**Foram**

Software Requirement Specification

for

Project-Work-I

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Pradyumna Rajnekar**

**EN21CS301561**

**&**

**Pranay Khandelwal**

**EN21CS301568**

Under the Guidance of

**Dr. Pinky Rane**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**AUG 2024 - DEC 2024**

**Introduction**

In today’s digital landscape, the proliferation of online platforms has transformed how individuals engage in discussions and collaborations. However, many users face significant challenges in finding and participating in structured, relevant conversations. Existing platforms often lack the necessary framework for focused, topic-specific discussions, leading to a fragmented and inefficient user experience. This is particularly problematic for students, professionals, and project collaborators, as the difficulty in accessing organized discussions can hinder productivity and limit the effectiveness of their interactions. The overload of unrelated content disrupts the coherence of conversations, making it hard for users to navigate and find communities that align with their specific needs.

The need for a more structured, topic-based platform is evident. Users require a solution that categorizes discussions and maintains a focused environment conducive to productive exchanges. Foram addresses this gap by offering a web application designed to create and join rooms dedicated to specific study or work-related topics. This structured approach ensures users can easily find and engage with conversations that align with their current needs, fostering a more organized and productive experience.

### **Problem Statement**

The proliferation of online discussion platforms has resulted in a chaotic environment where users struggle to find relevant, focused discussions on specific topics. This disorganization particularly impacts students, professionals, and project collaborators who require structured spaces to engage in meaningful interactions. Current platforms fail to effectively categorize discussions, leading to fragmented communication and a dilution of valuable exchanges. Consequently, there is a pressing need for a solution that offers dedicated environments for topic-specific discussions. Foram aims to address this issue by providing a web application that facilitates the creation and participation in organized discussion rooms, thereby enhancing productivity and promoting valuable engagement among users.

### **Functional Requirements**

1. **User Management:**
   * **Registration & Authentication:** Users can securely register, log in, and log out of their accounts.
   * **Profile Management:** Users can create, edit, and view their profiles, including preferences for topics and active discussion rooms.
2. **Discussion Room System:**
   * **Topic-Specific Rooms:** Users can create and join dedicated discussion rooms for specific topics, enhancing focused conversations.
   * **Access Control:** Users can only participate in rooms relevant to their selected topics to maintain organized discussions.
3. **Search Functionality:**
   * **Search Topics:** Users can search for specific discussion topics to quickly find relevant rooms.
4. **User Roles and Permissions:**
   * **Moderators:** Appointed users can manage discussion rooms, mute participants, and remove inappropriate content to maintain a constructive environment.
   * **General Users:** Participants can join discussions, edit their profiles, and manage their participation in various rooms.
5. **Notification System:**
   * **Activity Notifications:** Users receive notifications for messages and updates in rooms they are participating in, ensuring they stay informed about discussions.

### **Interface Requirements**

1. **User Interface (UI):**
   * **Main Dashboard:** Displays available discussion rooms and popular topics for engagement.
   * **Discussion Room Interface:** Features message input, chat history, and a list of participants, clearly indicating the current room.
   * **Responsive Design:** Optimized for both desktop and mobile devices for a smooth user experience.
2. **Navigation:**
   * **Room Switching:** Users can easily switch between discussion rooms through a simple menu.
   * **Search Bar:** Allows quick access to specific topics or rooms.
   * **Side Menu:** Provides easy access to user profiles and notifications for intuitive navigation.

### **Performance Requirements**

1. **Scalability:**
   * The system should be capable to scale up to support video and phone calls in the future.
2. **Latency:**
   * Messages within discussion rooms should ensure timely communication among users.
3. **Load Handling:**
   * The application must be able to manage traffic peaks..

### **Design Constraints**

1. **Platform Constraints:**
   * The app is designed for web browsers and is optimized for desktop and tablet screens. It is not intended for mobile devices in this version.
2. **Technological Stack:**
   * The app must use the following technologies:
     + SQLite for the database.
     + Django for the backend.
     + HTML, CSS, and JavaScript for the frontend.
3. **Communication Protocol:**
   * The application will utilize standard HTTP/HTTPS for all communication between users and the server, without the use of WebSockets.
4. **Security:**
   * All communication must be encrypted using HTTPS.
   * User authentication must be implemented through secure methods, including session-based authentication or token-based mechanisms (e.g., JWT).

### **Non-Functional Requirements**

1. **Security:**
   * Users must be authenticated via secure login, and passwords should be stored securely using encryption.
2. **Usability:**
   * The UI should be intuitive and easy to navigate, particularly for new users.
   * Users should be able to join discussion rooms, search for topics, and manage their profiles with minimal effort.
3. **Availability:**
   * The system should aim for 99.9% uptime, ensuring that discussion rooms and user services are consistently available.
4. **Maintainability:**
   * The codebase will adhere to best practices in structuring and will include comprehensive documentation for future developers.
   * The application should support easy updates for discussion room functionality and the addition of new topics.

### **Roadmap**

**Phase 1 (Weeks 1-2):**

* Develop user registration, login, and authentication modules.
* Set up SQLite database and integrate it with the Django backend.
* Implement basic discussion room functionality, allowing users to create and join rooms.

**Phase 2 (Weeks 3-4):**

* Develop the topic search and filtering system for easy navigation.
* Build the frontend using HTML, CSS, and JavaScript, focusing on the dashboard and discussion room UI.
* Implement access control to prevent spoiler discussions based on user statuses.

**Phase 3 (Weeks 5-6):**

* Optimize performance, focusing on handling high loads and ensuring chat message latency is under 2 seconds.
* Conduct user testing sessions to gather feedback on usability and functionality.
* Make necessary improvements based on user feedback.

**Phase 4 (Weeks 7-8):**

* Finalize the design and enhance usability based on testing insights.
* Prepare for launch, including addressing any last-minute issues.
* Launch beta testing for a broader user base and continue to collect and address feedback.

### **Appendices**

1. **Technology Stack Details:**
   * **SQLite:** Lightweight relational database for storing user data, discussion messages, and topic information.
   * **Django:** Backend framework for handling requests, routing, and managing user authentication.
   * **HTML/CSS/JavaScript:** Technologies used for building a dynamic and responsive frontend interface.
   * **HTTP/HTTPS:** Protocols for secure communication between users and the server.
2. **Glossary:**
   * **Discussion Room:** A virtual space where users can engage in focused conversations on specific topics.
   * **Spoiler Prevention:** A mechanism to restrict users from viewing discussions about content they have not yet completed.
   * **User Authentication:** The process of verifying the identity of a user to ensure secure access to the application.
   * **JWT (JSON Web Tokens):** A compact, URL-safe means of representing claims securely between parties, used for user session management.