Bhavan Patel

Manpreet Singh

Umaimah Syed

Amna Tiwana

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 |
| 7/25/2020 | 1.8 | Reviewing the Document | Team |
| 7/26/2020 | 1.9 | Final Document | Team |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Assignment #2 – Template 1

1. Purpose iii
   1. Scope 5
   2. Definitions, Acronyms, Abbreviations 5
   3. References 5
   4. Overview 5
2. Overall Description 6
   1. Product Perspective 6
   2. Product Architecture 8
   3. Product Functionality/Features 8
   4. Constraints 8
   5. Assumptions and Dependencies 8
3. Specific Requirements 9
   1. Functional Requirements 9
      1. Common Requirements: 9
      2. \_\_Account\_\_\_ Module Requirements: 9
      3. \_\_Add Account/Login\_\_\_ Module Requirements: 9
      4. \_Chat Box\_\_\_\_ Module Requirements: 9
      5. \_Listen and Receive\_\_\_\_ Module Requirements: 10
      6. \_Delete\_\_\_\_ Module Requirements: 10
   2. External Interface Requirements 10
   3. Internal Interface Requirements 10
4. Non-Functional Requirements 11
   1. Security and Privacy Requirements 11
   2. Environmental Requirements 11
   3. Performance Requirements 11

# Purpose

The purpose of our program is to provide a service for its users allowing them to communicate with other clients through the internet or the local network.

## Scope

Our program is designed to communicate in a simple and fast manner.

## Definitions, Acronyms, Abbreviations

IP - Internet Protocol

GUI – Graphic User Interface

UML – Unified Modeling Language

SRS – Software Requirements Specification

SDS – Software Design Specification

## 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

## Overview

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

# Overall Description

## 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 name and password, stable

internet connection}  
 Post-conditions: {Log out; close the program

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 and returns the check is true

3. Allowing user access to the chat room and account details

4.Lookup friends to add

5. System finds and connects user

with a 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: \* {This programmed is designed in a way that

developers can have access to it.

Also be able to code the program and all features listed correctly}

Primary Actor: {Developer }

Pre-conditions: {should have access to the IDE so they can update the

code. }

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 that 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}

## Product Architecture

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

## Product Functionality/Features

-Login Logout

-reset password

-Connect With friends -Look up using username

-Delete Account

-send and receive messages

## Constraints

-No Group chats

-No multiple windows for chat

-Only Text messaging sending and receiving allowed

-Chat cannot be saved

## 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

# Specific Requirements

## Functional Requirements

### Common Requirements:

-Internet access

-Need an account

-Need Operating System

### \_\_Account\_\_\_ Module Requirements:

-Needs username

-Needs passwords

-Needs Security Answer

-Needs status of the user (online, offline, blocked)

-Needs Printwriter object to chat with other users or with server

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

### \_\_LoginOnly \_\_\_ Module Requirements:

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

-Log in or Log out/close

-Prompt user if the login information is incorrect

### \_\_SignUp \_\_\_ Module Requirements:

-Will need a username, password, and a security answer

-After creating the account, it will log you in

### \_ ClientSide\_\_\_\_Module Requirements:

-Need one friend to chat

-Need to be able to communicate with server

-Correct format of text for example as string

-Needs to be logged in to chat with users

-Needs to create the socket when user log in

-Needs to close the socket user log out.

-Message GUI design needs login confirmation from ServerSide to chat with a user

### 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 be verified 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

### \_ Delete\_\_Module Requirements:

* Delete account
* Delete friend list
* Not implemented in the current version of the chat system

### \_ GUI\_\_Module Requirements:

-Design the Layout of the Chat Box: Signup and Login button

## External Interface Requirements

N/A

## Internal Interface Requirements

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

# Non-Functional Requirements

## Security and Privacy Requirements

-Message needs to be encrypted

-Password data entry is obscured

## Environmental Requirements

N/A

## Performance Requirements

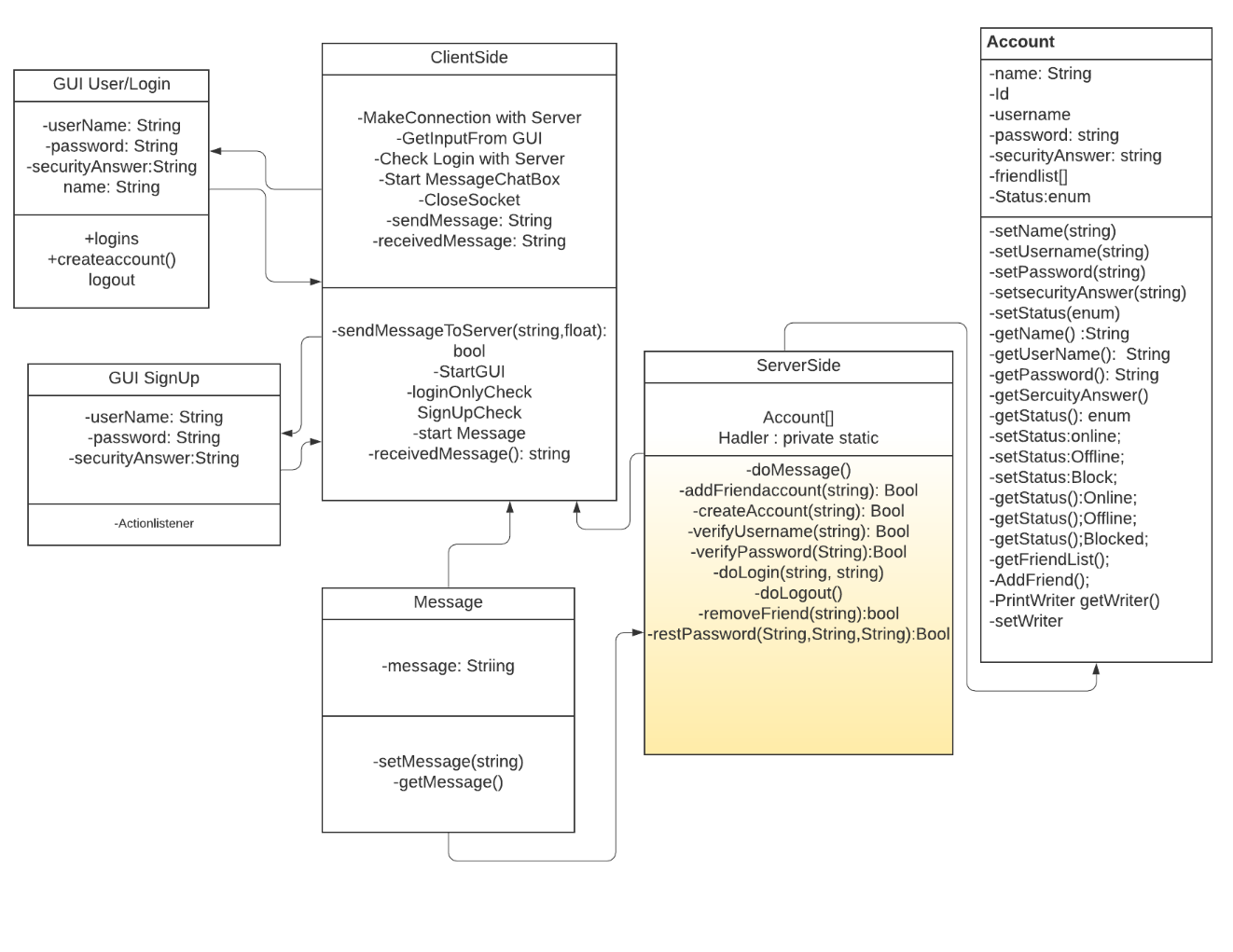
Windows OS and Mac OS (any machine that can run java)

**Flow Chart**

A close up of a map

Description automatically generated

**UML Diagram**

****

**Project Schedule**

**A screenshot of a social media post

Description automatically generated**

**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, Umaimah Syed

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