CS 319 Object-Oriented Software Engineering

Deliverable 1

Use Case Diagrams / NFRs / Tech Stack

S2T8

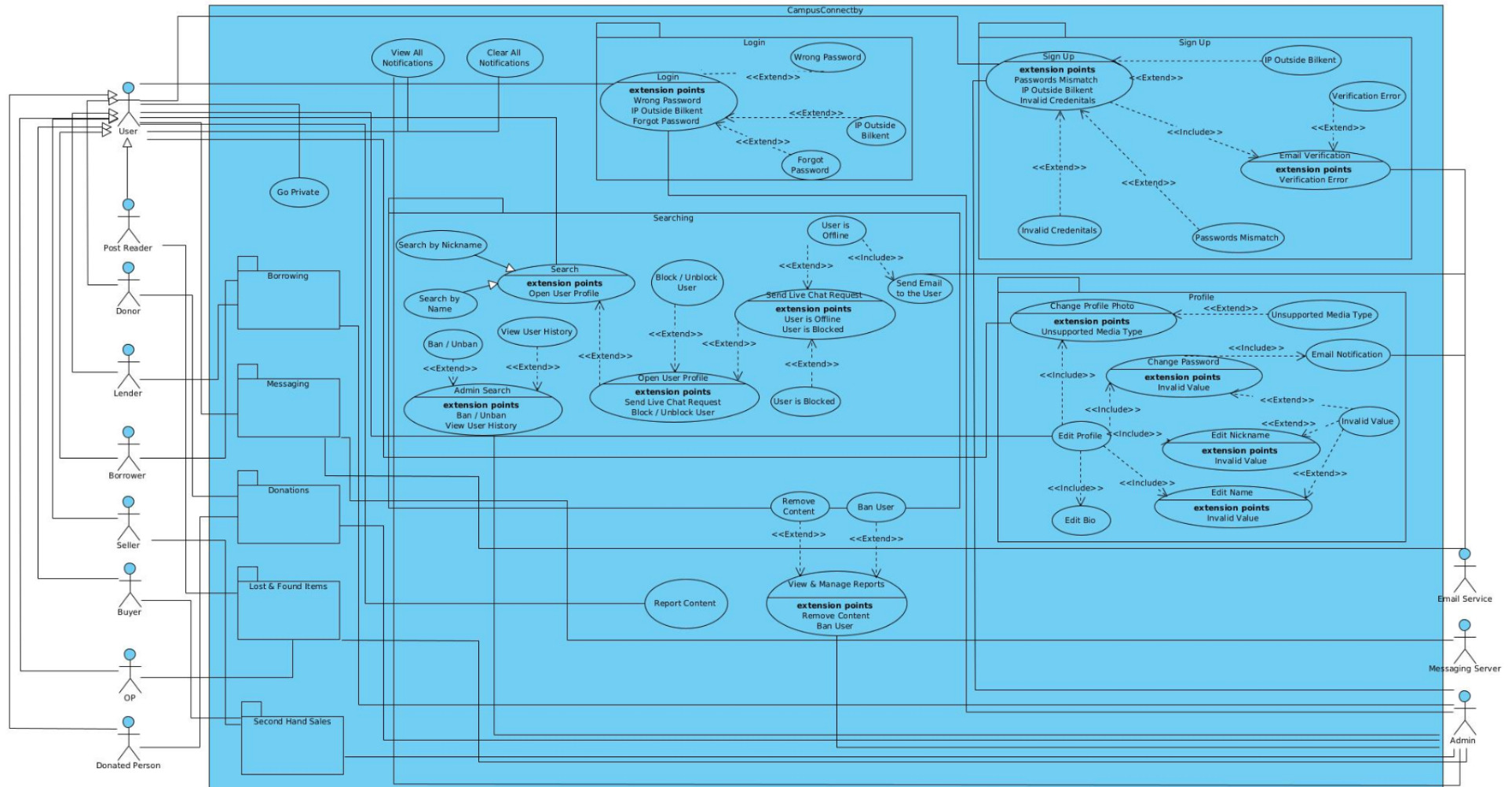| Bilginer Oral	Ercan Bahri Nazlıoğlu	Alper Biçer	Mohamed Amine Boukattaya	Eren Karakaş
22103163	21903151	22003097	22301131	22002722

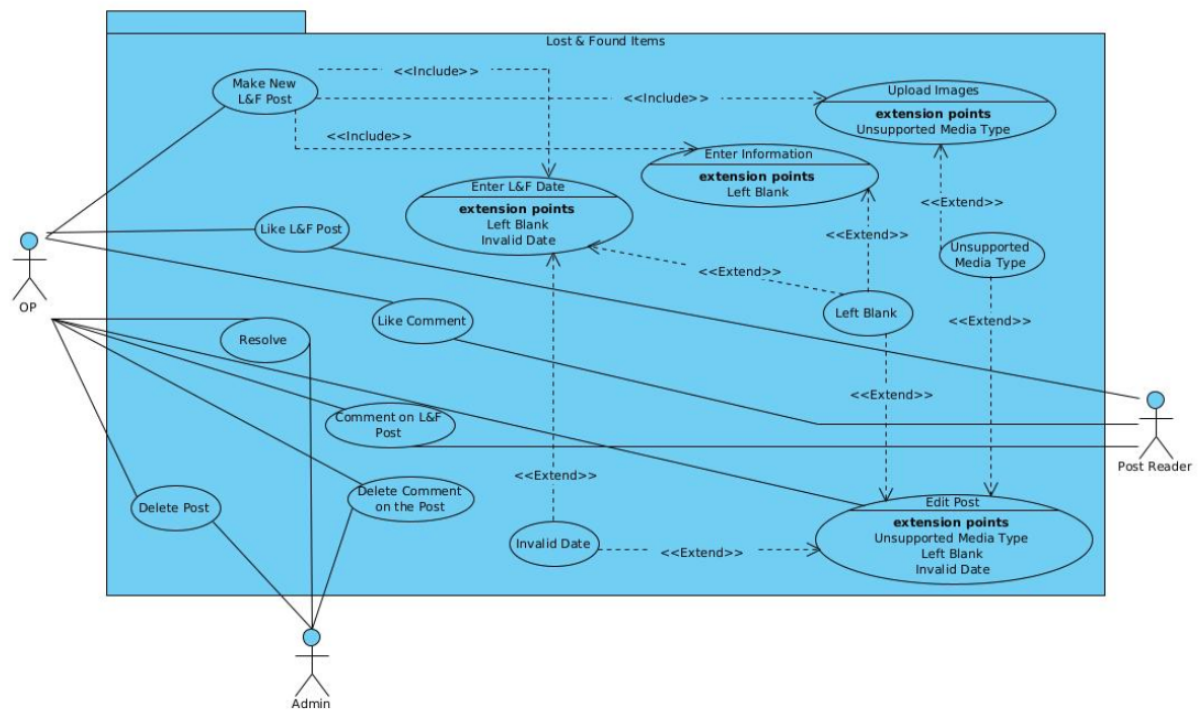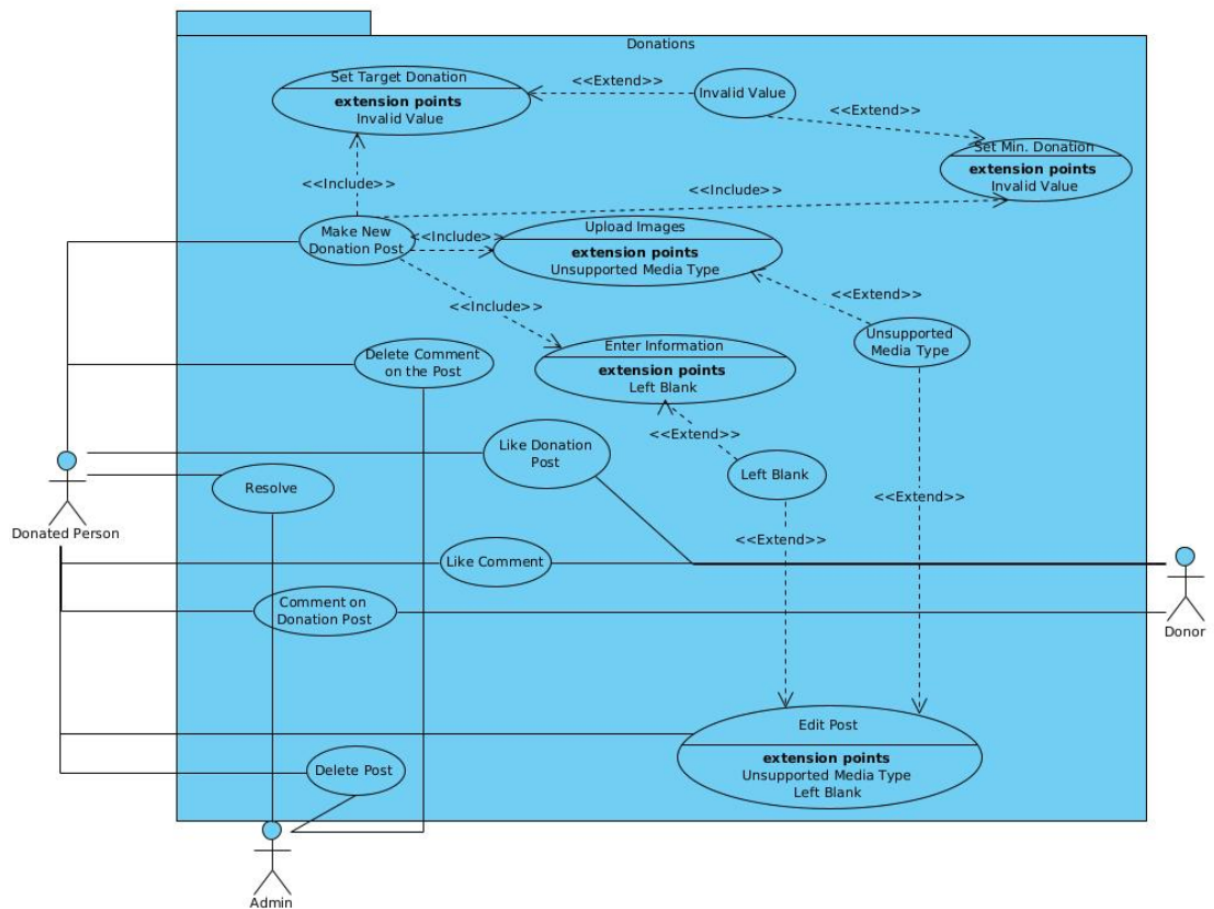# Table Of Contents

Use Case Diagrams

1. CampusConnect System

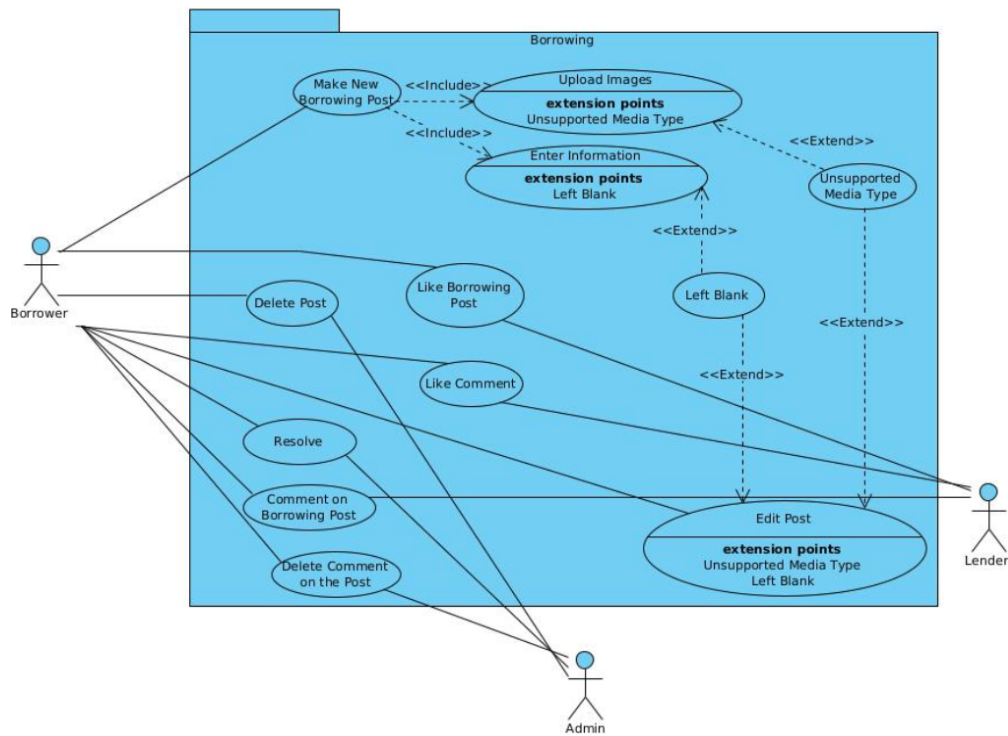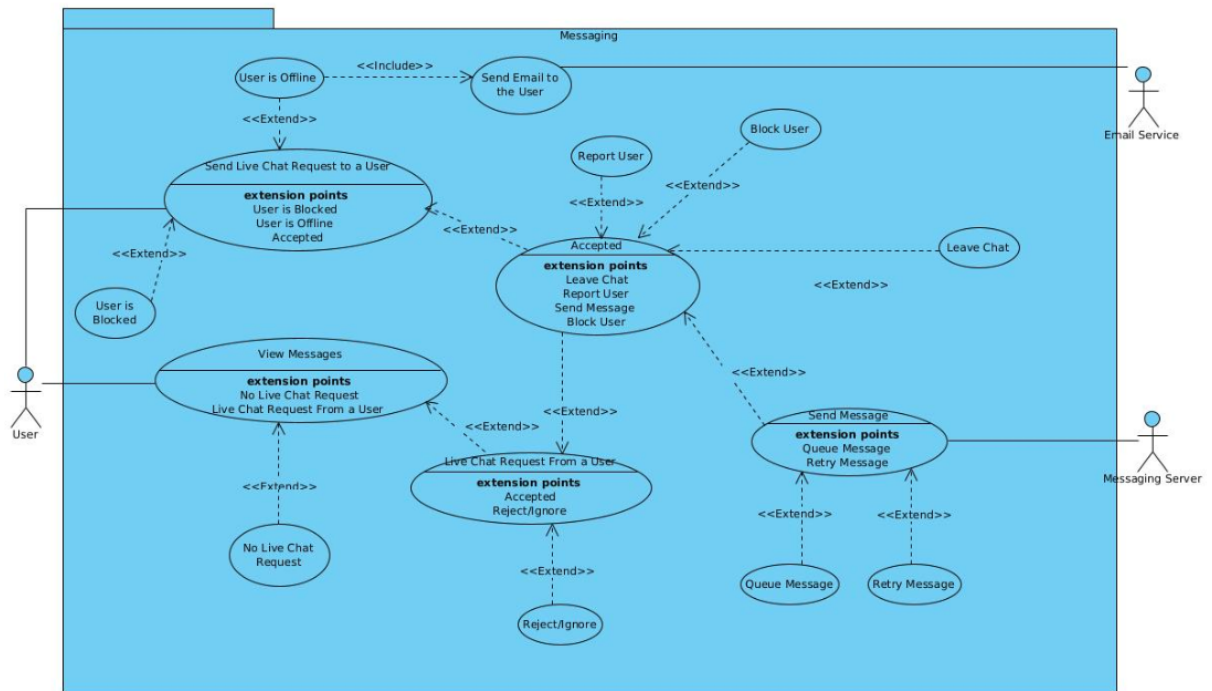## 2. Lost & Found Items

3. Donations

4. Borrowing

5. Second Hand Sales
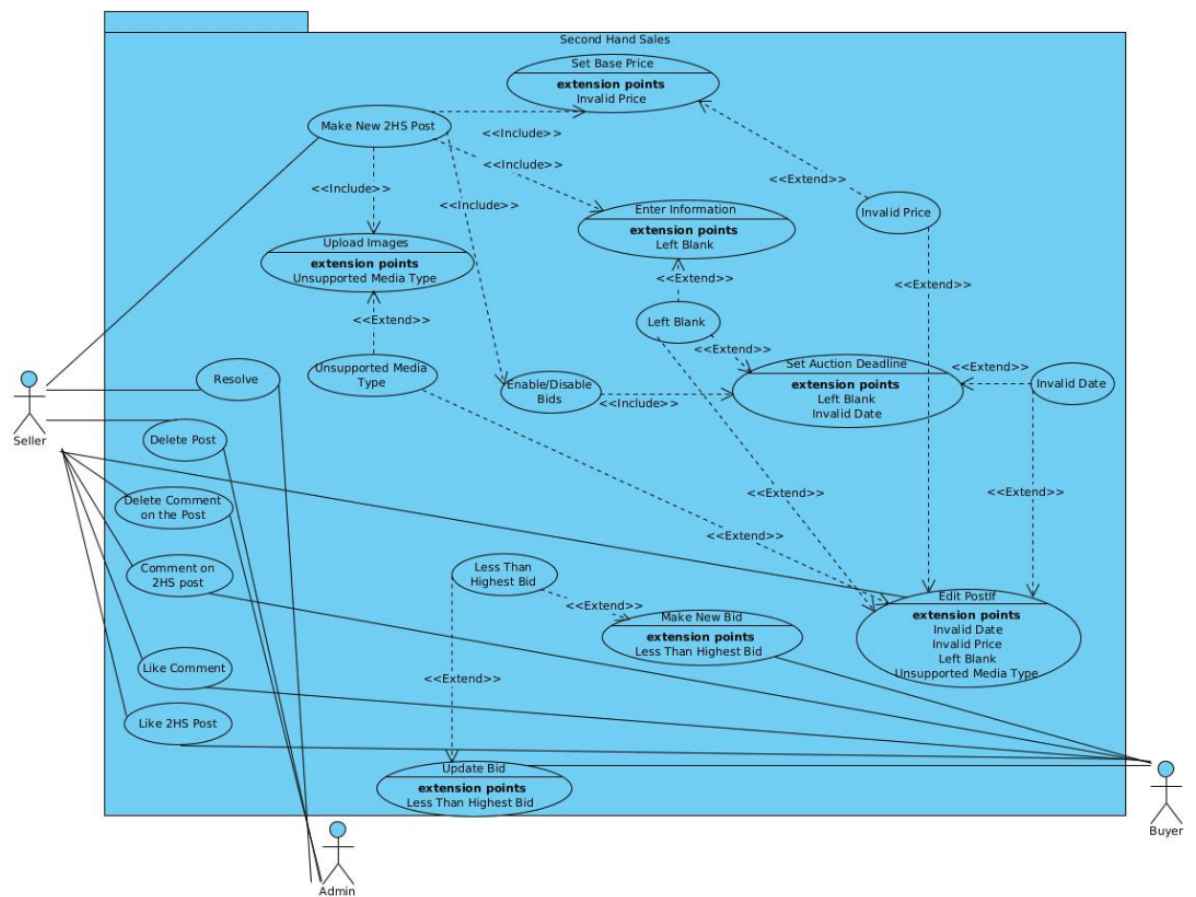
6. Messaging

Textual Specifications

CampusConnect System

Name	Login
Participating Actor(s)	User, Admin
Entry Condition	Open the web application through a web browser.
Exit Condition	Successfully log in to the system.

Flow of Events	1. Open the web application through a browser.
	2. Enter credentials to log in to the system.
	    2.1 If the password is incorrect, the login process fails.
	    2.2 If connected via a network that is not within Bilkent, the login process fails.
	    2.3 If clicked the "Forgot Password" button, the user is routed to a new page in which he/she can change password.

Name	Sign up
Participating Actor(s)	User
Entry Condition	Open the web application through a web browser.
Exit Condition	Successfully sign up to the system.

Flow of Events	1. Open the web application through a browser.
	2. Enter the necessary information to sign up to the system.
	    2.1 If entered invalid credentials, the signup process fails (e.g. impossible Bilkent ID).
	    2.2 If connected via a network that is not within Bilkent, the sign up process fails.
	    2.3 If the passwords entered by the user don't match, the sign up process fails.
	3. Afterwards, a verification e-mail is sent to the user. The user must enter the verification key to finalize the sign up process.
	    3.1 If the verification keys do not match, the sign up process fails.

Name	View All Notifications
Participating Actor(s)	User, Admin
Entry Condition	Must be logged in to the system.
Exit Condition	Click "Home" Button
Flow of Events	1. Click the "Notifications" button while in homepage. 2. View all the notifications (likes, comments etc.).

Name	Clear All Notifications
Participating Actor(s)	User, Admin
Entry Condition	Must be logged in to the system.
Exit Condition	Click "Home" Button
Flow of Events	1. Click the "Notifications" button while in homepage. 2. Clear all the notifications (likes, comments etc.).

Name	Report Content
Participating Actor(s)	User
Entry Condition	Must be logged in to the system.
Exit Condition	N/A
Flow of Events	1. Click the "Exclamation Mark (!)" that lies at the top of a post or a comment. 2. Enter the justification for the report.

Name	Go private
Participating Actor(s)	User
Entry Condition	Must be in homepage.
Exit Condition	Click "Go Public" button.
Flow of Events	1. Users may choose to "go private". This allows users to post, comment or like anonymously. This feature has been influenced by the Instagram page "Bilkent_itiraf_ediyor".

Name	View & Manage Reports
Participating Actor(s)	Admin
Entry Condition	Must have opened the "Reports" page.
Exit Condition	Go back to main page.

Flow of Events	1. Open the "Reports" page to see the reports sent by users.
	2. View all the reports.
	    2.1 If the content being reported is considered inappropriate / harmful, the admin removes it.
	    2.2 If the admin thinks that a particular user uploads inappropriate / harmful content, he/she can ban that user.

Name	Search
Participating Actor(s)	User, Email Service
Entry Condition	Must have clicked the "Magnifier (🔍)" icon.
Exit Condition	Go back to main page.

Flow of Events	1. Click the "Magnifier (🔍)" icon.
	2. For users, it is possible to search users both by nickname and name.
	    2.1 If the profile of the searched user is opened, users may
	        2.1.1 Block / Unblock that user.
	        2.1.2 Send Live Chat Request to that user.
	           a. If the user that sends the request is blocked by the other user, this operation fails and a warning is displayed on the screen.
	           b. If that user is not online at that time, this operation fails and an e-mail is sent to him/her.

Name	Admin Search
Participating Actor(s)	Admin
Entry Condition	Must have clicked the "Magnifier (🔍)" icon.
Exit Condition	Go back to main page.

Flow of Events	1. Admins search for users by Bilkent IDs. When they open users' profiles, through this way, they can
	    1.1 Ban / Unban that user.
	    1.2 View that user's activity history (his / her posts and comments, for example).

Name	Edit Profile
Participating Actor(s)	User
Entry Condition	Click on the "Profile" page.
Exit Condition	Go back to main page.
Flow of Events	1. Open the "Profile" page.
	2. Users can edit profiles in this page.
	2.1 Users can change profile photo. If the user upload an unsupported media type (e.g. system expects a .png file but the user uploads a .pdf file) this operation fails.
	2.2 Users can edit their bio.
	2.3 Users can edit their nickname. If their new nickname is invalid (e.g. 0 characters long) then this operation fails.
	2.4 Users can edit their name (actual name). If their new nickname is invalid (e.g. 0 characters long) then this operation fails.
	2.5 Users can change their passwords.
	2.5.1 If they enter an invalid password (for example, a password that does not satisfy the security requirements), this operation fails.
	2.5.2 After they change their password, an e-mail is sent to them to ensure everything is being done on their behalf.

Lost & Found Items

Name	Make New L&F Post
Participating Actor(s)	OP (Original Poster)
Entry Condition	Must have clicked "Upload New Post > L&F Post" in the homepage.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Upload new Lost & Found post to the system.
	1.1 Upload related images. If the media type is not supported, this operation fails.
	1.2 Enter the date when the item was lost or found. If this field is left empty, or the date is invalid (a future date, for example) this operation fails.
	1.3 Enter related information. If this field is left empty, this operation fails.

Name	Like L&F Post
Participating Actor(s)	OP, Post Reader
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Like the L&F post.

Name	Like Comment
Participating Actor(s)	OP, Post Reader
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Like the comment

Name	Comment on L&F Post
Participating Actor(s)	OP, Post Reader
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Comment on the L&F post.

Name	Delete Post
Participating Actor(s)	OP, Admin
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Remove the post from the system.

Name	Delete Comment on the Post
Participating Actor(s)	OP, Admin
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Comment on the L&F post.

Name	Edit Post
Participating Actor(s)	OP
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.

Flow of Events	1. Edit the post.
	  1.1 If unsupported media type is uploaded to the system, this operation fails.
	  1.2 If any of the fields is left empty, this operation fails.
	  1.3 If invalid date is entered, this operation fails.

Name	Resolve
Participating Actor(s)	OP, Admin
Entry Condition	Must be in the homepage or the OP's profile page.
Exit Condition	If accessed through OP's profile page, go back to the homepage.
Flow of Events	1. Resolve the post. This means the issue has been resolved and no longer needs attention (e.g. the item was found and given back to its owner).

Donations

Name	Make New Donation Post
Participating Actor(s)	Donated Person
Entry Condition	Must have clicked "Upload New Post > Donation Post" in the homepage.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.

Flow of Events	1. Upload new Donation post to the system.
	   1.1 Upload related images. If the media type is not supported, this operation fails.
	   1.2 Set a target donation (e.g. the target is $1000). If the value is invalid (e.g. a negative number) this operation fails.
	   1.3 Set the minimum possible donation. If an invalid value is entered, this operation fails.
	   1.4 Fields 1.3 and 1.4 can be left empty as the term "donation" encompasses a wide range of potential situations beyond just monetary contributions.
	   1.5 Enter related information. If this field is left empty, this operation fails.

Name	Like Donation Post
Participating Actor(s)	Donated Person, Donor
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Like the Donation post.

Name	Like Comment
Participating Actor(s)	Donated Person, Donor
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Like the comment

Name	Comment on Donation Post
Participating Actor(s)	Donated Person, Donor
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Comment on the Donation post.

Name	Delete Post
Participating Actor(s)	Donated Person, Admin
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Remove the post from the system.

Name	Delete Comment on the Post
Participating Actor(s)	Donated Person, Admin
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Comment on the Donation post.

Name	Edit Post
Participating Actor(s)	Donated Person
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.

Flow of Events	1. Edit the post.
	  1.1 If unsupported media type is uploaded to the system, this operation fails.
	  1.2 If any of the required fields is left empty, this operation fails.

Name	Resolve
Participating Actor(s)	Donated Person, Admin
Entry Condition	Must be in the homepage or the Donated Person's profile page.
Exit Condition	If accessed through Donated Person's profile page, go back to the homepage.
Flow of Events	1. Resolve the post. This means the issue has been resolved and no longer needs attention (e.g. target donation was reached).

Borrowing

Name	Make New Borrowing Post
Participating Actor(s)	Borrower
Entry Condition	Must have clicked "Upload New Post > Borrowing Post" in the homepage.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.

Flow of Events	1. Upload new Borrowing post to the system.
	    1.1 Upload related images. If the media type is not supported, this operation fails.
	    1.2 Enter related information. If this field is left empty, this operation fails.

Name	Like Borrowing Post
Participating Actor(s)	Borrower, Lender
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Like the Borrowing post.

Name	Like Comment
Participating Actor(s)	Borrower, Lender
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Like the comment

Name	Comment on Borrowing Post
Participating Actor(s)	Borrower, Lender
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Comment on the Borrowing post.

Name	Delete Post
Participating Actor(s)	Borrower, Admin
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Remove the post from the system.

Name	Delete Comment on the Post
Participating Actor(s)	Borrower, Admin
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Comment on the Borrowing post.

Name	Edit Post
Participating Actor(s)	Borrower
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.

Flow of Events	1. Edit the post.
	   1.1 If unsupported media type is uploaded to the system, this operation fails.
	   1.2 If any of the fields is left empty, this operation fails.

Name	Resolve
Participating Actor(s)	Borrower, Admin
Entry Condition	Must be in the homepage or the Borrower's profile page.
Exit Condition	If accessed through Borrower's profile page, go back to the homepage.
Flow of Events	1. Resolve the post. This means the issue has been resolved and no longer needs attention.

Second Hand Sales

Name	Make New Second Hand Sales Post
Participating Actor(s)	Seller
Entry Condition	Must have clicked "Upload New Post > 2HS Post" in the homepage.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.

Flow of Events	1. Upload new 2HS post to the system.
	   1.1 Upload related images. If the media type is not supported, this operation fails.
	   1.2 Enter related information. If this field is left empty, this operation fails.
	   1.3 Set base price for the item that is being sold. If the base price is invalid, this operation fails.
	   1.4 Enable/Disable bids. Enabling bids will allow the buyers to make bids in an auction.
	      1.4.1 If bids are enabled, the user is required to set an auction deadline to specify when the

auction ends. However, if the user leaves this field empty, or enters an invalid date, this operation fails.

Name	Like 2HS Post
Participating Actor(s)	Seller, Buyer
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Like the 2HS post.

Name	Like Comment
Participating Actor(s)	Seller, Buyer
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Like the comment

Name	Comment on 2HS Post
Participating Actor(s)	Seller, Buyer
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Comment on the 2HS post.

Name	Delete Post
Participating Actor(s)	Seller, Admin
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Remove the post from the system.

Name	Delete Comment on the Post
Participating Actor(s)	Seller, Admin
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Comment on the Seller post.

Name	Edit Post
Participating Actor(s)	Seller
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.

Flow of Events	1. Edit the post.
	    1.1 If unsupported media type is uploaded to the system, this operation fails.
	    1.2 If any of the required fields is left empty, this operation fails.
	    1.3 If invalid base price is selected, this operation fails.
	    1.4 If invalid auction deadline is selected, this operation fails.

Name	Resolve
Participating Actor(s)	Seller, Admin
Entry Condition	Must be in the homepage or the Seller's profile page.
Exit Condition	If accessed through Seller's profile page, go back to the homepage.
Flow of Events	1. Resolve the post. This means the issue has been resolved and no longer needs attention (e.g. the item is sold).

Messaging

Name	View Messages
Participating Actor(s)	User
Entry Condition	Must have clicked the "Messages" icon (✉) from the homepage.
Exit Condition	Go back to homepage

Flow of Events	1. View if there are any live chat requests. There may or may not be live chat requests from other users.
	   1.1 If there is a live chat request from a user, it is possible to accept or reject/ignore it.
	     1.1.1 If rejected, user is routed to the same messages page.
	     1.1.2 If accepted, user is routed to a live chat screen, in which he/she can chat with another user in real-time.
	
	       User can block the other user or can be blocked by the other user. In that case, the live chat is cut off immediately and the user is returned to the "Messages" page.
	
	       The user can also report the other user if he/she performs inappropriate behavior.
	
	       The users can leave the live chat by clicking the "Leave" button.
	
	        a. If a message cannot be delivered, message is added to a queue.
	        b. If a message drops from the queue without being delivered, message can be chosen to resend.

Name	View Messages
Participating Actor(s)	User
Entry Condition	Must have clicked the "Messages" icon (✉) from the homepage.
Exit Condition	Go back to homepage

Flow of Events	1. Users will be able to send live chat requests to other users.
	   1.1 If the user that sends the live chat request is blocked by the other user, this operation fails.
	   1.2 If the user that is sent the live chat request is offline, this operation fails and an email is sent to

that user, informing his/her that there was a live chat request.

1.3 If the request is rejected, user is routed to the same messages page.

1.4 If the request is accepted, user is routed to a live chat screen, in which he/she can chat with another user in real-time.

User can block the other user or can be blocked by the other user. In that case, the live chat is cut off immediately and the user is returned to the "Messages" page.

The user can also report the other user if he/she performs inappropriate behavior.

The users can leave the live chat by clicking the "Leave" button.

a. If a message cannot be delivered, message is added to a queue.
b. If a message drops from the queue without being delivered, message can be chosen to resend.

Non-functional Requirements

In developing CampusConnect, we recognize the significance of non-functional requirements in ensuring the project's success. This section of the report highlights the non-functional requirements of the project with a focus on usability, security, maintainability, and portability to highlight their importance in the development of our application.

a. Usability

Our main objective is to make life easier for the Bilkent community by offering them a platform in which they can find everything related to their specific needs. The user interface's design follows this objective, offering an intuitive and visually pleasing interface while also guaranteeing that users can effortlessly navigate the platform and reach various features. The term usability places a significant emphasis on performance; hence, our goal is to uphold an average page load time of less than 3 seconds to enhance the user experience, as measured through tools like Google PageSpeed Insights. Because our community has prior experience with similar applications like Instagram or Facebook, understanding most of the flow of the application will not take users more than a few seconds.

b. Security

CampusConnect will be carefully designed to ensure a secure and protected digital environment for its users. Security has core importance in the design of the application. As part of this commitment, our team will implement a restriction that confines user access to only when said users are within the Bilkent network. In essence, this means that users who are not connected to the Bilkent network will not be allowed to use the application. The implementation of this safety measure will involve an IP address check every time a user tries to use the application.

Furthermore, user privacy is a core aspect of our platform. To respect our users' privacy, we allow them to personalize their profiles, putting them in full control of the information they choose to expose to other users.

As a component of our safety measures, a dedicated team of administrators will be responsible for upholding the system's security and removing bad actors. Users will be able to report any posts, comments, or messages that raise concerns, and the administrators will possess the authority to remove any content that is deemed inappropriate or harmful.

c. Maintainability

Ensuring the application remains manageable and adaptable is crucial for its long-term success. To achieve this, our team will follow the following paradigms:

Modularity: The design phase will involve implementing small, testable units and merging them afterwards. This approach allows us to detect and fix any potential bugs before they lead to detrimental results. Moreover, our adoption of the Object-Oriented Programming (OOP) paradigm complements the modularity approach. OOP is an essential part of our development strategy, as it inherently encourages the creation of self-contained and reusable components, promoting a modular and organized codebase.

Documentation: We aim to establish comprehensive and extensive documentation for the source code, accompanied by detailed comments integrated into the codebase. By doing so, we aim to create a knowledge-rich resource that not only aids in understanding the application's source code but also simplifies the maintenance process. This approach will save valuable time, eliminating the need to backtrack and recollect the details of the code's functionality.

Version Control: Using Git & Github will be an essential aspect of the project to ensure maintainability. With Git, we will have a robust version control system, enabling us to track changes, manage code versions, and collaborate seamlessly among team members. This not only simplifies the process of identifying and rectifying issues but also facilitates the integration of new features and enhancements. By following these approaches, we aim to achieve a well-organized and maintainable application, streamlining code management and project collaboration.

d. Portability

The application is expected to be cross-platform, meaning that it will be designed to function seamlessly across multiple platforms. As such, our team intends to create it to be compatible with various operating systems, such as Windows, GNU/Linux, and MacOS, offering an inclusive user experience. Furthermore, our team strives to support a variety of web browsers, including but not limited to Mozilla Firefox, Google Chrome, and Microsoft Edge.

Tech Stack

We have opted to employ a combination of React.js for the front-end, FastAPI for the back-end, and PostgreSQL for the database in our application's tech stack. Several reasons support our choice of this tech stack.

Front-End with React.js:

Our first choice, React.js, stands as a robust selection for our front-end development. React.js is a highly popular front-end framework that provides extensive and readily accessible documentation. This is essential for our team since our team first needs to learn the basics of web development and web frameworks. The abundance of documentation will accelerate this learning process and help reduce the complexities of various intricacies of web development.

Furthermore, React.js is known for its performance and efficiency, aligning with the non-functional requirements of our project. Its capability of rendering user interfaces and efficiently updating data has the potential to elevate the overall user experience. Since we want our application to be scalable after the release, React.js emerges as the best fit for the front-end component.

Back-End with FastAPI:

We decided to go with FastAPI for the back-end of our project, which is a Python-based framework. We chose to use FastAPI because of its commitment to streamlined development. FastAPI significantly reduces the amount of code that developers need to write, distinguishing itself from other back-end frameworks like Django, Spring Boot, or Node.js with Express.

The decreased development workload holds significant value for us, particularly because our team members are still in the initial phases of familiarizing themselves with this back-end framework. FastAPI's abundant online documentation adds to its attractiveness. Having access to well-documented resources translates into shorter development times, making it a wise decision to accelerate the back-end development of our project.

Database Component with PostgreSQL:

We've adopted PostgreSQL as our database management system, a well-known and time-tested option known for its reliability and extensive feature set. The reliability aspect holds the most importance in our project, particularly for our future scalability goals.

While PostgreSQL's strong reputation for dependability aligns perfectly with our long-term vision, it is necessary to acknowledge that this reliability comes with some trade-offs. The implementation phase might require a slightly higher level of complexity and time investment compared to alternatives like SQLite.

In summary, our choice to construct this technology stack, which includes React.js, FastAPI, and PostgreSQL, has been influenced by a thorough evaluation of the amount of available documentation, user-friendliness, performance, and reliability each element offers. This tech stack serves as the foundation upon which we will construct our application, providing the framework for a successful software solution.