**Outline:**

1- Introduction……………………………………………………………………

1.1- Problem……………………………………………………………………

1.2- Solution……………………………………………………………………

2- Objectives………………………………………………………………………

3- High-Level Functionality………………………………………………………

3.1- Application Overview……………………………………………………

3.2- Functional Requirements…………………………………………………

3.3- Non-Functional Requirements……………………………………………

4- Stakeholders……………………………………………………………………

5- Process model…………………………………………………………………

6- Project Risks……………………………………………………………………

7- Project Needs…………………………………………………………………

8- Measurements…………………………………………………………………

9- Tools……………………………………………………………………………

10- Project Schedule………………………………………………………………

11- Project pay-offs………………………………………………………………

12- User Interfaces………………………………………………………………

**1- INTRODUCTION**

**1.1- Problem**

Currently, universities use web-site systems to support students and faculty in their academic activities. These systems facilitate tasks such as course enrollment, messaging, and viewing announcements for students, while enabling instructors to upload materials and manage their courses. However, these functionalities often come from different applications and software, which are not unified into a single system. In addition, different universities use different applications for their systems.

Moreover, nearly all of these systems lack social features, and while, more than 1 application for 1 university makes it challenging for students and faculty to navigate between multiple platforms.

**1.2- Solution**

To address the navigation issues faced by students and faculty, we propose unifying multiple universities and their functionalities into a single, unified system. This system will be accessible via the internet and through a mobile application. Our primary motivation for this solution is to make people not struggle to use multiple systems, which can indirectly lead to contractual lock-ins and high maintenance costs. By offering a cost-effective, all-in-one functional system, we aim to streamline university operations and reduce expenses.

Additionally, to enhance the social environment, we will upgrade existing profile systems by adding personal background sections, profile icons, and photos. We will also introduce student communities, which will be created and maintained by students based on specific interests or contexts. These communities will enable students to seek help, interact with other people, and improve their social skills.

After the development phase, universities can sign up by contacting us, and we will ensure that only verified students from registered universities have access to the system.

**2- OBJECTIVES**

Here are some of the objectives on this Project that we aimed to achieve both after launch and in development process.

1- Designing a simple GUI so that everyone can understand the system easily. For example, without any knowledge, users can find a way to access the help section about the platform. This can be measured by surveys or customer feedback.

2- Implementing everything according to plan with the most efficiency and safety. The system must be implemented so that usage of resources is minimalized without loss of functionality, and in any type of big issue (system slowdown due to overusage, DBMS crash), the system must ensure users receive minimal harm.

3- Average response time for course enrollments below 60 milliseconds.

4- Existing communities should immediately transferred from old system to our new system after launch.

5- Security of application should be maintained continuously by us.

6- After launching the project, we expect to get at least 2 universities working with us in the first year.

**3- HIGH-LEVEL FUNCTIONALITY**

**3.1- Application Overview**

Our developed software is mobile application which has mainly 5 parts, more shown in gui part of report. And explained briefly in here.

1- Mails:

Users can recieve or send mails to any people here.

2- Announcements:

Students or teachers will get announcements and can read from here.

3- Community:

Students can create or join communities for diverse events.

4- Courses

Students or teachers can download-upload materials here, and students can enroll courses or any kind of course actions done here, from online-asynchronous courses to document requests.

5- Profile

People can write public personal informations here and upload photo, icons and this section can be displayed in the community section.

**3.2- Functional Requirements**

Here are some of the functional requirements that we specified and explained briefly here. Some of them explained in the overview.

1- System should provide managing course materials to teachers.

2- Asynchronous lectures.

3- Announcements.

4- Providing profile section.

5- System activity report:

System must save activities of communities end of every month in database, these informations will stay at least 6 months. And stay accessible from authorized peoples.

6- Mailing system for communicating with both students and teachers.

**3.3- Non-Functional Requirements**

Here are 6 non-functional requirements specified for Project.

1- Easy to navigate user interface, even for the first-time users.

2- System should be available for users 24/7 with at most 5 second downtime.

3- System must give response not longer than 0.1 seconds whenever user sent to email anyone, or in course enrollments.

4- When accessed to courses section, system must ask additional pin code for more security.

5- System must maintain the performance while getting larger by new universities added to system.

6- System should support both iOS and Android on mobile application.

**4-STAKEHOLDERS**

**5- PROCESS MODEL**

We selected scrum for this Project which is simple framework and theory for general value creation with group of people.

Reasons to choose this model listed below;

1- Flexibility, since our Project related with education and education linked with too many regulations, and practices. We need to update our application continously and maintain the validity of the Project.

2- Team collaboration, scrum makes teams more effective in agile aproaches. In this Project we can only maintain these team effectively by using scrum.

3- Risk management, scrum enables us to see and identify risks earlier and take action accordingly.

4- Overall scrum enables team members to take responsibility and increase their motivations during Project development and sprints are makes team members stay focused on goals.

We have done sprint planning as follows;

Sprint 1: Course section

Course section’s interface and pin section implementation with material management, document requests, asynchronous lecture support.

Sprint 2: Profile section

Profile section gui with public personal background information section, photograph taking functionalities with course completion badges, and achievement indicators.

Sprint 3: Community section

Community section implementation in gui, which students can join communities they want and they can have discussios here, exchange ideas, joining events etc.

Sprint 4: Communication

Communication will be provided mostly by mails section, in this sprint we planned implementation of mails, how people will use it and gui design.

Sprint 5: Reports and analysis

Reporting course completion rates and students progress systems will be developed.

Sprint 6: Security

All necessary systems for log in and registeration to system. Seperate authorization levels will be determined like administrator, student, teacher etc. User passwords will be stored with necessary encryption methods.

**6- PROJECT RISKS**

**7- PROJECT NEEDS (software hardware support)**

**8- MEASUREMENTS**

**9- TOOLS**

**10- PROJECT SCHEDULE**

We created gantt chart for our Project and Scheduled the Project tasks. Here is the explanation of Schedule tasks shown below.

1-Project management. (41 days)

In this task we firstly define goals in 5 days and conduct meetings with stakeholders, we will review requirements and plan the Project in detail.

2- Training. (37 days)

This part consist of training of selected tools fort his Project.

3- Design. (17 days)

Designing ui models, database schema and defining system architecture.

4- Test case preperation (12 days)

Specifying test plans and writing test cases.

5- Implementation(131 days)

Longest task in our Project, consist of implementation of all gui parts, and functionality.

Gantt Chart is shown below;

A screenshot of a computer

Description automatically generated

Fig.1 Project-management to implementation.

A screenshot of a computer

Description automatically generated

Fig.2 implementation.

A screenshot of a computer

Description automatically generated

Fig.3 Implementation. (cont.)

**11- PROJECT PAY-OFFS**

**12- USER INTERFACES**

**13- CONCLUSİON**

In this report we discussed all of the work done for our Software Project UniLearn. We have discussed problem definition, Project objectives, requirements and overview of the Project, scheduling, risks, measurements… clearly.