# YEDITEPE UNIVERSITY
# CSE 344 PROJECT
# ANALYSIS REPORT


**CSE STUDENTS:**

Ali Abbasi Dolatabadi
Gamze Nur Erdem
Ece Tipici
Elif Dikmen
Muhammet Ramazan Dolaş

**COURSE INSTRUCTOR:**

Associate Prof. Dr. Mert Özkaya

# Analysis Report

## 1. INTRODUCTION

### 1.1. PURPOSE

In a bustling college environment, students often face barriers when attempting to access academic advice or seek guidance from professors. An appointment system erases these barriers by providing a user-friendly interface through which students can effortlessly schedule appointments with relevant professors.

The purpose of this app is to facilitate an appointment system for students seeking face-to-face meetings with their professors. The app aims to serve college students as a centralized platform for scheduling meetings with professors. It strives to simplify the appointment scheduling process, eliminating the need for extensive email communication and reducing the risk of overlapping appointments. Moreover, it offers students the flexibility to book appointments at their convenience, thereby optimizing their time management strategies and enabling them to balance academic pursuits with personal commitments. The system also aims to integrate the application with universities' student information systems and mobile applications. This integration enables professors and students to manage their appointments through a single system.

In summary, this appointment scheduling app empowers students by streamlining communication with professors, eliminating the hassle of email back and forth, and ensuring a smooth scheduling process. By offering a centralized platform accessible through mobile devices, the app allows students to manage their time effectively and prioritize their academic success.

## 1.2. BACKGROUND

### 1.2.1. Introduction

Domain analysis is an important step in developing well-structured software. It has a huge effect on understanding the research area and has beneficial effects on developing a better system.

The field that we are working on is communication between professors and students in universities. Traditionally, students arrange appointments with professors by initiating email conversations. However, this process can encounter challenges. Professors, overwhelmed by busy schedules, might miss or not respond to emails promptly, leading to unfulfilled appointment requests. Emails can also get lost in spam folders, further complicating communication.

Another issue arises when professors unintentionally double-book appointments or when students forget scheduled meetings. Additionally, there are instances where professors become available at short notice, requiring a system to promptly notify interested students.

Furthermore, urgent situations may arise where professors need to cancel appointments, posing challenges in communicating changes promptly. To address these complexities, an efficient and instant communication system becomes crucial.

### 1.2.2. General Knowledge About the Domain

There are many students who experience communication breakdowns with professors at universities. Therefore, this application addresses one of the most important problems of students by making the communication between professors and students planned. Here are some of the considerations we are putting on:

- User-friendly interfaces for students to view professors' availability and schedule appointments.
- Integration with university systems to access professors' schedules and course information.
- Notifications and reminders to help students and professors stay organized and informed about upcoming appointments.
- Mobile accessibility, allows students to schedule appointments on the go from their smartphones or tablets.
- Customization options for professors to set their availability preferences and appointment durations.

## 1.2.3. Customers and Users

In the context of an appointment scheduling system for college students, the customers and users can be identified as follows:

- Customers:

Universities and Colleges: These institutions are the primary customers as they would typically procure or implement the appointment scheduling system for the benefit of their students and faculty.
Administrators and Academic Support Staff: Administrators and support staff within universities and colleges are also considered customers as they would oversee the implementation and management of the appointment scheduling system.

- Users:

Students: College students are the primary users of the appointment scheduling system. They utilize the system to schedule appointments with professors for academic advising, mentorship, or assistance with coursework.
Professors and Faculty Members: Professors and faculty members are also users of the system, as they manage their availability and appointments with students through the platform.

By understanding the distinct roles of customers and users, the appointment scheduling system can be tailored to meet the needs and preferences of each group, ultimately enhancing the user experience and effectiveness of the system.

## 1.2.4. The Environment

This appointment app is easy to use, understandable, accessible, and user-friendly. It can be downloaded on any mobile device. We aim to provide as customized an application as possible to ensure the most accurate time management and communication of each student and professor. This application will also integrate with universities' student information systems and mobile applications to ensure ease of use.

Upon launching the application, users are greeted with a distinguished interface offering two distinct options: Professors and Yeditepe Students. For a seamless login experience, the application leverages existing usernames and passwords from the OBS system. It is very important for the usability of this application. After securely logging in with their respective usernames and passwords, users gain immediate access to the comprehensive calendar system of professors affiliated with Yeditepe University. In the appointment scheduling interface, the students will first select their faculty and major. Following this selection, all professors corresponding to their chosen faculty and major will be displayed on the screen. After selecting their preferred professor, students will gain access to their weekly appointment schedule button. In addition to viewing the weekly schedule button, the system also indicates

the current availability status button of the professor. If the professor is available instantly the ”currently available” button will be turned green and students may prefer to visit their office without a prior appointment. However, if the professor is not presently available and students wish to discuss any urgent problem with them, they can request an appointment through the system.
After clicking the weekly schedule button, students will be presented with their appointment calendars, displaying the available dates for scheduling.When they select the desired available dates, students can proceed to the “schedule your appointment” part. Additionally, there is a preferred option to provide a reason for the appointment. Moreover, they would edit their appointments on the next page. At the final stage, a summary of the appointment details will be displayed. Once students have reviewed the information and ensured its accuracy, they can proceed to confirm the appointment, and their scheduling process will be completed.

### 1.2.5. Tasks and Procedures Currently Performed

In this domain, numerous tasks are currently undertaken, with the most common ones including:

- Displaying a list of professors in each major.
- Indicating the current availability status of each professor for instant meetings.
- Presenting the available time slots in a calendar for scheduling appointments with a professor.
- Listing upcoming week's appointments for each student user.
- Displaying the week's appointments for each professor, along with their descriptions.

### 1.2.6. Competing Software

Online appointment scheduling software has revolutionized the way various industries manage appointments, streamlining processes and increasing efficiency for both businesses and their customers. This technology offers a convenient and user-friendly way to schedule appointments, optimize time management, and improve customer service.

Online appointment systems are currently being used in many sectors such as health, education, beauty, etc., and provide a lot of convenience to their users. But there is one area that is overlooked and that is universities.

While online appointment systems have permeated various sectors, universities often lack a dedicated platform for students to schedule meetings with their professors outside of class time. This creates a scheduling gap:

Limited Options: Students typically rely on emailing professors directly, a method prone to missed messages and potential delays.

Inefficiencies: Office hours may not accommodate everyone's schedule, and students might struggle to find suitable meeting times.

Lack of Transparency: Students often have limited visibility into professor availability, leading to frustration and wasted time.

This gap in university appointment scheduling highlights the need for a dedicated online system. Here's why such an application would be essential:

Improved Accessibility: Students can easily view professor availability and book appointments outside of office hours, catering to individual needs and schedules.

Increased Efficiency: Professors can manage their appointments more effectively and reduce scheduling conflicts, allowing them to dedicate focused time to student meetings.

Enhanced Communication: Both students and professors benefit from a clear and organized communication channel, fostering a more productive learning environment.

Reduced Wait Times: Online appointment scheduling reduces back-and-forth communication and potential delays in scheduling meetings.

By implementing a university-specific online appointment system, universities can bridge the current gap and create a more efficient, accessible, and student-centered learning environment. This application would empower students to take a more proactive role in managing their academic support and enhance communication with their professors.

## 1.2.7. Similarities Across Domains and Organizations

In today's dynamic university environment, scheduling efficient meetings can be a challenge. Inspired by successful online scheduling across various industries, this application offers a unique solution for streamlined student-teacher interactions. Here is how this university-specific appointment system leverages existing trends in online scheduling and tailors them to the specific needs of student-teacher interactions within academia.

This application mirrors existing systems in key areas:

Booking Convenience: Similar to healthcare appointments, students can view professor availability and book slots online, streamlining scheduling and communication.

Clear Communication: Like beauty appointments, your system fosters clear channels between students and professors, minimizing conflicts and ensuring focused interactions.

Increased Accessibility: Mirroring online tutoring platforms, your system offers flexibility by allowing appointments outside of office hours, catering to individual student needs.

Beyond these similarities, this application offers a unique university solution:

Tailored Features: Focus on student-teacher interactions with features like appointment types for office hours, advising, or project discussions.

Potential Integrations: Explore connecting with university systems for seamless information access and scheduling.

Mobile-First Design: Consider optimizing for mobile use, allowing students to schedule appointments on the go.

By understanding the need for efficient appointment scheduling across domains, this application leverages this trend to create a valuable university-specific solution. It empowers students, fosters communication, and contributes to a more productive learning experience.

# 1.3. MOTIVATION

## 1.3.1. Statement of Problems with the Existing System

The current system for scheduling appointments between students and professors in college environments often relies heavily on email communication. This approach presents several limitations that undermine efficiency and can lead to frustration for both students and professors.

- **Inefficiency:** Back-and-forth emails requesting availability, confirming appointments, and potentially clarifying meeting details can be time-consuming for both sides. This can lead to delays in scheduling appointments and reduce valuable time that could be spent on academics.
- **Missed Appointments:** Reliance on email can lead to missed appointments due to forgetfulness, overflowing inboxes, or lack of clear communication about appointment details. This appointment scheduling system addresses this issue by incorporating an automated reminder feature. Students and professors will receive notifications beforehand (app notification) to remind them of their upcoming appointments.
- **Conflicting Booking:** Professors may face double bookings if they manage their availability through manual methods like personal calendars or notes. This can lead to confusion and inconvenience for both professors and students.
- **Limited Visibility of Student Messages:** Professors often receive a high volume of emails, making it difficult to prioritize or quickly identify student messages requesting appointments. This can lead to delays in student responses and missed opportunities for communication.

### 1.3.2. The New System

To overcome drawbacks in other systems, we've created an innovative system. For effortless operation on both Android and iOS devices, we highly suggest using Flutter.

- At the start of each week, professors are required to set their available hours for the upcoming week. This allows students to easily browse through a clear and user-friendly calendar view to find an appointment slot that suits them.
- If the student is registered into the “request appointment” system they will get a notification when the professor clicks on the available button.
- Also, students can filter the professors by faculty and major to easily find the intended professor.
- Students log in to the appointment management system using their credentials provided by the university. They can access the system through a web portal or a dedicated mobile application. Once logged in, students can view the availability of professors or academic advisors within their department. The system displays a calendar view showing open time slots for appointments.
- Students can specify the reason for the appointment during booking, and appointments are managed by both students and teachers. Professors have the ability to cancel appointments in case of urgent situations. Notifications are sent to remind individuals of appointments before the scheduled meeting.
- The application can be easily integrated with universities' obs systems.

## 1.4. STRUCTURE OF THE DOCUMENT

The analysis report consists of 6 main sections which are

- Introduction
- Functional Requirements
- Non-functional Requirements
- System Models
- Definitions, Acronyms, and Abbreviations
- Glossary & References respectively.

The introduction, which is the first section, contains a description of the document's aim. This section is supported by a background that provides a synopsis of the subject. This part also discusses the rationale by describing the current system's flaws and the suggested system's promise to address them by including new features. In-depth explanations of these supplementary features' advantages for the system are provided.

The encapsulate requirements encapsulate the functional aspects of the system. Use Case diagrams are utilized to visually represent and comprehensively describe the system's features. Moreover, a prioritized list is established to underscore the varying importance levels of different functionalities. Within this section, the use case specifications offer detailed insights into the requirements.

The non-functional requirements are mentioned in the third section. These requirements are supported with Volere templates for the purpose of ease of use. These templates provide in-depth details on the associated needs, including stating their nature, the justification for them, and their importance.

System models are shown in the fourth section. Models for objects and classes are built in this section. This section also includes a description of the user interface. At the end of the document, important keywords for the project and their definitions are explained in the fifth section. Finally, the glossary part expresses what websites are going to be used while doing this project.

## 2. FUNCTIONAL REQUIREMENTS

### 2.1. Description of the System Functionalities

**Login Function:**

Students first navigate to the “Students” button, while professors access their account by selecting the “Professors” button. Then users enter their personal information. They click the "Login" button.

**Faculty and Major Selection Function:**

This feature allows students to choose their faculty and major from the side menu. This information is crucial for the app to function properly, as it tailors the student experience by displaying relevant professors based on their selection.

**Request Appointment Function:**

This feature allows students to see professors' current availability and receive notifications.

**Cancel Appointment Function:**

This feature allows both students and professors to cancel their appointments.

**Reason of Appointment Function:**

This feature allows students to specify the purpose of their appointment.

**Upcoming Appointment Notification:**

This feature sends notifications to the students for upcoming appointments 1 hour in advance and asks the student if they want to cancel the appointment.

**Display of Professors:**

The system shall display department professors on the screen according to the student's department.

**Edit Appointment:**

The system allows the student to edit his/her appointment. Editing an appointment allows the student to cancel an appointment, change the appointment time and edit the appointment message.

**Currently Available Notification:**

This feature is activated by professors when they are instantly available and a notification is sent to all students who have made an appointment request in advance.

**Confirm Appointment Function:**

This feature allows students to verify the details of the appointments such as date,time and purpose.

**Arrange Schedule Calendar Function:**

This feature allows professors to arrange date and time slots.Besides,they are able to reschedule the appointment when needed.

**Instant Appointment Function:**

The instant appointment function allows students to schedule meetings with professors without having to wait for the next available time slot. Students who are interested in scheduling an immediate appointment can request one and register to receive notifications. Later, when the professor is available, they can select the 'currently available' button. Once this button is selected, all students who registered for immediate appointments will receive a notification.

**Display Appointments Function:**

The system shall allow users to display details of their appointment.

**Selecting Professor and Time Function:**

This functionality allows students to schedule appointments with professors. Once a professor is selected, the app will display their available appointment slots based on their pre-configured schedule. Students can then choose a convenient time slot that fits their schedule.

**View Calendar Function:**

This feature allows users to view their upcoming appointments within the app.

## 2.2. Description of the System Users

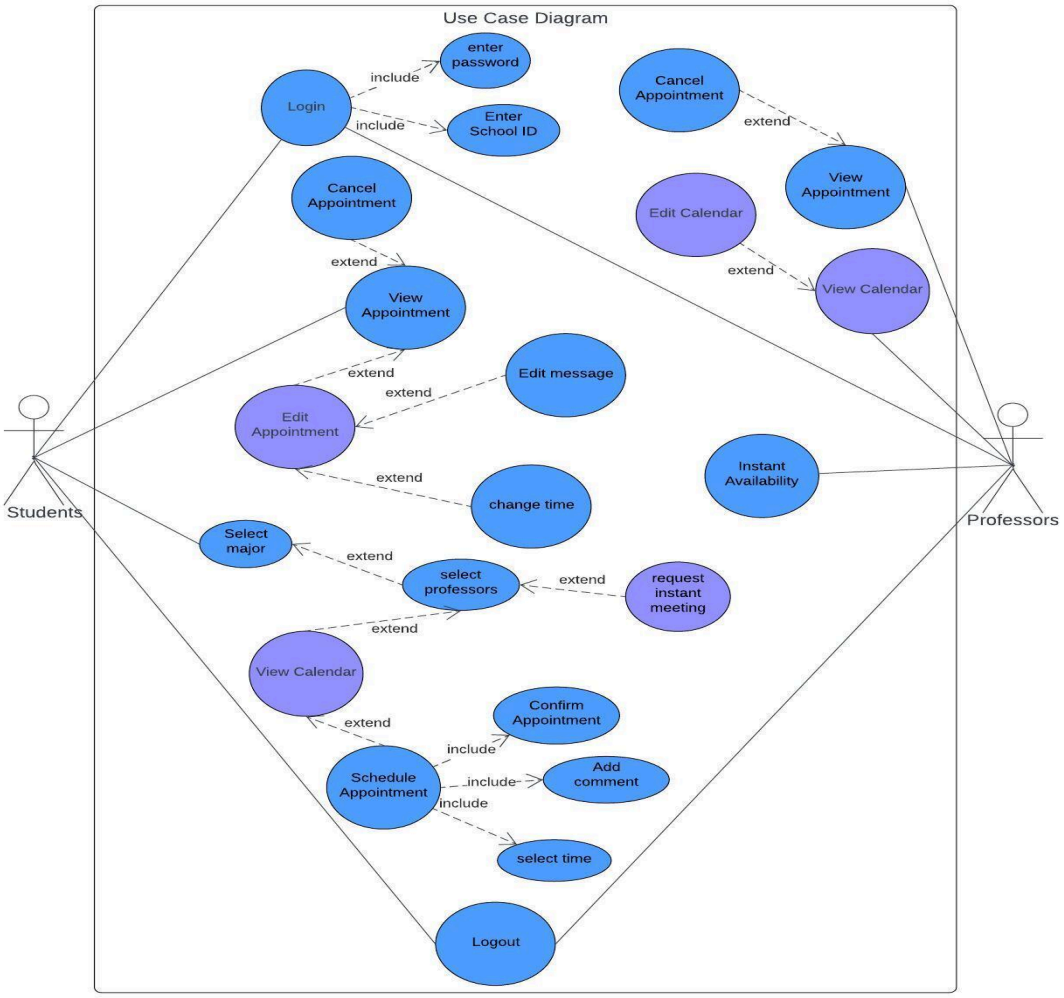
The system users for an appointment app can be divided into two categories: Students and Professors.

Students are the driving force behind appointment requests. Students will utilize the app to schedule appointments with their professors for academic discussions, project guidance, etc.., view professor availability and choose convenient meeting times, and manage their appointments, including cancellation, rescheduling, and receiving reminders. Professors will utilize the app to manage their appointment schedules through the app. Professors will be able to set their available meeting times and define appointment durations to accommodate their schedule, view all student appointment requests in a centralized location and cancel ,or reschedule appointments based on their availability and urgency.

Administrators are in charge of app maintenance and management, including updating the appointment database, guaranteeing data security, and controlling user accounts. Software developers, database administrators, and other IT experts who are in charge of the app's technical components may be considered administrators.

## 2.3. Specific Requirements

### 2.3.1 Use Case Diagram




## 2.3.2. Use Case Priority List

### 2.3.2.1. High Priority

- Log in
- Entering school/staff id
- Entering password
- Schedule Appointment
- Requesting instant appointment
- Viewing professor availability
- Set instant availability
- Edit calendar
- Selecting professor
- View calendar

#### 2.3.2.2. Medium Priority

- Log out
- Selecting faculty/major
- Editing appointment
- Viewing appointment

#### 2.3.2.3. Low Priority

- Support

## 2.3.3.Use Case Specifications

| |
|---|
| **USE CASE ID:** UC1 |
| **USE CASE:** Log In |
| **ACTORS:** Professor, Student |
| **RELATED USE CASES:**<br>Includes: Entering User Name, Entering Password |
| **PRECONDITION:**<br>The user has opened the application. |
| **MAIN FLOW:**<br>1. The user opens the application and clicks to “log in” button.<br>2. The user enters the “User Name Button” and enters “Password Button” .<br>3. The user clicks to “Enter Button” and completes this level. |
| **POST CONDITION:**<br>The user is logged in to the system. |

| |
|---|
| **USE CASE ID:** UC2 |
| **USE CASE:** Log Out |
| **ACTORS:** Students,Professors |
| **RELATED USE CASES:**<br>**Includes:** - |
| **PRECONDITION:**<br>The user has opened the application |
| **MAIN FLOW:**<br>1. The student and professors opens the application<br>3. The user clicks to “Log Out Button” and completes this level. |
| **POST CONDITION:**<br>The end user is logged out to the system. |

| |
|---|
| **USE CASE ID:** UC3 |
| **USE CASE:** Support |
| **ACTORS:** Students, Professors, Admin |
| **RELATED USE CASES:**<br>- |
| **PRECONDITION:**<br>The user should login to the app and click on profile. |
| **MAIN FLOW:**<br>1. The user should login.<br>2. Select the profile button.<br>3. Select the support button. |
| **POST CONDITION:**<br>The user can get support from the admin. |

| |
|---|
| **USE CASE ID:** UC4 |
| **USE CASE:** Entering school/staff id |
| **ACTORS:** Students and Professors |
| **RELATED USE CASES:**<br>**Related entities:** Log in |
| **PRECONDITION:**<br>The user has opened the application and selected "Student" or "Professor"**.** |
| **MAIN FLOW:**<br>1. The end user clicks the "Student" or "Professor" button depending on whether they are a student or professor<br>2. The end user chooses the "School ID" or "Staff ID" field.<br>3. The end user enters his/her school/staff id. |
| **POST CONDITION:**<br>The end user completes the "Entering School/Staff ID" action. |

| |
|---|
| **USE CASE ID:** UC5 |
| **USE CASE:** Entering password |
| **ACTORS:** Students and Professors |
| **RELATED USE CASES:**<br>**Related entities:** Log in |
| **PRECONDITION:**<br>The user has opened the application and selected "Student" or "Professor"**.** |
| **MAIN FLOW:**<br>4. The end user clicks the "Student" or "Professor" button depending on whether they are a student or professor<br>5. The end user chooses the "Password" field.<br>6. The end user enters his/her password. |
| **POST CONDITION:**<br>The end user completes the "Entering Password" action. |

| |
|---|
| **USE CASE ID:** UC6 |
| **USE CASE:** Schedule appointment |
| **ACTORS:** Students |
| **RELATED USE CASES:**<br>Select professor |
| **PRECONDITION:**<br>The user opens the application logs in to the application |
| **MAIN FLOW:**<br>1. The user opens the application and selects the professor with whom they want to make an appointment.<br>2. The user selects the time they want to make an appointment in the calendar of the professor they choose.<br>3. The user enters the reason for the appointment.<br>4. The user clicks the "Schedule Appointment" button and an appointment is created. |
| **POST CONDITION:**<br>The user completes the "scheduling appointment" action. |

| |
|---|
| **USE CASE ID:** UC7 |
| **USE CASE:** Edit Calendar |
| **ACTORS:** Professor |
| **RELATED USE CASES:**<br>**Related entities:** View Calendar |
| **PRECONDITION:**<br>The user has opened the application and select “View Calendar” |
| **MAIN FLOW:**<br>1. The end user opens the application and clicks the “View Calendar” button.<br>2. The end user clicks to the “Change Calendar”<br>3. The end user sets the available hours in the program.<br>4. The end user clicks to the "Save the Changes" button. |
| **POST CONDITION:**<br>The user has changed available or unavailable hours on the calendar. |

| |
|---|
| **USE CASE ID:** UC8 |
| **USE CASE:** Selecting professor |
| **ACTORS:** Student |
| **RELATED USE CASES:**<br>**Related entities:** Selecting faculty, Selecting major |
| **PRECONDITION:**<br>After the student selects their faculty and major, they should then choose the professor with whom they want to schedule an appointment. |
| **MAIN FLOW:**<br>1. The student selects the faculty/major of the professor.<br>2. The student selects the professor |
| **POST CONDITION:**<br>The student completes the “Selecting professor” action. |

| |
|---|
| **USE CASE ID:** UC9 |
| **USE CASE:** Edit Appointment |
| **ACTORS:** Student |
| **RELATED USE CASES:**<br>**Related entities:** Cancel appointment |
| **PRECONDITION:**<br>The user logged in to the application and created an appointment |
| **MAIN FLOW:**<br>1. The user opens the application and selects the “My Appointments” tab.<br>2. The user clicks on the “Edit” button of the appointment he/she wants to edit.<br>3. The user can optionally change the appointment time and appointment message from the opened professor's calendar.<br>4. The user clicks on the "Save Changes" button. |
| **POST CONDITION:**<br>The user completes the "Editing appointment" action. |

| |
|---|
| **USE CASE ID:** UC10 |
| **USE CASE:** View Appointment |
| **ACTORS:** Student, Professor |
| **RELATED USE CASES:**<br>**Related entities:** Schedule appointment, Selecting Professor, |
| **PRECONDITION:**<br>The student first selects a professor then schedules an appointment. After the appointment is scheduled it can be viewed by students and professors using “View Appointment” action. |
| **MAIN FLOW:**<br>1. The end user should navigate to the 'All Appointments' section.<br>2. From there, to view the appointment details, they should click on 'Details'. |
| **POST CONDITION:**<br>The student completes the “View Appointment” action. |

| |
|---|
| **USE CASE ID:** UC11 |
| **USE CASE:** Select Major/Faculty |
| **ACTORS:** Students |
| **RELATED USE CASES:**<br>**Related entities:** Login, Entering school/staff id, Entering password |
| **PRECONDITION:**<br>To be able to select a major and faculty you first need to login. |
| **MAIN FLOW:**<br>1. The student should login.<br>2. Select Major/Faculty |
| **POST CONDITION:**<br>The student completes the “Select Major/Faculty” action. |

| |
|---|
| **USE CASE ID:** UC12 |
| **USE CASE:** Viewing Calendar |
| **ACTORS:** Students, Professors |
| **RELATED USE CASES:**<br>**Related entities:** Login, Selecting Professors |
| **PRECONDITION:**<br>The student should login and later on should select a professor.<br>The professor should only login. |
| **MAIN FLOW:**<br>1. The user should login.<br>2. If the user is a student, they should first select a professor and then click on the 'Weekly Schedule' button to view the calendar.<br>3. If the user is a professor, they should click on the “View Calendar” button. |
| **POST CONDITION:**<br>The student completes the “Viewing Calendar” action. |

| |
|---|
| **USE CASE ID:** UC13 |
| **USE CASE:** Viewing professor availability |
| **ACTORS:** Students |
| **RELATED USE CASES:**<br>**Related entities:** Login, Select professor, View calendar |
| **PRECONDITION:**<br>The student should login and later on should select a professor. |
| **MAIN FLOW:**<br>1. The student should login.<br>2. The student should select the professor they want to schedule an appointment with.<br>3. The student should check the professor's calendar to see the available appointment times. |
| **POST CONDITION:**<br>The student completes the “Viewing professor availability” action. |

| |
|---|
| **USE CASE ID:** UC14 |
| **USE CASE:** Request instant appointment |
| **ACTORS:** Students |
| **RELATED USE CASES:**<br>**Related entities:** Viewing professor availability, setting instant availability |
| **PRECONDITION:**<br>The user has opened the application and selected a professor. |
| **MAIN FLOW:**<br>1. The user logs in to the application and selects a professor.<br>2. The user clicks to the "Request Instant Meeting" button. |
| **POST CONDITION:**<br>The user completes the "Requesting Instant Meeting" action and when the professor he/she chooses clicks on the "Currently Available" button, the student receives a notification that the professor is currently available. |

| |
|---|
| **USE CASE ID:** UC15 |
| **USE CASE:** Set instant availability |
| **ACTORS:** Professor |
| **RELATED USE CASES:**<br>**Related entities:** Login |
| **PRECONDITION:**<br>To set instant availability, the user must be logged in to the application.. |
| **MAIN FLOW:**<br>1. The user logs in to the application.<br>2. The user clicks on the “Currently Available" button. |
| **POST CONDITION:**<br>The professor has set instant availability then the system sends notification about professor’s instant availability to the students who requested instant appointment. |

# 3.Non-functional Requirements

**3.1.1**

The system shall allow the user to login in 30 seconds.

**3.1.2**

The system shall be a phone application.

**3.1.3**

The system shall be available 7/24.

**3.1.4**

The system shall back up data.

**3.1.5**

The system shall have an interface.

**3.1.6**

The system shall be compatible with a wide range of devices and platforms.

**3.1.7**

The system shall be able to host 40.000 users at the same time.

**3.1.8**

The system shall record previous activities for 3 years.

## 3.1. Volere Template Requirement

| **Requirement ID: 1** | **Requirement Type: NFR (Performance)** | **Event/Use case #** |
|---|---|---|
| **Description:** The system shall allow the user to login in 30 seconds. | | |
| **Rationale:** The system should provide users to be able to access their account quickly. Extended waiting times during account login may cause the customer to abandon the login. This problem is solved if the cooldown is limited to 30 seconds. | | |
| **Fit Criteria:** If the waiting time exceeds 30 seconds, the user must be re-entered with a message informing the user. | | |
| **Priority:** High | | |

| **Requirement ID: 2** | **Requirement Type: NFR (Implementation)** | **Event/Use case #** |
|---|---|---|
| **Description:** The system shall be a phone application. | | |
| **Rationale:** The system shall be online and accessible as a global service | | |
| **Fit Criteria:** Accessible as an online service | | |
| **Priority:** High | | |

| **Requirement ID: 3** | **Requirement Type: NFR (Performance)** | **Event/Use case #** |
|---|---|---|
| **Description:** The system shall be available 7/24 | | |
| **Rationale:** This is very important for users to access the system. Users should be able to access the system at any time of the day. By providing this performance, the system can improve the user experience and increase the value of the service provided. | | |
| **Fit Criteria:** To ensure that users can log in and out of the system at any time. This can help improve service performance over time. | | |
| **Priority:** High | | |

| **Requirement ID: 4** | **Requirement Type: NFR (Archiving)** | **Event/Use case #** |
|---|---|---|
| **Description:** The system shall back up data. | | |
| **Rationale:** The solution needs to be storage all necessary information into a database | | |
| **Fit Criteria:** Information can be accessible in a database | | |
| **Priority:** High | | |

| **Requirement ID: 5** | **Requirement Type: NFR** | **Event/Use case #** |
|---|---|---|
| **Description:**The system shall have an interface. | | |
| **Rationale:**The solution needs to have an intuitive and user-friendly interface for users to interact with | | |
| **Fit Criteria:**Users can easily interact with the system through the interface | | |
| **Priority:**High | | |

| **Requirement ID: 6** | **Requirement Type : NFR (Compatibility)** | **Event/Use case #** |
|---|---|---|
| **Description:**The system shall be compatible with a wide range of devices and platforms. | | |
| **Rationale:** Users may access the system using mobile phones, and may use different operating systems, such as iOS, or Android. By ensuring compatibility, the system can increase its reach and accessibility, and provide a consistent user experience regardless of the device or platform being used. | | |
| **Fit Criteria:**The system running smoothly and efficiently on all devices and platforms, without significant lag or delays. | | |
| **Priority:**High | | |

| **Requirement ID: 7** | **Requirement Type: NFR (Performance)** | **Event/Use case #** |
|---|---|---|
| **Description:**The system shall be able to host 40.000 users at the same time. | | |
| **Rationale:** The system should allow 40.000 users to login to the application at the same time.In this way, many users can log into the application and there will be no loss of users. | | |
| **Fit Criteria:**If the number of simultaneous users exceeds 100.000, a message should be sent to both the admin and the user stating that the user limit has been exceeded. | | |
| **Priority:**Low | | |

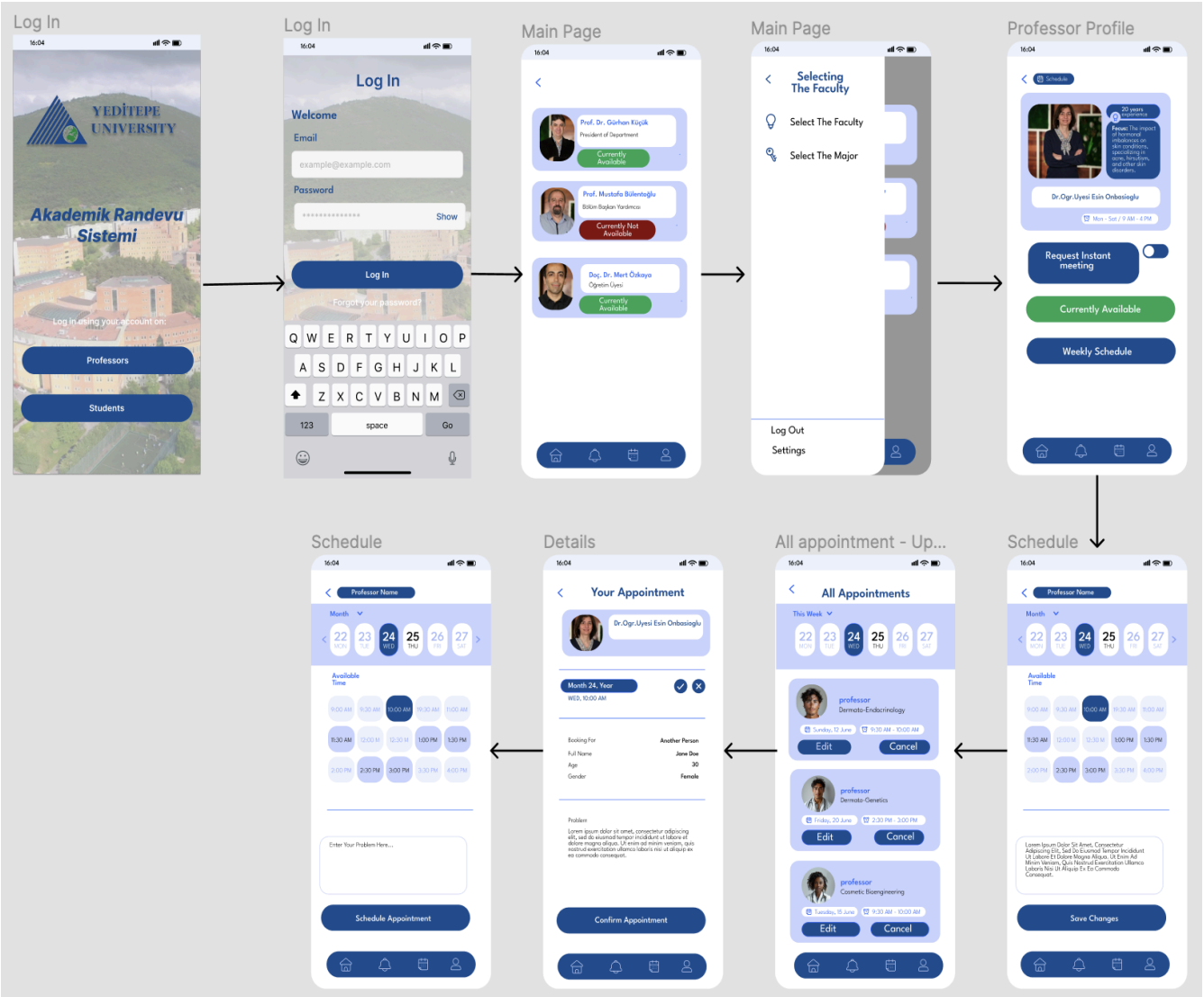
| Requirement ID: 8 | Requirement Type: NFR (Archiving) | Event/Use case # |
|---|---|---|
| **Description:**System shall record previous activities for 3 years. | | |
| **Rationale:**Visibility of historical data is essential for tracking progress and development. On top of that, this can help ensure compliance, aid in audit and traceability efforts, and support forensic investigations in case of a security breach | | |
| **Fit Criteria:**Accessibility and documentation of past records | | |
| **Priority:**Low | | |

# 4. SYSTEM MODELS

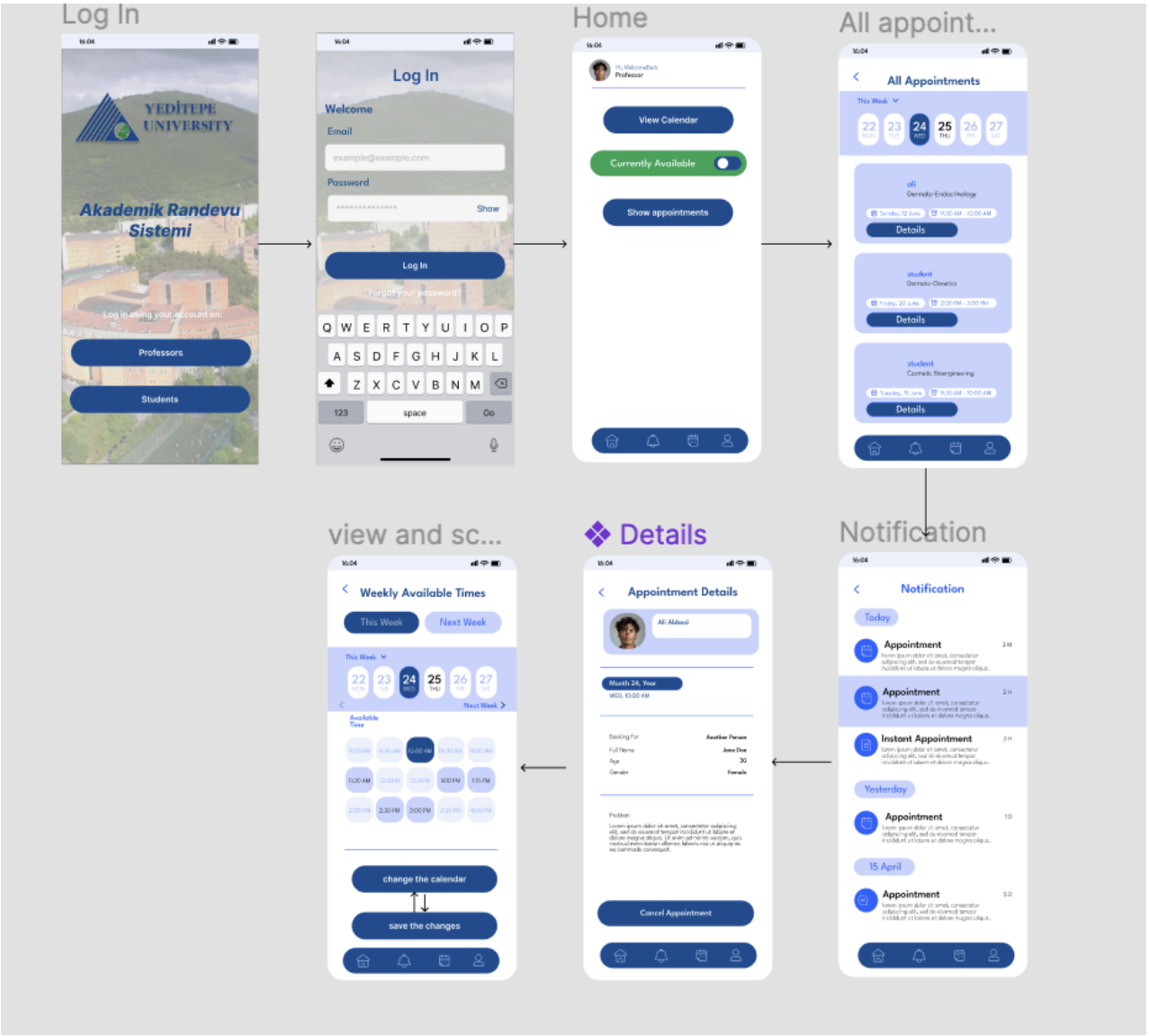
## 4.1. Object And Class Models

## 4.2. User interface - navigational paths and screen mock-ups

## Student's screen mock-up:



**Professor's screen mock-up:**




## 5. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

**Domain:** A domain is a name that identifies a website or a network of computers on the internet. It is essentially the address that people use to
access a website.

**Use Case Diagram:** A description of how users will perform tasks on your website.

**NFR:** Non Functional Requirement

**FR:**Functional requirements

**Instant meeting:** An instant meeting requires a spontaneous gathering or discussion that occurs without prior scheduling.

## 6. GLOSSARY & REFERENCES


Volere Requirement Specification Template: https://www.reqview.com/blog/2019-02-27-news-volere-requirements-specification-template/

How to Write a Proper Mobile App Requirements Document in 5 Steps – Nix (Online): https://nix-united.com/blog/how-to-write-a-proper-mobile-app-requirements-document-in-5-steps/

Use Case Specification Guideline – Best Tips & Guidance for 2023 – Business Analyst Mentor: https://businessanalystmentor.com/use-case-specification-guidelines/

Mozilla Developer MDN. Using a Database with Mongoose documents. Retrieved March 28, 2023, from / https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose/

Visual Paradigm International. (n.d.). Free Class Diagram Tool. Retrieved March 28, 2023, from/https://online.visual-paradigm.com/diagrams/solutions/free-class-diagram-tool/

Figma:https://www.figma.com/file/laVnFHV7N0hzAtHUEARFRz/CSE344-Project?type=design&node-id=0-1&mode=design&t=KFbRripkCa9bGKvB-0

Draw.io:https://app.diagrams.net/#G1pHcvFi9DQklCMOT3tRsra-x0RBE6hNbj%23%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

Lucidchart.app:https://lucid.app/lucidchart/16d38b0a-2ae0-459c-95d7-afb6a67d8c1e/edit?invitationId=inv_e1f043f9-ed9d-4548-8bfc-072804dc571c&page=0_0#