**COMSATS University Islamabad,   
Abbottabad Campus**

**SOFTWARE DESIGN DESCRIPTION   
(SDD 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*Fuzzel Jamil**

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

**Table of Contents**

Table of Contents

[Introduction 5](#_Toc170195858)

[Design methodology and software process model 5](#_Toc170195859)

[System overview 5](#_Toc170195860)

[Architectural design 6](#_Toc170195861)

[Design models 7](#_Toc170195862)

[Data Flow Diagrams: 7](#_Toc170195863)

[SSD 10](#_Toc170195864)

[Activity Diagram: 19](#_Toc170195865)

[Data Design 20](#_Toc170195866)

[Major Data Entities 20](#_Toc170195867)

[Databases and Data Storage 20](#_Toc170195868)

[Human interface design 24](#_Toc170195869)

**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

Roomy, a web application designed to bridge the gap in traditional messaging by fostering dynamic discussions through real-time communication and collaboration within themed chat rooms. This SDD serves as a comprehensive technical blueprint, outlining functionalities, features, and system architecture. By following this document, the development team will ensure alignment with project requirements, ultimately delivering a user-centric product that meets the evolving needs of online interaction.

# Design methodology and software process model

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 of 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.

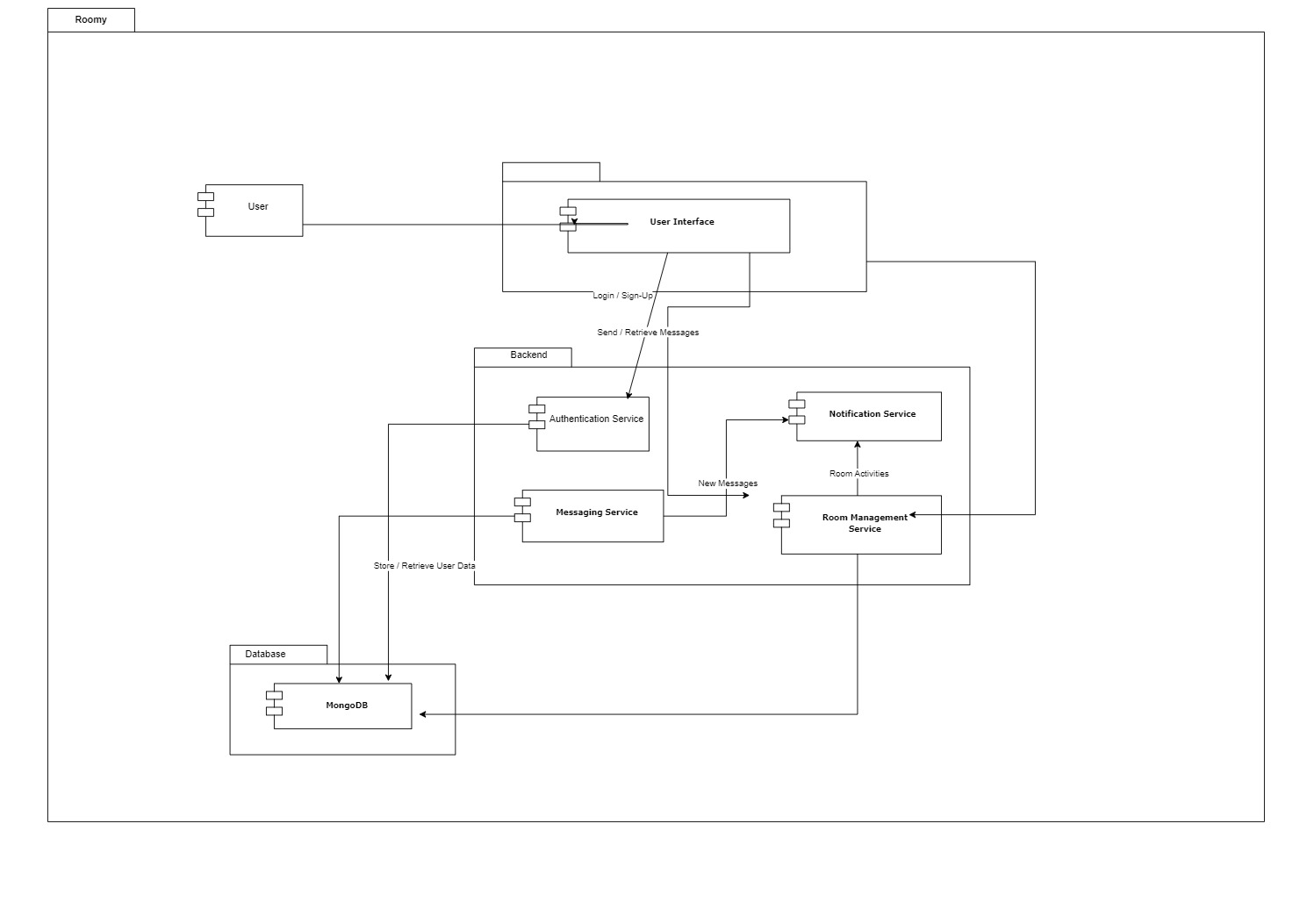
**Flexibility and Adaptability:**

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

# System overview

Roomy is a comprehensive online platform designed to facilitate diverse forms of communication and collaboration through chat rooms. The primary goal of Roomy is to provide users with a versatile and engaging environment where they can create, join, and manage chat rooms for various purposes, ranging from casual conversations to professional meetings.

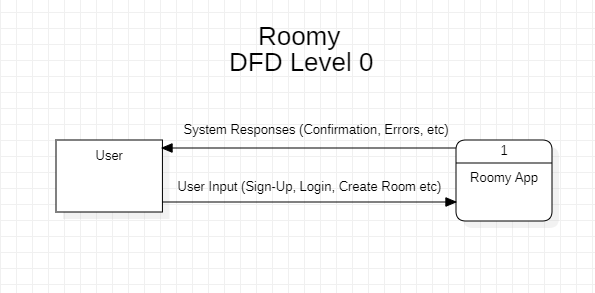
# Architectural design



# Design models

## Data Flow Diagrams:

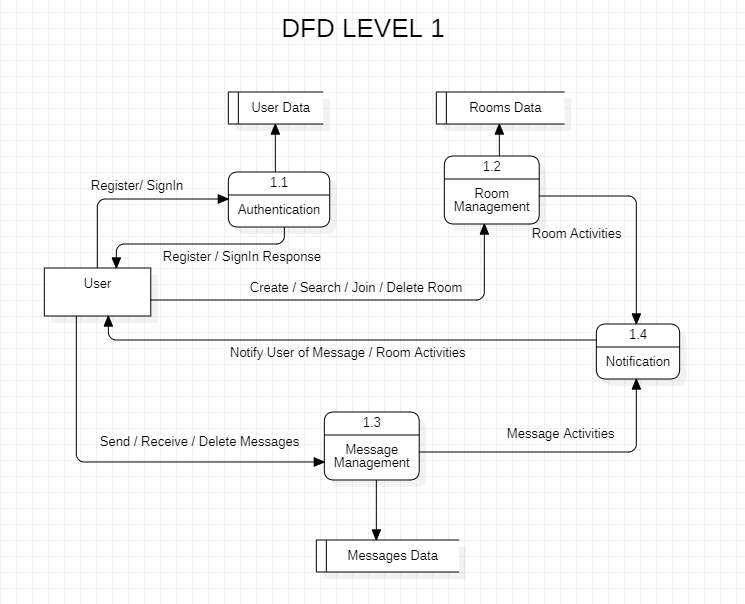
**DFD LEVEL 0:**

****

**Description:**

The Level 0 DFD provides an overview of Roomy as a single process with its external entities. It shows the interactions between users and the Roomy system, highlighting the main input and output data flows. Users interact with Roomy to register, log in, manage profiles, create and join chat rooms, invite others, search for rooms, and log out. The system processes these interactions and provides responses accordingly.

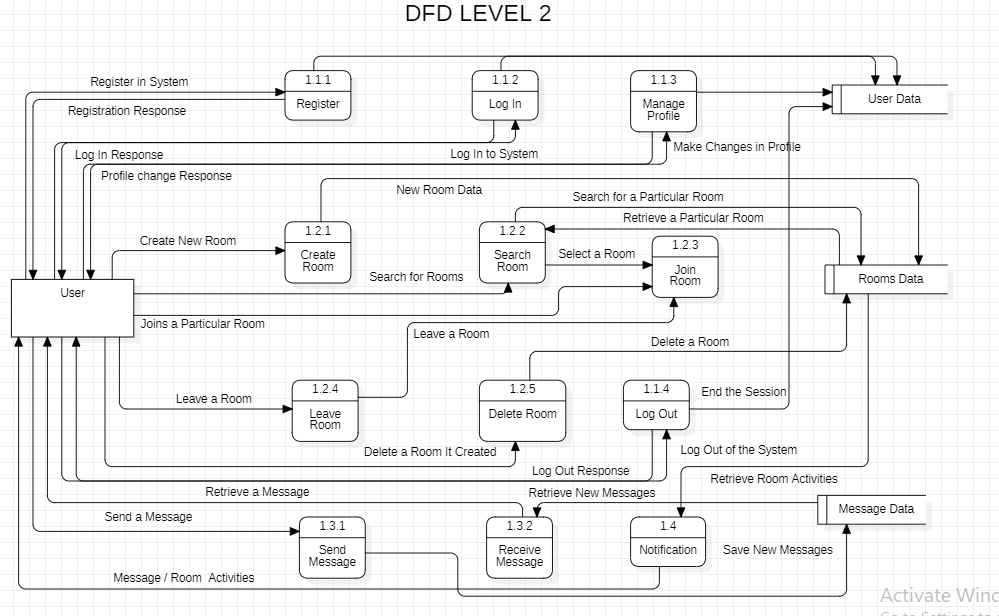
**DFD Level 1:**

****

**Description:**

The Level 1 DFD breaks down the main process into its sub-processes, providing more detail on how Roomy handles user interactions. Key sub-processes include user authentication, profile management, chat room management, communication handling (text/video), invitation and search management, and security processes. Each sub-process shows its inputs, outputs, and data stores involved.

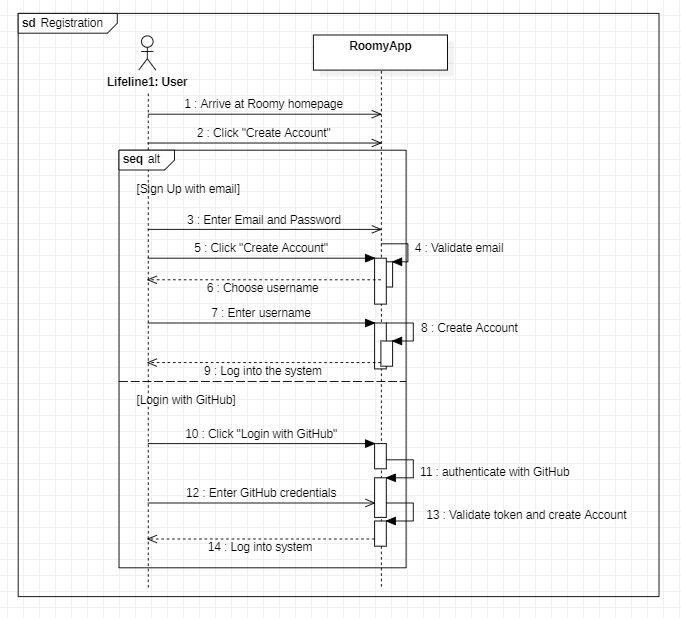
**DFD Level 2:**

****

**Description:**

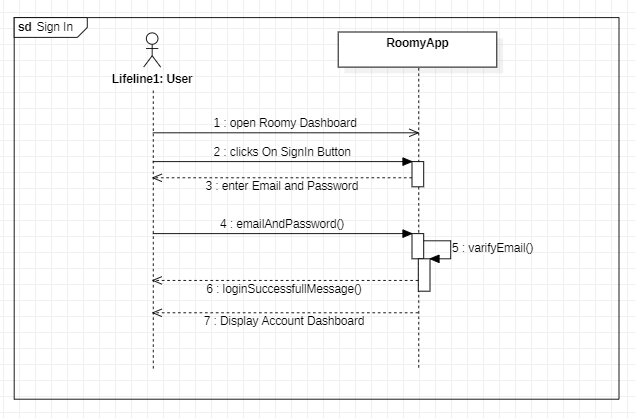
The Level 2 DFD provides an even more detailed view of the sub-processes identified in Level 1. Each sub-process is further broken down to show specific actions and data flows involved in completing the tasks.

## SSD

****

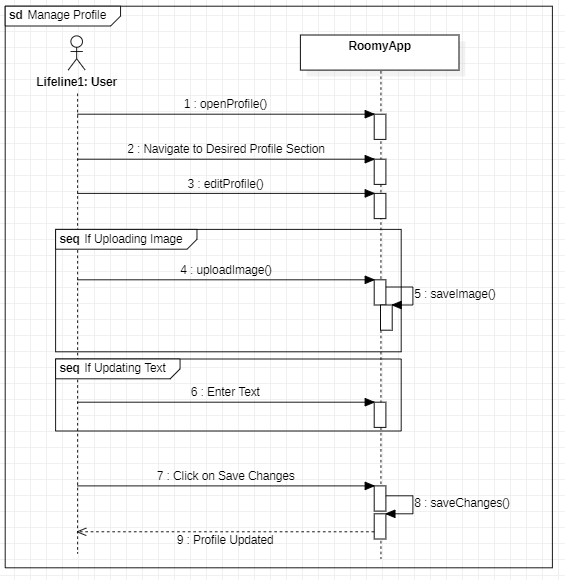
**Description:**

This SSD illustrates the process by which a new user signs up for Roomy by providing necessary information such as an email address and password, which the system then validates before creating a new user account and sending a confirmation response.

****

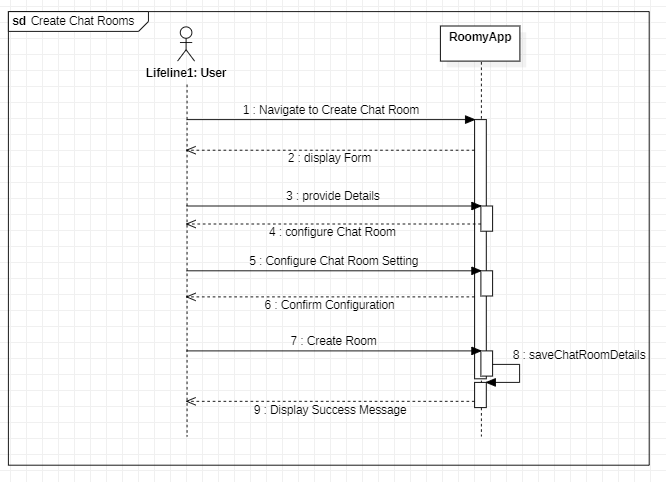
**Description:**

This SSD outlines how an existing user logs into their Roomy account by entering their email address and password, with the system validating the credentials and granting access upon successful validation.

****

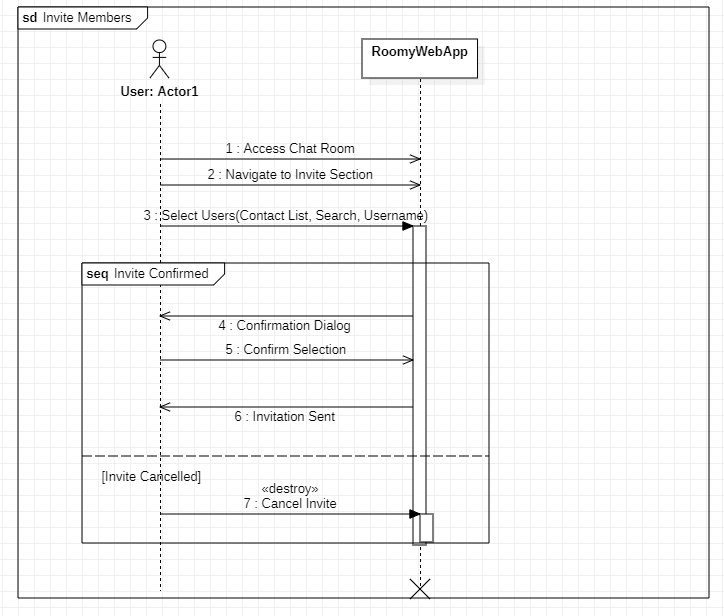
**Description:**

This SSD details the steps a user takes to customize their profile by updating their interests, bio, and profile picture, which the system then validates and saves before confirming the update to the user.

****

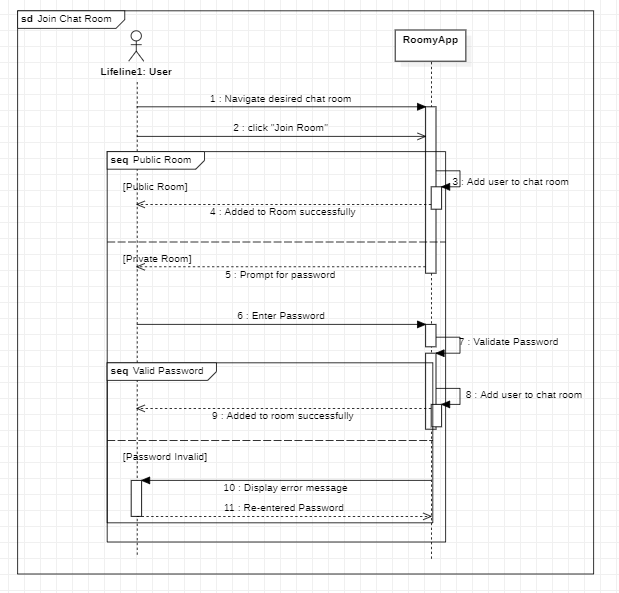
**Description:**

This SSD shows how a user creates a new chat room by providing a room name, which the system then uses to generate the chat room and notify the user of its successful creation.

****

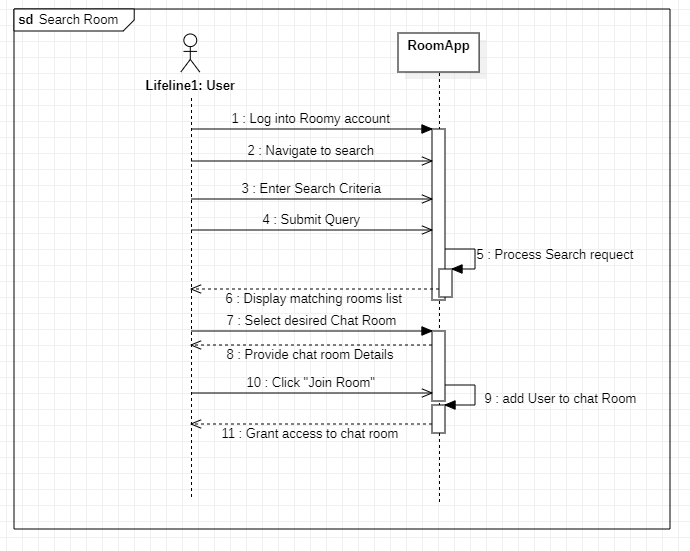
**Description:**

This SSD depicts how a user invites other members to a chat room by generating an invitation link or sending an email invite, which the system processes and sends to the invited users.

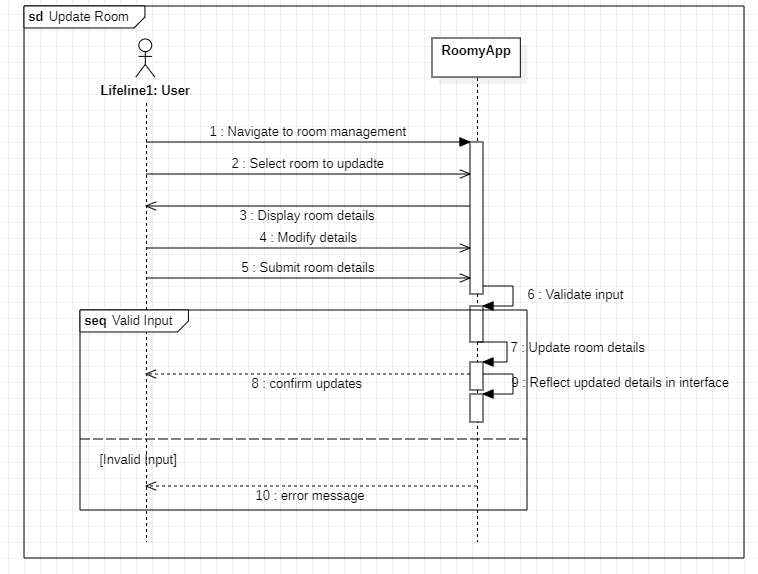
****

**Description:**

This SSD illustrates how a user joins a chat room via an invitation link, with the system validating the link and granting access to the chat room.

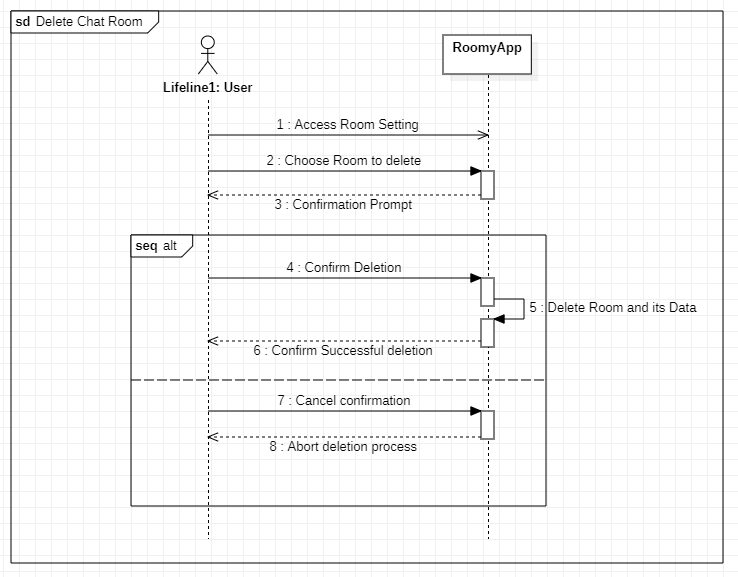
****

**Description:** This SSD outlines the process a user follows to search for chat rooms by name, with the system retrieving and displaying relevant chat rooms based on the search query.

****

**Description:**

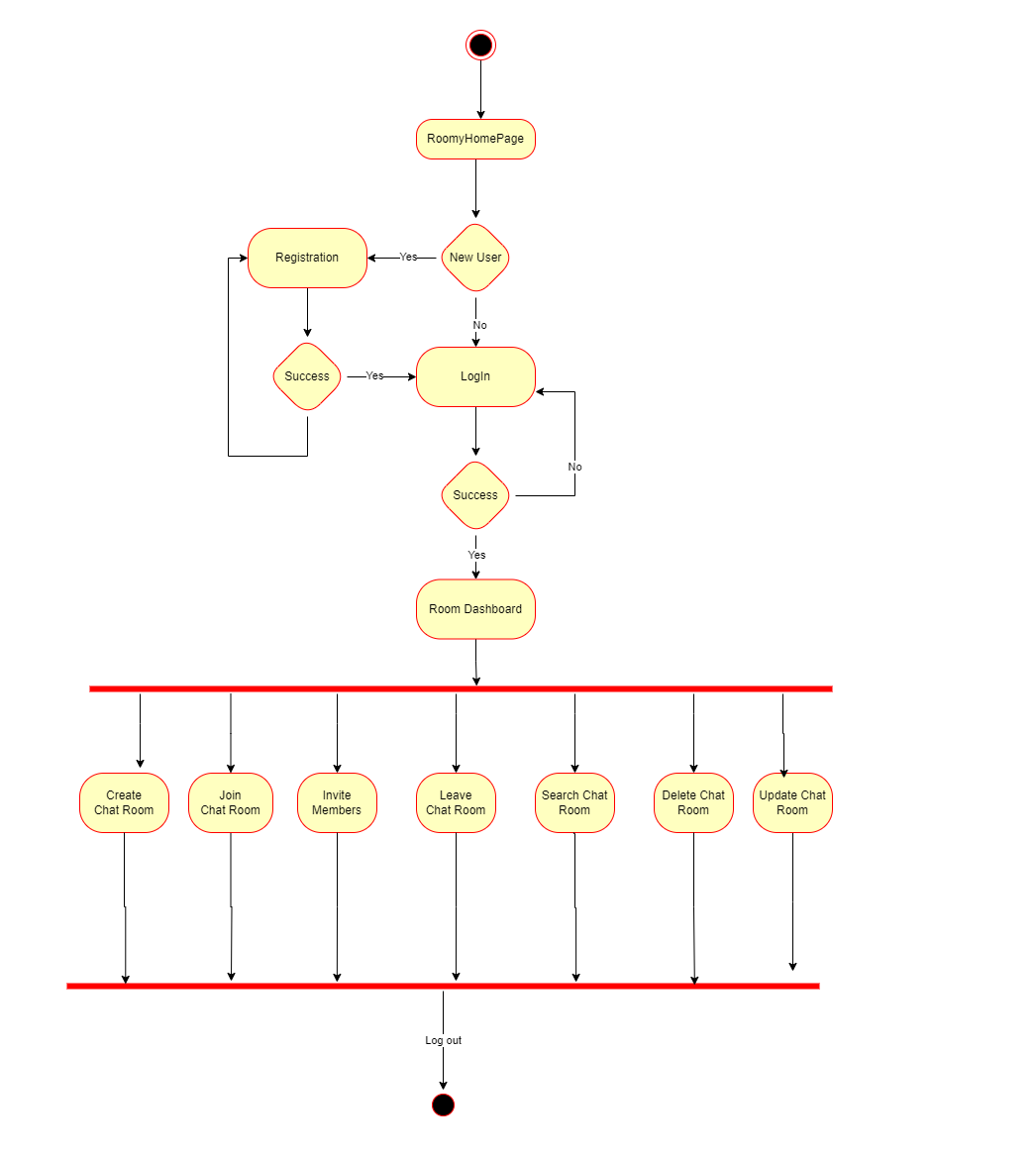
This SSD describes the process by which a user or group admin updates the room name, with the system validating and saving the new name before confirming the update to the user.

****

**Description:**

This SSD shows how a user or group admin deletes a chat room, with the system removing the chat room and notifying the user of its successful deletion.

## Activity Diagram:



**Description:**

The activity diagram for Roomy illustrates the flow of activities that users and the system perform to complete various tasks. It includes user actions such as registering, logging in, managing profiles, creating and joining chat rooms, inviting others, searching for rooms, and logging out. The diagram also highlights system processes such as validating user inputs, retrieving and updating data, and handling authentication and security.

# Data Design

**Transformation of Information Domain into Data Structures**

Roomy's information domain includes user interactions and system functionalities related to user management, chat rooms, communication, and security. These are transformed into structured data entities stored in a relational database.

## Major Data Entities

**User:**

**Attributes:** user\_id, email, password\_hash, name, bio, profile\_picture, created\_at, updated\_at

**Function:** Stores user account and profile information.

**ChatRoom:**

**Attributes:** room\_id, room\_name, creator\_id, is\_temporary, expiry\_time, created\_at, updated\_at

**Function:** Stores details of chat rooms.

**Message:**

**Attributes:** message\_id, room\_id, user\_id, content, type, created\_at

**Function:** Stores messages within chat rooms.

## Databases and Data Storage

Database: Roomy uses a DBMS (MongoDB).

Below is a JSON code that includes example documents for each of the collections (User, ChatRoom, Message, Invitation, and PasswordResetRequest). This JSON code is designed for saving data and used to initialize Roomy MongoDB database.

{

"users": [

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1a" },

"email": "abc@example.com",

"password\_hash": "hashed\_password\_1",

"name": "ABC",

"bio": "Developer and tech enthusiast.",

"profile\_picture": "path/to/john.jpg",

"created\_at": { "$date": "2023-06-25T12:00:00Z" },

"updated\_at": { "$date": "2023-06-25T12:00:00Z" }

},

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1b" },

"email": "jane.smith@example.com",

"password\_hash": "hashed\_password\_2",

"name": "Jane Smith",

"bio": "Designer and artist.",

"profile\_picture": "path/to/jane.jpg",

"created\_at": { "$date": "2023-06-25T12:00:00Z" },

"updated\_at": { "$date": "2023-06-25T12:00:00Z" }

}

],

"chatRooms": [

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1c" },

"room\_name": "General Chat",

"creator\_id": { "$oid": "60d5ec50f80d5a5b5d723a1a" },

"is\_temporary": false,

"expiry\_time": null,

"created\_at": { "$date": "2023-06-25T12:00:00Z" },

"updated\_at": { "$date": "2023-06-25T12:00:00Z" }

},

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1d" },

"room\_name": "Project Room",

"creator\_id": { "$oid": "60d5ec50f80d5a5b5d723a1b" },

"is\_temporary": true,

"expiry\_time": { "$date": "2023-07-01T12:00:00Z" },

"created\_at": { "$date": "2023-06-25T12:00:00Z" },

"updated\_at": { "$date": "2023-06-25T12:00:00Z" }

}

],

"messages": [

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1e" },

"room\_id": { "$oid": "60d5ec50f80d5a5b5d723a1c" },

"user\_id": { "$oid": "60d5ec50f80d5a5b5d723a1a" },

"content": "Hello, everyone!",

"type": "text",

"created\_at": { "$date": "2023-06-25T12:01:00Z" }

},

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a1f" },

"room\_id": { "$oid": "60d5ec50f80d5a5b5d723a1c" },

"user\_id": { "$oid": "60d5ec50f80d5a5b5d723a1b" },

"content": "Hi, John!",

"type": "text",

"created\_at": { "$date": "2023-06-25T12:02:00Z" }

}

],

"invitations": [

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a20" },

"room\_id": { "$oid": "60d5ec50f80d5a5b5d723a1c" },

"sender\_id": { "$oid": "60d5ec50f80d5a5b5d723a1a" },

"recipient\_email": "alice@example.com",

"invitation\_link": "http://example.com/invite/12345",

"created\_at": { "$date": "2023-06-25T12:03:00Z" }

}

],

"passwordResetRequests": [

{

"\_id": { "$oid": "60d5ec50f80d5a5b5d723a21" },

"user\_id": { "$oid": "60d5ec50f80d5a5b5d723a1a" },

"reset\_token": "token\_12345",

"created\_at": { "$date": "2023-06-25T12:04:00Z" },

"expires\_at": { "$date": "2023-06-26T12:04:00Z" }

}

]

}

### Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **DFD** | DFD Level 2 | Function Name(s) |
| R1 | User Registration | User Registration Process | Register user |
| R2 | Log In | User Authentication Process | Log In user |
| R3 | Manage Profile | Profile Management Process | Update user profile |
| R4 | Create Chat Room | Create Chat Room Process | Create chat room |
| R5 | Invite members | Invite members Process | Invite other members |
| R6 | Join Chat Room | Join Chat Room Process | Join chat room |
| R7 | Update Chat Room | Update Chat Room Process | Update chat room |
| R8 | Search Chat Room | Search Chat Room Process | Search chat room |
| R9 | Delete Chat Room | Delete Chat Room Process | Delete chat room |
| R10 | Log Out | Log Out process | Log out user |

# Human interface design

The interfaces for Roomy encompass user management, chat room functionality, messaging, invitations, and password recovery. Users can register, authenticate, and manage their profiles. Chat rooms can be created, joined, and managed, allowing users to send and receive messages in real-time. Invitations can be sent to invite others to chat rooms, and a password recovery process ensures account security through reset requests and token-based password resets. These interfaces collectively provide a comprehensive framework for seamless interaction and communication within Roomy.