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

Iteration 2

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

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. As a platform created exclusively for the Bilkenters, we have incorporated these services into the university life fabric. Whether you are a student tackling your academic journey, a faculty member shaping the future of education, or a staff member contributing to the university's ecosystem, OnlyBilkent makes you connect with others on campus. Our commitment is to empower students, faculty members, and staff with a single, user-friendly platform where they can effortlessly buy and sell items, reunite with lost belongings, and simplify borrowing processes. While other systems may exist for these purposes, OnlyBilkent stands apart by simplifying and enriching the lives of newcomers and longtime university members. We are dedicated to the students, instructors, 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 lesson materials or dormitory belongings, finding people to carpool with or just getting valuable advice about school life. We have observed some of these ways:

- **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.
    - **Challenges:** Posts on Instagram are transient, making it easy for information to get lost, and there is no effective search functionality.
- **External Applications:** Bilkent students also use external apps like "sahibinden" 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, potentially causing 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 students and incorporate advanced search and filtering functions for enhanced usability. 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 lesson materials, lost and found, changing dormitories, second-hand dorm furniture sales, carpooling, animal adoption, seeking lesson and teacher advice, and requesting private tutoring services. Users can select their preferred category for posting, facilitating interaction between users looking to buy products, arrange carpooling, and more. Additionally, we will introduce board user representatives of student clubs, who will have profiles dedicated to making announcements about club activities, thus simplifying the process of disseminating information about club events to the Bilkent University student body.

3.2 Actors

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

- User: Bilkent University student who logs in with their university email. After signing in, they can customize their profile and add posts to their desired categories. They can directly message the poster if they are interested in a post.
- Board: Representative of a Bilkent University student club. It can only be added by the admin. To get added, a user from the club has to make a request. They cannot post on the categories or messages. They can only post and edit their announcement of their club activities.
- Admin: Manages the requests for Board accounts. Can view posts. If users get reported, check the reports and ban the users accordingly.

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. This includes optimizing database queries to handle increased traffic efficiently.

3.3.2 Safety

- Authentication: Authentication system where users need to log in with their own university e-mail addresses to ensure that they are from Bilkent University. This means that each student will have only one account associated with their university e-mail address.
- Personal privacy: The information that the user wants to share will be available for other system users.
- Encryption: Encryption is essential to safeguard sensitive information when it is shared or stored.
- Access Control: Access control to restrict unauthorized access to specific platform areas. Lecturers won't be authorized to access recommendation forums. Users should only be granted access to data and features for which they have been authorized.
- Secure File Upload: It is crucial to implement security measures to prevent any malicious file uploads, such as uploading photos of secondhand objects. These measures will include verifying file types, imposing size limits, etc.

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. Using redundant database systems with replication to ensure data remains available even if one database server fails will minimize data loss.

3.3.4 Usability

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

- Intuitive User Interface: It will allow users to navigate the OnlyBilkent quickly. Ensuring a logical layout, consistent design elements, and straightforward menus will promote ease of use.
- Feedback Mechanisms: Feedback mechanisms will be included, such as notifications, to notify users of real-time events, second-hand deals, etc.

- Search and Navigation: Search functionality and navigation tools to help users quickly find specific posts in sections within the platform. We will include filters and sorting options for results to be easily accessible.
- Personalization Options: The OnlyBilkent will allow users to personalize the OnlyBilkent by saving their notification settings. 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.

- Modular Architecture: Modular architecture where application components are organized into separate modules will make it easier to update or extend the OnlyBilkent's features without affecting the entire system.
- Scalable Infrastructure: By using cloud services, the addition of new features will be more accessible.

3.3.6 Compatibility

- Responsive Design: Given that users will use various devices to share on OnlyBilkent, our application should be compatible with various devices. Responsive design implementation will be done, such as touch-friendly controls, etc., to ensure that OnlyBilkent remains accessible and user-friendly on various devices and screen sizes.

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

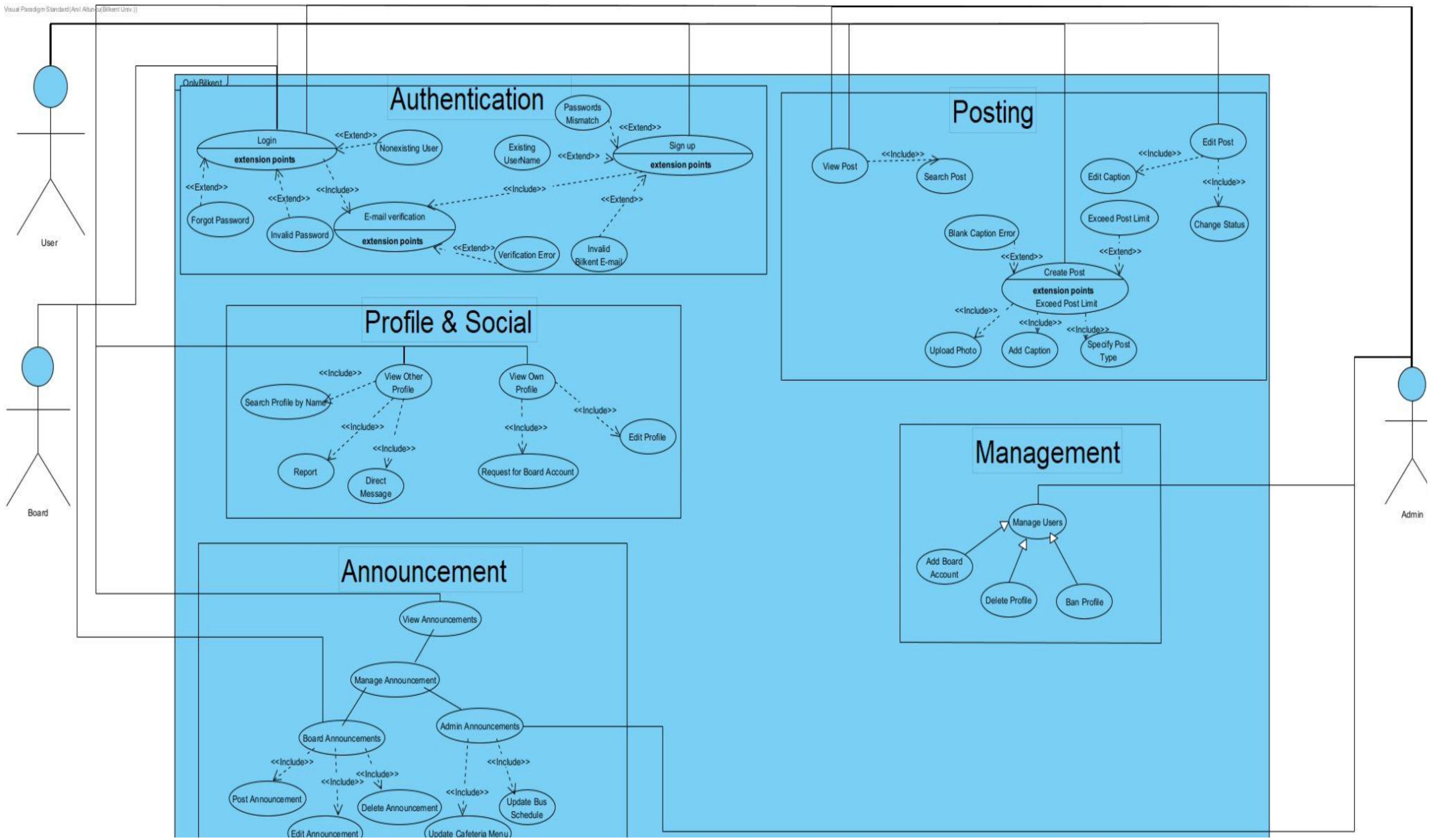


Fig. 1: Use Case Diagram for the Campuswide Connection System

Authentication Package

Use case name: Login

Participating actors: OnlyBilkent users

Flow of events:

1. The user inputs their Bilkent University email and password into the mail and password boxes.
2. If the user enters his/her mail incorrectly,
    a. "Mail does not exist" error message will be displayed.
3. Else, if the user enters his/her password incorrectly,
    a. "Wrong Password" error message will be displayed.
4. Else, the user successfully completes the login process.

Entry condition: Opening the website by using a web browser.

Exit condition: The login process fails, or the user chooses to sign up.

Use case name: Sign up

Participating actors: OnlyBilkent users

Flow of events:

1. The user decides to sign up and clicks the "Sign Up" button.
2. If the user's username is not unique,
    a. "Please enter a unique username" message will be displayed.
3. If the Bilkent mail the user entered is invalid,
    a. "Please enter a valid Bilkent mail" message will be displayed.
4. If the two password boxes for verification do not match
    a. "Passwords do not match" error message will be displayed.
5. Else, the user completes the sign-up process successfully.

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

Exit condition: Clicking the "Return" button or completing the sign-up process.

Profile and Social Package

Use case name: Search profile by Username

Participating actors: OnlyBilkent users

Flow of events:

1. OnlyBilkent 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. Then, the user can click the profile he/she wants to see.

Entry conditions: Searching names through the search bar.

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

Use case name: View other profiles

Participating actors: OnlyBilkent users, Admin, Board

Flow of events:

1. The user can scroll the user profiles.
2. If the user finds the wanted profile from the profile page results page, the user clicks on the profile.

Entry conditions: Navigating the user to the profile page results.

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

Use case name: No profile with searched Username

Participating actors: OnlyBilkent users

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: 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

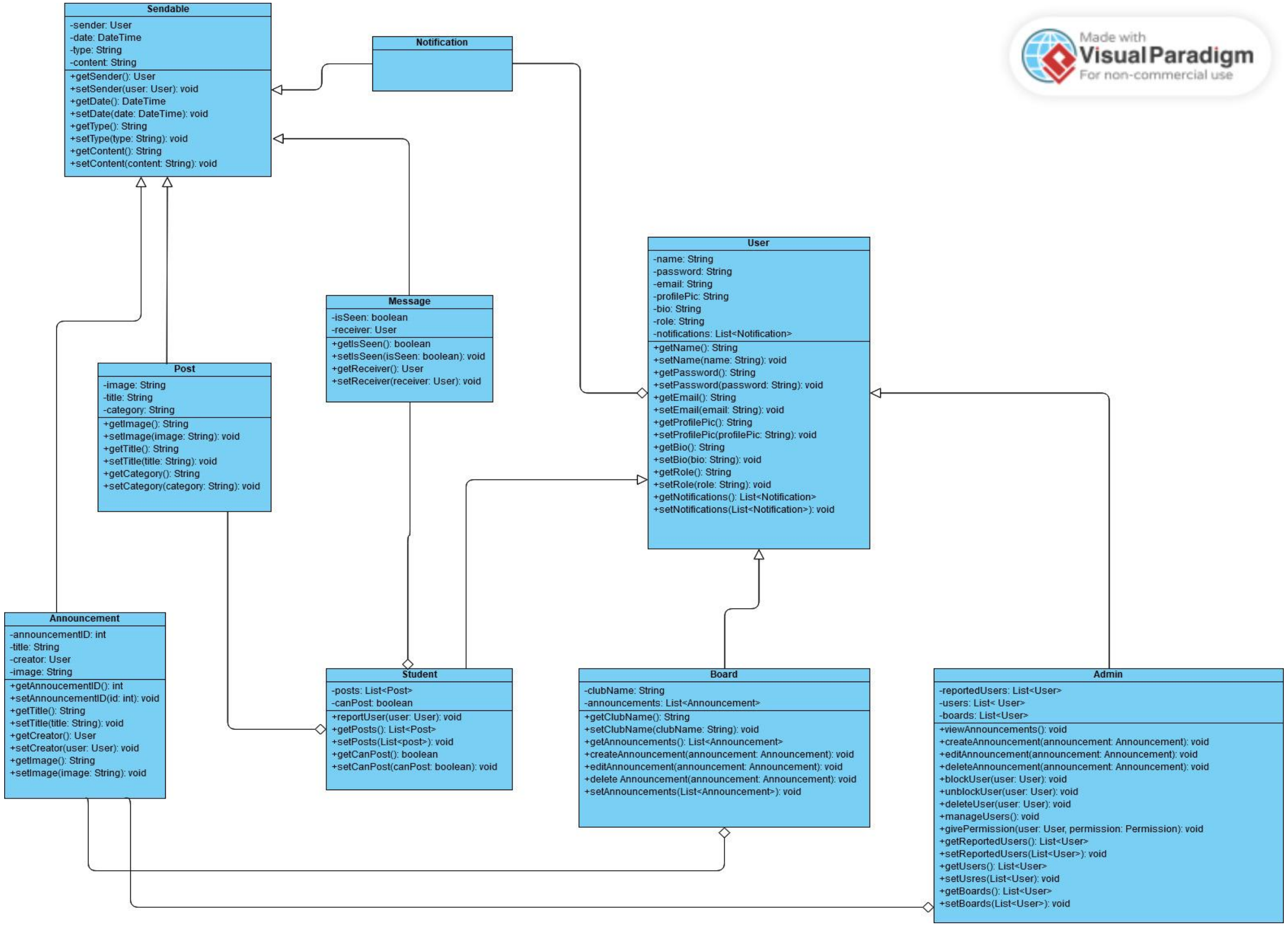

Sendable
-sender: User
-date: DateTime
-type: String
-content: String
+getSender(): User
+setSender(user: User): void
+getDate(): DateTime
+setDate(date: DateTime): void
+getType(): String
+setType(type: String): void
+getContent(): String
+setContent(content: String): void
Notification
Made with
Visual Paradigm
For non-commercial use
Message
-isSeen: boolean
-receiver: User
+getIsSeen(): boolean
+setIsSeen(isSeen: boolean): void
+getReceiver(): User
+setReceiver(receiver: User): void
Post
-image: String
-title: String
-category: String
+getImage(): String
+setImage(image: String): void
+getTitle(): String
+setTitle(title: String): void
+getCategory(): String
+setCategory(category: String): void
User
-name: String
-password: String
-email: String
-profilePic: String
-bio: String
-role: String
-notifications: List<Notification>
+getName(): String
+setName(name: String): void
+getPassword(): String
+setPassword(password: String): void
+getEmail(): String
+setEmail(email: String): void
+getProfilePic(): String
+setProfilePic(profilePic: String): void
+getBio(): String
+setBio(bio: String): void
+getRole(): String
+setRole(role: String): void
+getNotifications(): List<Notification>
+setNotifications(List<Notification>): void
Announcement
-announcementID: int
-title: String
-creator: User
-image: String
+getAnnoucementID(): int
+setAnnouncementID(id: int): void
+getTitle(): String
+setTitle(title: String): void
+getCreator(): User
+setCreator(user: User): void
+getImage(): String
+setImage(image: String): void
Student
-posts: List<Post>
-canPost: boolean
+reportUser(user: User): void
+getPosts(): List<Post>
+setPosts(List<post>): void
+getCanPost(): boolean
+setCanPost(canPost: boolean): void
Board
-clubName: String
-announcements: List<Announcement>
+getClubName(): String
+setClubName(clubName: String): void
+getAnnouncements(): List<Announcement>
+createAnnouncement(announcement: Announcement): void
+editAnnouncement(announcement: Announcement): void
+delete Announcement(announcement: Announcement): void
+setAnnouncements(List<Announcement>): void
Admin
-reportedUsers: List<User>
-users: List< User>
-boards: List<User>
+viewAnnouncements(): void
+createAnnouncement(announcement: Announcement): void
+editAnnouncement(announcement: Announcement): void
+deleteAnnouncement(announcement: Announcement): void
+blockUser(user: User): void
+unblockUser(user: User): void
+deleteUser(user: User): void
+manageUsers(): void
+givePermission(user: User, permission: Permission): void
+getReportedUsers(): List<User>
+setReportedUsers(List<User>): void
+getUsers(): List<User>
+setUsres(List<User): void
+getBoards(): List<User>
+setBoards(List<User>): void

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 an 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

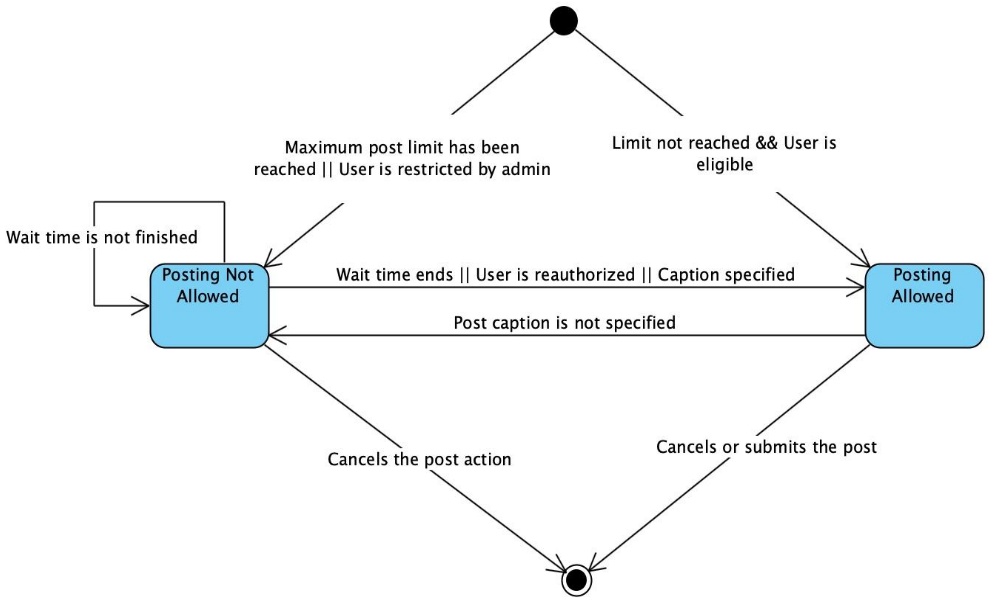

Maximum post limit has been reached || User is restricted by admin
Limit not reached && User is eligible
Wait time is not finished
Posting Not Allowed
Wait time ends || User is reauthorized || Caption specified
Posting Allowed
Post caption is not specified
Cancels the post action
Cancels or submits the post

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 restricted 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

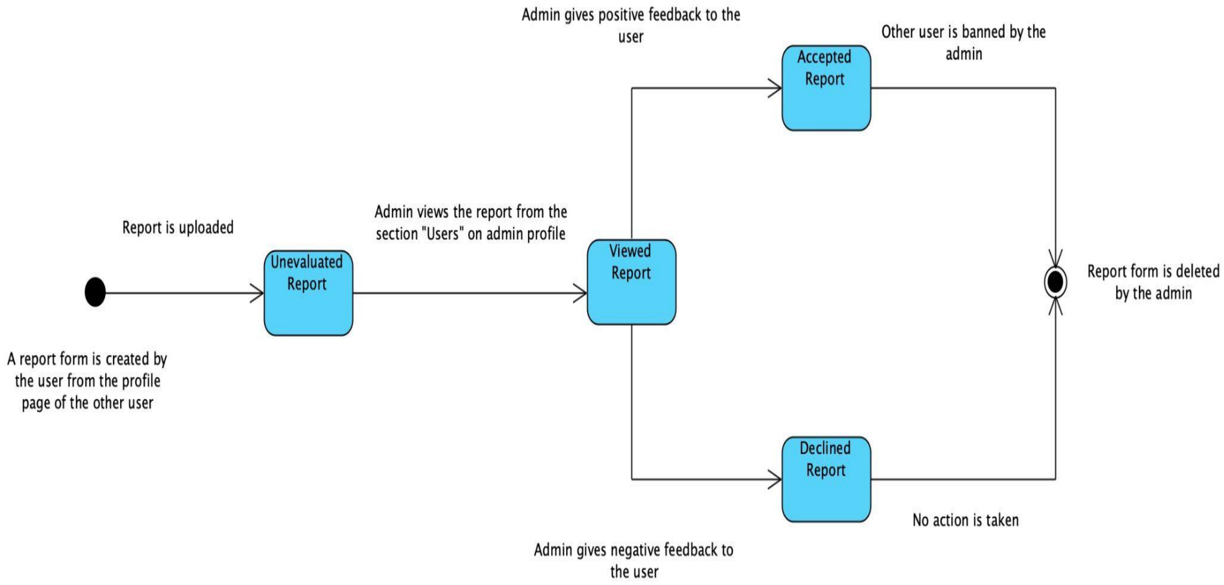

Admin gives positive feedback to the user
Other user is banned by the admin
Accepted Report
Report is uploaded
Admin views the report from the section "Users" on admin profile
Unevaluated Report
Viewed Report
Report form is deleted by the admin
A report form is created by the user from the profile page of the other user
Declined Report
No action is taken
Admin gives negative feedback to the user

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

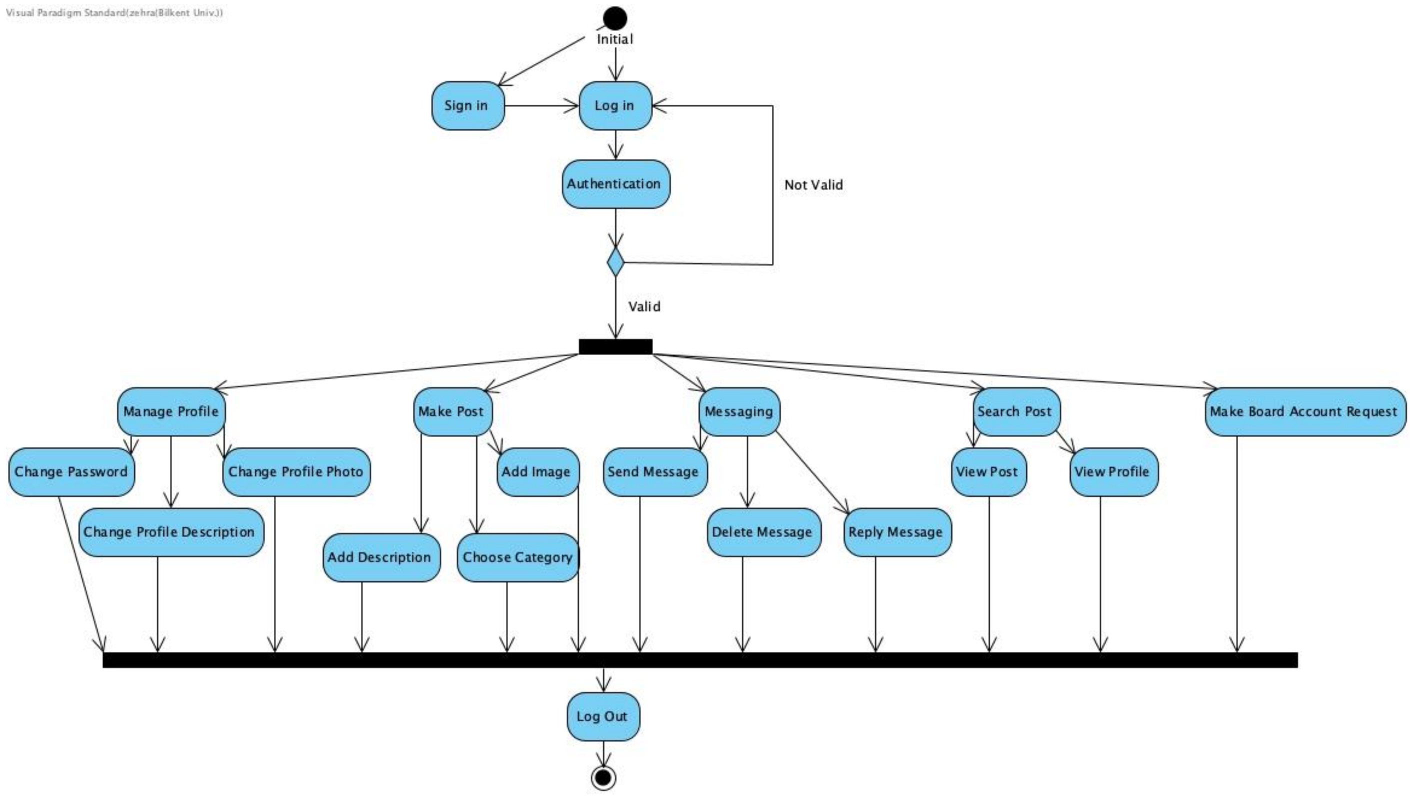


Fig. 5: User Signs In and Log In Into Their Account, Activities They Can Do In Their Account and Log Out

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

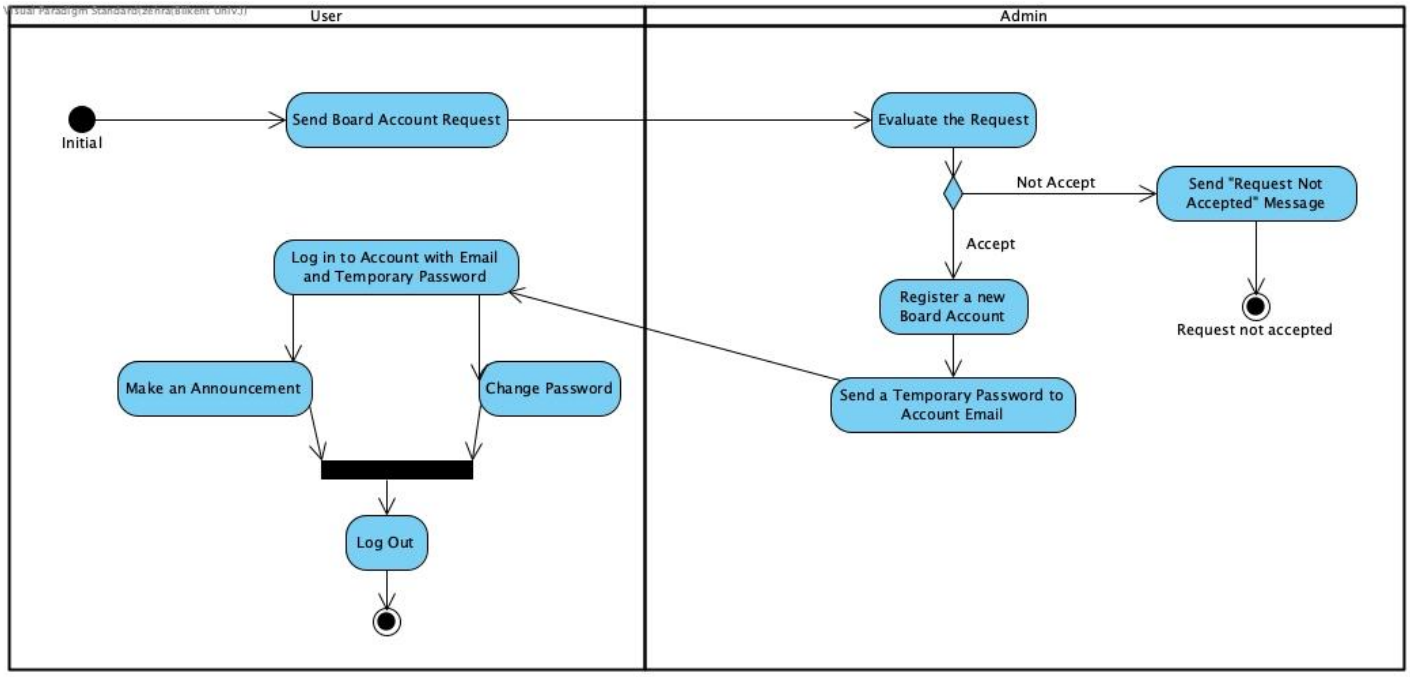

Visual Paradigm Standard(zehra(Bilkent Univ.))
User
Admin
Initial
Send Board Account Request
Evaluate the Request
Not Accept
Send "Request Not Accepted" Message
Accept
Register a new Board Account
Request not accepted
Log in to Account with Email and Temporary Password
Make an Announcement
Change Password
Send a Temporary Password to Account Email
Log Out

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

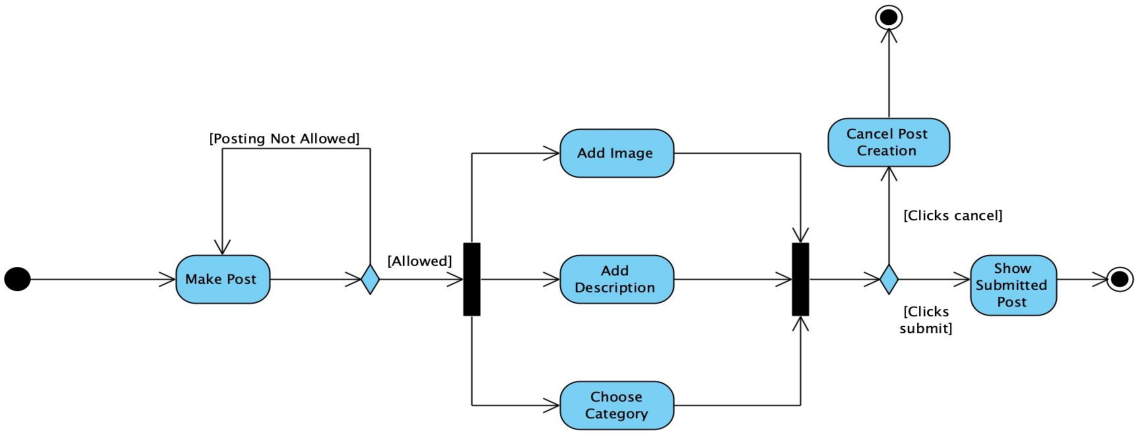


Fig. 7: Activity Diagram for Post Creation Process

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. Those who navigate can add images, add descriptions, 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

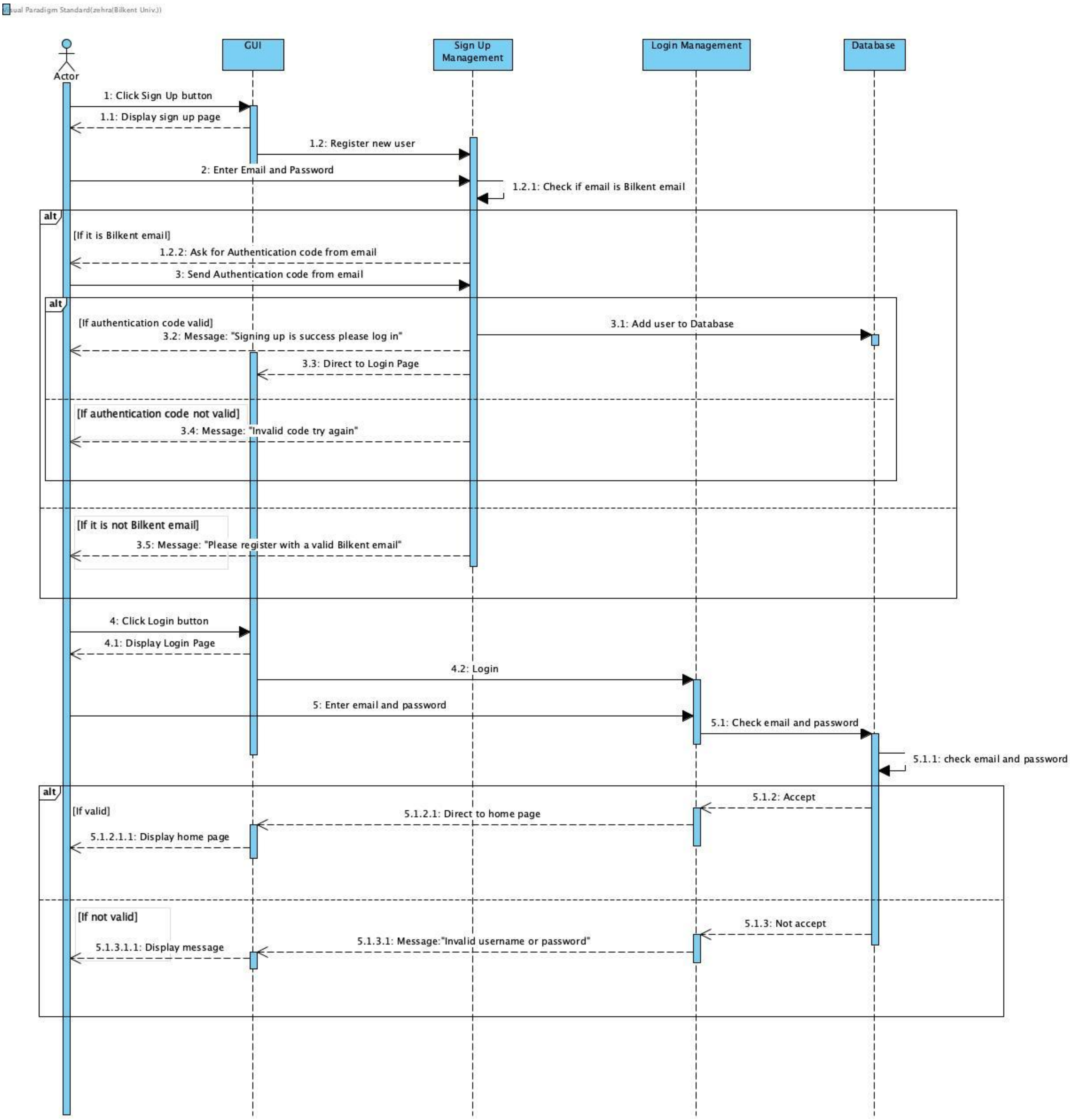

Visual Paradigm Standard(zehra(Bilkent Univ.))
Actor
GUI
Sign Up Management
Login Management
Database
1: Click Sign Up button
1.1: Display sign up page
1.2: Register new user
2: Enter Email and Password
1.2.1: Check if email is Bilkent email
alt
[If it is Bilkent email]
1.2.2: Ask for Authentication code from email
3: Send Authentication code from email
alt
[If authentication code valid]
3.1: Add user to Database
3.2: Message: "Signing up is success please log in"
3.3: Direct to Login Page
[If authentication code not valid]
3.4: Message: "Invalid code try again"
[If it is not Bilkent email]
3.5: Message: "Please register with a valid Bilkent email"
4: Click Login button
4.1: Display Login Page
4.2: Login
5: Enter email and password
5.1: Check email and password
5.1.1: check email and password
alt
[If valid]
5.1.2: Accept
5.1.2.1: Direct to home page
5.1.2.1.1: Display home page
[If not valid]
5.1.3: Not accept
5.1.3.1: Message:"Invalid username or password"
5.1.3.1.1: Display message

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

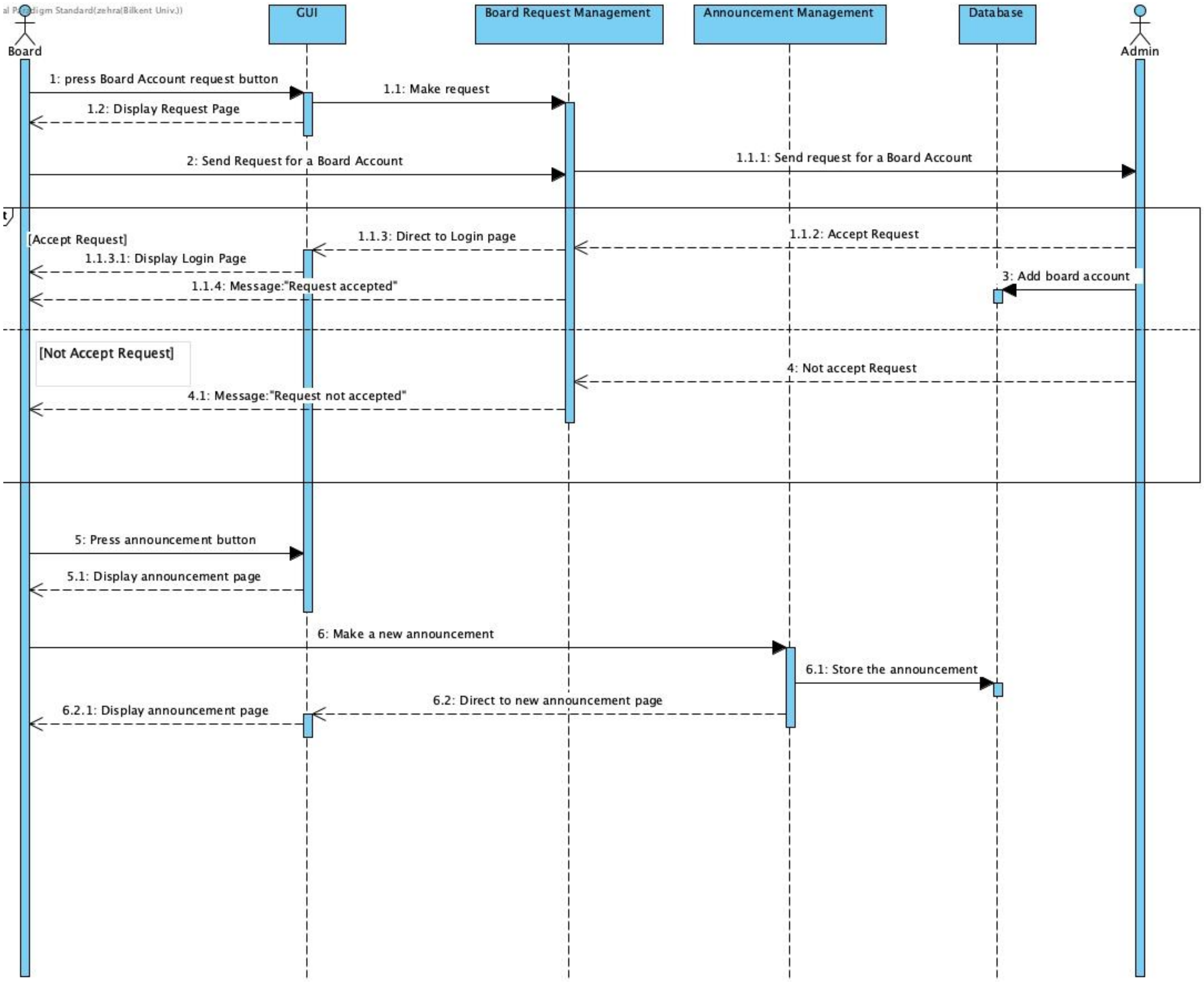

al Paradigm Standard(zehra(Bilkent Univ.))
Board
GUI
Board Request Management
Announcement Management
Database
Admin
1: press Board Account request button
1.1: Make request
1.2: Display Request Page
2: Send Request for a Board Account
1.1.1: Send request for a Board Account
[Accept Request]
1.1.3: Direct to Login page
1.1.2: Accept Request
1.1.3.1: Display Login Page
1.1.4: Message:"Request accepted"
3: Add board account
[Not Accept Request]
4: Not accept Request
4.1: Message:"Request not accepted"
5: Press announcement button
5.1: Display announcement page
6: Make a new announcement
6.1: Store the announcement
6.2: Direct to new announcement page
6.2.1: Display announcement page

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

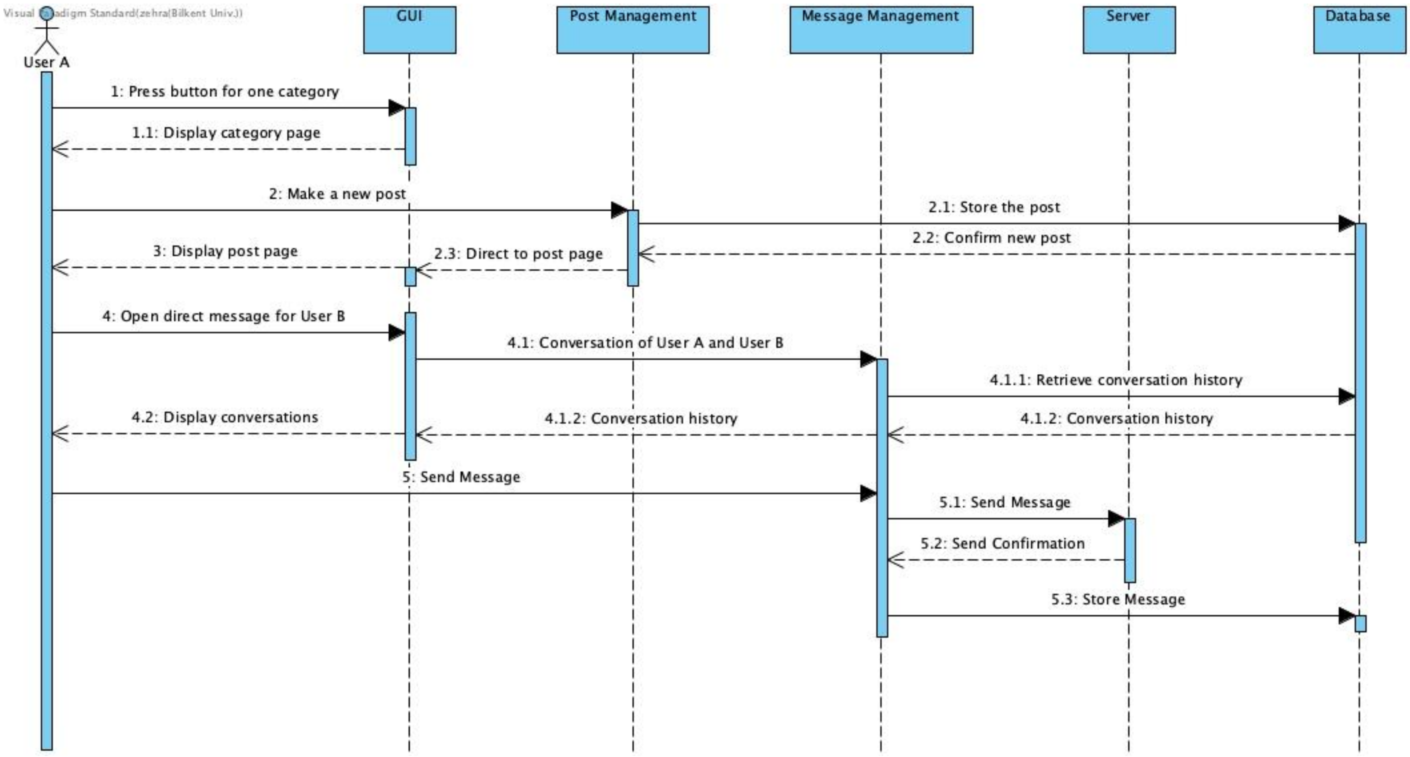
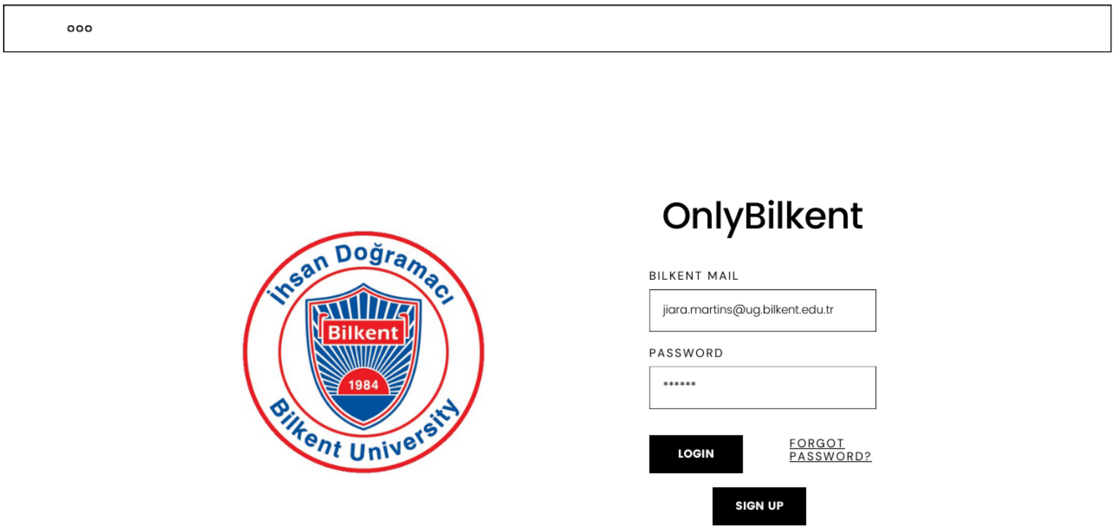

Visual Paradigm Standard(zehra(Bilkent Univ.))
User A
GUI
Post Management
Message Management
Server
Database
1: Press button for one category
1.1: Display category page
2: Make a new post
2.1: Store the post
2.2: Confirm new post
3: Display post page
2.3: Direct to post page
4: Open direct message for User B
4.1: Conversation of User A and User B
4.1.1: Retrieve conversation history
4.2: Display conversations
4.1.2: Conversation history
4.1.2: Conversation history
5: Send Message
5.1: Send Message
5.2: Send Confirmation
5.3: Store Message

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.

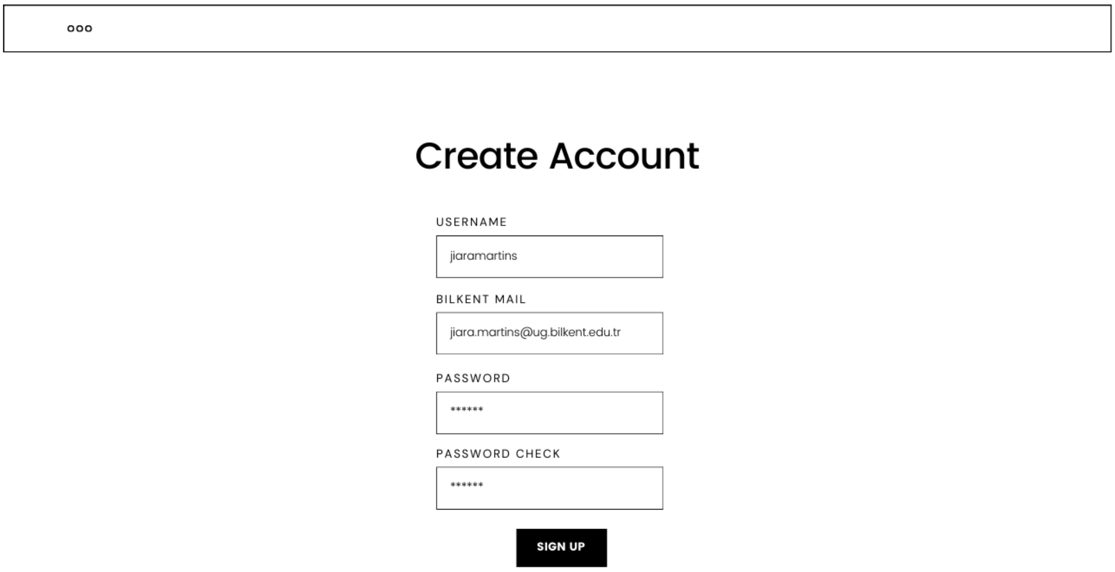
3.5.6 User Interface



BILKENT MAIL: jiara.martins@ug.bilkent.edu.tr

PASSWORD: ******

Fig. 11: Login Page



Create Account

USERNAME

jiaramartins

BILKENT MAIL

jiara.martins@ug.bilkent.edu.tr

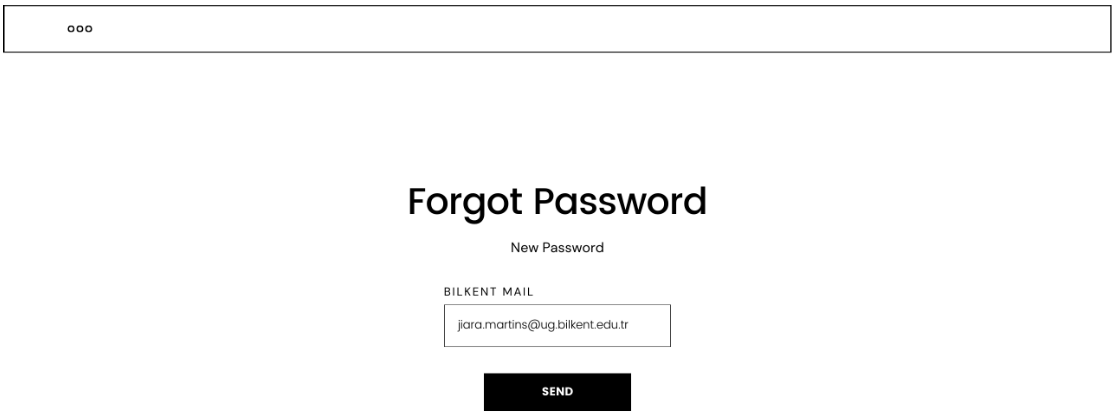
PASSWORD

******

PASSWORD CHECK

******

Fig. 12: Sign Up Page



Forgot Password

New Password

BILKENT MAIL

jiara.martins@ug.bilkent.edu.tr

Fig. 13: Forgot Password Page

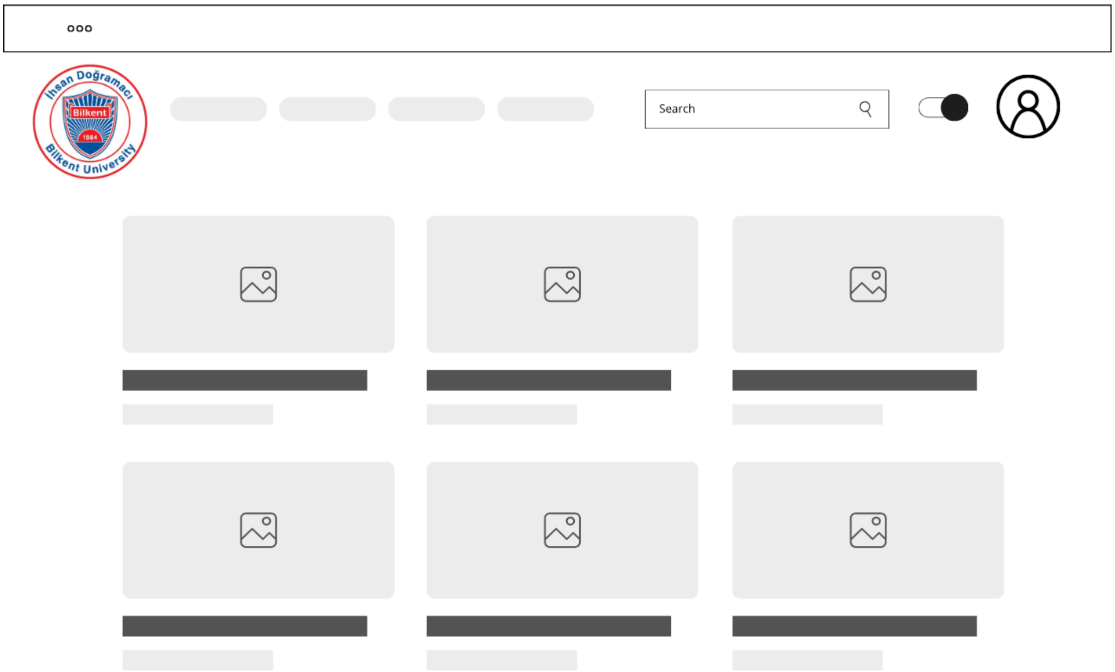


Fig. 14: Dashboard Page

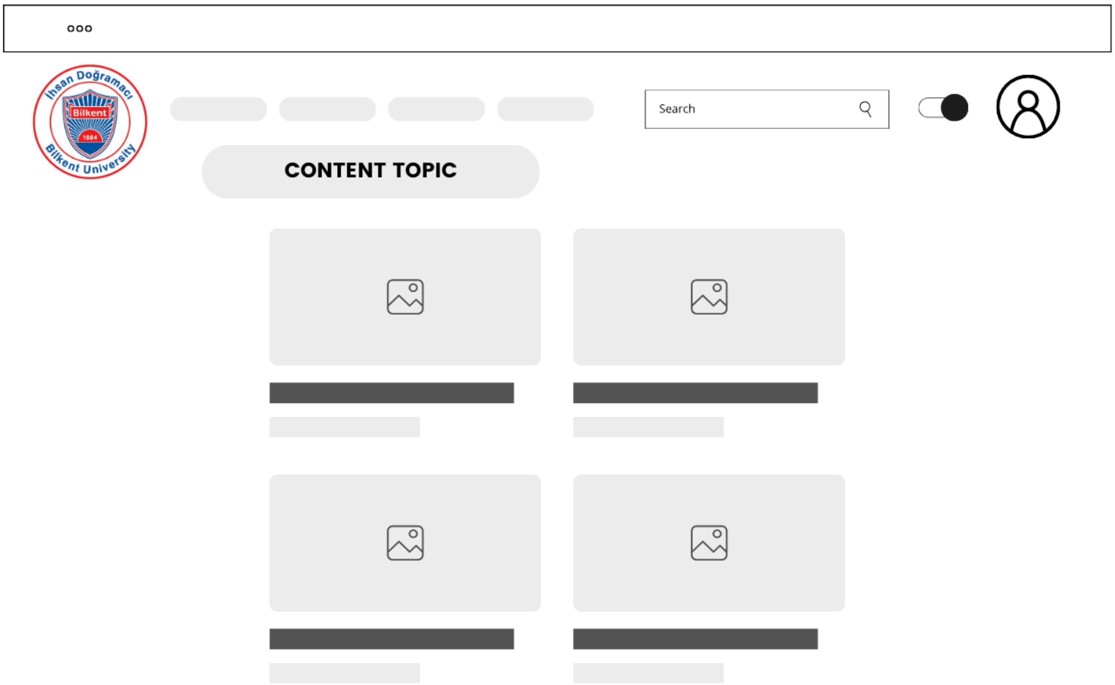
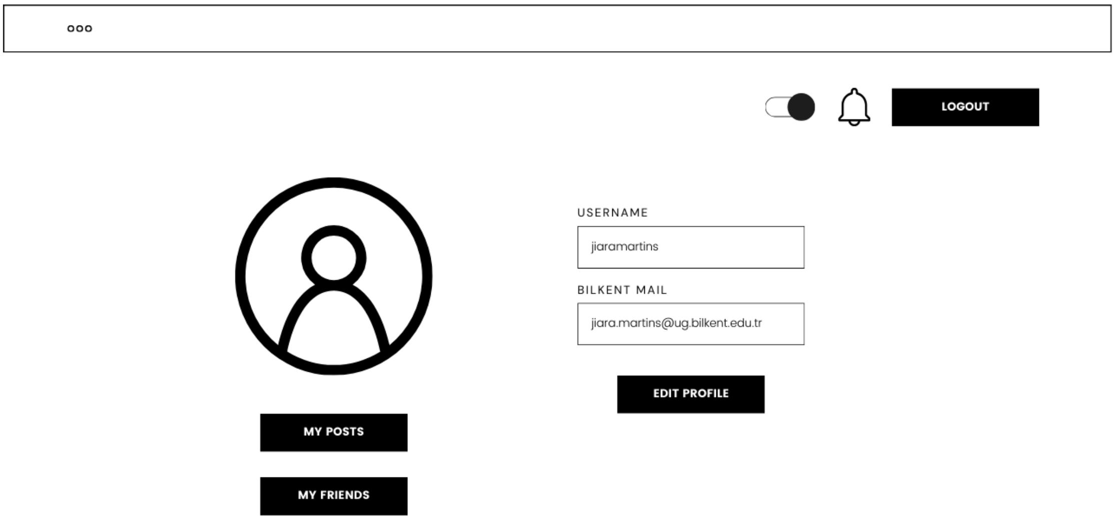


Fig. 15: Specific Content Page



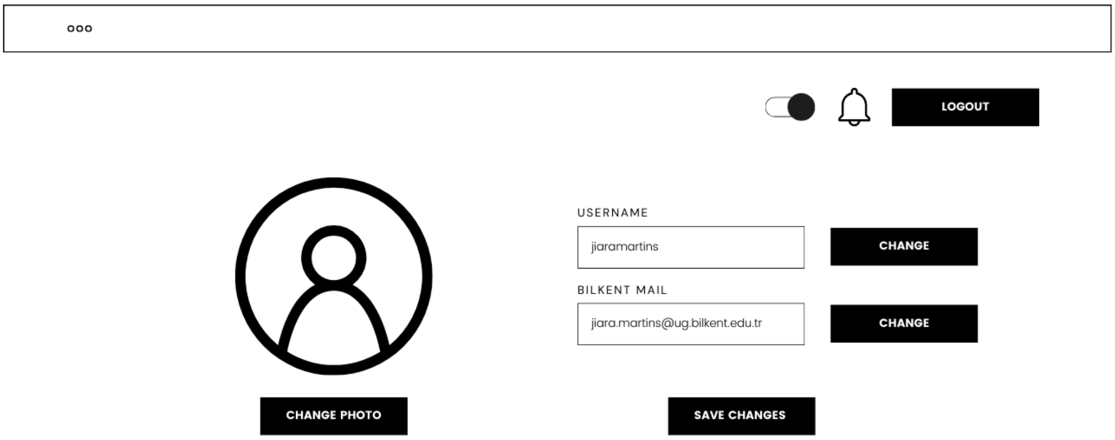
USERNAME

jiaramartins

BILKENT MAIL

jiara.martins@ug.bilkent.edu.tr

Fig. 16: Profile Page



USERNAME

jiaramartins

BILKENT MAIL

jiara.martins@ug.bilkent.edu.tr

Fig. 17: Edit Profile Page

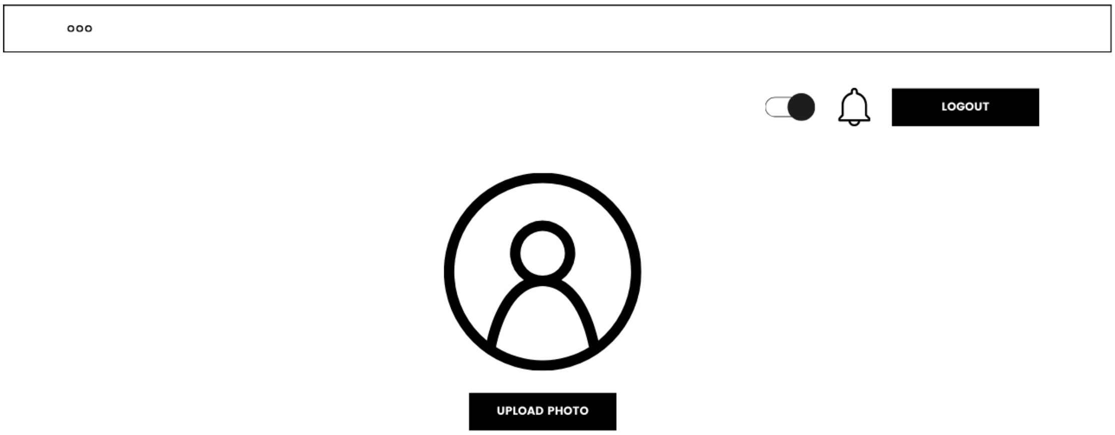


Fig. 18: Change Photo Page

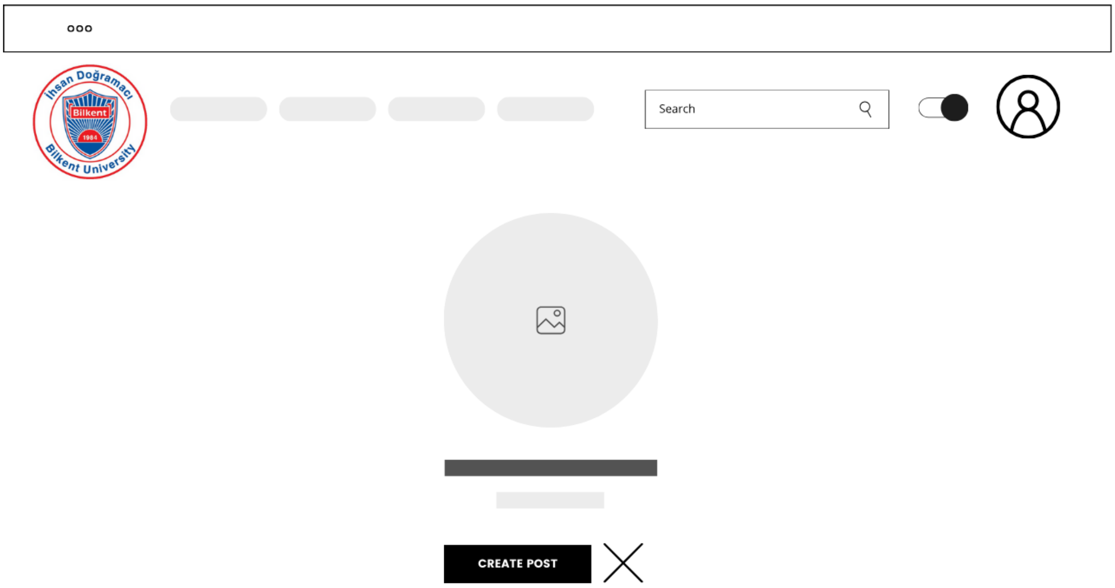


Fig. 19: Create Post Page

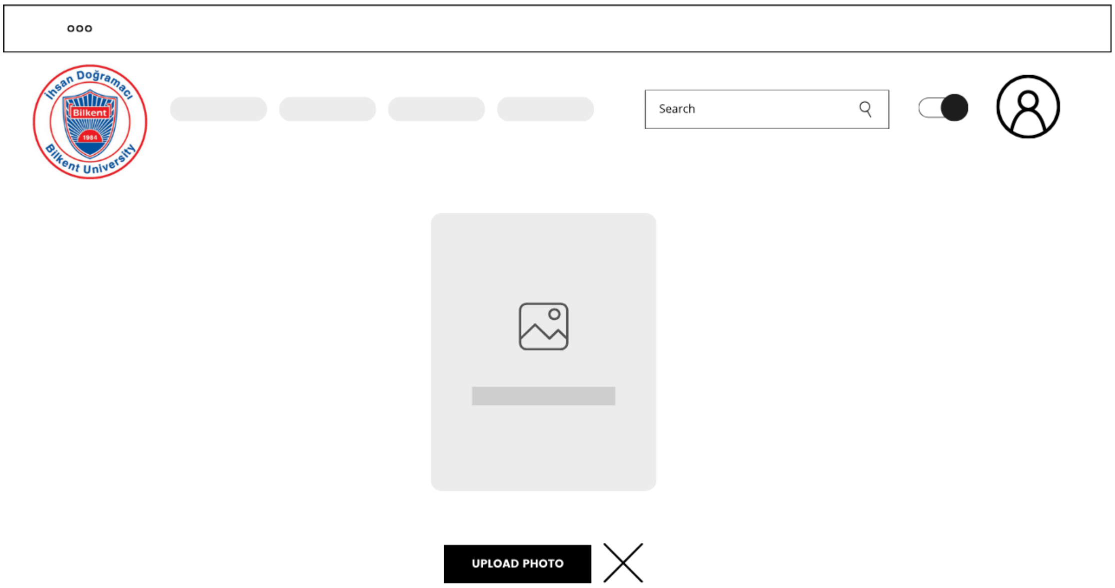


Fig. 20: Upload Post Picture Page

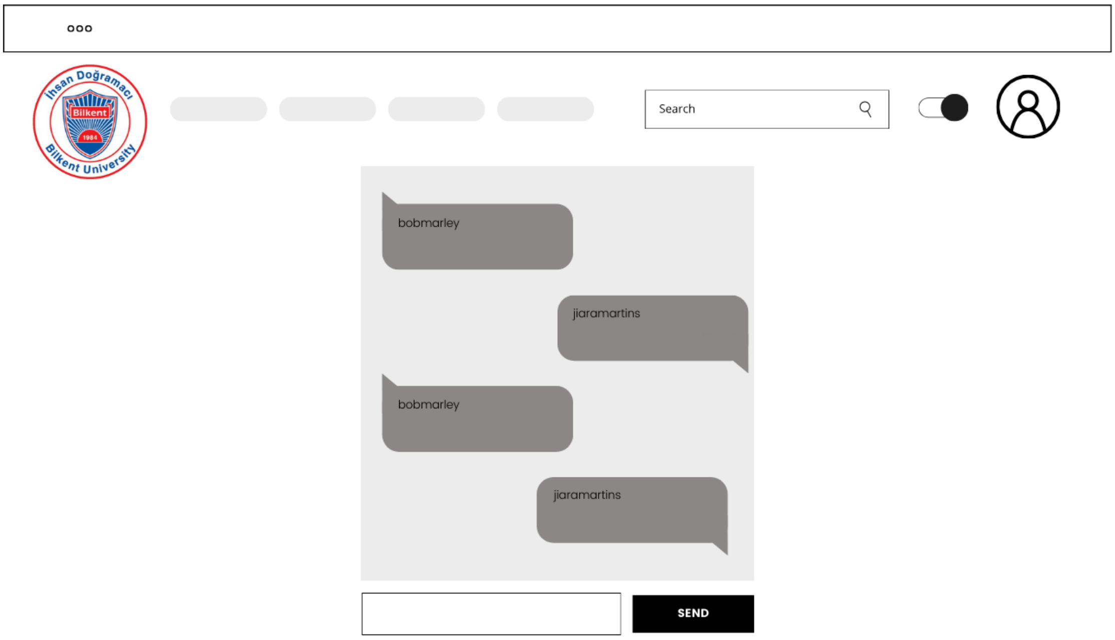
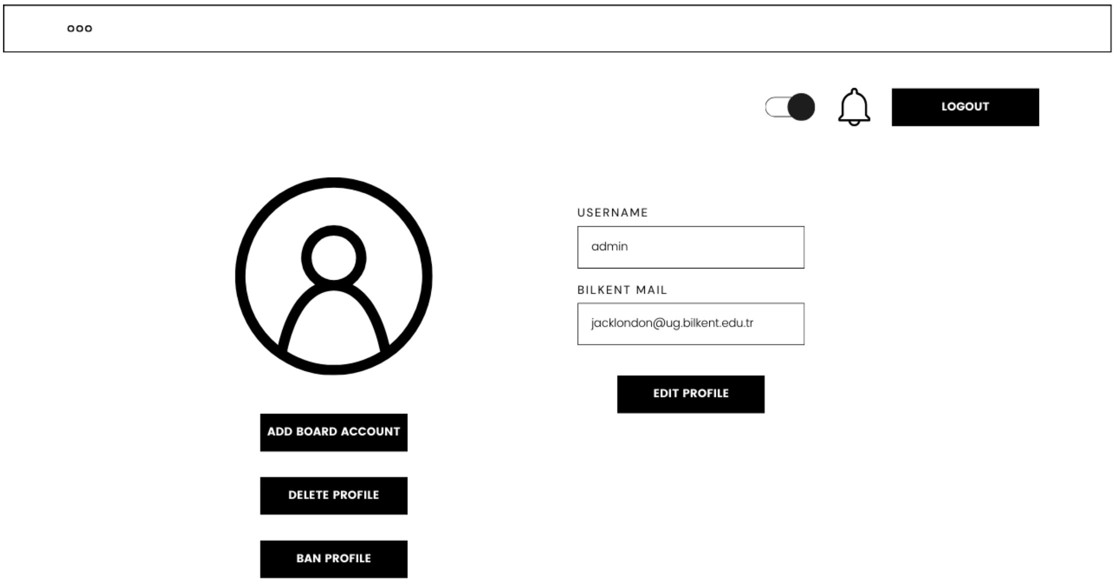


Fig. 21: Chat Page



USERNAME
admin

BILKENT MAIL
jacklondon@ug.bilkent.edu.tr

Fig. 22: Admin Profile Page

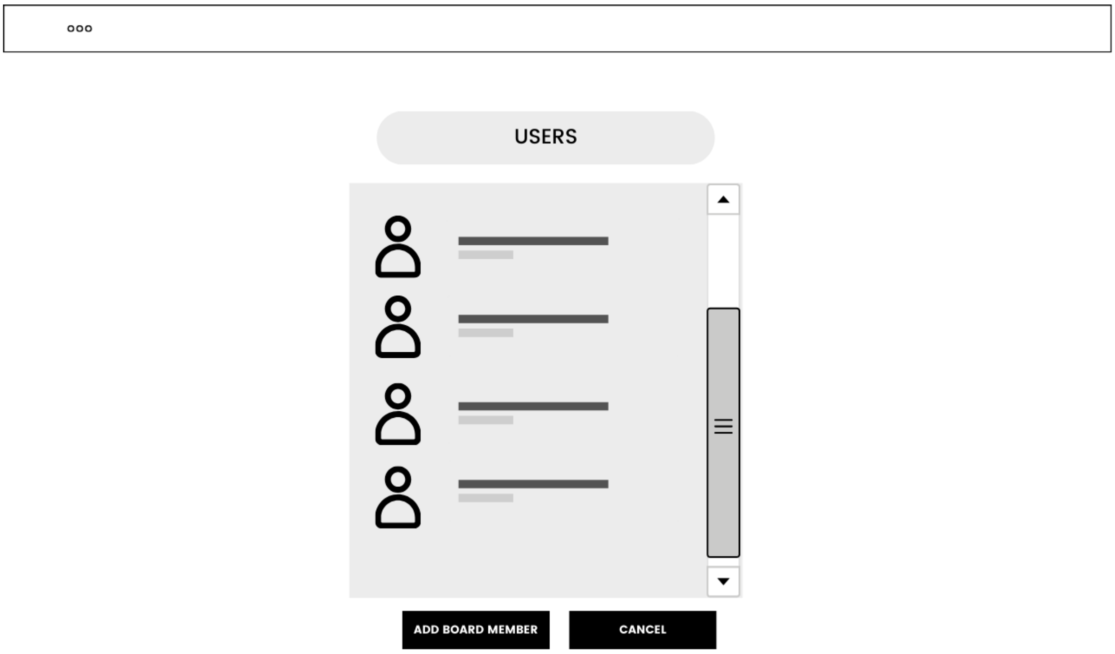


Fig. 23: Add Board Member Page

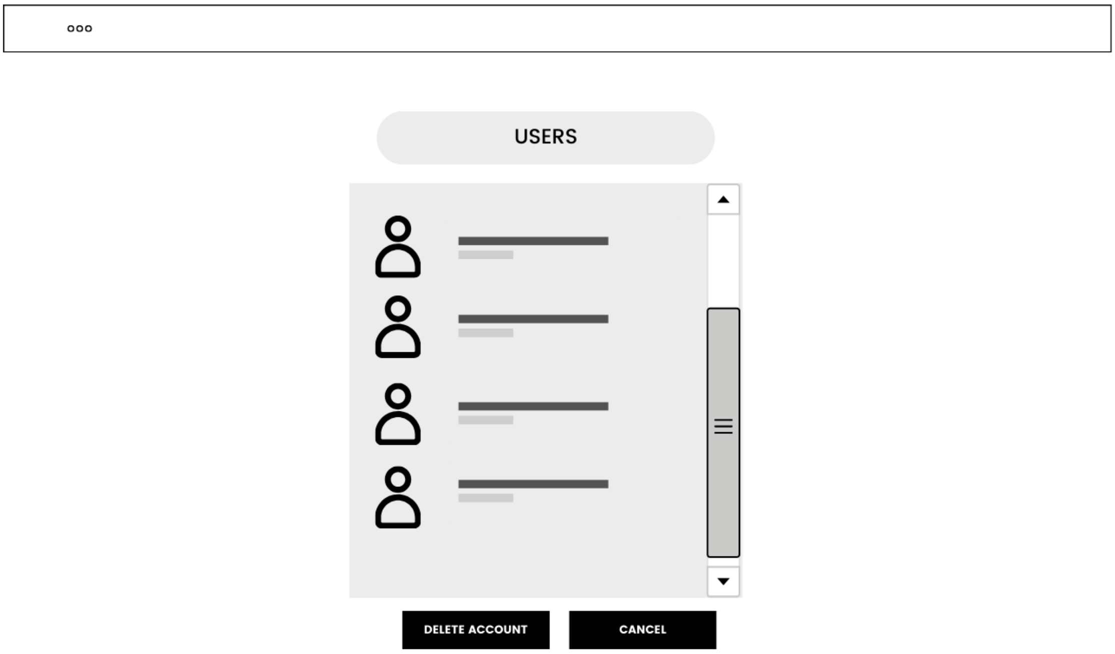


Fig. 24: Delete Account Page

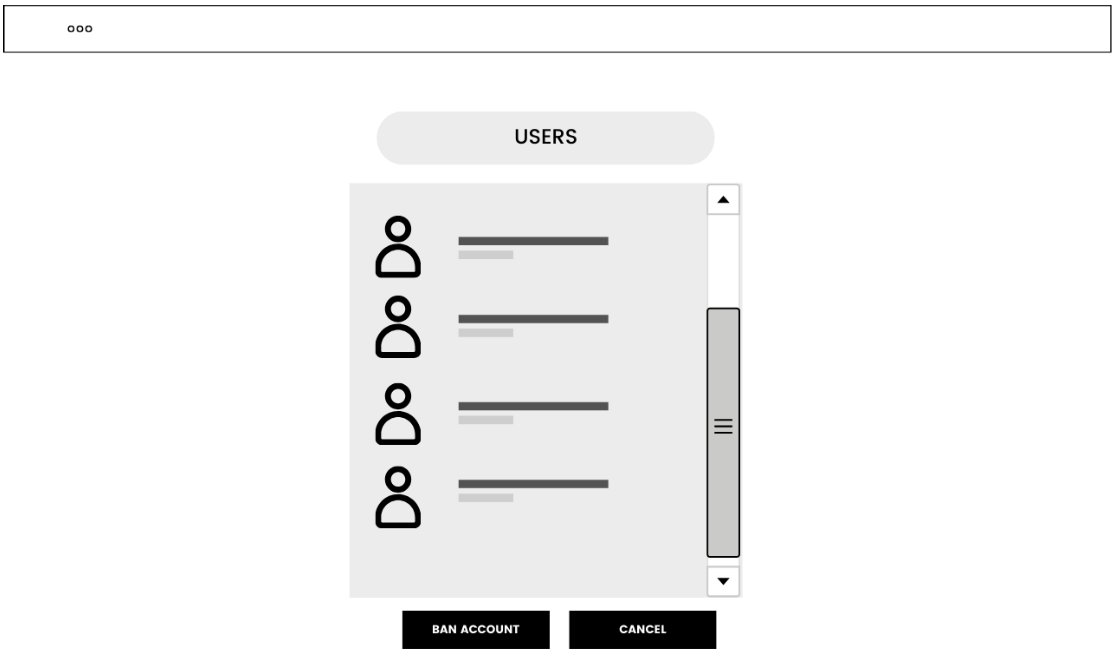


Fig. 25: Ban Account Page

3.6 Used Tech

Front-end: JavaScript

Back-end: Java

Database: 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.