



Bilkent University

Department of Computer Engineering

---

# Campuswide Connection System

*OnlyBilkent (team12 - 01)*

## Analysis Report

Anıl Altuncu - 21901880  
Zehra İyigün - 22002913  
İpek Sönmez - 22103939  
Bartu Albayrak - 22101640  
Gizem Gökçe Işık - 21803541

Instructor: Eray Tüzün  
Teaching Assistant(s): Yahya Elnouby, Selen Uysal

Final 1

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object-Oriented Programming Project course CS319.

# Contents

1 Introduction.....	3
2 Current System.....	3
3 Proposed System.....	4
3.1 Overview.....	4
3.2 Actors.....	4
3.3 Non-functional Requirements.....	4
3.3.1 Performance.....	4
3.3.2 Safety.....	5
3.3.3 Reliability.....	5
3.3.4 Usability.....	5
3.3.5 Maintainability.....	6
3.3.6 Compatibility.....	6
3.4 Pseudo Requirements.....	6
3.5 System Models.....	6
3.5.1 Use-Case Model.....	7
3.5.2 Class Diagram.....	15
3.5.3. State Diagram.....	17
3.5.3.1 State Diagram of Post Availability Status.....	17
3.5.3.2 State Diagram of Complaint Report Status.....	17
3.5.4 Activity Diagram.....	19
3.5.4.1 Activity Diagram of User Actions.....	19
3.5.4.2 Activity Diagram of Board Account Request.....	20
3.5.4.3 Activity Diagram of Post Creation.....	21
3.5.5 Sequence Diagram.....	22
3.5.5.1 Sequence Diagram for Registration and Login.....	22
3.5.5.2 Sequence Diagram for Requesting Board Account.....	23
3.5.5.3 Sequence Diagram for Making a Post and Messaging.....	24
3.5.6 User Interface.....	25
3.6 Used Tech.....	32
4 References.....	32

# Analysis Report

*OnlyBilkent*

## 1 Introduction

OnlyBilkent is designed to meet the diverse needs of the Bilkent University community, providing a solution for four key posts: second-hand sales, lost and found items, borrowing, and donations. Additionally OnlyBilkent provides a medium for posting and messaging as well. As a platform created exclusively for the Bilkenters, we have incorporated these services into a single application. Whether you are a student tackling your academic journey, a board member of a university club organizing events, or a graduate, OnlyBilkent makes you connect with others on campus. Our commitment is to empower students, university club boards, and graduates with a single, user-friendly platform where they can effortlessly buy items they have been searching for, sell their course materials, reunite with lost belongings, donate their unused items and simplify borrowing processes. While other systems may exist separately for each of the mentioned purposes, OnlyBilkent stands apart by simplifying and enriching the lives of university members with accessibility. Therefore, we are simply dedicated to the students, university club boards, and graduates of Bilkent University with a vision to foster a more connected and harmonious campus experience.

## 2 Current System

Bilkent University students have created numerous ways to get support from each other, whether for selling their second-hand materials or dormitory belongings, finding people to carpool with or just getting valuable advice about school life. We have observed some of these ways are as the following:

- **bilkent\_itiraf Instagram:** An "bilkent\_itiraf" Instagram account serves as a central platform for Bilkent University students.

- **Advantages:**

- It's exclusive to Bilkent students and promotes a sense of community.
- It serves as a platform for sharing information about campus events, student organizations, academic resources etc.
- It allows students to connect with their peers, share their concerns, and provide support to one another potentially leading to collaborative opportunities.
- It can be a source of valuable advice, whether it's related to academic matters, choosing courses, finding study spots etc.

- **Disadvantages:**
  - Instagram posts are transient, meaning they can easily get buried in a user's feed as new content is posted. This can make it hard to find specific information or discussions.
  - Instagram's search functionality is limited and not suitable for finding specific content or topics discussed in the past.
  - Lack of organization of posts by topic makes it difficult for students to find information or discussions relevant to their needs.
- **External Applications:** Bilkent students also use external apps like "sahibinden", "gardrops" and "dolap" for various purposes.
- **Wider User Base:** These platforms have a broader user base.
- **Reliability Issues:** The larger user base can make connecting with the intended audience harder. Buyer and seller being far away from each other might be considered as a problem as well. Moreover, problematic bargaining can lead to a breakdown of the negotiation, increased conflict and inefficiency in the negotiation process. The aforementioned cases potentially cause reliability problems when selling educational materials.

### 3 Proposed System

#### 3.1 Overview

Our project aims to merge the support mechanisms currently utilized by Bilkent University students into one web application. This application will be exclusively for Bilkent University student network and incorporate advanced search and filtering functions for enhanced usability. Only Bilkent Users can access personalized profiles upon registration with their Bilkent University email addresses and successful email verification. The web application will categorize posts into various functionalities, including selling, buying, lending, borrowing, donating course materials, lost and found, changing dormitories, carpooling, animal adoption, seeking course advice, and requesting private tutoring services etc. Bilkent University Students and Graduates can select their preferred category for posting, facilitating interaction between users looking to buy products, sell products, arrange carpooling, and more. Additionally, we will introduce Board Members, who will be representatives of student clubs, will have profiles dedicated to making announcements about club events. This simplifies the process of disseminating information about club events to the Bilkent University student network.

#### 3.2 Actors

As a result of our requirement analysis[1], we identified four types of actors:

- **OnlyBilkent User:** Base actor type for all other actor types. If a user is not assigned to a specific actor type on sign up or later on, the user will be evaluated as an OnlyBilkent User. If a user only has the OnlyBilkent User actor type, the user has limited access to the system. For example, an OnlyBilkent User can participate in authentication.
- **Student:** OnlyBilkent User becomes Student upon the registration process. This happens when a Bilkent University student logs into OnlyBilkent with his/her university email. After signing in, Students can customize their profile, add posts to categories buy, sell, borrow, lend, donate etc. They can also search for posts in the aforementioned categories. They can directly message the poster if they are interested in a post. They will be able to receive notifications about new posts in their selected categories and they will be able to receive notifications about Announcements from Board Members as well.
- **Graduate:** OnlyBilkent User becomes Graduate upon the registration process. This happens when a Bilkent University graduate logs into OnlyBilkent with his/her graduate email. After signing in, Graduates can customize their profile, add posts to categories sell, lend and donate. They will be able to receive notifications about their posts. They will also be able to receive notifications about Announcements if the Board Member is selected for that Announcement to be shared with Graduates.
- **Board Member:** Representative of a Bilkent University student club. This role can only be given by the Admin. To be a Board Member, an OnlyBilkent User from a university club has to make a request. After the evaluation of the request by the Admins, they can become Board Members. Board Members cannot post on the categories or message other OnlyBilkent Users. They can add, remove and edit their announcements about their club events.
- **Admin:** This is a role for the maintainers of OnlyBilkent application. There will be several Admins for the application to work smoothly. An Admin can create other Admin accounts. Admin can view posts and remove posts upon reports. Admin can also create and delete categories for the posts upon the demand and feedback from OnlyBilkent Users. Admin can add, remove and ban users upon reports from the users as well. Additionally Admin manages requests for Board Member accounts.

### 3.3 Non-functional Requirements

#### 3.3.1 Performance

Throughput: Ensure that OnlyBilkent is designed to accommodate the high number of simultaneous users, particularly the over 13,000 students at Bilkent University and numerous graduates. This includes optimizing database queries to handle increased traffic efficiently.

- The email will be sent to the user within a minute after he/she is signed up.
- The login and logout process will take less than 5 seconds.
- Navigation among different pages in the app will take less than 5 seconds.
- Posting process will take less than 10 seconds.

### **3.3.2 Safety**

Authentication: Authentication system where OnlyBilkent users need to log in with their own university e-mail addresses to ensure that they are from Bilkent University.

- This means that each user will have only one account associated with their university e-mail address.
- AES 128 Encryption will be used to encrypt the passwords of the users. After encryption the password will be stored in the database.
- Admins will control the access authorization of the users.
- To verify the file type during the posting process, only JPEG and PNG files will be allowed for upload. For every JPEG and PNG file the maximum file size will be 5mb and overall upload size in one time will be 25mb.

### **3.3.3 Reliability**

Users often seek to search for specific posts on OnlyBilkent, whether for educational purposes or to stay updated on campus events. Therefore, minimizing the data loss caused by system crashes is reasonable.

- Database backups will be performed every 12 hours. Using redundant database systems with replication to ensure data remains available even if one database server fails will minimize the data loss by 90%.
- In MongoDB data read and write processes are also quicker than MySQL since all information is stored in a single document.
- The uptime rate will be above 95% for OnlyBilkent.

### **3.3.4 Usability**

Our aim is to ease and fasten communication among the student network of Bilkent University. Therefore, the system should be easy to use and non-complicated.

- Consistent design elements such as consistent symbols for post categories throughout every page of the application will be maintained. Straightforward and simple left side menu will display post categories, notifications and announcements.
- Posting, searching for a post, making announcement processes will be done with 2 clicks.

- Looking at notifications and announcements will be within a single click.
- Unread notifications and announcements will appear in bold in the list view.
- Sorting posts, notifications and announcements regarding date will be available within 2 clicks.
- Easy on the eyes colors will be used in the background such as Light Lavender (230, 230, 250), Light Peach (255, 218, 185), Light pink (255, 182, 193) etc.
- A dark mode option will be provided as well.

### **3.3.5 Maintainability**

Given the possibility of future needs, such as incorporating new features, it is significant to design our application with maintainability in mind.

- OnlyBilkent components will be organized into separate modules such as post module, buyer seller module, lender borrower module, announcement module etc.

### **3.3.6 Compatibility**

We will employ a responsive design approach to ensure that OnlyBilkent functions well on various screen sizes such as 1920x1080, 1366x768, 1280x720 etc.

## **3.4 Pseudo Requirements**

1. Implementation Requirement:
  - a. The backend of our system should be implemented with an Object Oriented Programming Language.
2. The implementation of the project should be in the form of a web-based application.
3. The project's source code should be trackable on GitHub using Git.

## **3.5 System Models**

### 3.5.1 Use-Case Model

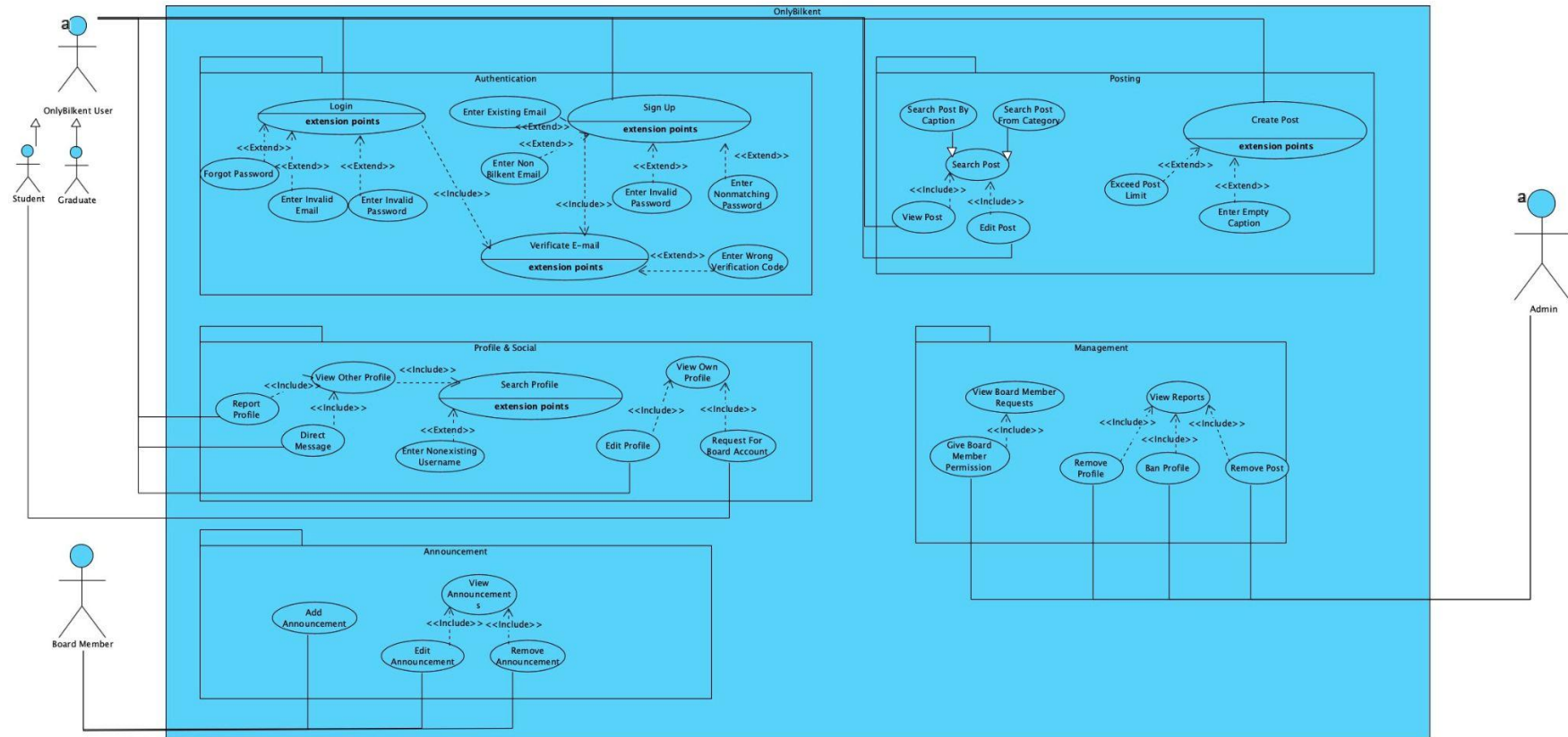


Fig. 1: UML Use Case Diagram of OnlyBilkent Application



## **Authentication Package**

**Use case name:** Login

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. The user decides to log in and clicks the "Login" button.
2. The user inputs their Bilkent University email and password into the mail and password boxes.
3. If the user enters his/her mail incorrectly,
  - a. "Invalid Email" error message is displayed.
4. Else, if the user enters his/her password incorrectly,
  - a. "Invalid Password" error message is displayed.
5. Else, the user successfully logs in.

**Entry condition:** Opening the app using a web browser.

**Exit condition:** User logs in or login process fails.

**Use case name:** Sign up

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. The user decides to sign up and clicks the "Sign Up" button.
2. If the user's email is not unique,
  - a. "Existing Email" error message is displayed.
3. Else if the user's email is not a Bilkent email,
  - a. "Non-Bilkent Email" error message is displayed.
4. Else if the two passwords entered by the user do not match,
  - a. "Non-Matching Passwords" error message is displayed.
5. Else if the user enters an invalid password,
  - a. "Invalid Password" error message is displayed.
6. Else, the user successfully signs up.
  - a. A verification mail is sent to the entered email address.

**Entry condition:** Clicking the sign up button on the first page, which is the login page.

**Exit condition:** Clicking the cancel button or completing the sign-up process.

**Use case name:** Verificate Email

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. The user signs up and opens his/her Bilkent email.
2. Enters verification code from his/her mail to the app.
3. If the entered verification code matches with the sent verification code,
  - a. User successfully signs up.
4. Else if the entered verification code does not match with the sent verification code,

- a. A new verification code is sent to the user.

**Entry condition:** User successfully signs up.

**Exit condition:** Entering the correct verification code from his/her email to the app.

### **Profile and Social Package**

**Use case name:** Search Profile

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. The user decides to search for a user by their username by writing the username into the search bar.
2. The user clicks the search button after writing the username to the search bar.
3. If a user with the entered username does not exist,
  - a. "Nonexistent Username" error message is displayed.
4. Else if users with the entered username exist,
  - a. The user clicks the profile he/she wants to view.

**Entry conditions:** Searching names through the search bar.

**Exit conditions:** Entering nonexistent username or clicking on the profile that the user wants to view.

**Use case name:** View Others Profile

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. User searches a username using search bar.
2. If the searched user exists,
  - a. User clicks on the desired profile,
    - i. The user can report the profile that he/she views.
    - ii. The user can direct message to the profile that he/she views.

**Entry conditions:** User searches a username using search bar and clicks on the desired profile.

**Exit conditions:** Clicking on the wanted user profile or another page.

**Use case name:** No profile with searched Username

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. User search a user by their username by writing the username into the search bar.
2. The user clicks the search button.
3. The system triggers an error message if no user with the provided username is found in the search results.
  - a. The error message informs the user that no user with the specified username was found.

4. The user has the option to modify the search criteria or take other actions.

**Entry conditions:** Searching usernames through the search bar.

**Exit conditions:** Navigate to the wanted profile or another page of OnlyBilkent.

**Use case name:** Direct message

**Participating actors:** OnlyBilkent users

**Flow of events:**

1. User decides to message a specific user and clicks on his/her profile page.
2. The user clicks on the "Message" button and sends a message to him/her directly.

**Entry conditions:** Users should navigate to the profile page of the specific user and click on the "Message" button.

**Exit conditions:** Navigate to the other page of the OnlyBilkent.

**Use case name:** View Own Profile

**Participating actors:** OnlyBilkent User

**Flow of events:**

1. User clicks on his/her profile.
2. The user clicks on the edit button.
  - a. The user makes changes to his/her profile.
    - i. User clicks on the save button.
  - b. The user does not make changes to his/her profile.
    - i. User clicks on the cancel button.
3. The user clicks on the request board member account button.

**Entry conditions:** User clicks on his/her profile.

**Exit conditions:** User clicks on another page.

**Use case name:** Edit Profile

**Participating actors:** OnlyBilkent users, Board

**Flow of events:**

1. User decides to edit the profile and enters his/her profile.
2. The user clicks the "Edit" button.
3. The user makes the necessary updates and clicks the "Save" button to save the changes.

**Entry conditions:** The user navigates to his/her profile page and clicks the "Edit" button.

**Exit conditions:** The user clicks on the "Save" button to save the changes to the profile or decides to navigate to another page.

## **Posting Package**

**Use case name:** View Post

**Participating actors:** OnlyBilkent users, Admin

**Flow of events:**

1. User decides to view posts and clicks to the dashboard.
2. From the dashboard page, the user clicks the specific context from which they wish to view posts.

**Entry conditions:** The user clicks on the specific content to view the corresponding post.

**Exit conditions:** The user navigates to the dashboard.

**Use case name:** Search Post

**Participating actors:** OnlyBilkent users, Admin

**Flow of events:**

1. User decides to search posts and writes the keyword to the search bar what he/she wants to search.
2. By the matched keywords, the user can see posts.

**Entry conditions:** The user enters a specific keyword into the search bar.

**Exit conditions:** The user navigates to the dashboard.

**Use case name:** No post found with searched Keyword

**Participating actors:** OnlyBilkent Users

**Flow of events:**

1. User searches for posts containing a specific keyword by entering it into the search bar.
2. The user clicks the search button, the user receives search results.
3. The system triggers an error message if no posts containing the provided keyword are found in the search results.
  - a. The error message informs the user that no posts with the specified keyword were found.
4. The user can modify the search criteria or take other actions.

**Entry conditions:** The user enters a keyword in the search bar and clicks the search button.

**Exit conditions:**

1. The user receives an error message indicating no posts with the specified keyword were found.
2. The user may modify the search criteria, perform a new search, or continue using the platform.

**Use case name:** Create Post

**Participating actors:** OnlyBilkent users

**Flow of events:**

1. The user decides to create a post and clicks on the specific context area.
2. The user clicks the "Create Post" button and creates his/her posts.
3. The user chooses the category of his/her post and can add images and descriptions to their post.

**Entry conditions:** The user clicks the "Create Post" button.

**Exit conditions:** The user navigates to the posts page.

**Use case name:** Blank Caption Error

**Participating actors:** OnlyBilkent Users

**Flow of events:**

1. The user creates a post and clicks on the specific context area.
2. The user proceeds to enter a caption in the provided field.
3. The user submits the caption field without entering any text.
4. The system detects that the caption field is empty or contains only white spaces.
5. The system triggers an error message to inform the user about the blank caption error.
  - a. The error message includes information on the requirement to provide a non-empty caption.

**Entry conditions:**

1. The user clicks the "Create Post" button.
2. The user submits an empty or whitespace-only caption.

**Exit conditions:**

1. The user receives an error message indicating a blank caption error.
2. The user is prompted to provide a valid, non-empty caption.
3. The user can either enter a valid caption and resubmit, or they can choose to cancel the captioning process.

**Use case name:** Exceed Post Limit

**Participating actors:** OnlyBilkent Users

**Flow of events:**

1. User composes and submits posts on the platform.
2. The user posts content until they reach the predetermined post limit.
3. The system tracks the number of posts the user makes and enforces the post limit.
4. User attempts to post content beyond the established limit.
5. The system detects attempts to exceed the post limit and restricts users from making additional posts.
6. The system displays an error message to inform the user of the post limit.

**Entry conditions:**

1. OnlyBilkent has a predefined limit on the number of posts a user can make.
2. The user is unaware of the post limit.

**Exit conditions:** The system provides feedback to the user regarding the post limit violation, which includes the number of posts remaining before the limit resets.

**Use case name:** Upload Photo

**Participating actors:** OnlyBilkent users

**Flow of events:**

1. User navigates to the "Create Post" then "Upload Photo" feature.
2. The user selects a photo from their device.
3. The user provides any caption related to the photo (optional).
4. The user initiates the upload process.
5. The system uploads the photo to the user's account.
6. The system confirms the successful upload.

**Entry conditions:** The user clicks the Upload Photo button.

**Exit conditions:**

1. The photo has been successfully uploaded to the user's account.
2. The user decides to cancel the upload process.

**Use case name:** Add Caption

**Participating actors:** OnlyBilkent users

**Flow of events:**

1. User selects a post from their account to add a caption.
2. The user clicks on the photo to open it.
3. The user locates the "Add Caption" option.
4. The user enters a caption for the photo.
5. User reviews and confirms the caption.
6. The system associates the caption with the selected photo.

**Entry conditions:** The user is logged into their account and selects a post to add a caption.

**Exit conditions:**

1. The caption has been successfully added to the selected photo.
2. The user decides not to add a caption and cancels the process.

### **Announcement Package**

**Use case name:** View Announcements

**Participating actors:** OnlyBilkent users, Admin, Board

**Flow of events:**

1. User decides to view announcements of student events and clicks to event announcement page

**Entry conditions:** User navigates to the dashboard and views all the announcements.

**Exit conditions:** The user can navigate to the dashboard or posts page.

**Use case name:** Manage Announcement

**Participating actors:**

Primary Actor: Board

Secondary Actor: OnlyBilkent Users

**Flow of events:**

1. Board logs into their account with appropriate permissions on the platform.
2. Board navigates to the "Manage Announcements" or similar section.
3. The system provides options to create, edit, or delete announcements.
4. Board selects one of the options based on their intent:
  - a. To create a new announcement: Board enters a title, content, and any additional details for the new announcement.
  - b. To edit an existing announcement: Board selects the announcement, modifies the content, and saves the changes.
  - c. To delete an announcement: Board selects an announcement to be removed from the platform.
5. After taking the intended action, the system updates the announcements accordingly.
6. If the action taken affects users, they will be notified through Notifications.
7. Users will be able to see the changes in the "View Announcements" use case.

**Entry conditions:** Board clicks the Manage Announcements button.

**Exit conditions:**

1. The admin has successfully created, edited, or deleted announcements.
2. Users, if affected, will see the updated announcements during their "View Announcements" interaction.

**Management Package**

**Use case name:** Manage Users

**Participating actors:** Admin

**Flow of events:**

1. Admin decides to perform some operations: add board account, delete user, ban user. For this purpose, the admin navigates to his/her profile page and clicks the "Users" button for the list of the users.
2. If the admin decides to add a board account,
  - a. Admin registers a new Board Account with the club's provided email and pushes the "Add board member" button.
3. Else, if the admin decides to delete user
  - a. Admin selects the user to be deleted and clicks the "Delete user" button.
4. If the admin decides to ban user

- a. Admin selects the user to be deleted and clicks the "Ban user" button.

**Entry conditions:** Admin enters his/her profile page.

**Exit conditions:** Admin can navigate to the dashboard or log out.

**Use Case Name:** View User Details

**Participating Actors:** Admin

**Flow of Events:**

1. Admin navigates to his/her profile page and clicks the "Users" button for the list of users.
2. Admin selects the user they want to view details from the Users list.
3. The system displays user information, including their profile, posts, and activity.

**Entry Conditions:** The Admin accesses their profile page.

**Exit Conditions:** Admin has reviewed the user's details and may return to the list of users or perform additional actions.

**Use Case Name:** Unban User

**Participating Actors:** Admin

**Flow of Events:**

1. Admin navigates to his/her profile page and clicks the "Users" button for the list of users.
2. Admin clicks on the user from the Users list.
3. The system provides an option to unban the user.
4. Admin confirms the unbanning action.
5. The system updates the user's status from "banned" to "active."

**Entry Conditions:** The Admin is logged in and accesses their profile page.

**Exit Conditions:**

1. Admin has successfully unblocked the user.
2. Admin can choose to return to the list of users or take further actions.



### 3.5.2 Class Diagram

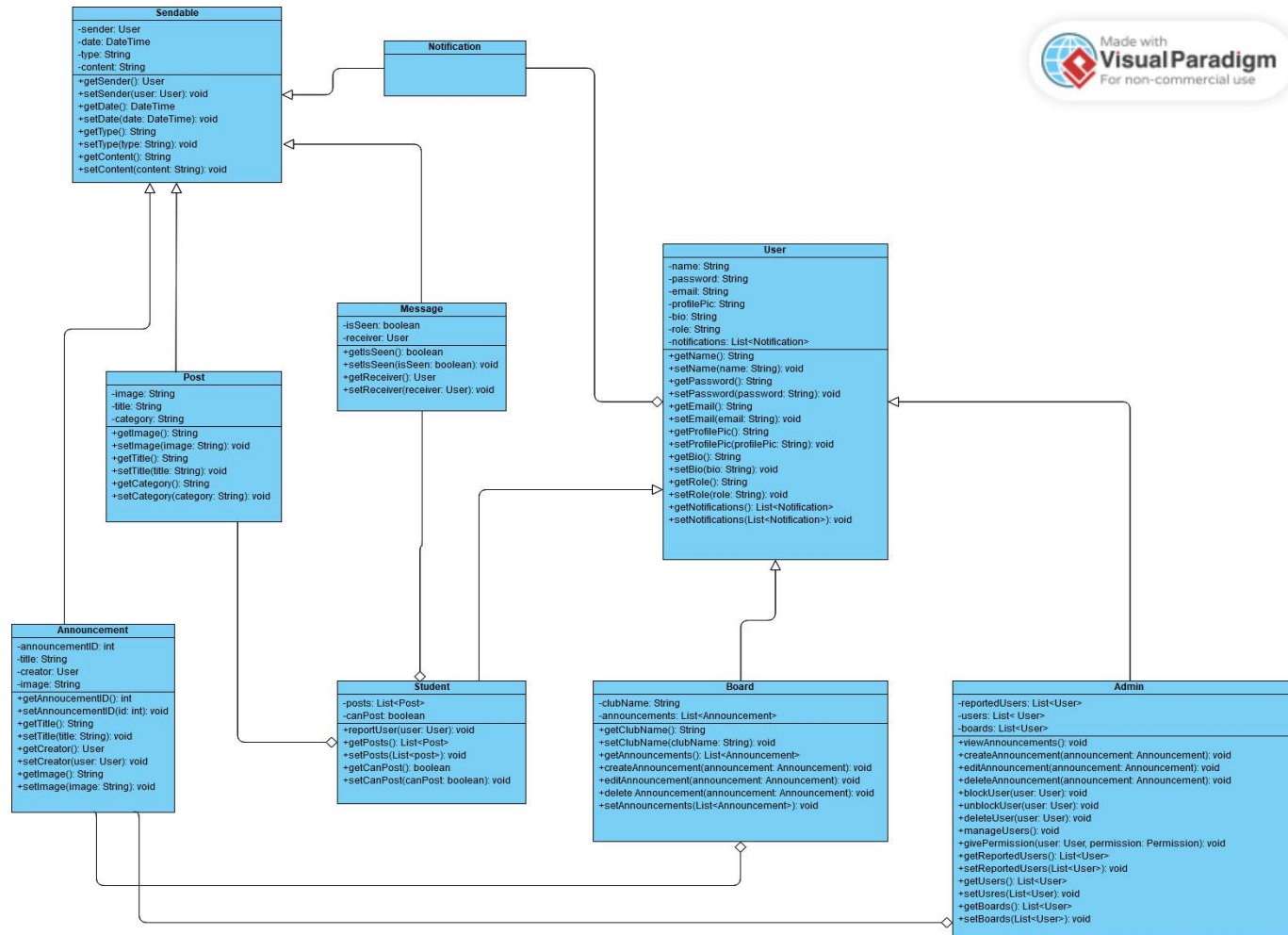


Fig. 2: High-Level UML Class Diagram of OnlyBilkent

**User:** The User class serves as an abstract class, encompassing common attributes shared by all users on the OnlyBilkent platform, including their names, email addresses, profile pictures, passwords, bios, roles (such as student, board, and admin), and a notifications list, given that every user can receive notifications. Various instances of users, such as students, board members, and admins, are instantiated from this abstract class. Users can update their personal information through the settings.

**Student:** The Student class is a child class of abstract User class. Unlike other users, they can submit posts in various categories. The student holds the list of his/her own posts on his profile. Moreover, students can check whether they are allowed to create a post. Besides the posting feature, they can also send direct messages to other students. If a student encounters an inappropriate behavior, he/she can create a complaint report against another user.

**Board:** The Board class is a child class of the abstract User class. Each board will have a unique club name. This type of user can neither create posts nor send messages. However, they can make announcements of their club activities. They can reach their previous announcements through their profiles in order to edit or delete an announcement.

**Admin:** The Admin class is a child class of abstract User class. Admin is responsible for keeping the peace of the application. Admin will have a list of users from which he/she can manage the permissions given to users. Also, admins can view the reports that students create and decide whether to ban the student or not. Besides banning them, they can also permanently delete the user from the application. Finally, they can create an announcement and edit the announcements already created by a board.

**Sendable:** The Sendable class serves as an abstract class, encompassing common attributes of posts, announcements, messages, and notifications such as sender, date, type and content.

**Notification:** The Notification class is a child class of the abstract Sendable class. Every user will have a notification list on their profile page. With the existence of a notification bar, users will be updated with the newest events.

**Message:** The Messages class is a child class of abstract Sendable class. Students can communicate through direct messaging. Students will have access if their messages have been received or seen by the other user.

**Post:** The Post class is a child class of abstract Sendable class. Posts can only be created by students. A post consists of a title, description, and images. Students should specify the category in which he/she will submit.

**Announcement:** The Announcement class is a child class of abstract Sendable class. Unlike other sendables, each announcement will have a unique id. Announcements can only be made by Board accounts.

### 3.5.3. State Diagram

#### 3.5.3.1 State Diagram of Post Availability Status

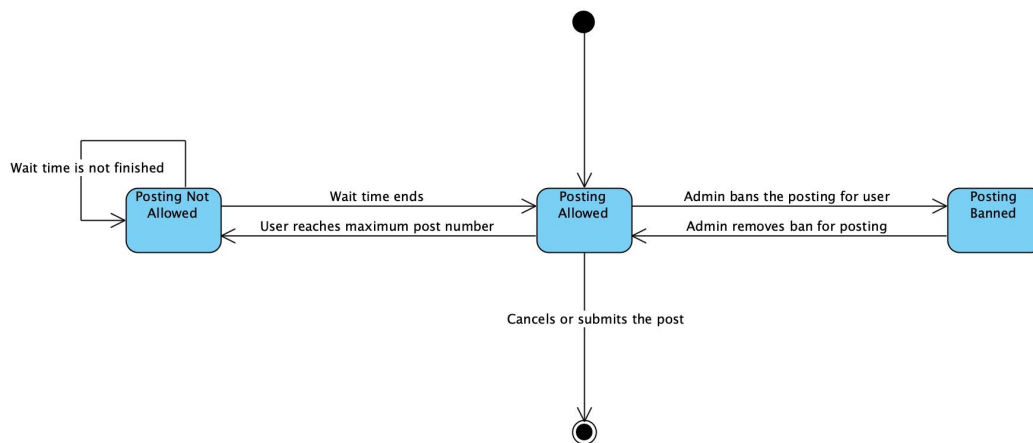
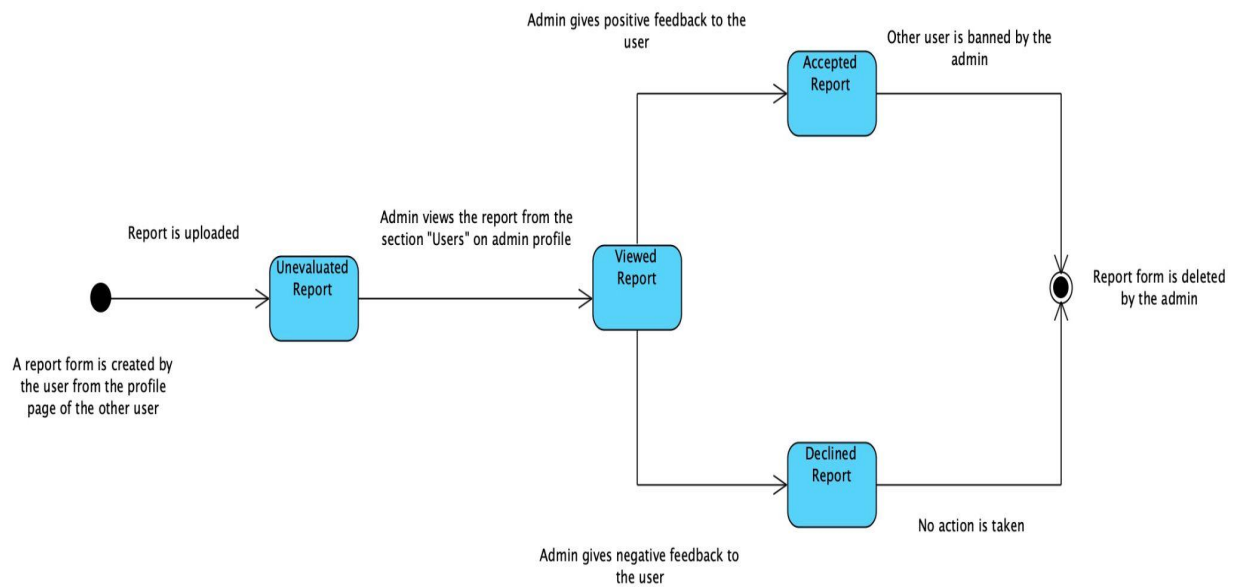


Fig. 3: State Diagram That Shows the Allowance of Creating Post

This state diagram shows the availability of the creating posts. Users will have a maximum post limit for a specified time to avoid unnecessary information. When a user reaches the limit or is banned by the admin, he/she will not be able to post anything. Otherwise, the user should be able to navigate to the post-specification page. Users should also specify the caption of the post.

#### 3.5.3.2 State Diagram of Complaint Report Status



*Fig. 4: State Diagram That Shows the Process of Evaluation of the Complaint Reports*

This state diagram shows the transition of the Complaint Report created by a user. When a user creates a report against another user, the admin can view the report. Until the admin notices the complaint report, it stays as an Unevaluated Report. If an admin views the report from his/her profile page, users will understand that the admin has viewed the report, and they can see their report is under consideration. As the admin decision is finalized, users will claim feedback on whether their report has been accepted or declined. Finally, the admin will delete the report to close the case.

### 3.5.4 Activity Diagram

#### 3.5.4.1 Activity Diagram of User Actions

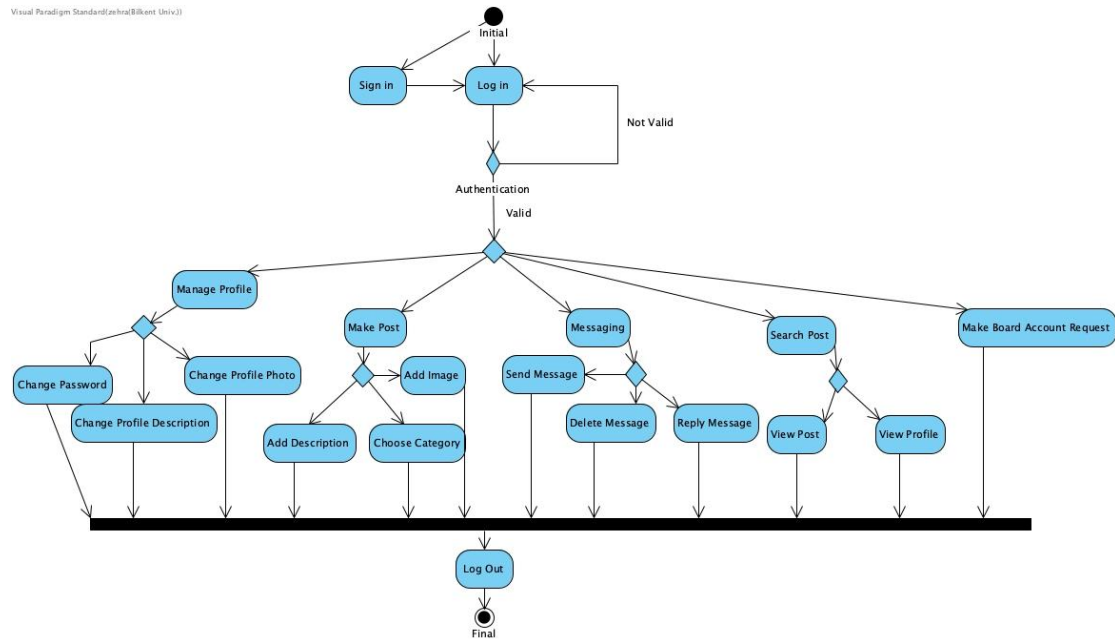


Fig. 5: User Signs In and Log In Into Their Account, Activities They Can Do In Their Account and Log Out

The activity diagram provides an overview of the critical actions and interactions available to users on our platform. It begins with the user signing in and, after successful authentication, offers a range of activities. Users can manage their profiles, edit personal information, and upload profile pictures. They can create and manage posts and share images and descriptions in their posts. Additionally, users can send messages to other users, facilitating direct communication and collaboration. The search feature allows them to find specific posts or users by choosing a category, and then they can view posts and the profile of the sender of the post. Users also have the option to request a board account for club representation, fostering student involvement. Finally, they can log out to conclude their session.

### 3.5.4.2 Activity Diagram of Board Account Request

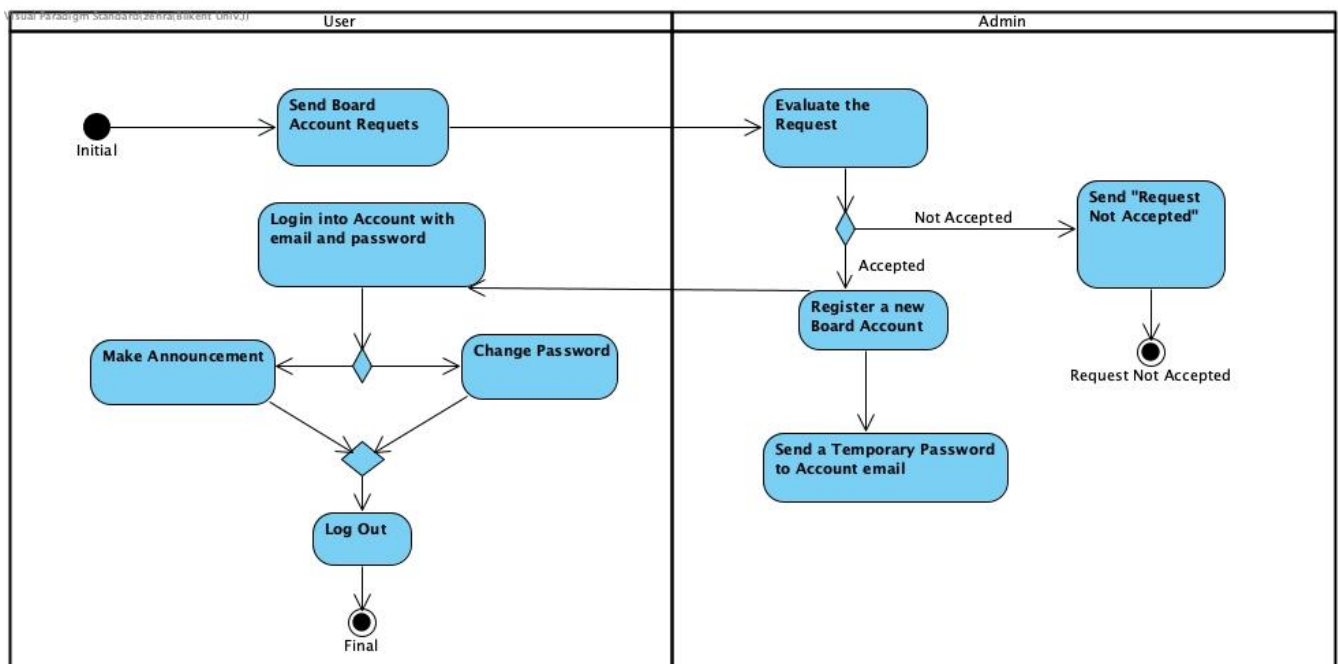


Fig. 6: User Makes a Request For Board Account, Admin Evaluates the Request, Board Account Makes Announcement of Their Activity

In order to have a Board Account, users have to make a request. They submit a formal request through our platform, including essential information, an email address of their student club, and the purpose for which the board account is required. Once the request is received, it is processed by our system. The Admin reviews the application, and if approved, the user is granted access to a board account. A temporary password for their account is sent to their email address. Later, they can log in with that address and password to their account and change their password. Board Account users can only make announcement posts.

### 3.5.4.3 Activity Diagram of Post Creation

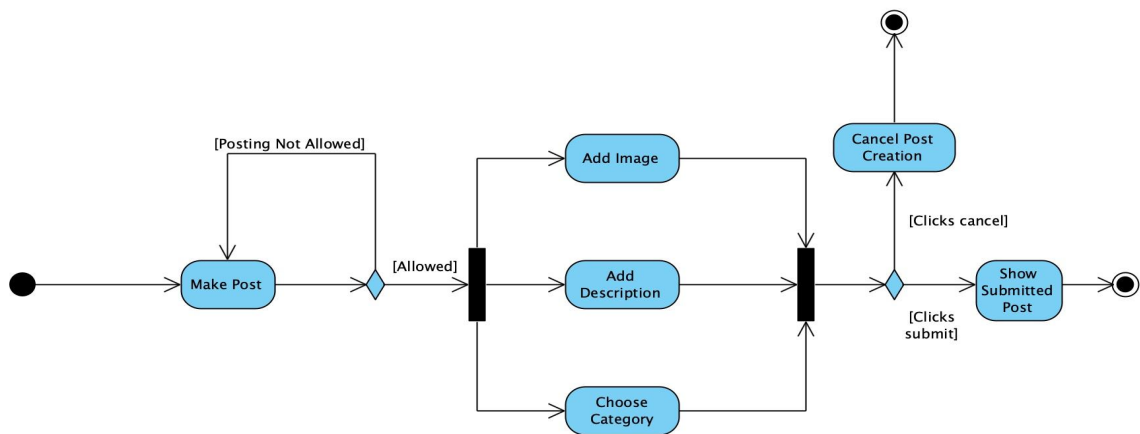


Fig. 7: Activity Diagram for Post Creation Process

This activity diagram describes the process of submitting posts to a particular category. Firstly, users can navigate to the actual post-edit page if they are allowed to. Users have a certain post limit. If they exceed that limit they cannot post for a certain time. Those who can post can add images and descriptions to their posts and choose the particular category they want to share. Lastly, they will either submit or cancel the post-creation process.

### 3.5.5 Sequence Diagram

#### 3.5.5.1 Sequence Diagram for Registration and Login

Equal Paradigm Standard(zehra@Bilkent Univ.)

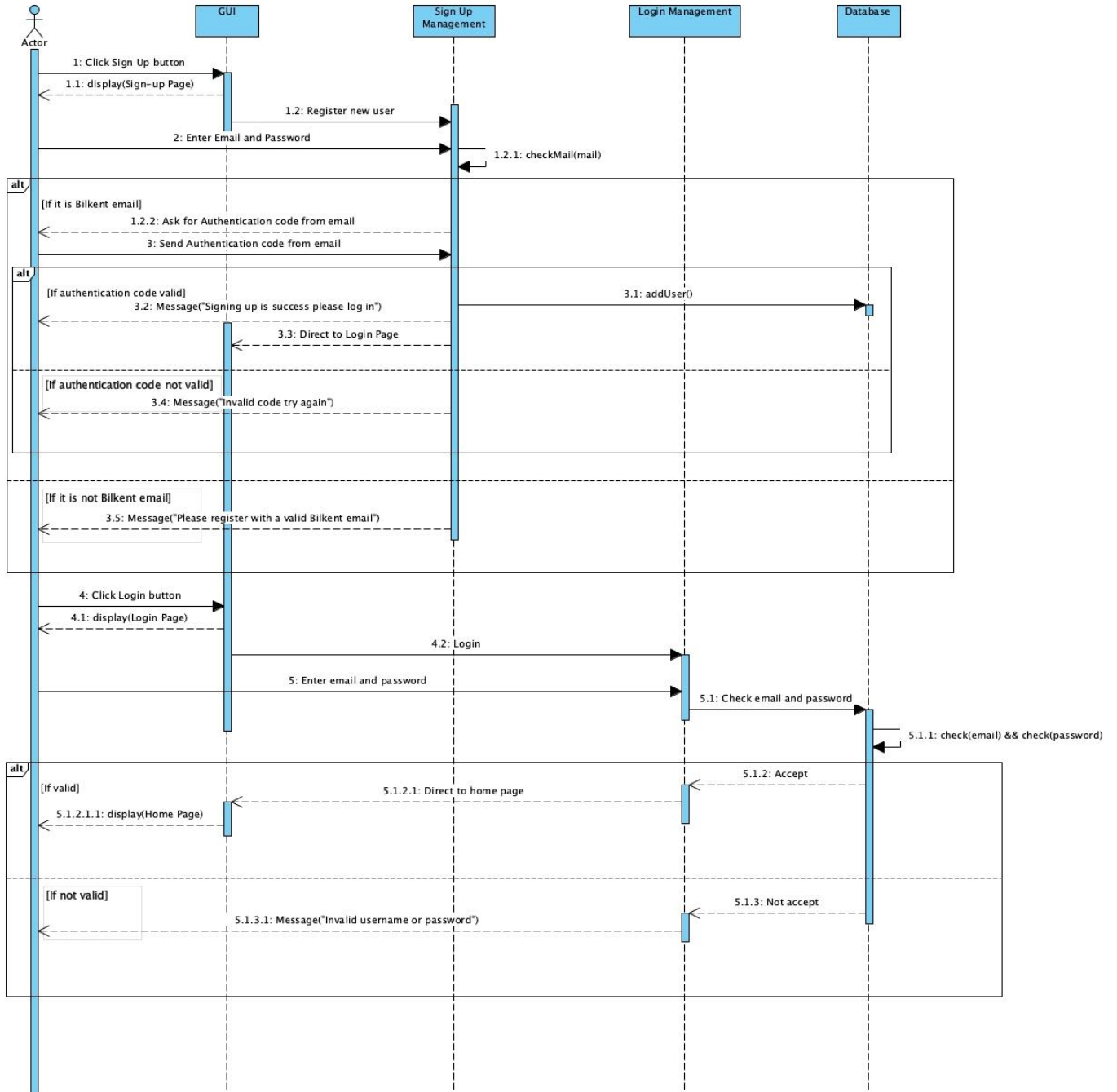


Fig. 8: User Signs Up With Bilkent Email, Confirm Email is Valid Then Login With the Email



Users sign in with their Bilkent University email account and a password. Upon entering their email, the system checks whether it is a Bilkent University email address, and a verification code is sent to their provided email address. The user must then correctly input this verification code to proceed. Once the code is verified, an account is created for the user. Users can log in using their Bilkent University email and a password. After authentication, they are directed to their dashboard or provided an error message.

### 3.5.5.2 Sequence Diagram for Requesting Board Account

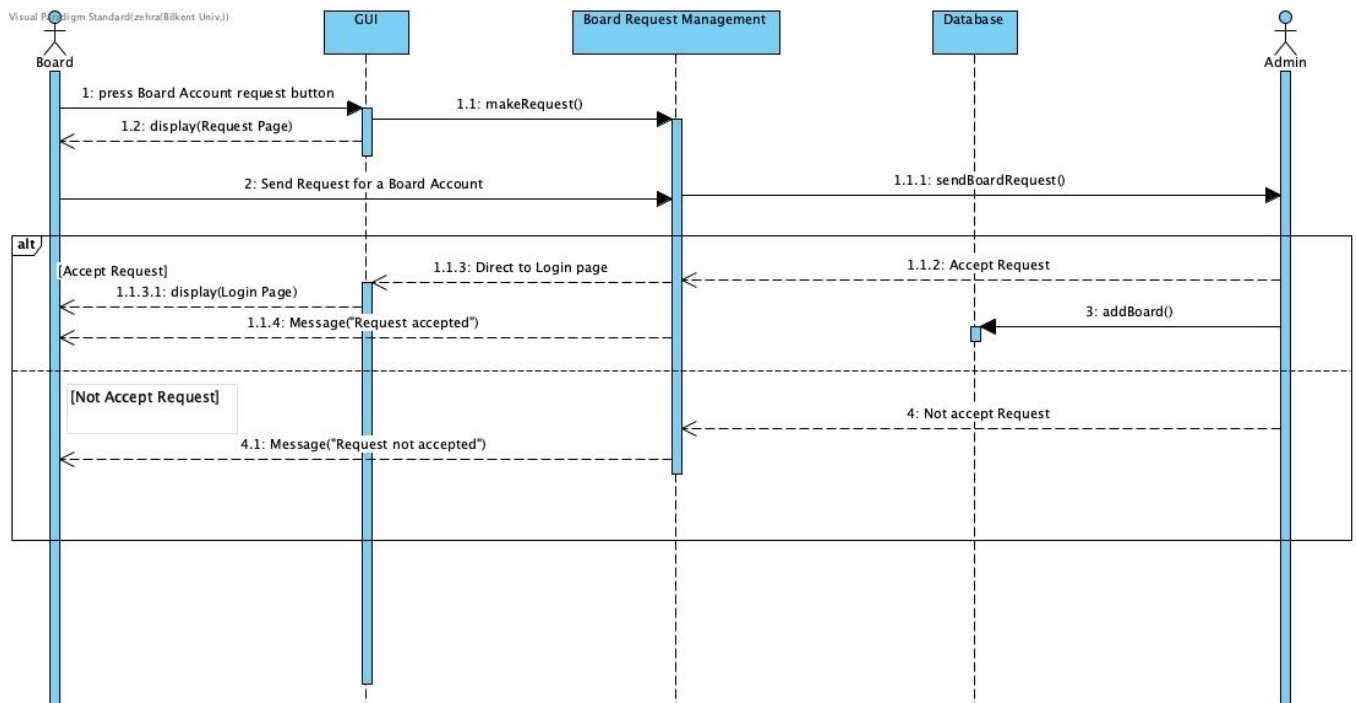


Fig. 9: User Makes a Request For Board Account, Admin Evaluates the Request, Board Account Makes Announcement of Their Activity

The presented sequence diagram outlines the procedure for users to request and obtain a board account, specifically designed for representatives of student clubs. This functionality empowers student club representatives to make announcements and manage club activities effectively. The sequence initiates when users express their intention to request a board account. They submit a formal request through our platform, including essential information, an email address of their student club, and the purpose for which the board account is required. Once the request is received, it is processed by our system. The Admin reviews the application, and if approved, the user is granted access to a board

account. A temporary password for their account is sent to their email address. Later, they can log in with that address and password to their account and change their password. Board Account users can only make announcement posts. This implementation supports users to obtain the necessary tools to enhance student club engagement and communication on our platform.

### 3.5.5.3 Sequence Diagram for Making a Post and Messaging

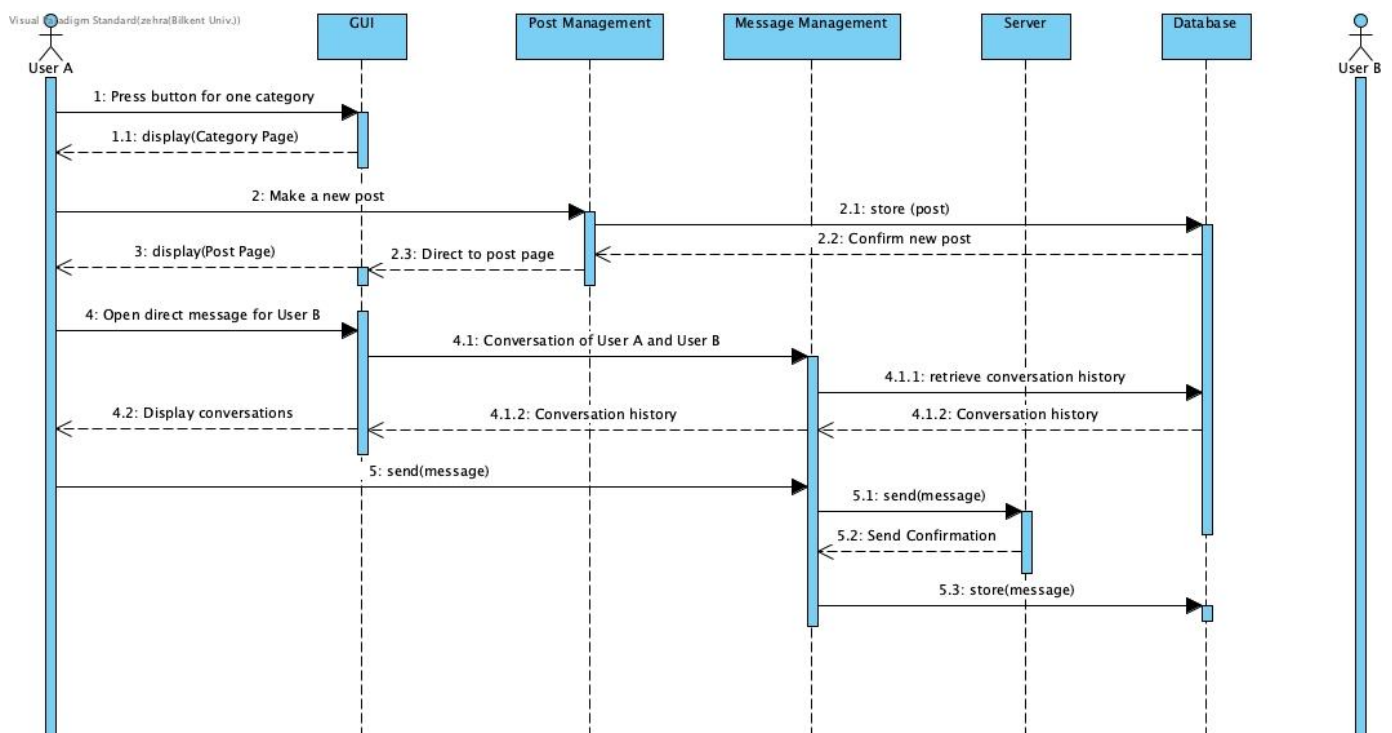



Fig. 10: User Makes a Post By Selecting Category of Their Post and Message Other Users

Users make a post by selecting a category that best suits their post's purpose, such as 'Lost and Found,' 'Second-hand Selling School Materials,' or 'Finding Dorm Friends.' Once a category is chosen, users proceed to compose their post, where they have the option to attach images and provide a description to convey their message effectively. Users can also message other users about their posts. For example, if a user is interested in a material another user is selling, they can directly message them about the material and set up a meeting to buy it. The diagram captures the flow of user interactions and system responses, ensuring that posts are organized and categorized according to their nature, making it easier for other users to discover and engage with the content.

### 3.5.6 User Interface

ooo



## OnlyBilkent

BILKENT MAIL

PASSWORD

[FORGOT  
PASSWORD?](#)

[LOGIN](#) [SIGN UP](#)

Fig. 11: Login Page

ooo

## Create Account

USERNAME

BILKENT MAIL

PASSWORD

PASSWORD CHECK

[SIGN UP](#)

Fig. 12: Sign Up Page

ooo

## Forgot Password

New Password

BILKENT MAIL

jara.martins@ug.bilkent.edu.tr

SEND

Fig. 13: Forgot Password Page

ooo



Search



Fig. 14: Dashboard Page

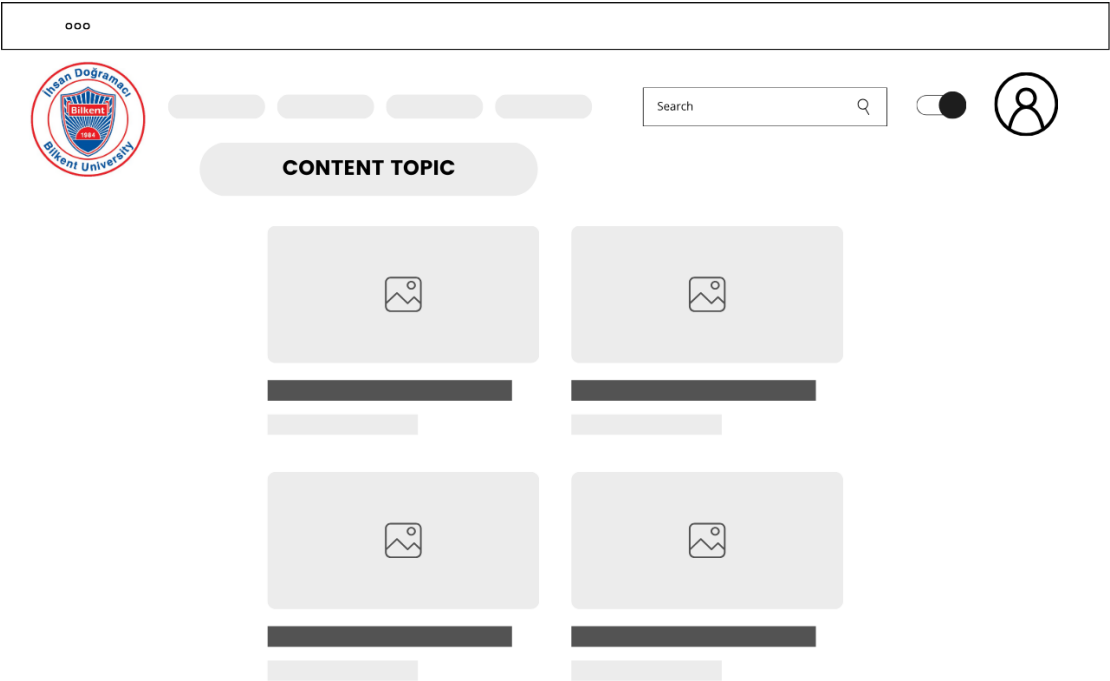


Fig. 15: Specific Content Page

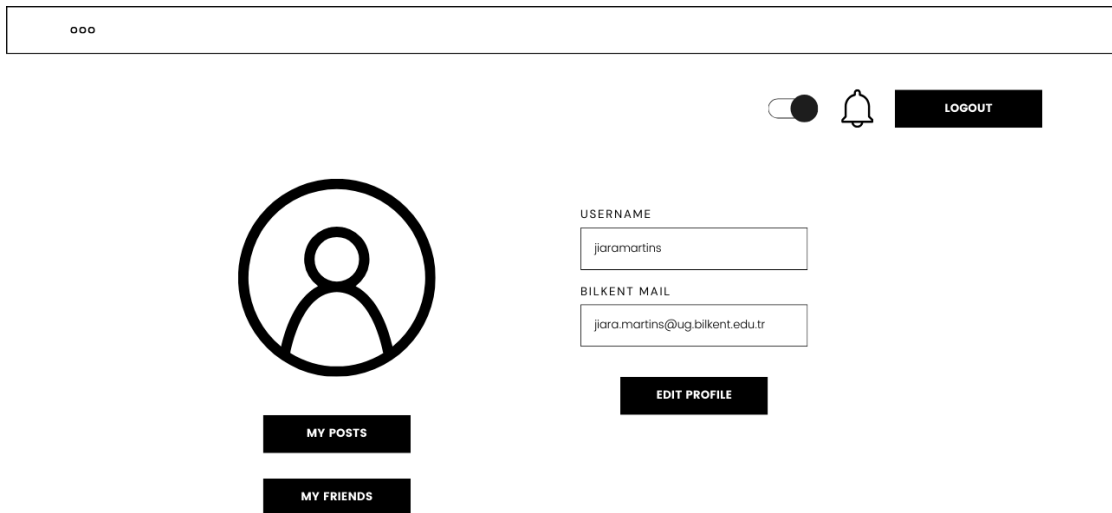


Fig. 16: Profile Page

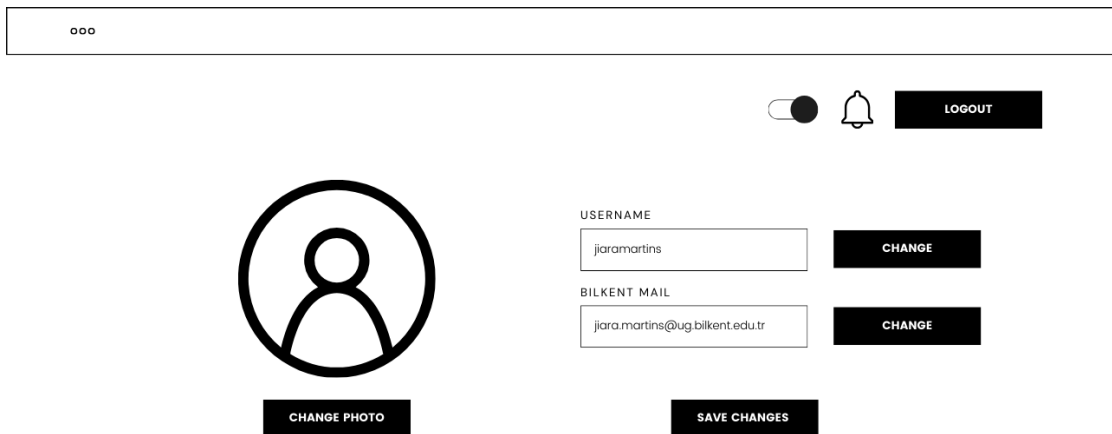


Fig. 17: Edit Profile Page

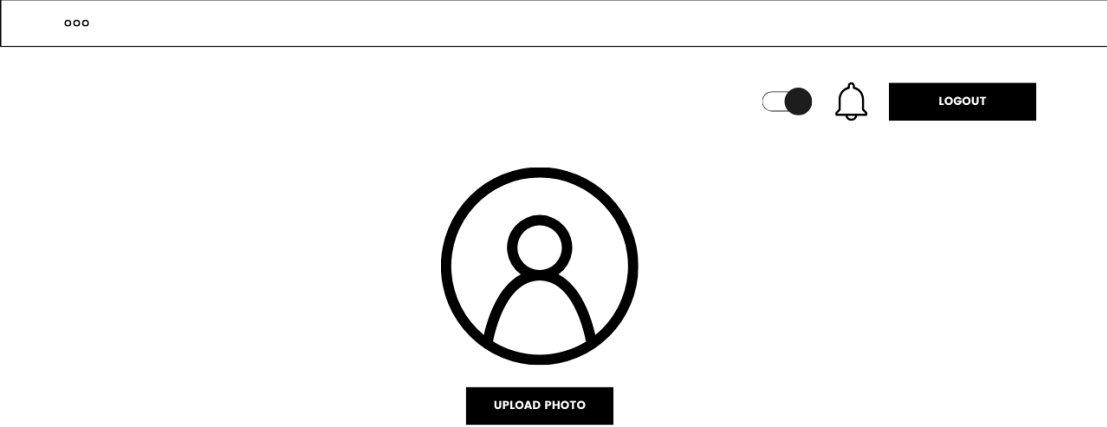


Fig. 18: Change Photo Page

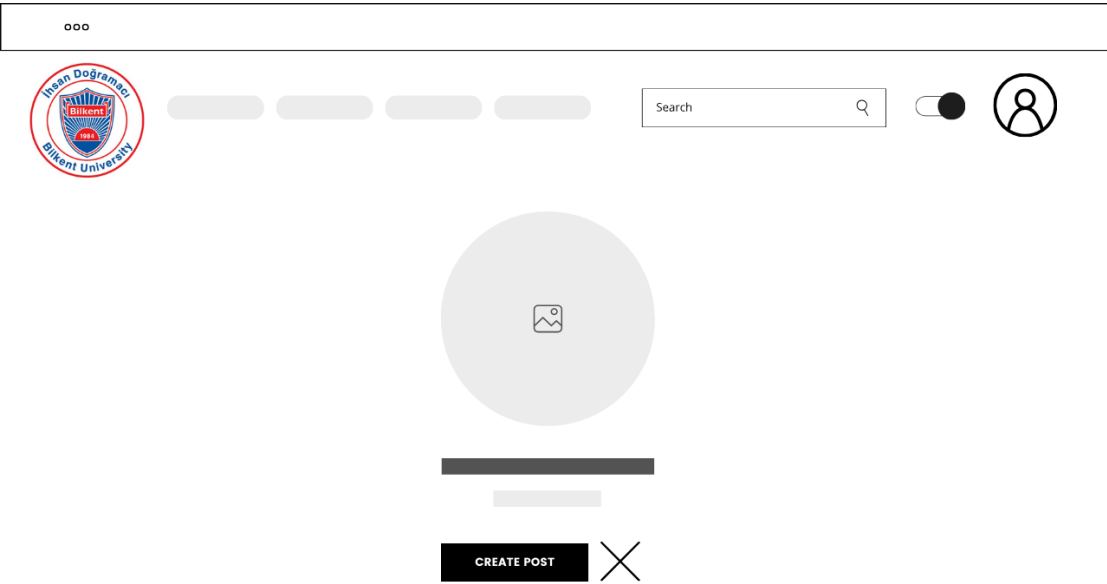


Fig. 19: Create Post Page

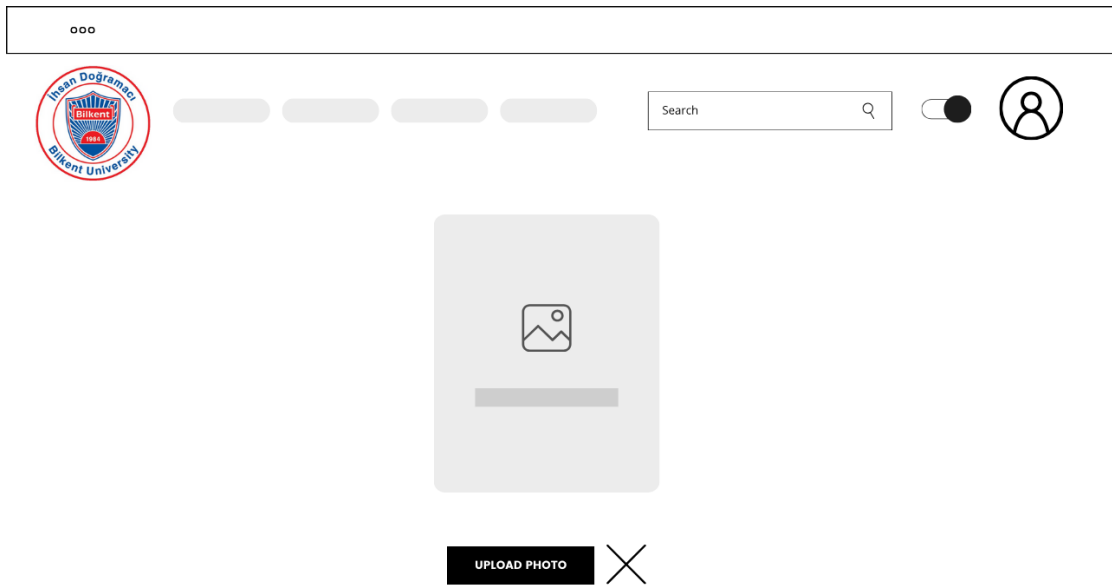


Fig. 20: Upload Post Picture Page

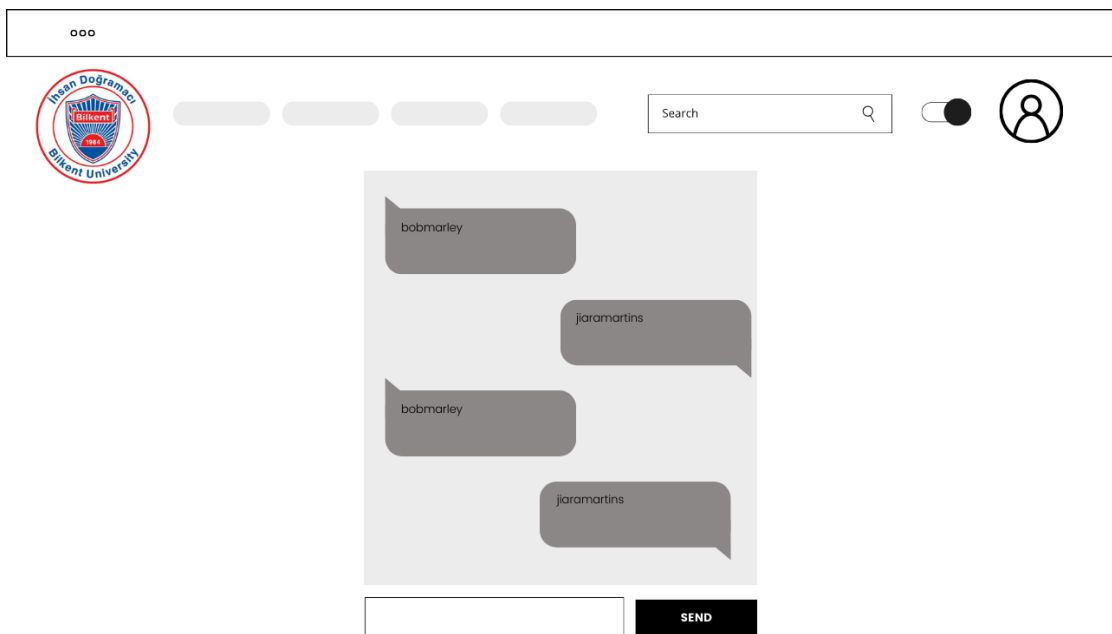


Fig. 21: Chat Page



ooo

LOGOUT

ADD BOARD ACCOUNT

DELETE PROFILE

BAN PROFILE

USERNAME

admin

BILKENT MAIL

jacklondon@ug.bilkent.edu.tr

EDIT PROFILE

Fig. 22: Admin Profile Page

ooo

USERS

Do

Do

Do

Do

Do

ADD BOARD MEMBER

CANCEL

Fig. 23: Add Board Member Page



Fig. 24: Delete Account Page



Fig. 25: Ban Account Page

### 3.6 Used Tech

**Front-end:** JavaScript

We will be using JavaScript for frontend because it is supported by the majority of web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. It also includes frameworks like React and Vue which enables responsive design.

**Back-end:** Java

We will use Java since it has object oriented principles. Additionally, Java has numerous libraries and frameworks that can provide ease for back-end development.

**Database:** MongoDB

MongoDB allows easy storage of different types of information which is useful for OnlyBilkent since it will have a variety of post types and a variety of users. We will also be using the querying capabilities of MongoDB.

## 4 References

- [1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.