Department of Computer Science

The University of Hong Kong

COMP 4801 Final year project

Interim Report

Group:

fyp23029

Project Title:

All-in-one mobile application for elderly

Group Member (UID):

Ng Tsz Hei (3035780192)

Supervisor:

Professor Wu, Chuan

Date of Submission:

21, January 2024

I) Abstract

It has recently been observed that the number of telephone deception cases in Hong Kong is rising. Within the recorded cases, a significant portion of the victims are elderly. Additionally, incidents of missing and accidents involving the elderly are concerning events that society should address. To mitigate these problems, developing specialized software that better protects the personal and property safety of the elderly is an effective solution. However, existing software in the market is not tailored for the elderly and does not meet their needs. This paper aims to document and demonstrate the entire software development lifecycle of an all-in-one smartphone application for the elderly, from preliminary design to implementation and testing. The deliverables include the application itself, its user interface design, database design, system architecture, and test case design. The main features to be achieved are location tracking, body status monitoring, fall detection, telephone deception prevention, and GPT integration. With these features, the safety of the elderly will be guaranteed, and their lives could become easier. At the current stage of the project, the preliminary design of the application has been completed. The next step is to start the implementing and testing stages of the project.

II) Acknowledgement

I would like to express my deepest gratitude to my supervisor, Professor Wu Chuan, from the Computer Science department of HKU, for her invaluable advice and guidance throughout the project. Without her assistance, this paper would not have been successfully and smoothly completed.

I would also like to acknowledge Mr. Matthew Anderson from the Centre for Applied English Studies of HKU for his language and writing support. With his assistance, the paper has been written in a more structured and professional manner.

Table of Contents

III) List of Figures

IV) Abbreviations

Abbreviation	Definition
UI	User Interface
IDE	Interactive Developing Environment
GPT	Generative Pre-trained Transformer
API	Application Programming Interface
UML	Unified Modelling Language

1. Project Introduction

This section will emphasize the introduction and relevant information regarding this project, and it consists of five parts. Section 1.1 will focus on the background of this project, while Section 1.2 will discuss the motivation behind it. Section 1.3 will highlight the current research gap in this topic, and Section 1.4 will state the objectives and deliverables of this project. Lastly, Section 1.5 will provide an outline of the remaining report.

1.1 Project background

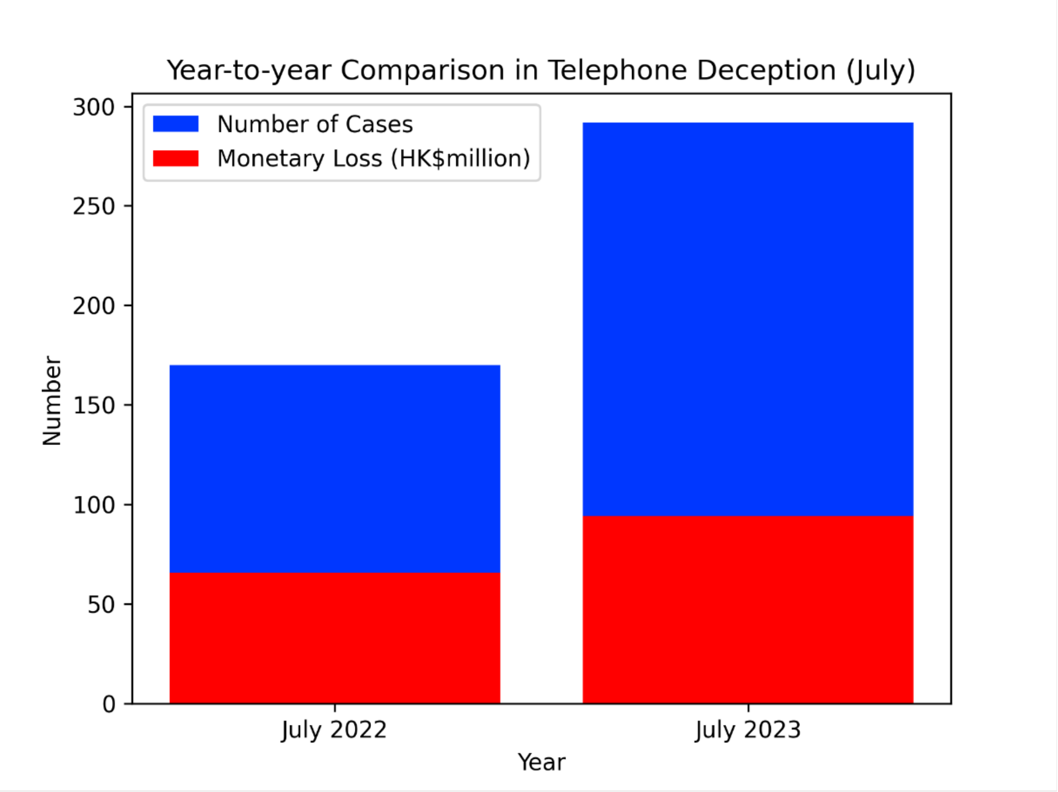


Figure 1 Year-to-year comparison in telephone deceptions' figures (July) [1]

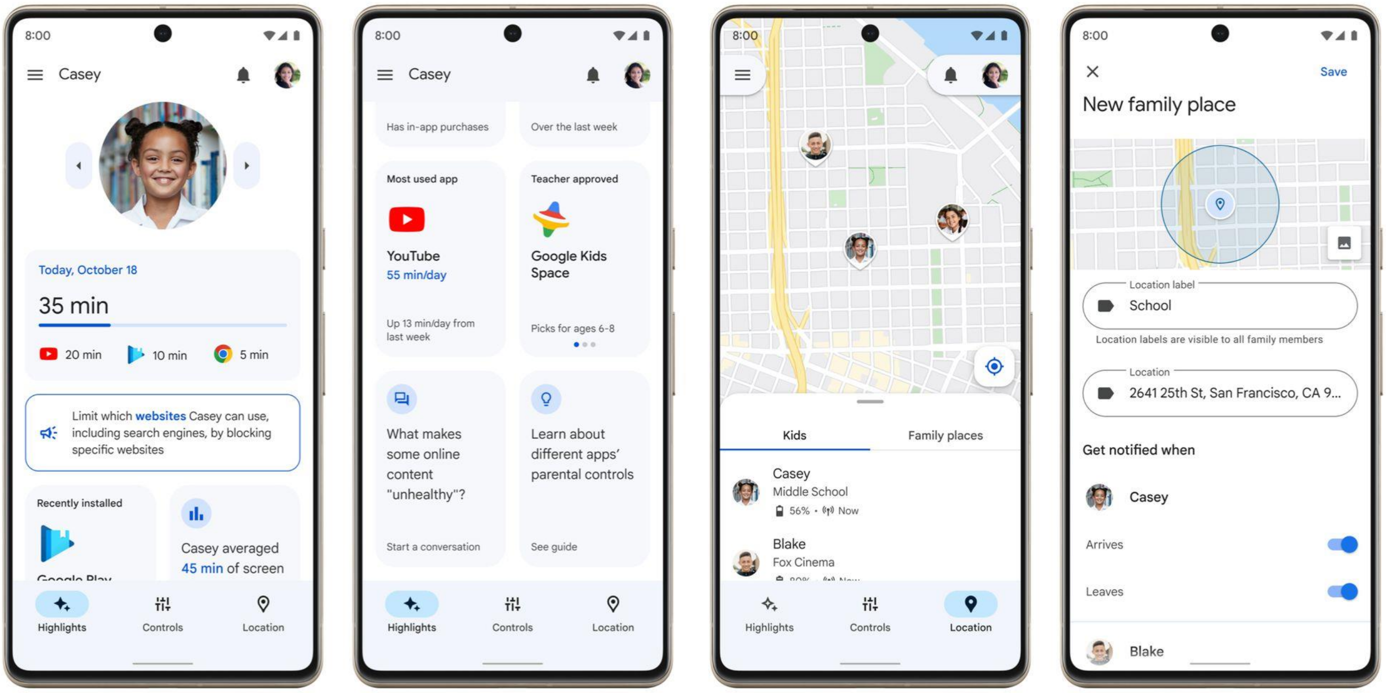
Figure 1 shows a year-to-year comparison of telephone deceptions in Hong Kong provided by the Anti-Deception Coordination Centre of the Hong Kong government. The figure indicates a rise in the number of telephone deception cases from 170 to 292, representing a 71.76% growth, and a rise in monetary loss from 65.65 million to 94.2 million, with a 43.49% growth, from July 2022 to July 2023 [1]. This highlights the surge in the frequency and monetary losses of telephone deception in recent years, necessitating the need for action to address the worsening situation.

In addition, according to the *Legislative Council Panel on Security Initiatives for Preventing and Combating Deception Cases*, there are two prevalent modi operandi of telephone deception known as "Pretend Officials" and "Guess Who." In 2020, "Pretend Officials" cases accounted for approximately 57% of all telephone deception cases (a total of 1150 cases), with 30% of the victims aged 61 or above. On the other hand, "Guess Who" cases accounted for 43% of the cases, with 60% of the victims aged 61 or above [2, p.4]. Overall, more than 40% of telephone deception victims are elderly individuals aged 61 or above, which represents a significant portion. This concerning trend highlights the need for extra awareness and measures to address the issue.

1.2 Project Motivation

In addition to telephone deception, the physical vulnerability of the elderly and their susceptibility to geriatric diseases such as Alzheimer's disease and dementia, which cause memory and cognitive decline, are concerning factors. Elderly individuals going missing or experiencing accidents, such as fainting and falling, are events that require our attention. Furthermore, the elderly may need assistance in problem-solving for daily life challenges and integrating into a rapidly changing society. However, their family members may not always be available to provide help. It would be preferable to have someone available to offer guidance and support to the elderly, such as providing advice on traveling to specific destinations and helping them bridge the gaps they may face in society.

1.3 Research Gap



8:00
Casey
Today, October 18
35 min
20 min 10 min 5 min
Limit which websites Casey can use, including search engines, by blocking specific websites
Recently installed
Google Play
Casey averaged 45 min of screen

8:00
Casey
Has in-app purchases
Over the last week
Most used app
YouTube
55 min/day
Up 13 min/day from last week
Teacher approved
Google Kids Space
Picks for ages 6-8
What makes some online content "unhealthy"?
Start a conversation
Learn about different apps' parental controls
See guide

8:00
Kids
Family places
Casey
Middle School
56% · Now
Blake
Fox Cinema

8:00
Save
New family place
Location label
School
Location labels are visible to all family members
Location
2641 25th St, San Francisco, CA 9...
Get notified when
Casey
Arrives
Leaves
Blake

Figure 2 User interface of Google Family Link App [3]

As solutions to the aforementioned problems, several smartphone applications already exist in the market. For instance, Whoscall is an unknown call identification application that utilizes data science to analyze a large volume of call data and create an optimal model for call identification [4]. Google Family Link is a parental control application that offers a location tracking function and allows parents (or family members) to manage the digital devices and accounts of their children (or elderly relatives) [5]. Apple Health is an application that connects to smartwatches and organizes users' important health information, which can be used to monitor the health conditions of the elderly [6].

However, these applications are not specifically designed with the elderly in mind, and their design language may not cater to their specific needs. For example, Figure 2 displays screenshots of the Google Family Link app [3], where the user interface appears complicated with an abundance of information. Additionally, the small text size may make it difficult for the elderly to comfortably read the content. These issues can create challenges for the elderly in learning and using these applications. Moreover, managing multiple applications simultaneously can be problematic for elderly users. Therefore, an all-in-one smartphone application that is specifically designed to be elderly-friendly would be a better solution to help them address the aforementioned problems.

1.4 Project Objective and Deliverables

The goal of this project is to develop an all-in-one, cross-platform mobile application called Elder Link, that is designed specifically for the elderly to address the issues mentioned above, enhance their personal and property safety, and improve their quality of life. To achieve the goal, there are 5 main features to be implemented and they are as follows:

1) A calls filtering feature to identify suspicious call and prevent telephone deception.
2) A location tracking feature to assist searching in case of elderly missing.
3) A body status monitoring feature to detect dangerous situation like high blood pressure and prompt the family members to take emergent action.
4) A fall detection feature to detect falling cases and notify family members to take subsequent action.
5) A GPT chatbot feature to provide personalized support to the elderly.

The deliverables of this project will include the application itself, as well as related documentation such as user interface design, database design, system architecture, and test case design.

1.5 Outline of remaining report

The subsequent sections of this paper will concentrate on the complete software development process of the application. Section 2 will discuss the methodology employed to accomplish the project. Section 3 will cover the current results, along with the accompanying discussion. In section 4, expected results in the coming semester and the remaining work plan will be discussed. Finally, in section 5, a conclusion will be provided, summarizing the project and outlining the next steps.

2. Project Methodology

This section will emphasize the methodology utilized to complete this project, which is the tools and environment setup employed for developing the application. Section 2.1 will introduce the front-end tool and Interactive Development Environment (IDE) utilized for developing the application, highlighting their advantages. Section 2.2 will focus on the back-end setup of the application. Section 2.3 will discuss the software employed for modeling the application structure. Section 2.4 will delve into the software used for designing the user interface. Lastly, Section 2.5 will explore the software utilized for version control and project management, and section 2.6 will explore the methods used for testing.

2.1 Front End and IDE

To build the user interface of the application, React Native, a Javascript library for creating user interfaces, is used. React Native's declarative UI paradigm and JavaScript enable it to wrap native code and interact with native APIs, allowing for cross-platform development. This means that the application will run seamlessly on both Android and IOS smartphones, providing users with a consistent experience regardless of the brand of their smartphone [7].

For the IDE, Visual Studio Code will be utilized. It is a powerful text editor that offers a range of helpful features for coding, such as syntax highlighting, quick debugging, built-in Git commands, and customizable extensions based on the programming languages and services required by developers [8]. These robust and convenient features can significantly ease the development process and accelerate the pace of development.

2.2 Back End

For the database and server of the application, we will be utilizing Firebase by Google. Firebase is a robust app development platform that offers a wide range of cloud services for various backend functionalities, including database setup, server hosting, authentication, and user analytics. With Firebase, developers can benefit from high performance and stability in their applications. Additionally, Firebase ensures a high level of security, prioritizing the protection of user privacy [9].

2.3 Modeling of application structure

To create UML diagrams, specifically use case diagrams, sequence diagrams, class diagrams, and ER diagrams for the database design, Lucidchart will be utilized. Lucidchart is an intelligent diagramming application that facilitates team collaboration and communication [10].

A use case diagram provides a high-level overview of the relationships and interactions between actors (users) and the system. It helps in understanding the system's functionalities and the roles of different actors involved. [11]

A sequence diagram illustrates the order in which objects interact and represents runtime scenarios. It visualizes the flow of messages between objects and helps in understanding the dynamic behavior of the system [11].

A class diagram describes the structure of the system by modeling its classes, attributes, operations, and relationships between objects. It provides a static view of the system's architecture and aids in understanding the relationships and dependencies between different classes [12].

In addition to these UML diagrams, Lucidchart will also be used to create an ER diagram for modeling the database design, which helps in visualizing the structure and relationships of database entities.

2.4 UI prototype

Figma will be used to design the Graphical User Interface for the application. Figma is a UI design application that enables designing realistic interactive prototypes for quick iteration of flows and states. It offers high team visibility, which enables team cooperation, as well as providing tools for turning the UI prototype into real code [13] to enhance development efficiency.

2.5 Version control and project management

GitHub will be used for version control and project management. It is a website and cloud-based service designed to assist developers in storing and managing their code, as well as tracking and controlling changes to it. GitHub utilizes Git, an open-source, distributed version control system that allows each developer to have a complete copy of the codebase and its history on their own computers.

Version control is implemented through branching and merging. Branching involves creating a duplicate of the source code in a new branch, allowing changes to be made without affecting the original main branch. Merging, on the other hand, entails integrating the finished and well-tested code back into the main branch, making it official. Every branching and merging action is tracked and can be reverted if necessary, enabling a return to previous checkpoints in case critical bugs are encountered.

Furthermore, GitHub provides project management features such as progress statistics and contribution tracking [14].

2.6 Testing

The React Native Testing Library will be utilized for both unit testing and integration testing. Unit testing is a type of software testing that focuses on validating the functionality of individual units or components within a software system. Its goal is to ensure that each unit functions correctly and meets the specified requirements [15].

Integration testing, on the other hand, is conducted after unit testing and aims to verify that the interface between two units or modules is functioning correctly and transferring data accurately. This testing process ensures that no errors occur when merging different units together [16].

User acceptance testing, on the other hand, is performed manually by real end users. Its purpose is to evaluate whether the application fulfills their requirements in a real-world scenario and determines if the overall user experience is satisfactory [17]. Testing on whether the fall detection will work as expected in falling demonstrations will also be a part of the user acceptance test.

3. Current Result

This section will focus on presenting the current results achieved so far. At this stage of the project, the preliminary design of the application has been completed. Section 3.1 will describe the use cases of the application. Section 3.2 will describe the underlying execution logic for each use cases. Section 3.3 will summarize the components involved in the execution logic. Section 3.4 will focus on the database design. Section 3.5 will discuss the user interface of the application.

3.1 Use case design

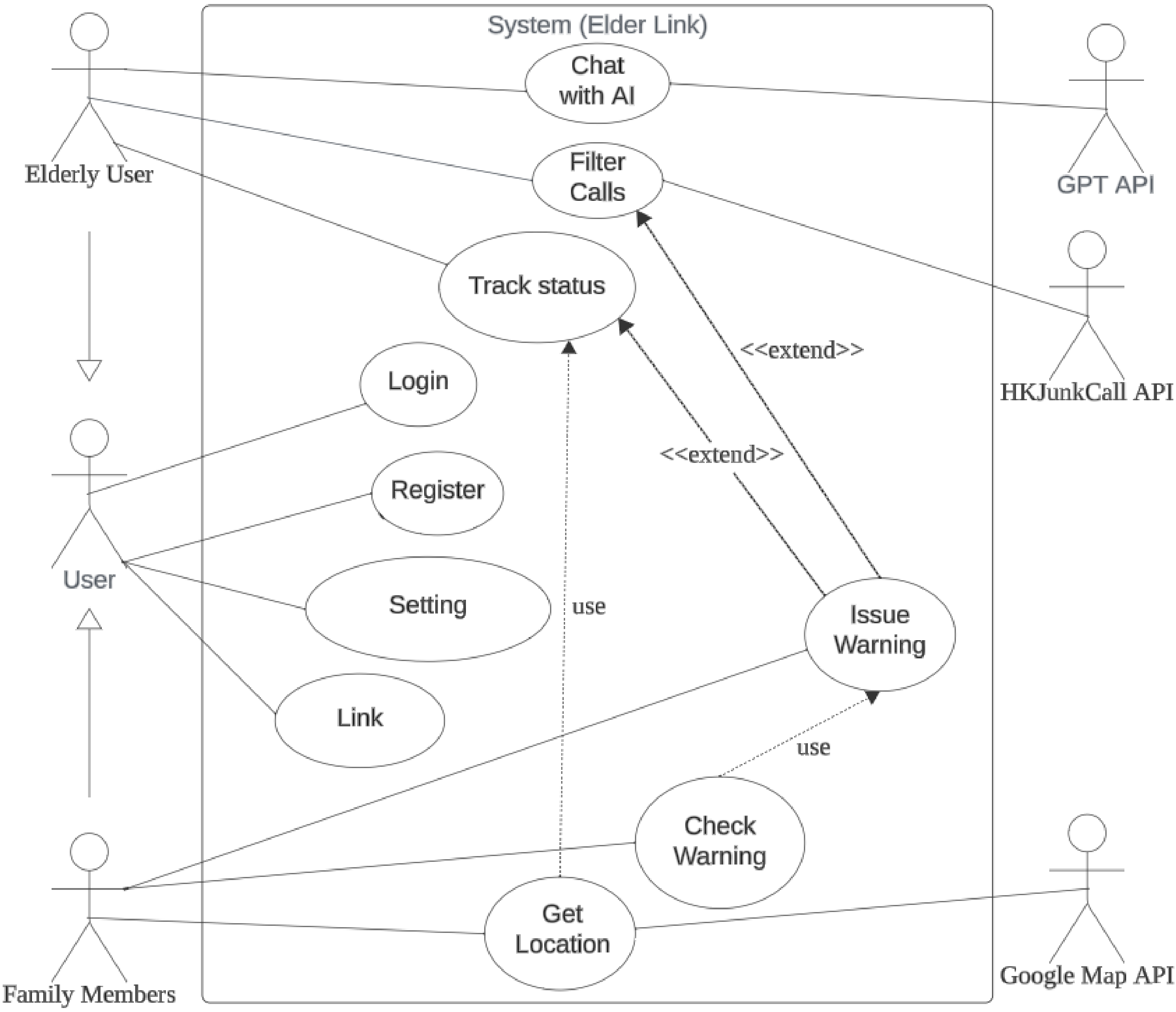


Figure 3 Use Case Diagram of Elder Link

The above diagram shows the use cases of the application. The users are specialized into two types of targeted users, namely elderly user and their family members.

Firstly, all the users are expected to perform the Register, Login, Setting, and Link use cases. Register use case allows users to register an account. Login use case allows users to access the registered account. Setting use case allow users to enable or disable the features in the application according to their own needs and set an upper and lower limit of blood pressure to trigger the body status monitoring mechanism. Link use case allows users to establish connections to other users.

Secondly, the elderly users are expected to perform the Chat with AI, Filter calls, and Track status use cases. Chat with AI use case allows elderly users to send a query to a GPT model through an external GPT API and retrieve its response as if chatting with a real person in a messenger app. Filter calls use case will be triggered automatically when there is an incoming call, and it will identify if the incoming call is suspicious by interacting with HKJunkCall API, which is an API that takes a phone number as input and return the identified type of phone number. If the call is identified suspicious, issue warning use case will be triggered, and a warning will be issued to the linked family members. Track status use case will be triggered constantly for every predefined period of time to get the up-to-date status of the elderly users. The location status will be stored and issue warning use case may be triggered if falling is detected or body status of elderly is abnormal.

Lastly, linked family members are expected to perform the Check warning and Get location use cases. Check warning use case allows family members to check previous issued warnings by elderly users. Get location use case allows family members to retrieve the location of elderly users. It also interacts with Google Map API to visualize the location.

3.2 Execution Logic

This section will focus on the underlying execution logic and sequence for each of the use case. A sequence diagram will be used to visualize and assist the presentation of the execution logic.

3.2.1 Register

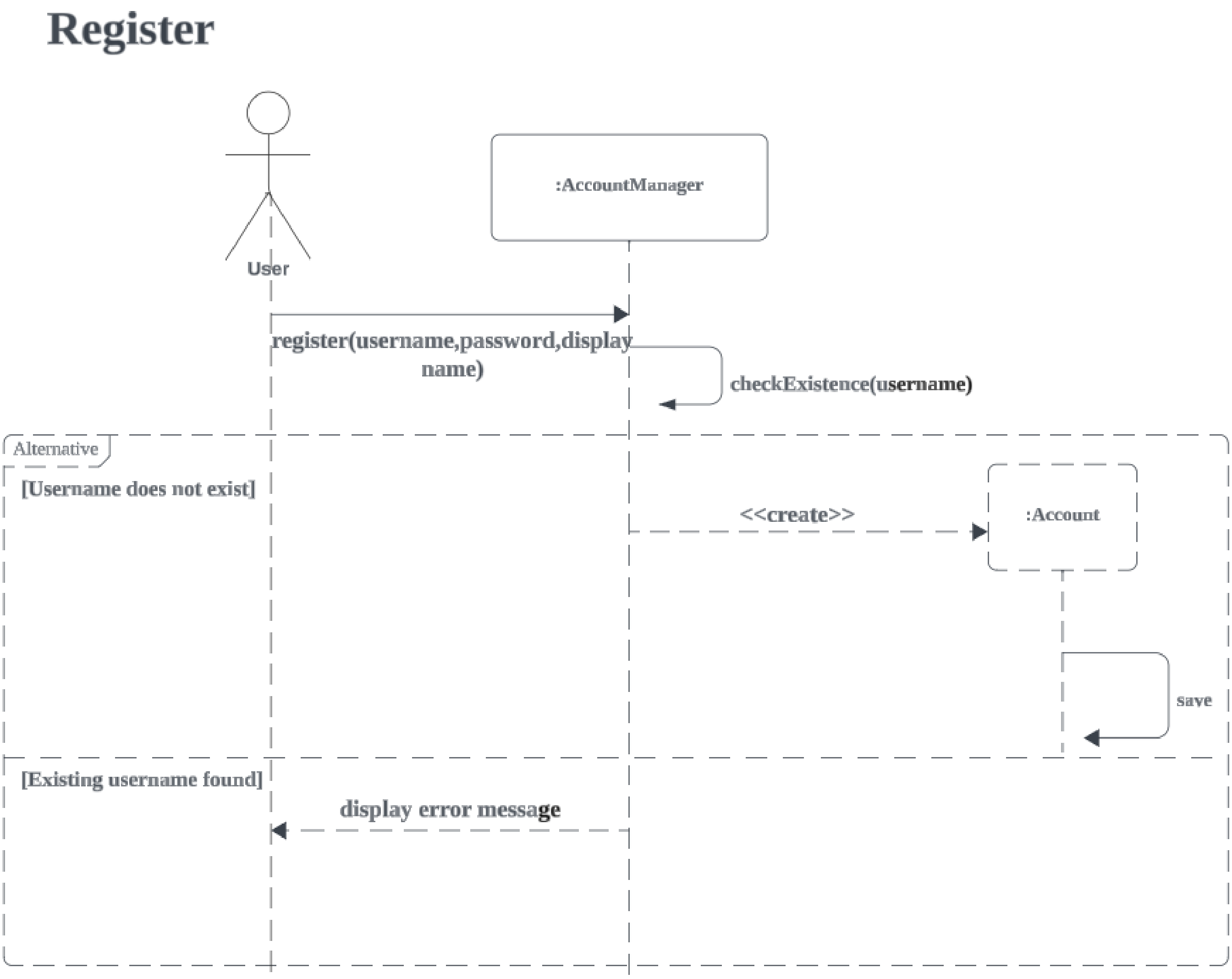


Figure 4 Sequence Diagram of register

Above is the sequence diagram of Register use case. During registration, the user will send the desired username, password and display name (name displayed to other users) to the AccountManager. Then the AccountManager will query the username in the database to check if it exists. If the username does not exist and it is valid, A new Account object is created and saved in the database by the AccountManager. If an existing username is found in the database, the registration is invalid and AccountManager will display an error message in the user interface.

3.2.2 Login

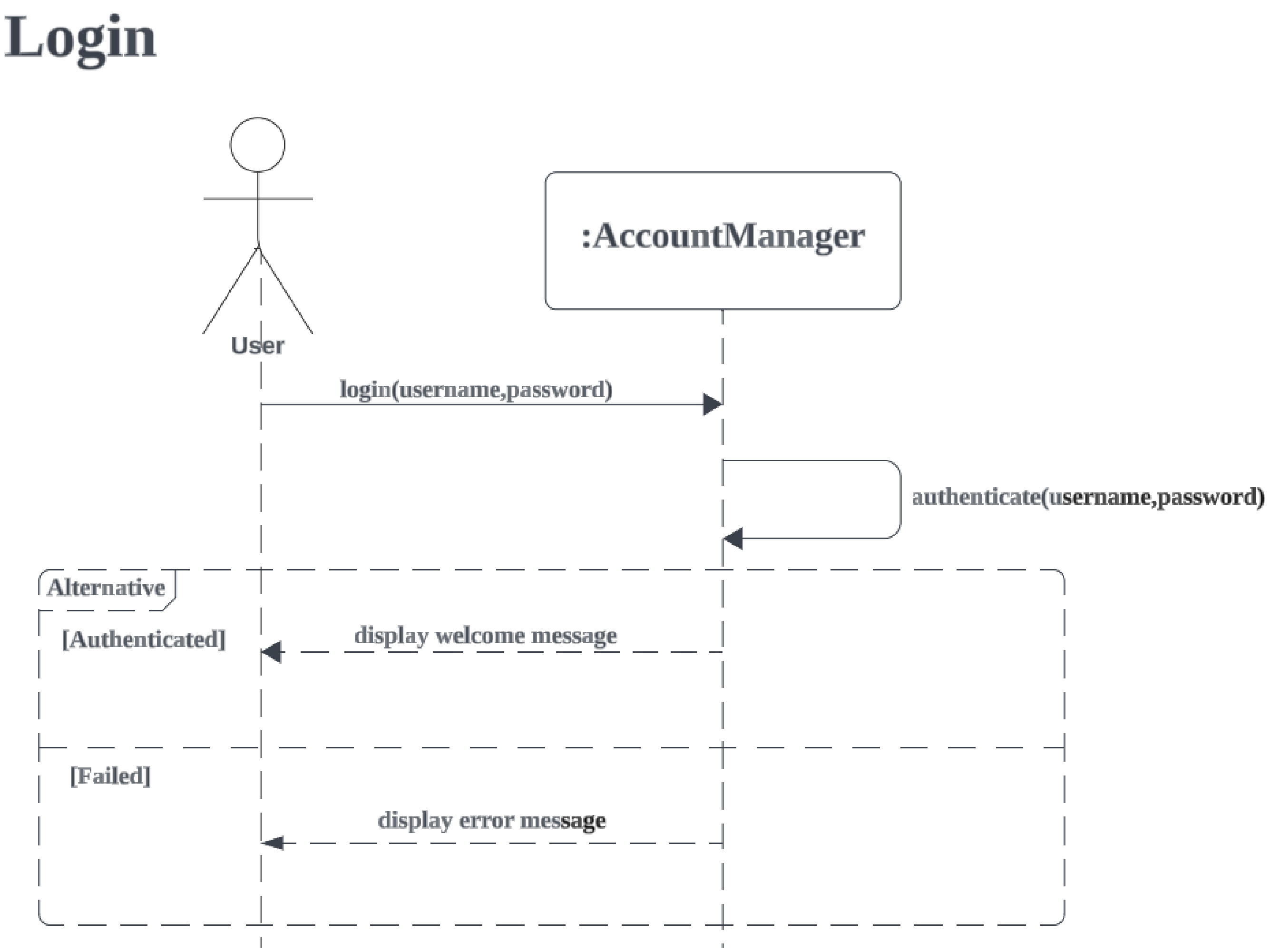


Figure 5 Sequence Diagram of Login use case

Above is the sequence diagram of login use case. During login, user will first send their username and password to the AccountManager. Then the AccountManager will query and authenticate the username and password in the database. If authentication succeeds, AccountManager will display a welcome message to the user. If authentication fails and the login is invalid, AccountManager will display an error message.

3.2.3 Setting

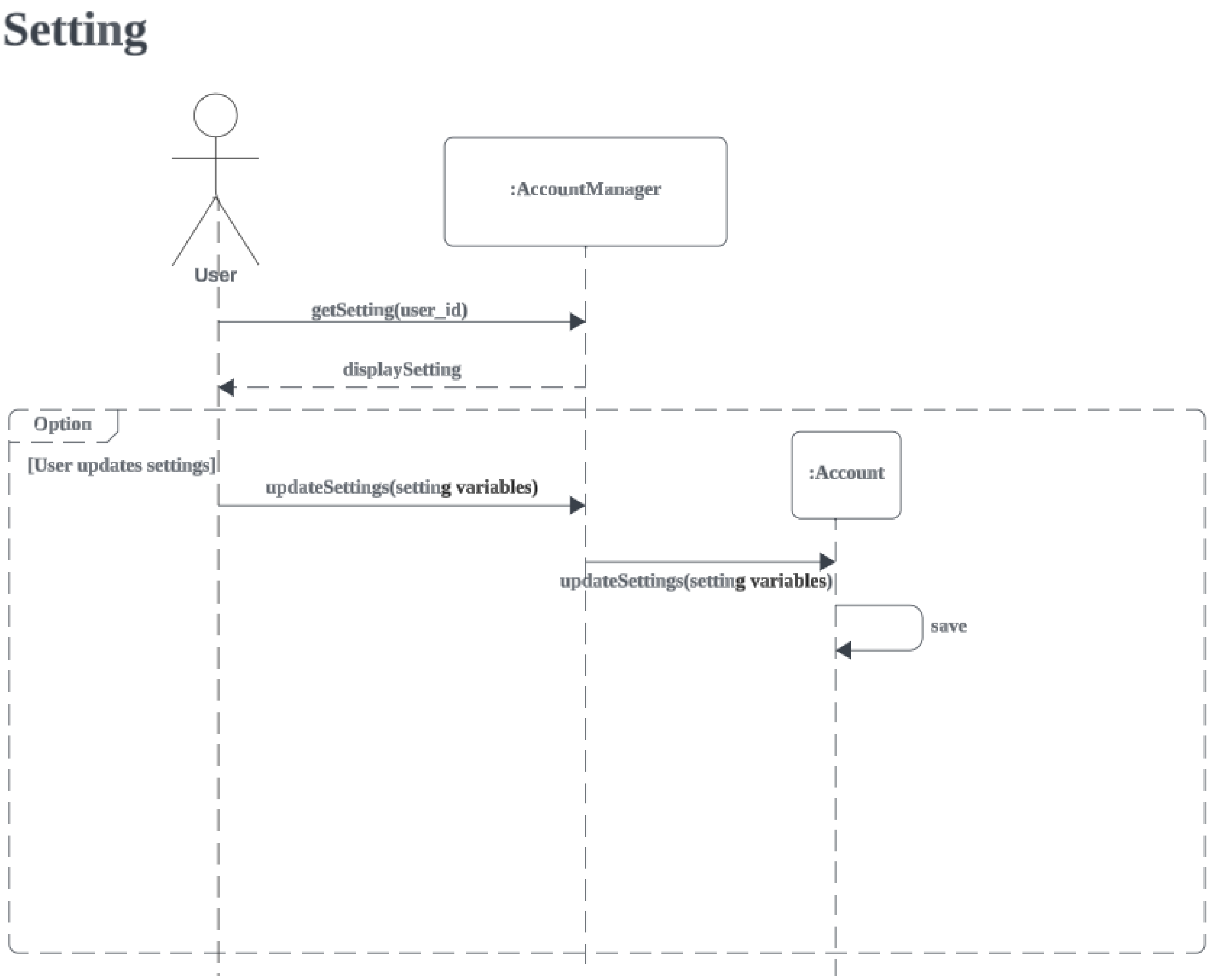


Figure 6 Sequence Diagram of setting use case

Above is the sequence diagram of the setting use case. The user will first retrieve the up-to-date settings from the AccountManager. Then AccountManager will retrieve the Account object in database and display the settings to the user. The user can choose the update the setting. Upon choosing to update the settings, AccountManager will update the Account object with the new set of settings and save it in the database.

3.2.4 Link

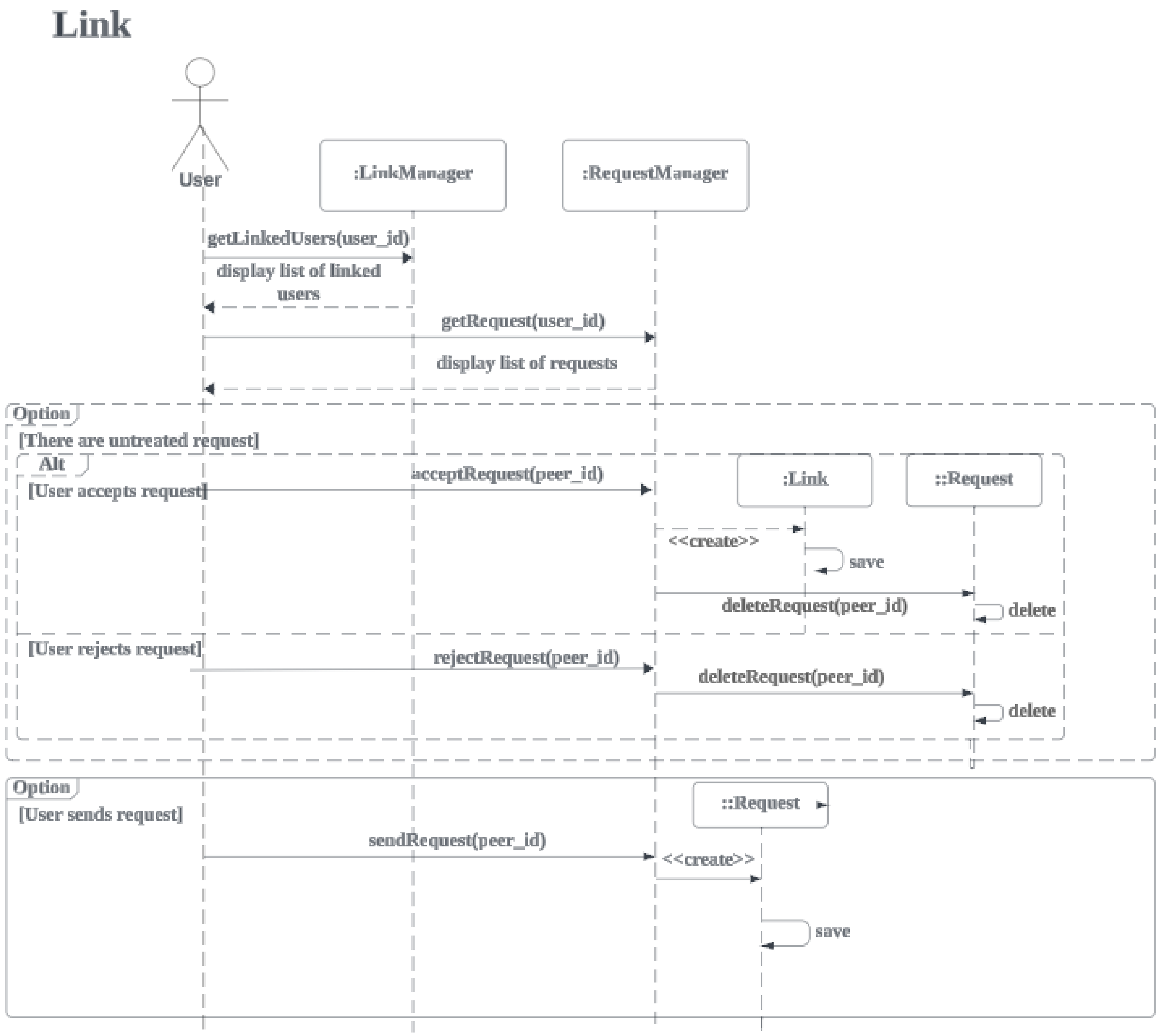


Figure 7 Sequence Diagram of Link

Above is the sequence diagram of Link use case. The user will first get the linked users list and request list from LinkManager and RequestManager. Then LinkManager and RequestManager will display the list of linked users and list of request respectively to the user.

If there are untreated link request, user can choose to accept or reject it. If user chooses to accept, RequestManager will create a new Link object that contains the (user_id,peer_id) pair and save it in the database. Afterwards, it will delete the current request record in the database as a result. If user chooses to reject the request, RequestManager will directly delete the request record in the database.

The user can also choose to send a request to a peer. A new Request object specifying (user_id,peer_id) will then be created and saved in database.

3.2.5 Chat with AI

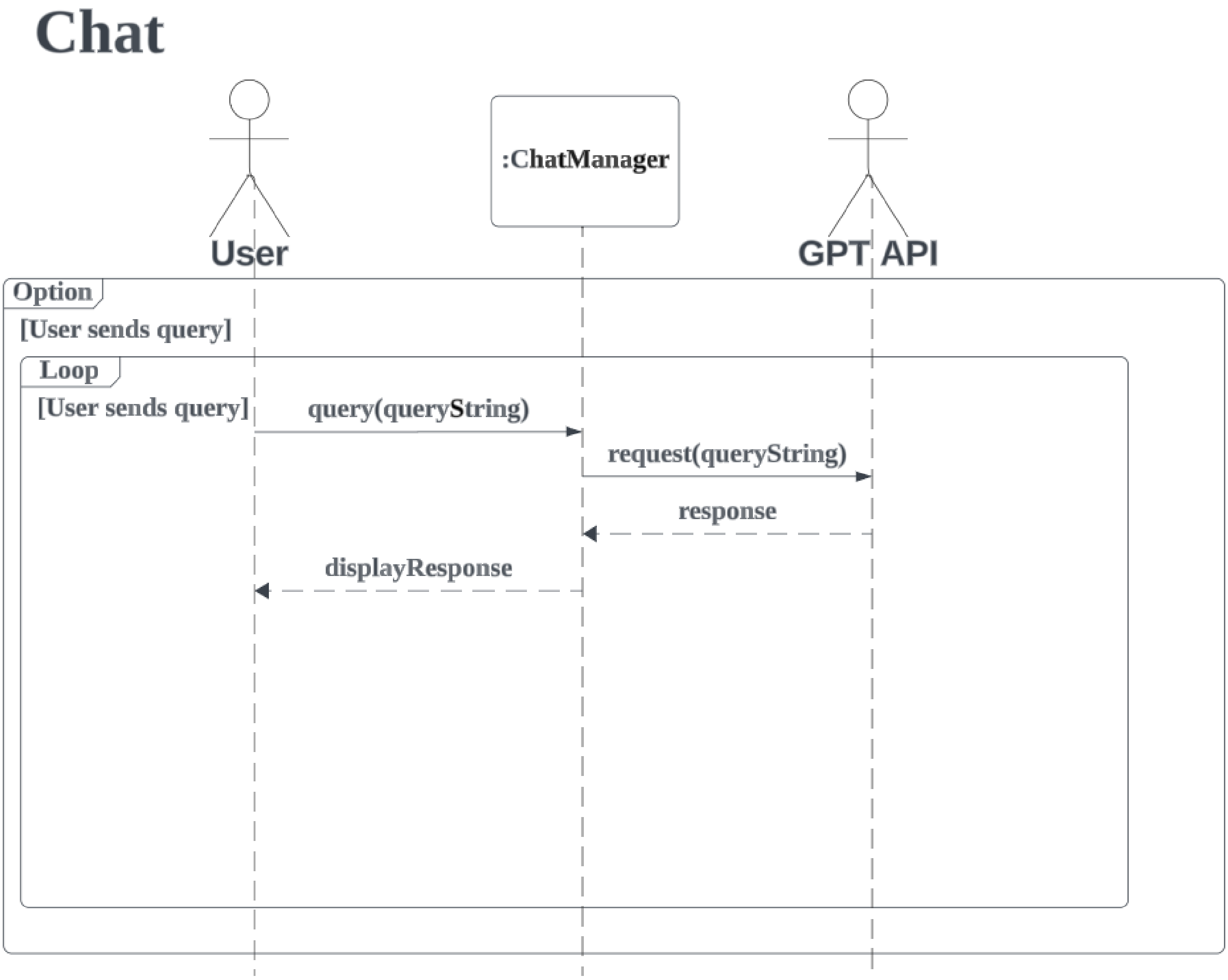


Figure 8 Sequence diagram of Chat with AI use case

Above is the sequence diagram of Chat with AI use case. The user can choose to send a query to the chatbot and every time a query is sent, ChatManager will send a request that contains the query string to the external GPT API and retrieve its response. Upon receiving the response, ChatManager will display the response to the user.

3.2.6 Filter Calls

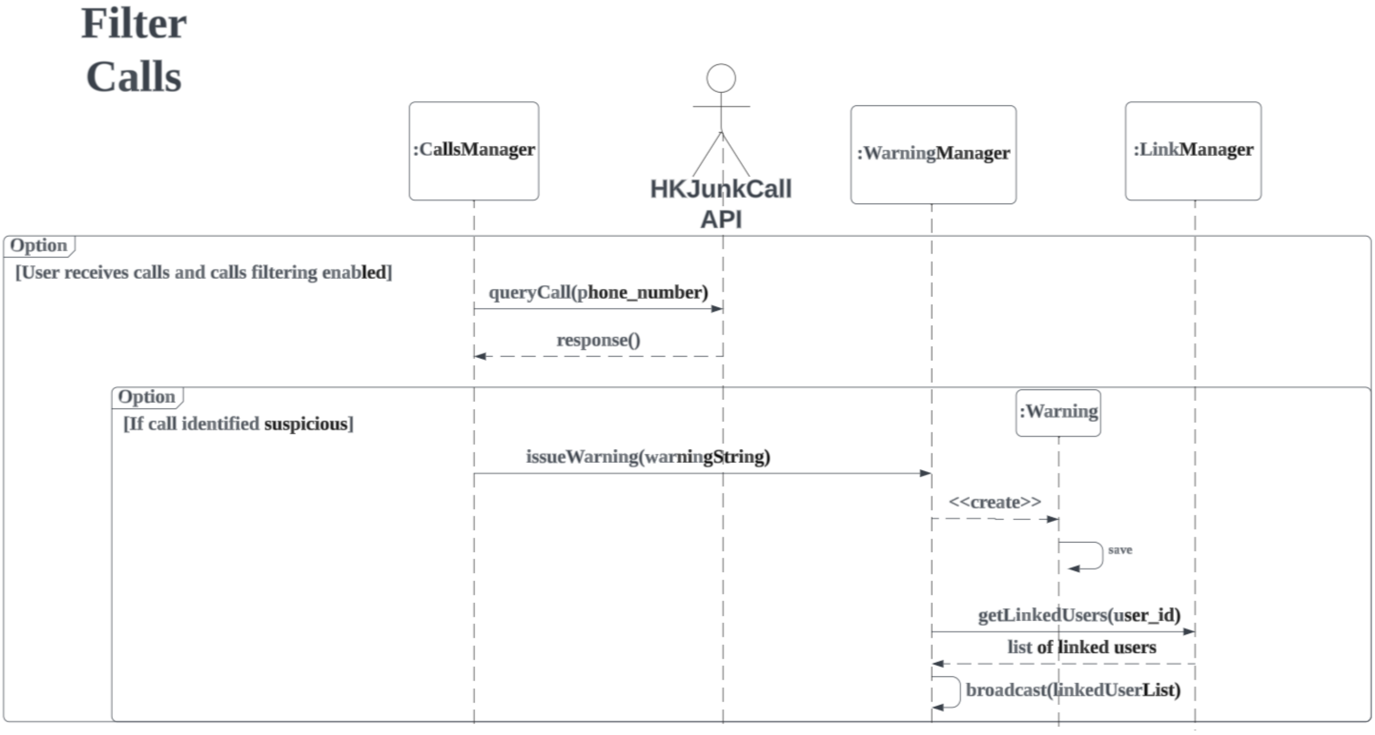


Figure 9 Sequence diagram of Filter Calls use case

Above is the sequence diagram of Filter Calls use case. If the calls filtering feature is enabled, every time there is an incoming phone call, CallManager will query the call via the HKJunkCall API and retrieve the response. If the response indicates that the phone number is suspicious, an warning string will be generated by the CallsManager and WarningManager will be called with the warning string. WarningManager will then first create a Warning object that contains the user_id and warning string and save it in database. After that, it will get the list of linked users from LinkManager and broadcast the warning message to all the linked users.

3.2.7 Track Status

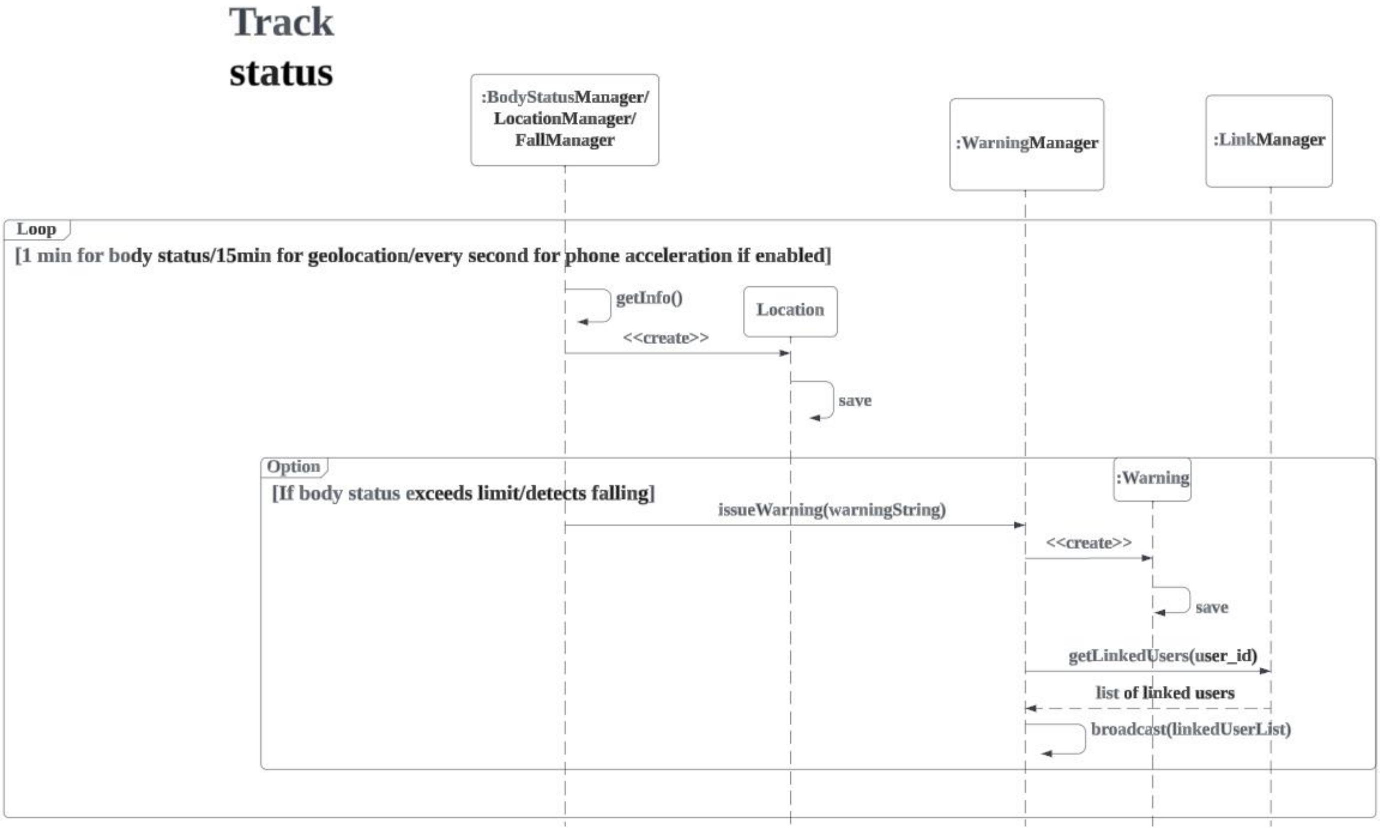

Track status
:BodyStatusManager/ LocationManager/ FallManager
:WarningManager
:LinkManager
Loop
[1 min for body status/15min for geolocation/every second for phone acceleration if enabled]
getInfo()
Location
<<create>>
save
Option
[If body status exceeds limit/detects falling]
:Warning
issueWarning(warningString)
<<create>>
save
getLinkedUsers(user_id)
list of linked users
broadcast(linkedUserList)

Figure 10 Sequence diagram of Track Status use case

Above is the sequence diagram of Track Status use case. If the corresponding feature is enabled, BodyStatusManager will get the body status (blood pressure) for every minute, LocationManager will get the geolocation and store the latest location in database for every 15 minute, while FallManager will get the phone acceleration for every second. If the body status exceeds the predefined limit or a falling is detected, BodyStatusManager and FallManager will generate the corresponding warning string and call WarningManager. The detail of issuing a warning is the same as described before.

3.2.8 Check Warning

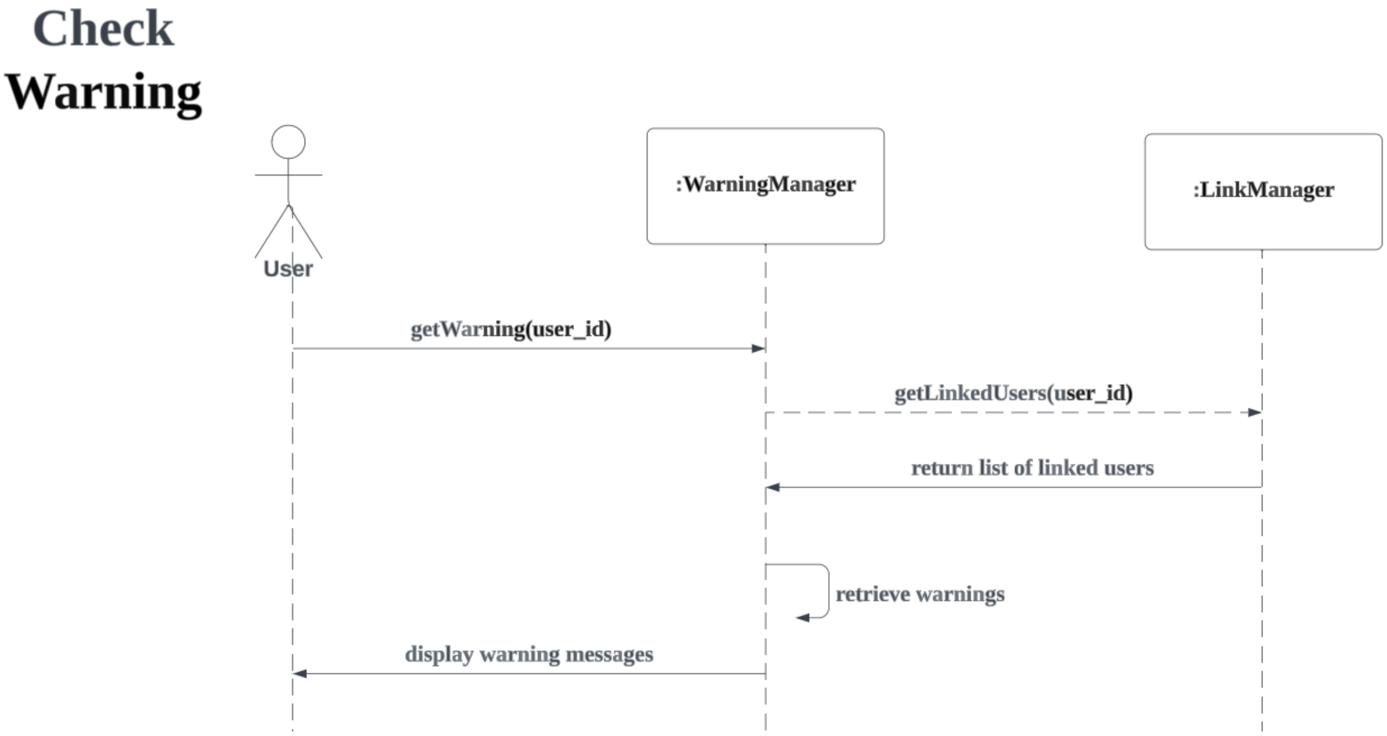


Figure 11 Sequence diagram of Check Warning use case

Above is the sequence diagram of the Check Warning use case. User can get previous issued warnings from WarningManager. WarningManager will then get the list of linked user from LinkManager and retrieve the associated warnings in the database, finally display it to the user.

3.2.9 Get Location

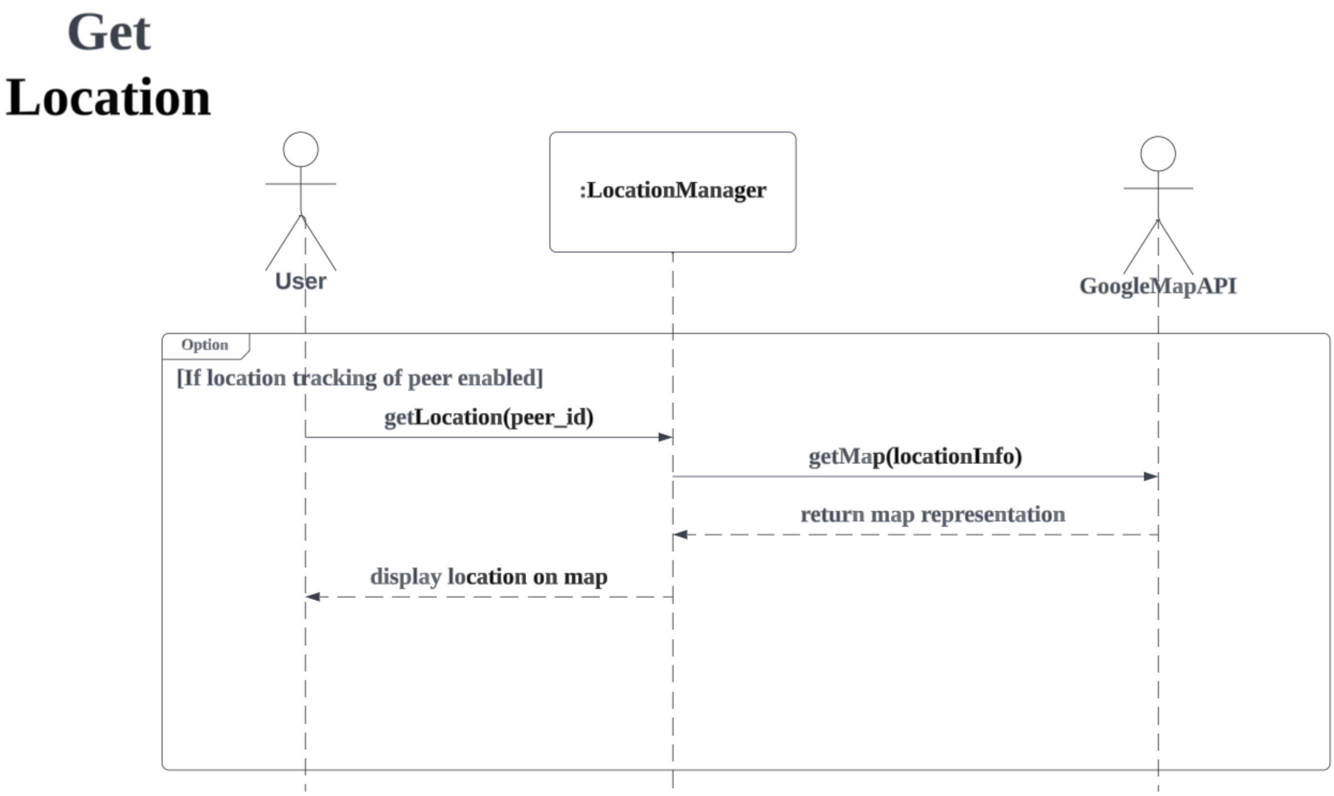


Figure 12 Sequence diagram of Get Location use case

Above is the sequence diagram of Get Location use case. If the location tracking feature of the peer is enabled, the user can get the location of the peer from LocationManager. The LocationManager will then retrieve the location information of the peer in the database and get the map representation by calling the GoogleMapAPI. Finally the LocationManager will display the location visualized on the map to the user.

3.3 Classes involved in the application

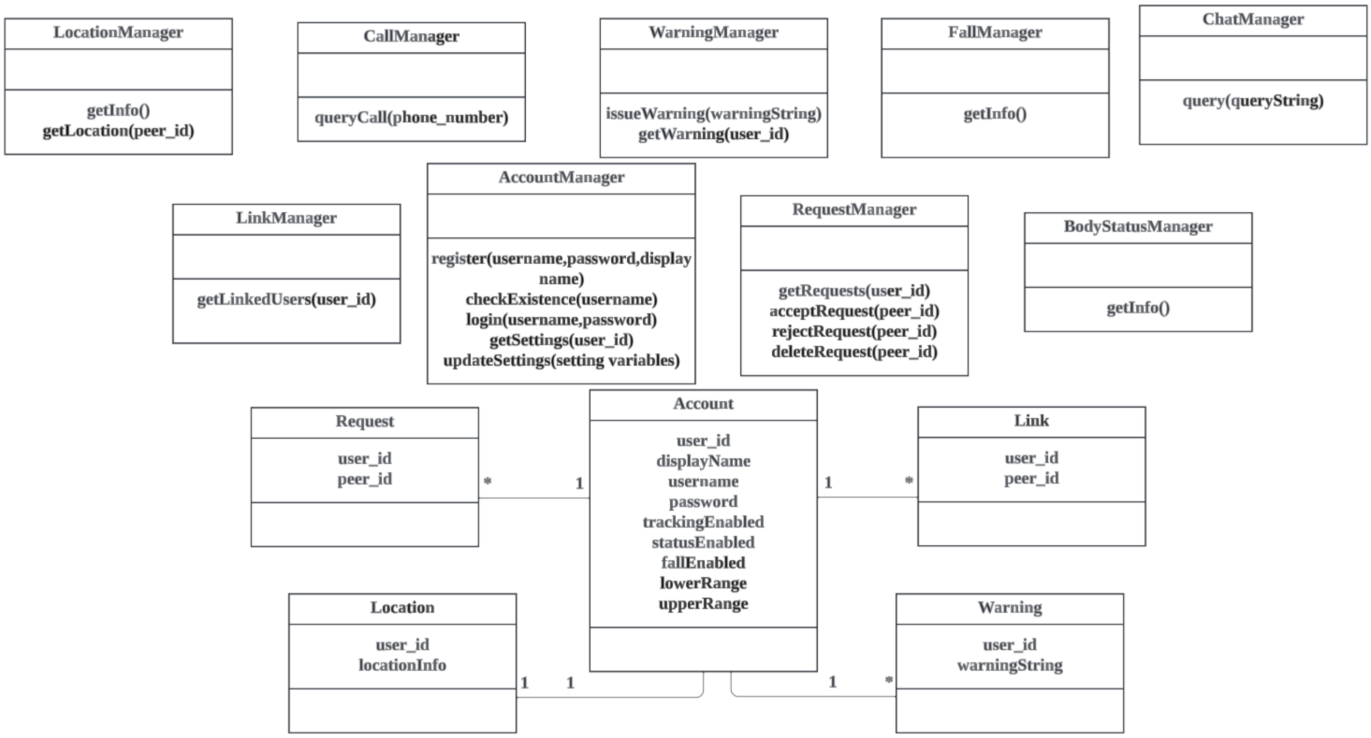


Figure 13 Class diagram of Elder Link

Above is the class diagram of the application. It is a summary of the managers and objects involved in the execution logic (shown by previous sequence diagram). Managers are classes that play a specific role and are responsible for a specific component or functionality of the system. Other classes are used to organize the required information or data related to a specific functionality. The class diagram can be used to set up classes efficiently during the development process.

3.4 Database Design

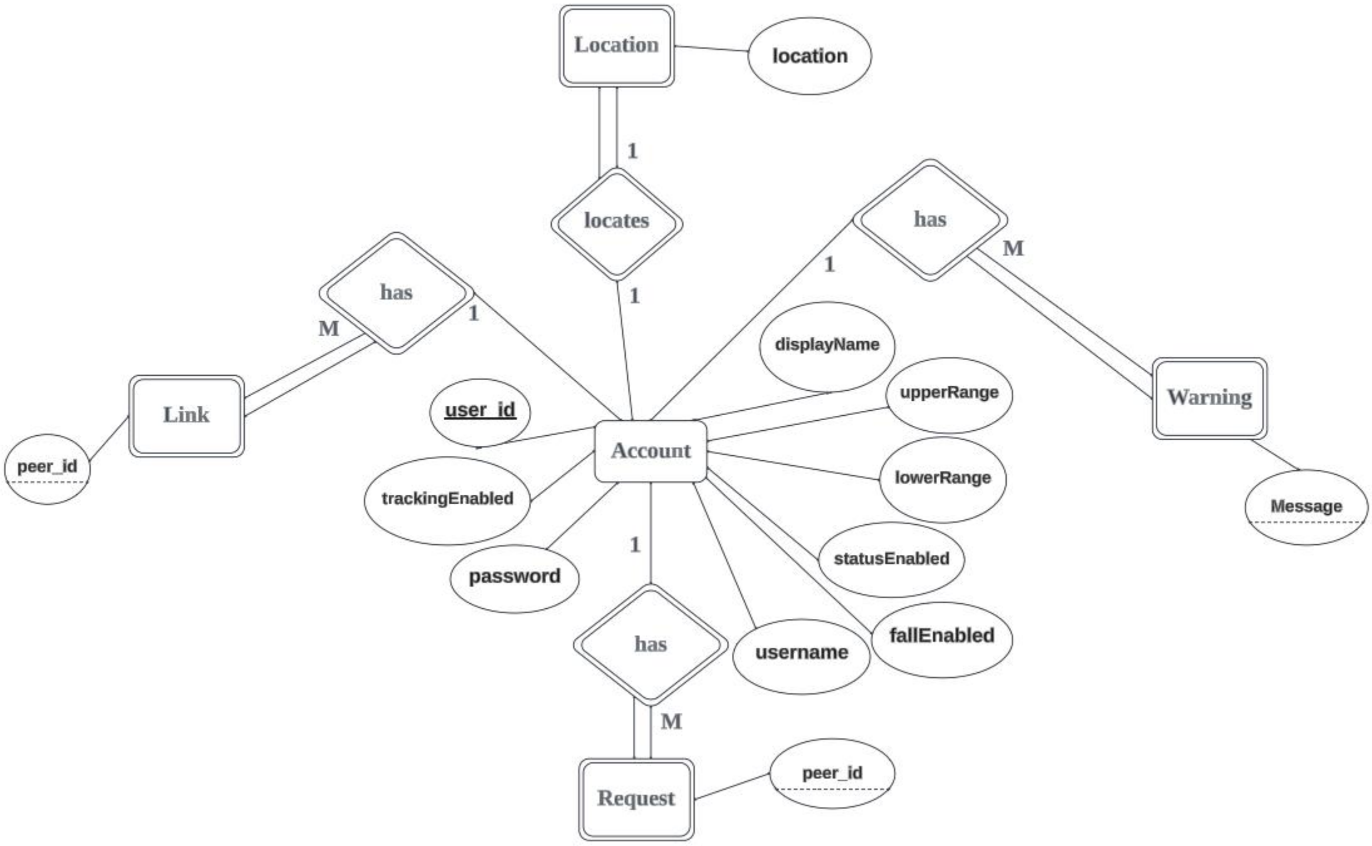


Figure 14 Entity Relationship diagram of Elder Link's database

Above is the entity relationship diagram of the application's database, which outlines the tables and their columns to be stored in the database as well as the relationship between tables.

Account entity has 9 attributes, including a primary key user_id; username, password, and display name that are strings; trackingEnabled, statusEnabled, and fallEnabled that are Booleans; as well as upperRange and lowerRange (for blood pressure) that are intergers. Account has an one to one relationship with Location, which means one account has at most one current location. It has an one to many relationship with Link, Warning and Request, which means that an account can create multiple links, issue multiple warnings and involved in multiple requests. All the participations in the relationships are partial, which means it is not necessary for an account to have any location, link, warning and request.

Location entity, which is a weak entity, has 2 attributes, including user_id (foreign key from Account) and location that is represented in a string. Location has a one to one relationship with Account, which means that each location can only associate with one account. The participation is total, which means that when every location must be linked to an account.

Link entity, which is a weak entity, has 2 attributes, including user_id (foreign key from Account) and peer_id that is an integer. Link has a many to one relationship with Account, which means that each link can only involve at most two accounts but not more. The participation is total, which means that when every link must be linked to two accounts.

Warning entity, which is a weak entity, has 2 attributes, including user_id (foreign key from Account) and warning message that is a string. Warning has a many to one relationship with Account, which means that each warning can only associate with one account. The participation is total, which means that when every warning must be linked to one account.

Request entity, which is a weak entity, has 2 attributes, including user_id (foreign key from Account) and peer id that is an integer. Request has a many to one relationship with Account, which means that each request can only associate with two accounts but not more. The participation is total, which means that when every request must be linked to two accounts.

3.5 User Interface Design

This section will focus on the Figma user interface design of different pages in the application.

3.5.1 Login and Register Pages

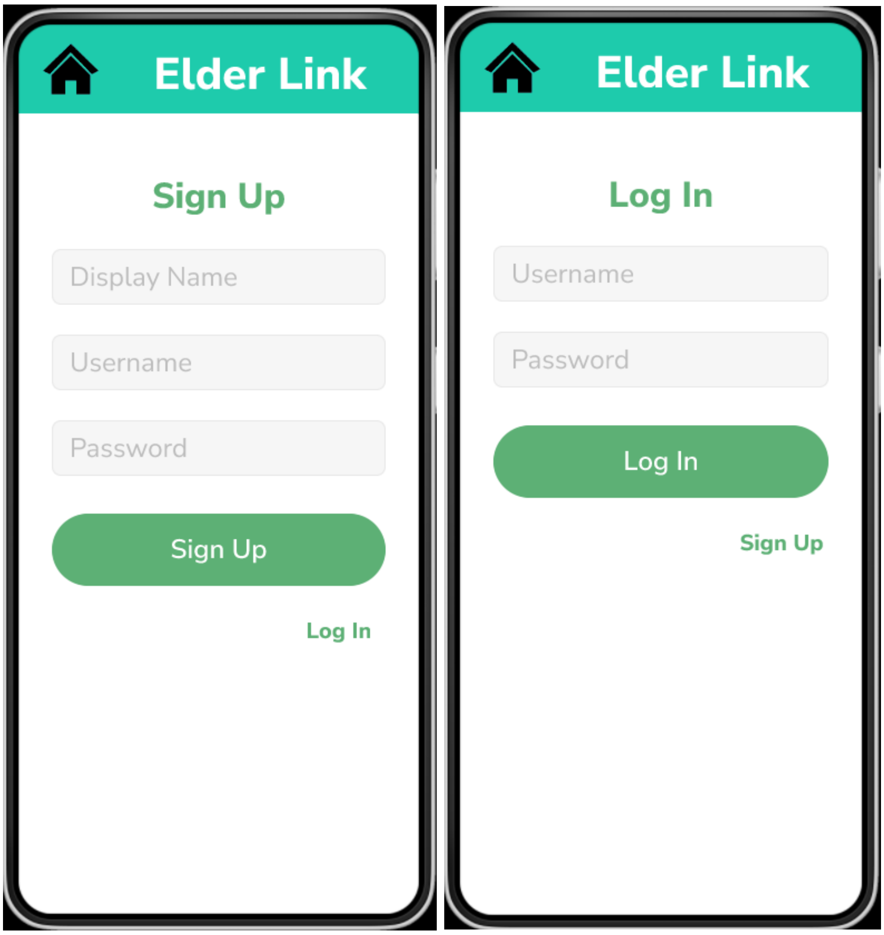


Figure 15 & 16 Register Page and Login Page

Above are the register and login pages of the application. In the Register page, user can fill in the display name, username, and password, then click the sign up button to submit their registrations. In login page, user can fill in their username and password, then click the login button to authenticate and log in.

3.5.2 Main Page

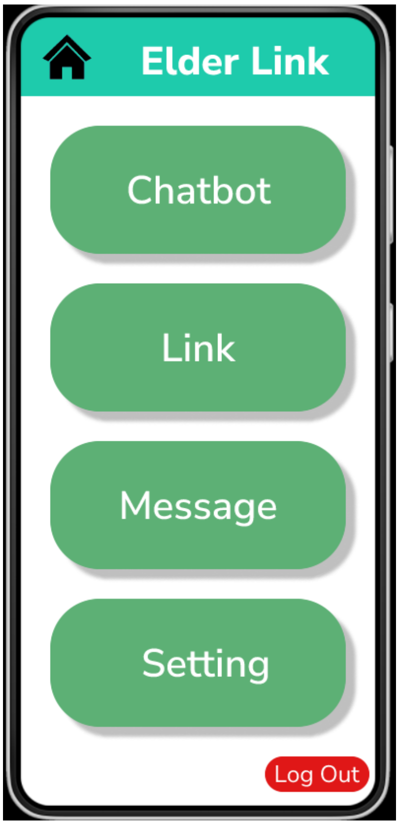


Figure 17 Main Page

Above is the main page of the application. It consists of navigation to chatbot, link, message, and setting pages, as well as a log out button which will navigate back to the login page.

3.5.3 Chatbot Page

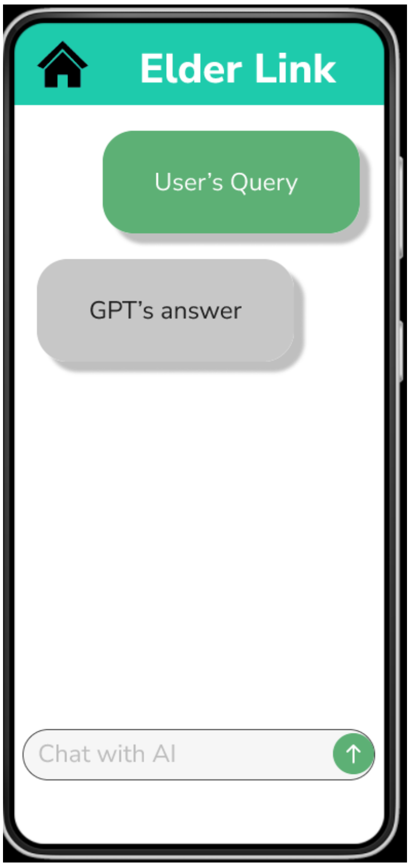


Figure 18 Chatbot Page

Above is the chatbot page of the application. User can type in their queries in the textbox and click on the button to submit their queries. The queries and responses will be displayed as if messages in messenger apps.

3.5.4 Link Pages

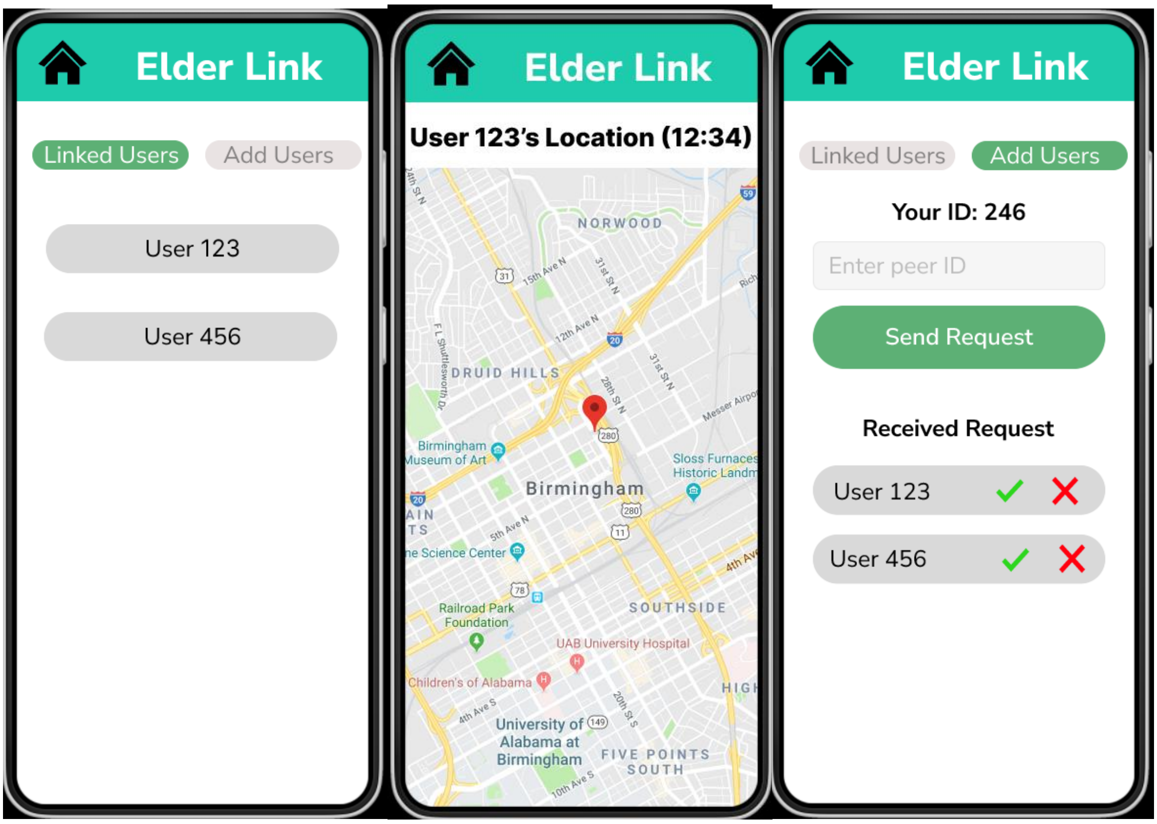
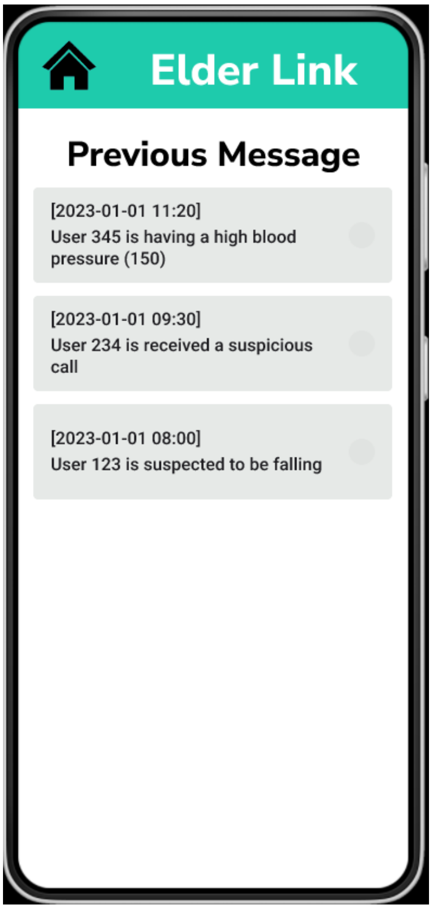


Figure 19 & 20 & 21 Link Pages

Above are pages in the Link component of the application. In the first page, the list of linked users will be displayed. If any of the linked users is clicked, it will enter the second page, which displays the location of the selected linked user. If the Add Users button in the first page is clicked, the third page will be shown. The third page displays the user id of the user, a text box and a Send Request button for user to enter their peer's id and send a request. There is also an area that displays all received requests and allows users to accept and reject the requests by clicking the tick or cross buttons.

3.5.5 Previous Message Page



Previous Message

[2023-01-01 11:20]
User 345 is having a high blood pressure (150)

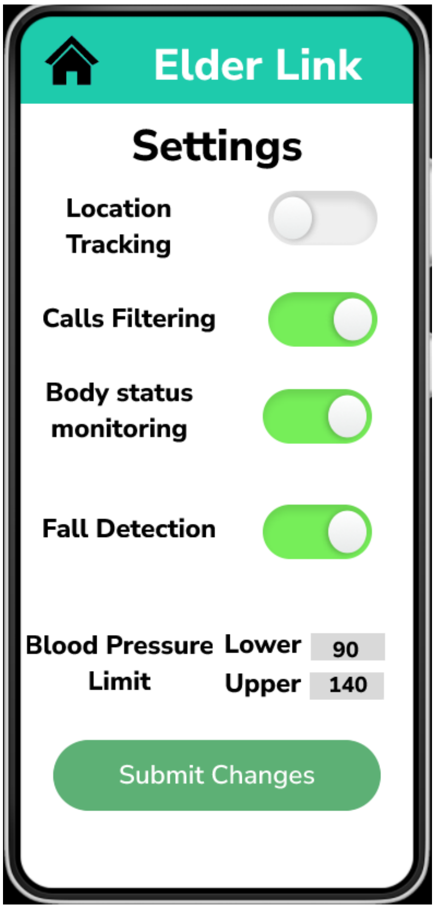
[2023-01-01 09:30]
User 234 is received a suspicious call

[2023-01-01 08:00]
User 123 is suspected to be falling

Figure 22 Previous Message Page

Above is the previous message page which displays all the previous warning messages issued by linked users.

3.5.6 Setting Page



Location Tracking

Calls Filtering

Body status monitoring

Fall Detection

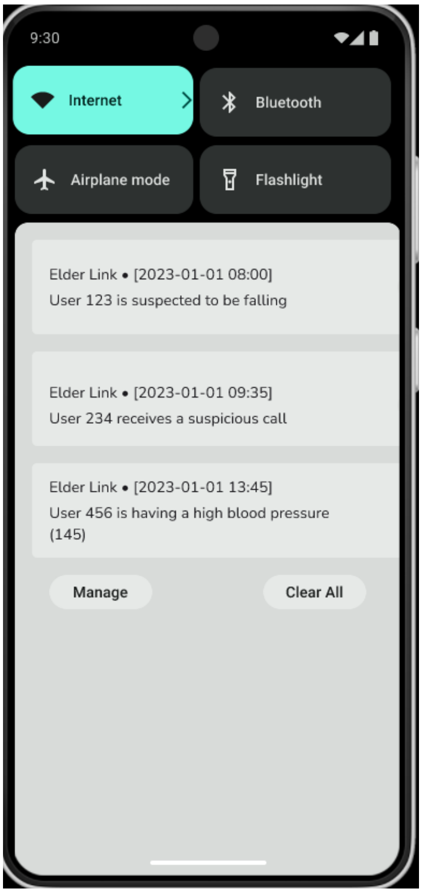
Blood Pressure Limit Lower 90

Upper 140

Figure 23 Setting Page

Above is the setting page of the application. It consists of toggle buttons to enable or disable the features, text boxes to enter the lower and upper limit of blood pressure, as well as a summit button to save the changed to database.

3.5.7 Notification



Elder Link • [2023-01-01 08:00]
User 123 is suspected to be falling

Elder Link • [2023-01-01 09:35]
User 234 receives a suspicious call

Elder Link • [2023-01-01 13:45]
User 456 is having a high blood pressure (145)

Figure 24 Notification

Above is the sample notification when a linked user triggers any of the warning.

4 Expected future results

This section will outline the expected future results and the plan in the second semester. Section 4.1 – 4.4 will describe the expected implementation, testing, and documentation results, as well as a work plan in the second semester.

4.1 Implementation

The source code of important sections in the application will be discussed, including how it works and what packages are involved.

4.2 Testing

Testing process and test cases will be described and discussed. Analysis of the performance of the application will also be provided.

4.3 Documentation

Related set of documentations, including final report, poster and video for project exhibition will be finished.

4.4 Work Plan

Testing and implementation will be interleaved, and all the essential features are expected to be finished within February. Starting from March, time will be spent on finishing the documentation as well as integrating potential new features into the application or refining the application.

5. Conclusion

Telephone deception, missing elderly persons, accidents, and the generation gap are notable problems that elderly individuals may encounter in today's society. To address these issues, this paper aims to harness the power of technology and develop a customized mobile application for the elderly.

The application will have several key functions, including location tracking, smartwatch connectivity, calls filtering, fall detection, and integration with GPT (Generative Pre-trained Transformer) technology. To achieve these functions, preliminary designs such as user interface, system, and database design will be created, followed by implementation and testing.

At this stage of the project, the preliminary design, including the use case design, execution logic, classes, database, and user interface are completed, which all together well define a system and enable efficient implementation during the development process.

As of now, the project progress remains on schedule. The major task in the second semester is to complete the implementation and testing of the application, as well as finish the related documentation like final report, poster and video.

6. Reference

[1] Anti-Deception Coordination Centre, *Scam Statistics*

https://www.adcc.gov.hk/en-hk/statistic.html (accessed: Oct 1, 2023)

[2] Security Bureau, Hong Kong Police Force, *Legislative Council Panel on Security Initiatives for Preventing and Combatting Deception Cases,* May 2021

https://www.legco.gov.hk/yr20-21/english/panels/se/papers/se20210601cb2-1110-3-e.pdf (accessed: Oct 1, 2023)

[3] Sarah Perez, *Google's Family Link updates reflect the pandemic's impact on how parents view screen time,* 16 March, 2021

https://techcrunch.com/2021/03/16/googles-family-link-updates-reflect-the-pandemics-impact-on-how-parents-view-screen-time/ (accessed: Nov 24, 2023)

[4] Whoscall official website

https://whoscall.com/en/about (accessed: Nov 24, 2023)

[5] Google Family Link official website

https://families.google/intl/en/familylink/ (accessed: Nov 24, 2023)

[6] Apple Official Website

https://www.apple.com/hk/en/ios/health/ (accessed: Nov 24, 2023)

[7] React Native official website

https://reactnative.dev/ (accessed: Oct 1, 2023)

[8] Visual Studio Code official website

https://code.visualstudio.com/ (accessed: Oct 1, 2023)

[9] Firebase official website

https://firebase.google.com/ (accessed: Oct 1, 2023)

[10] Lucidchart official website

https://www.lucidchart.com/ (accessed: Oct 1, 2023)

[11] Lucidchart , *Introducing Types of UML Diagrams | Lucidchart Blog*
https://www.lucidchart.com/blog/types-of-UML-diagrams (accessed: Oct 1, 2023)

[12] Lucidchart , *UML Class Diagram Tutorial*

https://www.lucidchart.com/pages/uml-class-diagram (accessed: Oct 1, 2023)

[13] Figma official website

https://www.figma.com/ (accessed: Oct 1, 2023)

[14] Kinsta, *What Is GitHub? Kinsta A Beginner's Introduction to GitHub*, Dec 13, 2022

https://kinsta.com/knowledgebase/what-is-github/ (accessed: Oct 1, 2023)

[15] GeeksforGeeks, *Unit Testing | Software Testing,* 06 Feb, 2023

https://www.geeksforgeeks.org/unit-testing-software-testing/ (accessed: Oct 1, 2023)

[16] GeeksforGeeks, *Software Engineering | Integration Testing,* 06 Apr, 2023

https://www.geeksforgeeks.org/software-engineering-integration-testing/ (accessed: Oct 1, 2023)

[17] GeeksforGeeks, *Software Testing – UAT (User Acceptance Testing),* 29 Mar, 2023

https://www.geeksforgeeks.org/software-testing-uat-user-acceptance-testing/ (accessed: Oct 1, 2023)