Online Chat Application

Wesley King - Sindhu Balakrishnan

# **Project Overview**

An online text chatting application which allows users to exchange messages with each other.

# **Functional Requirements**

* As a user of the client application, I would like to exchange messages with other client application users.
* As a user of the client application, I would like to have a one to one conversation with other client application users.
* As a user of the client application, I would like to have a multi-user conversation with other client application users.
* As a user of the client application, I would like to have the option to set my presence to one of Available, Unavailable (Invisible), or Offline.
* As a user of the client application I would like my presence information to be available to other client application users.
* As a user of the client application I would to see the presence information of other client application users.
* As a user of the client application, I would like to have the ability to add users to a contact list.
* As a user of the client application, I would like to have the ability to remove users from my contact list.
* As a user of the client application, I would like to have the ability to block users from my contact list.
* As a user of the client application, I would like to see the chat history of conversations with other client application users.
* As a user of the client application, I need to authenticate with the chat server.
* As a user of the client application, I would like to have single sign on access.
* As a user of the client application, I would like to be able to send messages to users by

specifying ‘#’ and the user login.

* As a user of the client application, I would like to have the option to send emotions.
* As a user of the client application, I would like the server to suggest responses based on my previous chat history.
* As an administrator of the server, I would like to add and manage users in the system.

# **Non-Functional Requirements**

* The application should have both a client(s) and a server.
* The client will run on the windows operating system.
* The server will run on an Unix operating system.
* The client application will use C# under Windows for the client side
* The server application will use Go under a Unix based operating system.
* The server application will use sqlite database to store user information and conversation history
* The server application will use a python module with tensorflow to provide suggested responses.

# **Use-Cases**

|  |  |
| --- | --- |
| Scenario 1 |  |
| Description | A user can have a one on one conversation with another user. |
| Preconditions | User1 and User2 both exist in the system and have each other in their contact lists.  Actors: User1 and User2 |
| Normal Sequence | 1. User1 authenticates with the server. 2. User2 authenticates with the server. 3. User1 selects User2 from his/her contact list. 4. User1 sends a message saying “Hello :)” to User2. 5. User2 receives a message saying “Hello :)” from User1. |
| Postcondition | The message sent from User1 to User2 is stored in the database. |

|  |  |
| --- | --- |
| Scenario 2 |  |
| Description | A user can have a multi user conversation with other users. |
| Preconditions | User1, User2, and User3 both exist in the system. User2 and User3 are in User1’s contact list. |
| Normal Sequence | 1. User1 authenticates with the server. 2. User2 authenticates with the server. 3. User3 authenticates with the server. 4. User1 selects User2 from his/her contact list. 5. User1 adds User3 to the conversation. 6. User1 sends a message saying “Hello :)” inside the conversation. 7. User2 and User3 receive a message saying “Hello :)” inside the conversation from user1. |
| Postcondition | The message sent in the multi user conversation is stored in the database. |