Education Learning and Management System

Elin Nilsson	-	nileli@chalmers.se
Shaphan Manipaul	-	shaphan@chalmers.se
Tindra Järgenstedt	-	tindra@jargenstedt.se
Chiranjibee Satapathy	-	chiranjibee2012@gmail.com
Eric Erlandsson Hollgren	-	eric.e.hollgren@gmail.com
Achyut Madhavan	-	gusmadac@student.gu.se
Priyanka Pentela	-	priyanka.pentela@gmail.com

22 - 10 - 2023

V3.0

Date	Versions	Comment	Author(s)
2023-09-16	**R 1.0**	● Gave a high level description. ● Defined the goal and scope. ● Listed the stakeholders business goals. ● Goal domain tracing done. ● Derived the core functionalities.	Elin Nilsson, Shaphan Manipaul, Tindra Järgenstedt, Chiranjibee Satapathy, Eric Erlandsson Hollgren, Achyut Madhavan, Priyanka Pentela
2023-09-19		Minor changes to R 1.0	
2023-09-19		Made minor changes to the high level description	Eric Erlandsson Hollgren
2023-09-21		Refined Context diagram and Goal Domain Tracing	Chiranjibee Satapathy
2023-09-21		Stakeholders updated with representatives	Achyut Madhavan Shaphan Manipaul
2023-09-21		Updated Use Case diagram and Core functionalities	Eric Erlandsson Hollgren Chiranjibee Satapathy
2023-09-26		Assumptions were added	Tindra Järgenstedt
2023-09-26		Quality Requirements along with P-Language	Tindra Järgenstedt, Eric Erlandsson Hollgren
2023-09-28		Added Data Requirements ● Class diagram ● Data dictionary	Chiranjibee Satapathy Tindra Järgenstedt, Eric Erlandsson Hollgren, Shaphan Manipaul
2023-09-28		Added Functional Requirements using Task Description Tables	Chiranjibee Satapathy, Shaphan Manipaul, Achyut Madhavan, Eric Erlandsson Hollgren,

			Elin Nilsson, Tindra Järgenstedt
2023-09-29		Added Prioritization	Chiranjibee Satapathy, Elin Nilsson, Tindra Järgenstedt, Achyut Madhavan, Shaphan Manipaul
2023-09-29		Added System Requirements and Software Requirements	Elin Nilsson, Tindra Järgenstedt
2023-09-29	**R 2.0**	Added Traceability matrix	Chiranjibee Satapathy, Elin Nilsson, Tindra Järgenstedt, Achyut Madhavan, Shaphan Manipaul
2023-10-10		Added a new use case and made changes to the use-case diagram	Chiranjibee Satapathy, Shaphan Manipaul, Achyut Madhavan, Eric Erlandsson Hollgren, Elin Nilsson, Tindra Järgenstedt
2023-10-10		Updated the traceability table based on the updated use cases	Chiranjibee Satapathy, Shaphan Manipaul, Achyut Madhavan, Eric Erlandsson Hollgren, Elin Nilsson, Tindra Järgenstedt
2023-10-10		Added the UI Prototypes	Chiranjibee Satapathy, Shaphan Manipaul, Achyut Madhavan, Eric Erlandsson Hollgren, Elin Nilsson, Tindra Järgenstedt
2023-10-10		Added more detail to System requirements	Eric Erlandsson Hollgren
2023-10-17	**R 3.0**	Added more description to the final prioritization.	Elin Nilsson, Tindra Järgenstedt

1. High-level description

This document outlines the requirements for the new Education Learning Management System (ELMS). Today, Canvas is a popular system used by many universities in Sweden. However, the system has many areas of improvement. The universities often rely on external third party systems like Slack or Messenger for communication between students and teachers as the messaging and notification system in canvas is in many ways flawed. Canvas also lacks a way of handling rooms. There is currently no way of finding rooms on the campus and to book rooms using the system. The requirements on the new system strives to fulfill both these flaws and more that currently exist within Canvas.

The new ELMS has the primary goal to establish a platform for both students and teachers, ensuring they have a consistent and uniform access to the resources and facilities that they would need during their time learning and teaching at the university. These resources encompass various elements such as course material like slides and presentations, assignment questions, announcements from teachers, and the literature that is relevant to that course. For teachers, the ELMS will provide new functionalities to administer and document courses, keep track of the student's progress, time management tools, and report scores.

The new system will also include integrations of a map for finding rooms, a way to book group rooms and lecture halls and a messaging system to ease the communication with teachers and fellow peers.

1.1 Goal and scope

The overall goal is to develop a new ELMS for all universities in Sweden. This platform will provide course pages, timetables, a communication system, a room booking system, and map functionality. The functionalities are integrated in such a way that it is uniform across all platforms, enabling the user to access the required information efficiently. The figure below illustrates the scope of the system using a context diagram.

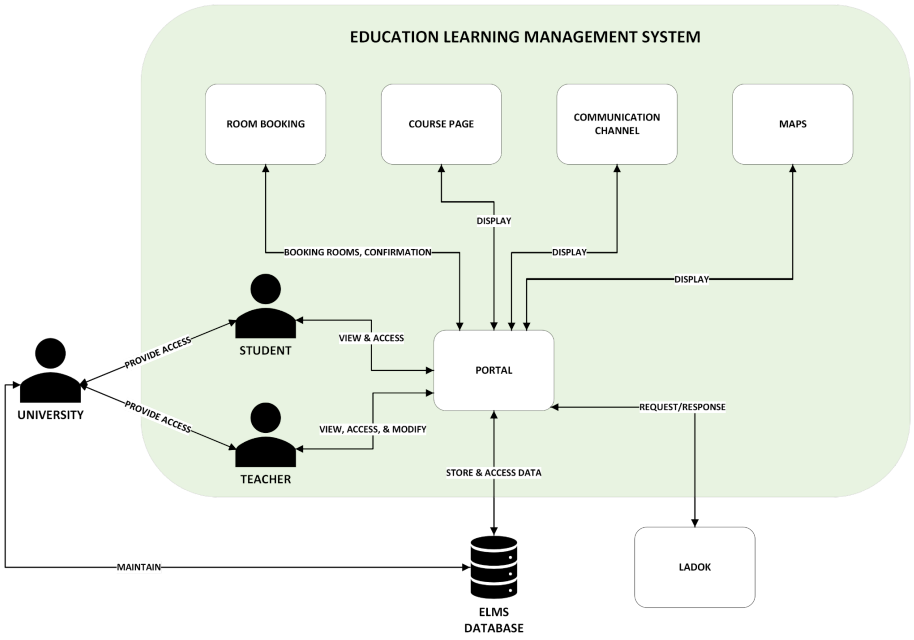

Figure 1: Context diagram for the ELMS

Legend:

- **Green Box**: Shows the product domain.
- **Arrows:** They show the interaction between parts of the system as well as what event the arrow is associated with.
- **View**: Viewing the ELMS application
- **Access**: Accessing the ELMS application
- **Modify:** Modify the content in the ELMS application
- **Display:** The content that the ELMS requests and displays it to the user.

1.2 Business case and stakeholder map

This section describes the stakeholders of the ELMS and how it will generate value for them. This includes potential users as well as funders and why the software is worth developing.

1.2.1 Business Goals

The ELMS will mainly be used by students, teachers, and the IT staff members, as a platform to access material, timetables, communicate, and book rooms. In particular, the ELMS aims to reach the following business goals:

Business Goal 1 (BG1): Provide uniform and consistent functionality for every user across web and mobile platforms. The ELMS aims to provide an integrated Time Management system and a maps system that allows students and teachers access to lectures and room booking services. It will also provide a uniform user interface and access to various third-party services.

Business Goal 2 (BG2): Unify the various third-party dependencies into the ELMS. It will be done by developing the system backend within the organization - eliminating the need for the university to rely on multiple web and app services and reducing the resources spent on maintaining those services.

1.2.2 Goal Domain Tracing

The Goal Domain Tracing allows us to relate our goals to specific events. The functions View, Access, Modify and Display relate to the ones in the context diagram above.

Goal	View	Access	Modify	Display	QR1 (Usability)
Business Goal 1 Uniform and Consistent Functionality	x	x	x	x	x
Business Goal 2 Unify 3rd-Party Dependencies	x	x		x	

Table 1: Goal Domain Tracing

1.2.3 Stakeholders

Name	Relationship	Representative	Power/Sentiment
University	Potential Customer	IT Department	High, may fund system
Teachers	Potential User	Michael Heron, Program Coordinator of the N2GDT Program, heronm@chalmers.se	High, user of the system
Students	Potential User	- Students from the Advanced Requirements Engineering Course. -Students from the Game Design Course. -Other Students.	High, user of the system

Table 2: Stakeholders

Stakeholder 1: University

The university is the stakeholder that may fund the system. The university is the one responsible for providing data to the database that is then accessed by the platform to update and provide information on the platform.

Stakeholder 2: Teachers

The teachers are one of the users of the system. They are the ones creating course pages based on templates that exist in the system. They are also able to create submissions and announcements on the platform. Teachers should be able to both view the platform as well as edit and create information.

Stakeholder 3: Students

The students are one of the users of the system. Students want to access the platform to see their courses, their assignments, their timetables etc. They should be able to view the platform but not edit it.

1.3 Core functionality

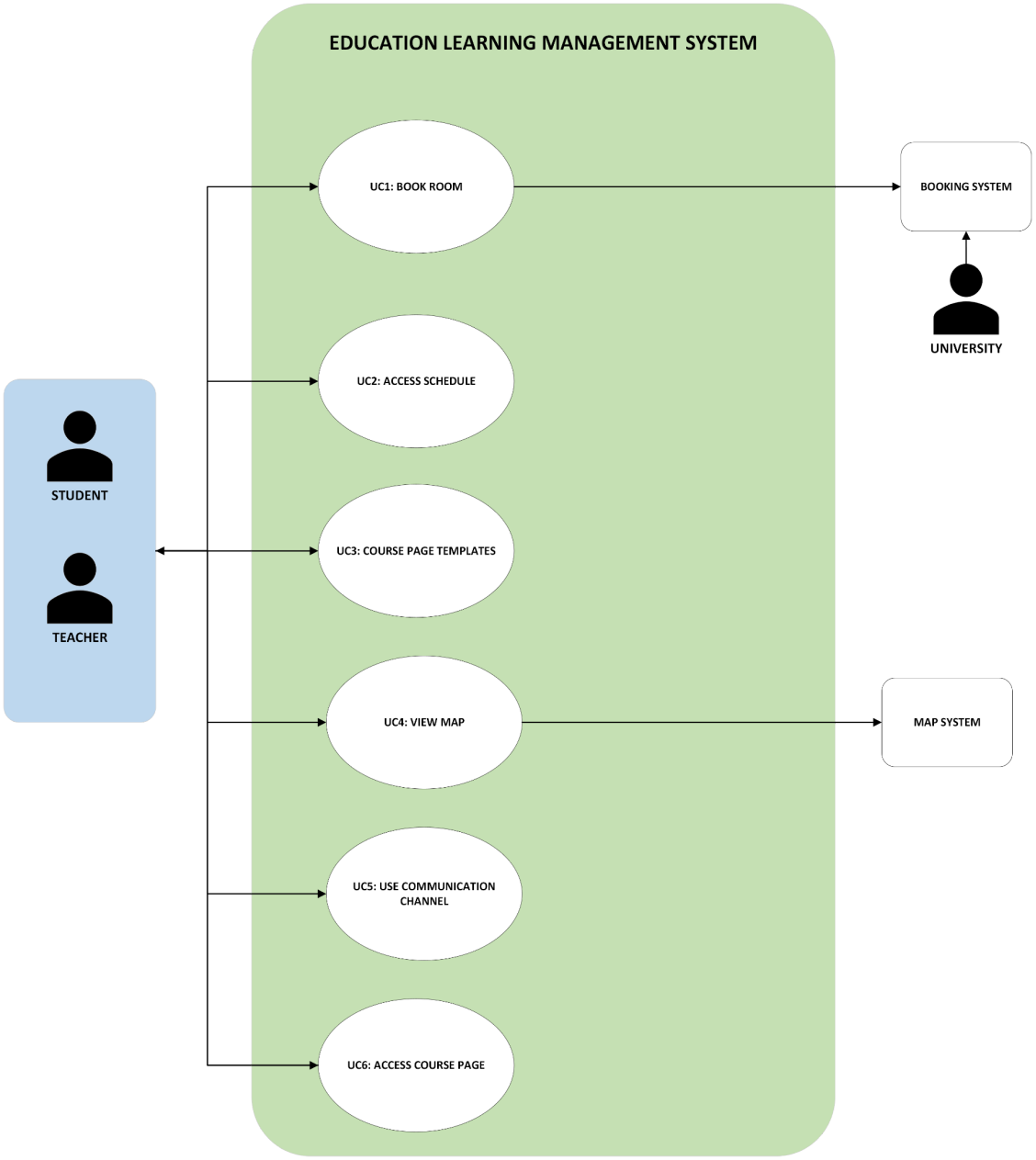

Figure 2: Use case diagram of ELMS

Figure 2 shows the core functionality that our new ELMS system brings to the users besides the assumptions made in section 1.3.1. By providing the new core functionalities we will ease the use of the system for the users.

The use cases shown in **Figure 2** are further explained below in the form of user stories.

UC1 As a user, I want to be able to book rooms so that I can have a place to study, teach, and do group tasks at school. I do this by using the room booking system that is integrated in the ELMS.

UC2 As a user, I want to see a schedule of course events such as lectures, workshops, and assignment deadlines.

UC3 As a teacher, I want to be able to edit the course page so that my students can access the correct information and material. I do this by using the templates provided by the system.

UC4 As a user, I want to access a map within the platform to find rooms and lecture halls on campus so I don't have to use another application for this.

UC5 As a user, I want a proper communication channel to connect with my fellow peers and also receive important information from the university.

UC6 As a user, I want to be able to access course pages in order to view and handle assignments along with other course activities in order to not miss anything.

1.3.1 Assumptions

For the core functionality that we describe in section 1.3 we assume that the new ELMS will include the baseline functionality from the old Canvas system. These functionalities include a way of submitting, creating and grading assignments, creating groups, accessing course pages, receiving and creating announcements as well as a file handling system.

1.4 Performance Requirements, Specific Quality Requirements, Constraints

This section describes the non-functional requirements that are important in order to reach the project-goals. Table 3 estimates the importance of different quality factors.

	Critical	**Important**	**As usual**	**Unimportant**	**Ignore**
Operation					
Integrity/Security			X		
Correctness			X		
Reliability/Availability				X	
Usability		QR1			
Efficiency				QR4	
Revision					
Maintainability			X		
Testability			X		
Flexibility			X		

Transition					
Portability		QR2			
Interoperability		QR3			
Reusability				X	
Installability			X		

Table 3: Quality factors for the ELMS

QR1 - Usability:

The User Interface for the systems needs to be easy to use. There is a lot of functionality and important information available which needs to be easy to find and use. Therefore the user should be able to get to the desired page with no more than three clicks on average from the homepage.

Example: A new user who has never used the system before and wants to quickly access information. If there is a need to click more than three times the user will be frustrated and confused as to where to find the information.

QR2 - Portability:

The User Interface needs to look the same across different platforms and applications. The user should be able to find the same functionality and information in the same place no matter what device or platform the user is using.

Example: A student usually uses the web application to see which groups they belong to. When using the mobile application instead, they can't find their group since that information is not accessed in the same way on the two different platforms.

QR3 - Interoperability:

The ELMS must achieve a minimum of 98% uptime in its connection with the external map- and booking system, as measured over a year. This is to ensure seamless functionality without the necessity of downloading third-party apps.

Example: The connection between the map system and the EMLS needs to work correctly in order to prevent the need for downloading another app provided by a third party.

QR4 - Efficiency:

The User Interface does not need to load as fast as other websites (no more than 2 seconds) since the users of the system can not access the same information elsewhere. This makes the quality requirement less important but we still should aim for a recommended load time of no more than 2 seconds.

Example: The new ELMS is not a time-critical system, and the user can perform operations with a

comfortable margin for response time without being frustrated.

2. User Requirements Specification

2.1 Data requirements

In this section, we describe the data requirements for our proposed system.

2.1.2 Class Diagram

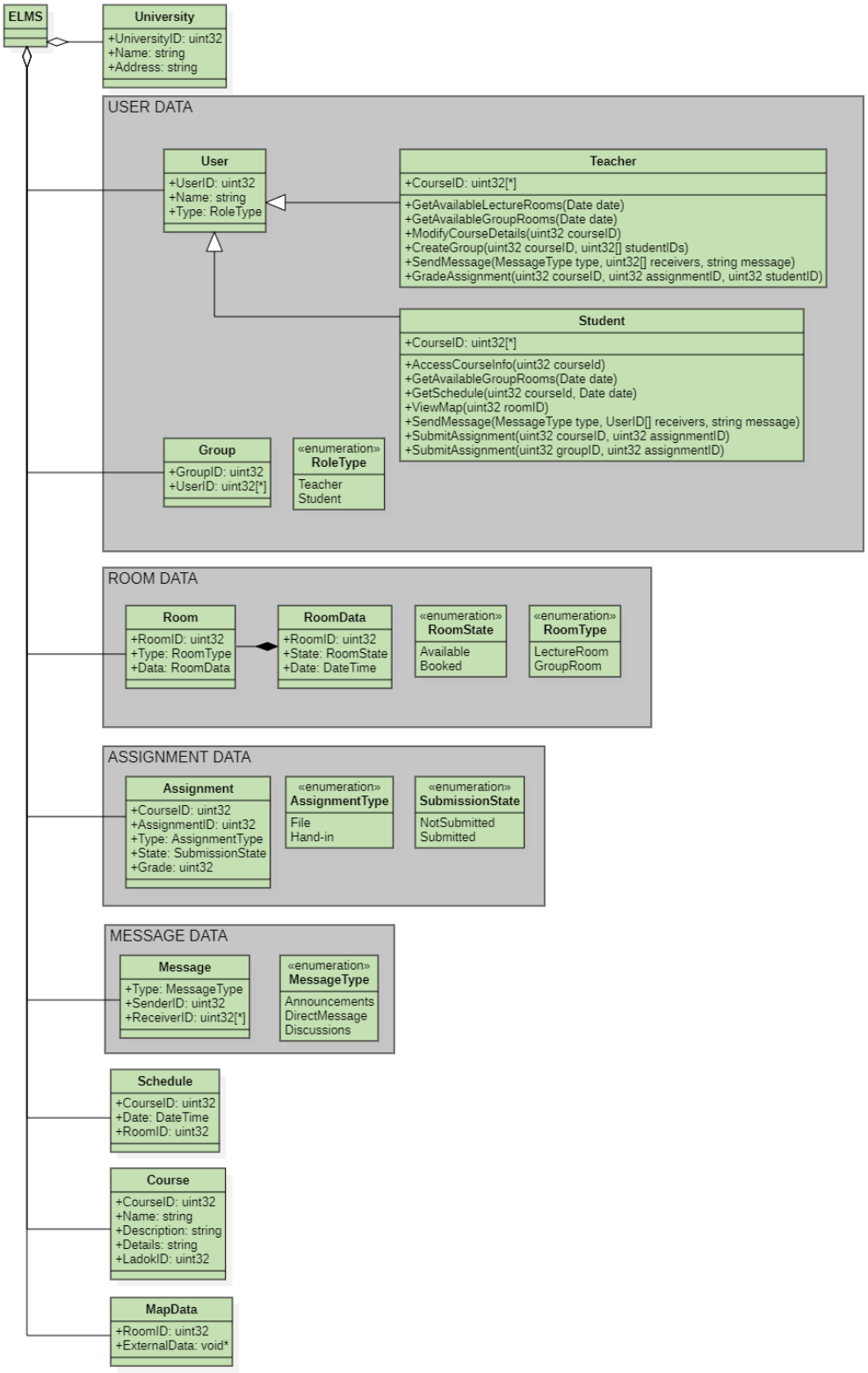

Figure 3: Class diagram of ELMS

The Figure 3 above represents the data artifacts of our proposed ELMS using a class diagram.

All the data descriptions and data artifacts are briefly explained using data dictionaries in the following section.

2.1.3 Data Dictionary

DR1	**Class: University** The universities as the customers provide the ELMS as a tool to the teachers and students.
	Examples: 1. University of Gothenburg 2. Chalmers University of Technology
	Attributes: 1. UniversityID - The unique ID for the university. 2. Name - Name of the university. 3. Address - Address of the university.

Table 4: Data dictionary for DR1

DR2	**Class: User** The users of the ELMS system who will generally access it on a daily basis.
	Attributes: 1. UserID - The unique ID for the user. 2. Name - The name of the user. 3. RoleType - The user role in using the ELMS system.

Table 5: Data dictionary for DR2

DR2.1	**Class: Teacher** The set of users of the ELMS system who will access the system
	Methods: 1. GetAvailableLectureRooms(Date date) : Room[] - Returns an array of lecture rooms available on the provided date. 2. GetAvailableGroupRooms(Date date) : Room[] - Returns an array of group rooms available on the provided date. 3. ModifyCourseDetails(uint32 courseID) : void - Allows for

	modifying the course details page or editing the course template for the given Course ID. 4. CreateGroup(uint32 courseID, uint32[] studentIDs) : GroupID - Returns the GroupID after creating a group for the specified CourseID and the given students. 5. SendMessage(MessageType type, uint32[] receiverIDs, string message) : void - Sends a message of the specified MessageType to the users in the receiverIDs array. 6. GradeAssignment(uint32 courseID, uint32 assignmentID, uint32 studentID) : void - The teacher grades the student on the particular assignment for the course specified by their IDs.
	Attributes: 1. CourseID [] - List of courses the teacher teaches or supervises.

Table 6: Data dictionary for DR2.1

DR2.2	**Class: Student** The other set of users of the ELMS system who will access the system
	Methods: 1. AccessCourseInfo(uint32 courseID) : void - Provides the student access to the particular course as specified in the ID. 2. GetAvailableGroupRooms(Date date) : Room[] - Returns an array of group rooms available on the provided date. 3. GetSchedule(uint32 courseID, Date date) : Schedule - Returns the schedule of the particular course on the provided date. 4. ViewMap(uint32 roomID) : MapDate - Displays the map from the Campus Map API for the specified room. 5. SendMessage(MessageType type, uint32[] receiverIDs, string message) : void - Sends a message of the specified MessageType to the users in the receiverIDs array. 6. SubmitAssignment(uint32 courseID, uint32 assignmentID) : SubmissionState - Returns the updated SubmissionState for the assignment submitted by the student. 7. SubmitAssignment(uint32 groupID, uint32 assignmentID) : SubmissionState - Returns the updated SubmissionState for the assignment submitted by the student group.
	Attributes: 1. CourseID [] - List of courses the student is registered.

Table 7: Data dictionary for DR2.2

DR3	**Class: Group** A set of Users
	Examples: 1. A subset of users forming a group with a name, i.e. Group 10
	Attributes: 1. GroupID - uint32 providing a identifier id for a group 2. UserID - uint32[*] providing a list of references to all the ids of students within that group

Table 8: Data dictionary for DR3

DR4	**Class: Room** A room with its room types and a timestamp
	Examples: 1. Lecture Rooms in Jupiter building. 2. Group Study Rooms in Jupiter building.
	Attributes 1. RoomID - The unique identifier of the list of rooms 2. Type - Provide information whether its a lecture room or a group room 3. Data - Provide information whether the rooms are currently booked or available for booking.

Table 9: Data dictionary for DR4

DR4.1	**Class: RoomData** The room specific data that contains the availability status of the room with a timestamp
	Attributes: 1. RoomID - Unique room identifier 2. State - enum providing the current state of the room (booked, available) 3. Date - DateTime providing timestamp for the booking

Table 10: Data dictionary for DR4.1

DR5	**Class: Assignment** Assignments and its submission types, of a course along with its grade
	Examples: 1. R2 Assignment in Requirements Engineering course.
	Attributes 1. CourseID - The unique ID for the course related to the assignment 2. AssignmentID - The unique ID for the assignment 3. Type - The AssignmentType that states whether it's a file or a hand-in. 4. State - The SubmissionState that states whether it's submitted or not submitted. 5. Grade - The grade for the assignment

Table 11: Data dictionary for DR5

DR6	**Class: Message** A messaging system consisting of data about the sender, the receiver and the channel through which the message goes through
	Examples: 1. Announcements regarding a particular course. 2. Direct Messages from teachers regarding assignment feedback.
	Attributes: 1. Type - The type of the message that is sent. Could be announcements, discussions or direct messages 2. Sender ID - The unique identifier of the user that sends the message. 3. Receiver ID - The unique identifier of the user/users that receive the message.

Table 12: Data dictionary for DR6

DR7	**Class: Schedule** The time and place of the course taking place
	Attributes: 1. CourseID - The unique ID for the course 2. Date - The date connected to the scheduled event 3. RoomID - The unique ID for the room that the course takes place in

Table 13: Data dictionary for DR7

DR8	**Class: Course** A course along with its name, course description, details and how it is associated with Ladok.
	Attributes: 1. CourseID - Unique identifier for the course 2. Name - Name of the course 3. Description - Description of the course 4. Details - Details about the course 5. LadokID - Identifier for the course in Ladok

Table 14: Data dictionary for DR8

DR9	**Class: Map Data** Map display data we get from CampusMap API.
	Attributes: 1. RoomID - The room we are looking for in the ELMS system. 2. ExternalData - The data received from the CampusMap API.

Table 15: Data dictionary for DR9

2.2 Functional requirements

This section describes the functional requirements for the new ELMS system.

<table>
<tr><td>Task 1 - FR1</td><td colspan="2">Access Course Information - UC6</td></tr>
<tr><td>Purpose</td><td colspan="2">Support students to access the course information on the ELMS from a mobile application or web browser.</td></tr>
<tr><td>Trigger</td><td colspan="2">The student has selected the Course Information Page.</td></tr>
<tr><td>Pre-condition</td><td colspan="2">The student is registered in their university and has used their university email address to access ELMS.</td></tr>
<tr><td rowspan="5">Frequency</td><td colspan="2">The student might access the Course Page daily.</td></tr>
<tr><td>Sub-Tasks:</td><td>Example Solution:</td></tr>
<tr><td>1. Show Registered Courses

2. The student selects one of the Registered Courses.</td><td>Show on display the list of registered courses the student is registered for.

Show on display the detailed view of the course selected by the student.</td></tr>
<tr><td colspan="2">Variants:</td></tr>
<tr><td colspan="2">1. If the student has not registered for any course, allow the student to request registration into available courses for the semester.
2. The student might access the Schedule Viewer for the selected course.</td></tr>
</table>

Table 16: Task description for FR1

Task 2 - FR2	**Room Booking - UC2**	
Purpose	Support students/teachers to book group study rooms/lecture rooms on the ELMS from a mobile application or web browser.	
Trigger	The student/teacher has selected the Room Booking Page.	
Pre-condition	The student/teacher is registered in their university and has used their university email address to access ELMS.	
Frequency	The student/teacher might access the Room Booking 1-2 times a week.	
	Sub-Tasks:	Example Solution:
	1. Show available rooms for the current week(s).	Show on display the list of available rooms along with time slots.
	2. The student/teacher selects the available room for a particular time slot.	Show on display a confirmation message for the booked room.
	3. Show the booked lecture/group study rooms.	Show on display the list of booked rooms along with time slots.
	Variants:	
	1. The student/teacher can view the available slots for a particular time slot. 2. The student/teacher can cancel/edit the booked room.	

Table 17: Task description for FR2

Task 3 - FR3	**Schedule Viewing - UC2**	
Purpose	Support the student to view the schedule of their courses.	
Trigger	The student has selected the schedule page.	
Pre-condition	The student is using the university email address to access ELMS.	
Frequency	The student might access the Schedule Page daily.	
	Sub-Tasks:	Example Solution:
	1. The student selects the course or courses they want to view in their schedule. 2. The student filters on what they want to see, e.g., Lectures, deadlines or exams.	Shows the schedule for the Lectures along with the Lecture room, assignment deadlines and exams.

Table 18: Task description for FR3

Task 4 - FR4	**Course Page Editing - UC3**
Purpose	Supports the teachers to edit/update the course details, materials, etc. regarding a particular course.
Trigger	The teacher has selected an existing course page or selects from a course template page.
Pre-condition	The teacher is using the university email address to access ELMS.
Frequency	The teacher may access the Course Page 2-3 times a month.
	Sub-Tasks:
	1. The teacher gives the course details. 2. The teacher edits the course pages at any given time. 3. The teacher creates student groups. 4. The teacher adds students/teaching assistants/ manually.
	Variants:
	1. The teacher allows students to create groups.

Table 19: Task description for FR4

Task 5 - FR5	**View Maps - UC4**	
Purpose	Support the users to navigate around the campus and to locate group rooms and lecture rooms.	
Trigger	The user is using the maps system.	
Pre-condition	The student has used their university email address to access ELMS.	
Frequency	The student might access the maps system daily.	
	Sub-Tasks:	Example Solution:
	1. The student selects to view the location of a campus building.	The system shall display the map view through the Campus Map integration.
	Variant	
	1. The student selects to view the booked room location.	

Table 20: Task description for FR5

Task 6 - FR6	**Communication Channel - UC5**	
Purpose	Support the student/teacher to communicate important information amongst each other.	
Trigger	The student/teacher is using the messaging system.	
Pre-condition	The student/teacher is using the university email address to access the ELMS.	
Frequency	The student/teacher might access the Communication Channel daily.	
	Sub-Tasks:	Example Solution:
	1. The student/teacher chooses who they want to send the message to. 2. The student/teacher can view their inbox.	The system displays the text box to select the message receivers. The system displays the previous messages received by the user.
	Variants:	
	1. The student/teacher can view their messages.	

Table 21: Task description for FR6

Task 7 - FR7	**Assignment Submission - UC6**	
Purpose	Support submitting assignments for students.	
Trigger	The student has selected the Course page	
Pre-condition	The students have used their university email address to access ELMS and are accessing the Assignments Page.	
Frequency	The student might access the Assignments Page 1-2 times a month.	
	Sub-Tasks:	Example Solution:
	1. The student accesses the Assignment Page for submission. 2. The student selects a particular assignment to submit.	Display the list of assignments that the student can submit. Show details for the assignment submission. Show the button to upload/re-upload the assignment file.
	Variants:	
	1. The student accesses the Assignment Page to view the grades.	

Table 22: Task description for FR7

Task 8 - FR8	**Assignment Handling - UC6**	
Purpose	Support viewing and grading assignments for teachers.	
Trigger	The teacher has selected the Course page	
Pre-condition	The teacher is using the university email address to access ELMS.	
Frequency	The teacher might access the Assignments Page 1-2 times a month.	
	Sub-Tasks:	Example Solution:
	1. The teacher accesses the Assignment Page for viewing submissions.	Display the list of assignments that the students/student groups have submitted.
	2. The teacher selects a particular assignment to review.	Display the selected assignment for review.
	3. The teacher assigns a grade and optional feedback for the assignment.	Provide an input box for grading the assignment and the text box to add feedback.

Table 23: Task description for FR8

2.3 Detailed Performance Requirements, Specific Quality Requirements, Constraints

In this section we will explain how we plan to measure our quality requirements using P-Language.

QR1: Usability	**The user should be able to get to the desired page with no more than three clicks.**
SCALE	The average number of clicks the user performs
METER	Measure the number of clicks in-house by accessing a new piece of information
MUST	Must be less than 4 clicks
WISH	Three clicks

PLAN	Three clicks
PAST	On average much more than three clicks

Table 24: P-Language for QR1

QR2: Portability	**The user should be able to find the same functionality and information in the same place no matter what device or platform the user is using.**
SCALE	The average time to find a functionality on another device
METER	We measure how long it takes to find information or functionality on another device, in seconds
MUST	Must be the same across all platforms
WISH	Equally quick on all platforms
PLAN	Difference in Two seconds across platforms
PAST	Some functionality can not be found across all platforms

Table 25: P-Language for QR2

QR3: Interoperability	**The ELMS must achieve a minimum of 98% uptime in its connection with the external map- and booking system, as measured over a year.**
SCALE	The average percentage of uptime in a year
METER	We measure the time our service is up and running, and divide it by total time
MUST	Be running 98% of the time
WISH	99.9% uptime
PLAN	98% of uptime
PAST	Down due to maintenance during the night

Table 26: P-Language for QR3

2.4 Proposed prioritization

In this section the functional requirements are prioritized by performing the 100$ test and Ranking method. In the subsections below the result from each technique is shown. The final prioritization was derived by looking at both of the outcomes from the two techniques. The two different outcomes were similar but had some differences. We discussed the differences in the group from the stakeholders perspective and came up with the final prioritization based on what seemed most important. The final prioritization from the two methods combined is the following:

1. Access Course Information
2. Schedule Viewing
3. Communication Channels
4. Assignment Submission
5. Assignment handling
6. View Maps
7. Course Page Editing
8. Room Booking

2.4.1 100$ test

The test was conducted by using personas. One person acted as the University, two persons acted as a group of teachers and two persons acted as a group of students. The stakeholder groups were given 100$ each and were free to divide the money as they liked, based on how important they found each requirement. Table 27 illustrates how each stakeholder group spent their money.

Stakeholder	FR	Amount	Stakeholder	FR	Amount	Stakeholder	FR	Amount
University	**FR3**	30$	**Teachers**	**FR4**	20$	**Students**	**FR1**	28$
	FR1	21$		**FR8**	20$		**FR6**	24$
	FR5	15$		**FR6**	15$		**FR3**	22$
	FR7	11$		**FR7**	15$		**FR2**	12$
	FR6	10$		**FR1**	15$		**FR5**	10$
	FR2	6$		**FR3**	8$		**FR7**	4$
	FR8	4$		**FR5**	5$		**FR8**	0$
	FR4	3$		**FR2**	2$		**FR4**	0$

Table 27: 100$ test - Individual Stakeholders

Stakeholder	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8
University	21$	6$	30$	3$	15$	10$	11$	4$
Teachers	15$	2$	8$	20$	5$	15$	15$	20$
Students	28$	12$	22$	0$	10$	24$	4$	0$
Total amount:	**64$**	**20$**	**60$**	**23$**	**30$**	**49$**	**30$**	**24$**

Table 28: 100$ test - Cumulative results

When combining the funding done by all the stakeholders by simply adding up the money spent on each requirement the final prioritization looked as follows: (Table 8 shows the total $ for each requirement).

1. Access Course Information
2. Schedule Viewing
3. Communication Channels
4. View Maps & Assignments submission
5. Assignment handling
6. Course Page Editing
7. Room Booking

2.4.2 Ranking

Ranking was performed the same way as the 100-dollar test but instead of giving each requirement money, the stakeholder personas groups ranked the requirement from 1-10. After that, the average rank of the requirement decides how high it is prioritized. For requirements receiving the same score, the one given a lower number by any stakeholder group is prioritized higher. The ranking by each stakeholder and the total ranking is shown in table 29.

Stakeholder	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8
University	2	4	1	8	3	6	5	7
Teachers	4	8	6	1	7	3	5	2
Students	1	4	3	8	6	2	5	7
Total ranking:	**7**	**16**	**10**	**17**	**16**	**11**	**15**	**16**

Table 29: Ranking method for prioritization

Final prioritization for ranking:

1. Access Course Information
2. Schedule Viewing
3. Communication Channel
4. Assignment Submission
5. Assignment Handling
6. View Maps
7. Room Booking
8. Course Page Editing

3. System Requirements

In section 3.1, we provide detailed system and software requirements for the Room Booking, the Schedule Viewing as well as the Communication Channels.

3.1 System requirements

SR-ID	System req.	SWR-ID	Software req.	FR-ID
SR2.1	The system shall display the rooms with their availability status for the chosen day.	**SWR2.1**	The software shall know the status of the rooms availability.	**FR2**
SR2.2	The system shall send a confirmation email to the users email address when a room is successfully booked.	**SWR2.2**	The software shall book the chosen room and update the database with the new availability status.	**FR2**
SR2.3	The system shall show the users booked rooms and an option to cancel bookings.	**SWR2.3**	The software shall allow cancellation of rooms and update the database accordingly.	**FR2**

Table 30: System and Software requirements respective to FR2

SR-ID	System req.	SWR-ID	Software req.	FR-ID
SR3.1	The system shall display the events as a schedule for all the courses of the user.	**SWR3.1**	The software shall handle the events and their times of each course.	**FR3**
SR3.2	The system shall receive input from the user to filter the schedule and display an updated version.	**SWR3.2**	The software shall update the schedule based on the input from the system.	**FR3**

Table 31: System and Software requirements respective to FR3

SR-ID	System req.	SWR-ID	Software req.	FR-ID
SR6.1	The system shall allow the user to choose which users to send a message to.	**SWR6.1**	The software shall provide a list of users that are able to receive a message.	**FR6**
SR6.2	The system shall display received messages and if they are read or not.	**SWR6.2**	The software shall handle messages and their status.	**FR6**
SR6.3	The system shall categorize the messages based on their type. Eg. announcement or direct message.	**SWR6.3**	The software shall provide the system with information about the type of each message.	**FR6**
SR6.4	The system shall send a notification to the user when a new message is sent to them.	**SWR6.4**	The software shall send a signal to the messaging system when a message has been sent.	**FR6**

Table 32: System and Software requirements respective to FR6

3.2 Traceability

In this section, we do traceability wherein we link each artifact to each of its respective artifacts.

Business Goal	Use Case	Functional Req.	Data Req.	Quality Req.	System Req.	Software Req.
BG1	UC1	FR2	DR2 DR4	QR1 QR3	SR2.1-3	SW2.1-3
	UC2	FR3	DR2 DR4 DR7	QR1 QR2	SR3.1-2	SWR3.1-2
	UC3	FR4	DR2.1 DR4 DR7	QR2		
	UC4	FR5	DR9	QR3		
	UC5	FR6	DR2 DR6	QR1 QR2	SR6.1-4	SWR6.1-4
	UC6	FR7	DR2.2 DR5 DR8			
		FR8	DR2.1 DR5 DR8			
		FR1	DR2.2 DR5 DR8	QR1 QR2		
BG2	UC1	FR2	DR4	QR1 QR3	SR2.1-3	SWR2.1-3
	UC4	FR5	DR9	QR3		

Table 33: A Traceability table showing the relationships between various artifacts of the proposed system

3.3 UI Prototype

This section shows the UI prototypes for the different core functionalities described in section 1.3.

Figure 4 The Home Page

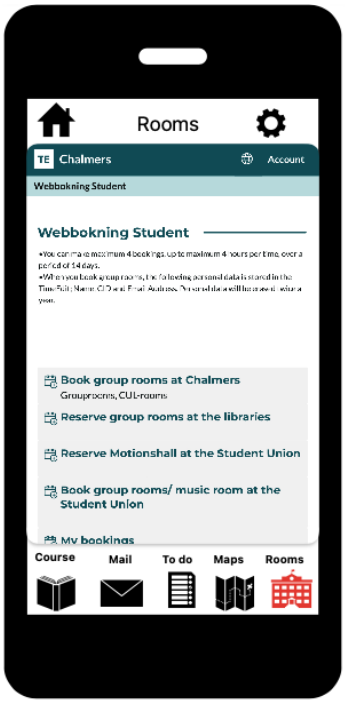

Figure 5 Room Booking

Figure 6 Schedule Viewing

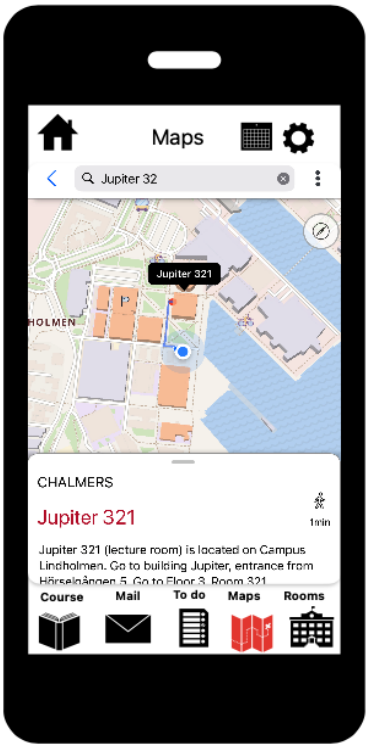

Figure 7 View Map

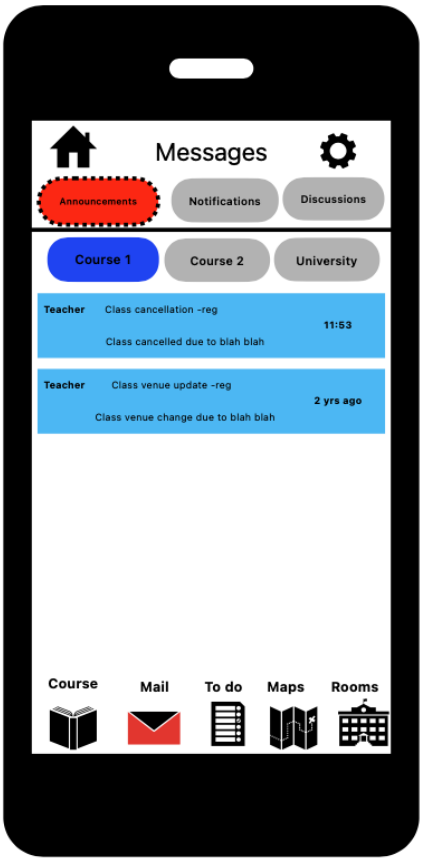

Figure 8 Notification channel

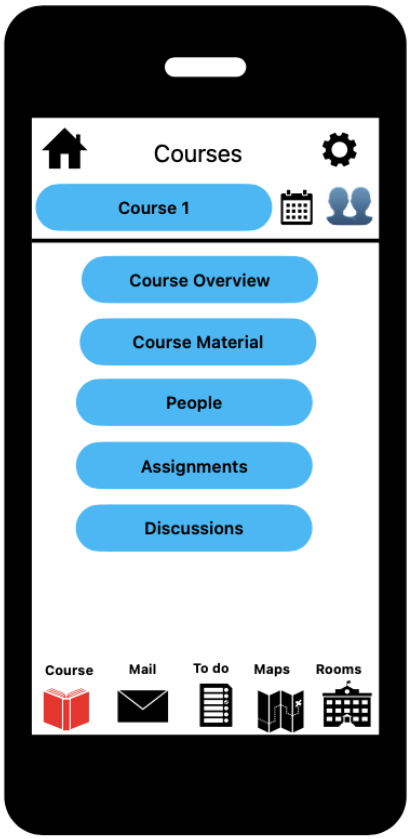

Figure 9 Course Page