



Bilkent University

Department of Computer Engineering

CS 319 - Object-Oriented Software Engineering
Fall 2023

Deliverable 1

Use Case Diagram
Non-Functional Requirements
Tech Stack

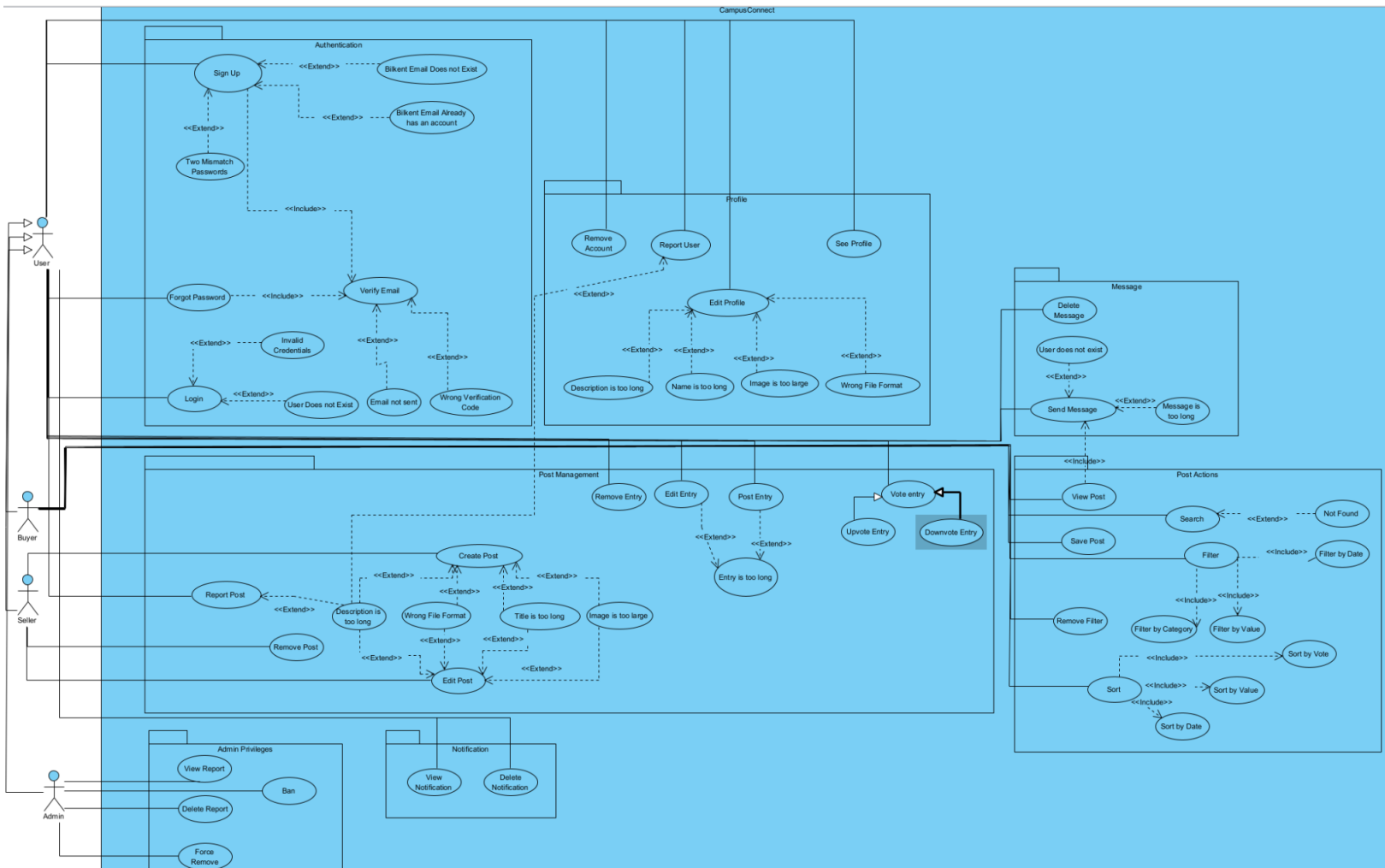
Team 5

Cahit Ediz Civan (22003206)
Efe Kaan Fidancı (22102589)
Emir Tuğlu (22003165)
Görkem Kadir Solun (22003214)
Mete Enes Yılmaz (22102849)

Table of Contents

Table of Contents	2
Use Case Diagram	3
Use Case Diagram Explanations	4
Authentication Package	4
Profile Package	5
Message Package	7
Post Management	7
Post Actions	10
Admin Privileges Package	12
Notification Package	13
Non-functional Requirements	14
Usability	14
Reliability	14
Security/Safety	14
Performance	14
Maintainability	15
Tech Stack	15

Use Case Diagram



Use Case Diagram Explanations

Authentication Package

1. **Name:** Login
 2. **Participating Actor:** User
 3. **Entry Condition:** Opening the app through a web browser
 4. **Exit Condition:** The user signs in, or the login process fails.
 5. **Flow of Events:**
 - 5.1. The user fills in the credentials and presses the login button.
 - 5.1.1. If the username/email or password does not match, the user is prompted to re-enter them.
 - 5.2. The user logs in.
-
1. **Name:** Sign Up
 2. **Participating Actor:** User
 3. **Entry Condition:** Clicking the sign-up button on the login page
 4. **Exit Condition:** Signing up, the sign-up process fails, or clicking the back button
 5. **Flow of Events:**
 - 5.1. The user opens up the sign-up page
 - 5.2. Provides the required information (email, username, password, etc.)
 - 5.3. Clicks the sign-up button
 - 5.3.1. If the user did not provide a Bilkent Email(bilkent.edu.tr), "Email should be a bilkent email" error message will be displayed.
 - 5.3.2. If the provided Bilkent Email already has an account, "Email has already an account" error message will be displayed.
 - 5.3.3. If two entered passwords are mismatched, "Passwords mismatch" error message will be displayed.
 - 5.4. The user will be asked to verify his/her email.
 - 5.5. Registration will be completed after email verification.
-
1. **Name:** Verify Email
 2. **Participating Actor:** User
 3. **Entry Condition:** Signing up or requesting a new password via forgot password page
 4. **Exit Condition:** Verifying the email or abandoning the verification page
 5. **Flow of Events:**
 - 5.1. The user either signs up or requests a new password.
 - 5.2. The user receives an email containing a verification code.

- 5.2.1. If the user did not receive the email, he/she can request a resending of the email.
- 5.3. The user enters the code into the related box on either the sign-up or forgot password pages.
 - 5.3.1. If the entered code is wrong, an error message will be displayed.
- 5.4. Verification will be completed after the user enters the correct code.

1. **Name:** Forgot Password
2. **Participating Actor:** User
3. **Entry Condition:** Clicking the forgot password button on the login page
4. **Exit Condition:** Changing the password, or abandoning the forgot password page
5. **Flow of Events:**
 - 5.1. The user opens up the forgot password page.
 - 5.2. Enters his/her email address.
 - 5.3. The user is redirected to the email verification page if an account has that email address.
 - 5.3.1. If there is no account with that email address, "There is no account with registered with that email address" error message will be displayed.
 - 5.4. The user is redirected to the password change page after completing the email verification.
 - 5.5. The user enters the new password and clicks the change password button.
 - 5.6. The user is redirected to the login page.

Profile Package

1. **Name:** Edit Profile
2. **Participating Actor:** User
3. **Entry Condition:** Clicking the edit profile button on the user's profile page.
4. **Exit Condition:** Clicking the done button or cancel button on the edit profile page
5. **Flow of Events:**
 - 5.1. The user enters the profile page.
 - 5.2. The user clicks the edit profile button.
 - 5.3. The user reenters the description.
 - 5.3.1. If the description is too long, "Description is too long" message will be displayed.
 - 5.4. The user reenters the name.

- 5.4.1. If the name is too long, "Name is too long" message will be displayed.
- 5.5. The user uploads a new profile picture.
 - 5.5.1. If the image is too large, "Image is too large" message will be displayed.
 - 5.5.2. If the file is in the wrong file format, "Wrong File Format" message will be displayed.
- 5.6. The user will be able to change the profile.

- 1. **Name:** See Profile
- 2. **Participating Actor:** User
- 3. **Entry Condition:** Opening up a post and clicking the seller.
- 4. **Exit Condition:** Exiting through a back button from a profile.
- 5. **Flow of Events:**
 - 5.1. Clicking the username or profile picture of a seller.
 - 5.2. The buyer will be directed to the seller's profile.

- 1. **Name:** Remove Account
- 2. **Participating Actor:** User
- 3. **Entry Condition:** Clicking the remove account button on the profile page
- 4. **Exit Condition:** Clicking the I am sure button or cancel button when prompted with the warning "Are you sure you want to remove your account?"
- 5. **Flow of Events:**
 - 5.1. The user enters the profile page.
 - 5.2. The user clicks the edit profile button.
 - 5.3. The user opens the edit page.
 - 5.4. The user clicks the remove account button.
 - 5.5. The user accepts the "I am sure" button when prompted with the warning, "Are you sure you want to remove your account?"
 - 5.6. The user is directed to the login page.

- 1. **Name:** Report User
- 2. **Participating Actor:** User
- 3. **Entry Condition:** The user opens the user's profile and clicks the report user button.
- 4. **Exit Condition:** Reporting the user or clicking the cancel button.
- 5. **Flow of Events:**
 - 5.1. The user clicks the report user button.
 - 5.2. The user fills up the report user form.

- 5.2.1. If the description is too long, “Description is too long” error message is displayed.
- 5.3. The user successfully reports the user.

Message Package

1. **Name:** Send Message
2. **Participating Actor:** User
3. **Entry Condition:** The user opens up the messages window and selects a conversation
4. **Exit Condition:** The message is successfully sent or cannot be sent.
5. **Flow of Events:**
 - 5.1. The user writes the message content to the given space.
 - 5.2. The user pushes the send message button.
 - 5.2.1. If the user has removed his/her account, “User does not exist” message will be displayed.
 - 5.2.2. If the message is too long, “Message is too long” message will be displayed.
 - 5.3. The message is sent, and the message will be shown.
1. **Name:** Delete Message
2. **Participating Actor:** User
3. **Entry Condition:** The user opens up the messages window and selects a conversation
4. **Exit Condition:** The user deletes the message
5. **Flow of Events:**
 - 5.1. The user opens the message history
 - 5.2. The user who has sent the message clicks the delete message.
 - 5.3. The message will be deleted.

Post Management

1. **Name:** Create Post
2. **Participating Actor:** Seller
3. **Entry Condition:** Opening up related(secondhand, borrow, etc.) page and clicking the create post button.
4. **Exit Condition:** Creating the post or closing the create post window.
5. **Flow of Events:**
 - 5.1. The seller clicks the related page’s create post button.
 - 5.2. The seller fills up the create post form.

- 5.2.1. If the title is too long, "Title is too long" error message is displayed.
- 5.2.2. If uploaded files are in the wrong file format, "Wrong file format" error message is displayed.
- 5.2.3. If the description is too long, "Description is too long" error message is displayed.
- 5.2.4. If images are too large, "Images are too large" error message is displayed.
- 5.3. The seller successfully creates the post.

1. **Name:** Remove Post
2. **Participating Actor:** Seller
3. **Entry Condition:** The seller opens the related(secondhand, borrow, etc.) page and clicks his/her post.
4. **Exit Condition:** Clicking the I am sure button(removing the post) or cancel button when prompted with the warning "Are you sure you want to remove your post?"
5. **Flow of Events:**
 - 5.1. The seller enters his/her post.
 - 5.2. The seller clicks the remove post button, only visible to the post's creator.
 - 5.3. The user successfully removes his/her post from the network.
 - 5.4. The user is prompted with a "You successfully removed your post" message.
 - 5.5. The user is directed to the related(secondhand, borrow, etc.) page.

1. **Name:** Edit Post
2. **Participating Actor:** Seller
3. **Entry Condition:** The user opens up a post belonging to himself/herself and clicks the edit post button.
4. **Exit Condition:** Editing the post or clicking the cancel button
5. **Flow of Events:**
 - 5.1. The seller opens up the edit post window
 - 5.2. The seller fills up the edit post form.
 - 5.2.1. If the title is too long, "Title is too long" error message is displayed.
 - 5.2.2. If uploaded files are in the wrong file format, "Wrong file format" error message is displayed.
 - 5.2.3. If the description is too long, "Description is too long" error message is displayed.
 - 5.2.4. If images are too large, "Images are too large" error message is displayed
 - 5.3. The seller successfully edits the post.

1. **Name:** Report Post
2. **Participating Actor:** User
3. **Entry Condition:** The user opens a post and clicks the report post button.
4. **Exit Condition:** Reporting the post or clicking the cancel button.
5. **Flow of Events:**
 - 5.1. The user clicks the report post button.
 - 5.2. The user fills up the report post form.
 - 5.2.1. If the description is too long, "Description is too long" error message is displayed.
 - 5.3. The user successfully reports the post.

1. **Name:** Vote Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Clicking the up/downvote button again.
5. **Flow of Events:**
 - 5.1. The user clicks the up/downvote entry button.
 - 5.2. The user successfully up/downvotes the post.

1. **Name:** Upvote Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Clicking the upvote button again.
5. **Flow of Events:**
 - 5.1. The user clicks the upvote entry button.
 - 5.2. The user successfully upvotes the entry.

1. **Name:** Downvote Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Clicking the downvote button again.
5. **Flow of Events:**
 - 5.1. The user clicks the downvote entry button.
 - 5.2. The user successfully downvotes the entry.

1. **Name:** Post Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Posting the entry or clicking the cancel button.
5. **Flow of Events:**

- 5.1. The user writes his/her entry.
 - 5.1.1. If the entry is too long, "Entry is too long" error message is displayed.
- 5.2. The user successfully posts the entry by clicking the send button.

1. **Name:** Edit Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Editing the entry or clicking the cancel button.
5. **Flow of Events:**
 - 5.1. The user edits their entry.
 - 5.1.1. If the entry is too long, an "Entry is too long" error message is displayed.
 - 5.2. The user successfully edits the entry by clicking the finish button.

1. **Name:** Remove Entry
2. **Participating Actor:** User
3. **Entry Condition:** Opening up a post on the forum page.
4. **Exit Condition:** Clicking the I am sure button(removing the entry) or cancel button when prompted with the warning "Are you sure you want to remove your entry?"
5. **Flow of Events:**
 - 5.1. The user opens up his/her entry
 - 5.2. The user successfully removes the entry by clicking the remove entry button that becomes visible when clicking the entry.

Post Actions

1. **Name:** Search
2. **Participating Actor:** Buyer
3. **Entry Condition:** Clicking the search bar on the related pages(secondhand, borrow, etc.).
4. **Exit Condition:** Choosing a tab from the navigation bar or viewing a post.
5. **Flow of Events:**
 - 5.1. The buyer types the thing he/she wants to find in the search bar.
 - 5.2. The buyer clicks the search button.
 - 5.3. Matching posts are fetched.
 - 5.3.1. The "Not Found" message is displayed if there are no matching results.

1. **Name:** View Post
2. **Participating Actor:** Buyer
3. **Entry Condition:** Opening up a post.
4. **Exit Condition:** Exiting through a back button from a post.
5. **Flow of Events:**
 - 5.1. Clicks the details button of a post
 - 5.2. If the buyer wants to interact/buy with the post user, he/she can send a message.

1. **Name:** Save Post
2. **Participating Actor:** Buyer
3. **Entry Condition:** Opening up a post and clicking the save button of a post.
4. **Exit Condition:** Clicking the unsave button of a post which replaced the save button.
5. **Flow of Events:**
 - 5.1. Clicks the save button of a post.

1. **Name:** Filter
2. **Participating Actor:** Buyer
3. **Entry Condition:** Clicking the filter section on the left of the related pages(secondhand, borrow, etc.).
4. **Exit Condition:** Clicking the remove filter button.
5. **Flow of Events:**
 - 5.1. The buyer clicks the filter section.
 - 5.1.1. The buyer filters by category.
 - 5.1.2. The buyer filters by date.
 - 5.1.3. The buyer filters by price.
 - 5.2. The buyer clicks the apply button.
 - 5.3. Matching posts are fetched.

1. **Name:** Remove Filter
2. **Participating Actor:** Buyer
3. **Entry Condition:** Clicking the filter section on the left of the related pages(secondhand, borrow, etc.).
4. **Exit Condition:** -
5. **Flow of Events:**
 - 5.1. The buyer clicks the remove filter button.
 - 5.2. The buyer applies.
 - 5.3. Matching posts are fetched.

1. **Name:** Sort
2. **Participating Actor:** Buyer
3. **Entry Condition:** Clicking the sort dropdown menu on the top of the related pages(secondhand, borrow, etc.).
4. **Exit Condition:** Clicking the applied sort.
5. **Flow of Events:**
 - 5.1. The buyer clicks the sort dropdown menu.
 - 5.1.1. The buyer sorts by vote.
 - 5.1.2. The buyer sorts by date.
 - 5.1.3. The buyer sorts by price.
 - 5.2. The sort will be applied, and matching posts will be fetched.

Admin Privileges Package

1. **Name:** View Report
 2. **Participating Actor:** Admin
 3. **Entry Condition:** Clicking the existing reports button on the admin panel and selecting a report.
 4. **Exit Condition:** Exiting through a back button from a report.
 5. **Flow of Events:**
 - 5.1. The admin clicks the existing reports button.
 - 5.2. The admin opens up a report.
-
1. **Name:** Delete Report
 2. **Participating Actor:** Admin
 3. **Entry Condition:** Clicking the existing reports button on the admin panel and selecting a report.
 4. **Exit Condition:** Deleting the report.
 5. **Flow of Events:**
 - 5.1. The admin clicks the existing reports button.
 - 5.2. The admin opens up a report.
 - 5.3. The admin clicks the delete button.
-
1. **Name:** Force Remove
 2. **Participating Actor:** Admin
 3. **Entry Condition:** Opening up a post.
 4. **Exit Condition:** Removing the post or exiting through a back button from a post.
 5. **Flow of Events:**
 - 5.1. Press the remove post button (only visible to the admin) alongside a post in the flow.

1. **Name:** Ban
2. **Participating Actor:** Admin
3. **Entry Condition:** Admin must look at a user profile and press the ban button.
4. **Exit Condition:** Banning the user or clicking the cancel button.
5. **Flow of Events:**
 - 5.1. The admin goes to the profile page of a user.
 - 5.2. The admin clicks the ban user button on the right side of the user's profile photo.

Notification Package

1. **Name:** View Notification
 2. **Participating Actor:** User
 3. **Entry Condition:** Opening up the notifications window.
 4. **Exit Condition:** -
 5. **Flow of Events:**
 - 5.1 Notification from the app signals the user to check for updates.
 - 5.2 The user opens the notification content.
-
1. **Name:** Delete Notification
 2. **Participating Actor:** User, Admin
 3. **Entry Condition:** Clicking delete notification button.
 4. **Exit Condition:** -
 5. **Flow of Events:**
 - 5.1 The user opens up the notifications window.
 - 5.2 The user clicks the delete notification button.

Non-functional Requirements

Usability

- Bilkent Exchange Network must have an intuitive and user-friendly interface, making it easy for users and administrators to use the app without extensive effort. We will design an interface similar to already known social media applications (such as FaceBook, WhatsApp, Instagram, etc.) to give our application a user-friendly factor.
- The application should be accessible to all users by complying with accessibility standards.
- Bilkent Exchange Network should have a responsive design to support different devices and browsers. The elements on the screen should be resized according to the user's screen size.
- Bilkent Exchange Network should have concise and clear user documentation for UI elements. With this, it will be ensured that messages are informative enough, and guides will be able to guide users toward resolution.

Reliability

- Bilkent Exchange Network should be available with at least a 95% uptime rate, ensuring users can access it whenever they want.
- The application should handle errors gracefully. It should provide informative and precise error messages to prevent bad user experience.

Security/Safety

- Bilkent Exchange Network should include robust user authentication and authorization mechanisms to ensure only authorized users can access and modify data using strong password policies and multi-factor authentications.
- Data stored in the database, such as user accounts and posts, must be reliably stored and protected by access controls and data encryption.
- Sensitive data, such as user passwords, must be encrypted with the SHA-256 algorithm before being stored in the database.
- Bilkent Exchange Network should respect and protect user information by sharing only the necessary information with other users.

Performance

- The application should respond to the requests coming from the users quickly(<1s) to provide a good user experience.
- The verification email should be sent to users within 1 minute after the request.
- Navigation among different pages should take less than 3 seconds.

Maintainability

- Bilkent Exchange Network's code should be well-structured according to object-oriented design principles and documented to support future updates.

Tech Stack

- **TypeScript**
 - A widely-used programming language for web development.
 - A statically typed superset of JavaScript that helps catch errors at compile-time and enhances code quality.
 - Provides improved code maintainability, better support for tools, and a more robust development experience than plain JavaScript.
 - It will be used for both client-side and server-side development.
- **React**
 - A popular JavaScript library for building user interfaces.
 - Allows for the creation of dynamic web applications.
 - We will use it to create a user interface and deliver an exceptional user experience.
- **Node.js**
 - An open-source, server-side runtime environment for executing JavaScript code.
 - Ideal for building scalable and high-performance back-end applications.
 - It will be used to run TypeScript code on the back end.
- **Next.js**
 - A framework for building React applications with server-side rendering and routing.
 - It helps improve performance.
 - Next.js will be our full-stack framework for building this application.
- **MongoDB**
 - A NoSQL database that stores data in JSON format.
 - Suitable for handling large amounts of unstructured data.
 - We chose MongoDB as the database management system since the posts in our application will contain a lot of unstructured data, which requires a NoSQL database like MongoDB.