CS319 – Object-Oriented Software Engineering

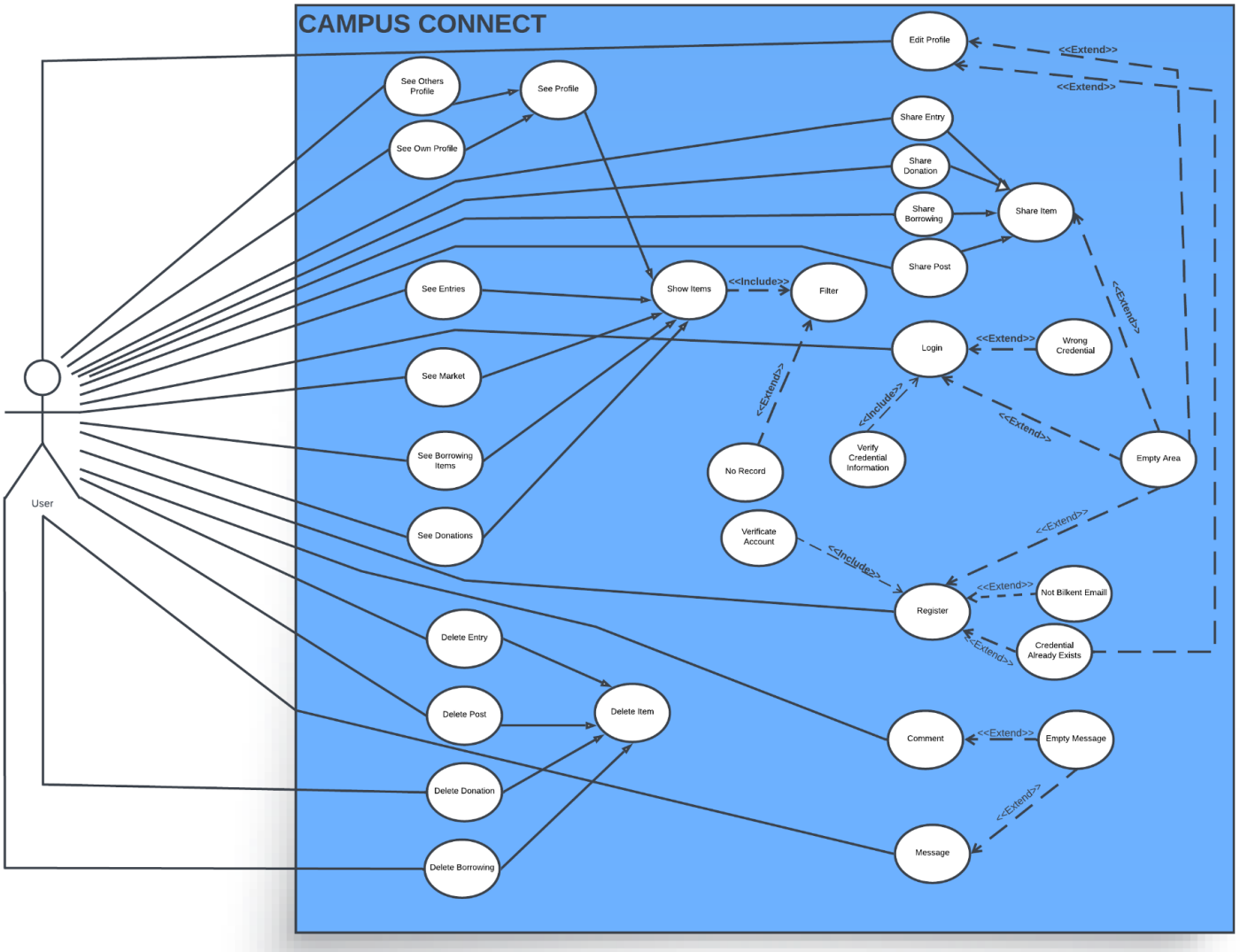
D1 Report

Section 3

Team 4 – FACID

Çağatay **AKPINAR**	**22003508**
İlhami **ULUĞTÜRKKAN**	**22102546**
Alphan **TULUKCU**	**22003500**
Feza Emir **ÇELİK**	**22101910**
Deniz Can **ÖZDEMİR**	**22003854**

1.0 Use Case Diagram



Use Case Name: Login
Participating Actor: User
Flow Of Events:

- User enters username and password.
- If one of the parts is empty
  - “Please fill this area” message pops up and the process continues until the user fills it.
- If the username or the password is not found in database
  - “Check your username and password.” message occurs and the login process restarts.
- If both username and password is found and correct, the user logs in successfully.

Entry Conditions: Opening the app through the internet.
Exit Conditions: User logs in or the login process fails.

Use Case Name: Register

Participating Actor: User
Flow Of Events:
- User fills the username, name, surname, mail and password parts.
- If one of the parts is empty
  - “Please fill this area” message pops up and the process continues until the user fills it.
- If the user does not use his/her bilkent mail
  - “Please Enter Your Bilkent Mail” message pops up and the registration process restarts.
- If all the information is proper
  - a verification mail goes to the relevant mail address.

Entry Conditions: User clicks the button “Register” on the “Log-In” page.
Exit Conditions: User clicks the “Back to Log In” button or register process fails.

Use Case Name: Verificate Account
Participating Actor: User, Database
Flow Of Events:
- Database system sends a verification code to the Bilkent mail of the user.
- User enter this verification code to the verification page of CampusConnect
- If entered, the verification code is matched with the verification code in mail.
  - Account is successfully verified
- If entered, the verification code is matched with the verification code in mail.
  - Account is not successfully verified and an error pop-up appears on screen.

Entry Conditions: User clicks the button “Verificate Account” on the “Main” page.
Exit Conditions: User clicks the “Enter code” button correctly or enters the wrong code.

Use Case Name: See Own Profile
Participating Actor: Actor
Flow Of Events:
- If a user wants to see its accounts properties, he/she clicks the “Profile” button on the menu.
- All of his/her shared posts and entries show up on the page.
  - If the user does not have any shared entries or posts
    - “No entries” and “No posts” messages occur.

Entry Conditions: User clicks “Profile” button on the menu.
Exit Conditions: User clicks another page's button.

Use Case Name: See Others’ Profile
Participating Actor: User
Flow Of Events:
- User sees wanted user’s shared items and account properties.

Entry Conditions: User clicks to another user's name through its shared post's or entry's detailed page.
Exit Conditions: User clicks another page's button.

Use Case Name: Edit Profile
Participating Actor: User
Flow Of Events:

- User edits his/her profile's properties.
- If a property is not filled
  - "Please fill this area" message pops up and the editing process continues.
- If changed username already exists
  - "Username already exists" message pops up and the editing process continues.
- If all the parts filled
  - "Please fill this area" message pops up and the editing process continues.

Entry Conditions: User clicks the "Edit Profile" button on the "Profile" page.
Exit Conditions:

1. If the editing process ends successfully and the user returns to the "Profile" page.
2. If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: Share Entry
Participating Actor: User
Flow Of Events:

- Click the "Add" button on the main page.
- Select entry from "Entry" button
- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture) for new entry.
- If one of the mandatory areas is empty.
  - "Please fill all mandatory areas" message pop-up and the entry process continues.
- If all the parts filled properly
  - Entry is successfully shared.

Entry Conditions: User clicks the "Add" button on the "Main" page.
Exit Conditions:
1.If the sharing entry process ends successfully and the user returns to the "Profile" page.
2.If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: Share Post
Participating Actor: User
Flow Of Events:

- Click the "Add" button on the main page.
- Select post from "Post" button

- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture, Price) for new entry.
- If one of the mandatory areas is empty.
  - “Please fill all mandatory areas” message pop-up and the entry process continues.
- If all the parts filled properly
  - Post is successfully shared.

Entry Conditions: User clicks the “Add” button on the “Main” page.
Exit Conditions:
1.If the sharing post process ends successfully and the user returns to the “Profile” page.
2.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Share Borrowing
Participating Actor: User
Flow Of Events:
- Click the “Borrowing” button on the main page.
- Click the “Add Borrowing” button on “Borrowing” page.
- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture) for new borrowing.
- If one of the mandatory areas is empty.
  - “Please fill all mandatory areas” message pop-up and the share borrowing process continues.
- If all the parts filled properly
  - Borrowing is successfully shared.

Entry Conditions: User clicks the “Borrowing” button on the “Main” page.
Exit Conditions:
1.If the sharing borrowing process ends successfully and the user returns to the “Profile” page.
2.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Share Donation
Participating Actor: User
Flow Of Events:
- Click the “Donation” button on the main page.
- Click the “Add Donation” button on “Donation” page.
- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture) for new donation.
- If one of the mandatory areas is empty.
  - “Please fill all mandatory areas” message pop-up and the share donation process continues.
- If all the parts filled properly
  - Donation is successfully shared.

Entry Conditions: User clicks the “Donation” button on the “Main” page.
Exit Conditions:
1.If the sharing donation process ends successfully and the user returns to the “Profile” page.

2.If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: See Entries
Participating Actor: User
Flow Of Events:

- Click the "Form" button on the main page.
- View all entries from the "Forum" page.
- User filter entries according to category and subcategory
- If there is entries which is proper to filters
  - User view entries according to his/her filters
- If there is no entries which is proper to filters
  - User view empty forum page.

Entry Conditions: User clicks the "Forum" button on the "Main" page.
Exit Conditions:
1.If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: See Market
Participating Actor: User
Flow Of Events:

- Click the "Market" button on the main page.
- View all posts from the "Market" page.
- User filter posts according to category and subcategory
- If there is posts which is proper to filters
  - User view posts according to his/her filters
- If there is no posts which is proper to filters
  - User view empty market page.

Entry Conditions: User clicks the "Market" button on the "Main" page.
Exit Conditions:
1.If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: See Borrowing Items
Participating Actor: User
Flow Of Events:

- Click the "Borrow" button on the main page.
- Click the "All Borrowings" button from the "Borrow" page.
- View all borrowing items from the "All Borrowings" page.
- User filter borrowings according to category and subcategory
- If there is borrowing which is proper to filters
  - User view borrowings according to his/her filters
- If there is no borrowing which is proper to filters
  - User view empty "All Borrowings" page.

Entry Conditions: User clicks the “Borrow” button on the “Main” page.
Exit Conditions:
1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: See Donations
Participating Actor: User
Flow Of Events:
- Click the “Donation” button on the main page.
- Click the “All Donations” button from the “Donation” page.
- View all donations from the “All Donations” page.
- User filter donations according to category and subcategory
- If there is donations which is proper to filters
  - User view donations according to his/her filters
- If there is no donation which is proper to filters
  - User view empty “All Donations” page.

Entry Conditions: User clicks the “Donation” button on the “Main” page.
Exit Conditions:
1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Comment
Participating Actor: User
Flow Of Events:
- Click the “Forum” button on the main page.
- View all entries from the “Forum” page.
- Click one of the entries from the “Forum” page.
- Write a comment to the comment area below the entry.
- Click the “Comment” button and send a comment.

Entry Conditions: User clicks the “Forum” button on the “Main” page.
Exit Conditions:
1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.
2.If the user comments successfully and directed “Forum” page again.

Use Case Name: Message
Participating Actor: User
Flow Of Events:
- Click the “Market” button on the main page.
- View all posts from the “Market” page.
- Click one of the posts from the “Forum” page.
- Click the “Message” button below the post.
- Directed to the “Chat” page and started messaging.

Entry Conditions: User clicks the “Market” button on the “Main” page.

Exit Conditions:
1.If the user clicks to another page's button on the menu, the user goes to the relevant page.

Use Case Name: Delete Entry
Participating Actor: User
Flow Of Events:
- Click the "Profile" button on the main page.
- User views his/her profile information from the "Profile" page.
- Clicks his/her entries from the "Profile" page.
- Click the "Delete" button on the "Entry" page.
- Entry is successfully deleted.

Entry Conditions: User clicks the "Profile" button on the "Main" page.
Exit Conditions:
1.If the user clicks the "Delete" button on the "Entry" page.

Use Case Name: Delete Post
Participating Actor: User
Flow Of Events:
- Click the "Profile" button on the main page.
- User views his/her profile information from the "Profile" page.
- Clicks his/her posts from the "Profile" page.
- Click the "Delete" button on the "Post" page.
- Post is successfully deleted.

Entry Conditions: User clicks the "Profile" button on the "Main" page.
Exit Conditions:
1.If the user clicks the "Delete" button on the "Post" page.

Use Case Name: Delete Donation
Participating Actor: User
Flow Of Events:
- Click the "Profile" button on the main page.
- User views his/her profile information from the "Profile" page.
- Clicks his/her donations from the "Profile" page.
- Click the "Delete" button on the "Donation" page.
- Donation is successfully deleted.

Entry Conditions: User clicks the "Profile" button on the "Main" page.
Exit Conditions:
1.If the user clicks the "Delete" button on the "Donation" page.

Use Case Name: Delete Borrowing
Participating Actor: User
Flow Of Events:

- Click the “Profile” button on the main page.
- User views his/her profile information from the “Profile” page.
- Clicks his/her borrowings from the “Profile” page.
- Click the “Delete” button on the “Borrowing” page.
- Borrowing is successfully deleted.

Entry Conditions: User clicks the “Profile” button on the “Main” page.
Exit Conditions:
1.If the user clicks the “Delete” button on the “Donation” page.

2.0 Non-Functional Requirements

2.1 Performance

Performance is the heartbeat of CampusConnect, with every aspect meticulously tuned to ensure lightning-fast interactions. Our login and logout processes are designed to take less than 2 seconds, allowing users to access their accounts swiftly. Navigating between pages is a seamless experience, taking less than 2 seconds to transition effortlessly. Whether you're uploading a post or entry with images, we prioritize speed – pictures from our database to your screen in under 5 seconds and post creation in less than 2 seconds, even with photographs. In the background, the entire system data is dynamically backed up to ensure data integrity and reliability. The importance of such performance measures cannot be overstated. A rapid, responsive system enhances user satisfaction, engagement, and the overall success of our platform. With CampusConnect, every second counts in making your online campus experience exceptional. These performance measures are vital to ensure that users have a smooth and efficient experience when using the CampusConnect platform. The speed of these actions directly impacts user engagement and the overall functionality of the website. Rapid performance is key to user satisfaction and success in providing an exceptional online campus experience.

2.2 Safety/Security

At CampusConnect, security and safety are paramount. We've implemented stringent measures to safeguard user data, ensuring that personal information is encrypted and stored securely within our database. In CampusConnect password of users stored in database as hashed form so even admins cannot reach the passwords of users. Additionally, the app boasts robust authentication and authorization mechanisms, limiting access exclusively to individuals within the Bilkent University community. Account verification feature of our app provides the chance of creating an in-campus community in the CampusConnect app. The importance of these security measures cannot be overstated. They are the foundation of user trust and confidence in our platform. As guardians of sensitive data, we prioritize the protection of personal information and access controls to ensure that CampusConnect remains a safe and secure digital space for our users, fostering trust, peace of mind, and a thriving online community.

2.3 Usability

Usability is a cornerstone of CampusConnect's design philosophy, and it holds immense importance in delivering an exceptional user experience. Our user interface is meticulously crafted to be intuitive, accessible, and responsive, catering to users of different abilities. We have focused on creating an easy-to use and simple user interface that ensures a seamless experience. Places of all buttons are on the top menu and easy to find&use for all users. Also while adding post, entry, borrowing and donation; blanks which must be filled are located at the right of page. The app's compatibility with popular web browsers, including Chrome, Safari, and Opera, guarantees that users can access CampusConnect with ease using any web browser. Additionally, we've chosen color tones (Orange&White) that are easy on the human eye to prevent visual fatigue and ensure a pleasant reading experience. Our buttons are designed to be clear and user-friendly, streamlining interactions. By prioritizing usability, we aim to provide an accessible, enjoyable, and efficient platform for all users, promoting engagement and making CampusConnect a user's first choice.

2.4 Reliability

Reliability is at the core of CampusConnect, and its significance cannot be

overstated. Our commitment to offering a dependable platform means that the CampusConnect app will be available 24/7, with minimal downtime reserved exclusively for essential maintenance. Furthermore, we've taken measures to ensure the system can recover gracefully from any unforeseen system failures, all without the loss of any critical data. We understand that reliability is the linchpin of user trust and satisfaction. It's the assurance of uninterrupted service, data integrity, and a seamless experience for our users. By prioritizing reliability, we aim to provide a platform that users can depend on, day in and day out, fostering a strong sense of trust and loyalty within our digital community.

3.0 Tech Stack

3.1 Back-End (Server Side):

Python: All back-end stages of Campus Connect is based on Python, and its frameworks for web development. Python is a high-level programming language which is known as readable, and it supports various programming approaches such as object-oriented, functional etc. It has wide usage in web development, artificial intelligence, data analysis, and automation. One of the most significant advantages is that its extensive libraries and frameworks make backend development more efficient.

Django: Campus Connect is a web-based project arised with the Django framework. Django is a high-level Python web framework providing almost everything a web-project needs such as object modeling (ORM), admin panel, and authentication support. Suitable for building scalable and maintainable web applications.

Django Rest Framework (DRF): Django Rest Framework is used for building APIs for Campus Connect. Its features include serialization for ORM and non-ORM data sources, authentication policies, and extensive documentation. Ideal for developing complex, database-driven websites and enabling web services/APIs.

3.2 Front-End (Client Side):

HTML: HTML is supported by the Django framework to provide a user interface after API calls. The standard markup language used to create web pages. Its main advantage for Campus Connect is to design all the web content and connect with back-end with calling urls with respect to users choice for all web pages.

CSS: CSS, namely Cascading Style Sheets is used to make the style of the HTML contents better and more specific. It can control the layout of multiple web pages all at once so that the design pattern of the Campus Connect is more stable.

JavaScript: JavaScript is used in the front-end stage of Campus Connect. It makes the HTML pages responsive and increases its functionality.

Bootstrap: A front-end framework for developing responsive and advanced user interfaces. It includes HTML and CSS design templates for various interface components, and with some JavaScript extensions. Campus Connect front-end stage contains several Bootstrap components in particular pages.

3.3 Database:

MySQL: MySQL is a relational database management system that uses structured query language (SQL) for accessing and managing the data stored in relational databases. Campus Connect is based on ORM objects such as profile, post, entry, category objects etc., which are stored in MySQLdatabase provided by Amazon Web Services.

Google Firebase (Pyrebase4 Framework): Firebase is a platform developed by Google for storing visual components, and each image in Campus Connect is stored there. Pyrebase4 is a Python wrapper for the Firebase API, and it ensures the storing of images with basic back-end code parts.

3.4 Server Infrastructure:

EC2 (Amazon Web Services): EC2 (Elastic Compute Cloud) is a part of Amazon's cloud-computing platform allowing users to use virtual machines for various applications to deploy the web-applications. Campus Connect is planning to deploy a virtual machine with EC2.

3.5 Version Control and DevOps Tools:

Git: A distributed version-control system for tracking changes in source code during software development. All of the version controls and code sharing is handled with Git and GitHub for Campus Connect.