Software Requirements Specification

for

Instant Messaging System

Prepared by Project Team 4

(Christopher Sellers, Jerry Zhu, Aaron Lam)

Instructor: Sara Farag

Course: CS 410

Date: 04-30-2019

Table of Contents

Revision History

Primary Authors	Date	Reason For Changes	Version
Jerry Zhu	4/22/19	Added Section 1.x	0.1
Aaron Lam, Christopher Sellers, Jerry Zhu	4/23/19	Added Section 2.2	0.2
Christopher Sellers	4/23/19	Added Section 3.1 & 3.2	0.3
Aaron Lam	4/24/19	Added Section 2.x	0.4
Jerry Zhu	4/24/19	Added Section 3.3 & Revised Section 3.2	0.5
Aaron Lam, Christopher Sellers, Jerry Zhu	4/26/19	Added Section 4.x	0.6
Jerry Zhu	4/28/19	Added Section 5	0.7
Christopher Sellers, Jerry Zhu	4/28/19	Added Section 6	0.8
Aaron Lam, Christopher Sellers, Jerry Zhu	4/29/19	Revise all sections & Added section 1.6	0.9
Aaron Lam, Christopher Sellers, Jerry Zhu	4/30/19	First submission for SRS document	1.0
Jerry Zhu	5/25/19	Added use case story in Section 3.2, Added extension point in Section 3.3, Revise Section 5.2 & 5.3	1.1
Chris Sellers	5/25/19	Clarified points in Section 3.1 Added information for 3.1 Added points for Section 4.3	1.2
Aaron Lam	5/27/19	Added figure title for Section 2	1.3

1. Introduction

Instant messaging through mobile platform is an important aspect in the modern world. To be able to send and receive messages from afar allows people to communicate efficiently. Although instant messaging technology has made major advances in the past decade, IM applications in the current market still struggle in fundamental user aspects: Usability and extensibility. To compete in the market, there will be extra emphasis on these aspects when designing the base version of the IM mobile application.

1.1 Document Purpose

This document illustrates the design requirements of version 1.0 of an instant messaging application. Details regarding the application's core functionalities, interfaces, use case scenarios, and characteristics are covered in the future sections. The instant messaging application is intended to be a stand-alone software that allows for future extensions and plugins.

1.2 Product Scope

The IM application that will be developed is designed to improve usability; the application will be intuitive and easy to use. In addition, this application will serve as a foundation for future projects. New features will be easy to incorporate into the base version. High usability and extensibility are the core values that will distinguish the proposed IM application from the ones currently in the market. To satisfy the basic needs of the clients, existing features provided by other IM applications will included: instant messaging system, account settings management, etc.

1.3 Intended Audience and Document Overview

This document provides an overview of the software requirements and functionalities to inform the project manager, Sara Farag, and the product owner. Section 2 (Overall Description) covers the core functionalities, intended use of the application, possible constraints, and dependencies. Section 3 (Specific Requirements) describes the functional and behavior requirements through use case templates and diagrams, respectively. Section 4 (Other Non-functional Requirements) discusses the essential characteristics that the application must possess in order to have marketable value: security, maintainability, availability, and usability.

It is highly recommended for all readers to view section 2 first to acquire a general understanding of the IM application. Section 3 offers insight on the application development and detailed analysis of each use case, which the project manager may find important. Since the product owner is responsible for representing the interest of the clients, it is recommended for him/her to read section 4 to understand how the application will improve the client's quality of life.

1.4 Definitions, Acronyms, and Abbreviations

- IM = Instant messaging
- UI = User interface
- UX = User Experience
- SDK = Software Development Kit
- API = Application Programming Interface
- AI = Artificial Intelligence
- CRUD = Create, Read, Update, Delete

1.5 Document Conventions

Throughout the document, there may be unfamiliar acronyms. To understand them, refer to section 1.4 (Definitions, Acronyms, and Abbreviations).

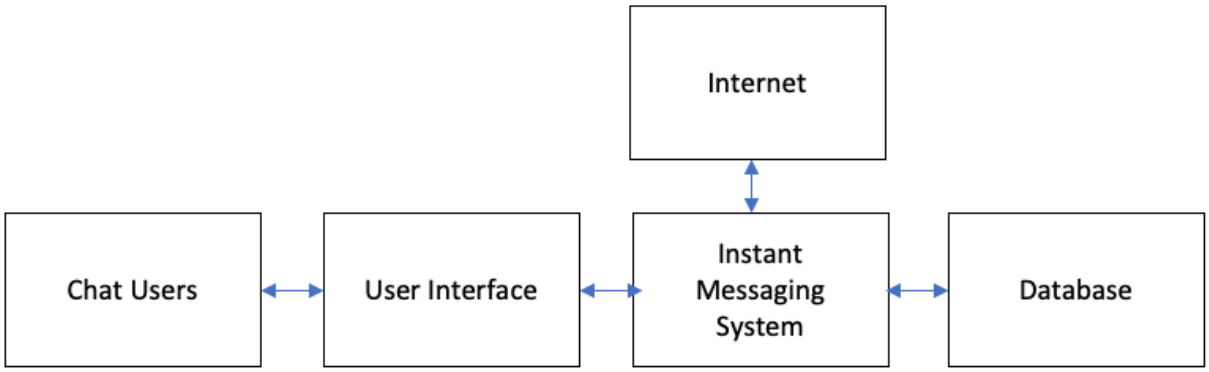
1.6 References and Acknowledgments

Android Developers. (2019). *Platform Architecture | Android Developers*. [online] Available at: https://developer.android.com/guide/platform [Accessed 30 Apr. 2019].

Socialcompare.com. (2019). *Android versions comparison | Comparison tables - SocialCompare*. [online] Available at: http://socialcompare.com/es/comparison/android-versions-comparison [Accessed 30 Apr. 2019].

2. Overall Description

2.1 Product Perspective



Product Perspective Diagram

The instant messaging application is a new, self-contained product. Chat users can send action to instant messaging system by interacting with the UI. After receiving the user actions from UI, the application will do the tasks (e.g: update database, retrieve messages from database, etc.) and possibly render the display on the UI. While chat users are using the instant messaging application, we have to make sure that the system is connected with stable internet.

2.2 Product Functionality

- Account functions
  - Register new account
  - Manage account settings
  - Login/Logout to account

- Chat functions
  - Start/Leave chat (User limit)
  - Join chat
  - View chat
  - Go back to chats interface

- Messaging functions
  - Send messages
  - Browse message history
  - Receive notifications

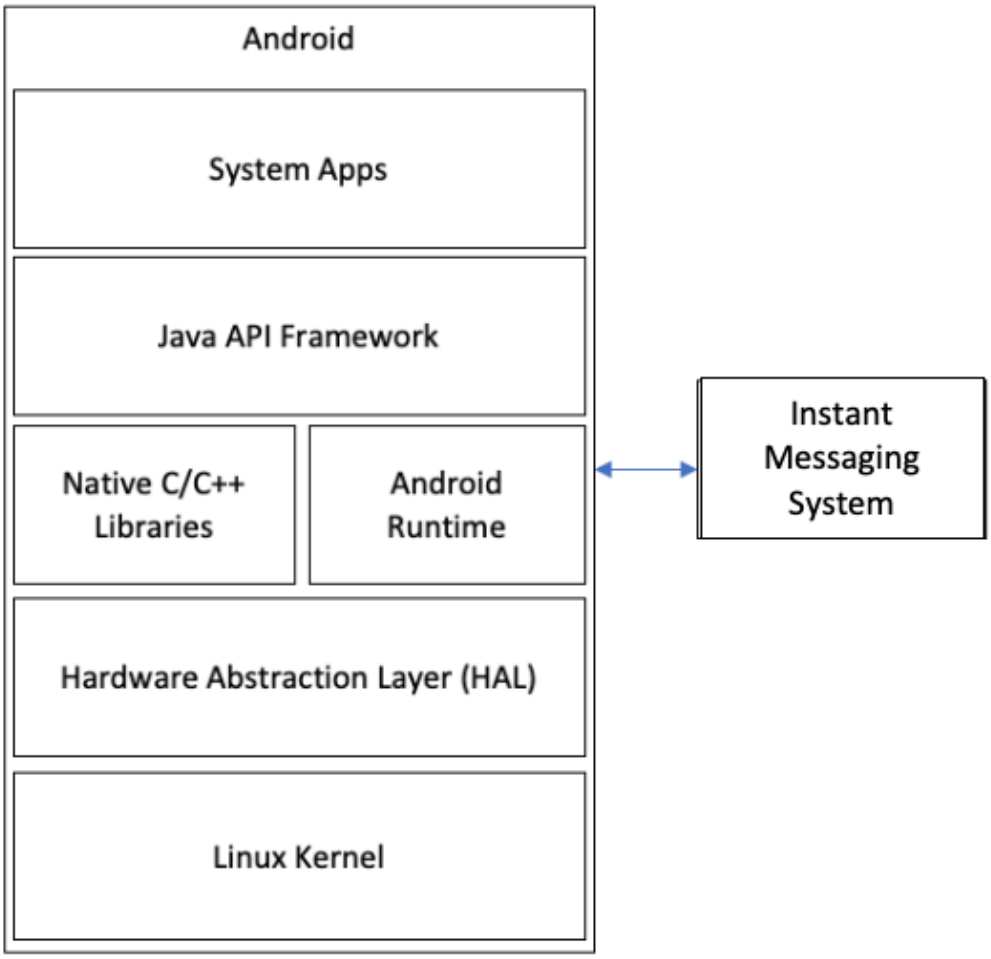
2.3 Users and Characteristics

Users	**Chat User**	**Tech Support**	**Developer**	**Marketing Person**
Use	While using the app	On support Calls	Develop and implement features on system	Receive customer feedback
Functions	All functions from section 2.2	Can retrieve information from customer service database	Can do CRUD on database; have testing environment available	Can collect customer feedback when alpha / beta version is launched
Expertise	Good behavior chat users	Familiar with customer service database	Familiar with database, UI, and the whole instant messaging system	Familiar with the product and its features
Security / Privilege	Only able to access UI	Access to customer service database	All uncensored data and codebase	Only able to access UI

Order of user importance in descending order:

- Chat user
- Developer
- Tech Support
- Marketing Person

2.4 Operating Environment



Android Operating System Diagram

We will be using Android as our operating system. Android is an open source, Linux-based software stack created for mobile devices. In this project, our instant messaging Android application will only communicate with the Android operating system through the Java API framework and Android Runtime. The instant messaging application development should not require digging into hardware parts such as HAL and Linux Kernel.

This project will use API 15 (Android 4.0.3) as the minimum SDK version. Even though API 15 is not the latest SDK version, our instant messaging application will run on approximately 100% of Android devices by targeting API 15 and later.

2.5 Design and Implementation Constraints

1. Does not apply to smart watch and TV
Smart watch and TV will require Android version 5.0 or higher (API 21 or above).

2. Neural networks API for Artificial Intelligence is not available
Neural networks API can make our application be able to extend AI features. However, this API only supports on Android version 8.1 or above (API 27 or above). Currently, only 1.1 % of Android devices run on version 8.1 or above.

3. Does not support material design style
Material design style requires Android version 5.0 or higher (API 21 or above). Some of the animations might not be available in our instant messaging app UI.

2.6 User Documentation

Document	Description	Format
Chat User Guide	Instructions for using chat app	Webpage
Developers Documentation	Brief description of introducing project structure, system architecture, etc.	Word / PDF
Tech Support Manual	Instructions for troubleshooting chat users' issues and concerns	Word / PDF

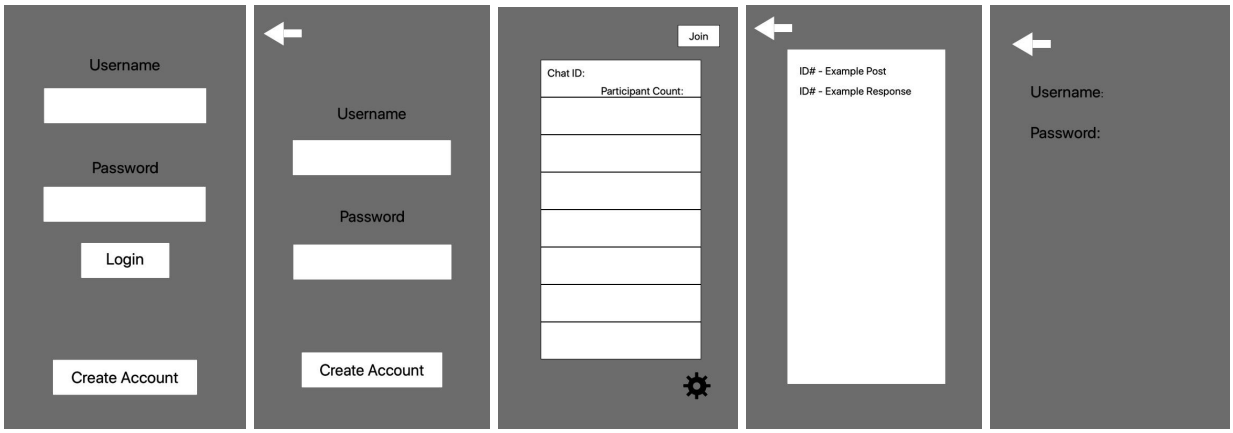
2.7 Assumptions and Dependencies

- All chat users' Android device versions are 4.0.3 or above
- Requires Android API 15 or above
- User phones have stable wifi connection
- UI and some of the functionalities may change during the development
- Future versions of the application will also operate on the devices mentioned above

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces



Login Page Create Account Main Page Chat Panel Account Settings

3.1.2 Hardware Interfaces

No special libraries will be necessary for the hardware to communicate with the application. All hardware interfaces are part of the user's device.

3.1.3 Software Interfaces

Persistent data will be stored in SQLite because it is the default built-in storage in Android OS. Android version 4 or higher is required. The application will use post and get to send and receive messages from the database. Login credentials will also be verified through post transactions.

3.1.4 Communications Interfaces

FTPS will be used as the standard communication protocol. This implies the connection between each device and the server will be encrypted. An encrypted connection is key for this project to ensure all data is protected and handled properly.

3.2 Functional Requirements

Create account

- When the user does not have an account they will click the “Create Account” button on the Login Interface.
    - Establish a connection with the database.
    - Post the username and password input to the database.
        - If the username is available create the account.

Manage account settings

- After the user has signed in, they will be able to click the gear icon to manage their account.
    - The user will be able to modify their username and password.
    - A user can logout from this menu.

Login to account

- Once the application is launched the Login Interface will be displayed.
    - A user inputs their login credentials
        - If the credentials are correct the next interface will appear
        - If the credentials don’t match an error will be displayed and the application will remain on the current page.

Chat functions

- A user has the ability to start a chat by clicking on a new chat slot and specifying the number of slots available within the chat.
    - The new chat channel will have an ID generated and a password assigned by the user.
- Each chat can also be removed by swiping the specific chat to the left and selecting delete.
- Chats can be joined by clicking the Join button on the Chat Panel Interface.
    - The correct chat ID and password are necessary.
- All chat channels are visible on the Chat Panel Interface.
- When in a channel the user can return to the main panel through the return arrow.

Messaging functions

- A message can be sent by typing in the provided text box and clicking the Send button.
- The Chat Interface displays all previous messages sent in the current channel.
- A notification is posted to the chat channel on the main interface to display any new messages.

User Story 1: As a user, I need to be able to create a new account, login, logout, and configure account settings.

Account Registration

Use Case Number	1
Summary	The user needs to be able to create an account.
Actor	Application user
Trigger	“Create new account” button
Primary Scenario	User clicks the “Create new account” button on the login interface. User is directed to the account registration interface.
Alternative Scenario	N/A
Exceptional Scenario	If phone number or email has been already taken, clear the entries and notify the user.
Pre-Condition	N/A
Post-Condition	Notifies user that new account has been created. User is directed to the login interface.
Assumptions	Phone number will be used as the username. Every account has a unique phone number. Passwords do not have to be unique.

Account Login

Use Case Number	2
Summary	The user needs to be able to login.
Actor	Application user
Trigger	Login button
Primary Scenario	User clicks the login button.
Alternative Scenario	N/A
Exceptional Scenario	If username and password are invalid, application clears entries and notifies user of invalid login entry.
Pre-Condition	N/A
Post-Condition	If the username and password are valid, user is directed to chats interface.
Assumptions	N/A

Account Logout

Use Case Number	3
Summary	The user needs to be able to logout.
Actor	Application user
Trigger	Logout button
Primary Scenario	User clicks the logout button on the chats interface.
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in.
Post-Condition	User is directed to the login interface.
Assumptions	N/A

Account settings

Use Case Number	4
Summary	The user needs to be able to change account settings.
Actor	Application user
Trigger	“Account settings” button
Primary Scenario	User clicks the “Account settings” button on the chats interface.
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in.
Post-Condition	User is directed to accounts settings interface.
Assumptions	N/A

User Story 2: As a user, I want to be able to start and view a specific chat. I want to be able to navigate back to the chats interface. I want to be able to join and leave a chat group.

Start chat

Use Case Number	5
Summary	The user needs to be able to start a chat and specific the maximum members that can be in the chat.
Actor	Application user
Trigger	“Start chat” button
Primary Scenario	User clicks on the “Start chat” button on chats interface.
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in.
Post-Condition	A new chat panel is created in the chats interface. The user is now a member of the chat.
Assumptions	If the user starts a chat, he/she intends on being a member of the chat.

Leave chat

Use Case Number	6
Summary	The user needs to be able to leave a chat.
Actor	Application user
Trigger	“Leave chat” button
Primary Scenario	User clicks on the “Leave chat” button on the chat interface.
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in. User must be a member of the chat.
Post-Condition	User is directed to the chats interface. The user is no longer a member of the chat that he/she left.
Assumptions	If there are no members in chat, the chat is no longer needed and will get deleted.

Join chat

Use Case Number	7
Summary	The user needs to be able to join a specific chat.
Actor	Application user
Trigger	Join button
Primary Scenario	User types in the chat ID in a textbox and clicks on the join button in the chats interface.
Alternative Scenario	N/A
Exceptional Scenario	The chat ID the user typed is invalid. The user will be notified of invalid chat ID and the entry in the textbox will get cleared.
Pre-Condition	User must be logged in.
Post-Condition	User is then directed to the chat interface that he/she picked.
Assumptions	N/A

View chat

Use Case Number	8
Summary	The user needs to be able to view a specific chat
Actor	Application user
Trigger	Chat panel
Primary Scenario	User clicks on one of the chat panels that he/she wants to view.
Alternative Scenario	N/A
Exceptional Scenario	The chat, that the user wants to view, has been deleted. User is notified that the chat does not exist anymore.
Pre-Condition	User must be logged in. User must be a member of the chat.
Post-Condition	User is then directed to the chat interface that he/she picked.
Assumptions	N/A

Return to chats interface

Use Case Number	9
Summary	The user is in a specific chat and wants to return to the chats interface.
Actor	Application user
Trigger	Back arrow button
Primary Scenario	User clicks on the back arrow button on the chat interface
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in. User must be a member of the chat
Post-Condition	User is directed to the chat interface
Assumptions	N/A

User Story 3: As a user, I want to be able to send and receive messages. I want to be able to browse through past messages and receive a new message notification.

Send message

Use Case Number	10
Summary	The user needs to be able to send messages in the chat
Actor	Application user
Trigger	Send button
Primary Scenario	User types message in textbox and presses the send button.
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in. User must be a member of the chat
Post-Condition	The message gets posted on the chat for other users to see. Other users get a new message notification on their chats interface.
Assumptions	Other users want to be notified of a new message that they received.

Receive new message notification

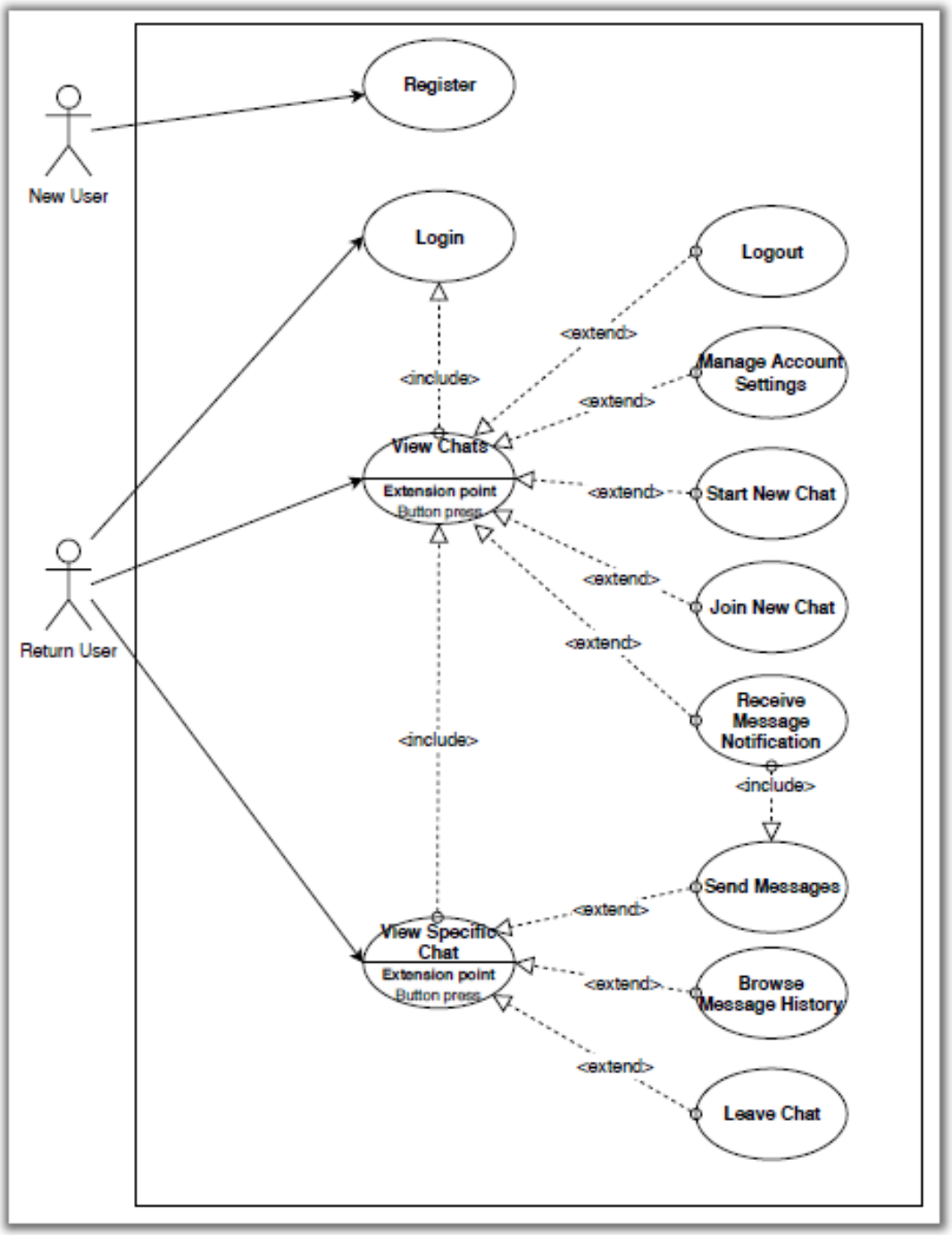
Use Case Number	11
Summary	The user needs to be notified when he/she receives a new message.
Actor	Application user
Trigger	Send button
Primary Scenario	Another user sends a message. A notification pops up next to the corresponding chat panel in the chats interface.
Alternative Scenario	N/A
Exceptional Scenario	The user is already viewing the chat that received the new message. There will no notification in this case.
Pre-Condition	User must be logged in. User must be a member of the chat. Another user must send a new message.
Post-Condition	If the user views the chat that received the new message, the message notification for that chat will go away.
Assumptions	N/A

Browse message history

Use Case Number	12
Summary	The user needs to be able to browse past messages
Actor	Application user
Trigger	Message scrollbar
Primary Scenario	User uses the message scrollbar to view the past messages
Alternative Scenario	N/A
Exceptional Scenario	N/A
Pre-Condition	User must be logged in. User must be a member of the chat
Post-Condition	N/A
Assumptions	N/A

3.3 Behaviour Requirements

3.3.1 Use Case View



The two actors in the system are the new and return application users. The new user must register a new account to become a return user. A return user is able to participate in every use case except registering.

Receive Message Notification

A user will receive a message notification only if another user sends a new message in chat. Users are only able to send messages if they are in a specific chat. The message notification will popup on the chats interface next to the corresponding chat panel.

View Specific Chat

A user must be able to view specific chats. This use cases requires the user to first login and then choose the specific chat, that he/she wishes to view, from the chats interface.

4. Other Non-functional Requirements

4.1 Performance Requirements

- Time requirements
    - Any messages sent will not take more than 2 seconds to appear in chat.
    - Navigating between the chat and chats interfaces will not take more than 3 seconds
    - New message notifications will not take more than 5 seconds to appear on the chats interface
    - Verifying an account login or creating a new account will take more than 5 seconds.
- Storage requirements
    - The database needs to be able to store at least 1GB per user. Information that will be stored are user and chat information and chat messages.
    - The database needs to be able to store information from at least 1000 users.
- Concurrency requirements
    - The application must be able to handle at least 1000 concurrent users. In the event that the applications becomes increasingly popular, steps towards supporting more concurrent users must made.

4.2 Safety and Security Requirements

- Passwords will be saved as hash values within database.
- Chat dialogue will be encrypted within the database and before messages are sent.
- All sensitive information will be encrypted or hashed before being stored or transmitted. This application will be moderately secure.
- Safeguards for malicious account logins will not be implemented.
- Messages will be stored until all participants have left a chat.
- All hashed values will use a rotating salt.
- After 5 failed login attempts, the user will be locked from login for 10 minutes.

4.3 Software Quality Attributes

4.3.1 Security

- The encryption method will be verified each month to ensure the data transfer and storage of previous messages remain secure.
- New salts will be introduced to the hashing algorithms on a weekly basis.
- FTPS establishes an encrypted connection between the user and server for all data transactions.

4.3.2 Maintenance

- Chat channels over 30 days old will be removed from the database to free up space and improve performance.
- Multiple servers will host and mirror the databases for load balancing and simplicity when maintaining systems.
- Inactive accounts will be removed after one calendar year to maintain reasonable sign-in times.

4.3.3 Availability

- The application is expected to have zero down time.
- Users always have access to their accounts and previous messages.
- Chat channels will be accessible at any time.

4.3.4 Usability

- Each function should be intuitive and easily accessible.
- All menus should follow the same structure and format.
- Pages, buttons and other components will be non-blocking and respond quickly.

5. Planning

5.1 Team Structure

- Team member 1, Aaron Lam: Lead Frontend Developer, Team Consultant
- Team member 2, Jerry Zhu: Backend Developer, Lead Tester
- Team member 3, Chris Sellers: Lead Backend Developer, Project Leader

5.2 Estimation and Feasibility (BASIC SCHEDULE)

We plan to keep up with the software engineering course syllabus. Our schedule will be as close as the CS410 course schedule. All the listed functionality will be minimum and implemented by the end of this quarter.

Basic version functionalities

- Account functions
    - Register new account
    - Login/Logout to account

- Chat functions
    - Start chat
    - View chat

- Messaging functions
    - Send messages

Sprint #1 Backlog

- Register new account
- Login/Logout to account
- Start chat

Sprint #2 Backlog

- View chat
- Send messages

Sprint Schedule

Each sprint will last approximately a week. A sprint will be decomposed into three phases.

- Phase 1: Design (2 days)
- Phase 2: Implementation (3 days)
- Phase 3: Testing/Bug Fixing (2 days)

5.3 Process Model

All members will be participate in every aspect of the software development process. Each member will designate a majority of their effort on specific parts based on expertise and experience.

Since Aaron Lam is the most knowledgeable in android application development, he will be responsible for leading the creation and design of the graphical user interface. In addition to being the lead front-end developer, he will act as the team consultant: Offer advice and assistance in resolving issues the team encounters.

Chris Sellers and Jerry Zhu are more familiar with back-end development, so they will designate most of their time designing the software interface, writing the implementations, and integrating the software components. More specifically, Jerry will be in charge of designing and performing tests: Unit tests, integration tests, and system tests. Chris will be responsible designing the software architecture. As project leader, he will set up sprint goals for the team. In addition, he will set up team meetings to discuss issues and evaluate progress.

6. Conclusion

A strong foundation is essential when developing any mobile application. It ensures that the application does not stray away from its core values. In this case, the core values of this proposed IM application are high usability and extensibility: the application should be intuitive to users and extensible for future projects. When a mobile application loses its core value, its purpose and auxiliary features become obscure and superficial, respectively.

To prevent that scenario from occurring, this SRS was developed to provide clarity about our project. Each portion of the project is described along with the requirements and specifications related to each component. The development and implementation of each component will be more complex than the descriptions provided within the SRS. Each description and explanation within this document is intentionally written to provide a refined but broad understanding of the product. This will also be a reference point for all developers during the development and maintenance stages.

Appendix A – Data Dictionary

Data Item	Functional requirement	Input	Output	Description	State Variable	Possible States
User account information	User registration	Phone number, password	Account created notification	Register new account	isRegistered(String phoneNumber)	True, False
Account setting	Manage setting	Phone number, password	Account info changed notification	Change phone number/password	hasChanged()	True, False
Phone number	User login	Phone number	N/A	Used for validating user's login	String phoneNumber	Constant
Password	User login	Password	N/A	Used for validating user's login	String password	Constant
Message notification	Receive notifications	Send message action	Notification pop-up	User is notified when there's a new unread message	hasRead(Message newMessage)	True, False
Chat User Limit	Start chat	User limit	User limit restriction set	User specifies max user limit in a chat	int userLimit	Constant
Message	Send message	Message text	Triggers new message notification to others	User sends message to chat for others to view	hasSent(Message newMessage)	True, False
Chat ID	Join chat	Chat ID number	Allows access to chat	User joins chat by entering chat ID	isValidID(int chatID)	True, False
Chat member	View chat	Phone number	Allows access to chat	User can view chat if he/she is a member of the chat	isMember(String phoneNumber)	True, False