



İZMİR EKONOMİ ÜNİVERSİTESİ

Software Project Final Report

UniLearn

Group-1

Mehmet Şakir ŞEKER	20210601056
Demirkan YILDIZ	20220601077
Sarp DEMİRTAŞ	20220601016
Cavit KAYA	20220601045
Melik Fırat GÜLTEKİN	20210601029
Sertan ÜNAL	20170601032

OUTLINE

1 - INTRODUCTION	2
1.1. Problem	2
1.2 Solution	2
2 - OBJECTIVES	2
3 - HIGH-LEVEL FUNCTIONALITY	3
3.1. Application Overview.....	3
3.2. Functional Requirements.....	4
3.3. Non-Functional Requirements	4
4 - STAKEHOLDERS.....	4
5 - PROCESS MODEL	6
6 - PROJECT RISKS	7
7 - PROJECT NEEDS	9
7.1. Hardware Needs.....	9
7.2. Software Needs.....	9
7.3. Support Needs	9
8 - MEASUREMENTS	10
9 – TOOLS	10
10 - PROJECT SCHEDULE	11
11 - PROJECT PAY-OFFS	13
12 - USER INTERFACES	14
13 – FUTURE PLANS	18
14 - CONCLUSION.....	19

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

The aim of this project is to develop a digital platform that combines academic and social functions for universities. This system aims to facilitate access to information for students and faculty, reduce operational costs and improve communication. The main objectives include the items listed below:

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

The software we developed is a comprehensive application designed to combine and facilitate academic and social activities at universities. The app is divided into five main sections, each serving a different purpose to improve user experience and functionality:

1- Mails:

Users can send and receive e-mail within the university network, ensuring uninterrupted communication between students, faculty and administrative staff.

2- Announcements:

This section creates a central location for all important university announcements. Students and teachers can easily access and read university news, keeping them informed about critical events, deadlines and information.

3- Community:

Students can create or join various communities based on their interests or academic needs. This feature encourages increased social interaction and networking among students.

4- Courses

The course management system allows students to enroll in courses, download and upload course materials, and participate in various course-related activities. Instructors can manage their courses, distribute materials, and perform administrative tasks efficiently. This section supports both synchronous and asynchronous learning, meeting a variety of educational needs.

5- Profile

Users can create their profiles by adding personal information, photos, and icons, and personalize them by adding their skills and educational background. These profiles can be

viewed in the community section and help improve social interactions and create a more connected university community.

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- The system should allow universities to upload and manage their course materials.
- 2- All announcements related with a user on the platform, information about courses, social messages, and notifications of events will be sent to the user
- 3- Application must allow users to customize their profiles, ex. writing public personal information, and a bio.
- 4- Students can view announcements or events that other universities allow them to do.
- 5- The system should report/save activity information about student communities and university clubs end of every month to database, these informations will stay at least 6 months and will be accessible by authorized people, ex. university IT manager.
- 6- The system should deliver interactive lessons, quizzes, assignments, and project tasks to students. Offering real-time interactive classes such as video conferences or live streams.

3.3. Non-Functional Requirements

Here are 6 non-functional requirements specified for Project.

- 1- The system has user-friendly interface, even for the first-time users.
- 2- The system should be available for users 24/7 with at most 5 second downtime.
- 3- The 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- The system must maintain the performance while getting larger by new universities added to system.
- 6- The system should support both iOS and Android on mobile application.

4 - STAKEHOLDERS

There are eight types of stakeholders in this project;

- Students
- Lecturers
- Scrum Team
- Student Affairs
- University Administration
- Publishers

- Scholarships
- Career Center

1. Students

Students have benefit from the software throughout their courses. Since our software is based on academic purposes, they are the most important internal stakeholder group. They are going to benefit from the richness of content and information, easiness of accessing materials visually. They also can give feedback to development team about design of the program.

2. Lecturers

Lecturers are one of the stakeholders of our software. They have benefits from the popularity of their university and if they are independent lecturers, they directly have benefits from the popularity of the software. They are going to find the possibility of spread their fame and advancing in their career.

3. Scrum Team

As they have benefit in the development, scrum team is also the direct creator of the project, with the suggestions of other stakeholders such as students and lecturers. If the program is successful, they are going to earn a lot of respect in software community, and they are going to be more preferable for upcoming projects in the future.

4. Student Affairs

The student affairs office is an important stakeholder of the project in real life, as it is the primary communication center that provides services such as accommodation, scholarships and academic consultancy for students who have registered or will enroll at the university.

5. University Administration

University administration is the main stakeholders responsible for the general operation and management of the institution. These may include positions such as the university rector, department heads and the board of directors. Management group establishes the policy of the university by overseeing the budget and academic programs. They evaluate the general effectiveness of the student information system and make strategic decisions.

6. Publishers

Publishers are stakeholders who produce and distribute textbooks, supplementary resources, and academic materials. They develop textbooks and related resources by collaborating with universities. They also provide valuable data on updating materials in the system.

7. Scholarships

Scholarship offices are units that provide financial support to students at the university and manage scholarships. They can use the system to monitor the academic performance of students and determine whether they meet the scholarship requirements.

8. Career Center

Career centers help a university on achieving career development, employment processes and professional goals. They provide internship and job opportunities and career counseling to students.

This center is an important stakeholder in terms of student connections, career training and development.

5 - PROCESS MODEL

For this project, we chose Scrum, a simple yet effective framework that will make it easier to create value within a team. Scrum's adaptability and focus on iterative progress make it an ideal choice for our project, which involved developing a comprehensive application for the university's academic and social activities. The main reasons why you should choose this model are:

- 1- Flexibility, since our Project related with education and education linked with too many regulations, and practices. We need to update our application continuously and maintain the validity of the Project.
- 2- Team collaboration, scrum makes teams more effective in agile approaches. 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- Continuously stakeholder meetings and inspection, scrum has efficient method to maintain validity, called sprint review. In these meetings stakeholders have important place for checking the validity of the requirements implemented during sprint.
- 5- In scrum everything is transparent and can be seen by all stakeholders, especially who will receive the results of the Project.
- 6- 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. So, we will use scrum also for it's values.

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 discussions 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 registration 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

Combined risk table shown below with descriptions

Likeli-Hood Rank	Impact Rank	Combined Rank	RISK DESCRIPTION
1	3	1	Management Risks: Arising from project managers and executors, these can lead to project failure due to incorrect execution. These include prioritizing tasks incorrectly, failing to fulfill responsibilities, lacking resources, hiring inexperienced personnel, inadequate experience or staffing, lack of resource planning, failure to facilitate team communication, and similar factors
6	4	2	Insecure implementation : Tool related bugs, or any kind of vulnerability due to implementation process is inevitable. And if some of the bugs that not identified before launch can give serious problems to users.
5	5	3	Tool-related bugs & exceptions: As we use front-end libraries for our project, it isn't feasible to learn all the details without abstraction. Even if the documentation is enough for React (we have selected this), our programmers can make a mistake using that library. Then, the program might have a bug that could be identified later in the project. Or some exceptions, which are easier to identify and fix in sprints, can occur. However, it is not likelihood that getting bugs or exceptions from only the abstraction of the library.

11	1	4	Requirements not-matching / validation risk: even if we are developing this software for any kind of university, our approach or understanding may be irrelevant and unnecessary. We are planning to discuss this with experienced university managers before starting a new sprint to update and get more accurate requirements if needed. However, we are most likely sure about the approach because of the scope and idea of the project. Therefore, this risk easily gets the last position in the likelihood rank. But if we can't get the universities attention, the impact will be the highest on this list.
10	2	5	Timing risk / requirement changes: While the project is being carried out, every new feature and requirement that is not defined at the beginning of the project and is wanted to be included later threatens the planned completion time of the project. In Agile processes, the changes should be included in the project and the prioritization process should be started with the previously determined demands. If this process is not like this, things will become very complicated and need to be solved. it gets harder.
2	10	6	Timing risks / management: It is the type of risk that prevents or causes the project to be realized or completed within the expected time due to incorrect task and material sharing.
3	9	7	Technical Support Risk: Users may encounter technical issues while using the platform, and inadequate technical support may lead to dissatisfaction and disengagement.
4	8	8	Content Quality Risk: The quality of course materials uploaded to the platform may vary, leading to dissatisfaction among users.
7	6	9	Exceptions: Errors or defects in the code can lead to malfunctions, crashes, or unexpected behavior in the software.
8	7	10	Contractual and legal risks: Include changing needs, market-driven programs, health and safety issues, government regulations, and product warranty issues.

9	11	11	Budget (cost) risk: These are risks that lead to financial problems as a result of unrealistic budget estimates. These risks If this happens, tables change and costs increase. If Infrastructure Costs (server, database services, network infrastructures), Content Production Costs and Legal and License Costs are not calculated well, they may bring additional risks to the budget.
---	----	----	---

7 - PROJECT NEEDS

In this project, process is going to require some hardware, software and support needs.

7.1. Hardware Needs

- Memory: Samsung 970 EVO Plus 1TB NVMe M.2 SSD
- Processor: AMD Ryzen 7 5800X 8-Core 16-Thread AM4 Processor
- **Graphics card:** NVIDIA GeForce RTX 3060 Ti
- **Monitor:** HP 24mh FHD Monitor
- **Input devices:** Logitech MX Master 3 Wireless Mouse, Logitech K780 Multi-Device Wireless Keyboard

7.2. Software Needs

- **Integrated development environment (IDE):** Visual Studio Code
- **Version control system:** GitHub Desktop
- **Collaboration and communication tools:** Slack, Microsoft Teams, or Discord
- **Unit testing frameworks:** Tools like JUnit (Java), pytest (Python), or Jest (JavaScript)
- **Database management systems:** Amazon RDS
- **Documentation tools:** Confluence, Notion, or Google Docs
- **Design tools:** Sketch or Adobe XD

7.3. Support Needs

- Technical support for users
- Update and maintenance support
- Technical support for development team
- Database management
- Tool learning sources

All need's feature may change as the project advances, because new stakeholder requirements may occur, some unexpected delays may happen and our current system may not provide stability and required sufficiency.

In order to avoid imbalances, we are going to consider features/resources/time triangle.

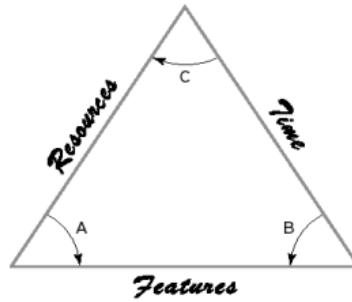


Figure 1: Features/Resources/Time Triangle

8 - MEASUREMENTS

There are different software measurements needed in this project:

- **Features:** Total number of features added in the current development cycle
- **Effort on training:** Total amount of hours to spent on each tool that used in the project.
- **Requirements in backlog:** Total number of new user stories added to the backlog.
- **Amount of API:** Total number of APIs integrated into the project.
- **Optimization effectiveness:** Impact of each optimization on system performance.
- **Time spent on testing:** Total number of hours spent on testing throughout the project.
- **Amount of fixed bugs:** Total bug that are fixed.
- **Counting Days:** Total number of days spent on the project.

9 – TOOLS

Software tools are chosen by using a comparison graph:

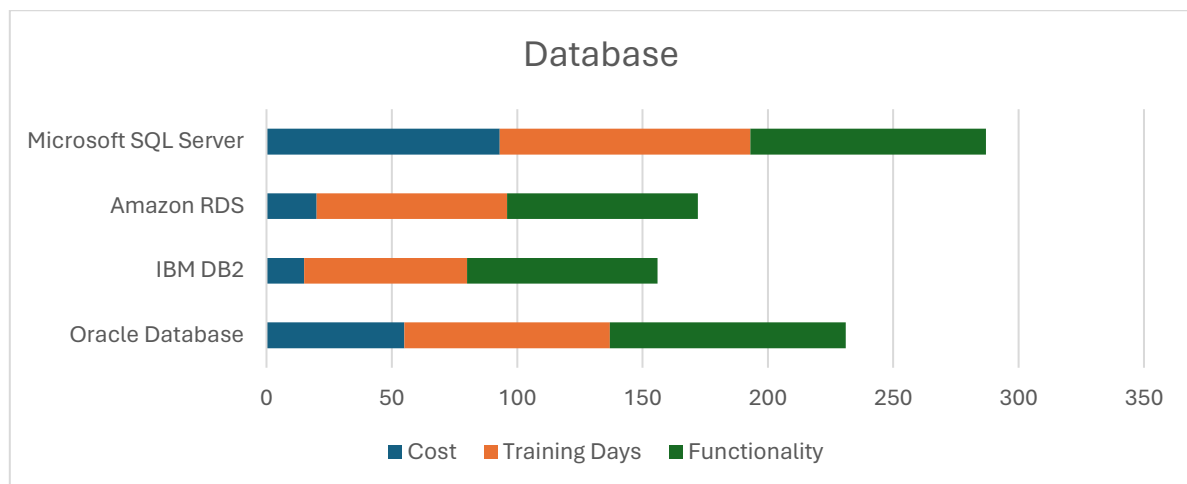
1. Database Tool Cost/Training/Functionality Data

Tool	Oracle Database	IBM DB2	Amazon RDS	Microsoft SQL Server
Cost	\$3,454	\$969	\$6,240	\$1,260
Training Days	14	11	13	17
Functionality	85	40	90	70

2. Normalized Database Tool Cost/Training/Functionality Data

Tool	Oracle Database	IBM DB2	Amazon RDS	Microsoft SQL Server
Cost	55.14	15.58	100	20.16
Training Days	82.35	64.71	76.47	100
Functionality	94.44	44.44	100	77.78

3. Normalized Database Tool Graph



We decided to use Amazon RDS, because its managed service and cost- effectiveness. Its higher functionality rate to lower training time ratio is an important reason why we chose it.

4. Additional Tools

- **IBM DB2:** IBM DB2 is the cloud-native database built to power low-latency transactions and real-time analytics at scale. Built on decades of innovation in data security, scalability and availability, you can use Db2 to keep your applications and analytics protected, highly performant and resilient, anywhere.
- **React:** React is a framework that employs Webpack to automatically compile React, JSX, and ES6 code while handling CSS file prefixes. React is a JavaScript-based UI development library.
- **Oracle Database:** Oracle makes software, called database management systems, to create and manage databases.

Our tool selection list includes Amazon RDS (6240\$), React (0\$) and Visual Studio Code (2500\$). As a total outcome, we need to pay 8740\$ for tool costs per year.

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;

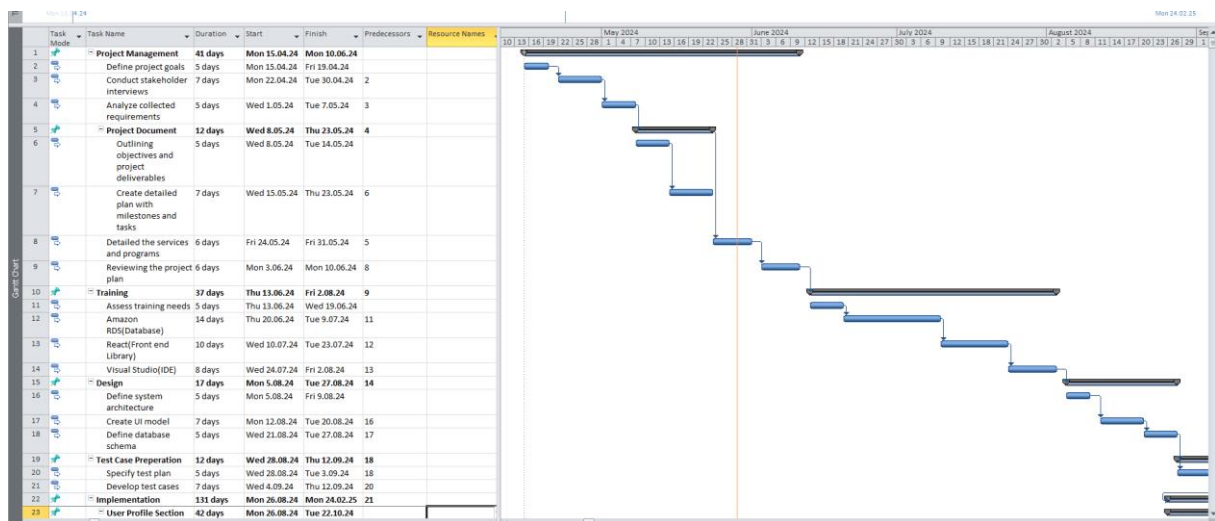


Fig.1 Project-management to implementation.

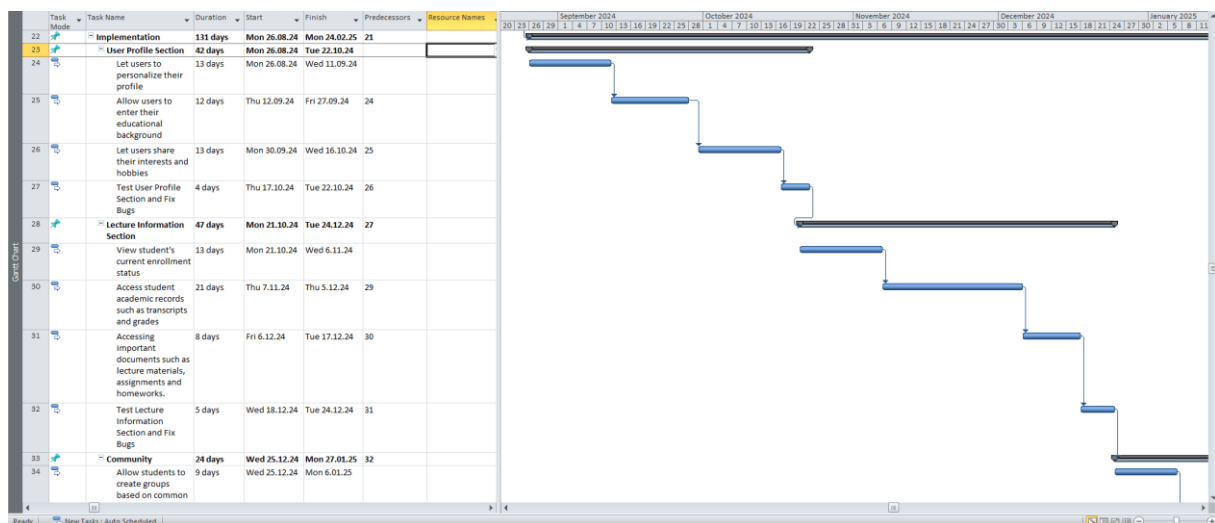


Fig.2 implementation.

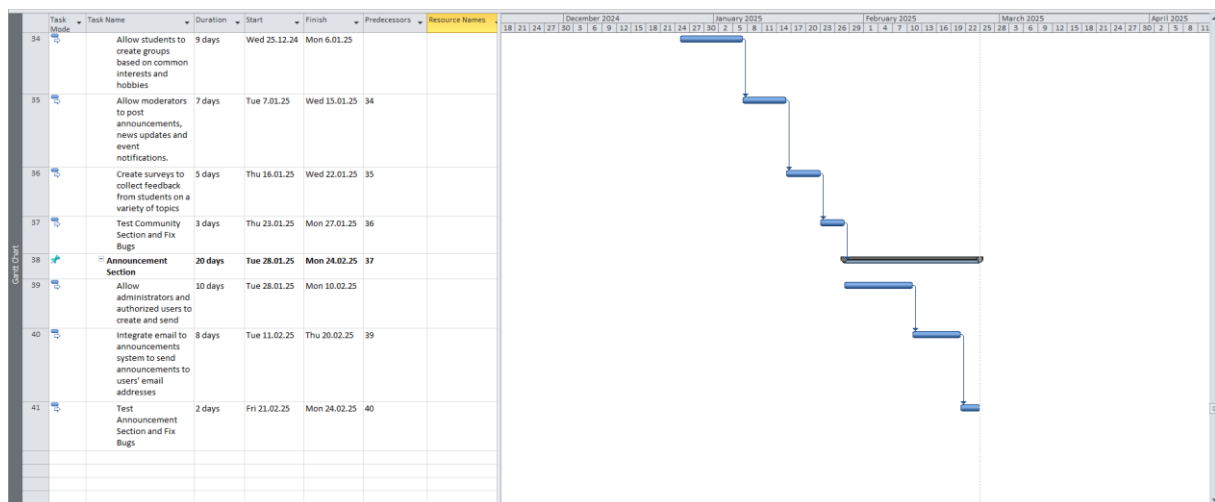


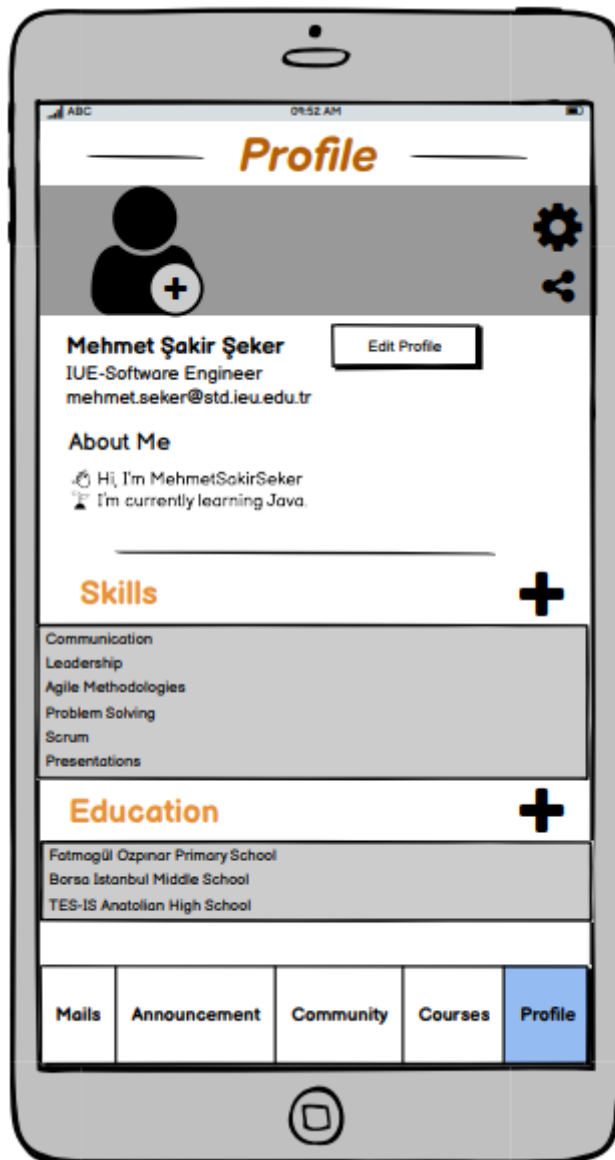
Fig.3 Implementation. (cont.)

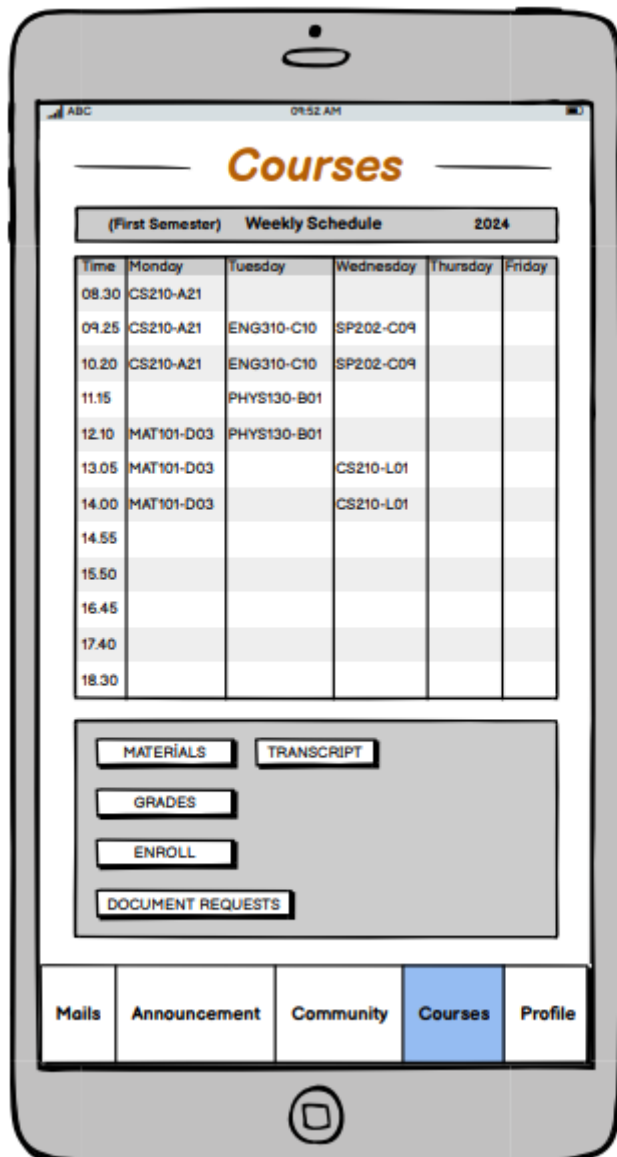
11 - PROJECT PAY-OFFS

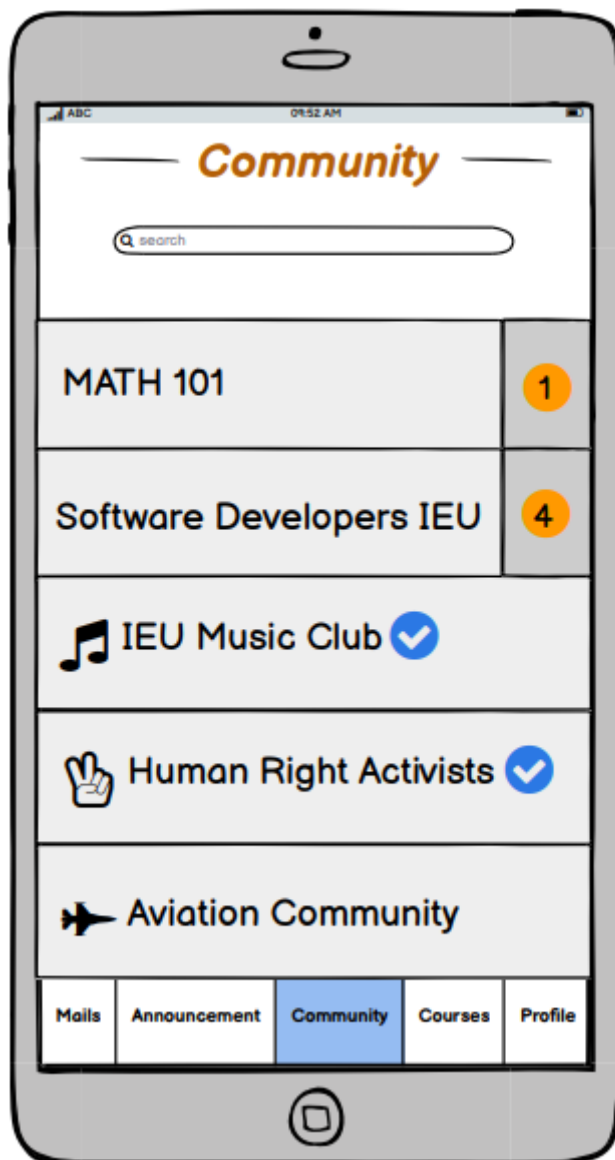
If the project is successfully implemented;

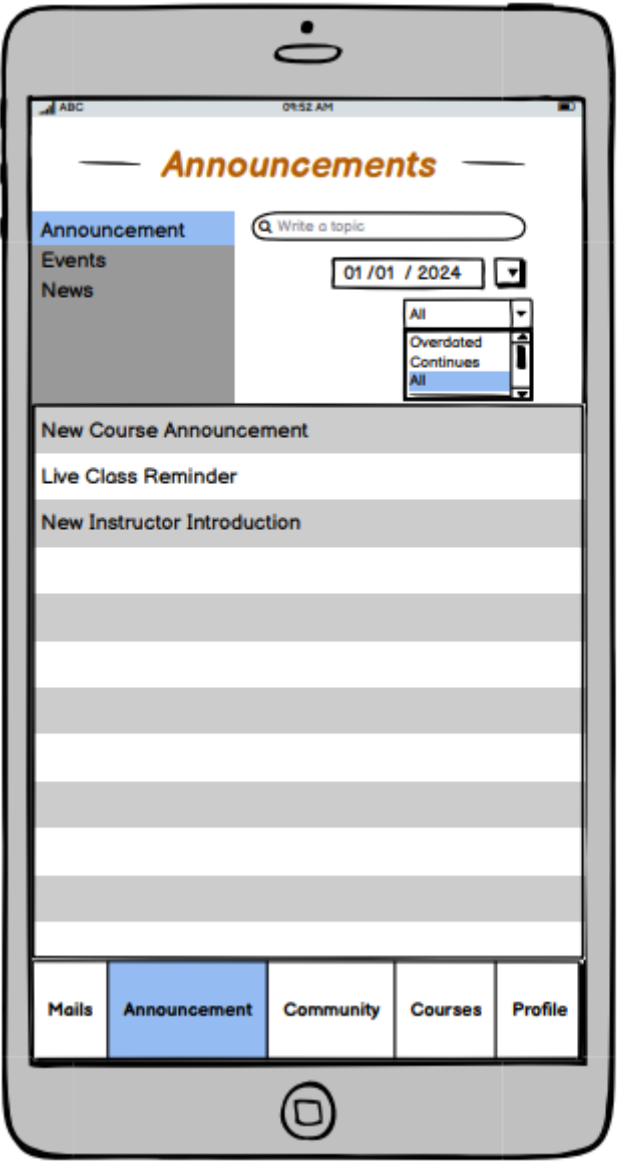
- All universities will be collected in a single application
- University lecturers will have the opportunity to make themselves known on a global scale
- Students will have the opportunity to socialise on a global scale
- Team members are going to obtain priceless professional experience
- These payoffs will maximize the production quality and performance of team members.

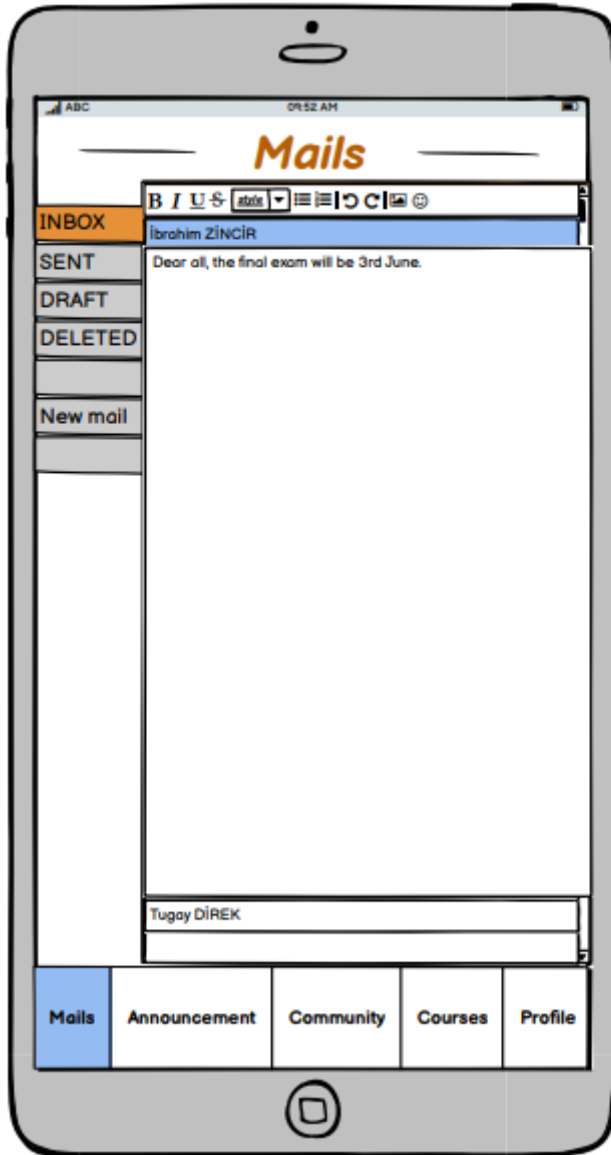
12 - USER INTERFACES











13 – FUTURE PLANS

Our future objectives for this project include increasing the platform's capabilities and developing new features to better support institutions' academic and social demands. We intend to incorporate advanced analytics and reporting tools to provide more detailed insights into student performance and engagement. We also intend to integrate AI-powered individualized learning routes and recommendations to tailor the educational experience to the specific needs of each learner. To encourage increased collaboration, we will integrate real-time collaboration tools and virtual classrooms that allow for synchronous learning and group projects. Expanding language support and accessibility features will make our platform more inclusive and user-friendly for various global audiences. We also intend to form relationships with educational material producers to enhance existing resources and materials. These future

enhancements will strengthen our platform's position as a comprehensive, cutting-edge solution for modern educational institutions.

14 - CONCLUSION

Ultimately, our project aims to create a unifying method in the way universities manage their academic and social activities by providing a unified, user-friendly digital platform. By integrating functions such as course management, communication, community engagement, and user profiling into a single application, we aim to streamline university operations and improve the overall experience for students and faculty. Implementing Scrum as our process model allowed us to maintain flexibility, foster effective team collaboration, and manage risks efficiently. Our focus on designing an intuitive interface, ensuring high performance and security, and encouraging social interaction between users underlines our commitment to providing a comprehensive solution that meets the evolving needs of modern educational institutions. We believe that this platform will not only simplify administrative tasks but also enrich the academic and social environment in universities, ultimately contributing to a more connected and efficient educational experience.