MODERN SOFTWARE ENGINEERING

PROJECT MILESTONE- 2

TEAM 3

BATCHANGARI ROHITH CHANDRA GOUD

BHARGAV VECHA

VINAY KUMAR REDDY KANDULA

PRODUCT NAME: PRIVATE CHAT BOX

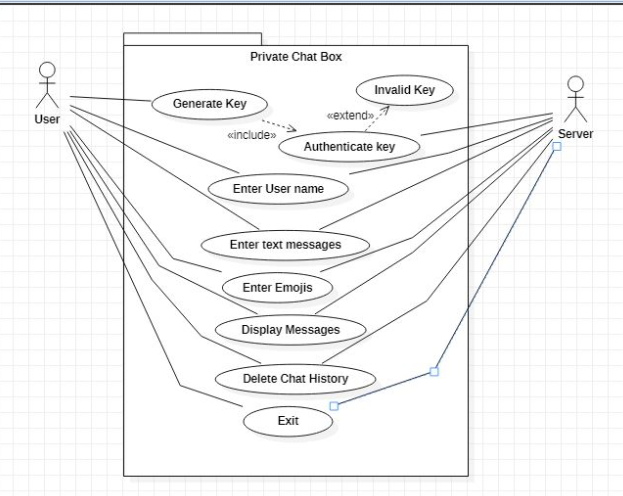
SYSTEM OVERVIEW:

Development of a mode of communication between users, by developing a chat box in order to provide privacy. The end user will have a user interface using which they can respond and create messages to be shared between each other. The potential audience of this product can be, students of a class, a group of friends, strangers who wants to share their insights on a topic and for users wanting to share messages or communicate. This software will provide users with a chat room where about 25 people at the same time can communicate with each other using text and emojis. The most important feature of this product is that it can be used without providing a mobile number, E-mail address or any social media account information unlike most of the application that are found on web. No personal information is shared on the internet and is known by the peer user as privacy is one of the most concerning issue in today’s world. The users can join a personal chat room by entering a specific key shared among them, to communicate and the whole chat will be deleted once they leave the chat session.

TECHNICAL REQUIREMENTS:

Technical capabilities required to perform this task are, having knowledge of Java Script, CSS (Cascading Style Sheets) and using Microsoft Visual Studio.

* Functional Requirements:

  
Actors: User, Server.

Entry Condition: The user either needs to generate the key to create a new chat room or enter the key to an existing chatroom.

Use Cases:

* **Generate Key**: When the user needs to create a new chat room, he needs to generate a key.
* **Authenticate Key**: The key is then authenticated in the server to check whether the key exists or not.
* **Invalid Key**: If the key entered is not present in the server, the user receives an error message stating the key is invalid.
* **Enter Username**: The user needs to enter a username which is a display name. It appears in the chat when we send the message.
* **Enter Text Message**: The user enters his message in the text area and presses enter. The message gets delivered in the chat room and other members can see it.
* **Enter Emojis**: The user can also communicate with emojis.
* **Display Messages**: The user can see the messages sent by other users.
* **Delete Chat**: When the users leave the chat room, the messages are all cleared in the server.
* Nonfunctional Requirements:

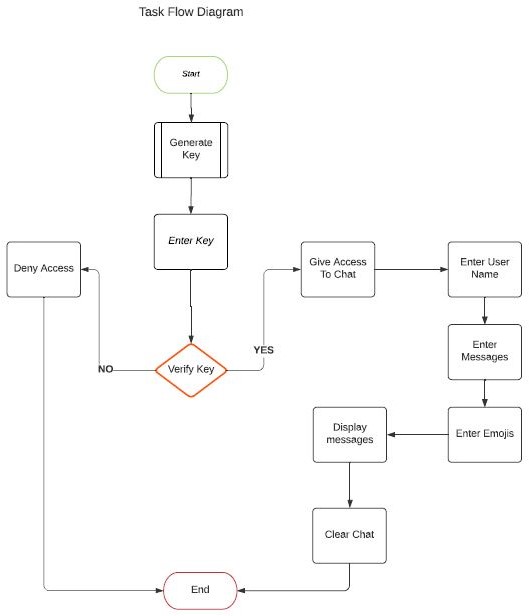
Security: As there is no personal information is taken, there is no issue of data theft. The messages in the chat room are also deleted when the room is left. This increases the security of data.

Capacity: The number of people allowed per room are restricted to 25.

Performance: As we use socket, it helps in connecting the threads and response time will be less. It also increases throughput as it orders the threads and processes them.

Reliability: As no information is taken from the user, there is no issue of loss of personal information.

* User Interface Specification: In the user interface, we can observe there will be option to enter the message, send emojis and view the messages.
* User Task Flow:



* Input/ Output Data Specification: The user enters the message in the text area and sends it into the chat box. The user can also communicate through emojis. He can also view the messages sent by other users present in the same room.

Acceptance Criteria:

* **How can I access the chat box?** A user generates a key and shares it with his classmates or friends. When they enter same key, they all enter a chat roomwhere they can talk to others through messages and emojis.
* **How secure is the chat?** Once we exit the chat room, the messages are cleared from the server.
* **What happens when members in the chat room message at the same time?** Based on the incoming timestamp of the message in the server, the messages are displayed in that order.

Verification/ Validation:

* The software undergoes testing on all its features.
* We check whether the key is generated and is getting validated or not.
* We check for messages getting delivered and received properly.
* The clearing of the messages is done at the server. It is done with utmost care as it must be cleared totally from the server. This feature undergoes rigorous testing.

Review of Progress:

* Finalized the requirements.
* Installed required software.
* Started creating socket for integrating threads.

Revised Schedule:

Integrating Emojify. 03/06/2021

Testing the development. 03/20/2021

Frontend Development:

Creating Web Pages. 04/03/2021

• User entry page

• Authenticating Room Key

• Creating Text Box

Integrating Frontend and Backend. 04/17/2021

Final Testing. 05/01/2021