Bhavan Patel
Manpreet Singh
Umaimah Syed
Amna Tiwana

Phase 5

Software Requirements Specification

Revision History

Date	Revision	Description	Author
06/12/2020	1.0	Initial Version	Team
06/24/2020	**1.1**	**Worked on UML diagram**	**Team**
06/14/2020	1.2	Made the flow chart	Manpreet Singh
06/14/2020	1.2	UML Diagram Design	Team
06/15/2020	1.3	SRS Document Added more content	Team
6/16/2020	1.4	SRS Finished First Draft	Team
7/7/2020	1.5	Started Coding	Team
7/8/2020	1.6	Updated UML Design	Team
7/15/2020	1.7	Updated UML Design, Specification	Team

Table of Contents

1. Purpose

Purpose of our program, multiuser communication system. Provides a service for its users allowing them to communicate thew the internet one on one chat.

1.1. Scope

Our program will is simple fast way to communicate

1.2. Definitions, Acronyms, Abbreviations

IP-Internet Protocol

1.3. References

Use Case Specification Document – Step 2 in assignment description

UML Use Case Diagrams Document – Step 3 in assignment description

Class Diagrams – Step 5 in assignment description

Sequence Diagrams – Step 6 in assignment description

1.4. Overview

Our Program allows users to communicate one on one over the internet.

2. Overall Description

2.1. Product Perspective

Actor-User/client

Actor-Developer

Use Case ID: {User}

Use Case Name: {Anyone Who is going to be using our chat service}

Relevant Requirements: * {User needs a machine that can install software correctly }

Primary Actor: {User }

Pre-conditions: {Need Valid Email and set valid password, and Stable internet connection}

Post-conditions: {Log out, close the program completion of this use. Effects on other systems and actors may also be described.}

Basic Flow or Main Scenario: {

1.User enters the login info
2. Server checks info is valid return the check is true
3. Allowing user access to the chat room and account details
4.Lookup Friends to Add
5.System Finds Friend and Connects user with friend
6.Allows access to chat
7.Log out
8.System Saves Data and chat
}

Extensions or Alternate Flows: {

1. User Enters Login Info Incorrectly
2. Server Checks info and returns False
3. User enters wrong information 3 times in row
4.System Prompts user to reset password
5.User enters new password
6.system updates password
7.User enters new login info
8 . System Checks Info returns true
9.repeat steps in main scenario 3 onwards }

Exceptions: { Incorrect password
User not found,

user offline.}

Use Case ID: {Developer}
Use Case Name: {Has access to code, allowed to make changes to program and update as needed.}

Relevant Requirements: * {Also has designed the program in way that none developers can't access it. Also be able to code the program and all features listed correctly}

Primary Actor: {Developer }

Pre-conditions: {Has access to perform updates to code using IDE.

Post-conditions: {Check and update program if needed}

Basic Flow or Main Scenario: { To Update or edit code
1. Use IDE to change or update code
2.system update server
3.Close IDE

Extensions or Alternate Flows: { Abusive User that needs to be Taken off the Platform Revoke access
1. Access Data Base and Find Username
2.Send email to user their access has been revoked
3. Update data base take note this email is put under revoked
4.delete account from data base
5.Finish update

Exceptions: {fail to update, If OS is updated to new version our software might be out of date, data can get corrupted on server end}

2.2. Product Architecture

Account, User, ServerSide, ClientSide, DeleteAccount, Message,

2.3. Product Functionality/Features

-Login Logout
-reset password
-Connect With friends -Look up using email or username
-Delete Account
-send and receive messages
-Administer Can Suspend Account

2.4. Constraints

-No Group chats

-No multiple windows for chat

-Only Text messaging sending and receiving allowed

-Need Valid Email address from user to create account

-Chat can not be saved

2.5. Assumptions and Dependencies

List appropriate assumptions

-Need Valid Username from user to create account

-Limited server recourses

-User will install the run the code on his JVM machine or JVM compatible machine

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Common Requirements:

-Internet access

-Need A account/Create

-Need Operating System

3.1.2. __Account___ Module Requirements:

-Needs name

-Needs passwords

-Needs Username

-Needs Security Answer

Validates All the information entered during login for user returns true or false only

3.1.3. __User ___ Module Requirements:

-Sign Up user Need Valid Email

-Password- minimum of six to 20 letters and numbers allowed.

-Log in or Log out

-Prompt user if Information is incorrect

3.1.4. _ ClientSide____Module Requirements:

-Need one Friend to Chat

-Need able to communicate with server

-Correct Format of Text for example as string

-Needs to be logged in order to chat with users

-Needs to create the socket when user log in

-Needs to close the socket user log out.

3.1.4_ ServerSide___ Module Requirements:

-Connect Two IP address with host

-Need to be able to communicate with user

-Correct Format of Text for example a string

-Account needs to get verify before giving the object to user

-Needs to be able to remove user, change password for user, add user and search user.

-need to create the socket for each user, so that multiple clients can login

-need to close socket as the user log out

3.1.5. _ Delete__Module Requirements:

-delete the username and password from the server

- delete friend list

3.1.6. _ GUI__Module Requirements:

-Design the Layout of the Chat Box and Login
-Set the font and color

3.1.7. _ UserInterference_e Requirements:

-User Command Process

3.1.8. _ Message__Module Requirements:

- The message needs to be string in order to process
- Need to able to connect with the ServerSide class and Userside class, to create the path to send and receive messages

3.2. External Interface Requirements

N/A

3.3. Internal Interface Requirements

We need at least two user accounts for our platform to work as attended fully. Need access to the server at all times for most of the key functions to work correctly.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

-Message needs to encrypt
-Password data entry is obscured

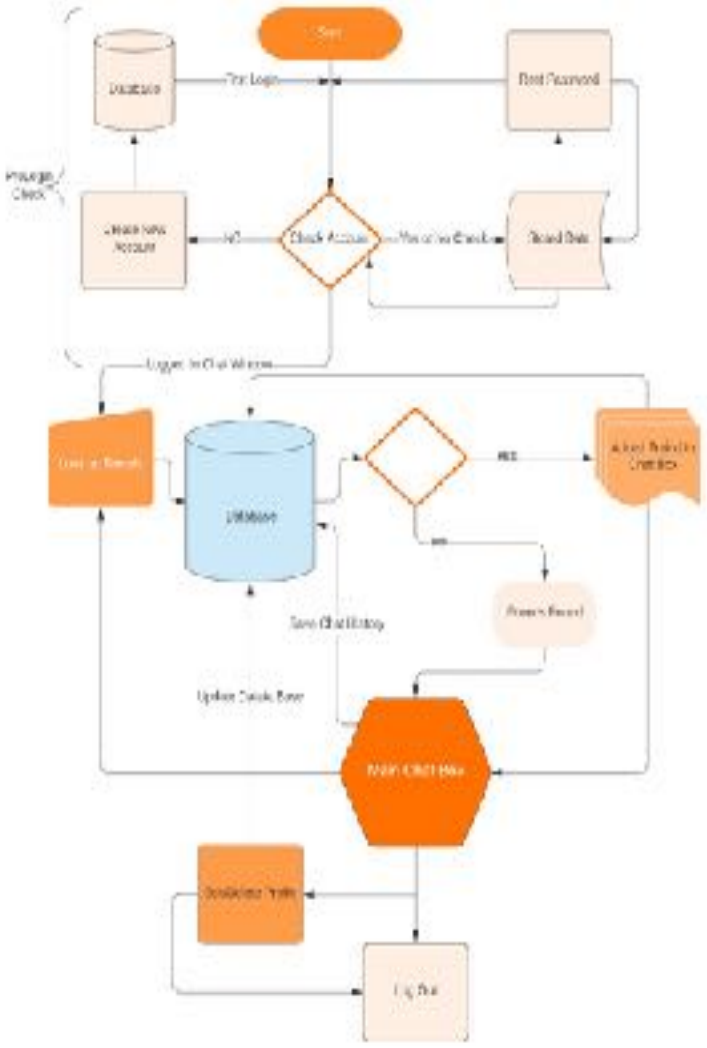
4.2. Environmental Requirements

N/A

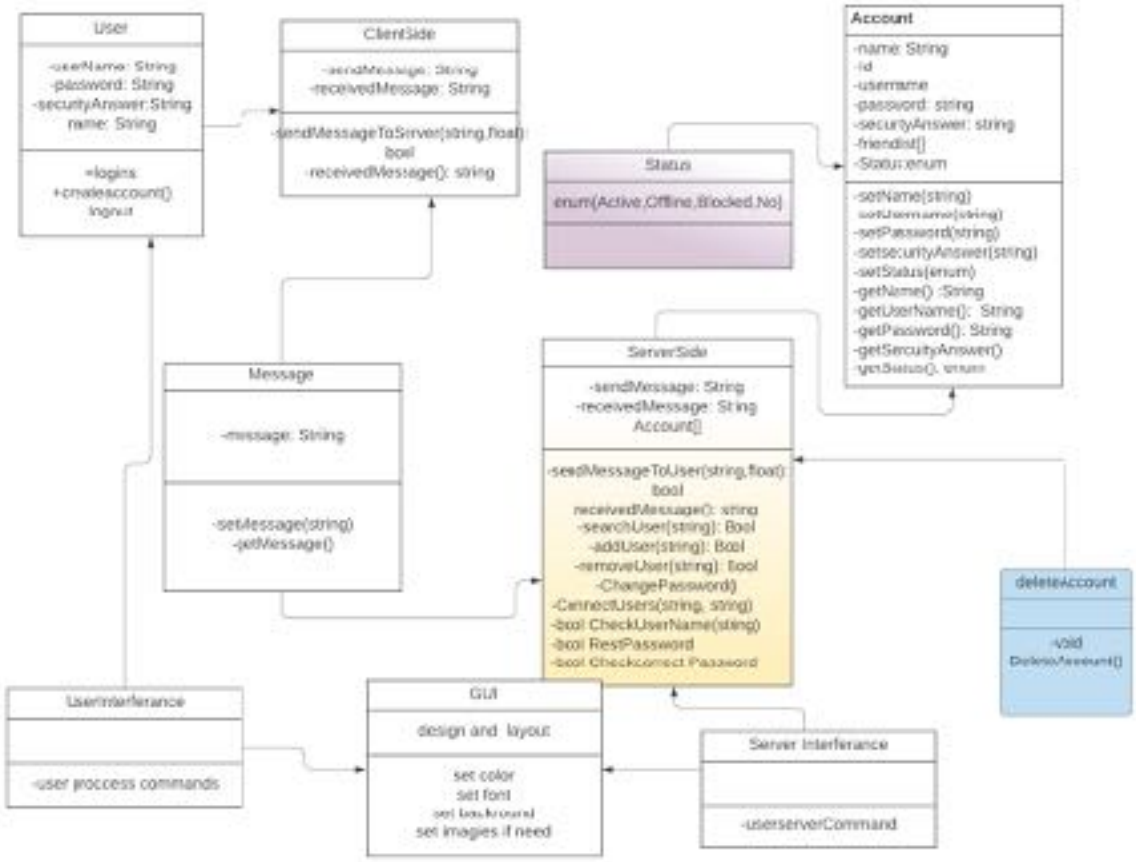
4.3. Performance Requirements

Windows OS and Mac OS any machine that can run java

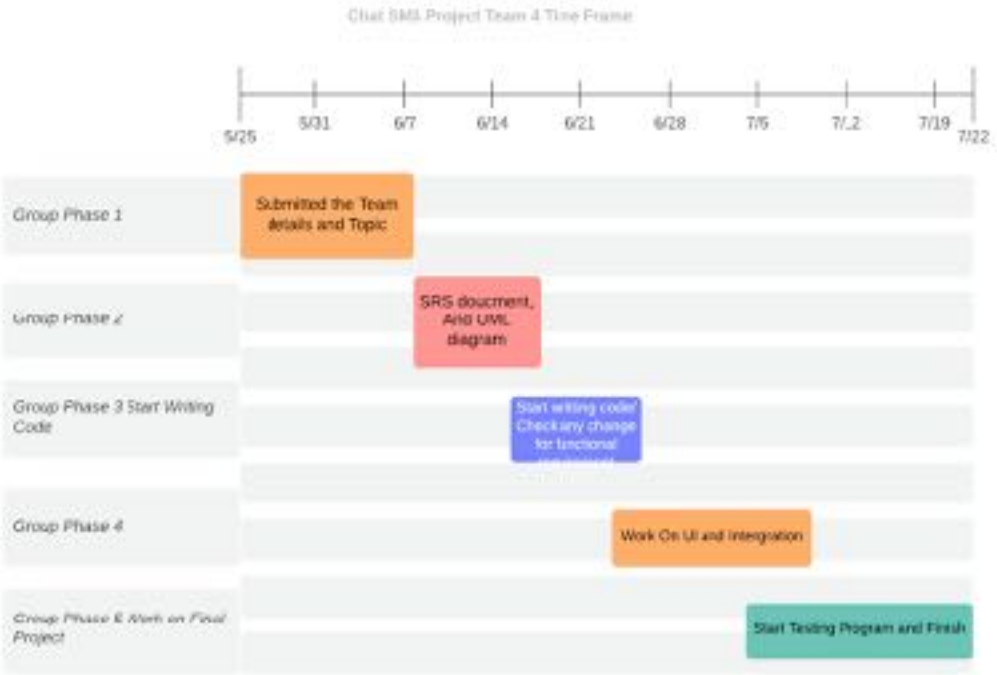
Flow Chart



UML Diagram



Project Schedule



Roles & Responsibilities

Server Side:
Team members: Umaimah Syed, Bhavan Patel
Roles and responsibilities: .

- Work on the sever code
- Some of the GUI interface
- Find the potential security loopholes.
- Work on deleteclass

Client Side:
Team Members: Amna Tiwana, Manpreet Singh,
Roles and responsibilities:

- Work on the client-side code.
- Implement on the user interface.
- Update the appropriate version of the code with the sever side developer.

- Work on the user class

Project Status Report

Group Progress:
- Narrowing down to specific needs and updated the design of overall project.
-Removed unnecessary classes
-Worked on SRS document's changes and UML class diagram.
-Did the most of coding for the project.
-Created the foundation of all classes
-Updated the SDS document
-Team is assigning tasks to individuals along with time Frame narrowing down
-Pushed classes to git repo of the team project

Individual Progress: