

Software Requirements Specification

for

Donation App

Version 1.0

Prepared by

Group

Sapan Sharma	6352819	Ss17hq@brocku.ca
Alanna McNulty	6275168	am17xy@brocku.ca
Alexander Bowles	6089957	ab16hc@brocku.ca
Alex Orvitz	5517693	ao13al@brocku.ca
Brian Jenkins	6063481	bj16hh@brocku.ca
Arooj Fateh	6246185	Af16iy@brocku.ca
Samantha Laffin	5017538	sl11hm@brocku.ca

Instructor: Naser Ezzati-Jivan

Course: COSC 4P02

Date: February 7 2021

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Glossary of Terms	1
1.3 Intended Audience	1
1.4 Product Scope	1
1.5 Document Overview	2
2. Overall Description	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics.....	3
2.4 Entity-Relationship Diagram	3
2.5 Operating Environment.....	4
2.6 Design and Implementation Constraints.....	4
2.7 User Documentation	5
2.8 Assumptions and Dependencies.....	5
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Hardware Interfaces	10
3.3 Software Interfaces	10
3.4 Communications Interfaces.....	10
4. System Features	11
4.1 Account Creation	11
4.2 Login.....	11
4.3 Profiles and Donation Management.....	12
4.4 Search Engine	12
4.5 Real-Time Chat Integration	13
4.6 Location Display for a Specific Non-Profit Organization	13
4.7 View Local Non-Profits.....	14
4.8 ‘Sign-Up’ for Specific Types of Donation or Non-Profit	15
4.9 Notifications.....	16
5. Other Nonfunctional Requirements.....	16
5.1 Performance Requirements	16
5.2 Safety Requirements	17
5.3 Security Requirements	17
5.4 Software Quality Attributes	17
5.5 Business Rules	18
6. Design Diagrams	18
6.1 Use Cases View	18
6.2 Deployment Diagram.....	22
6.3 Logical View.....	22
6.4 Sequence Diagrams.....	23
6.5 Class Diagram.....	28
6.6 Database Diagram.....	29

1. Introduction

1.1 Purpose

This document contains the complete design description of the Donation app. This includes but is not limited to the database design, architectural design, and the list of use cases which will be implemented by the system and various other diagrams.

1.2 Glossary of Terms

Term	Description
User	User is someone who uses and interacts with the app.
Admin	This term is used to refer to app administrators who have special permissions to manage and control the app.
Donor	This term is used to refer to people who are using the app to search for non-profit organizations and to donate.
Non-profit	This term will be used to refer to non-profit organizations.
Receiver	This term is used to refer to a user in need who can posts their situation and items in need.
UI	This term is short for user interface.

1.3 Intended Audience

This document is meant for software developers. The developer should have a technical background and be able to understand use-case diagrams, activity diagrams, sequence diagrams, class diagrams, state diagrams and interface designs.

1.4 Product Scope

The main goal of this system is to connect needy individuals and non-profit organizations with people who are willing to donate goods in a way that is accessible, fast, and easy for both the non-profit organizations and people who require help. This app will provide a platform where individuals can post their situations, non-profit organizations can post their campaigns, users can get information about the charity contributions of these organizations, and donors can list items which they want to donate.

1.5 Document Overview

This document follows the IEEE standards for Software Specification documents. Each requirement statement will have its own priority.

Chapter 1: Provides an introduction.

Chapter 2: Gives an overall description of the system.

Chapter 3: Outlines the external interfaces, including user interfaces.

Chapter 4: Gives descriptions of all the features of this system.

Chapter 5: Lists the non-functional requirements, such as performance, safety, and security.

Chapter 6: Lists all the design diagrams of this system.

2. Overall Description

2.1 Product Perspective

This application is a new self-contained product, which will provide users with the ability to find and donate to charities and non-profits of their choice and find more information in a centralized app. Various user will be able to register an account and log in with their unique profile to explore their donation options, and non-profit organizations can find potential donors.

The software will be utilized on android devices and will be written in Java. We will use Firebase as the Realtime database server and for backend infrastructure.

2.2 Product Functions

- 2.2.1. Users will be able to register their profile with their unique set of details such as email address and password initially. In future changes, we could allow for Google or Facebook quick login functionality.
- 2.2.2. Registered users will be able to sign in and post their donation items.
- 2.2.3. Users will be able to browse through a list of non-profit organizations and campaigns created by other users.
- 2.2.4. Users will be able to find organizations and campaigns closer to them through location proximity.
- 2.2.5. Users will be able to find organizations and campaigns using a search bar by entering names, tags or other descriptors.
- 2.2.6. Users will be able to send emails or messages to get more information from other users.
- 2.2.7. Users will be able to see their donation history and will be given a token of appreciation for each donation.
- 2.2.8. Users will be able to post information about their non-profit organizations or campaigns.

- 2.2.9. Non-profit organizations can send customized messages to all previous donors to alert them to their current needs.

2.3 User Classes and Characteristics

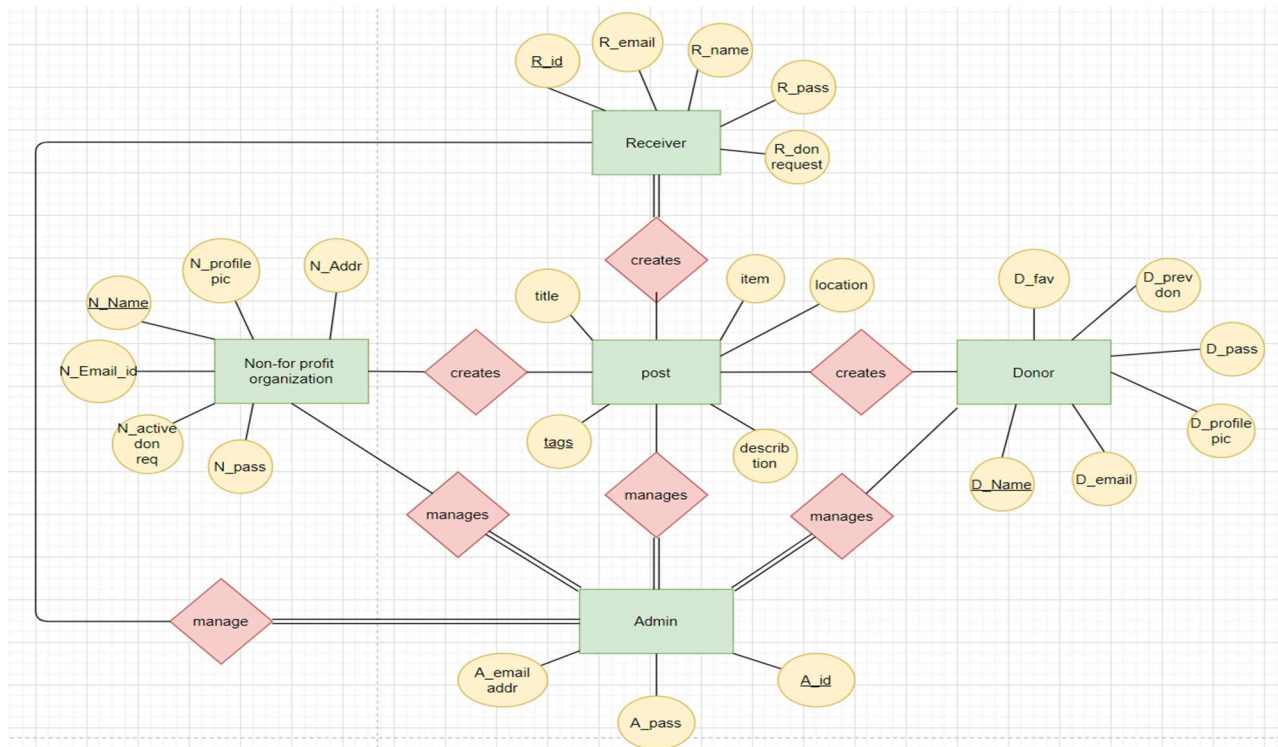
Users of the app should be able to see the list of non-profit organizations, campaigns and posts by people in need according to their preferences and descriptors like location, cause and favorites.

The app will support four types of user privileges, namely: Admin, Donor, Organization and receiver. Admin will have access to management and administrative functions, and other users will have access to different functions as mentioned below.

1. Admin
 - 1.1. Add/Delete profile
 - 1.2. Add/Delete posts
 - 1.3. Update profiles
 - 1.4. Add new owners
 - 1.5. Manages data
2. Donor
 - 2.1. See all the non-profit organizations and campaigns
 - 2.2. Post donation information
 - 2.3. Message
3. Organization
 - 3.1. Post campaigns and donation information
 - 3.2. Insights into donors
 - 3.3. Message
 - 3.4. Notify donors for urgent requests
4. Receiver
 - 4.1. Post things they need
 - 4.2. Connect with non-profit and donors
 - 4.3. Message

2.4 Entity-Relationship Diagram

This diagram shows the relationship between different user entities who can create posts and get more information from other user's profiles.



2.5 Operating Environment

The software will operate on Android OS, developed for use on Android 6.0 (Marshmallow) and above, since this OS version works on most Android systems but is still relatively recent.

Firebase will be used which provides backend as a service and real time database to be stored on Firebase's cloud service.

2.6 Design and Implementation Constraints

2.6.1 Developer Constraints

The application will be developed in the Java programming language. The application will utilize a Firebase Realtime Database for data storage and user authentication. Due to the limited scope of the project, the application will only be supported by Android mobile systems. Developers will utilize Google Maps API for location integration within the system.

2.6.2 User Constraints

The application requires users to have an Android device running Android 6.0 or above with a working touch screen and overlay keyboard. Users will be required to have an internet connection to access location information within the application. The user must enable location permissions to provide additional functionality within the app such as nearest pick up and drop of locations. The application will only be supported in English language initially.

2.7 User Documentation

User manual/Technical manual will be available. User help when using the application can be found in the 'help' tab within the application. The help tab will provide answers to frequently asked questions and users may contact the email address provided for further inquiries. If users have inquiries regarding other users such as specific companies or individuals found within the app, they will be directed to the requisite company website for further information.

2.8 Assumptions and Dependencies

2.8.1 Developer Dependencies

Changes to tools used to develop the application will influence the development environment. Namely, changes to Google Maps API and Firebase would influence development times and implementation of services.

2.8.2 User Dependencies

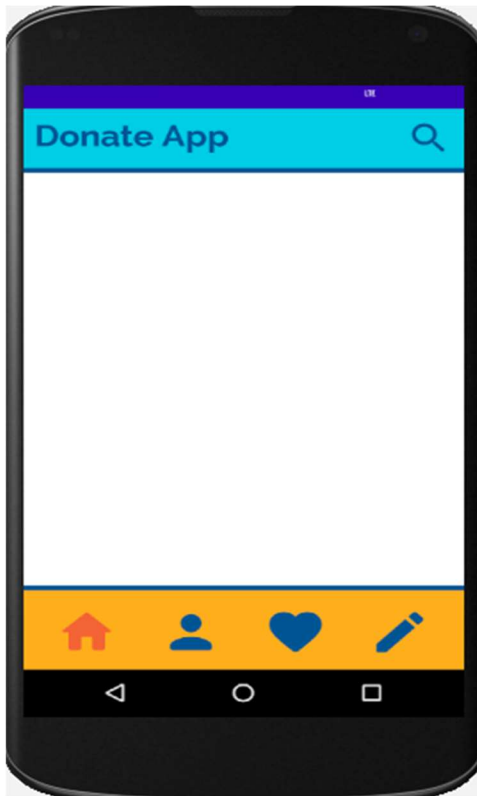
Users are required to have access to the internet. Users must be registered with a username and password in order to access the application. In order to utilize location features of the application, users are required to have either a working internet connection or have location permissions enabled to allow the application to provide information about nearby services.

3. External Interface Requirements

3.1 User Interfaces

- The user interface should be intuitive and easy to use, creating a logical representation of what software is intended to do.
- Proper branding, logos, fonts and color schemes should be implemented with the same GUI standards in the entire application.
- Accessibility features and non-flashy UI designs should be implemented.
- Consistency in UI should be maintained and should be compatible with most screen sizes and android devices.

*These are user interface mockups; final product may vary.



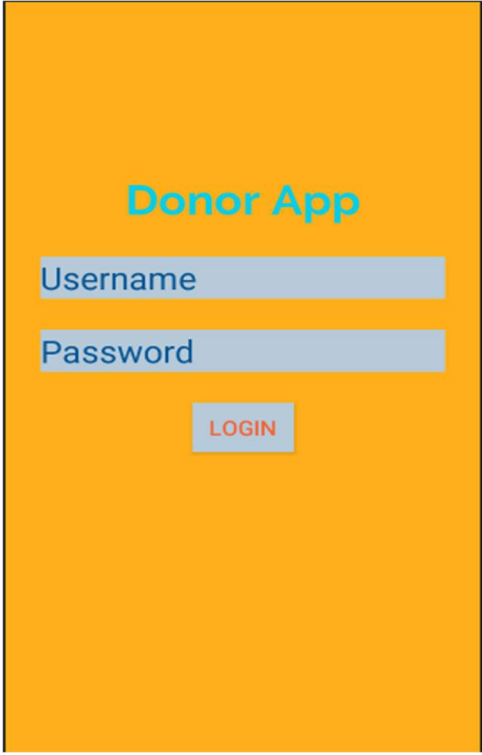
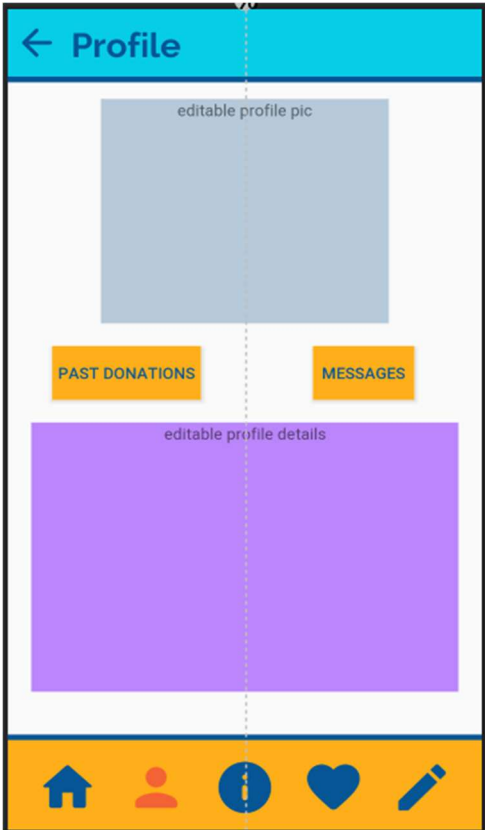
This screen is the Home screen for the application, with the main features being accessible:

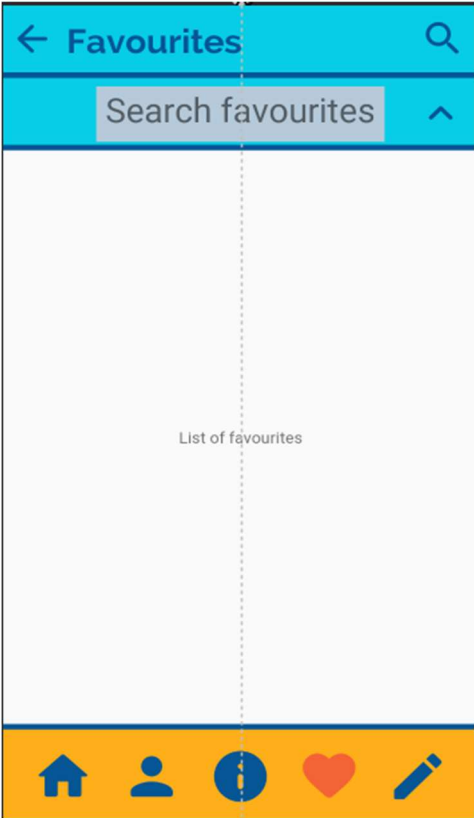

1. Search function (top right)
2. Home button
3. Profile button
4. Favourites button
5. Write Post button

The middle of the page will show updates and activity from favourite non-profits as well as announcements from local ones.



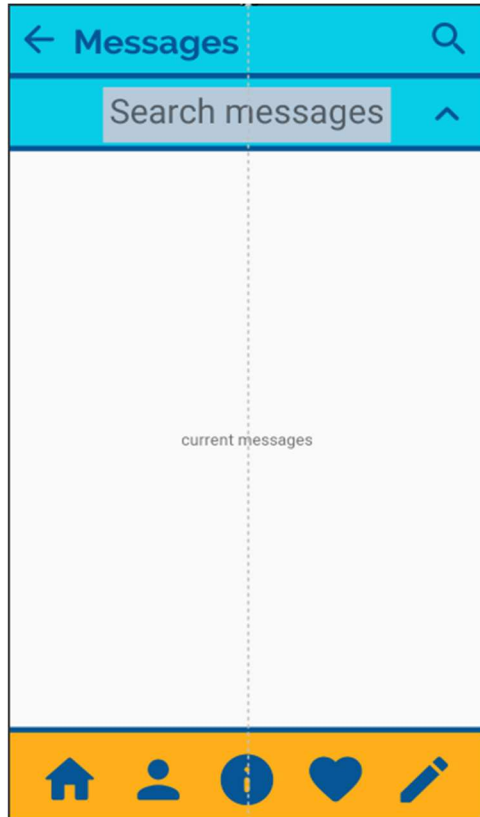
The Home screen again, showing the search function when the search button is pressed. When the Search is used, the middle of the screen will populate a scrolling list of non-profits based on the search and the location area set in the users profile.

 <p>The image shows a login screen for the 'Donor App'. It has an orange background. At the top, the text 'Donor App' is written in a light blue, sans-serif font. Below this, there are two light blue rectangular input fields. The first field is labeled 'Username' in a darker blue font, and the second field is labeled 'Password' in the same font. Below the password field is a light blue rectangular button with the word 'LOGIN' in red, uppercase letters.</p>	<p>This is the login screen which will appear when first using the application and upon startup.</p>
 <p>The image shows a profile screen for the 'Donor App'. It has a white background. At the top, there is a light blue header bar with a white back arrow icon and the word 'Profile' in white. Below the header, there is a large light blue square labeled 'editable profile pic'. Below this square are two orange rectangular buttons: 'PAST DONATIONS' on the left and 'MESSAGES' on the right. Below these buttons is a large purple rectangle labeled 'editable profile details'. At the bottom of the screen is an orange navigation bar with five white icons: a house, a person, a circle with a dot, a heart, and a pencil.</p>	<p>The profile screen will feature a changeable profile picture, as well as editable profile details:</p> <ul style="list-style-type: none">• Email• Username• Password• Possibly payment details• Location setting <p>The user can access their Past Donations as well as their ongoing Messages with non-profits.</p>

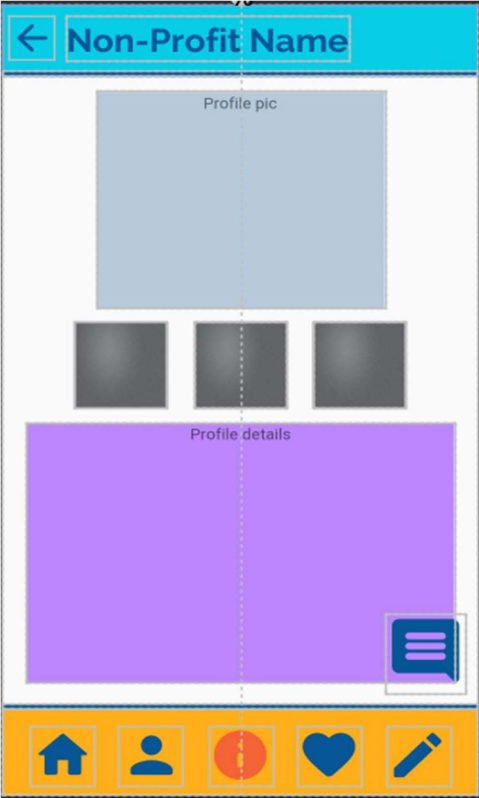
	<p>This screen will show an editable list of those charities and non-profits that the user has marked as favourites.</p>
	<p>This screen will allow users to write posts.</p>



This screen will show the past donations that the user has made. This is accessible through the user profile screen.



This screen is the messages screen, showing ongoing chats. This is accessible through the user profile screen.

	<p>This is an example screen of a non-profits profile page.</p> <p>Images can be uploaded: one profile picture and three additional images (logos, pictures of non-profit activities, etc.) and the address and contact details will be listed.</p> <p>The user has options to message the non-profit (with the hovering message button) and/or tap the Favorite (Heart) button to add this non-profit to their Favorites List.</p> <p>They can also find donation information through this screen.</p>
--	---

3.2 Hardware Interfaces

- App will be functional on Android devices (Android 6.0 and above) using touch screen and other mobile functionality.

3.3 Software Interfaces

- App will be designed and implemented using Android studio.
- App will interact with Firebase service to store user information and content.
- Java libraries and google android based libraries will be used.

3.4 Communications Interfaces

- Cellular or wireless Wi-Fi based connectivity is required to fetch data from the server and update the data on the server.

4. System Features

4.1 Account Creation

4.1.1 Description and Priority

The app provides the user with a GUI to register their account with email address and password to use the app. Users also need to include essential information such as their address, contact email, phone number and a brief description of themselves or their organization. Users must also declare themselves an organization or an individual to provide them with proper roles.

This feature is of high priority.

4.1.2 Stimulus/Response Sequences

1. The user opens the app from the app icon for the first time.
2. The user wants to use the app and register for a new account.
3. User enter details and submit request to create an account.

4.1.3 Functional Requirements

The app welcomes the user on the start page and displays the necessary information required to set up an account.

Possible errors and responses:

1. App should notify user if email address does not exist or not correct domain of email provider
2. App should notify users if they already have id with email entered.
3. App should notify users if any required field are left empty.

4.2 Login

4.2.1 Description and Priority

Upon selection of the profile tab while not currently logged in, users will be prompted for their username and password established previously upon account creation.

Upon incorrect input, users will be informed there was a mistake in their input, and subsequently be prompted once again for input. Upon correct input, users will successfully have access to their account.

This feature is of high priority.

4.2.2 Stimulus/Response Sequences

1. User enters login information.
2. User is granted access to account upon correct input.
3. User is re-prompted for login information upon incorrect input.

4.2.3 Functional Requirements

The app displays a login prompt for the user when the profile tab is selected and no account is currently logged on.

Possible errors and responses:

1. App should notify the user if there is an incorrect pairing between username and password.
2. App should notify the user if there is no account registered under the email entered.
3. App should notify the user if any required fields have been left empty.

4.3 Profiles and Donation Management

4.3.1 Description and Priority

Once registered, individuals and organizations will be able to upload their needs to the app. These needs can range from financial assistance to clothing items and everything in between. Donors, receivers and organizations can create posts to Each account will be accessible to the public through a 'public profile'. Here, donors will be able to see what the organization or individual is currently in need of along with the pertinent information collected during their account creation process. For ease of use, a map may also appear to indicate drop off locations for certain items. For more pressing items, users will be able to flag their posts as 'URGENT' making them more visible to potential donors.

This feature is a high priority.

4.3.2 Stimulus/Response Sequences

1. Users will be able to add their posts/campaigns on the home page by clicking on the add button.
2. This information will also be added to their public profile briefly so other users can see that information.

4.3.3 Functional Requirements

The app provides an easy-to-use UI to post donations and collects all the necessary information required to create the post. And then the app submits the data to Google Firebase database. App displays the user submitted posts on home page and profile from the Firebase database.

Possible errors and responses:

1. App should notify users if the requirements for the post are not met.
2. App should notify users if the data is not sent to the server due to a connection problem.
3. App should display the results aligned as designed for all display sizes and devices.

4.4 Search Engine

4.4.1 Description and Priority

In order for donors to choose the cause that suits them best, a search engine will be implemented. It is at this step where donors will be able to filter and browse through the needs of individuals or organizations. This key feature will also help link those

looking to donate specific items to the organizations who need them the most. For example, if an organization has made an 'URGENT' posting for knitted sweaters, donors with sewing as a hobby will be able to see where their skills are needed the most.

This feature is a medium priority.

4.4.2 Stimulus/Response Sequences

Users click on the search bar and can search with keywords, tags, names, and descriptors.

4.4.3 Functional Requirements

App should list relevant search results based on the queries.

Possible errors and responses:

If no relevant search result found, the screen displays 'no result found'.

4.5 Real-Time Chat Integration

4.5.1 Description and Priority

In the event that a donor requires more specific information from an organization or individual, a chat feature will be implemented.

This feature is medium priority.

4.5.2 Stimulus/Response Sequences

1. User clicks on the chat icon to begin a chat with another user.
2. User types their message in the text input field.
3. User clicks enter to send their message into the chat.

4.5.3 Functional Requirements

A functional interface connected with Google Firebase to provide real time messages with no delays. Privacy and encryption between chat should be maintained.

Possible errors and responses:

If there is no end user to receive message an email notification is sent to receiver party.

4.6 Location Display for a Specific Non-Profit Organization

4.6.1 Description and Priority

This feature will create a map which is displayed on every non-profit organizations profile. This map will have an icon indicating where the drop-off point for that organization is.

This feature is of low priority.

4.6.2 Stimulus/Response Sequences

1. The user searches for an organization.
2. The user selects an organization from the search results.
3. The map is displayed on the profile of the organization.

4.6.3 Functional Requirements

The user's device should be able to support the use of Google Maps.

Possible errors and responses:

1. If the device cannot support Google Maps, then the map shall be replaced with the text "Your device does not support Google Maps".
2. If a network issue does not allow the map to load properly, then the map shall be replaced with the text "Please check your network connection and try again."
3. If the non-profit organization does not collect donations at a drop off point, then no map will be displayed.
4. If another error not defined here occurs while loading the map, the map shall be replaced the text "An unexpected error has occurred."

4.7 View Local Non-Profits

4.7.1 Description and Priority

This filter feature allows the user to view all local nearby non-profit organizations. The locations shall be displayed on a map with numbered icons corresponding to each non-profit. The icons shall be numbered in order from shortest distance to the user to the largest. Underneath the map will be a list of the non-profits and their corresponding numbers.

The total number of non-profits displayed shall be limited to a specified kilometer radius, which the user is able to change. This should be changed using a slider, which has a range from one to one hundred kilometers, and a default value of ten kilometers.

The user shall be able to select a nonprofit both on the map and on the list.

Selecting a non-profit will send the user to that non-profit's profile.

This feature is of medium priority.

4.7.2 Stimulus/Response Sequences

1. The user checks off "Find Nearby Organizations".
2. The user selects a range using the slider.
3. The user hits "Search"
4. If there is text in the search bar, the search engine should search through the database first for the other terms, and then narrow this list down by only returning non-profits who are within the range specified.
5. If there is no text in the search bar, the search engine will display all non-profits within the range specified.

4.7.3 Functional Requirements

This feature requires the search engine feature to be implemented first. There are no errors listed for this requirement because if the search engine is not implemented it will be impossible to search in the first place.

This feature also requires that the user's device supports Google Maps, and that the user has allowed locations services to be accessed by the app.

Possible errors and responses:

1. If the device cannot support Google Maps, then the map shall be replaced with the text "Your device does not support Google Maps".
2. If a network issue does not allow the map to load properly, then the map shall be replaced with the text "Please check your network connection and try again."
3. If the non-profit organization does not collect donations at a drop off point, then no map will be displayed.
4. If another error not defined here occurs while loading the map, the map shall be replaced with the text "An unexpected error has occurred."
5. If no non-profits are returned from the search then there will be no map or list displayed, this are replaced with the text "No Results Found".
6. If the user has not allowed the app to use location services, the app will open a pop-up with the text "Allow the use of location services to view nearby non-profit organizations". If the user declines to allow the app to use location services, then no search will be carried out and the user will remain on the same page they were on before they opened the search.

4.8 'Sign-Up' for Specific Types of Donation or Non-Profit

4.8.1 Description and Priority

This feature will allow the user to sign up to be notified when a specific non-profit adds a request for some items.

Each non-profit's profile will have a checkbox with text beside it which says, "Sign-Up for Notifications". When clicked, the user will receive notifications whenever this non-profit adds a request for items. If the checkbox is not selected, then no notifications will be sent.

The user can also sign up to be notified when specific items are in need, such as canned goods or clothing. This can be done in the settings menu. There will be an option called "Notifications" that when opened opens a list of the different kinds of donation with checkboxes beside them. Above this list will be a slider that says "Distance: ", and options from one to one hundred kilometers. This slider will choose how close the non-profit has to be to the user in order for them to receive notifications. This will be saved when the user clicks the "Update Changes" button. If the user clicks the "Cancel" button than no changes will be made.

When a donation request which matches what the user has signed up for is added, then the user will receive a notification to their phone, for more information on notifications see the section below. The app's home screen will also display a list of the corresponding donation requests so the user can easily find them once they open the app.

4.8.2 Stimulus/Response Sequences

1. User goes to a non-profit's profile.
2. User selects "Sign-Up for Notifications".
3. User goes to Settings.
4. User selects "Notifications".
5. User selects a distance using the slider.
6. User selects some types of donation to receive notifications for.
7. User selects either "Update Changes" or "Cancel".

4.8.3 Functional Requirements

The user's device should allow the app to access location services.

Possible errors and responses:

If the user has not allowed the app to use location services, the app will open a pop-up with the text “Allow the use of location services to receive notifications from nearby non-profit organizations”. If the user declines to allow the app to use location services, then no changes will be made to the notifications added.

4.9 Notifications

4.9.1 Description and Priority

This feature will send the user’s phone a notification. This feature will be used for sending notifications that the user has signed up for (see Feature 4.8), and also when a nearby non-profit sends out a request marked as “URGENT”. This feature can be disabled in the Settings. Non-profits are considered nearby if they are within ten kilometers of the user’s current location.

4.9.2 Stimulus/Response Sequences

1. A non-profit adds a new donation request.
2. If this non-profit is within the kilometer range specified and matches with the user’s criteria, then a notification is sent to that user’s device.

4.9.3 Functional Requirements

This feature requires that the user allows the app to send them notifications.

Possible errors and responses:

If the user has not allowed notifications, then their device will not receive any until they go to their settings and allow the app to send notifications.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The following performance criteria will be the primary focus of this software development project:

5.1.1 System Responsiveness

On supported devices, the application should load and be ready for use within 2 seconds. To achieve this goal, it will be important to separate the user interface from any network or database communications by using threading and caching techniques. The user interface should be constructed in a manner which prevents the application from feeling sluggish. Therefore, each user interaction should update the interface within the 100 to 200ms threshold. For more costly or greedy interactions, it is important to indicate to the user that the application is still operating. In these instances, a loading bar or spinner may be utilized.

5.1.2 Workload

In order to meet the estimated initial demand for the application, the service should be built to support a minimum of 10,000 users with 100 simultaneous connections. With this as a consideration, database operations and network transmissions should be fine-tuned to ensure efficient communication.

5.1.3 Scalability

As the application grows in popularity, the userbase will be to interact with the database and networking systems in higher volumes. Ensuring that the response time requirement is still met during this process, the application will be built on a scalable platform such as Firebase. Further improvements to the database and network communications will need to be realized during the development process so that this growth in user-count can be accommodated. It will be at this stage in development where simultaneous connection limits and user-storage requirements will need to be raised.

5.1.4 Equal performance on different devices

When considering the diversity of Android mobile devices, the application should be built in a manner which prevents degradation when being used on lower end devices.

5.2 Safety Requirements

All data retained by app should be handled with user privacy in mind. Privacy for the users in their messages and contact details should be maintained and passwords stored on firebase will not be visible to admins.

5.3 Security Requirements

Proper security protocols will be essential to maintain privacy for all users. All login and location data will be protected through proper encryption. Only account administrators will have the ability to alter the information provided on the accounts main page. All chatrooms will only be accessible to preauthorized participants.

5.4 Software Quality Attributes

The following software quality attributes will be the primary focus for this software development project:

5.4.1 Availability

The app will be used by a variety of users and user types, all on different schedules. Functions of the app must be available at all times to facilitate the most extensive use.

5.4.2 Reliability

Some content on the app will be time sensitive, thus it is important for there to be extremely limited or no downtime.

5.4.3 Usability and Reusability

Technical experience should not be a limiting factor for using the app as users will likely have varying levels of technical expertise. Thus, the navigation and utilization of the app should be straightforward and clear. The need for charitable action is ever-present, therefore the app will need to attract repeat users. To facilitate this, user experience must be a positive one.

5.4.4 Concurrency

The app will need to be able to handle a large number of concurrent users as there is a large potential for groups of users to be accessing the same functions at the same time. Examples of this include live chatrooms, viewing profiles, and database queries.

5.5 Business Rules

App users will be binding by terms and conditions of app such as Retention of Right to Terminate/Ban Users, Retention of the Right to Change the App at Any Time and others. This app is strictly for non-profit organizations and charitable individuals. There will be no monetary or sale conditions attached to any item advertised on the app. As such, there are no charges associated with registration or participation on the platform for both organization and individual type accounts.

6. Design Diagrams

6.1 Use Cases View



6.1.1 Account Registration

Use Case Title	Account Registration
Description	User is prompted to create an account when starting the application for the first time. A user can use the app once the account is registered and profile details are filled.
Actor(s)	Donor, Organization, Receiver, Admin
Preconditions	The app is started by new user who does not have an account.
Postconditions	The user gets access to the benefits of the app.
Main Success Scenario	<ol style="list-style-type: none"> 1. User opens the app 2. User creates an account using email address and password. 3. Additional details are requested from user to create user profile. 4. User submits the request to create a new account and a success message is shown.
Alternative Step	<ol style="list-style-type: none"> 1. User exits the app. 2. User enters invalid email or characters. App informs the user of their error. 3. User enters the already registered email address. App informs the user of their error.

6.1.2 Account Login

Use Case Title	Account Login
Description	A user may log onto their account to access their posts and profile. The user will be prompted for their account information, which will be subsequently cross-referenced against the user database.
Actor(s)	Donor, Organization and Receiver
Preconditions	The user must have an existing account and on login page.
Postconditions	User has successfully logged in.
Main Success Scenario	<ol style="list-style-type: none"> 1. User selects the login screen. 2. User inputs their login information in the correct fields. 3. Login manager verifies the login information exists in the user database. 4. Upon successful verification, the user is logged into their account.
Alternative Step	<ol style="list-style-type: none"> 1. User exits the app. 2. User enters invalid email or characters. App informs the user of their error.

6.1.3 Delete User/Post

Use Case Title	Delete User/Post
Description	If a user post non related items or explicit material, admin can delete the post or user due to use case violation.
Actor(s)	Admin
Preconditions	A registered user violated the use case of the app.
Postconditions	Profile/post is deleted to reflect changes made by the admin.
Main Success Scenario	Admin uses firebase to delete the user.
Alternative Step	User is warned for potential removal.

6.1.4 Update Profile

Use Case Title	Update Profile
Description	Users can alter the information that appears when another user views their profile.
Actor(s)	Donor, Organization, Receiver, Admin
Preconditions	The user must have an existing account, on update profile page and be logged in.
Postconditions	Profile is updated to reflect changes made by the user.
Main Success Scenario	<ol style="list-style-type: none"> 1. User navigates to their profile and selects the edit button 2. User makes desired changes to their profile 3. User saves the changes using the save button
Alternative Step	User enters invalid information or leave some or all of the fields empty. App informs the user of their error.

6.1.5 Create Post

Use Case Title	Create Post
Description	Users can create and submit a campaign or their donation items according to their profile.
Actor(s)	Donor, Organization, Receiver
Preconditions	User must be logged in and on home page.
Postconditions	User is directed back to the main page after the post is submitted.
Main Success Scenario	<ol style="list-style-type: none"> 1. This use case is triggered when user clicks on plus icon which represents add post. 2. The user then proceeds to enter the required details into the fields. 3. Once all details such as description, drop-off, pickup are filled the user can submit the post for other users to see.
Alternative Step	User enters invalid information or leave some or all of the fields empty. App informs the user of their error.

6.1.6 Search – Organization

Use Case Title	Search Organizations
Description	Users can search organizations to which they want to donate or to receive from to get more information.
Actor(s)	Donor, Receiver
Preconditions	User must be logged in and on the search bar.
Postconditions	The results are shown to user.
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks on search bar from home page and enters the name or other detail of the organization. 2. User clicks on search button.
Alternative Step	App has no valid organization with the search name and displays 'No results found' message.

6.1.7 Search – User

Use Case Title	Search Users
Description	Users can search other users to which they want to donate or to receive from to get more information.
Actor(s)	Donor, Receiver, Organization
Preconditions	User must be logged in and on the search bar.
Postconditions	The results are shown to user.
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks on search bar from home page and enters the name or other detail of the organization. 2. User clicks on search button.
Alternative Step	App has no valid user with the search name and displays ‘No results found’ message.

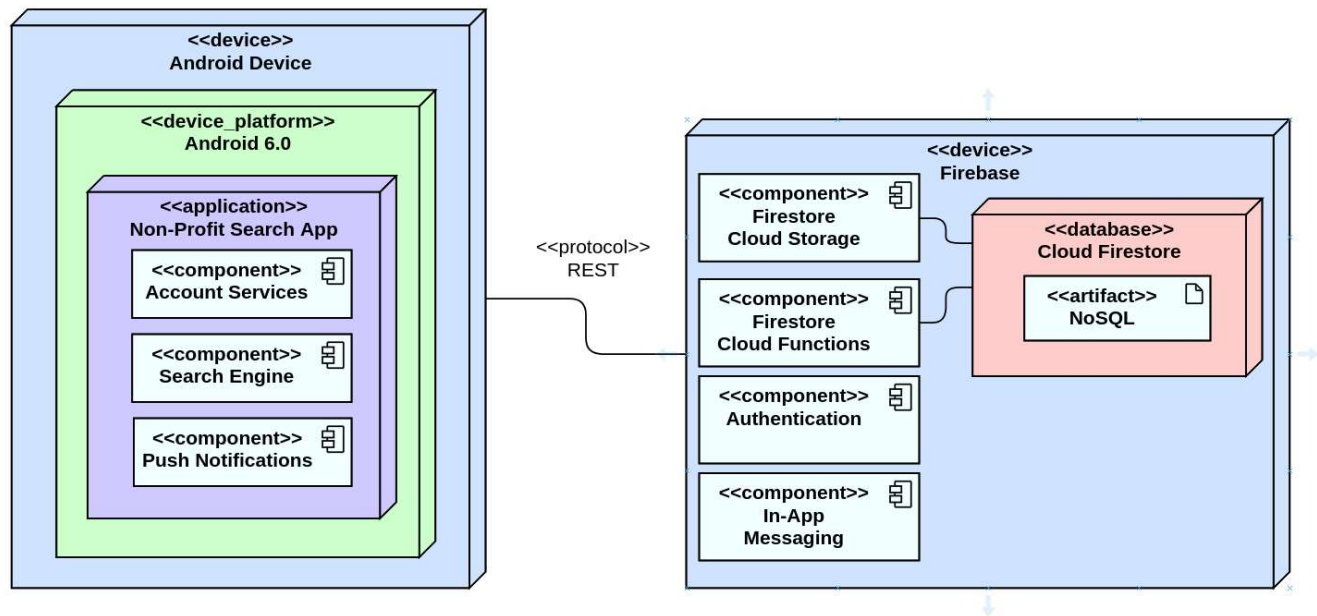
6.1.8 Filter Posts – Category

Use Case Title	Filter Posts by category
Description	Users can filter or sort posts according to his preference. The filters include location filter, donation cause filter, urgent.
Actor(s)	Donor, Receiver and Organization
Preconditions	User must be logged in, on search bar and applied features must be valid for result.
Postconditions	If there are posts to show, the posts are shown.
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks on search bar from home page and click on filters button. 2. User selects the desired filters. 3. User clicks on apply filter button.
Alternative Step	App has no valid filter results and displays ‘No results found’ message.

6.1.9 Messaging

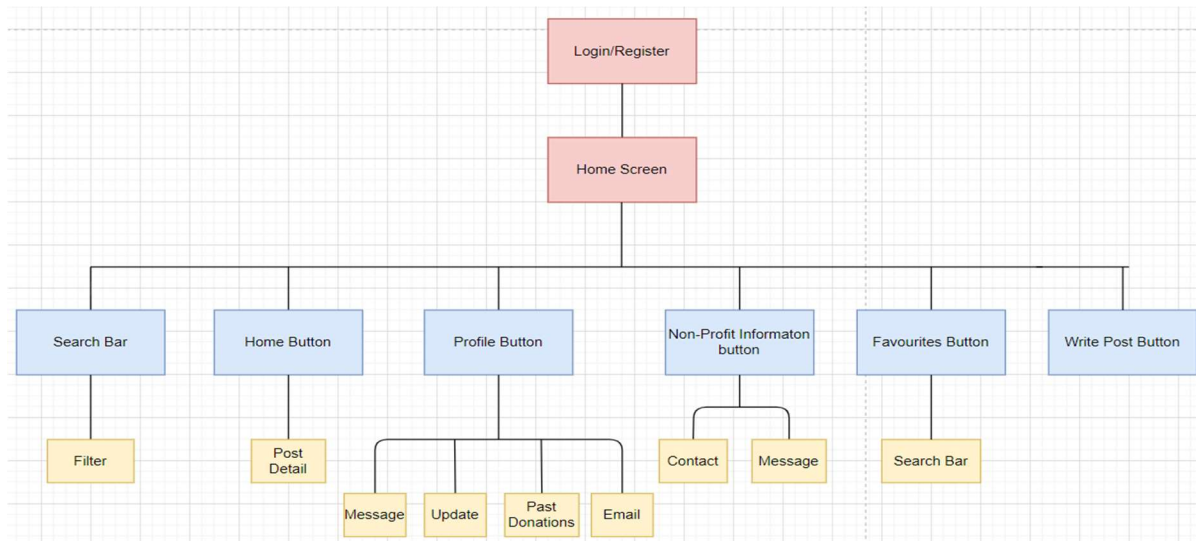
Use Case Title	Messaging
Description	Users may send private messages to other users to facilitate further discussion based on user posts or otherwise.
Actor(s)	Donor, Receiver, Organization
Preconditions	User(s) must be logged in
Postconditions	Messages must appear in a user's message tab and must be private to all non-associated users.
Main Success Scenario	A user selects the message icon on another users profile and records their message in the text field. When the message is complete, it is sent to the receiving user.
Alternative Step	User selects and existing message log to continue a previous dialog.

6.2 Deployment Diagram



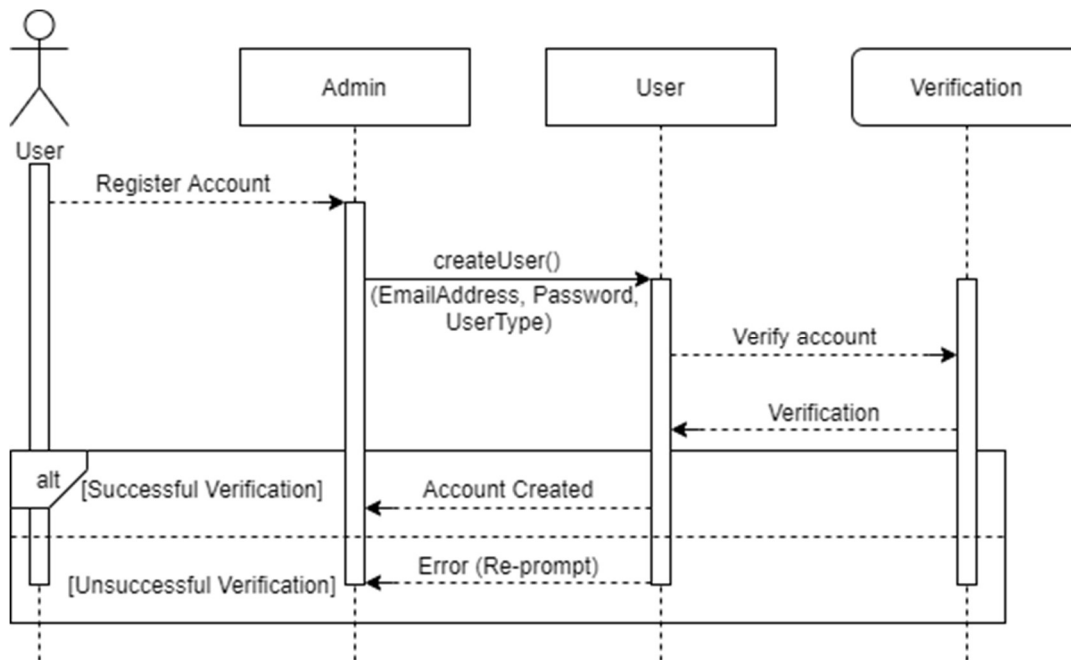
6.3 Logical View

Overall Logical view which provides the end user functionality and key abstractions.

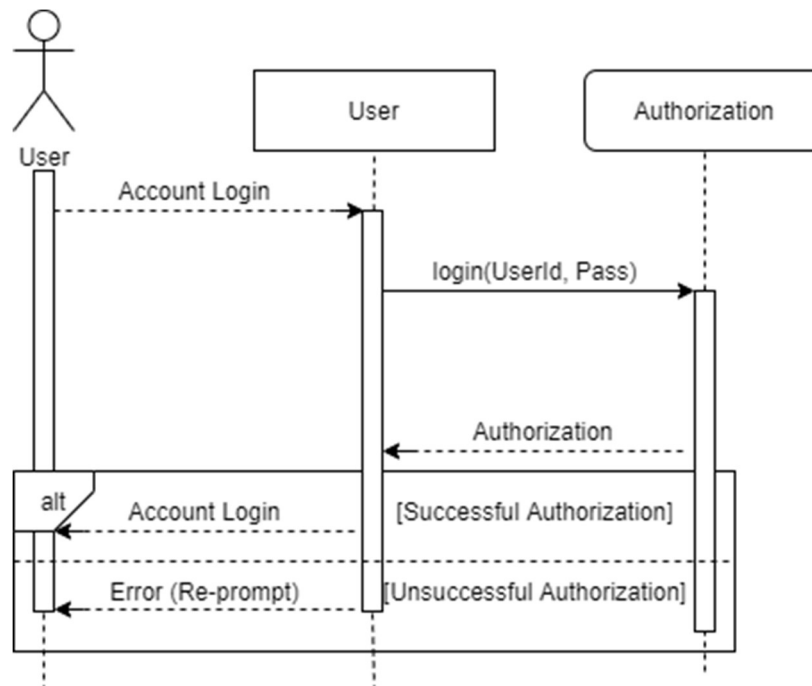


6.4 Sequence Diagrams

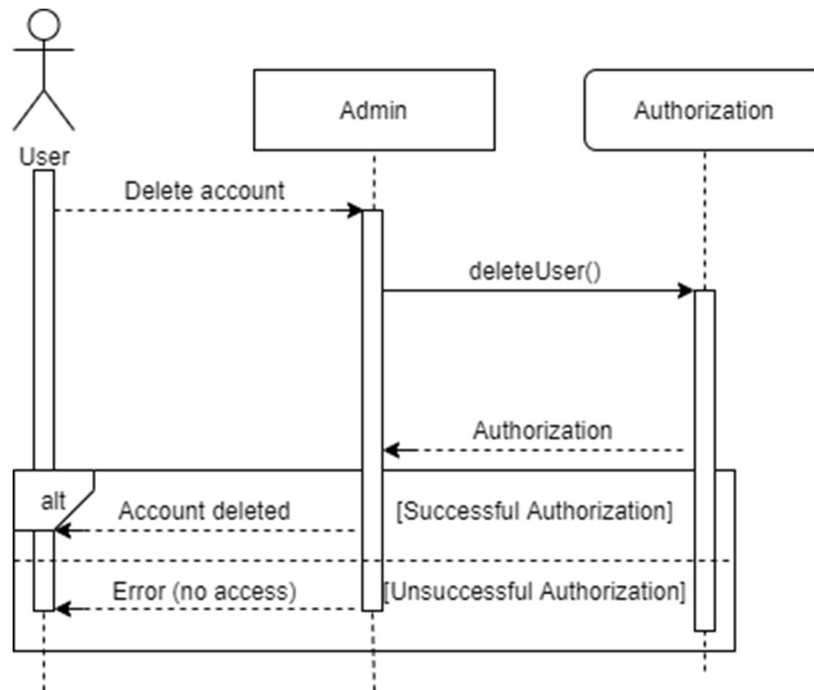
6.4.1 Account Registration (Xref: 6.1.1)



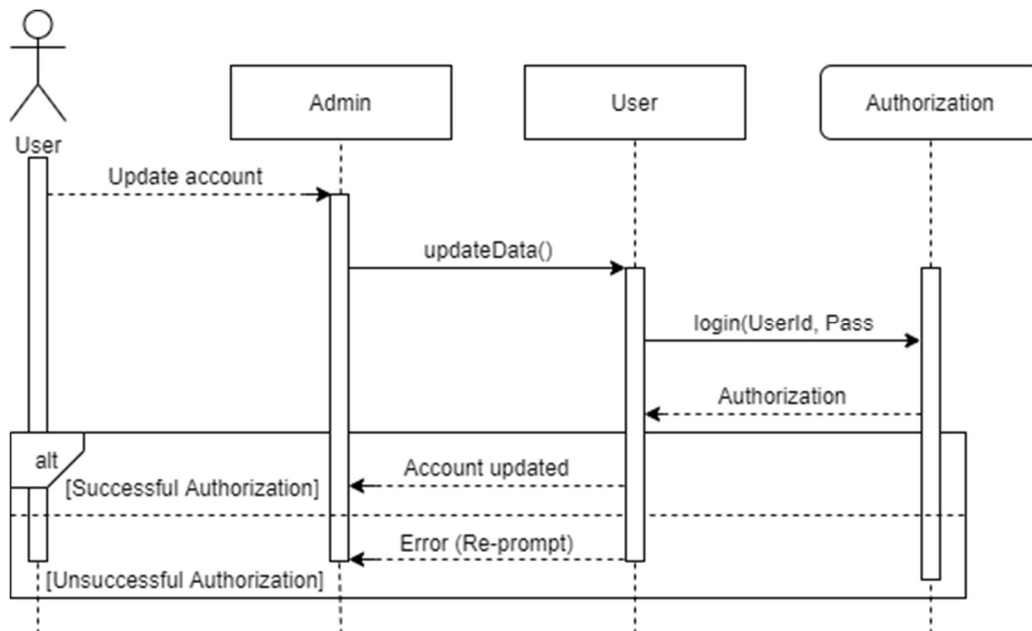
6.4.2 Account Login (Xref: 6.1.2)



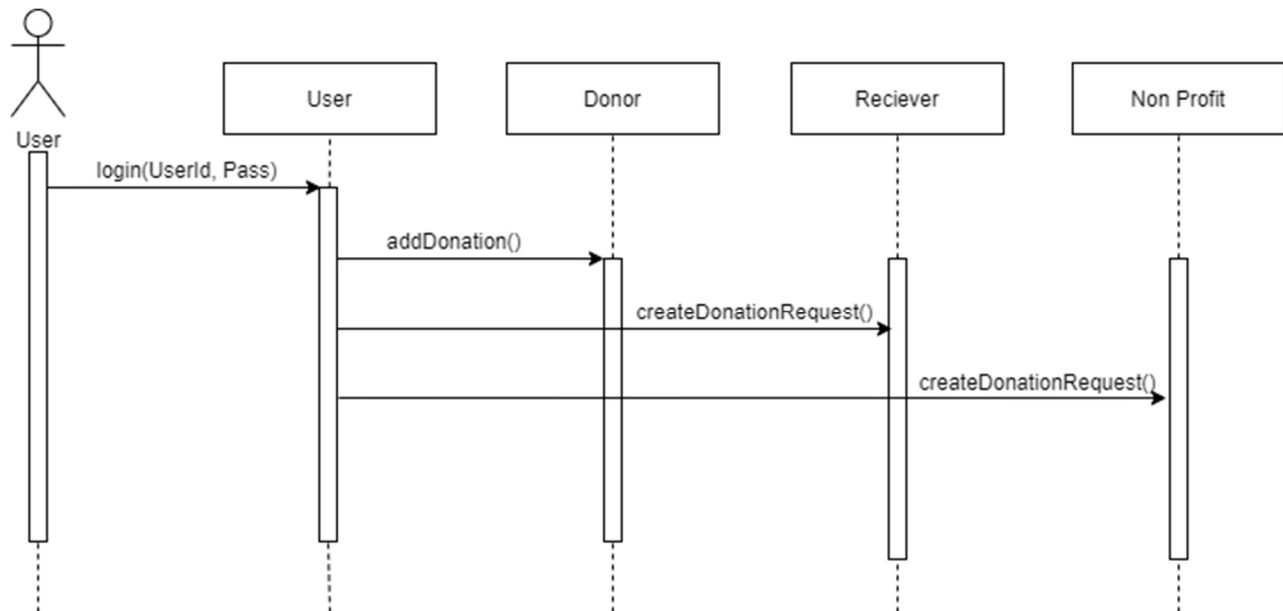
6.4.3 Delete User (Xref: 6.1.3)



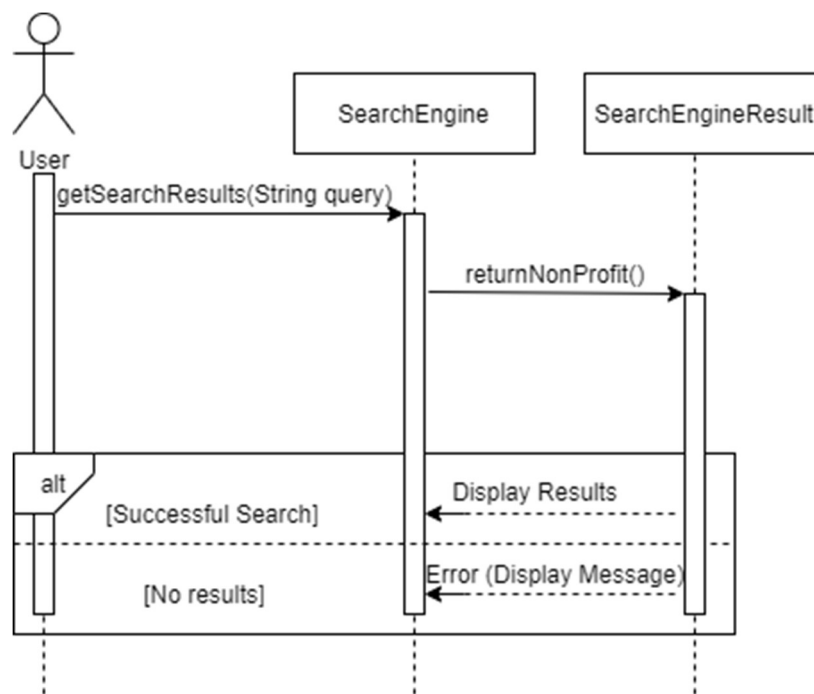
6.4.4 Update Profile (Xref: 6.1.4)



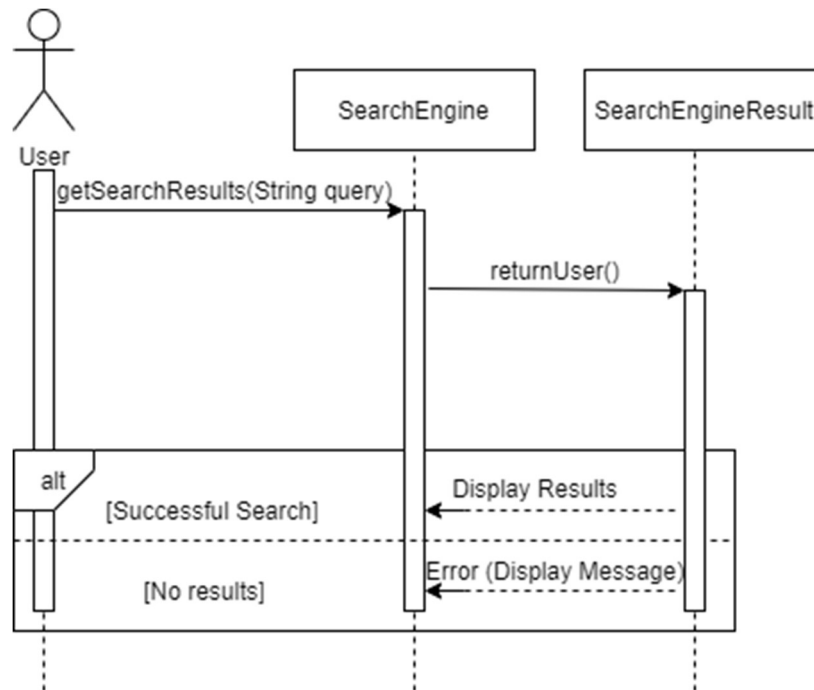
6.4.5 Create Post (Xref: 6.1.5)



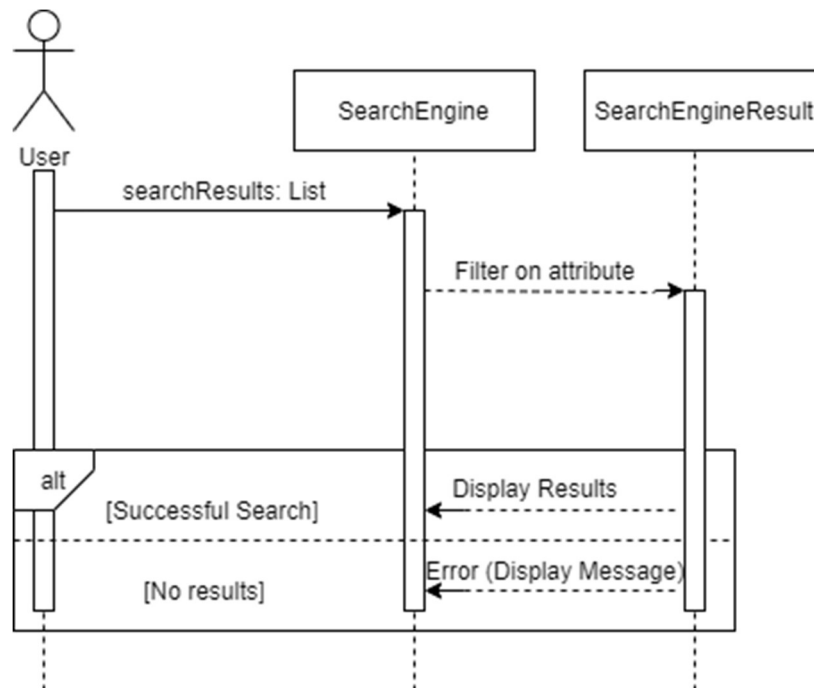
6.4.6 Search – Organization (Xref: 6.1.6)



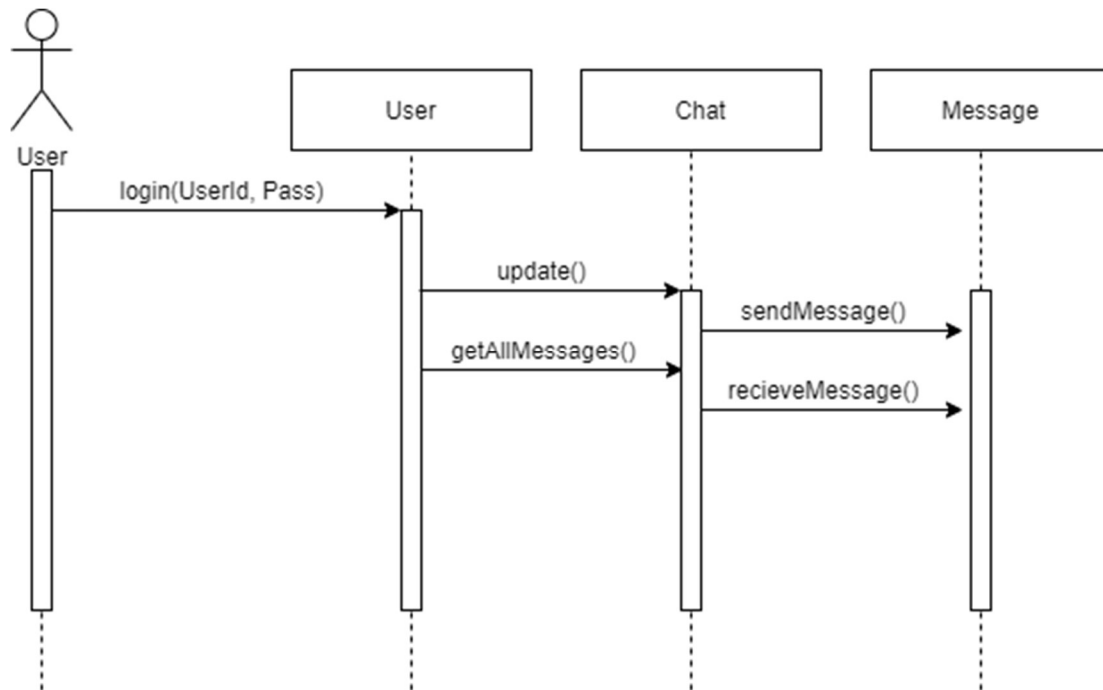
6.4.7 Search – User (Xref: 6.1.7)



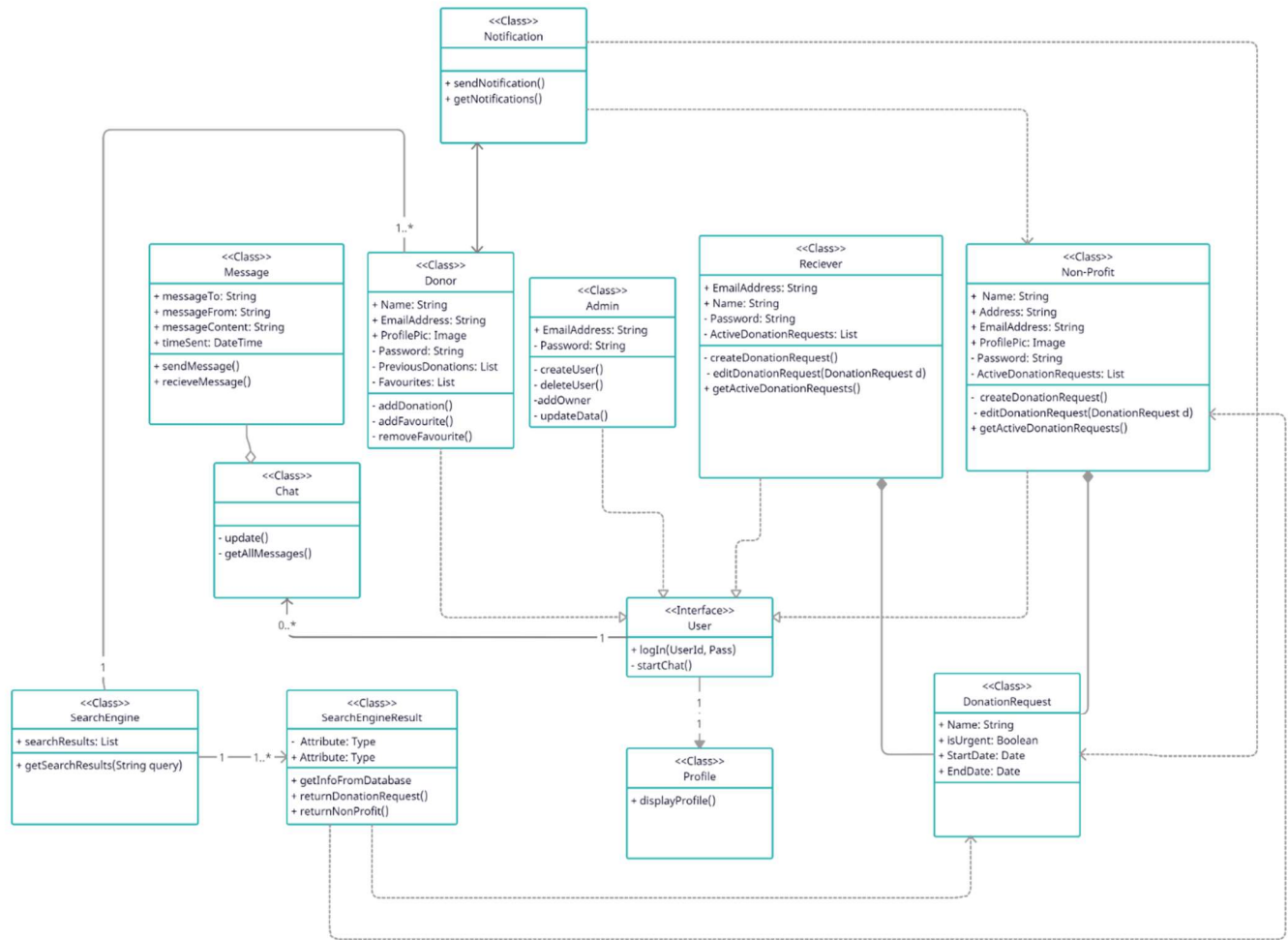
6.4.8 Filter Posts (Xref: 6.1.8)



6.4.9 Messaging (Xref: 6.1.9)



6.5 Class Diagram



6.6 Database Diagram

