**COMSATS University Islamabad,   
Abbottabad Campus**

**Project Proposal   
(SCOPE DOCUMENT)**

**for**

**ROOMY**  
Version 1.0

***By***

**Muhammad Saad Hussain CIIT/SP21-BSE-020/ATD**

**Muddassir Ali CIIT/SP21-BSE-016/ATD**

**Muhammad Haris CIIT/SP21-BSE-019/ATD**

***Supervisor*Sir Fuzel Jamil**

***Bachelor of Science in Software Engineering (2021-2024)***

|  |  |  |
| --- | --- | --- |
| **No.** | **Comment** | **Action** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**SCOPE DOCUMENT REVSION HISTORY**

**Supervisor Signature**

**Date:**

Table of Contents

[**Abstract** 5](#_Toc162133587)

[**Introduction** 5](#_Toc162133588)

[**Problem Statement** 6](#_Toc162133589)

[**Problem Solution for Proposed System** 7](#_Toc162133590)

[**Related System Analysis/Literature Review** 7](#_Toc162133591)

[**Advantages/Benefits of Proposed System** 8](#_Toc162133592)

[**Scope** 9](#_Toc162133593)

[**Modules** 9](#_Toc162133594)

[**Room Creation & Discovery:** 9](#_Toc162133595)

[**System Limitations/Constraints** 10](#_Toc162133596)

[Scalability: 10](#_Toc162133597)

[Real-time Audio Chat Challenges: 10](#_Toc162133598)

[Critical Mass 10](#_Toc162133599)

[Moderation Challenges: 10](#_Toc162133600)

[**Software Process Methodology** 11](#_Toc162133601)

[Agile Methodology: 11](#_Toc162133602)

[Focus on User Needs: 11](#_Toc162133603)

[Rapid Prototyping and Iteration: 11](#_Toc162133604)

[Flexibility and Adaptability: 11](#_Toc162133605)

[**Tools and Technologies** 11](#_Toc162133606)

[**Project Stakeholders and Roles** 12](#_Toc162133607)

[**Team Members Individual Tasks/Work Division** 13](#_Toc162133608)

[**Data Gathering Approach** 13](#_Toc162133609)

[Focused Interviews: 13](#_Toc162133610)

[Targeted Surveys: 13](#_Toc162133611)

[Competitor Analysis: 13](#_Toc162133612)

[**Concepts** 13](#_Toc162133613)

[**Gantt chart** 14](#_Toc162133614)

[**Mockups** 15](#_Toc162133615)

[**Conclusion** 17](#_Toc162133616)

[**References** 17](#_Toc162133617)

[**Plagiarism Report** 17](#_Toc162133618)

**Project Category: (**Select all the major domains of proposed project**)**

**B-**Web Application/Web Application based Information System

**Abstract**

Roomy is a web application designed to revolutionise online communication by fostering real-time interaction within themed chat rooms. Unlike traditional messaging platforms, Roomy focuses on temporary chat rooms cantered around specific interests or topics. This unique approach allows users to connect with like-minded individuals, engage in lively discussions (through text or audio conversations), and collaborate on shared passions in a fun and open environment.

Built on the robust MERN stack (MongoDB, Express.js, React.js, and Node.js).

Roomy empowers users create a profile with interests and a bio, can create and find chat rooms, can chat in real-time, can generate and schedule chat rooms.

**Introduction**

In today's fast-paced digital world, the way we connect, and exchange ideas is constantly evolving. While traditional messaging platforms offer a convenient way to stay in touch, they often lack the dynamism needed for in-depth discussions or spontaneous interactions. Roomy emerges to address this gap, offering a revolutionary approach to online communication.

Roomy is a dynamic web application designed to foster real-time conversations and collaboration within themed chat rooms. Unlike traditional messaging platforms, Roomy focuses on temporary chat rooms cantered around specific interests or topics. This unique approach allows users to:

* Connect with like-minded individuals who share their passions.
* Engage in lively discussions through text or audio conversations.
* Collaborate on shared interests in a fun and open environment.

Get ready to ignite discussions, discover new perspectives, and build meaningful connections – all within the innovative world of Roomy.

**Problem Statement**

**Q: What problem does your software solve?**

Roomy tackles the frustration of finding focused conversations on specific interests. Existing messaging apps keep you connected, but often lack the structure for in-depth discussions or quick connections around shared passions.

**Why are you developing this system?**

We're developing Roomy to bridge this gap. It offers a platform dedicated to real-time, interest-based interactions, fostering deeper connections, and igniting discussions among those who share specific passions.

Q: Does the same system already exist?

Similar messaging platforms exist, but Roomy sets itself apart with temporary, themed chat rooms. Unlike persistent connections, Roomy focuses on focused discussions around specific topics with these temporary spaces. It also prioritizes real-time engagement through features like text chat, audio conversations, and scheduling.

**If yes, how will a re-implementation aid your learning?**

Even with existing systems, re-implementing Roomy offers valuable learning. Building a platform with its unique features allows us to gain hands-on experience with the MERN stack for web development. We'll also learn user-centric design by implementing functionalities like room creation, search, and moderation tools. Finally, the project hones project management skills, teaching us to manage the development process from concept to launch.

**What skills do you expect to learn from this project?**

Through Roomy, we expect to develop a diverse skillset. We'll gain proficiency in the MERN stack, hone UI/UX design skills, and learn to manage projects efficiently. Importantly, we'll develop problem-solving skills by tackling technical challenges that arise during development. Roomy is not just about creating a communication tool, it's about equipping ourselves with valuable skills for the future.

**Problem Solution for Proposed System**

Roomy tackles the challenge of shallow online interactions by offering a space specifically designed for focused discussions on shared interests. Unlike traditional messaging apps, Roomy utilizes temporary, themed chat rooms. This allows users to:

**Dive right into conversations** centered around specific topics, eliminating the need to navigate broad chat histories.

**Connect with like-minded people instantly** who share their passion for the chosen topic.

**Engage in real-time discussions** through text chat or even live voice conversations, fostering a dynamic and immediate exchange of ideas.

**Related System Analysis/Literature Review**

While Roomy focuses on fostering real-time conversations around specific interests, several existing platforms offer communication features:

* **Slack:** Popular in workplaces, Slack provides text chat functionalities within persistent channels organized by teams or topics. However, its focus on work communication and limited discovery options makes it less suitable for casual, interest-based interactions.
* **Discord:** Primarily used by gamers, Discord offers voice and text chat functionalities within persistent servers organized by communities. Finding new communities based on fleeting interests can be challenging, and the lack of focus on temporary discussions limits the platform's applicability to broader interest exploration.
* **Telegram:** This messaging app allows group chats with persistent conversation history. While groups can cater to specific topics, the asynchronous nature and lack of features for real-time engagement hinder in-depth discussions.

**Roomy's Contribution:**

By analyzing the limitations of these existing systems, Roomy aims to bridge the gap. It offers a unique combination of:

* **Temporary, themed chat rooms:** Focused discussions on specific interests without the burden of persistent connections.
* **Real-time features:** Text chat, audio conversations, and scheduling enable immediate and dynamic interactions.
* **User-friendly discovery:** A search and discovery system facilitates finding rooms based on current interests.

Roomy goes beyond simply connecting users; it empowers them to engage in real-time, in-depth conversations around the topics they care about.

**Table 1 Related System Analysis with proposed project solution**

|  |  |  |
| --- | --- | --- |
| **Application Name** | **Weakness** | **Proposed Project Solution** |
| Slack | limited discovery options | Temporary, themed chat rooms, Real-time features, and User-friendly discovery |
| Discord | the lack of focus on temporary discussions | Temporary, themed chat rooms |
| Telegram | asynchronous nature and lack of features for real-time engagement | **Focus on Themed Discussions**  **Real-Time Engagement**  **Enhanced Discovery** |

**Advantages/Benefits of Proposed System**

* **Focused Discussions:**

Temporary, themed chat rooms eliminate the clutter of persistent chats and ensure discussions stay focused on the chosen topic.

* **Real-Time Engagement:**

Text chat, audio conversations, and scheduling features enable dynamic and immediate interaction for a more engaging experience.

* **Interest-Based Connections:**

The search and discovery system facilitate finding rooms based on current interests, allowing users to connect with like-minded people instantly.

* **Deeper Connections:**

By fostering focused discussions, Roomy encourages deeper connections and knowledge sharing among those who share specific passions.

* **Exploration and Learning:**

 The platform allows users to explore new interests by discovering and participating in diverse themed chat rooms.

**Scope**

Roomy, a web application for real-time, interest-based conversations, tackles limitations of existing platforms by offering temporary, themed chat rooms. Users can create and discover rooms based on their passions, engaging in dynamic text chat with features like threading and reactions.

Users can create Temporary groups, user can create audio chat rooms, can invite anyone in chat room, and anyone can request to join the chat rooms.

After the group chat

**Modules**

This describes a cool chat app where you can join or create groups on any topic you like. Chat with friends using text messages, emojis, and even voice chat in special rooms. You can even schedule group chats for later and customize your profile with your interests. To keep things friendly, there are tools for room creators to manage things and keep the chat respectful.

* **Register:** New users can sign up and create an account to access Roomy's features.
* **Login:** Existing users can securely log in to their accounts to engage in conversations.
* **Manage Profile:** Users have the flexibility to customize their profiles by adding interests, bios, and pictures.
* **Create Chat Rooms:** Users can initiate discussions by creating chat rooms based on various topics of interest.
* **Invite Members:** Users can extend invitations to others, encouraging participation in chat rooms.
* **Search Room:** Users can search their desired rooms with keywords and
* **Join Room:** Users can join existing chat rooms to participate in ongoing discussions.
* **Update Room:** Users can update the policies or the theme of the room.
* **Leave Room:** Users can leave chat rooms they are no longer interested in.
* **Log Out:** Users can securely log out of their Roomy accounts, ensuring privacy and account security.

**Fully dressed Use cases:**

**Registration:**

|  |  |
| --- | --- |
| **Use case ID** | UC -01 |
| **Use case Name** | Register |
| **Actors** | User |
| **Description** | User arrives at Roomy homepage and want to create an account, clicks the sign-up button, and creates an account to use Roomy’s features. |
| **Trigger** | The user wants to create a new account on Roomy. |
| **Preconditions** | The user has internet access.  The user has a valid email address. |
| **Postconditions** | The user a new Roomy account.  The user is logged into the application. |
| **Normal Flow** | The user arrives at the Roomy web application homepage.  Click on sign-up button.  The user enters the email and password.  The user clicks on the create account button.  The email address is validated.  The user chooses a username.  The user is logged into the system. |
| **Alternative flow**  Alternative flow 1 (invalid email) | The user has entered an invalid email.  The system asks the user to enter a valid email and repeat the whole process again. |
| **Alternative flow**  Alternative flow 2 (username already taken) | User enters a username which is taken by someone else.  The system asks the user to take another username.  The user takes another username.  The system |
| **Alternative flow**  Alternative flow 2  (Login with GitHub) | The user clicks the 'Login with GitHub' button.  The system redirects the user to GitHub for authentication.  The user enters their GitHub credentials.  GitHub authenticates the user and returns an authentication token to Roomy.  The system validates the token and creates an account for the user.  The user is logged into the system. |
| **Exceptions** | System overload: If the system is overloaded, an error message is displayed informing the user to try again later. |
| **Business Rules** | Passwords must meet specific complexity requirements (e.g., minimum length, combination of uppercase and lowercase letters, numbers, and symbols).  Usernames cannot contain special characters (except for allowed ones, e.g., underscore, period).  Email addresses must be unique. |
| **assumptions** | The user has a device capable of accessing the internet and running the Roomy web application.  The user understands basic internet navigation and form filling procedures. |

**Sign In:**

|  |  |
| --- | --- |
| **Use case ID** | UC -02 |
| **Use case Name** | Sign In |
| **Actors** | User |
| **Description** | User arrives at Roomy homepage and want to access account to use Roomy’s features, clicks the Sign-In button, and the user is granted access to its roomy account. |
| **Trigger** | The user desires to access their existing Roomy account. |
| **Preconditions** | The user has internet access.  The user has a valid email address. |
| **Postconditions** | The user is successfully logged into the Roomy application.  The user gains access to their account features and functionalities. |
| **Normal Flow** | The user arrives at the Roomy web application homepage.  The user clicks on the designated "Sign In" button.  The user enters their registered email address and password in the login form.  The user submits the login credentials by clicking the "Sign In" button.  The system verifies the entered email address and password against the stored account information.  Upon successful verification, the system logs the user into their Roomy account. The user is directed to their account dashboard or a designated homepage. |
| **Alternative flow**  Alternative flow 1 (invalid email) | The user entered their email address or password is incorrect, the user receives an error message prompting them to re-enter their credentials. They must repeat steps 3-5 until valid credentials are entered. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later |
| **Business Rules** | There will be only three attempts allowed for a Sign In.  For security reasons, user sessions can be configured to automatically expire after a period of inactivity. This would require users to log in again to maintain access. |
| **Assumptions** | The user has a device capable of accessing the internet and running the Roomy web application.  The user remembers their registered email address and password. |

**Manage Profile:**

|  |  |
| --- | --- |
| **Use case ID** | UC -03 |
| **Use case Name** | Manage Profile |
| **Actors** | User |
| **Description** | The user wants to manage their profile information within the Roomy platform. |
| **Trigger** | The user desires to personalize their profile or update their account information. |
| **Preconditions** | The user has a valid Roomy account.  The user is successfully logged into the application. |
| **Postconditions** | The user successfully modifies their profile information based on their actions.  The updated profile information is reflected in their account |
| **Normal Flow** | The user is logged into their Roomy account and has access to their profile settings.  The user identifies the profile section they wish to modify (e.g., edit bio, change profile picture, update interests).  The user edits their desired profile information within the designated section. This could involve entering text, uploading an image, or selecting options from a list.  The user confirms the modifications by clicking a "Save Changes" button or similar action.  The system processes the user's changes and updates their profile information accordingly. |
| **Alternative flow**  Alternative flow 1 (invalid Information) | If the user enters invalid information (e.g., exceeding character limits, inappropriate content), the system displays an error message specifying the issue. The user must correct the information and resubmit the changes. |
| **Alternative flow**  Alternative flow 2  (Image Upload Error) | If an error occurs during image upload (e.g., unsupported file format, exceeding file size limit), the system displays an error message. The user must choose a valid image and try again. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later |
| **Business Rules** | Bios and other text fields might have character limits to ensure conciseness and readability within the profile layout.  The system might enforce limitations on image file size and format (e.g., JPEG, PNG) to optimize storage and display. |
| **Assumptions** | The user understands basic internet navigation and form filling procedures.  The user has access to any necessary resources for profile updates (e.g., image files for profile picture). |

**Create Chat Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -04 |
| **Use case Name** | Create Chat Room |
| **Actors** | User |
| **Description** | This use case outlines the process for a user to initiate a new chat room on the Roomy platform for discussions on specific topics or themes, with options for text, audio, or video chat functionality. |
| **Trigger** | The user desires to create a dedicated space for focused conversations with other users around a particular interest, choosing the desired communication format |
| **Preconditions** | The user has a valid Roomy account.  The user is successfully logged into the application. |
| **Postconditions** | A new chat room is created on the Roomy platform with the user as the creator.  The user is automatically joined to the newly created chat room.  Depending on the chosen settings and the platform's capabilities, the chat room facilitates text-based conversations, audio calls, or video conferencing. |
| **Normal Flow** | The user navigates to the designated section for creating chat rooms (e.g., "Create Chat Room" button or menu option).  The user provides details to define the new chat room:  A descriptive name that reflects the chat room's central topic (e.g., "Book Club Discussions," "Movie Night Planning").  The user configures the chat room settings:  Choose the desired communication mode for the chat room: Text Chat, Audio Chat, or Video Chat. This selection determines the functionalities available within the room.  Choose whether the chat room is open to all users who can discover and join it, or if it requires an invitation for access.  If the room is private, the user can optionally set a password for an additional layer of control over who can join.  Room Creation: The user confirms the details and settings and initiates the creation process by clicking on a "Create Room" button or similar action.  The system successfully creates the new chat room based on the provided details and settings.  The user is automatically joined to the newly created chat room. |
| **Alternative flow**  Alternative flow 1 (Invalid Room Name) | If the user enters an invalid room name (e.g., exceeding character limit, containing inappropriate language), the system displays an error message prompting them to enter a valid name. The user must correct the name and resubmit. |
| **Alternative flow**  Alternative flow 2  (Room Name Already Taken) | If the chosen room name is already in use for another chat room, the system displays an error message suggesting alternative names. The user must choose a unique name and resubmit. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later. |
| **Business Rules** | Room names must adhere to specific guidelines regarding length and acceptable characters.  Inappropriate or offensive language might be restricted in room names.  The platform might have limitations on the number of rooms a single user can create simultaneously.  The availability of audio and video chat functionalities might depend on user device compatibility and system resources |
| **Assumptions** | The user understands the purpose of chat rooms on the Roomy platform.  The user has a clear idea of the topic or theme for their desired chat room and the preferred communication format (text, audio, or video). |

**Invite Members:**

|  |  |
| --- | --- |
| **Use case ID** | UC -05 |
| **Use case Name** | Invite Members |
| **Actors** | User |
| **Description** | This use case details the process for a user to invite other Roomy users to join a previously created chat room. |
| **Trigger** | The user, as the chat room creator or an existing member with permission, desires to invite specific users to participate in the conversation. |
| **Preconditions** | The user has a valid Roomy account.  The user is successfully logged into the application.  The user is a member of the chat room they wish to invite others to (applies if not the creator).  The chat room settings allow adding members via invitations (not applicable to public rooms).  The user has a valid Roomy account.  The user is successfully logged into the application. |
| **Postconditions** | The invited users receive a notification or message informing them about the invitation to join the chat room.  The invited users can choose to accept or decline the invitation.  Upon accepting, the invited users become members of the chat room and can participate in the conversation. |
| **Normal Flow** | The user accesses the desired chat room.  The user navigates to the designated section for inviting users (e.g., "Invite Members" button or menu option).  The user selects the specific users they want to invite from their contact list, search results, or by entering usernames.  The user confirms their selections and sends the invitations.  The system sends notifications or messages to the chosen users, informing them about the invitation and providing details about the chat room (optional). |
| **Alternative flow**  Alternative flow 1 (User not found) | f a user the inviter attempts to invite doesn't exist on Roomy, the system displays an error message prompting the user to verify the username. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later. |
| **Business Rules** | Users might have limitations on the number of invitations they can send simultaneously (to prevent spam).  Certain user roles might have restricted permissions to invite others (depending on room settings). |
| **Assumptions** | The user knows the usernames or has a way to identify the specific users they want to invite.  The user understands the chat room's settings regarding adding members via invitations. |

**Search Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -06 |
| **Use case Name** | Search Room |
| **Actors** | User |
| **Description** | This use case outlines the process for a user to search for and discover existing chat rooms on the Roomy platform. Users can find chat rooms based on specific topics, interests, or other criteria and join conversations that interest them |
| **Trigger** | The user desires to find and join a chat room related to a particular topic or interest. |
| **Preconditions** | The user has internet access.  The user has a valid Roomy account.  The user is successfully logged into the application. |
| **Postconditions** | The user successfully finds and joins a chat room of interest.  The user's participation in the selected chat room is facilitated by the system. |
| **Normal Flow** | The user logs into their Roomy account and navigates to the chat room search functionality.  The user enters search criteria (e.g., keywords, topics, chat room names) into the search bar.  The user submits the search query by clicking a "Search" button or similar action.  The system processes the search request and displays a list of chat rooms matching the search criteria.  The user reviews the search results and selects a desired chat room from the list.  The system provides details about the selected chat room (e.g., description, member count).  The user joins the selected chat room by clicking a "Join Room" button or similar action.  The system adds the user to the chat room and grants access to participate in the conversation. |
| **Alternative flow**  Alternative flow 1 (Confirmation Cancellation) | The user enters search criteria and submits the search query.  The system processes the search request but finds no chat rooms matching the criteria.  The system displays a message indicating no matching chat rooms were found and suggests alternative search terms or criteria. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later. |
| **Business Rules** | The system might limit the number of search results displayed at once to ensure performance.  Search functionality should account for variations in spelling and synonyms to improve search accuracy. |
| **Assumptions** | The user understands how to use the search functionality within the Roomy platform.  The user has a clear idea of the topics or themes they are interested in finding chat rooms for. |

**Join Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -07 |
| **Use case Name** | Join Room |
| **Actors** | User |
| **Description** | This use case details the process for a user to join an existing chat room on the Roomy platform, allowing them to participate in discussions and access room-specific features. |
| **Trigger** | The user finds a chat room they wish to join, either through search, invitation, or browsing. |
| **Preconditions** | The user has internet access.  The user has a valid Roomy account.  The user is successfully logged into the application.  The chat room exists and is accessible to the user based on its privacy settings. |
| **Postconditions** | The user becomes a member of the chat room.  The user can participate in conversations and access room-specific features. |
| **Normal Flow** | The user navigates to the desired chat room page.  The user clicks on the "Join Room" button or similar action.  If the room is public, the system immediately adds the user to the chat room.  If the room is private and requires a password, the user is prompted to enter the room password.  The user enters the password and submits it.  The system validates the password.  Upon successful validation, the system adds the user to the chat room.  The user gains access to the chat room and its features and can start participating in the conversation. |
| **Alternative flow**  Alternative flow 1 (Invalid Password) | The user attempts to join a private chat room by entering a password.  The system validates the password and finds it incorrect.  The system displays an error message prompting the user to re-enter the correct password.  The user re-enters the password and submits it again. |
| **Exceptions**  (Exception 1)  (System Overload) | If the system is overloaded, an error message is displayed, informing the user to try again later. |
| **Exceptions**  (Exception 2)  (Access Denied) | If the user attempts to join a room they are not permitted to join (e.g., due to restrictions or bans), an error message is displayed. |
| **Business Rules** | The system might limit the number of members in a chat room to ensure performance.  Passwords for private rooms must meet specific security requirements (e.g., minimum length). |
| **Assumptions** | The user understands the process of joining a chat room on the Roomy platform.  The user has the necessary permissions or information (e.g., password) to join private chat rooms. |

**Update Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -08 |
| **Use case Name** | Update Room |
| **Actors** | Room Manager |
| **Description** | This use case details the process for a Room Manager to update the details of an existing room on the Roomy platform. The Room Manager can modify various attributes of the room such as the room name, description, capacity, and features, ensuring that the room information remains current and relevant for users. |
| **Trigger** | The Room Manager decides to change or correct the details of an existing room. |
| **Preconditions** | The Room Manager has internet access.  The Room Manager has a valid Roomy account with the necessary permissions to manage room details.  The Room Manager is successfully logged into the application.  The room to be updated exists within the Roomy platform. |
| **Postconditions** | The room details are successfully updated in the system.  The updated details are reflected in any relevant search results, room listings, or other interfaces where room information is displayed. |
| **Normal Flow** | The Room Manager navigates to the room management section.  The Room Manager selects the room to be updated.  The System displays the current details of the selected room.  The Room Manager modifies the necessary room details (e.g., room name, description, capacity, features).  The Room Manager submits the updated room details.  The System validates the input to ensure all required fields are correctly filled and data is in the correct format.  The System updates the room details in the database.  The System confirms the successful update to the Room Manager.  The System reflects the updated room details in all relevant interfaces. |
| **Alternative flow**  Alternative flow 1 (Invalid Data Entry) | The System detects that some fields contain invalid data.  The System displays an error message indicating the specific issues.  The Room Manager corrects the invalid data.  Steps 5-9 are repeated. |
| **Exceptions**  (Exception 1)  (Database Error) | The System encounters an error while updating the database.  The System displays an error message indicating the failure.  The Room Manager decides to retry the update or cancel the operation.  If retrying, steps 5-9 are repeated. |
| **Business Rules** | The System should ensure that room names are unique within the same building or area.  The System should provide real-time validation for data entry fields.  The System should log the update operation with details of the Room Manager and the changes made. |
| **Assumptions** | The Room Manager has adequate training to use the system.  The network and system infrastructure are reliable and capable of handling update operations without significant delays. |

**Leave Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -09 |
| **Use case Name** | Leave Room |
| **Actors** | User |
| **Description** | This use case outlines the process for a user to leave an existing chat room on the Roomy platform, terminating their participation in that room. |
| **Trigger** | The user decides to leave a chat room they are currently a member of. |
| **Preconditions** | The user has internet access.  The user has a valid Roomy account.  The user is successfully logged into the application.  The user is currently a member of the chat room they wish to leave. |
| **Postconditions** | The user is no longer a member of the chat room.  The user no longer has access to the room's conversations and features. |
| **Normal Flow** | The user navigates to the chat room they wish to leave.  The user accesses the room settings or options menu.  The user clicks on the "Leave Room" button or similar action.  The system prompts the user for confirmation to prevent accidental departures.  The user confirms the decision to leave the room.  The system processes the request and removes the user from the chat room.  The user is redirected to the main chat room directory or their account dashboard. |
| **Alternative flow**  Alternative flow 1 (Confirmation Cancellation) | The user attempts to leave the chat room.  The system prompts the user for confirmation.  The user chooses to cancel the confirmation prompt.  The leave room process is aborted, and the user remains a member of the chat room |
| **Exceptions**  (Exception 1)  (System Overload) | If the system is overloaded, an error message is displayed, informing the user to try again later. |
| **Business Rules** | Users might be required to provide a reason for leaving the room, depending on the room's settings or policies.  Users should be notified if leaving the room will result in losing access to content or features of the room permanently. |
| **Assumptions** | The user understands the implications of leaving a chat room on the Roomy platform.  The user has navigated the Roomy platform sufficiently to locate the "Leave Room" option. |

**Log Out:**

|  |  |
| --- | --- |
| **Use case ID** | UC -10 |
| **Use case Name** | Log Out |
| **Actors** | User |
| **Description** | This use case outlines the process for a user to terminate their current session on the Roomy platform. Logging out ensures the user's account is no longer actively connected and accessible on the device. |
| **Trigger** | The user desires to end their current use of the Roomy platform and disconnect their account. This could be due to finishing their work, taking a break, or switching to another device. |
| **Preconditions** | The user's session is terminated on the device.  The user is no longer actively connected to the Roomy platform.  Any confidential information associated with the user's account is no longer readily accessible on the device. |
| **Postconditions** | The user is successfully logged into the Roomy application.  The user gains access to their account features and functionalities. |
| **Normal Flow** | The user accesses the account menu or profile settings within the Roomy application.  The user identifies a clear option labeled "Log Out" or similar terminology.  The user selects the "Log Out" option.  The system prompts the user for confirmation (optional) to prevent accidental logouts.  The user confirms the logout (if prompted).  The system successfully terminates the user's session and returns them to a login screen or non-account-specific area of the application. |
| **Alternative flow**  Alternative flow 1 (Confirmation Cancellation) | If the user chooses to cancel the confirmation prompt (if presented), the logout process is aborted, and the user remains logged in. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later. |
| **Business Rules** | None specifically for logging out, but security best practices regarding session management might apply. |
| **Assumptions** | The user understands the purpose of logging out and its impact on their current session. |

**System Limitations/Constraints**

### Scalability:

While the MERN stack is scalable, handling a very large user base with high concurrent connections could strain the system's resources. Optimizations and potential infrastructure upgrades might be necessary as Roomy grows.

Real-time Audio Chat Challenges:

Implementing a robust and high-quality audio chat feature requires addressing issues like latency, echo cancellation, and potential bandwidth limitations for users.

Critical Mass**:**

Roomy relies on a critical mass of users with diverse interests to create a vibrant and engaging platform. Encouraging user participation and room creation will be crucial.

### Moderation Challenges:

Effectively managing many rooms and ensuring a safe and respectful environment will require a robust moderation strategy, potentially involving human moderators.

**Software Process Methodology**

Here's a suitable software development methodology for Roomy and the reasoning behind it:

### Agile Methodology:

We plan to utilize the Agile methodology for developing Roomy. This iterative and user-centric approach allows for continuous development, testing, and feedback integration throughout the project lifecycle. This is ideal for Roomy because:

### Focus on User Needs:

Agile prioritizes user stories and feedback, enabling us to adapt Roomy's features and functionalities based on real-world user needs.

### Rapid Prototyping and Iteration:

Agile allows for the creation of functional prototypes early on, facilitating user testing and feedback loops to refine the application iteratively.

### Flexibility and Adaptability:

 The project can adapt to changing requirements or unforeseen challenges as we progress through development cycles.

**Tools and Technologies**

For the implementation of the Roomy platform, we have selected a robust set of tools and technologies to ensure efficient development, documentation, presentation, database management, and design. MS Visual Studio 2015 is our primary IDE for coding and debugging. MS Word 2015 is used for documentation, while MS PowerPoint 2015 facilitates presentations. MongoDB 2015 is our chosen DBMS for its flexibility and scalability. Draw.io is utilized for designing flowcharts and diagrams.

JavaScript 6.0 serves as the core programming language. React.js 18 is used for the front-end to build user interfaces, and Node.js 20 is employed for the back-end, providing a robust server-side runtime environment. This cohesive set of tools and technologies supports the comprehensive development and deployment of the Roomy platform.

**Tools and Technologies for Proposed Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools**  **And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| MS Visual Studio | 2015 | IDE |
| MS Word | 2015 | Documentation |
| MS Power Point | 2015 | Presentation |
| MongoDB | 2015 | DBMS |
| Star UML | 2018 | Designing |
| Draw.io | --- | Designing |
| **Technology** | **Version** | **Rationale** |
| JavaScript | 6.0 | Programming language |
| React.js | 18 | Front-End |
| Node.js | 20 | Back-End |
|  |  |  |

**Project Stakeholders and Roles**

Write down the project stakeholders and their roles.

**Table 3Project Stakeholders for Proposed Project**

|  |  |
| --- | --- |
| **Project Sponsor** | COMSATS University Islamabad, Abbottabad Campus |
| **Stakeholder** | Muhammad Saad Hussain  Muddasir Ali  Muhammad Haris  Supervisor: Sir Fuzel Jamil  Final year project committee |

**Team Members Individual Tasks/Work Division**

**Team Member Work Division for Proposed Project**

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Student Registration Number** | **Responsibility/ Modules** |
| Muhammad Saad Hussain  Muddassir Ali  Muhammad Haris | SP21-BSE-020  SP21-BSE-016  SP21-BSE-019 | Create Chat Rooms, Messaging Module, Join Room Create Groups  Invite Members, Manage Profile, Leave Room, Log Out  Registration, Sign-Up, Search room, Documentation |

**Data Gathering Approach**

The following approaches will be used for data collection for Roomy.

### Focused Interviews:

Exploring communication habits and ideal features for real-time interest chats.

### Targeted Surveys:

Gathering quantitative data on user demographics and desired functionalities.

### Competitor Analysis:

Identifying strengths, weaknesses, and opportunities in existing platforms.

**Concepts**

*Developing Roomy will involve learning real-time communication protocols like Web Sockets for seamless chat interactions. scalable web development with Node.js and cloud platforms, UI/UX design principles, and NoSQL databases with MongoDB and MQL.*

**Gantt chart**

Create the Grant Chart and provide estimated start and end dates of all proposed modules/tasks for each team member. Also identify the dependencies (which tasks cannot be started/completed, until the dependent task is completed). Gantt chart can be created using MS Project.

******

Figure 1Sample Gantt chart

**Mockups**

Insert minimum mockups (Usually 4-6 mockups) which show the major modules mentioned in the scope section of the document. Do not include mockups for Login, Signup, Forgot Password, Contact Us, About Us etc. If the project is a Web or a Smartphone Application, then include at-least three mockups from each part of the project. You can design mockup in any design tool for example pencil tool (<https://pencil.evolus.vn/>) or Balsamiq (<https://balsamiq.com/>)





**Conclusion**

Conclude this document. (Usually 4-5 sentences)

**References**

Mention the books, research papers, web links etc.

**Plagiarism Report**

Attach the Plagiarism report of your project scope document from library staff of turnitin tool (http://turnitin.com