Software Requirements Specification

for

Online Chat Application

Version 1.0

Prepared by Sanchit Singh (2K18/CO/317), Simarpreet Singh (2K18/CO/353)

Delhi Technological University

Table Of Contents

Revision History :

1. Introduction:

1.1. Purpose :

The purpose of this project is to create a chat application with a server and enable users to chat with each other.

To develop an instant messaging app to let users communicate seamlessly with each other in real time.

To develop an easy interface so that a newcomer/novice person should be able to avail all its benefits.

The development of this project centered on the development of a message protocol that would allow the application to properly log in users, send messages, and perform system maintenance.

1.2. Document Conventions:

- Client - any user who is using this type of service
- Server - the web server which is responsible for real time communication between clients

1.3. Scope:

- Broadcasting Chat Server Application is going to be a text communication software, it will be able to communicate between two computers using point to point communication.
- The limitation of Live Chat is it does not support audio conversations. To overcome this limitation we are concurrently working on developing better technologies.
- Companies would like to have a communication software wherein they can communicate instantly within their organization.
- The fact that the software uses an internal network setup within the organization makes it very secure from outside attacks.

1.4. Overview:

Section 2 of this document describes overview of the system in terms of General characteristics of the system, information about possible users of the system, possible constraints on the system, operating environment of the system etc. The Section 3 describes in detail functional as well as non-functional requirements of the system.

1.5. References:

- IEEE 830¬1998 standard for writing SRS document.
- Software engineering by KK Aggarwal
- Example- SRS document

2. Overall Description:

2.1. Product Perspective:

There is a two way communication between different clients and servers. This chat application cannot be used for group discussion. It allows users to find other logged in users.

2.2. Requirements Elicitation:

2.2.1. Use Case Approach:

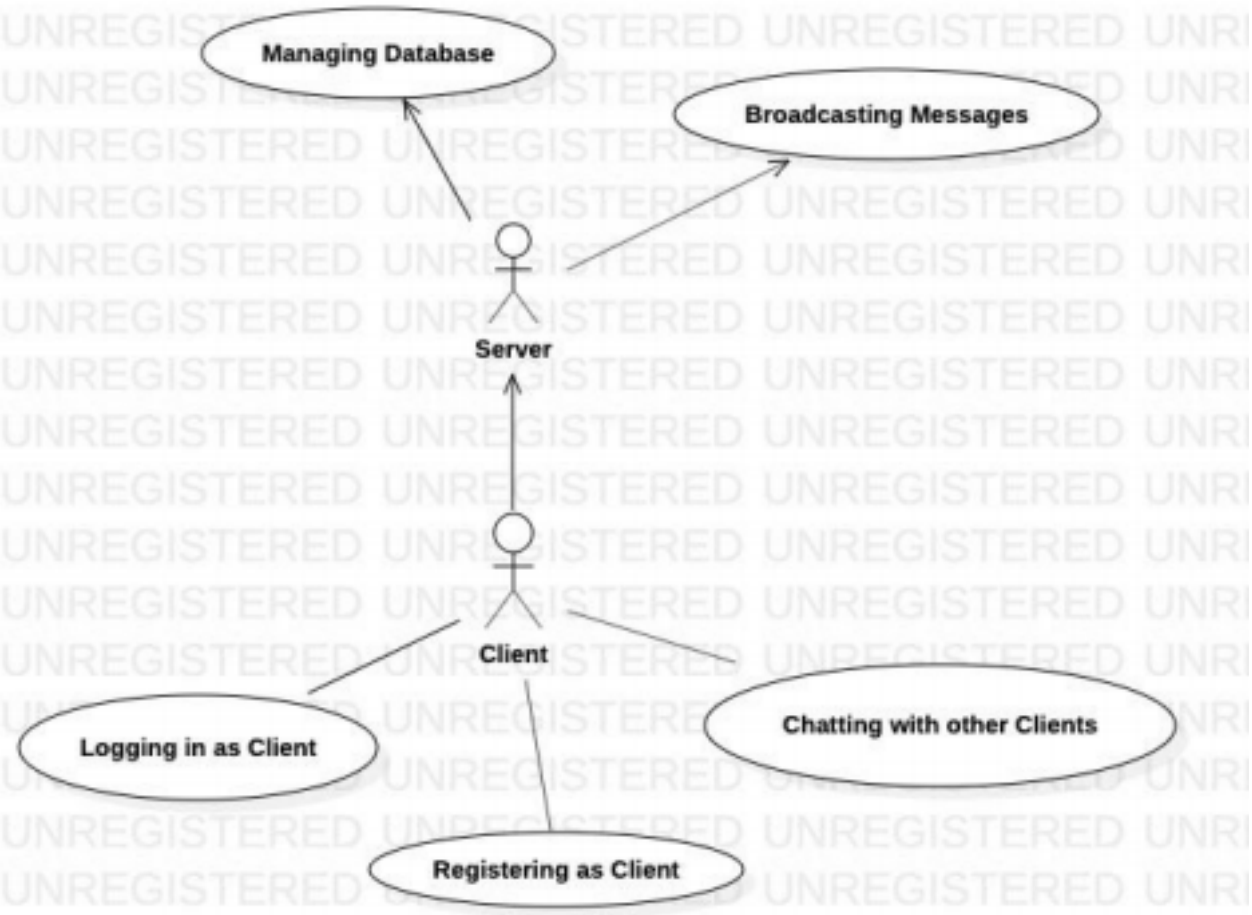


Fig 1. - Use Case Diagram of the Project

Use Case Description:

Use Case	
Use Case Name :	Registering as a Client
Actors:	Client 1
Description:	Client registers with a name, username and a password
Pre-Conditions:	The client has opened up the website and clicked on Signup
Post-Conditions:	The client gets a unique username and a password registered into the system.
Normal Flow:	**1.0 Registering as a client**

	1. Client goes to the website 2. Client enters Username, Password and Phone Number 3. Client Presses Sign up.
Alternative Flow:	**1.1 Username already exists** 1. Client enters another Username, Password and Phone Number. 2. Client Presses Sign up.

Use Case	
Use Case Name :	Chatting with another Client
Actors:	Client 1 and Client 2
Description:	Client 1 Adds Client 2 as a Friend and sends them a message on the chat screen
Pre-Conditions:	Client 1 knows Client 2's Username and Phone Number
Post-Conditions:	Client 1 and Client 2 are friends and can chat with each other
Normal Flow:	**1.0 Adding a Friend** 1. Client 1 presses the menu and selects Add Friend option 2. Client 1 enters Client 2's Username and Phone Number 3. Client 1 clicks Send Request 4. Client 1 and Client 2 are now friends. **2.0 Sending Client 2 messages** 1. Client 1 selects Client 2's name from the Chat list 2. Client 1 sends Client 2 a message.
Alternative Flow:	**1.1 Username or Phone Number of Client 2 is incorrect** 1. The website refreshes 2. Go back to **1.0**

2.3. Product Functions:

The Chat Application Supports the Following Functions:

- Functions by which the client can view messages

- Functions by which client can add friends
- Function by which client can delete friends

- Function by which client cannot send hate/offensive words
- Function by which client can login to the system
- Function to manage client signup by unique username and phone number

The access to these different functions is user restricted. E.g only the user can view their own chat.

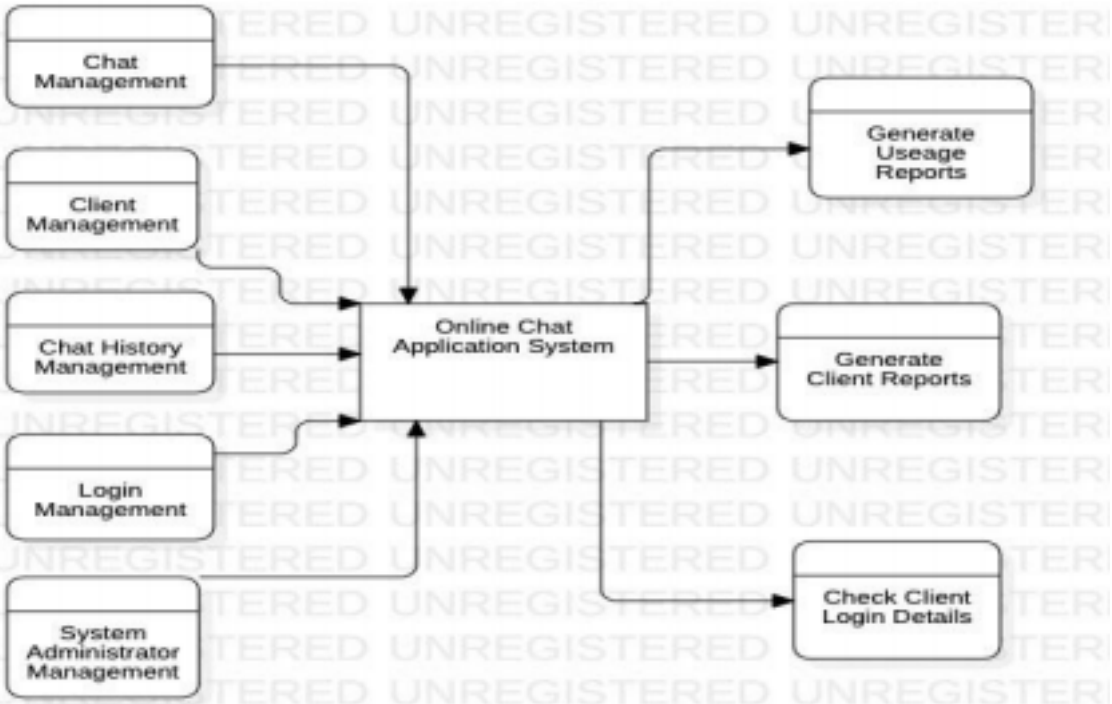
2.4. User Characteristics:

The users of this chat application are the Clients, they just need to know how to use a web browser, the website has an easy to understand GUI.

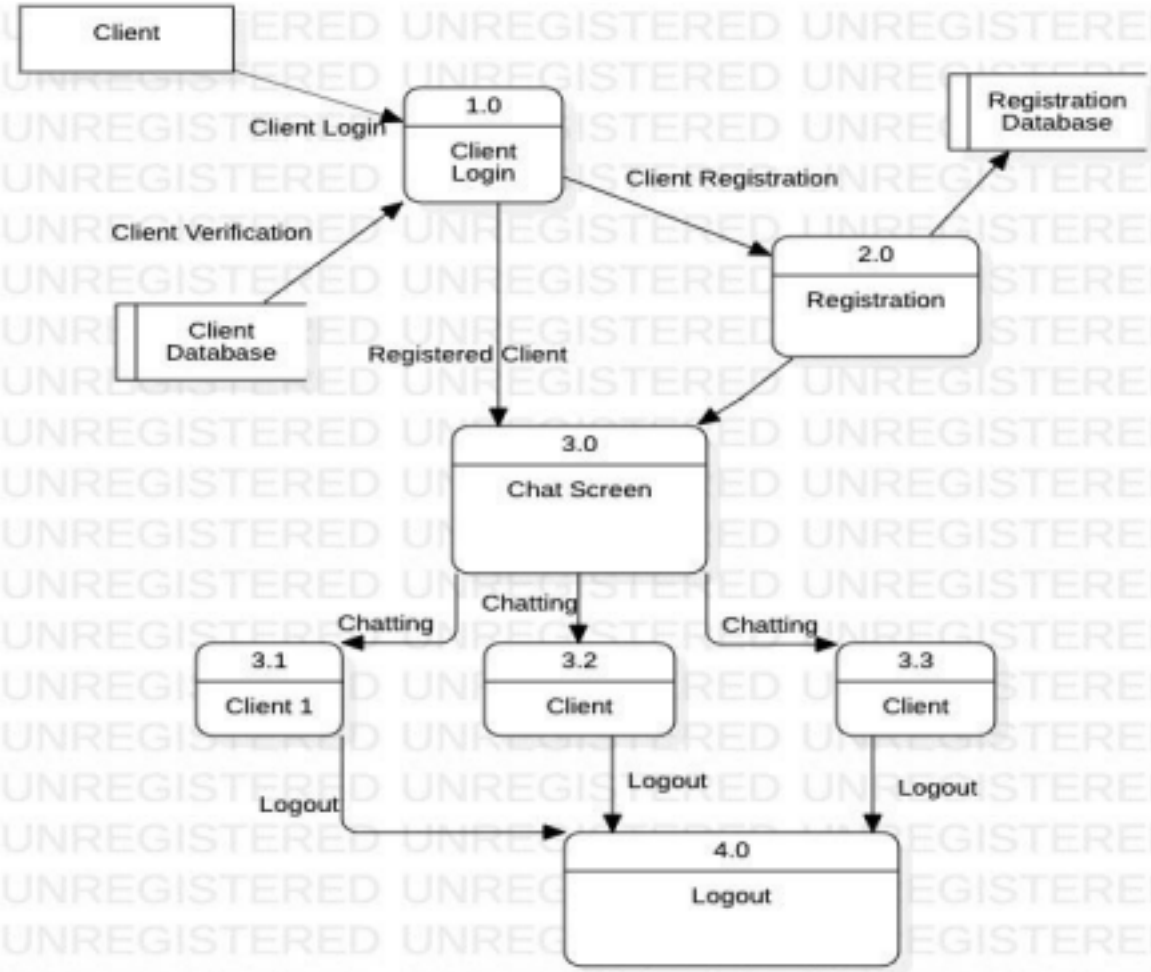
2.5. User Constraints:

- There should be an active LAN or internet connection on all terminals.
- Clients should know each other.
- There can be multiple clients.

2.6. Data Flow Diagram:

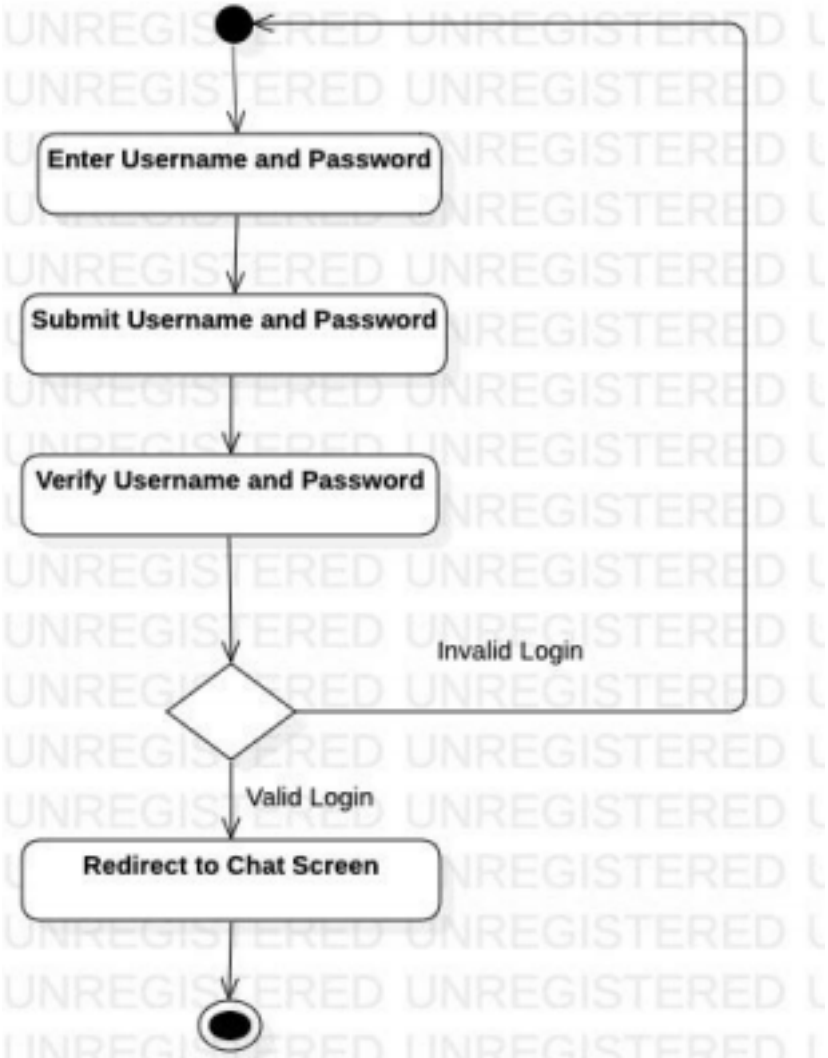


Level 0 DFD of Chat Application System

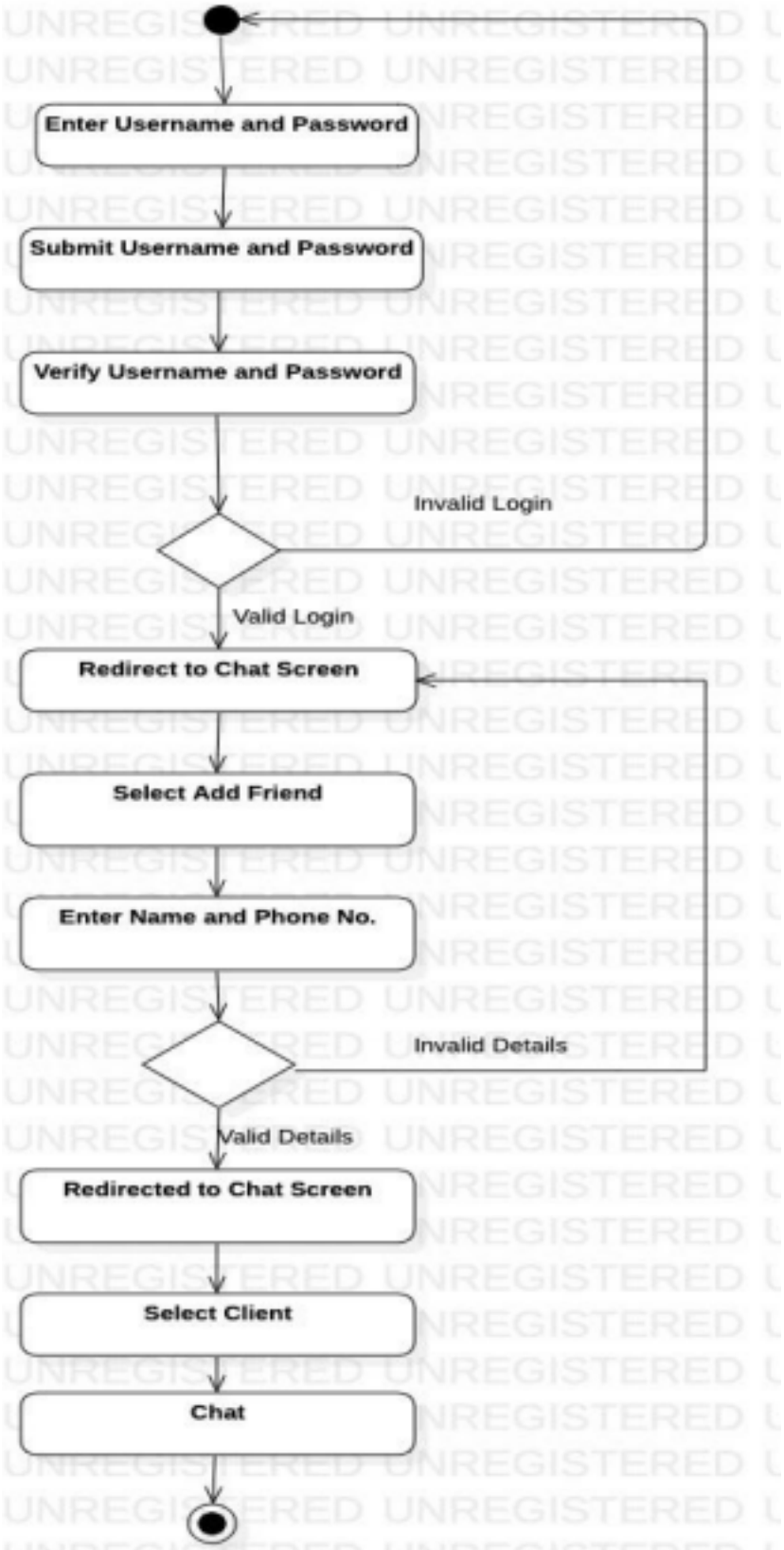


Level 1 DFD Diagram of Chat Application

2.7. Activity Diagram:



Activity Diagram for Logging into Chat Application



Activity Diagram for Adding another Client and Chatting

3. External Interface and Requirements: 3.1.

Operating System Requirements:

- Operating system (such as window 7, 8, xp, MacOS, Linux etc).

3.2. Software Requirements:

- A Web Browser (such as Google Chrome, Safari, Firefox, Opera etc).

3.3. Hardware Interfaces:

- Minimum requirements will be as follows:
  - 128 MB RAM required.
  - Processor with speed of 500 MHz.
  - Internet or LAN connection.
  - MOUSE: 2 or 3 button mouse
  - KEYBOARD: 101 key Keyboard

4. Specific Requirements:

4.1. Functional Requirements:

4.1.1. Login Menu function

This functional requirement is for prompting the user with the option to register for the chat application, logging in, or exit the program. It will take the form of a GUI **Register function (Login Menu aspect)**

This aspect of the login menu will ask the user for the name, username, and password of the client. It will check if the username has been taken and will close if the username is not taken and will go back to the main login menu.

4.1.1.1. Login function(Login Menu aspect)

This aspect will ask for the username and password. Errors will occur if a space is left blank, the username doesn't exist, or the password doesn't match with the username. If the username and password matches, you are online and able to message anyone else online.

4.1.1.2. Exit(Login Menu aspect)

This aspect will close the chat application.

4.1.2. Online Menu function

This function will give the option of seeing who is online, the option of sending a message to whoever is online, and the option to logout.

4.1.2.1. Who is online(Online Menu aspect)

This aspect will show who is online and will give the user the ability to click on a user and send a message to that user.

4.1.2.2. Send a message(Online Menu Aspect)

This aspect will give the user the ability to send a message to whoever they want who is online and selected by the user.

4.1.2.3. Logout(Online Menu aspect)

This aspect will give the option to logout of the chat application and will go back to the login menu..

4.2. Non - Functional Requirements:

4.2.1. User Friendly:

The chat application needs to be user friendly, when using its user interface.

4.2.2. GUI:

By using GUI's, it should make the application more user friendly and better to use instead of a command line. Buttons will be used.

4.2.3. Privacy:

Messages shared between users should be encrypted to maintain privacy.

4.2.4. Robustness:

In case a user's device crashes, a backup of their chat history must be stored on remote database servers to enable recoverability.

4.2.5. Performance:

The Application must be lightweight and must send messages instantly.

4.2.6. Block Offensive Words:

The Application must be able to block a list of offensive words from its outgoing messages.

4.3. Performance Requirements:

It will support all devices it is installed upon, the updates from database might get slower and laggy with every increase in number of devices initially about 100 users are easily supported.

4.4. Security Requirements:

A login system will be present for clients for added security which will involve each client having a Unique ID.

5. Process Model:

5.1. Spiral Model:

This model is a combination of Waterfall and Iterative model where each phase begins with a Design goal and Ends with client reviewing. So our software is developed in a series of incremental releases. It will be worked upon in multiple stages:

- The Communication stage where the requirements are discussed. The next phase will be planning where estimation of the project, scheduling the project and risk analysis while developing the project is done.
- The Modeling phase where analysis and design of the software is done.
- The Construction phase coding followed by testing is done.
- The Deployment phase of the project is delivered along with support. Then the feedback is taken from the customer.

This is how our first task region is completed. So after the first task region concept development takes place followed by software development in the next task region. Then system in enhancement and system maintenance takes place in the next tasks region accordingly.

Spiral model is useful because it's a large project, the releases are required to be frequent , risk and cost evaluation is there, the requirements are unclear and complex and the images might be required any time. So additional functionality or changes can be done at a later stage. Cost estimation becomes easy and development is fast. Also the features are added in a systematic way. Most importantly there is always space for customer feedback.