Software Requirements Specification

for

CENG Website Management Desktop App

v0.4 not approved

Özgesu Özen
Busenur Aktılav
Ayşenur Erdem
Berk Bildirici
Furkan Emre Yılmaz

Izmir Institute of Technology
12.03.2019

Table of Contents

List of Figures

Revision History

Name	Date	Reason for Changes	Version
Group 9	13.03.2019	Table of Contents added	v0.1.1
Group 9	14.03.2019	Renew Password GUI added	v0.1.2
Group 9	20.03.2019	Detailed Use Case Description Added	v0.2.0
Group 9	21.03.2019	ER Diagram Added	v0.3.0
Group 9	22.03.2019	UIs Updated	v0.3.1
Group 9	24.03.2019	Review Revision	v0.4.0

1. Introduction

1.1. Definitions, acronyms, and abbreviations

TERM	**DEFINITION**
SRS	Software Requirements Specification
CENG	Computer Engineering Department
CENG WM	Computer Engineering Department Website Management
Users	Admin/Administrator and Content Manager
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
Content Manager	Department Instructor(s), Assistant(s) or Secretary who is added by Admin
CM	Content Manager
Login	To connect the system interface
Logout	To end the connection to the system interface.
CRUD	Create, Read, Update, Delete
WM	Website Management
API	Application Programming Interface
GUI / UI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
Database	Collection of all the information monitored by this system.
ER Diagram	Entity relationship diagram that describes relations between specific domain knowledge.
Use Case	It is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal.
Context Diagram	It defines the boundary between the system and its environment

Java	It is a set of computer system and specifications
Extends	It is used when a use case adds steps to another first-class use case
Includes	It is used to extract use case fragments that are *duplicated* in multiple use cases.
Subsystem	Group of interconnected and interactive parts that performs an important job or task as a component of a larger system.
App	CENG Website Management App
Credential	Evidence of authority
Main Menu	The opening or main page of a website
Scrollpane	Scrollable field in app window
Priority	Order of priority for use cases in coding Low Medium High

1.2. Purpose

The purpose of this SRS document is to provide a detailed description of the requirements for the “CENG Website Management Desktop App” software. It will illustrate the purpose and complete description of the development system. It will also explain system constraints, interfaces, and interactions with other external applications.

The intended reader groups for this SRS document are secretaries, instructors or assistants of Computer Engineering Department who will use the system as Content Manager and chosen members of Computer Engineering Department who will use the system as Admin.

1.3. Scope

Computer Engineering Website Management App is a desktop application which helps users of CENG department to get control of the content of the CENG website. This system will be designed to maximize the users’ productivity by providing a tool to assist in automating the CRUD processes of the website content, which would otherwise have to be performed manually. The system will meet the users’ needs while remaining easy to understand and use.

More specifically this system is designed to allow users to manage all the courses in the department website. Additionally, users can make CRUD processes on the courses. The system allows admin to manage email lists and send events notification to the selected email lists. The system provides communication between users and CENG website. The system also contains a relational database containing the list of courses, staffs of CENG department, and users’ information.

1.4. References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

1.5. Overview

The remainder of this document includes four chapters which are overall description, specific requirements, appendices, and index.

The Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements. It is used to establish a context for the technical requirements specification.

The third chapter, Specific Requirements section, of this document, is written primarily for the developers and describes in technical terms the details of the functionality of the product.

The Appendices include all results of the requirement prioritization and a release plan based on them.

2. Overall Description

2.1. Product perspective

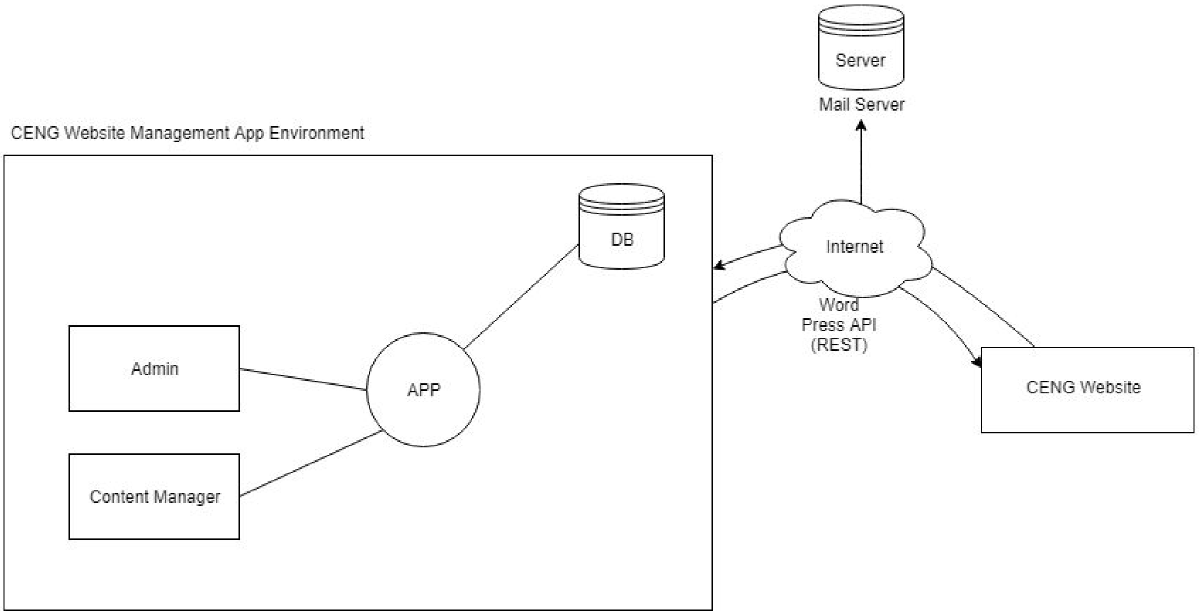


Figure 1: Context Diagram

The system has two main external capability. The first one is connecting CENG Website via WordPress API for updating the contents of the website and the second one is sending email to any email server on the internet to notify its internal email list about CENG Department events.

2.2. Product functions

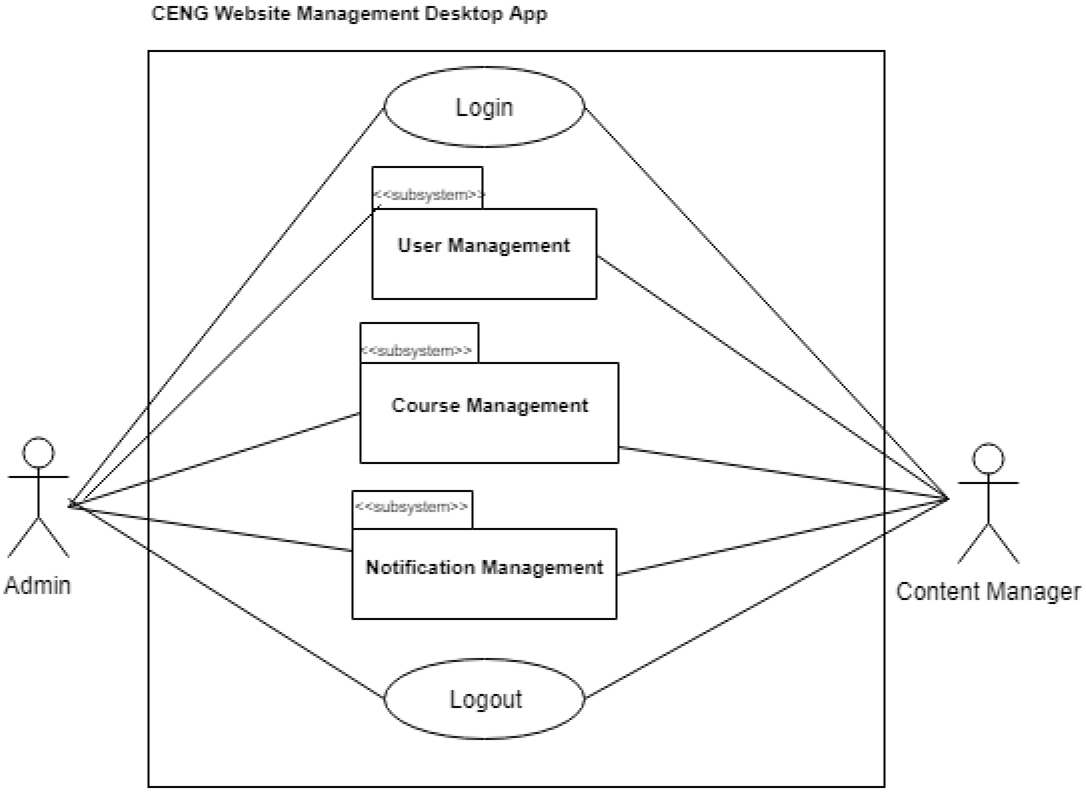


Figure 2: CENG Website Management Desktop App

2.2.1. Use Case: **Login**

Description: At the start of the app, all users must login to the system in order to make further actions.

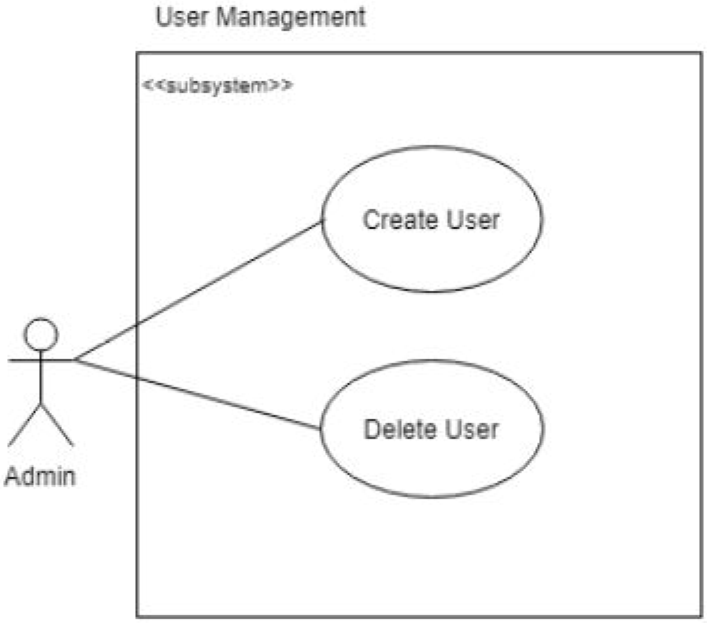


Figure 3: User Management

2.2.2. User Management Use Case Diagram

2.2.2.1. Use Case: **Create User**

Description: Admin creates users.

2.2.2.2. Use Case: **Delete User**

Description: Admin can delete users with their information.

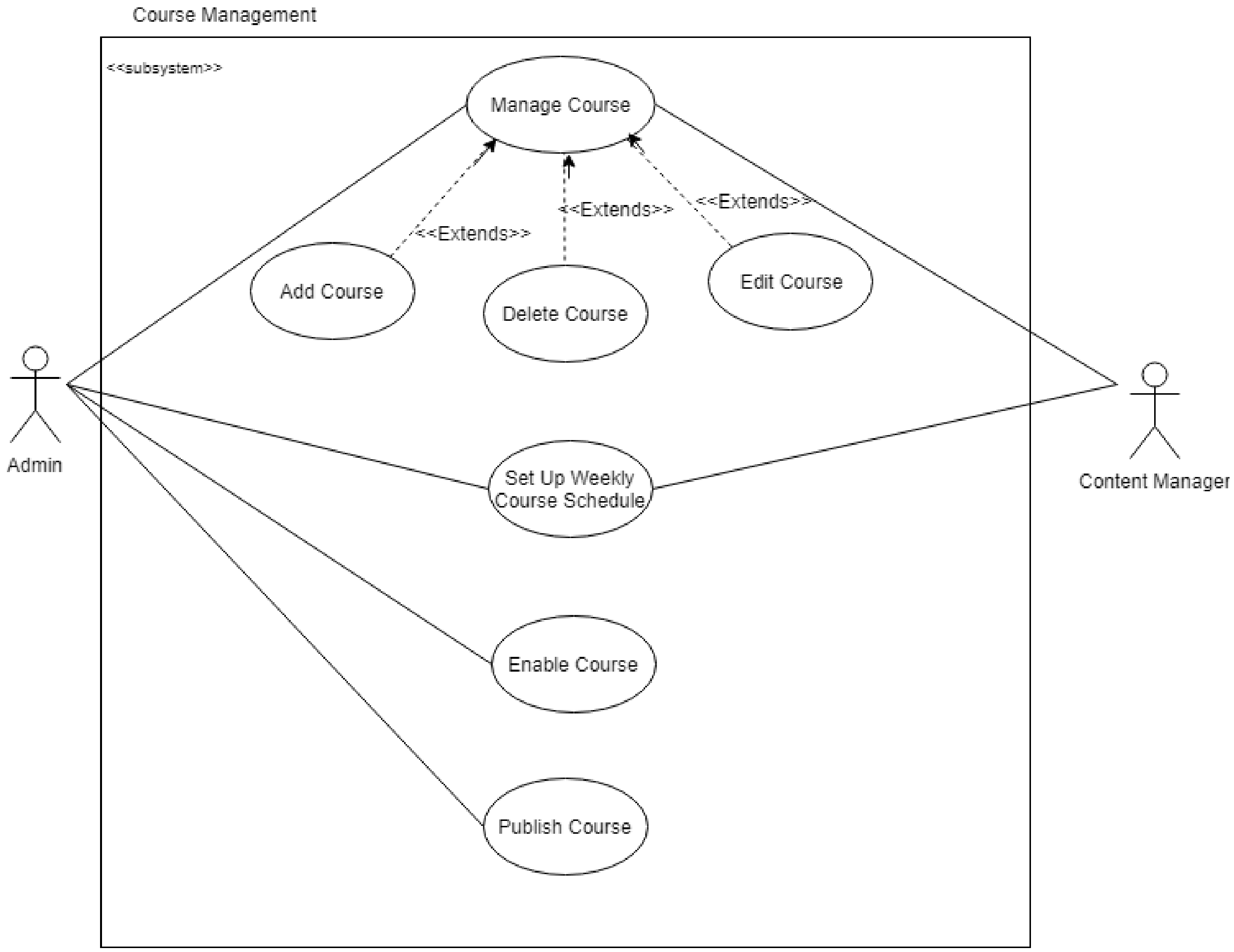


Figure 4: Course Management

2.2.3. Course Management Use Case Diagram

2.2.3.1. Use Case: **Manage Course**

Description: CM and Admin manage courses by adding, deleting and updating.

2.2.3.2. Use Case: **Enable Course**

Description: Admin enables courses for that semester.

2.2.3.3. Use Case: **Set Up Weekly Course Schedule**

Description: CM and Admin schedule the weekly course table of that semester.

2.2.3.4. Use Case: **Publish Course**

Description: Admin publish all courses to the website.

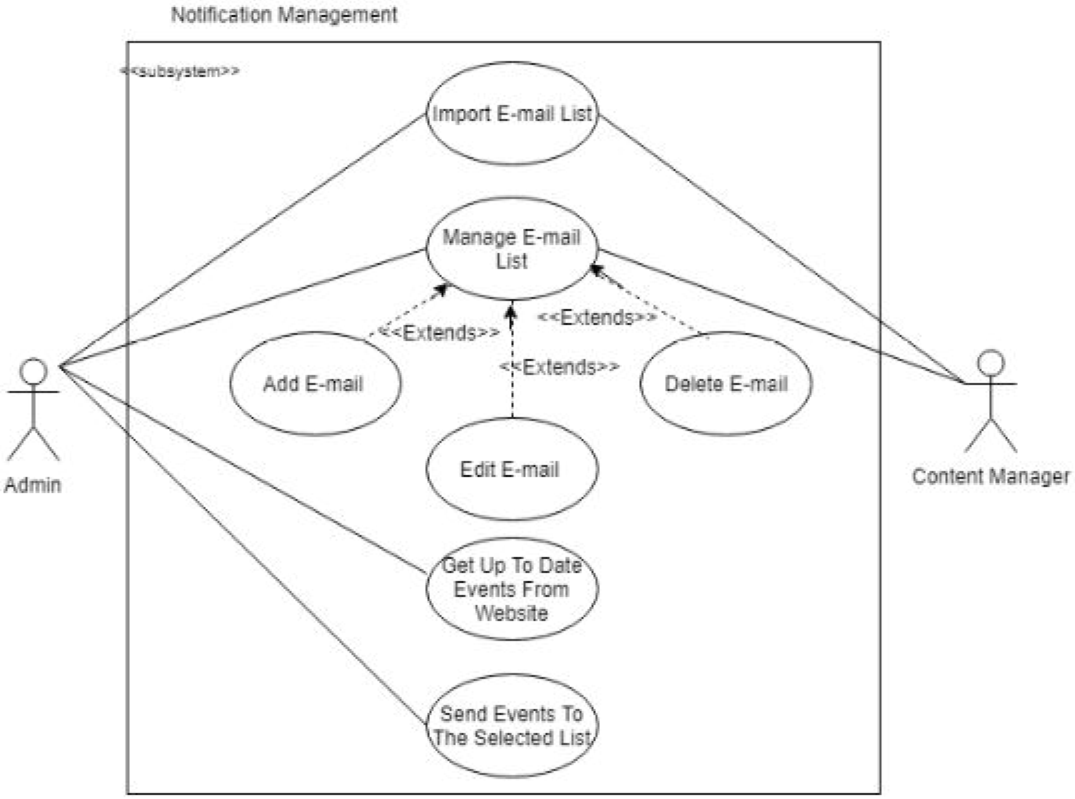


Figure 5: Notification Management

2.2.4. Notification Management Use Case Diagram

2.2.4.1 Use Case: **Import Email List**

Description: It allows users to import email list for the purpose of notifying possible events and announcements may take place at the CENG Department.

2.2.4.2. Use Case: **Manage Email List**

Description: Users can add, edit or delete emails from the emails list.

2.2.4.2.1. Use Case: **Add Email**

Description: Users can add email to the email list.

2.2.4.2.2. Use Case: **Edit Email**

Description: Users can edit email in the email list.

2.2.4.2.3. Use Case: **Delete Email**

Description: Users can delete email from the email list.

2.2.4.3. Use Case: **Get Up to Date Events from the Website**

Description: Admin is able to view existing event content in the CENG website using app.

2.2.4.4. Use Case: **Send Events to the Selected List**

Description: Admin is able to select related email list to send a notification to the member of the email list.

2.2.5. Use Case: **Logout**

Description: User(s) can end the session..

2.3. User characteristics

There are two types of users that interact with the system: admin(s) and content manager(s). Admin(s) is a person such as department instructor(s) or assistant(s) who is given specific permission for managing and controlling the system. Content manager(s) is also a person such as department instructor(s), assistant(s) or secretary who is added by admin. These users need to know English in order to use this application.

Each of these two types of users has different use of the system so each of them has their own requirements.

In user management, only admin can create and delete content manager(s) or admin(s) as a user.

In course management, only admin publishes and enables courses. Besides, both users can add, delete and edit courses. They can also set up a weekly course schedule.

In notification management, only admin can get up to date events from the website and send events to the selected lists. Also, users can import and manage email lists. They can add, edit and delete emails from the email list.

2.4. Constraints

The system depends on the HTTP protocol. If any changes happen in the HTTP protocol, it will affect the system reliability.

The system depends on the WordPress API. If any change happens in the WordPress API, it will affect system reliability.

In order to the system works properly, the database must be in the working state.

Users must preinstall Java 8 or above version in their computer in order to run the CENG WM Desktop App.

2.5. Assumptions and dependencies

There will be an admin in the system initially.

There will be maximum 3 admins and 5 content managers as a user.

CENG website should preserve existing design such as changing a label id for a component will affect the application.

2.6. Apportioning of requirements

A notification can be sent to the admin to publish the course.

User(s) can add new events and push them to the CENG Website.

The Turkish version of this application can be developed.

Editing user feature can be added to the application.

Adding and deleting instructor and assistant features can be added to the application.

Some of the search capabilities can be implemented or improved for ease of use.

3. Specific Requirements

3.1. External Interfaces

The only link to an external system is link to wordpress API in order to get information from CENG website and send updates to the CENG website.

3.1.1 User Interfaces

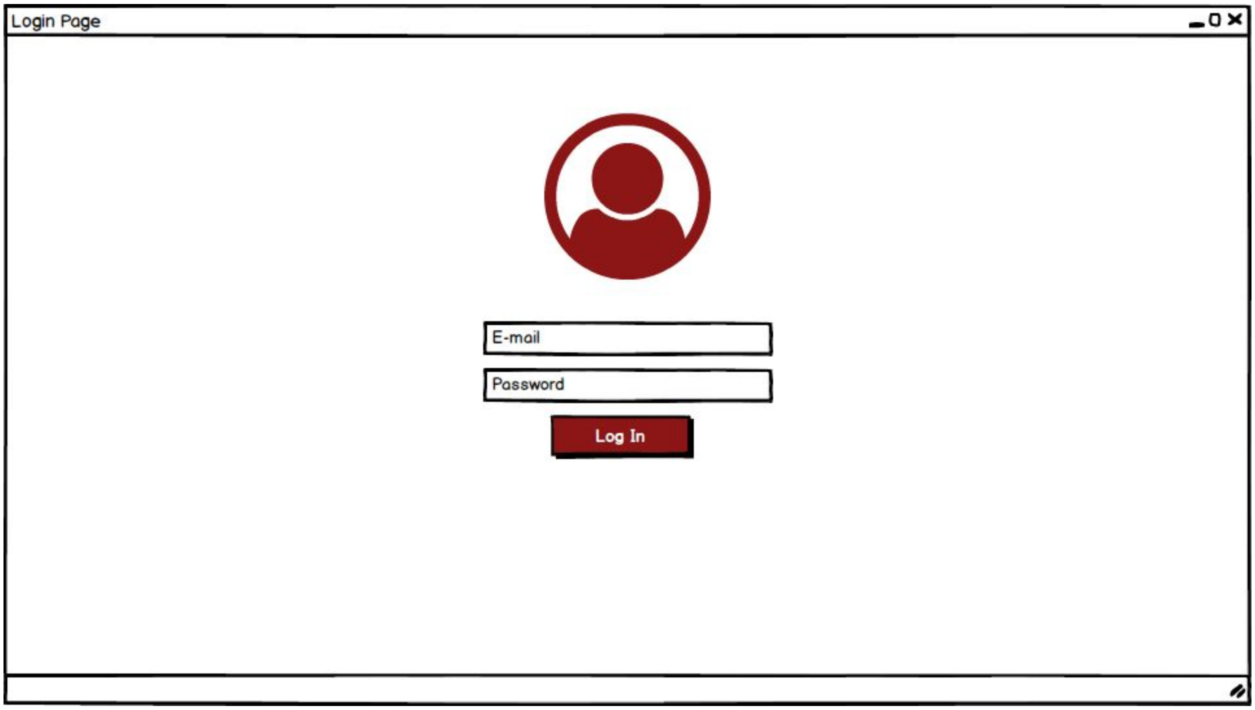


Figure 6: Login UI

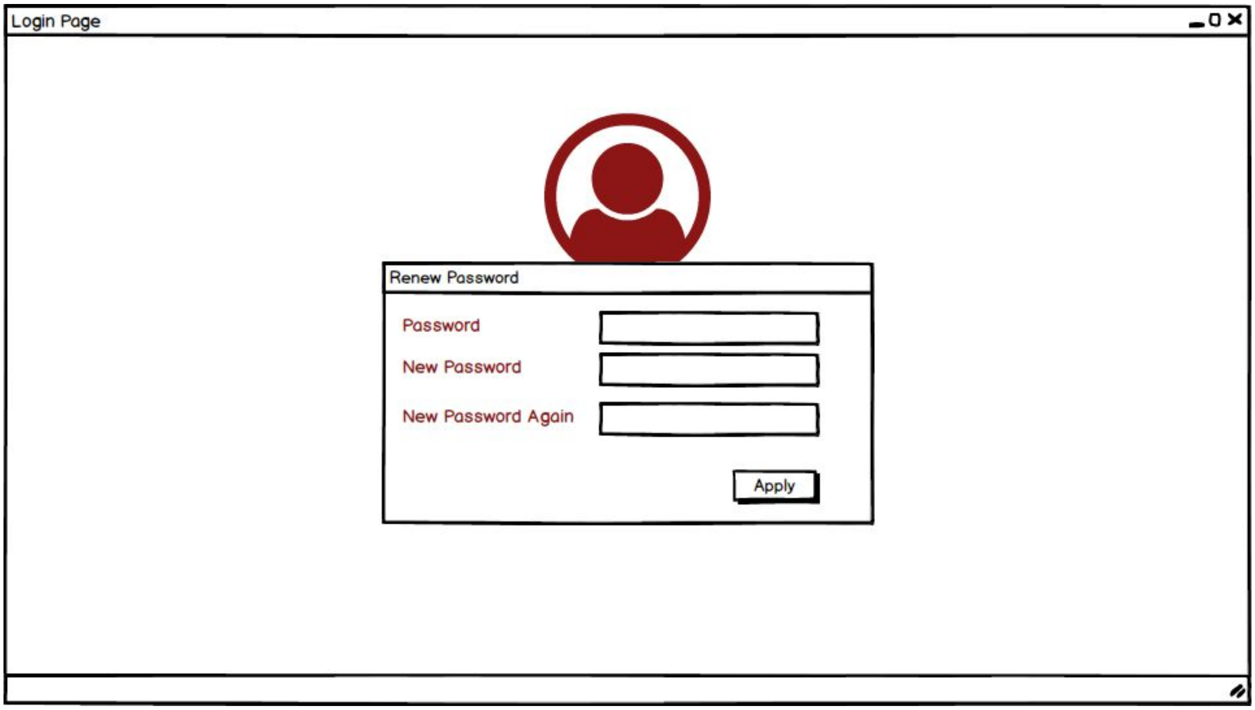


Figure 7: Renew Password UI

Use Case Name	Login
Trigger	User(s) starts the app by clicking icon of the app.
Precondition	User(s) have accessed the Login Page. User(s) shall renew their default passwords on Renew Password pop-up window before first time logging in. User(s) shall sign up before to the system.
Basic Path	1. User(s) fill their credentials to the fields. 2. User(s) click the *Login* button.
Alternative Paths	None.
Postcondition	User(s) log into the system.
Exception Paths	User(s) can not login in case of providing invalid credentials.
Other	Credentials include email(username) and password.
Priority	High

<u>WARNING</u>:After this point we assumed that system login completed successfully.

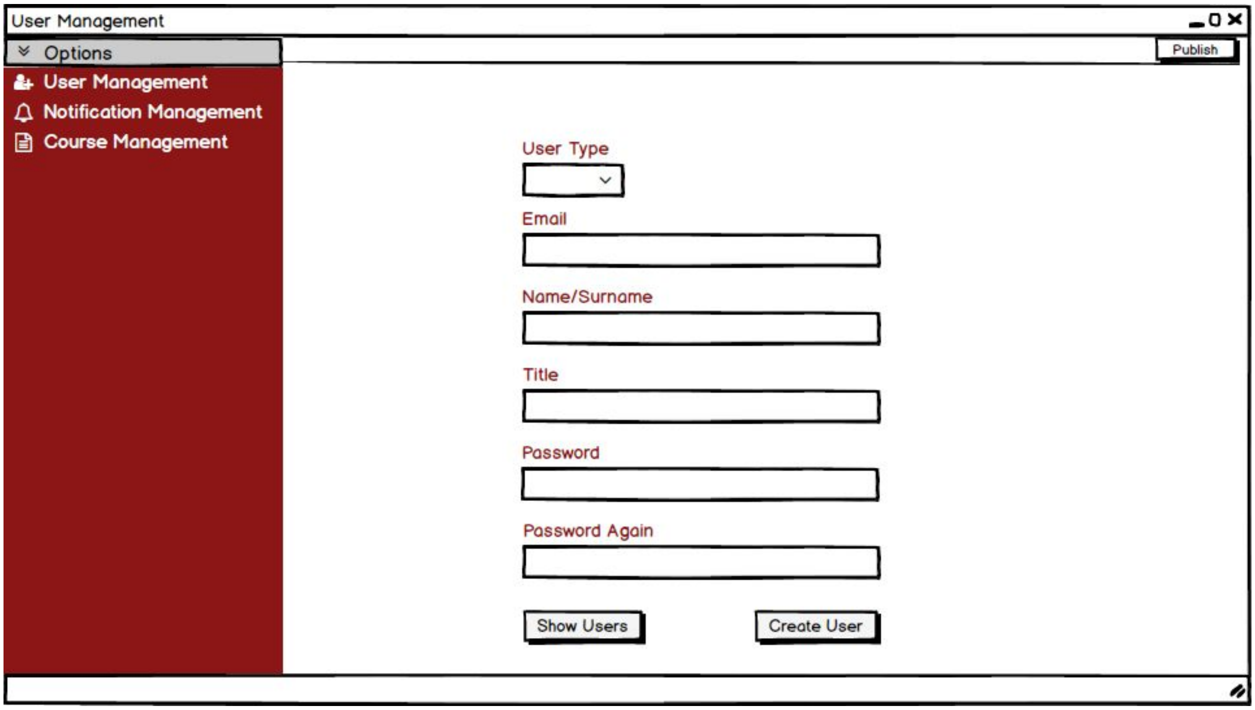
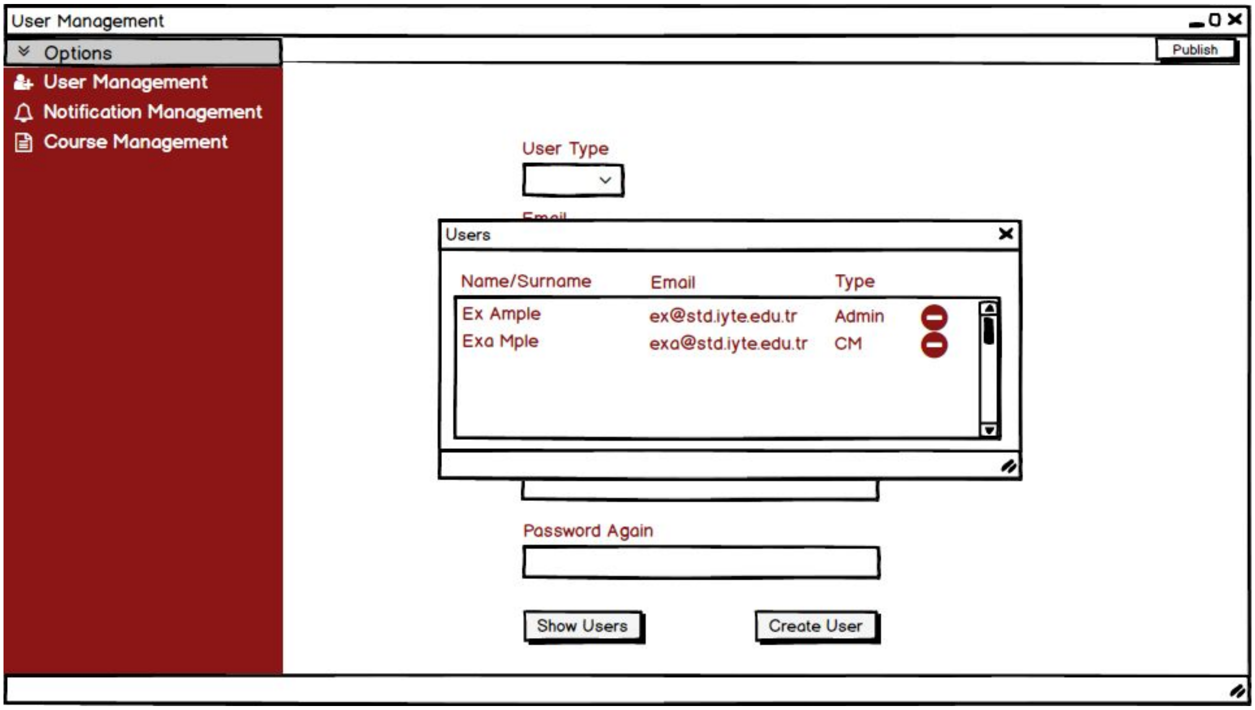


Figure 8: User Management UI

Use Case Name	Create User
Trigger	User(s) select *User Management* from main menu.
Precondition	User(s) shall be an admin.
Basic Path	1. User selects *user type* from dropdown list. 2. User fills username(email). 3. User fills password fields. 4. User clicks to the *create user* button. 5. New user is created and added to the database.
Alternative Paths	Without creating new user, User can show all the created users by clicking *Show Users* button.
Postcondition	User created in selected type.
Exception Paths	1. User can cancel user creation whenever they leave from User Management menu. 2. If password is not valid under specified circumstances in section 3.6, then user will be notified to change the password to a valid one. 3. If the passwords in *Password* and *Password Again* doesn't match, user will be asked to retype the password fields again.
Other	User information include Email, Name/Surname, Title Invalid input includes non matching password and empty fields.
Priority	High



Name/Surname	Email	Type
Ex Ample	ex@std.iyte.edu.tr	Admin
Exa Mple	exa@std.iyte.edu.tr	CM

Figure 9: Delete User UI

Use Case Name	Delete User
Trigger	User(s) select *User Management* from main menu.
Precondition	User(s) shall be an admin.
Basic Path	1. User clicks to *Show Users* button. 2. User find the user they want to delete from the pop-up page. 3. User clicks the *delete (-)* button to the right of the user type. 4. The selected user is deleted from the database.
Alternative Paths	None.
Postcondition	User deleted from database.
Exception Paths	User can cancel user deletion whenever they leave from *User Management* menu.
Other	User's information include Name/Surname, Email, user type.
Priority	Low

Use Case Name	Publish Course
Trigger	Admin(s) click the *Publish* button at the top of the page.
Precondition	User(s) must be an admin. There shall be created course by Admin(s) or CM(s)
Basic Path	1. User(s) create course. 2. Then the *Publish* button becomes clickable on the Admins' screen. 3. Admin(s) click the *Publish* button at the top of the page.
Alternative Paths	None.
Postcondition	The changes are updated in the CENG Website.
Exception Paths	Internet connection failure may occur. In that case user notified to try again.
Other	None.
Priority	High

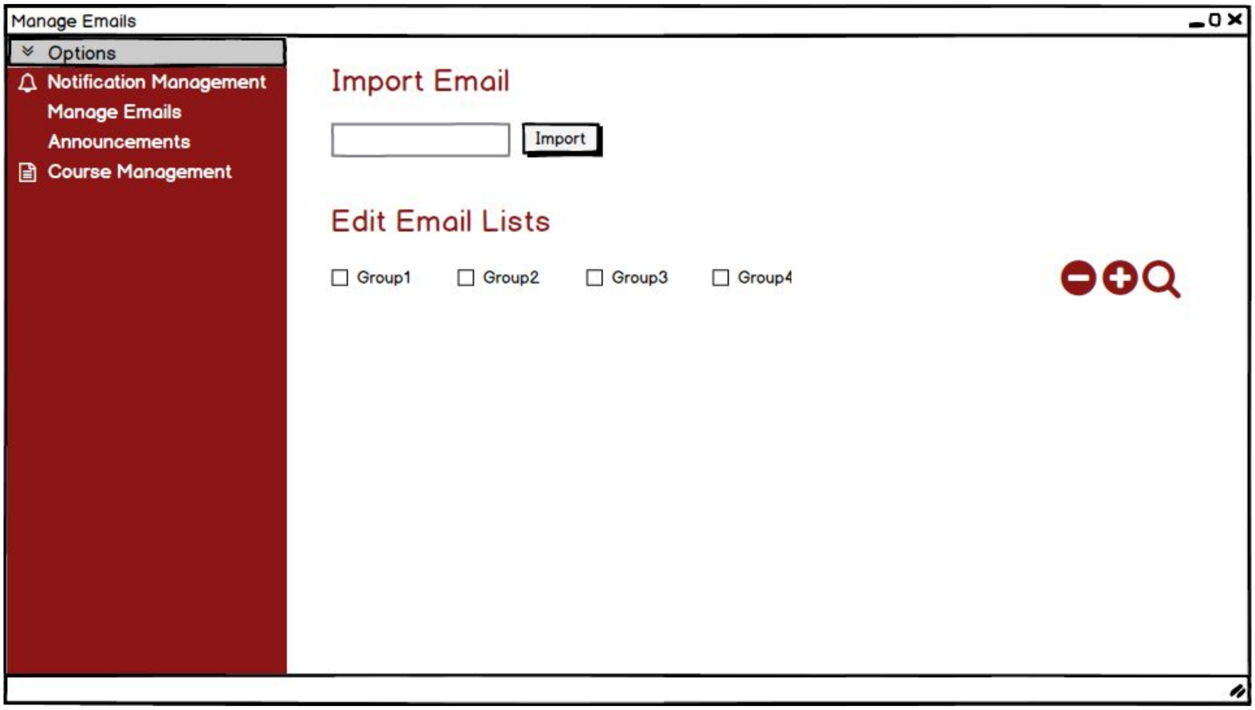


Figure 10: Manage Email UI

Use Case Name	Import Email List
Trigger	User(s) select the *Manage Emails* from the submenu of *Notification Management.*
Precondition	None.
Basic Path	1. User(s) click Import button. 2. User(s) browse in their local file system, find and select a valid email list text file. 3. User(s) click import button.
Alternative Paths	None.
Postcondition	The new mail list is imported to the app.
Exception Paths	User(s) can leave the Import Email page whenever they want.
Other	Valid text files includes .txt files. Each email separated by new line and file name as email list name.
Priority	Medium

Use Case Name	Manage Email List
Trigger	User(s) select the *Manage Emails* from the subsection of *Notification Management*
Precondition	None.
Basic Path	1. From the opening window, user can see *Import Email* and *Edit Email Lists* section.
Alternative Paths	None.
Postcondition	None.
Exception Paths	None.
Other	This use case extends *Add Email List, Edit Email List* and *Delete Email List* use cases*.*
Priority	Medium

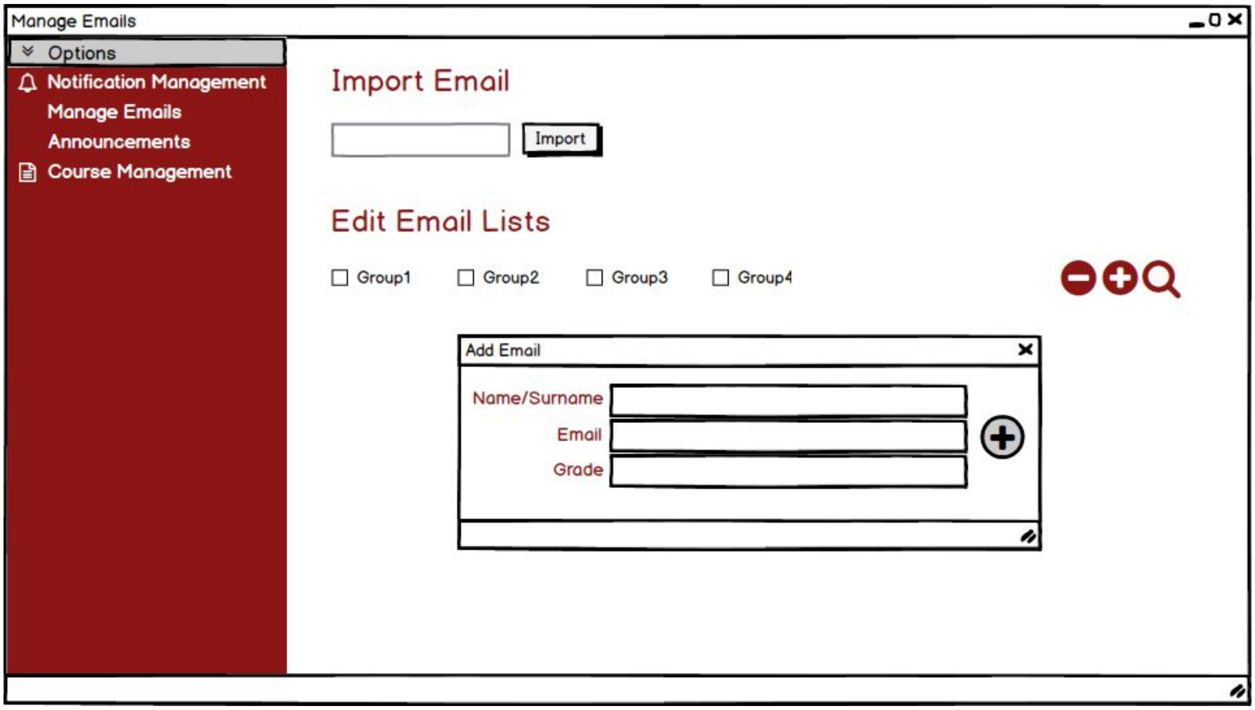
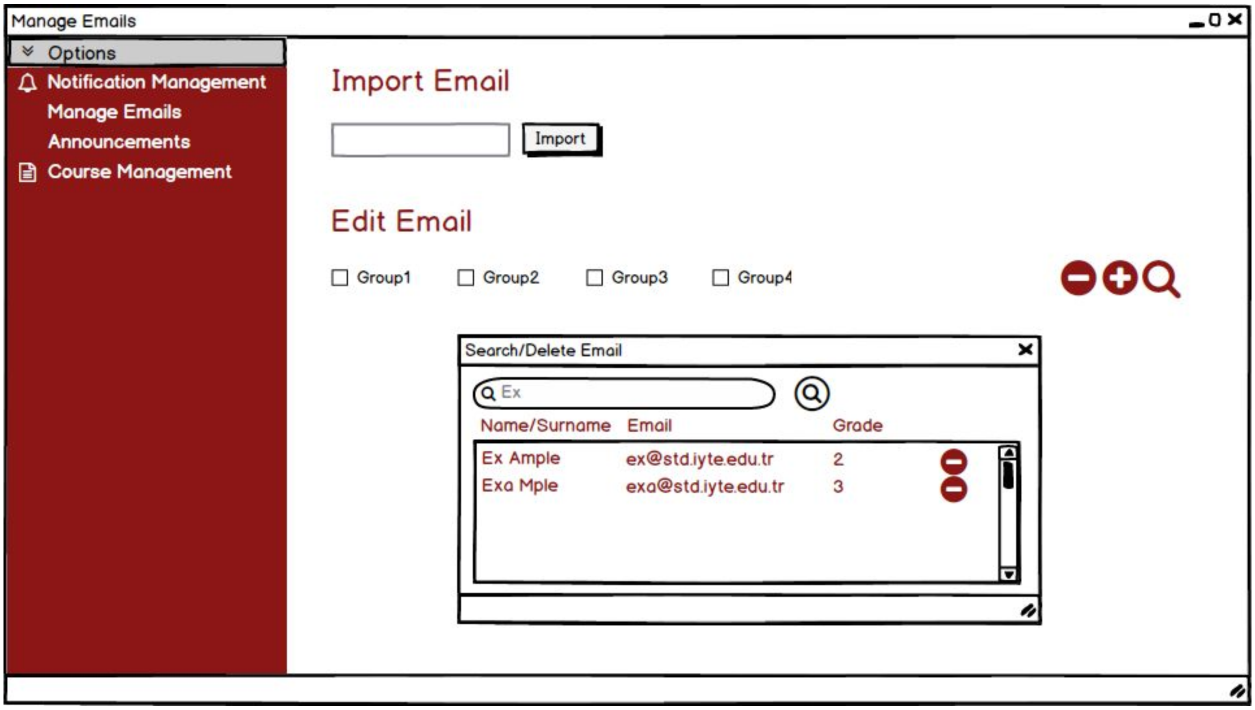


Figure 11: Add Email UI

Use Case Name	Add Email
Trigger	User(s) select the *Manage Emails* from the subsection of *Notification Management*
Precondition	The email shall not be in the system previously.
Basic Path	1. User first selects the email groups by clicking check boxes 2. User click the *plus* button 3. User fills the respective fields accordingly: a. User fills the name/surname field. b. User writes valid email. c. User writes the grade that student is at. 4. After these steps, user clicks the *add* button in order to add email.
Alternative Paths	None.
Postcondition	New email has been added to the selected email groups.
Exception Paths	1. While user adding new email to the selected groups, if the fields are filled with invalid or wrong inputs then system has no responsibility about it. Only way to fix this situation is for user to

	delete the previously wrong added email(according to *Delete Email* description) and add it again with the correct inputs. 2. If user selects the add button without choosing any email groups, then the system will give warning to select the respective email groups in order to add a new email. 3. If user clicks the *plus* button on the *Edit Email Lists* section, then change his/her mind and decided not to add new email, user can simply leave that page by clicking on the top right corner of pop-up window's *cross(x).* 4. If there is no email list created before, user should first import an email list by following *Import Email List* description table steps.
Other	None
Priority	Low



Name/Surname	Email	Grade
Ex Ample	ex@std.iyte.edu.tr	2
Exa Mple	exa@std.iyte.edu.tr	3

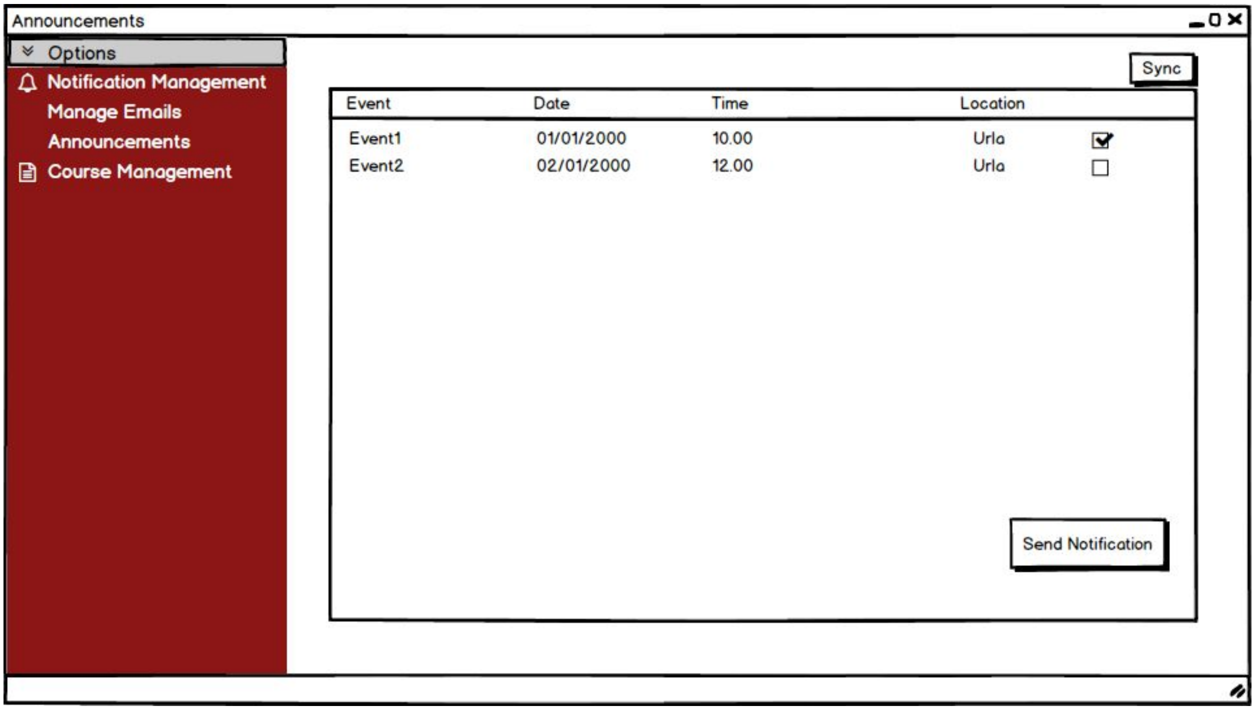
Figure 12: Manage Email Search & Delete UI

Use Case Name	Edit Email
Trigger	User(s) select the *Manage Emails* from the subsection of *Notification Management.*
Precondition	The email list that user(s) want to edit shall be in the app.

Basic Path	1. User(s) selects a email group. 2. User(s) clicks the search button and search for the mail address they want to edit. 3. User(s) fill the fields with new information to edit an email in the opening pop-up.
Alternative Paths	None.
Postcondition	The information in the renewed email list is updated in the app.
Exception Paths	1. If the user changes his mind and decides he does not want to search within the email group, user can simply leave that page by clicking on the top right corner of pop-up window's cross(x). 2. If the user clicks the search button without selecting any email group, the system gives a warning that the email groups must be selected in order to search within email groups. 3. If there is no previously created list of email, the user must first import an email list by following the steps in the Import Email List use case explanation.
Other	In order to edit the email, the fields include name/surname, email and grade.
Priority	Low

Use Case Name	Delete Email
Trigger	User(s) select the *Manage Emails* from the subsection of *Notification Management*
Precondition	The email shall be in the app which user want to delete..
Basic Path	1. User selects a email group. 2. User clicks the search button and search for the mail address they want to delete. 3. User click the minus button to delete email
Alternative Paths	None.
Postcondition	Email has been deleted from the app.
Exception Paths	None.
Other	This use case extends *Add Email List, Edit Email List* and *Delete Email List.*

Priority	Low

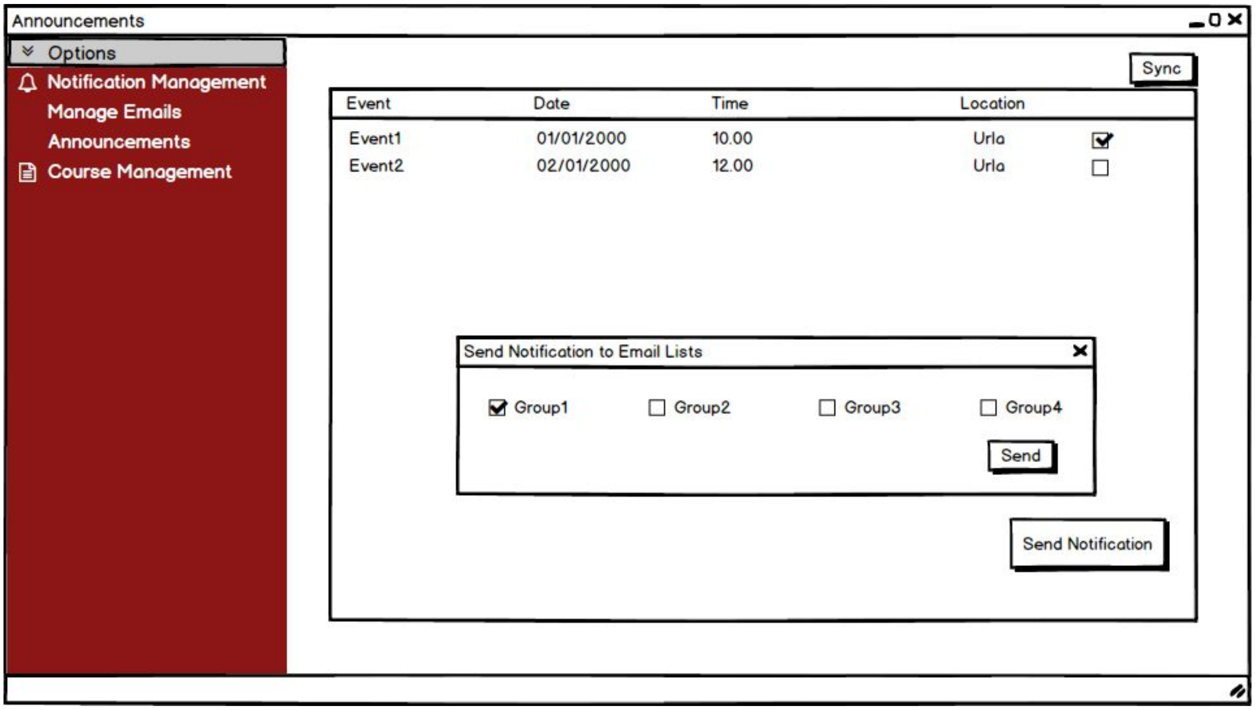


Event	Date	Time	Location	
Event1	01/01/2000	10.00	Urla	☑
Event2	02/01/2000	12.00	Urla	□

Figure 13: Announcements UI

Use Case Name	Get Up to Date Events from Website
Trigger	Admin selects the *Announcements* from the subsection of *Notification Management*
Precondition	Admin’s computer should be connected to the internet.
Basic Path	1. In the announcements admin are able to see the event content that extracted from CENG Website.
Alternative Paths	1. To refresh the menu sync button can be clicked.
Postcondition	CENG Website events listed in the content of scroll pane.
Exception Paths	1. Internet connection failure may occur user notified to click sync button.
Other	None

Priority	Medium



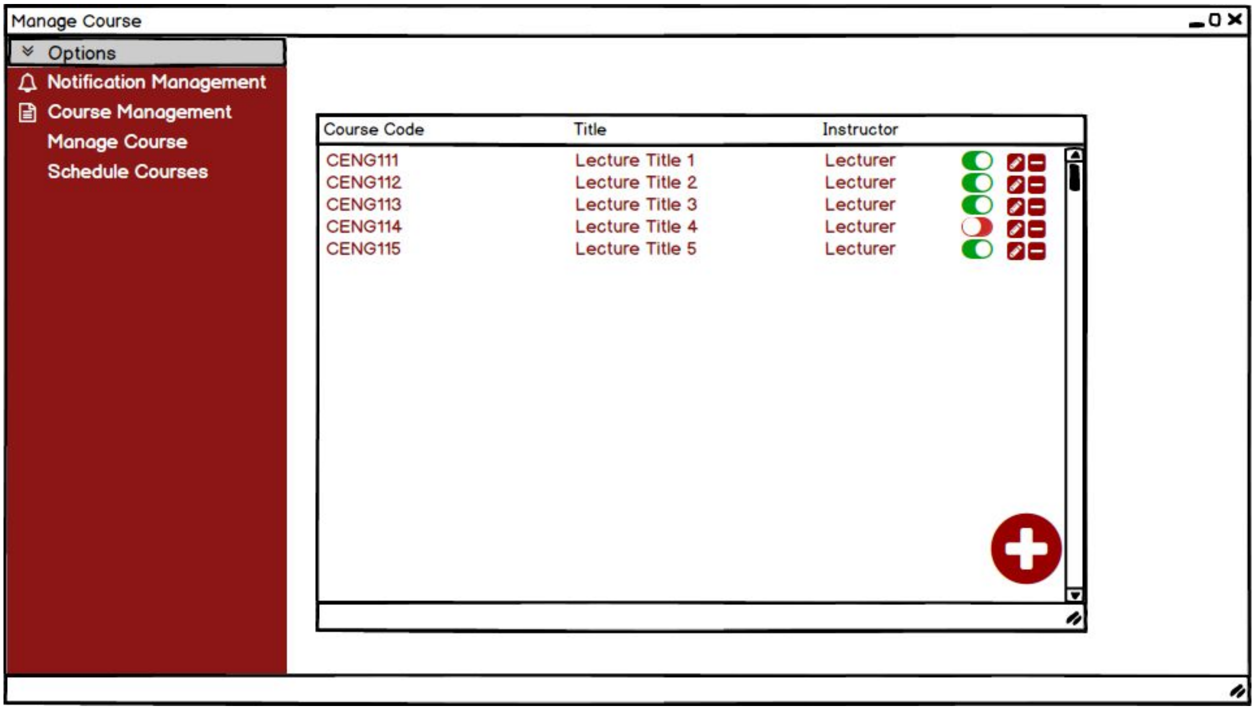
Event	Date	Time	Location	
Event1	01/01/2000	10.00	Urla	☑
Event2	02/01/2000	12.00	Urla	☐

☑ Group1 ☐ Group2 ☐ Group3 ☐ Group4

Figure 14: Send Notification to Email Lists UI

Use Case Name	Send events to the selected list
Trigger	Selecting respective email groups and pushing the *send* button.
Precondition	Admin should click the *Send Notification* button on *Announcements* There is at least one email list group to send notification.
Basic Path	1. Admin select the event(s) that will be send the email groups. 2. After selecting events, admin clicks on the *Send Notification* button. 3. Pop-up window appears which allows selecting the respective email list groups. 4. Admin selects the email list groups and clicks *send* button.
Alternative Paths	None.
Postcondition	The event information is sent to the email groups.
Exception Paths	1. If user(s) leave the page by clicking *cross sign(x),* selected events will not be sent.
Other	None.

Priority	Low



Course Code	Title	Instructor
CENG111	Lecture Title 1	Lecturer
CENG112	Lecture Title 2	Lecturer
CENG113	Lecture Title 3	Lecturer
CENG114	Lecture Title 4	Lecturer
CENG115	Lecture Title 5	Lecturer

Figure 15: Course Management UI

Use Case Name	Manage Course
Trigger	User(s) select *Manage course* from the subsection of *Course Management* menu .
Precondition	None.
Basic Path	1. From the opening window, user can see all the courses with their titles, course codes and instruction names. 2. Also, user can see if a course is an active course or not for that semester. 3. User can change the activity of courses by *enable/disable toggle switch* by following the instructions of *Enable Courses* description table*.* 4. User can edit courses by *edit* button by following the instructions of *Edit Course* description table. 5. User can delete courses by *delete(-)* button by following the instructions of *Delete Course* description table. 6. User can add new course by clicking the *add(+)* button by following the instructions of *Add Course* description table.

Alternative Paths	If the page is full, user can scroll down the page to look down.
Postcondition	Manage course window has opened.
Exception Paths	1. User can leave the page whenever they want by selecting the any other options from main menu. 2. If user can't find the course that s/he is looking for, the system has no responsibility for that. Only thing that user can do is to add a new course by following the *Add Course* description table instructions.
Other	This use case extends *Add Course, Delete Course* and *Edit Course.*
Priority	High

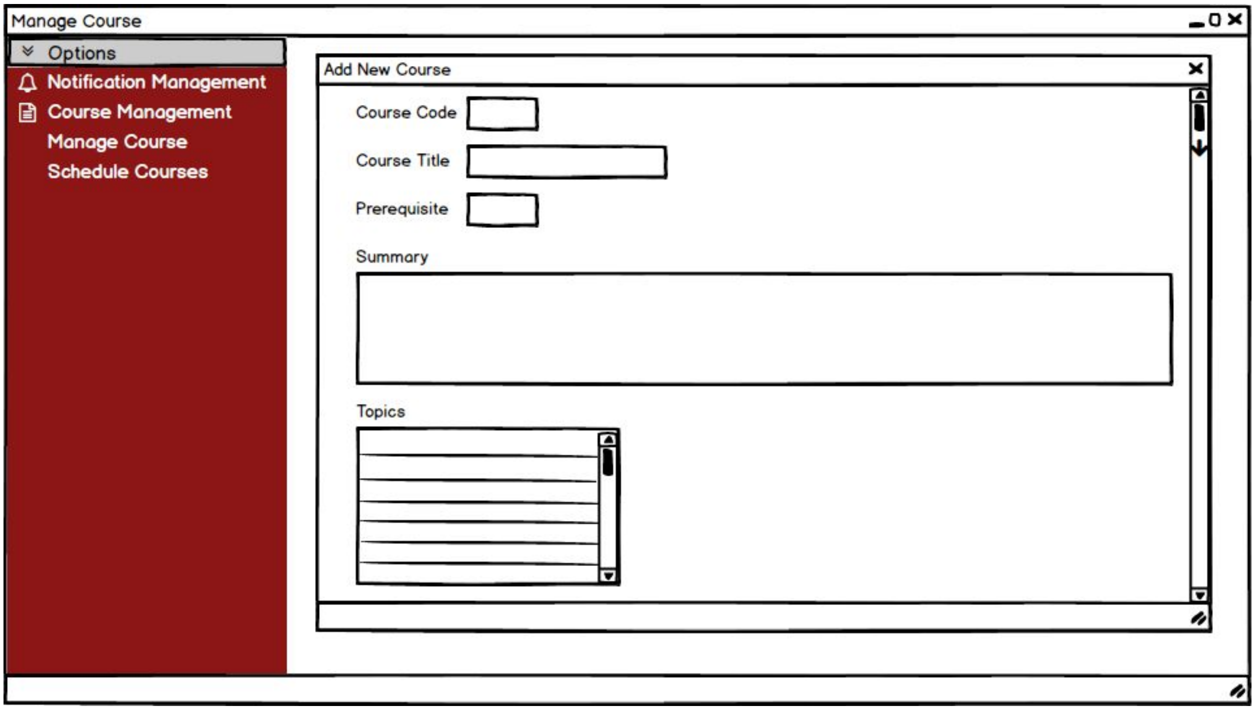


Figure 16: Course Management Add UI

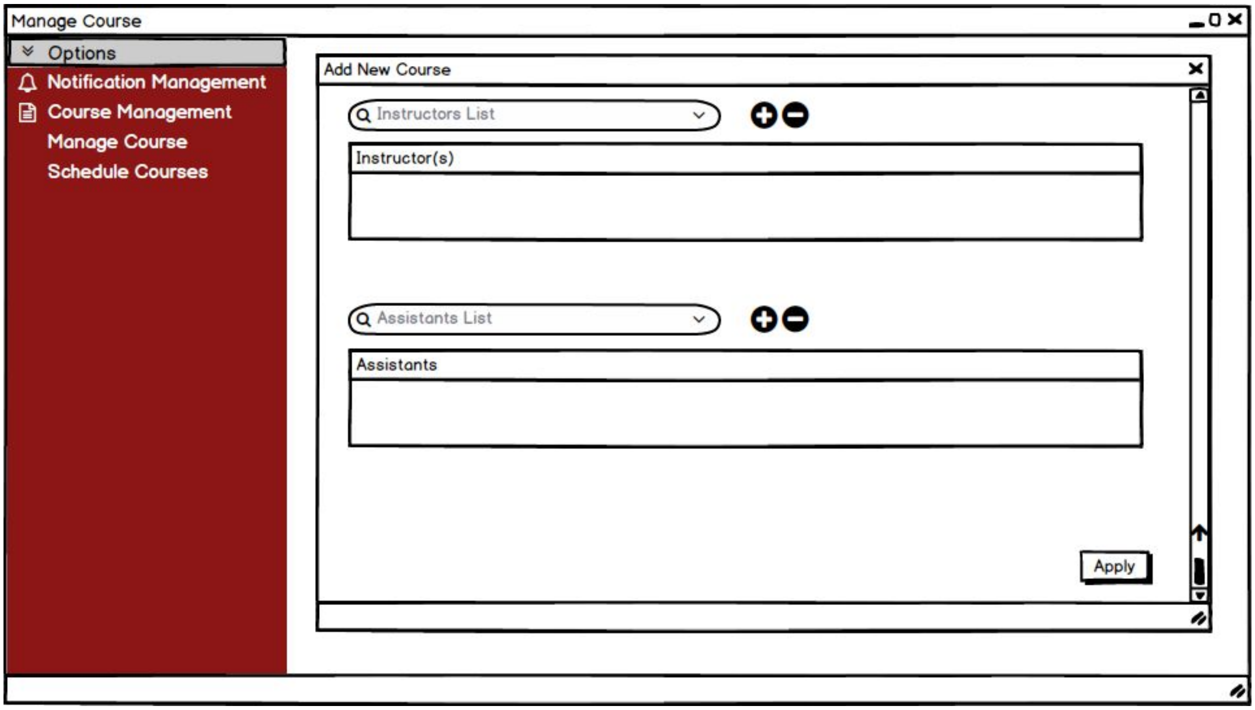


Figure 17: Manage Course Add Instructors UI

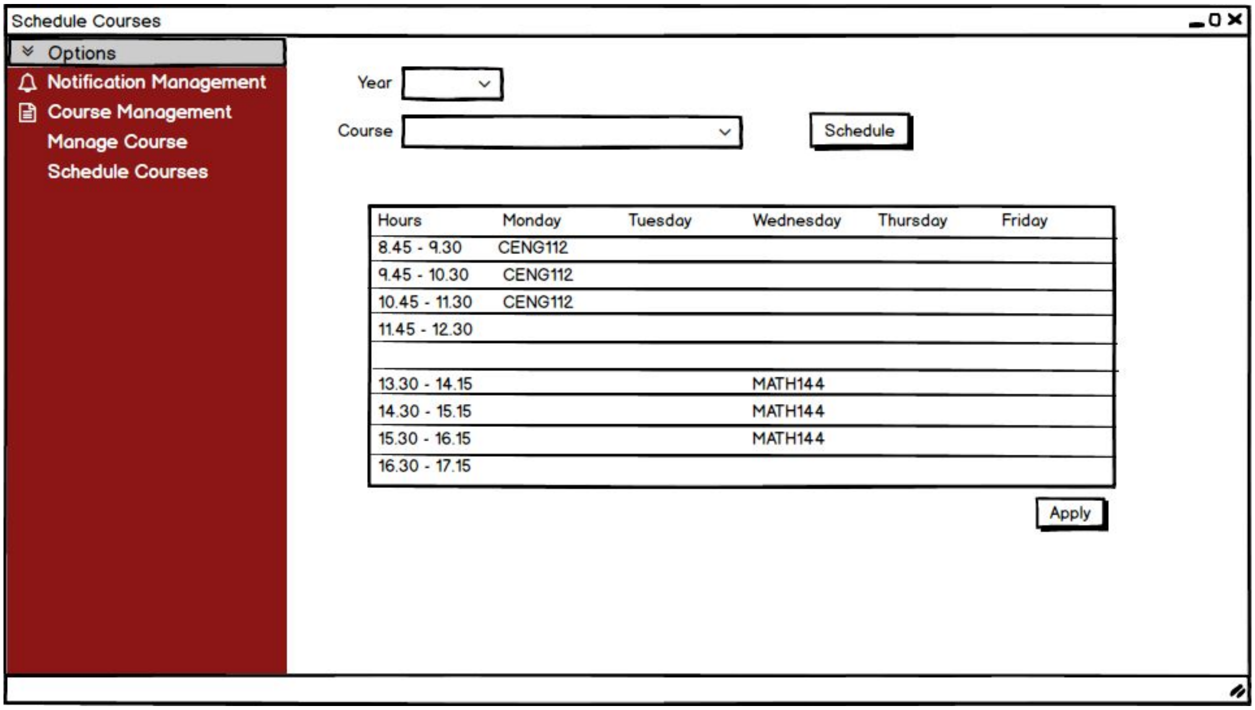
Use Case Name	Add Course
Trigger	User(s) click plus button.
Precondition	User(s) must be in manage course submenu The course shall not be in the system before that user(s) want to add.
Basic Path	1. User click *plus* button from the bottom right of page. 2. User fill course code, course title, prerequisite, summary and topic fields. 3. User select instructor(s) and assistant(s) a. User(s) search for instructor by typing in the instructor or assistant list field. b. User(s) select the instructor c. User(s) click plus button beside the instructor or assistant list field. 4. User click to the *Apply* button.
Alternative Paths	None.
Postcondition	The new course added to the database.
Exception Paths	Invalid data causes exception and user(s) warned to validate their data.

Other	Data includes possible fields in the add course screen.
Priority	Medium

Use Case Name	Delete Course
Trigger	User(s) select the *Manage Course* from the subsection of *Course Management.*
Precondition	The course we want to delete must be exist in the system.
Basic Path	1. The user must click the delete (-) button on the right side of the course he / she wants to delete.
Alternative Paths	None.
Postcondition	The course deleted from the database.
Exception Paths	None.
Other	None.
Priority	Low

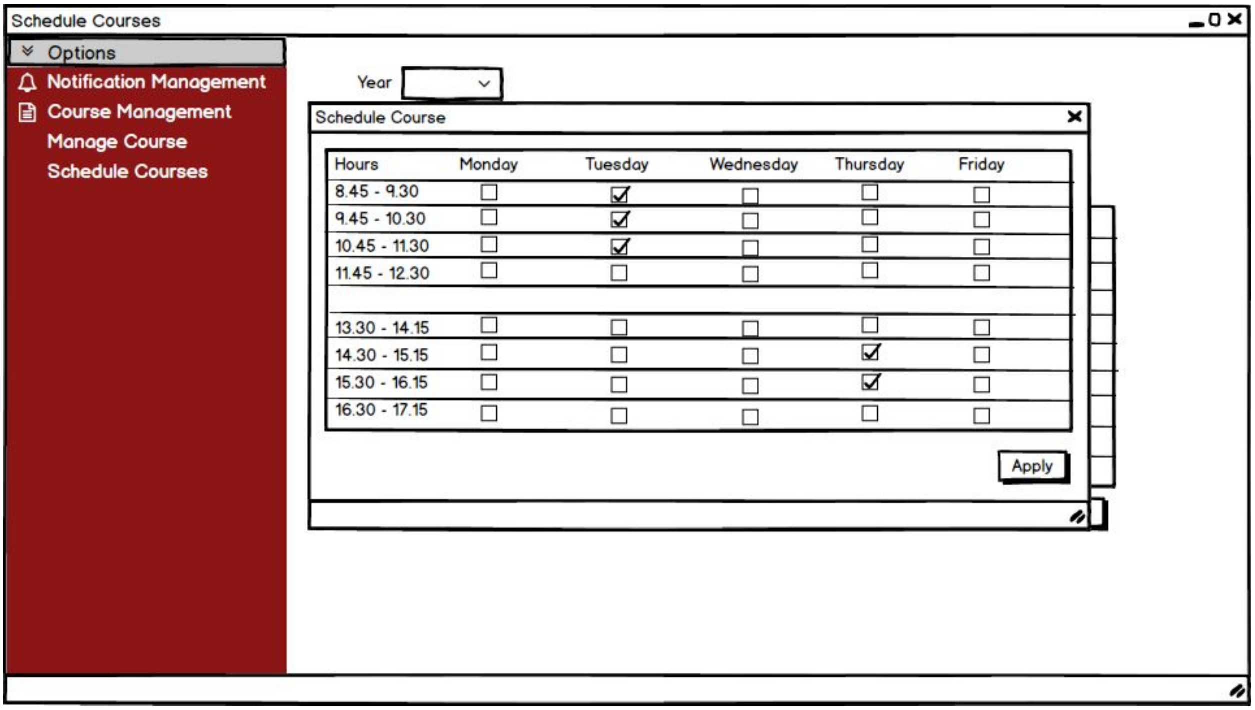
Use Case Name	Enable Courses
Trigger	User(s) select the *Manage Course* from the subsection of *Course Management*
Precondition	There must be at least a course that is added before.
Basic Path	1. User(s) switch the on/off button next to course information from the courses list.
Alternative Paths	None.
Postcondition	The enabled course has been updated as active course in database.
Exception Paths	1. User(s) can not do courses such as Calculus 1 and Calculus 2 active or inactive simultaneously. While one of them is active, the other one shall be inactive or vice versa.
Other	None.
Priority	Medium

Use Case Name	Edit course
Trigger	User(s) select the *Manage Course* from the subsection of *Course Management*
Precondition	The course must be in the system before that user(s) want to edit.
Basic Path	1. User(s) must select *edit* symbol next to the course information line. 2. User(s) can refill the course information such that course code, course title, prerequisite, summary and topic fields. 3. User(s) must click to the *edit* button to save the changes.
Alternative Paths	None.
Postcondition	The course info updated in the database.
Exception Paths	1. User(s) cannot edit course in case of providing invalid input. 2. If user(s) leave the window without clicking *edit* button, the changes will not be saved.
Other	None.
Priority	Low



Hours	Monday	Tuesday	Wednesday	Thursday	Friday
8.45 - 9.30	CENG112				
9.45 - 10.30	CENG112				
10.45 - 11.30	CENG112				
11.45 - 12.30					
13.30 - 14.15			MATH144		
14.30 - 15.15			MATH144		
15.30 - 16.15			MATH144		
16.30 - 17.15					

Figure 18: Weekly Schedule UI



Hours	Monday	Tuesday	Wednesday	Thursday	Friday
8.45 - 9.30	□	☑	□	□	□
9.45 - 10.30	□	☑	□	□	□
10.45 - 11.30	□	☑	□	□	□
11.45 - 12.30	□	□	□	□	□
13.30 - 14.15	□	□	□	□	□
14.30 - 15.15	□	□	□	☑	□
15.30 - 16.15	□	□	□	☑	□
16.30 - 17.15	□	□	□	□	□

Figure 19: Schedule Courses UI

Use Case Name	Set up weekly course schedule
Trigger	User(s) select the *Schedule Courses* from the subsection of *Course Management.*
Precondition	There must be at least one lecture that is enabled before. If user doesn't select year, then s/he will not move further in this page to schedule the weekly course schedule.
Basic Path	1. User selects a year that is going to be scheduled.(ex: 1st year, 2nd year, 3rd year, 4th year ...) 2. User selects the course in order to be added to the schedule. 3. User clicks the *Schedule* button. 4. From opened pop-up, user puts *check mark* to the respective day and lecture hour(s) 5. After selecting necessary lecture hours, user clicks the apply button and pop-up disappears. 6. The up-to-date weekly course schedule will be available for user.
Alternative Paths	1. If user wants to add another course to the schedule that conflicts another course, they can add that course by following above path.
Alternative Paths	1. If user wants to delete a course from schedule, first selects the specified year. 2. User selects the course to be deleted. 3. User clicks the *Schedule* button. 4. The pop-up screen appears. 5. From pop-up screen user, remove check marks from the respective day and lecture hour(s) 6. User selects *apply* button and pop-up disappears. 7. The up-to-date weekly course schedule will be available for user.
Postcondition	Finally by clicking the *Apply* button to save all the changes, the weekly course schedule is set up.
Exception Paths	1. If user makes some changes in the weekly course schedule and doesn't click the *Apply* button and try to leave the page, all the changes s/he has made will disappear. 2. After selecting a course and clicking the Schedule button, if user doesn't do any changes in the selectable weekly course schedule pop-up, by clicking Apply button or cross(x)(in the top right corner of the pop-up) user can leave that page. 3. After selecting a course and clicking the Schedule button, in the pop-up weekly course schedule window, if user makes some

	changes and doesn’t click Apply button and leave that page, all the changes s/he made will not be saved. 4. After selecting a course and clicking the Schedule button, in the pop-up weekly course schedule window, if user makes some changes and doesn’t click Apply button and clicks cross(x)(in the top right corner of the pop-up window), all the changes s/he made will not be saved.
Other	None.
Priority	High

3.2. Functions

3.2.1. Login

There shall be a predefined admin.

There shall be two types of users, admin and content manager.

All users shall be able to login to system using their email and password.

For the first log in there shall be a screen to change the password.

3.2.2. User Management

3.2.2.1. Create User

Predefined admin should be able to create admin.

Admin should be able to create content manager.

Content manager shall not be in the system before admin created it.

Emails shall be unique to each user.

3.2.2.2. Delete User

Content manager who will be deleted, firstly it should be in the system.

Admin should be able to delete content manager.

Admin should be able to delete another admin.

3.2.2. Course Management

3.2.2.1. Manage Courses

3.2.2.1.1. Add Course

The course that will be added shall not be added already.

Course name shall be written in format given below:

CENG (year)(code) -> CENG 113

If there are prerequisite courses, they shall be entered to the system.

All users shall write summary to add the course.

There shall be 14 weeks for each semester. Each week’s topic should be added to the course.

All users shall assign instructor(s) for selected course.

All users should assign assistant(s) for selected course.

3.2.2.1.2. Delete Course

All users shall not delete non-existing course.

All users should know the course code to be deleted.

All users can delete courses by searching course code.

All users can delete courses by selecting course code.

3.2.2.1.3. Edit Course

In order to edit a course, the course shall be in the system.

Except editing course code, all users may edit rest of the course information.

All users can edit:

course name,
pre-requirements,
summary,
topic of the course.

All users can edit courses by searching course code.

All users can edit courses by selecting course code.

All users can assign instructors or assistants for selected course.

3.2.2.2. Set Up Weekly Course Schedule

All users shall select a year in order to schedule the weekly course table.

All users may select or search enabled courses for that semester to add to the schedule table.

All users can add a course as many as they want.

All users can add more than one course to the same lecture hour of same day.

3.2.2.3. Publish Courses

Admin shall publish all courses and schedule on the website.

3.2.3. Notification Management

3.2.3.1. Import Email List

All users can import email list

All users can browse files on local computer to be imported.

3.2.3.2. Manage Email List

3.2.3.2.1. Add Email

All users should view the group(s).

All users can add email to the selected group(s).

All users can select the group(s).

3.2.3.2.2. Edit Email

All users can search email from selected groups.

All users can delete email from searched list

3.2.3.2.3. Delete Email

All users can edit email from searched list.

3.2.3.3. Get Up to Date Events From Website

Admin can synchronize and show up to date events.
Admin can select an event.

3.2.3.4. Send Events to the Selected List

Admin can send the selected event to the email group(s).

3.2.4. Logout

All user(s) may logout from CENG Website management desktop application whenever they want.

3.3. Performance requirements .

Users can add a course in about 60 seconds.
Users can update weekly course schedule in about 120 seconds.
Users can send a notification to email lists in about 30 seconds.
Simultaneous usage of desktop application may cause data loss.
Renew password window shows up in 0.5 second, after application has started.
All pop-up windows in the system show up in about 0.5 second

3.4. Logical database requirements

We have 3 entities (tables) in our database. These are: Course, Teaching Staff and User.
Course and Teaching Staff are connected with Is Given By relationship. Teaching Staff and User are connected with Can Be relationship. Course keeps 7 attributes in our database. These are as seen below course code, course title, prerequisite, topics, active, summary, position. Teaching Staff keeps 4 attributes as seen below: email, title, name, status. User keeps 6 attributes as seen below: user id, email, title, name, type, password. So, the primary keys of these entities are also attributes of relationships they are connected to.

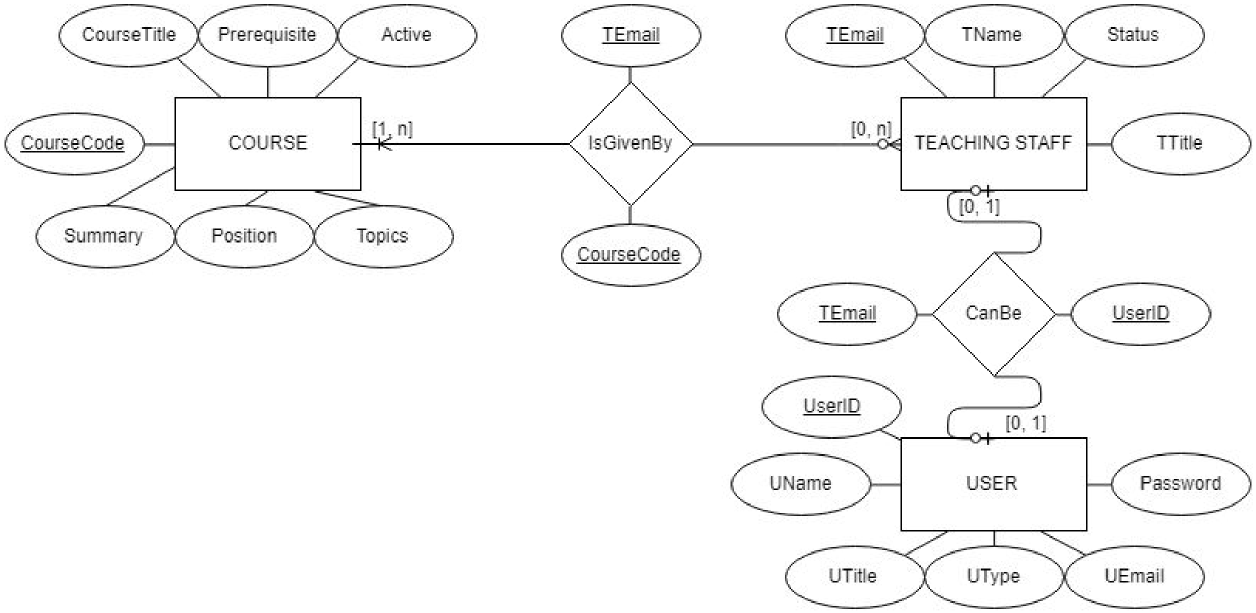


Figure 20: ER Diagram

3.4.1. Reduced Relation

Course (CourseCode, CourseTitle, Prerequisite, Topics, Active, Summary, Position)
TeachingStaff (TEmail, TTitle, TName, Status, UserID)
User (UserID, UEmail, UTitle, UName, UType, Password)
IsGivenBy (CourseCode, TEmail)
~~CanBe(UserID, TEmail)~~

3.4.2. Entities

3.4.2.1 Course Data Entity

Data Item	**Type**	**Description**	**Comment**
Course Code	String	The unique code for a course.	Used as a primary key in database.
Course Title	String	The name of the course.	None.
Prerequisite	String	Course that must be completed with a satisfactory grade before enrolling another course.	May be several prerequisite courses.

Topics	String	It tells the each week’s course content in a compact way.	There are 14 topics for a semester.
Active	Bit	It indicates if a course is active or not for that semester	It takes 0 or 1 values.
Summary	String	It gives a preview for the course.	None.
Position	Pointer	By this integer number, we will know where each lecture should be located in the weekly course schedule.	None.

3.4.2.2 Teaching Staff Entity

Data Item	**Type**	**Description**	**Comment**
TEmail	String	Email of teaching staff of CENG department.	Used as a primary key in database
TTitle	String	Levels of professional expertise	None.
TName	String	Name of teaching staff of CENG department.	None.
Status	Integer	It indicates if teaching staff is a faculty member or research assistant.	None.
UserID	Integer	ID number of each user.	Used as a foreign key in database

3.4.2.3 User Entity

Data Item	**Type**	**Description**	**Comment**
UserID	Integer	ID number of each user.	Used as a primary key in database
UEmail	String	Email of users	None.
UTitle	String	It indicates that if user is a faculty member, research	None.

		assistant or administrative staff.	
UName	String	Name of the users	None.
UType	String	It indicates if user is a Content Manager or Admin.	None.
Password	String	Password of users in hashed format.	None.

3.4.2.4 IsGivenBy

Data Item	**Type**	**Description**	**Comment**
Course Code	String	The unique code for a course.	Used as a primary key in database
TEmail	String	Email of teaching staff of CENG department.	Used as a primary key in database

3.5. Design constraints

3.5.1 Standard Development Tools

App core functionality will be developed by using Java 8 SE environment. For GUI implementation JavaFX and ControlsFX libraries will be used.

3.5.2 Hardware

Theoretically app can work all the Java Virtual Machine(JVM) implemented machines. However, our suggestion for hardware is:

Core 2 Duo and above CPU, minimum 3GB RAM, Ethernet or WiFi to establish internet connection for display output at least 1024x768 resolution screen.

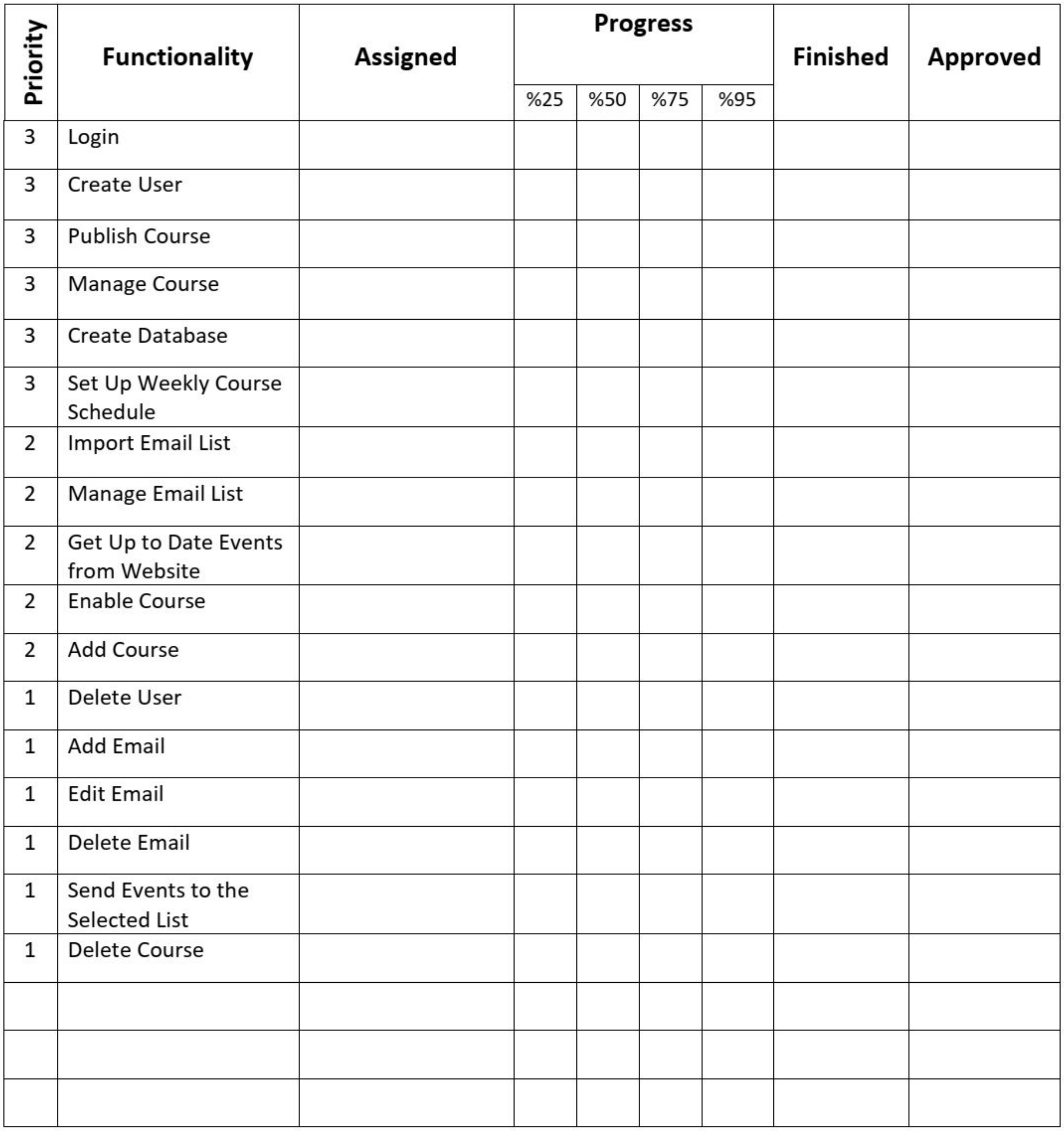
3.6. Software system attributes

User password shall be at least 6 character long.
User password shall be at most 14 character long.
User password shall contain at least a number.
User password shall contain at least an uppercase letter.
User password shall contain at least a symbol.

User password shall be kept as encrypted with SHA256 in database.
“$”,”\” and “=” shall not be accepted as username and user password.

4. Appendixes

Appendix A: Traceability Matrix



Priority	Functionality	Assigned	Progress				Finished	Approved
			%25	%50	%75	%95		
3	Login							
3	Create User							
3	Publish Course							
3	Manage Course							
3	Create Database							
3	Set Up Weekly Course Schedule							
2	Import Email List							
2	Manage Email List							
2	Get Up to Date Events from Website							
2	Enable Course							
2	Add Course							
1	Delete User							
1	Add Email							
1	Edit Email							
1	Delete Email							
1	Send Events to the Selected List							
1	Delete Course							