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 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

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 it harder to connect with the intended audience, 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 will incorporate advanced search and filtering functions for enhanced usability. Upon registration with their Bilkent University email addresses and successful email verification, users will have access to personalized profiles. 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 users, 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 seven different types of actors:

- User: Bilkent University student who logins with their university email. After signing in they can customize their profile and add posts into categories they want. If they are interested in a post they can directly message the poster.
- Board: Representative of a Bilkent University student club. Can only be added by the admin. In order to get added a user from the club has to make a request. They cannot post on the categories or message. They can only post and edit their announcement of their club activities.
- Admin: Manages the requests for Board accounts. Can vies post. If users get reported, check the reports and ban the users accordingly. Also manages the bus schedule and cafeteria menu announcements of Bilkent University.

3.3 Non-functional Requirements

3.3.1 Performance

Throughput: Ensure that the 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 1 account associated with their university e-mail address.
- Personal privacy: The information that the user wants to share will be available for other users of the system.
- Encryption: Encryption is essential to safeguard sensitive information when it is being shared or stored.
- Access Control: Access control to restrict unauthorized access to specific areas of the platform. 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 in the process of uploading photos of the second hand 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 the OnlyBilkent, whether for educational purposes or to stay updated on campus events. Therefore, it is reasonable to minimize the data loss caused by system crashes. Using redundant database systems with replication to ensure that data remains available even if one database server fails will minimize data loss.

3.3.4 Usability

Our aim is to ease and fasten the 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 quickly navigate the OnlyBilkent. 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 with real-time events, second hand deals etc.
- Search and Navigation: Search functionality and navigation tools to help users easily 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 the incorporation of new features, it is significant to design our application with maintainability in mind.

- Modular Architecture: Modular architecture where components of the application 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 easier.

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

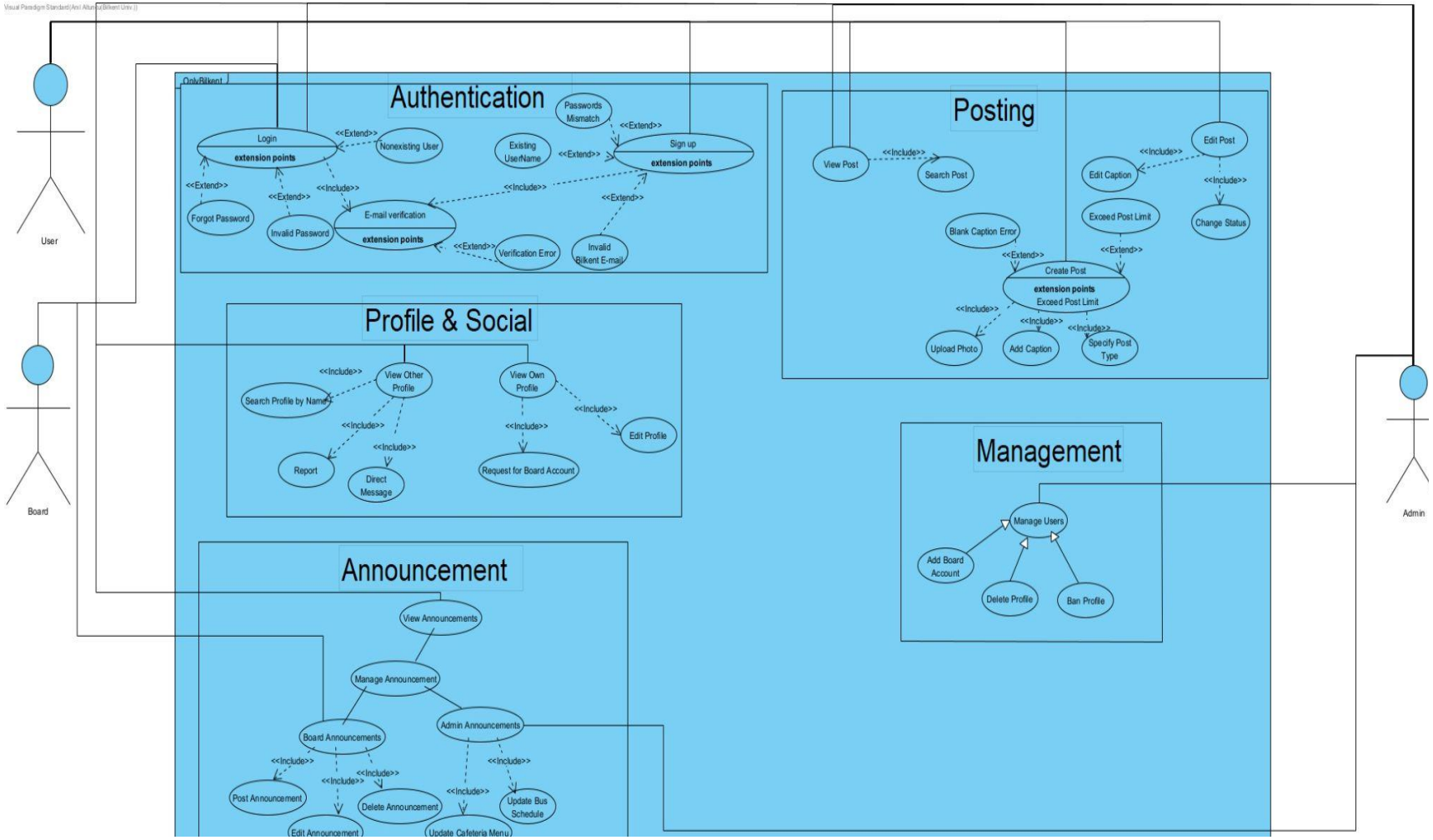

Fig. 1: Use Case Diagram for the Campuswide Connection System

Authentication Package

Use case name: Login
Participating actors: OnlyBilkent users
Flow of events:

1. The user inputs their email and password into the mail and password boxes.
2. If the user enters his/her mail incorrectly
    2.1 "Mail does not exist" error message will be displayed.
3. Else if the user enters his/her password incorrectly
    3.1 "Wrong Password" error message will be displayed.
4. Else the user successfully completes the login in process.
    4.1 Verification mail will be sent to the mail address.

Entry condition: Opening the website by using a web browser.
Exit condition: Login process fails or 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 username of the user is not unique
    2."Please enter a unique username" message will be displayed.
3. If the Bilkent mail the user entered is not valid
    3. "Please enter a valid Bilkent mail" message will be displayed.
4. If the two password boxes for verification does not match
    4.1 "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, Admin, Board
Flow of events:

1. OnlyBilkent user decides to search a user by their username by writing the username into the search bar.
2. After writing the username to the search bar, the user clicks the search button.
3. Then, the user can click the profile he/she wants to see.

Entry conditions: Searching names through 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 to 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 searches a user by their username by writing the username into the search bar.
2. User clicks the search button.
3. If no user with the provided username is found in the search results, the system triggers an error message.
    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 search bar.

Exit conditions: Navigating 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 to his/her profile page.
2. User clicks to 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: Navigating to the other page of the OnlyBilkent.

Use case name: Edit Profile

Participating actors: OnlyBilkent users

Flow of events:

1. User decides to edit the profile and enters his/her profile.
2. User clicks the “Edit” button.
3. User makes the necessary updates and clicks the “Save” button for saving the changes.

Entry conditions: User navigates to his/her profile page and clicks the "Edit" button.
Exit conditions: User clicks to the "Save" button for saving the changes of the profile or user 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 write 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 search for posts containing a specific keyword by entering the keyword in the search bar.
2. User clicks the search button, the user receives search results.
3. If no posts containing the provided keyword are found in the search results, the system triggers an error message.
    a. The error message informs the user that no posts with the specified keyword were found.
4. The user has the option to 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 that no posts with the specified keyword were found.

2. The user may choose to 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 the user clicks to the specific context area.
2. User clicks the "Create Post" button and creates his/her posts.

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:

5. The user decides to create a post and the user clicks to the specific context area.
6. User proceeds to enter a caption in the provided field.
7. User submits the caption field without entering any text.
8. The system detects that the caption field is empty or contains only white spaces.
9. 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. User continues to post content until they reach the predetermined post limit.
3. The system tracks the number of posts made by the user and enforces the post limit.
4. User attempts to post content beyond the established limit.

5. The system detects the attempt to exceed the post limit and restricts the user 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. 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. User selects a photo from their device.
3. User provides any caption related to the photo (optional).
4. 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: User clicks the Upload Photo button.
Exit conditions:

1. The photo has been successfully uploaded to the user's account.
2. 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 to.
2. User clicks on the photo to open it.
3. User locates the "Add Caption" option.
4. 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: User is logged into their account and selects a post to add a caption to.
Exit conditions:

1. The caption has been successfully added to the selected photo.
2. 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: User can navigate to dashboard or posts page.

Use case name: Manage Announcement
Participating actors:

Primary Actor: Admin

Secondary Actor: OnlyBilkent Users

Flow of events:

1. Admin logs into their account with appropriate permissions on the platform.
2. Admin navigates to the "Manage Announcements" or similar section.
3. The system provides options to create, edit, or delete announcements.
4. Admin selects one of the options based on their intent:
    a. To create a new announcement: Admin enters a title, content, and any additional details for the new announcement.
    b. To edit an existing announcement: Admin selects the announcement to be edited, modifies the content, and saves the changes.
    c. To delete an announcement: Admin 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: Admin 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 admin decides to add board account

2.1. Admin selects the user who to be the board member from the user list and clicks to “Add board member” button

3. Else if admin decides to delete user

3.1 Admin selects the user to be deleted and clicks “Delete user” button

4.Else if admin decides to ban user

4.1 Admin selects the user to be deleted and clicks “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 the users.
2. Admin selects the user they want to view details for from the Users list.
3. 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: Modify User Roles

Participating Actors: Admin

Flow of Events:

1. Admin navigates to his/her profile page and clicks the “Users” button for the list of the users.
2. Admin clicks on the user from Users list.
3. Admin modifies the role or permissions of a specific user.
4. System presents options for modifying the user's role, such as changing from a regular user to an admin.
5. Admin makes the necessary role changes and saves them.

6. The system updates the user's role accordingly.

Entry Conditions: Admin accesses their profile page and clicks the “Users” button.

Exit Conditions:

1. The Admin has successfully modified the user's roles.
2. The Admin can choose to 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 the users.
2. Admin clicks on the user from Users list.
3. The system provides an option to unblock the user.
4. Admin confirms the unblocking action.
5. 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.

3.6 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.