**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**for**

**<Roomy Chat Rooms>**  
Version 1.0

***By***

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

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

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

***Supervisor*Fuzel Jamil**

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

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 5**

2.1 Product perspective 5

2.2 Operating environment 5

2.3 Design and implementation constraints 5

**3.** **Requirement identifying technique 6**

3.1 Use Case Diagram 6

3.2 Use Case Description 6

**4.** **Specific Requirements 8**

4.1 System feature X 8

**5.** **Quality attributes** Error! Bookmark not defined.

5.1 Usability 9

5.2 Performance 9

**6.** **External interface requirements 35**

6.1 User interfaces 35

6.2 Software interfaces 35

6.3 Hardware interfaces 35

6.4 Communications interfaces 35

**7.** **Project Gantt chart 35**

**8.** **References 10**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Fuzel Jamil>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

This Software Requirements Specification (SRS) document outlines the requirements for Roomy, a dynamic web application designed for real-time communication and collaboration within themed chat rooms. The SRS provides a comprehensive guide to understanding the functionalities, features, and operational aspects of the application. It is structured to ensure clear communication among stakeholders, including developers, designers, and project managers, and to provide a detailed reference for the development and implementation phases of the project. By following this document, the development team will be able to align their efforts with the specified requirements, ensuring that the final product meets user needs and expectations.

**Purpose**

The purpose of this document is to define the requirements for the development of Roomy, a web application focused on facilitating real-time communication through temporary, interest-based chat rooms. This document specifies the functional and non-functional requirements of the application, providing a detailed description of its features and capabilities. It serves as a blueprint for the development process, guiding the team in building a robust and user-friendly platform. the SRS aims to ensure that all stakeholders have a clear understanding of the project's goals and deliverables.

**Scope**

Roomy is a web application designed to facilitate real-time communication and collaboration within themed chat rooms, utilizing the MERN stack (MongoDB, Express.js, React.js, Node.js). The application allows users to create and discover chat rooms based on their interests. Key features of Roomy include real-time text messaging with enhancements such as message threading, emojis, and reactions; audio chat rooms for voice conversations, with options to join, leave, and mute audio channels; and video call functionality for face-to-face interactions within chat rooms. Additionally, Roomy offers room scheduling with a calendar interface to help users plan and join rooms at convenient times. Users can create customizable profiles to add interests, bio information, profile pictures, and set notification preferences and privacy settings. Furthermore, the application includes moderation tools for room creators and administrators to manage activities. Roomy aims to create an engaging platform where users can connect with others who share similar interests, enabling lively and interactive discussions.

**Overall description**

**Product perspective**

We designed roomy is as an innovative communication platform, similar to existing applications like Discord, but with a unique feature: temporary chat rooms. These rooms can be automatically deleted after a specified time, making Roomy stand out in the market. While leveraging the strengths of established platforms. By combining real-time text messaging, audio and video chat capabilities, and room scheduling, Roomy aims to offer for users to connect and collaborate based on their interests.

**Operating environment**

Our system Roomy is a responsive web application designed to run seamlessly across various web browsers and devices. The software will operate correctly on popular web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari, ensuring compatibility with the latest and several previous versions of these browsers. The application is built to be fully responsive, providing an optimal user experience on desktop computers, laptops, tablets, and smartphones. Roomy will be accessible to users worldwide, with servers and databases hosted on reliable cloud platforms to ensure high availability and performance.

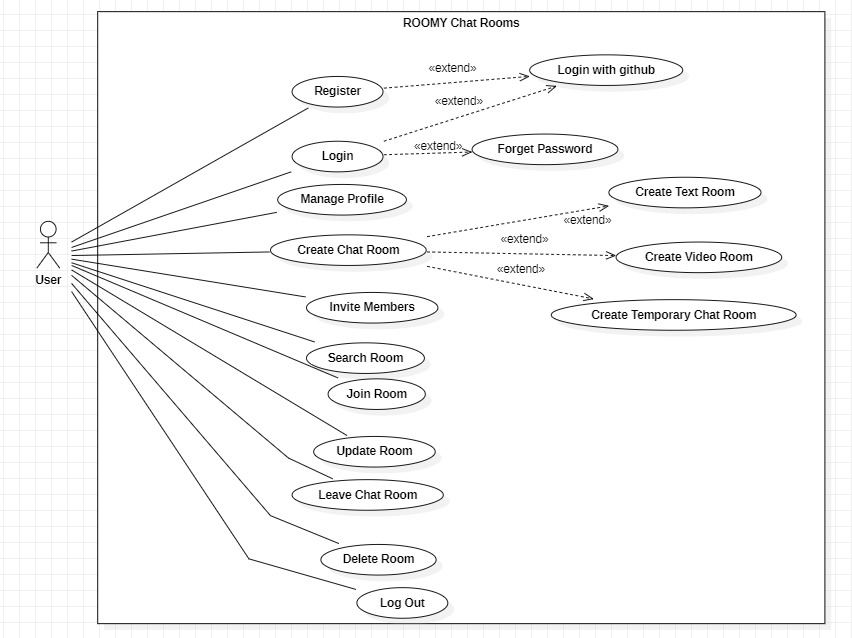
**Design and implementation constraints.**

Our application Roomy will be developed using the MERN stack, which consists of MongoDB, Express.js, React.js, and Node.js. This choice imposes certain constraints on the development process:

* CO-1: The system shall use MongoDB as the database engine. This NoSQL database is chosen for its flexibility in handling unstructured data and its scalability, which is crucial for managing the dynamic chat rooms.
* CO-2: The system shall use Express.js for backend API development. Express.js is selected for its simplicity and efficiency in building robust server-side applications, making it suitable for handling real-time communication features.
* CO-3: The system shall use React.js for the frontend user interface. React.js is chosen for its ability to create a dynamic and responsive user experience, essential for a web application that needs to handle real-time updates and user interactions smoothly.
* CO-4: The system shall use Node.js for server-side operations. Node.js is selected for its event-driven architecture, which is ideal for applications requiring real-time communication and high concurrency.
* CO-5: The system shall operate within the constraints of web technologies and standards. This includes ensuring compatibility with HTML5, CSS3, and JavaScript ES6+, and adhering to web accessibility standards to ensure the application is usable by as many people as possible.

**Requirement identifying technique.**

**Use case diagram**

****

Use case description.

**1. Register**

Description: New users can sign up and create an account to access Roomy's features. Upon registration, users provide necessary information such as email address and password and confirm password to create their account.

**2. Login**

Description: Existing users can securely log in to their accounts to engage in conversations. Users provide their login credentials (email address and password) to access their account and utilize Roomy's features.

**3. Manage Profile**

Description: Users have the flexibility to customize their profiles by adding interests, bios, and pictures. They can edit their profile information at any time to personalize their profiles and enhance their online presence.

**4. Create Chat Rooms**

Description: Users can initiate discussions by creating chat rooms based on various topics of interest. They provide a room name.

**5. Forget password**

Description: our application provide the functionality of forget password user can change their passwords.

**5. Invite People**

Description: Users can extend invitations to others, encouraging participation in chat rooms. They can invite others via email or shareable links, promoting user engagement and increasing participation in discussions.

**6. Search rooms**

Description: we provide the functionality of searching user can search room by the names.

**7. Create Audio and Video Chat Rooms**

Description: Registered users can create audio and video chat rooms for real-time voice and video conversations. They have the option to initiate audio or video discussions within Roomy, providing additional communication options beyond text-based chat rooms.

**8. Create Temporary Chat Rooms**

Description: Users can quickly establish temporary chat rooms for specific, time-bound discussions. These chat rooms are automatically deleted after a specified time period, allowing users to facilitate ad-hoc discussions and events within Roomy.

**9. join Rooms**

Description: register user can join the rooms by links through invitation.

**10. Schedule a Room**

Description: Users can plan by scheduling chat rooms for future discussions, ensuring timely engagements. They have the option to set specific dates and times for scheduled chat rooms, facilitating planning and coordination of discussions among users.

**11. Delete room**

Description: User or group admin has. The authority to delete a chat room.

**12. Update room**

Description: User or group admin has. The authority to update the room name.

**13. Log Out**

Description: Users can securely log out of their Roomy accounts, ensuring privacy and account security. Logging out clears all user session data and requires reauthentication for access, providing users with control over their account access and ensuring the security of their personal information within Roomy.

**Fully dressed Use cases:**

**Register:**

|  |  |
| --- | --- |
| **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 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 gave another username. |
| **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. User can click with GitHub user can easily login. |
| **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 Rooms:**

|  |  |
| --- | --- |
| **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. |

**Schedule Chat Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -08 |
| **Use case Name** | Schedule Chat Room |
| **Actors** | User |
| **Description** | This use case details the process for a user to create a chat room on the Roomy platform for a future date and time, allowing participants to plan their availability and join the discussion at the designated time. |
| **Trigger** | he user desires to create a dedicated space for focused conversations with other users at a specific time in the future, providing advanced notice and promoting participation. |
| **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 chat room is scheduled for a specific date and time in the future.  Depending on 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:  Descriptive name reflecting the chat room's central topic (e.g., "Weekly Team Meeting").  The user configures chat room settings and chooses the desired communication mode for the chat room: Text Chat, Audio Chat, or Video Chat (if applicable).  The user selects a date and time from a calendar interface for when the chat room should become active and accessible to participants.  The user optionally adds a description or agenda for the scheduled chat room, providing context and details to potential participants.  The user confirms details and settings and initiates the creation process by clicking on a "Schedule Room" button or similar action.  The system successfully creates the new scheduled chat room based on the provided details and settings.  The system automatically generates notifications or invites (optional) to inform relevant users about the upcoming chat room, including the date, time, and topic. |
| **Alternative flow**  **Alternative flow 1 (Schedule Conflict)** | If the chosen date and time conflict with existing limitations on the platform (e.g., exceeding maximum scheduling range), the system displays an error message prompting the user to select a valid time. |
| **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** | There might be limitations on the maximum scheduling range for chat rooms (e.g., cannot be scheduled months in advance).  Room names must adhere to specific guidelines regarding length and acceptable characters. |
| **Assumptions** | The user understands the concept of temporary chat rooms and their limited lifespan.  The user has a clear idea about the topic of their desired chat room, the preferred communication format (if applicable), and the intended duration for the conversation. |

**Create Groups (open / private:)**

|  |  |
| --- | --- |
| **Use case ID** | UC -09 |
| **Use case Name** | Create Groups (open / private) |
| **Actors** | User |
| **Description** | This use case outlines the process for a user to create a group on the Roomy platform, allowing them to manage a collection of users for focused communication and collaboration. The user can choose between an open group (discoverable and joinable by all users) or a private group (requiring approval for membership). |
| **Trigger** | The user desires to establish a persistent communication channel for a specific community of users around a shared interest, project, or team. |
| **Preconditions** | The user has a valid Roomy account.  The user is successfully logged into the application. |
| **Postconditions** | A new group is created on the Roomy platform with the user as the group owner.  The group settings reflect the chosen privacy level (open or private).  Depending on the platform's capabilities, the group can facilitate various communication features like chat rooms, discussion boards, file sharing, and member directories. |
| **Normal Flow**  **(Open group)** | The user navigates to the designated section for creating groups (e.g., "Create Group" button or menu option).  The user provides details to define the new group:  Descriptive name reflecting the group's purpose (e.g., "Book Lovers Club").  Optional group description providing further context and attracting potential members.  Selection of the "Open Group" privacy setting, making the group discoverable and allowing any user to join.  The user confirms details and settings and initiates the creation process by clicking on a "Create Group" button or similar action.  The system successfully creates the new open group based on the provided details and settings.  The group becomes visible to other users in search results or browsing categories (depending on the platform). |
| **Normal Flow (Private Group)** | The user follows steps 1 and 2 from the Open Group flow.  The user selects the "Private Group" privacy setting, requiring approval for membership.  The user optionally defines the approval process (e.g., manual approval by group owner or moderators, or question-based screening).  The user confirms details and settings and initiates the creation process.  The system successfully creates the new private group. |
| **Alternative flow**  **Alternative flow 1 (Invalid Group Name)** | If the user enters an invalid group name (e.g., exceeding character limit, containing inappropriate language), the system displays an error message prompting them to enter a valid name. |
| **Alternative flow**  **Alternative flow 2**  **(Group Name Already Taken)** | If the chosen group name is already in use for another group, the system displays an error message suggesting alternative names. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later. |
| **Business Rules** | Group names must adhere to specific guidelines regarding length and acceptable characters.  Inappropriate or offensive language might be restricted in group names and descriptions.  The platform might have limitations on the number of groups a single user can create. |
| **Assumptions** | The user understands the differences between open and private groups and chooses the appropriate privacy level for their needs.  The user has a clear idea about the purpose of the group and the target audience. |

**Update Room:**

|  |  |
| --- | --- |
| **Use case ID** | UC -010 |
| **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 -011 |
| **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. |

**Delete chat Rooms:**

|  |  |
| --- | --- |
| **Use case ID** | UC -012 |
| **Use case Name** | Delete chatrooms |
| **Actors** | Room manager |
| **Description** | This use case details the process for a group owner to permanently remove a group from the Roomy platform. |
| **Trigger** | The group owner determines that the group is no longer required or active and wishes to permanently remove it. |
| **Preconditions** | The user has a valid Roomy account.  The user is successfully logged into the application.  The user is the owner of the group. |
| **Postconditions** | The chosen group is permanently deleted from the Roomy platform.  All associated data, including chat history, member lists, and files, are removed.  Group members are no longer able to access the group or its content. |
| **Normal Flow** | The group owner accesses the group settings or management console.  The group owner chooses to confirm the dismantlement action.  The system permanently deletes the group and its data from the platform. |
| **Alternative flow**  **Alternative flow 1 (Confirmation Cancellation)** | If the group owner chooses to cancel the confirmation prompt, the dismantlement process is aborted, and the group remains intact. |
| **Exceptions** | In case of system overload, an error message is displayed, informing the user to try again later.  If the user attempting to dismantle the group is not the owner, the system displays an error message indicating insufficient permissions. |
| **Business Rules** | Dismantling a group is a permanent action and cannot be undone.  Group owners should exercise caution before dismantling a group, as data recovery might not be possible. |
| **Assumptions** | The group owner has carefully considered the decision to dismantle the group and understands the consequences.  The group owner has communicated the dismantlement to relevant members if necessary (depending on the situation). |

**Log Out:**

|  |  |
| --- | --- |
| **Use case ID** | UC -13 |
| **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. |

**Functional Requirements**

* **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.
* **Create audio and video chat Rooms:** register user can create a audio and video rooms.
* **Create Temporary Chat Rooms:** Users can quickly establish temporary chat rooms for specific, time-bound discussions.
* **Invite People:** Users can extend invitations to others, encouraging participation in chat rooms.
* **Search Room:** Users can search their desired rooms with the names
* **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
* **Schedule a Room:** Users can plan by scheduling chat rooms for future discussions, ensuring timely engagements.
* **Delete Chat Room:** Group creators have the authority to delete groups once discussions are concluded or no longer relevant.
* **Log Out:** Users can securely log out of their Roomy accounts, ensuring privacy and account security.

**Functional Requirement X**

**Functional Requirement 1 (User Registration)**

**Table 2 Show the functional requirement template**

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | User Registration |
| **Requirement** | The system shall allow new users to sign up and create an account to access Roomy's features. |
| **Source** | User requirement |
| **Rationale** | To allow new users to access the application and engage in conversations. |
| **Business Rule (if required)** | Users must provide a valid email address and create a password meeting security criteria. |
| **Dependencies** | None |
| **Priority** | High |

**Functional Requirement 2(User Login)**

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | User Login |
| **Requirement** | he system shall securely authenticate existing users to log in to their accounts and access Roomy's features |
| **Source** | User requirement |
| **Rationale** | To ensure secure access for existing users to engage in conversations. |
| **Business Rule (if required)** | Users must provide valid login credentials (email address and password) to access their accounts. |
| **Dependencies** | FR-1 (User Registration) - User login depends on the existence of registered user |
| **Priority** | High |

**Functional Requirement 3 (Manage Profile)**

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | User Login |
| **Requirement** | The system shall allow users to customize their profiles by adding interests, bios, and profile pictures. |
| **Source** | User requirement |
| **Rationale** | To provide users with the flexibility to personalize their profiles and enhance their online presence. |
| **Business Rule (if required)** | Users can edit their profile information at any time. |
| **Dependencies** | FR-1 (User Registration) and FR-2 (user login) – profile management is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 4 (Create Chat Rooms)**

|  |  |
| --- | --- |
| **Identifier** | FR-4 |
| **Title** | Create Chat Rooms |
| **Requirement** | The system shall enable users to initiate discussions by creating chat rooms based on various topics of interest. |
| **Source** | User requirement |
| **Rationale** | To facilitate user-driven conversations and community building within the application. |
| **Business Rule (if required)** | Users can set visibility and privacy settings for each chat room they create. |
| **Dependencies** | FR-1 (User Registration) and FR-2 (user login) – create chat rooms is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 5 (Create Audio and Video Chat Rooms)**

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | Create Audio and Video Chat Rooms |
| **Requirement** | Registered users can create audio and video chat rooms for real-time voice and video conversations. |
| **Source** | User requirement |
| **Rationale** | To provide users with additional communication options beyond text-based chat rooms. |
| **Business Rule (if required)** | Users must have access to microphone and camera permissions to create audio and video chat rooms. |
| **Dependencies** | FR-1 (User Registration) and FR-2 (user login) – Create Audio and Video Chat Rooms) is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 6 (Create Temporary Chat Rooms)**

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | Create Temporary Chat Rooms |
| **Requirement** | Users can quickly establish temporary chat rooms for specific, time-bound discussions. Source: User requirement |
| **Source** | User requirement |
| **Rationale** | To facilitate ad-hoc discussions and events within the application. |
| **Business Rule (if required)** | Temporary chat rooms will be automatically deleted after a specified time period. |
| **Dependencies** | FR-1 (User Registration) and FR-2 (user login) – Create Temporary Chat ) is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 7 (Invite People)**

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | Invite People |
| **Requirement** | Users can quickly establish temporary chat rooms for specific, time-bound discussions. Source: User requirement |
| **Source** | User requirement |
| **Rationale** | To promote user engagement and increase participation in discussions. |
| **Business Rule (if required)** | Users can invite others via email or shareable links |
| **Dependencies** | FR-2 (user login) FR-4 (Create Chat Rooms ) – invite people depends on user login and create chat rooms requirements |
| **Priority** | medium |

**Functional Requirement 8 (Schedule a Room)**

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Schedule a Room |
| **Requirement** | Users can plan by scheduling chat rooms for future discussions, ensuring timely engagements. |
| **Source** | User requirement |
| **Rationale** | To facilitate planning and coordination of discussions among users. |
| **Business Rule (if required)** | Users can set specific dates and times for scheduled chat rooms. |
| **Dependencies** | FR-2 (user login) – schedule a room is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 9 (Dismantle Group)**

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Dismantle Group |
| **Requirement** | Group creators have the authority to disband groups once discussions are concluded or no longer relevant. |
| **Source** | User requirement |
| **Rationale** | To facilitate planning and coordination of discussions among users. |
| **Business Rule (if required)** | Only the creator of a chat room has the authority to dismantle it. |
| **Dependencies** | FR-1 (User Registration) and FR-2 (user login) – Dismantle Group is depends on the existence of registered user and sign in |
| **Priority** | medium |

**Functional Requirement 10 (Log Out**)

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | Log Out |
| **Requirement** | Users can securely log out of their Roomy accounts, ensuring privacy and account security. |
| **Source** | User requirement |
| **Rationale** | To provide users with control over their account access and ensure security. |
| **Business Rule (if required)** | Logging out clears all user session data and requires reauthentication for access. |
| **Dependencies** | FR-2 (user login) – logout is depends on the existence of registered user and sign in |
| **Priority** | High |

**Non Functional Requirements**

**Usability**

**Simple interface:** The layout and navigation of Roomy should be easy for users of all technical backgrounds to understand. Users should be able to find what they need and perform actions quickly and efficiently.

**Clear error messages:** Roomy should provide clear instructions whenever a user needs to perform an action. Error messages should be informative and guide users towards resolving the issue.

**Performance**

**Fast page load times:** Roomy webpages should load quickly (e.g., within 3 seconds) to ensure a smooth user experience. This is especially important for real-time features like chat and audio conversations.

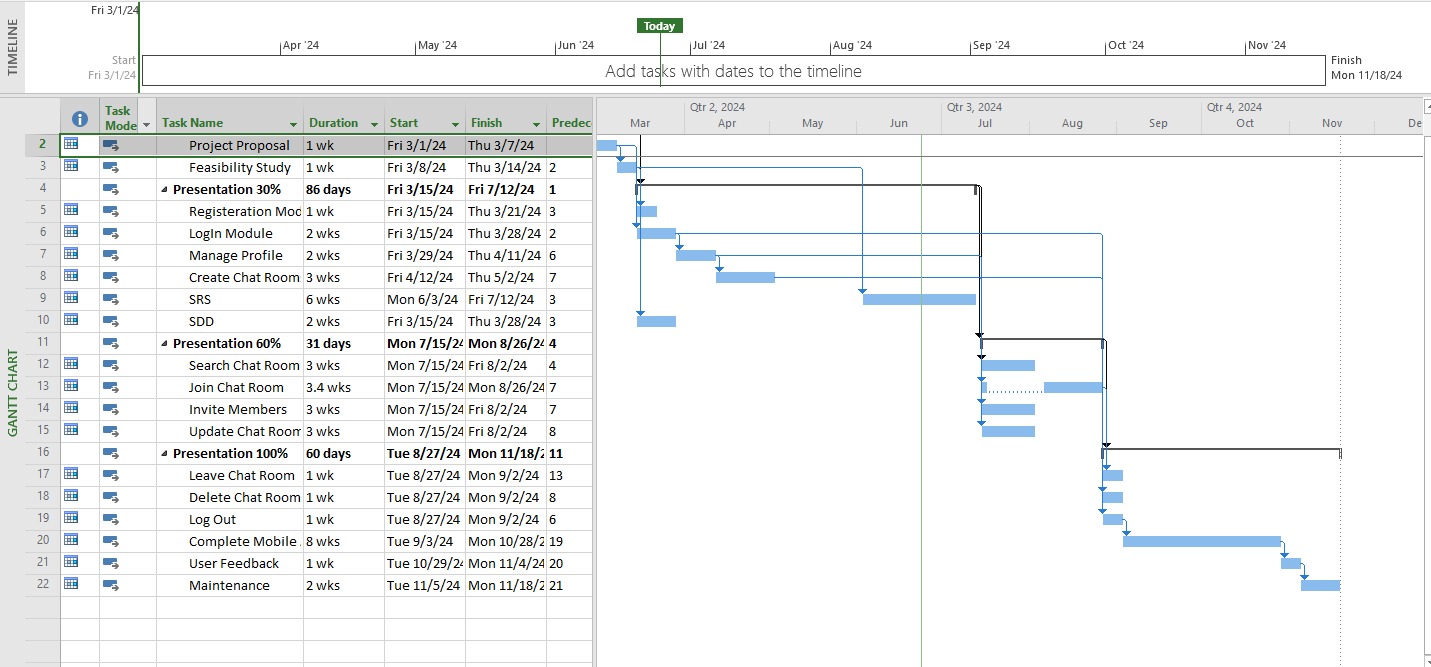
**Responsive design:** Roomy should adapt its layout seamlessly across different devices (desktop, mobile, tablet) for optimal user experience on any platform.

**Scalability**: Roomy should be able to handle a growing number of users and chat rooms without compromising performance.

**External interface requirements**



**Project Gantt chart**:



**References**