Software Design & Implementation

Project Plan

**Group 30 :**

Hannah Ashna Jacob

Hassaan Naveed

Jarad Johnson-Bailey

Nicholas McCaig

# Requirements List:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Requirement** | **Priority** | **Implications** | **Tasks** |
| 1 | Users **must** be able to send, receive and view messages through the application | MUST | Without this feature, the application’s primary function would not be possible as a message exchange platform relies on users being able to send and receive messages | **T2.1.0:** Setup and Configure Client/Server using MQTT  **T4.0.0:** Implement User Interface |
| 2 | Users **must** be able to create chat rooms (rooms with more than two contacts) | MUST | A key feature in a messaging platform is the ability to chat with more than one person simultaneously, hence, the need for chat rooms | **T2.1.1:** Setup and Configure Client/Server using MQTT  **T3.2.0:** Create separate user classes |
| 3 | The user that creates the chat room **must** be classified as Admin | MUST | This is to prevent other users from making modifications to a chat room when they did not create it | **T3.0.0:** Create separate user classes |
| 4 | The moderator **must** be able to invite and remove users from a chatroom | MUST | This allows for a chat room to expand its user base numbers or remove certain members if necessary | **T3.1.1:** Create separate user classes |
| 5 | Moderators **must** inherit all the admin permissions; however, Moderators **cannot** demote the Admin | MUST | The functionality of the Moderator and Admin is essentially the same, however, the Admin is the owner of the chat room, hence, moderators should not be allowed to demote or remove them as owner | **T3.1.0:** Create separate user classes |
| 6 | Application **must** provide a friendly User Interface (UI) | MUST | A welcoming and friendly UI ensures user retention and ease of use. The UI should not drive users away due to its complexity | **T1.0.0:** Design User Interface |
| 7 | Users **must** be able to see the active users in the chat room | MUST | Allowing a user to view other active users allows them to know who is available to chat | **T4.0.2:** Implement User Interface |
| 8 | Users **must** be notified when a new notification is received | MUST | A notification system ensures that users are always up to date with new conversations happening in their chats and chat rooms | **T5.0.0:** Implement event listeners |
| 9 | Clients **must not** connect directly to other clients without a server or a broker | MUST NOT | This is necessary as it would otherwise present itself as a security risk | **T2.0.0:** Setup and Configure Client/Server using MQTT |
| 10 | A server or a broker **must** allow multiple authorised clients to connect to it | MUST | A message exchange platform is going to have several users on it; hence, the broker must be capable of connecting with multiple clients | **T2.0.1:** Setup and Configure Client/Server using MQTT |
| 11 | Users **must** only access their space after the login | MUST | Having users access another user’s space would breach security and privacy protocols | **T6.0.1:** Implement security features and protocols |
| 12 | Passwords **must** be saved securely locally | MUST | This is necessary as it would otherwise present itself as a security risk | **T6.0.0:** Implement security features and protocols  **T9.0.0:** Setup a local database |
| 13 | Application **must** adhere to all local (and international) privacy laws | MUST | With laws like the General Data Protection Regulation (GDPR), modern-day applications must respect and handle user data appropriately | **T6.0.2:** Implement security features and protocols |
| 14 | Application **must** list all the personal contacts in the contacts pane | MUST | Users want to be able to view their contacts list and start a chat easily, this method provides them with the necessary tools in an intuitive manner | **T4.0.1:** Implement User Interface |
| 15 | The admin **should** be able to promote and demote users to moderators in chat rooms | SHOULD | This allows the Admin to provide chosen users with privileges to make modifications to the specific chat room; especially useful in cases where it is a large chat room, and the Admin cannot manage this on their own | **T3.0.1:** Create separate user classes |
| 16 | The moderator **should** be able to create and delete channels in the chatroom | SHOULD | Creating discussion threads within a chat room prevents the primary discussion thread from being flooded with several concurrent conversations | **T3.1.2:** Create separate user classes |
| 17 | The moderator **should** be able to delete a user’s messages in the chatroom | SHOULD | In the case a user sends an inappropriate message to a chat room, the moderator can then remove it immediately to avoid the message distressing other chat room members | **T3.1.3:** Create separate user classes |
| 18 | Users **should** be able to change their status | SHOULD | Users may be preoccupied with other tasks or do not wish to be disturbed, hence, would like the ability to modify their availability status within the application | **T7.0.0:** Develop user profile and control panel |
| 19 | User **should** be able to change their details including their picture | SHOULD | A customization feature that allows a user to share their details with their contacts (i.e., profile photo, name, email) | **T7.1.0:** Develop user profile and control panel |
| 20 | Files transfer **should** only start if the contact clicks on the link | SHOULD | This acts as a security feature to prevent the accidental download of malicious files | **T6.1.0:** Implement security features and protocols |
| 21 | Messages **should** be sent and received within 5-10 seconds | SHOULD | A performance requirement so that users do not spend too long waiting on a response from a contact | **T2.2.0:** Setup and Configure Client/Server using MQTT |
| 22 | Users **should** be logged off automatically after a specific amount of time | SHOULD | Prevents the system from being burdened by a user that is not active. This also doubles as a security feature to prevent an unsupervised account being compromised | **T6.2.0:** Implement security features and protocols |
| 23 | User pictures **could** be displayed in the channels | COULD | An alternative identification method to just using a user’s name | **T4.1.0:** Implement User Interface  **T7.1.1:** Develop user profile and control panel |
| 24 | Application **could** allow the exchange of files with contacts | COULD | Users may wish to share images or other files with another user, hence the need for file transfer capabilities between contacts | **T8.0.0:** Implement file transfer feature |
| 25 | Users **could** be able to send emoji’s | COULD | Implements an interactivity feature to the application, allowing users to express themselves better | **T10.1.0:** Consider the implementation of additional features |
| 26 | Messages **could** come with sent and read receipts | COULD | Allows a user to know who has read their message and who has not. This is especially useful for urgent or messages of high importance | **T10.0.0:** Consider the implementation of additional features |
| 27 | Application **could** display the full history of the conversation when a specific contact is selected | COULD | Conversation history allows a user to rely on the system for information they may have forgotten about, hence, allowing them to refer to their older conversations | **T2.3.0:** Setup and Configure Client/Server using MQTT |
| 28 | Offline messages **could** be stored in the client-side and transmitted to the target user(s) once they are online | COULD | A user may not have internet access but would still like to send a message, this method ensures that as soon as they are connected to the internet, they can send those pending messages | **T9.1.0:** Setup a local database |
| 29 | Application **could** run on both macOS and Windows-based desktops and laptops | COULD | Not all users use windows devices, hence, the need for an application that runs on other operating systems | **T10.2.0:** Consider the implementation of additional features |
| 30 | Application **could** have a light and dark mode | COULD | Users with visual impairments may require alternative application colour schemes to make the application usable | **T4.2.0:** Implement User Interface  **T7.2.0:** Develop user profile and control panel |
| 31 | Application **could** have text-size customisation | COULD | Users with visual impairments may require alternative text sizes to be able to read their chat messages | **T4.2.1:** Implement User Interface  **T7.2.1:** Develop user profile and control panel |
| 32 | The application **could** have several language options | COULD | International users may require an application with features written in a language they are familiar with to use it | **T4.3.0:** Implement User Interface  **T7.3.0:** Develop user profile and control panel |

# Risk Analysis & Mitigation Plan:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Number** | **Description of Risk** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Unclear or unrealistic requirements and scope | 2 | 5 | Ensure that all requirements are reviewed by all members of the team and are thoroughly discussed prior to being confirmed. In addition to this, actively seek out the support from experts (i.e., lecturers and industry professionals) to ensure that the project scope is attainable within the allocated development window |
| 2 | Insufficient knowledge and background research of messaging applications | 3 | 4 | Carry out an intensive research process prior to beginning the development process to ensure that all team members are well informed |
| 3 | Security breach due to passwords being compromised | 2 | 5 | Add password encryption and (if feasible) two-factor authentication. |
| 4 | Team member falls ill due to ongoing pandemic or is otherwise unable to support the team due to extenuating circumstances | 3 | 4 | Assign a task to two members, the first member being the primary member in charge of completing the task. In the case they are unable to, the task is then passed onto the secondary member to handle task completion. |
| 5 | Lost data due to technical failure | 2 | 4 | Make regular backups |
| 6 | Tasks go over allotted time | 2 | 3 | Give buffer for overrun time at end of project - work to a week before actual deadline |
| 7 | Team member overwrites an existing file's contents on accident | 3 | 1 | Regularly use version control software (i.e., Git - GitHub) so that the file contents can be easily reverted to an older version |
| 8 | Users struggle to use the application due to unintuitive user interface (UI) | 2 | 2 | Ensure that during the testing stages user feedback is gathered with regards to the usability of the UI |
| 9 | Major bug found in testing stage | 2 | 2 | Agile development allows for regular testing to prevent large scale bugs at the end of the project. |