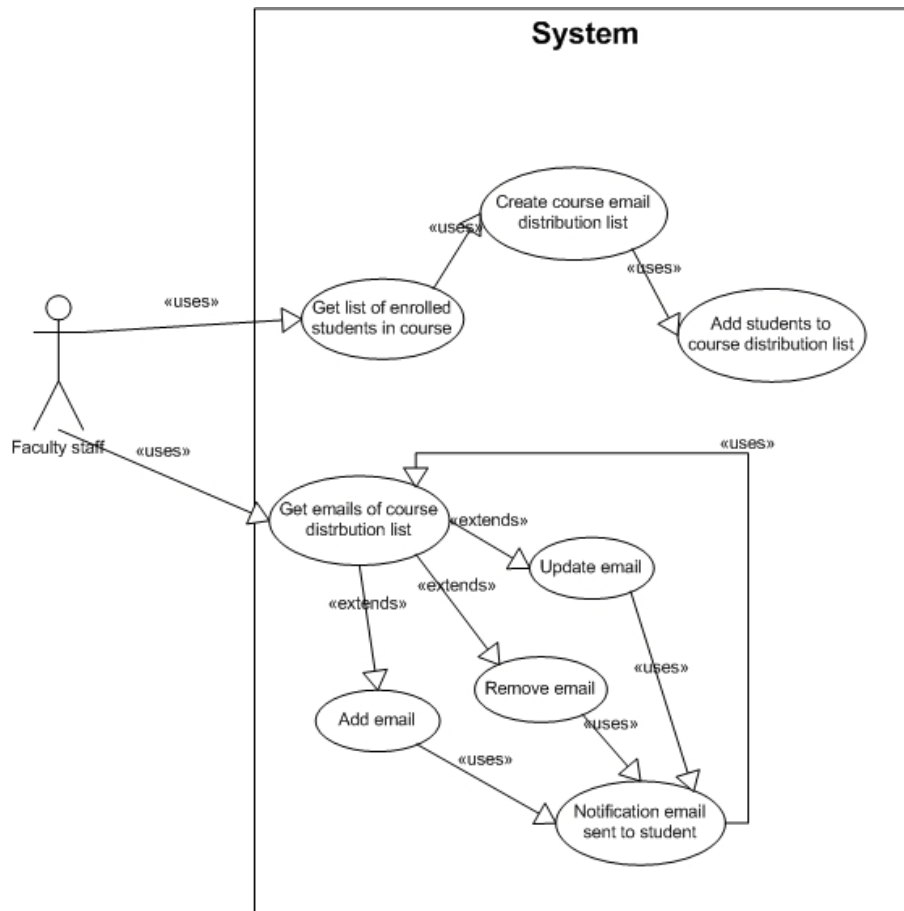
- Administrator should be able to manually add/update/remove emails from distribution lists
- Administrator should be able to remove students that have left the University/graduated from all distribution lists

- 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.

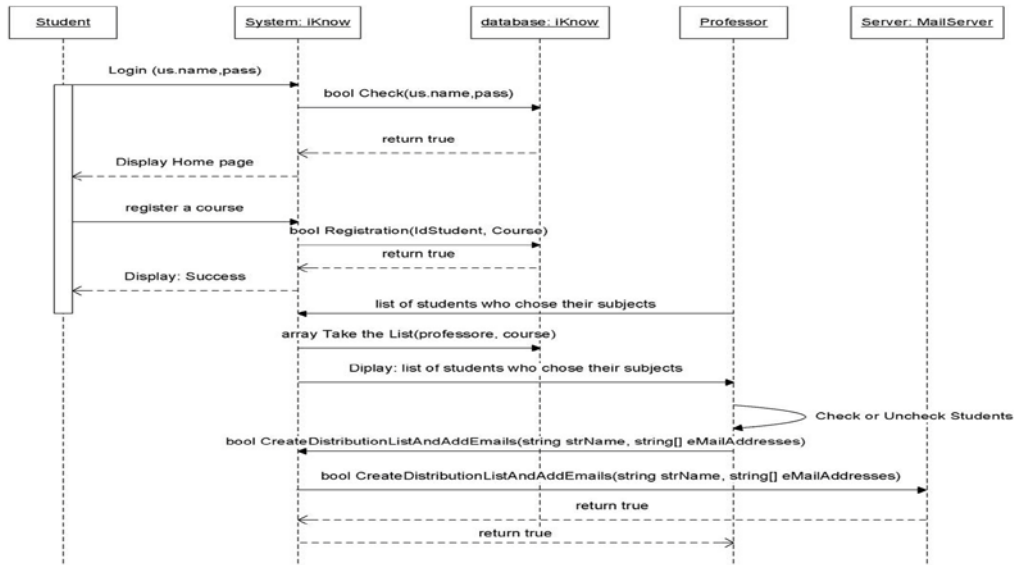


Use Case 4. Course distribution lists

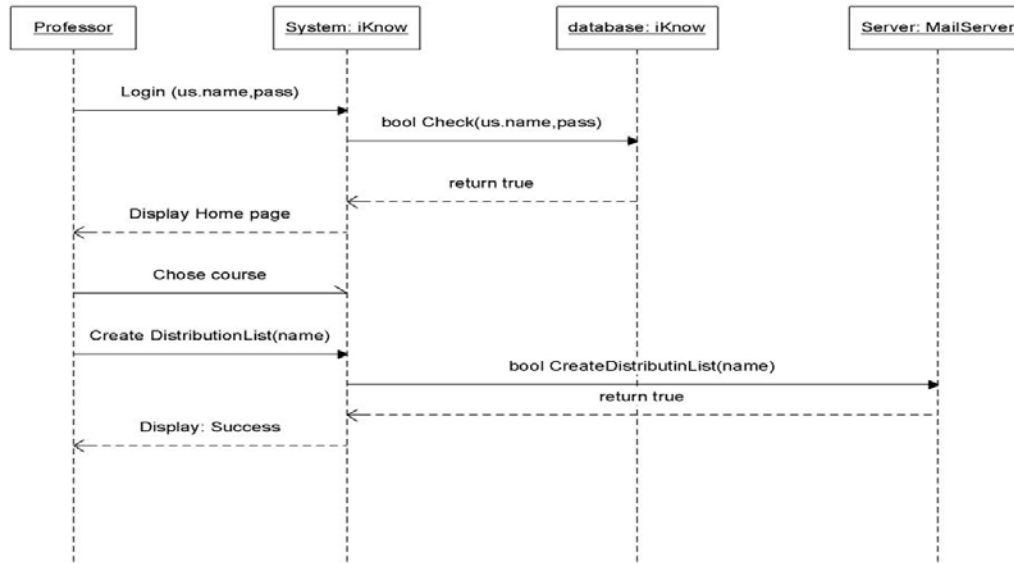
- Faculty staff should be able to create course distribution lists for the courses he/she teaches.
- Faculty staff should be able to create/update/remove individual emails from course list.



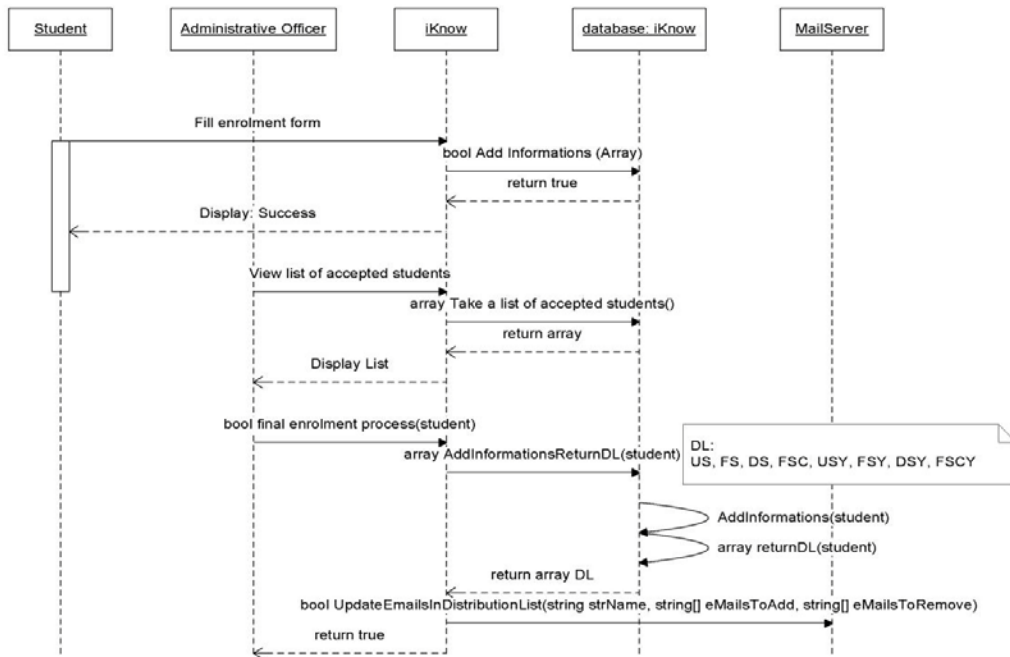
Sequence Diagram. Course enrolment



Sequence Diagram. Creating new Distribution List



Sequence Diagram. When a student is enrolled at University



Library systems

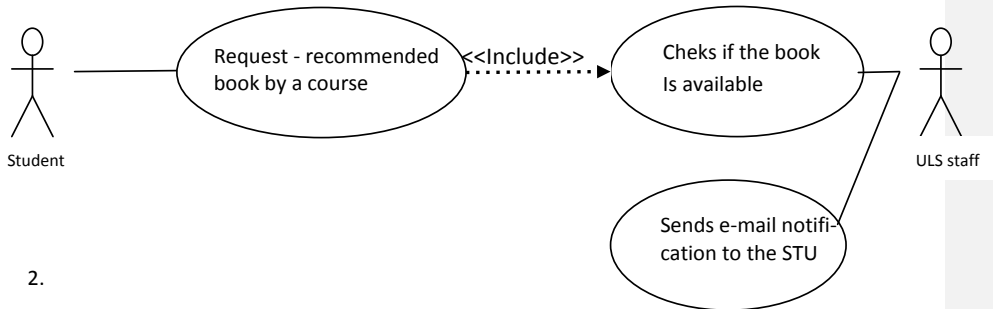
This section describes the technical interface between iKnow and other internal systems run at University.

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

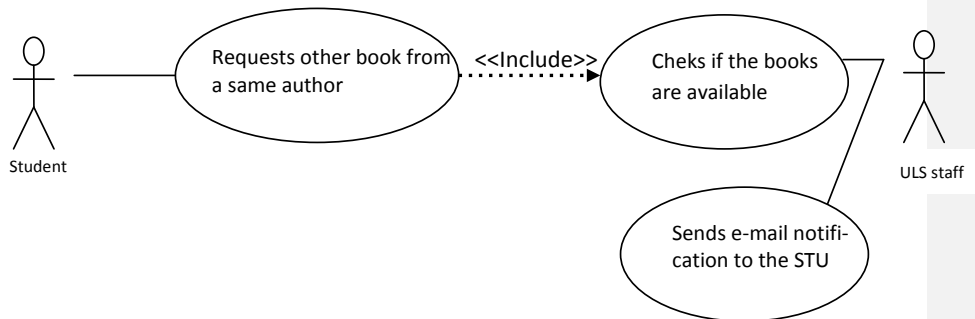
Web Service name	Description	Implemented At
ULS_WS	Enquires information on borrowed books by the students in order to finish the course or study and whether the service should be rejected or accepted depending on the book status within a course	University implements this web service as an addition to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

The LMS → ULS

1.

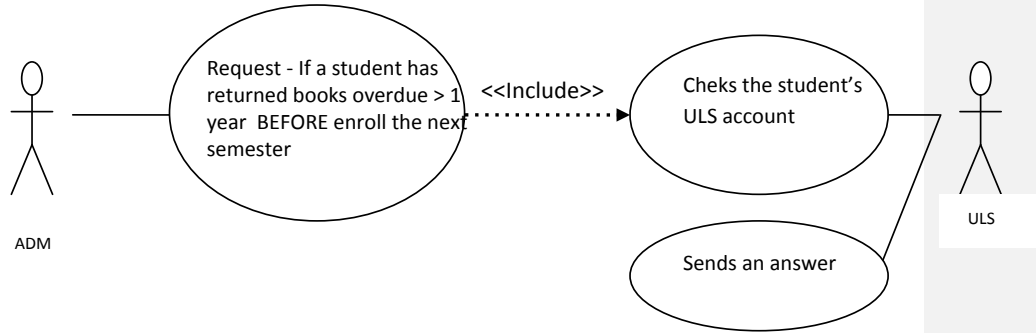


2.

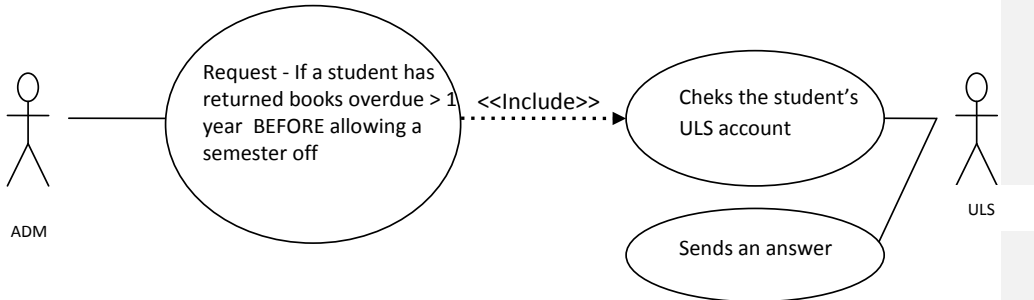


The CORE → ULS

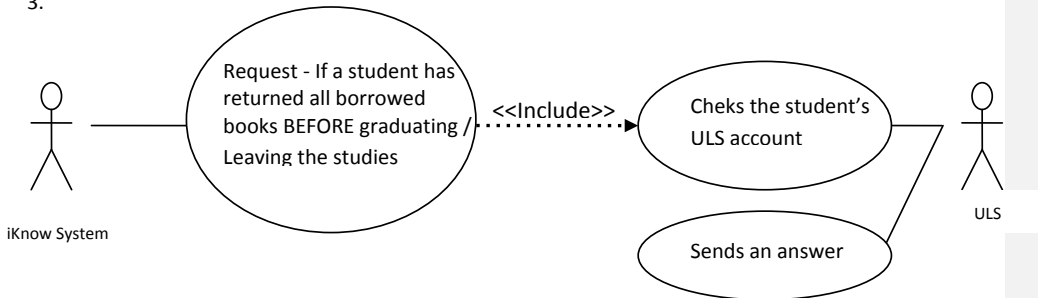
1.



2.



3.



4.

