PMU-Mentor

Raghad Aldossary

Table of Contents

1 Introduction

PmuMentor is an innovative mobile application specifically designed for the students of Prince Mohammad Bin Fahd University (PMU). The application aim to provides a mentorship platform where more experience students (mentors) can offer a guide, support, and advices to less experience students (mentees). Through this application, we strive to enhancing academic success, providing professional advice, and assisting in personal development, thereby contributing to building a cohesive and collaborative university community.

Today's academic environments, and students often faced challenges that can hinder their educational journey. PmuMentor seek to bridge the gap between student by fostering a supportive networks that encourages knowledge sharing, collaboration, and mutual grow.

2 Problem Statement

Lack of Communication Among Students

Many new students struggle to adapt to the new university environment. They may feel isolated, overwhelmed by academic demands, and uncertain about how to navigate university life. The absence of a structured support system makes it difficult for them to seek guidance and mentorship from their more experienced peers.

Limited Available Resources

Students often find it challenging to access reliable resources that can aid them in their studies or provide insights into professional opportunities. The lack of a centralized repository for educational materials, internships, scholarships, and career advice hampers their ability to make informed decisions about their academic and professional futures.

Absence of a Unified Platform

Currently, there is no centralized platform that connects mentors and mentees within the university. This fragmentation leads to missed opportunities for mentorship, collaboration, and community building. Without a dedicated space for peer interaction, students are less likely to engage with one another in meaningful ways that enhance their university experience.

3 Objectives of the System

Facilitate Communication and Mentorship

- **Create a User-Friendly Platform:** Develop an application that allows students to easily register, create profiles, and connect with one another based on shared interests and academic fields.
- **Enable Mentor-Mentee Connections:** Provide tools for mentees to find and connect with mentors who can offer personalized guidance and support.

Provide Resources and Support

- **Resource Sharing:** Allow mentors to share study guides, articles, tutorials, and information about internships, scholarships, and career opportunities.
- **Event Creation:** Enable mentors to organize virtual or in-person events such as workshops and webinars to disseminate knowledge and skills.

Enhance the University Community

- **Build an Interactive Community:** Foster an environment that encourages students to participate in group discussions, join chat rooms, and collaborate on projects.
- **Recognition and Rewards:** Implement a rewards system to recognize active mentors, encouraging continued participation and contribution to the community.

Ensure Accessibility and Inclusivity

- **Multiplatform Availability:** Develop the application using Flutter to ensure compatibility with both Android and iOS devices, making it accessible to all students.
- **Personal Development:** Support students in their personal growth by providing tools and resources that cater to diverse needs and learning styles.

4 Functional Requirements

1. **User Registration and Authentication**
    - **Account Creation**: Allow new users to create an account by providing necessary details such as first name, last name, email, password, and user type (mentor or mentee).
    - **Email Verification**: Send a confirmation email to verify the user's email address upon registration.
    - **Login**: Enable users to log in using their registered email and password.
    - **Password Recovery**: Provide options for users to recover or reset their passwords via email verification.
2. **Profile Management**
    - **Profile Editing**: Allow users to update personal information, including profile picture, bio, education, experience, skills, and interests.
    - **Privacy Settings**: Enable users to set privacy preferences for their profiles and control the visibility of their information.
    - **Social Media Linking**: Allow users to link their social media accounts to their profiles.
3. **Mentor-Mentee Matching**
    - **Search Functionality**: Enable mentees to search for mentors based on criteria such as field of study, interests, availability, and ratings.
    - **Connection Requests**: Allow mentees to send connection requests to mentors with an optional personalized message.
    - **Request Management**: Enable mentors to accept or decline connection requests and view pending requests.
4. **Event Management**
    - **Event Creation**: Allow mentors to create virtual or in-person events like workshops and webinars by providing event details.

- **Event Viewing**: Enable all users to view upcoming events, event details, and register to attend.
- **Event Notifications**: Send notifications to relevant users about new events and event updates.
- **Calendar Integration**: Display events in a calendar view within the application.

5. **Resource Sharing**
    - **Upload Resources**: Allow mentors to upload or link study guides, articles, tutorials, and information about opportunities.
    - **Resource Browsing**: Enable users to view, search, and download resources shared on the platform.
    - **Categorization**: Allow resources to be categorized and tagged for easier navigation.
6. **Communication System**
    - **Messaging**: Provide a messaging system for users to communicate individually in real-time.
    - **Group Chat**: Enable users to participate in group chat rooms focused on specific topics or interests.
    - **Notification of Messages**: Send notifications to users when they receive new messages.
7. **Notifications and Alerts**
    - **Activity Notifications**: Notify users about new messages, events, connection requests, feedback, and other relevant activities.
    - **Notification Center**: Provide a centralized location within the app where users can view all their notifications.
    - **Customization**: Allow users to customize which notifications they receive and how they are alerted (e.g., push notifications, email).
8. **Ratings and Feedback**

- **Submit Ratings**: Enable mentees to rate mentors on a predefined scale and provide written feedback.
- **Display Ratings**: Show average ratings and recent feedback on mentor profiles.
- **Feedback Notifications**: Notify mentors when they receive new ratings or feedback.

9. **Rewards System**
    - **Activity Tracking**: Track mentor activities such as sessions conducted, resources shared, and events hosted.
    - **Points and Badges**: Award points or badges to mentors based on their contributions.
    - **Leaderboard**: Display a leaderboard showcasing top mentors within the community.
    - **Profile Integration**: Display mentors' achievements and rewards on their profiles.
10. **Security and Privacy**
    - **Data Protection**: Secure all user data with encryption both in transit and at rest.
    - **Authentication**: Implement secure authentication mechanisms to prevent unauthorized access.
    - **Compliance**: Ensure compliance with data protection laws and regulations.
11. **Administrative Functions**
    - **User Management**: Allow administrators to manage user accounts, including suspending or deleting accounts if necessary.
    - **Content Moderation**: Provide tools for administrators to monitor and moderate content such as messages, resources, and feedback.
    - **Analytics and Reporting**: Offer administrative dashboards with analytics on user engagement and platform usage.
12. **Support and Help**
    - **Help Center**: Provide a help section with FAQs, tutorials, and guides.

- **Contact Support**: Allow users to contact support for assistance or to report issues.

5 Non-Functional Requirements

1. **Performance**
    - **Response Time**: The application should load main screens within 2 seconds under normal network conditions.
    - **Scalability**: Support up to 10,000 concurrent users without performance degradation.
    - **Real-Time Updates**: Ensure real-time features like messaging and notifications have minimal latency.
2. **Security**
    - **Data Encryption**: Use industry-standard encryption methods for data storage and transmission.
    - **Authentication and Authorization**: Implement robust authentication mechanisms and role-based access control.
    - **Vulnerability Protection**: Protect against common security threats such as SQL injection, XSS, CSRF, and others.
    - **Regular Security Audits**: Conduct periodic security assessments and penetration testing.
3. **Usability**
    - **User Interface**: Provide an intuitive and consistent user interface that is easy to navigate.
    - **Accessibility**: Comply with accessibility standards (e.g., WCAG 2.1) to support users with disabilities.
    - **Multilingual Support**: Design the system to support multiple languages if necessary.
4. **Reliability and Availability**
    - **Uptime**: Ensure the system is available 99.5% of the time.
    - **Error Handling**: Implement comprehensive error handling to maintain system stability.

- **Data Integrity**: Prevent data loss or corruption through transactional integrity and backups.

5. **Maintainability**
    - **Modularity**: Design the system with a modular architecture to facilitate updates and maintenance.
    - **Documentation**: Maintain thorough documentation for code, APIs, and system configurations.
    - **Code Quality**: Adhere to coding standards and best practices to ensure code readability and maintainability.
6. **Portability**
    - **Platform Compatibility**: Ensure the application runs smoothly on the latest versions of Android and iOS.
    - **Device Support**: Support a wide range of devices with varying screen sizes and resolutions.
7. **Scalability**
    - **Horizontal Scaling**: Design the system to allow scaling by adding more servers or resources.
    - **Cloud Readiness**: Utilize cloud services to handle increased load and storage requirements.
8. **Interoperability**
    - **API Integration**: Design APIs to allow integration with other systems or services if needed.
    - **Standards Compliance**: Use standard protocols and formats (e.g., RESTful APIs, JSON).

6 User Stories and Acceptance Criteria

User Story 1: User Registration

- **As a** new student,
- **I want** to create an account,
- **So that** I can access the PmuMentor platform and connect with mentors.

Acceptance Criteria:

1. The registration page includes fields for:
    - First Name (required)
    - Phone Number (optional)
    - Department (optional)
    - Email Address (required, must be a valid university email)
    - Password (required, minimum 8 characters, must include letters and numbers)
    - Confirm Password (must match the Password field)
    - User Type Selection (Mentor or Mentee)
2. The system validates all inputs and provides real-time feedback on any errors.
3. If all inputs are valid, the system creates a new user account.
4. A verification email is sent to the provided email address with a link to activate the account.
5. The user cannot log in until they have verified their email address.
6. If the email is already in use, the system informs the user and prompts them to log in or reset their password.

User Story 2: Email Verification

- **As a** new user,

- **I want** to verify my email address,
- **So that** I can activate my account and access the platform.

Acceptance Criteria:

1. Upon successful registration, the system sends a verification email containing a unique activation link.
2. The activation link remains valid for 24 hours.
3. Clicking the activation link activates the user's account and redirects them to the login page with a success message.
4. If the activation link has expired, the user is prompted to request a new verification email.
5. The system securely handles the activation process to prevent unauthorized access.

User Story 3: User Login

- **As a** registered user,
- **I want** to log in to the platform,
- **So that** I can access my personal dashboard and use the application's features.

Acceptance Criteria:

1. The login page includes fields for:
    - Email Address
    - Password
2. The system authenticates the user's credentials.
3. If authentication is successful:
    - The user is redirected to their dashboard.
    - A session is created to keep the user logged in.
4. If authentication fails:

- An error message is displayed stating that the email or password is incorrect.
- The user is allowed to attempt login again.

5. After 5 consecutive failed login attempts, the account is temporarily locked for 15 minutes.
6. A "Remember Me" checkbox allows users to stay logged in on trusted devices.

User Story 4: Password Recovery

- **As a** user who has forgotten my password,
- **I want** to reset my password,
- **So that** I can regain access to my account.

Acceptance Criteria:

1. The login page provides a "Forgot Password" link.
2. Clicking the link directs the user to a password recovery page where they can enter their registered email address.
3. The system sends a password reset email containing a secure, unique link.
4. The reset link is valid for 1 hour.
5. The user can set a new password that meets the password policy requirements.
6. Upon successful password reset, the user is notified and can log in with the new password.

User Story 5: Profile Management

- **As a** user,
- **I want** to update my profile information,
- **So that** others can learn about my background and interests.

Acceptance Criteria:

1. Users can access the "Edit Profile" section from their dashboard.
2. Users can update the following information:

- Profile Picture
- Bio/Summary
- Education Details
- Work Experience
- Skills and Interests
- Contact Information

3. The system validates input fields for proper format (e.g., no special characters in names).
4. Users can set privacy settings to control who can view specific profile sections.
5. Changes are saved and immediately reflected on the user's profile.
6. If the user changes their email address, email verification is required for the new address.

User Story 6: Mentor-Mentee Matching

- **As a** mentee,
- **I want** to search for mentors,
- **So that** I can find someone who aligns with my academic and professional goals.

Acceptance Criteria:

1. The search function allows filtering by:
    - Field of Study
    - Areas of Expertise
    - Availability (e.g., times, days)
    - Language Preferences
    - Rating
2. Search results display mentors with:
    - Profile Picture

- Name and Title
- Brief Bio
- Average Rating

3. Users can click on a mentor's profile for more detailed information.
4. Users can send a connection request directly from the search results or profile page.
5. The system handles large volumes of data efficiently, providing quick search results.

User Story 7: Connection Requests

- **As a** mentee,
- **I want** to send a connection request to a mentor,
- **So that** I can establish a mentorship relationship.

Acceptance Criteria:

1. Users can compose an optional personalized message when sending a connection request.
2. The mentor receives a notification of the new connection request.
3. The mentor can view the mentee's profile before accepting or declining the request.
4. The mentor's decision is communicated back to the mentee via notification.
5. The system updates the status of the request accordingly.

User Story 8: Event Creation

- **As a** mentor,
- **I want** to create events,
- **So that** I can educate and engage with mentees.

Acceptance Criteria:

1. The event creation form includes fields for:
    - Event Title

- Description
- Date and Time
- Location or Virtual Meeting Link
- Event Type (Workshop, Webinar, etc.)
- Maximum Number of Participants (optional)

2. The system validates all inputs and ensures required fields are filled.
3. Upon creation, the event is added to the events calendar and visible to all users or a targeted audience.
4. Notifications are sent to mentees who match the event's criteria.
5. The mentor can edit or cancel the event after creation, with notifications sent to registered participants.

User Story 9: Event Registration

- **As a** user,
- **I want** to register for events,
- **So that** I can participate and learn.

Acceptance Criteria:

1. Users can view event details from the events calendar or list.
2. Users can register for an event by clicking a "Register" button.
3. The system confirms registration and adds the event to the user's personal calendar within the app.
4. Users receive a confirmation notification and reminders as the event date approaches.
5. If the event reaches maximum capacity, the registration is closed, and users are informed.

User Story 10: Resource Sharing

- **As a** mentor,
- **I want** to upload educational resources,
- **So that** mentees can access additional learning materials.

Acceptance Criteria:

1. The resource upload form accepts:
    - File Upload or External Link
    - Title
    - Description
    - Category
    - Tags
2. The system supports common file types and enforces a maximum file size limit.
3. Uploaded resources are scanned for viruses and malicious content.
4. Resources are displayed in a searchable library accessible to all users.
5. Users can filter resources by category, tags, or keywords.

User Story 11: Messaging and Communication

- **As a** user,
- **I want** to send messages to other users,
- **So that** I can communicate directly with them.

Acceptance Criteria:

1. Users can initiate conversations from profiles or connection lists.
2. The messaging interface shows the conversation history in chronological order.
3. Messages are delivered in real-time with read receipts.
4. Users receive notifications for new messages when not in the messaging interface.

5. Users can report or block others for inappropriate communication.

User Story 12: Group Chat Rooms

- **As a** user,
- **I want** to join group chat rooms,
- **So that** I can participate in community discussions.

Acceptance Criteria:

1. Users can view a list of available chat rooms categorized by topics.
2. Users can join or leave chat rooms at any time.
3. Messages in chat rooms are displayed in real-time.
4. Users can create new chat rooms and set them as public or private.
5. Moderation tools are available to manage inappropriate content.

User Story 13: Ratings and Feedback

- **As a** mentee,
- **I want** to provide feedback on my mentor,
- **So that** I can acknowledge their support and inform others.

Acceptance Criteria:

1. After a mentorship session, the system prompts the mentee to rate the mentor.
2. Ratings are on a scale of 1 to 5 stars.
3. Mentees can write a textual review or feedback.
4. The mentor is notified of new feedback.
5. The system filters and removes any inappropriate or offensive content in reviews.

User Story 14: Notifications Settings

- **As a** user,

- **I want** to customize my notification preferences,
- **So that** I receive only relevant alerts.

Acceptance Criteria:

1. Users can access notification settings from their profile or app settings.
2. Users can enable or disable notifications for:
    - Messages
    - Connection Requests
    - Event Updates
    - Feedback and Ratings
    - System Announcements
3. Changes to settings are saved and immediately take effect.
4. Users can choose between push notifications, email notifications, or both.

User Story 15: Security and Privacy

- **As a** user,
- **I want** my data to be secure,
- **So that** I trust the platform with my personal information.

Acceptance Criteria:

1. All sensitive data is encrypted during transmission and storage.
2. The system uses secure authentication methods and stores passwords securely (e.g., hashed with salt).
3. Users can review and accept the privacy policy and terms of service upon registration.
4. Users can delete their accounts and data permanently upon request.
5. Regular security updates and patches are applied to the system.

7 System Design

7.1 Software Architecture

The PmuMentor application is designed using a modular and scalable architecture to ensure maintainability, flexibility, and ease of development. The system adopts a three-tier architecture consisting of:

1. **Presentation Layer (Frontend)**
    - **Technology**: Flutter (Dart language)
    - **Responsibility**: User interface and user experience (UI/UX) components, handling user interactions.
2. **Business Logic Layer (Backend)**
    - **Technology**: RESTful APIs developed using Node.js with Express.js or Python with Django/Flask.
    - **Responsibility**: Application logic, processing user requests, implementing business rules, and interacting with the data layer.
3. **Data Layer (Database)**
    - **Technology**: Firebase Firestore or an SQL database like PostgreSQL/MySQL.
    - **Responsibility**: Data storage and retrieval, managing user data, events, resources, messages, and other persistent information.

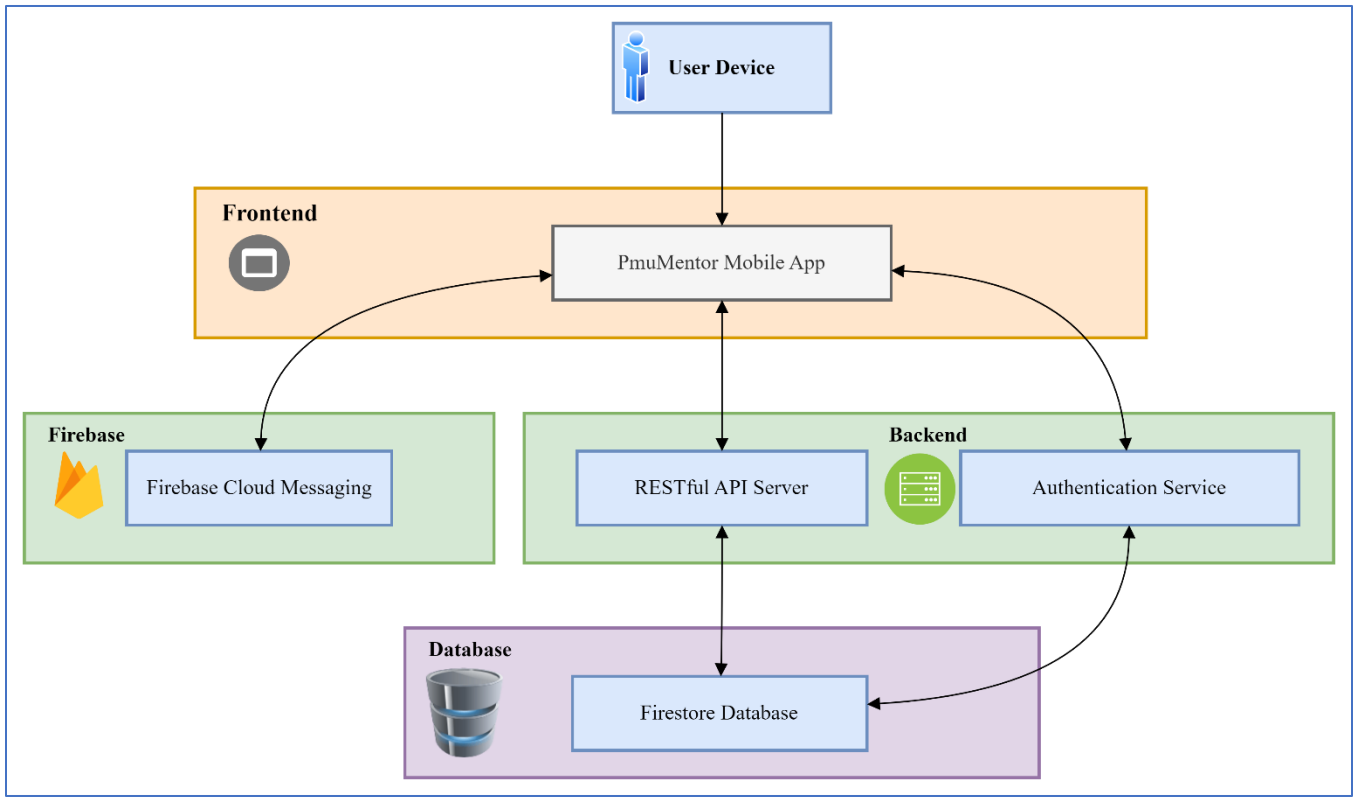
Additional Components:

- **Authentication and Authorization Service**
    - Implemented using Firebase Authentication or custom JWT-based authentication.
    - Manages user sign-up, login, password recovery, and access control.
- **Real-Time Communication Service**

- Uses WebSockets (e.g., Socket.IO) or Firebase Realtime Database for real-time features like messaging and notifications.

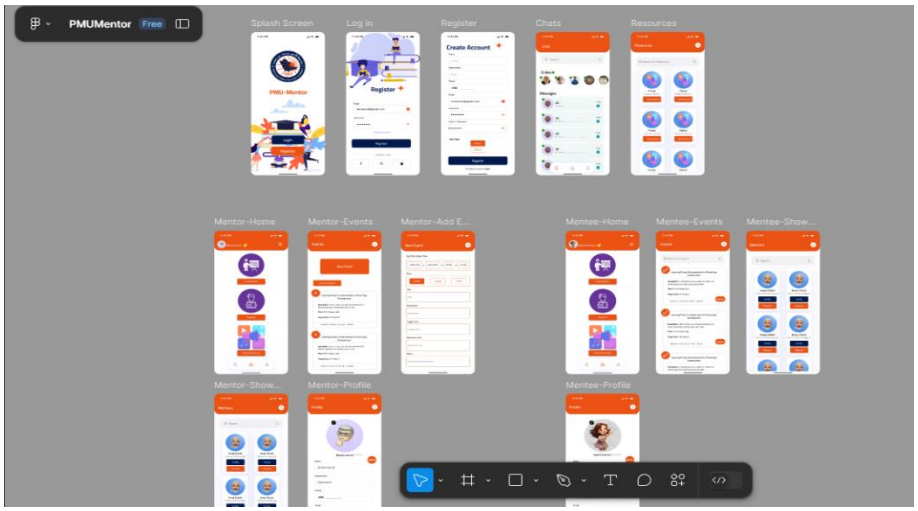
- **Push Notifications**
  - Implemented using Firebase Cloud Messaging (FCM) to send notifications to users about new messages, events, or other important updates.

Deployment Architecture Diagram:

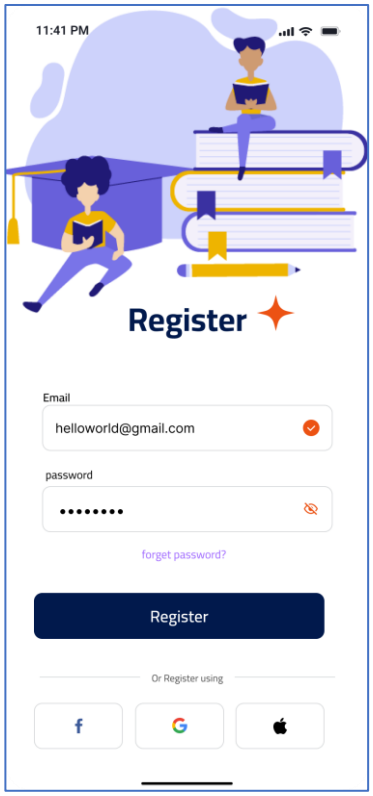


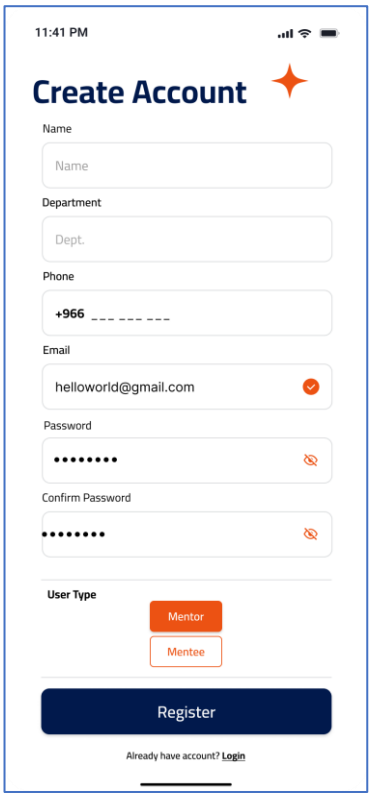
7.2 UI Design (Mockups)

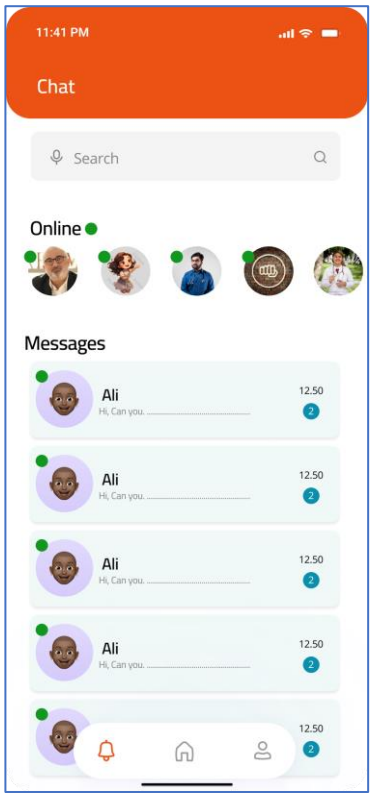
We used Figma to design the Interfaces



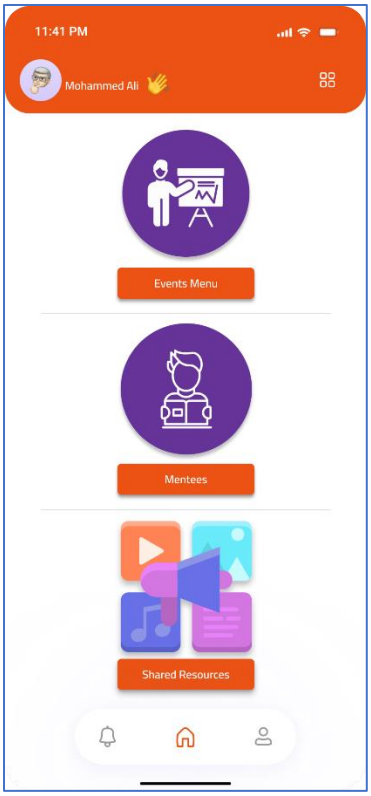
Interfaces for Whole Users:

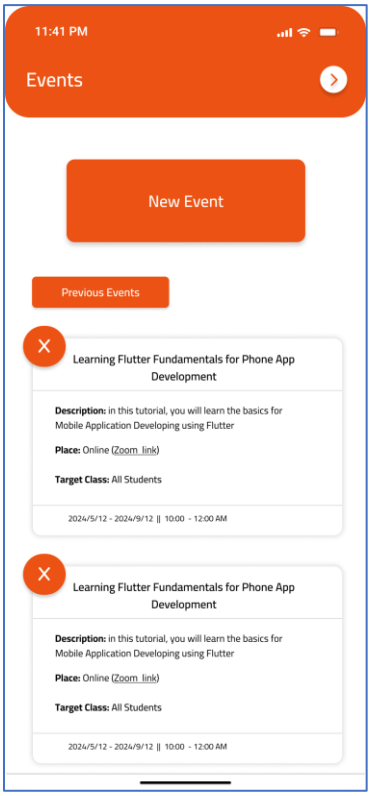


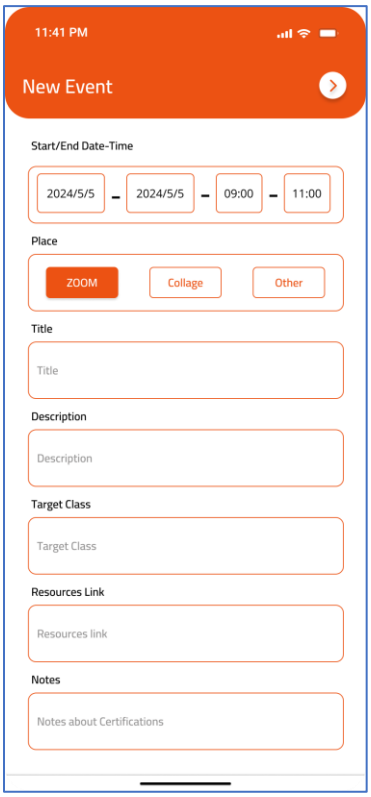


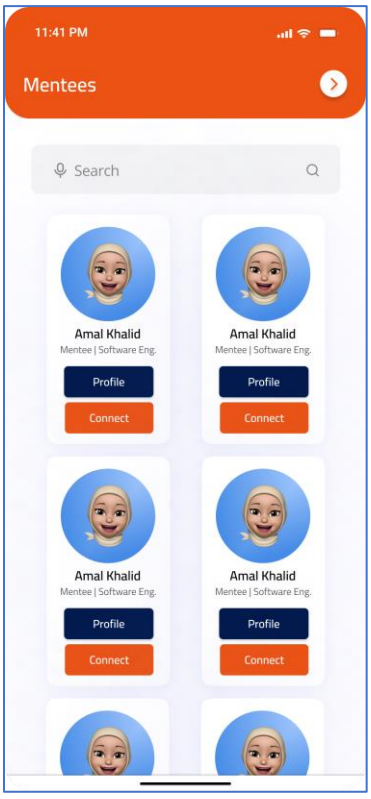


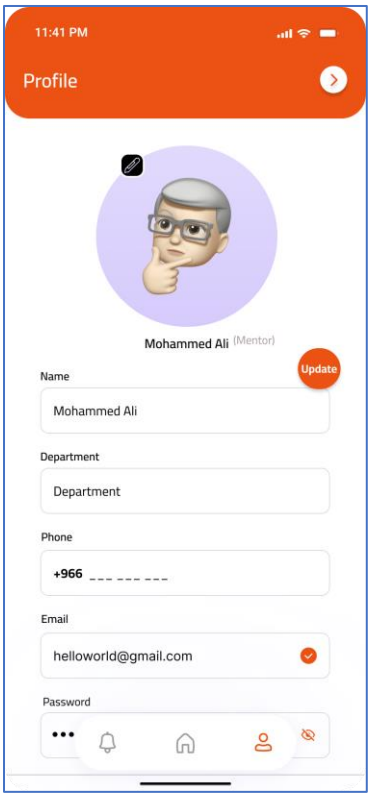
Mentor Interface:



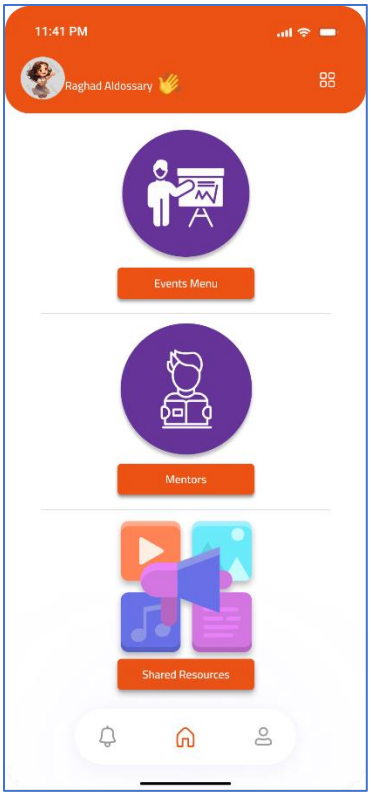


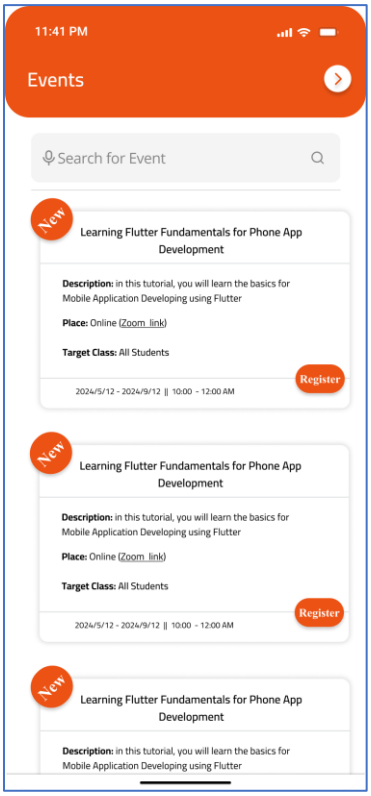


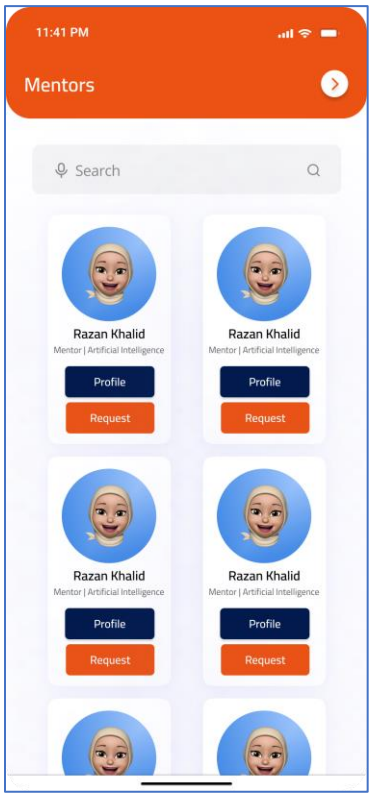


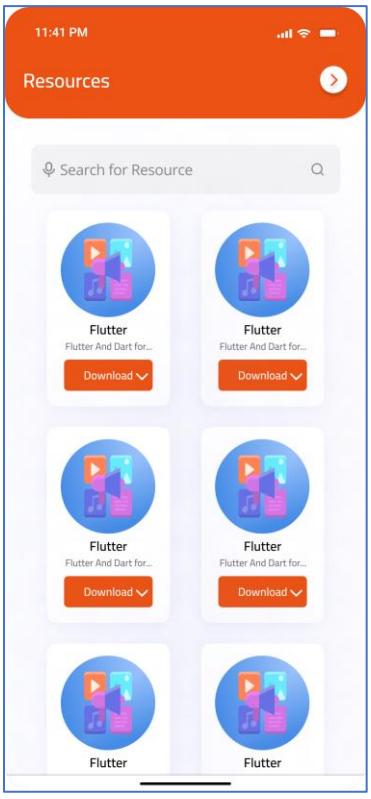


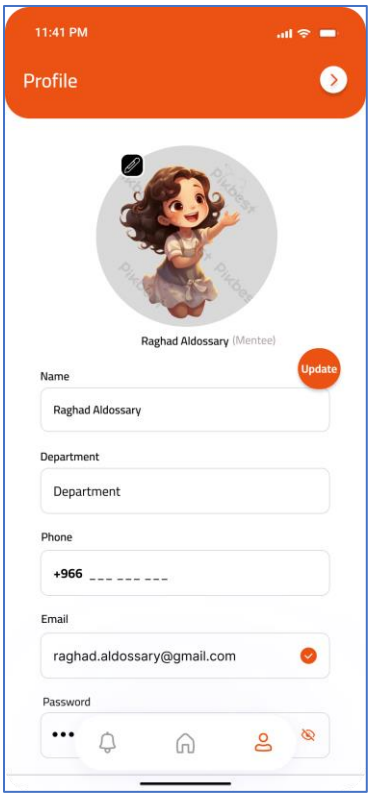
Mentee Interfaces:







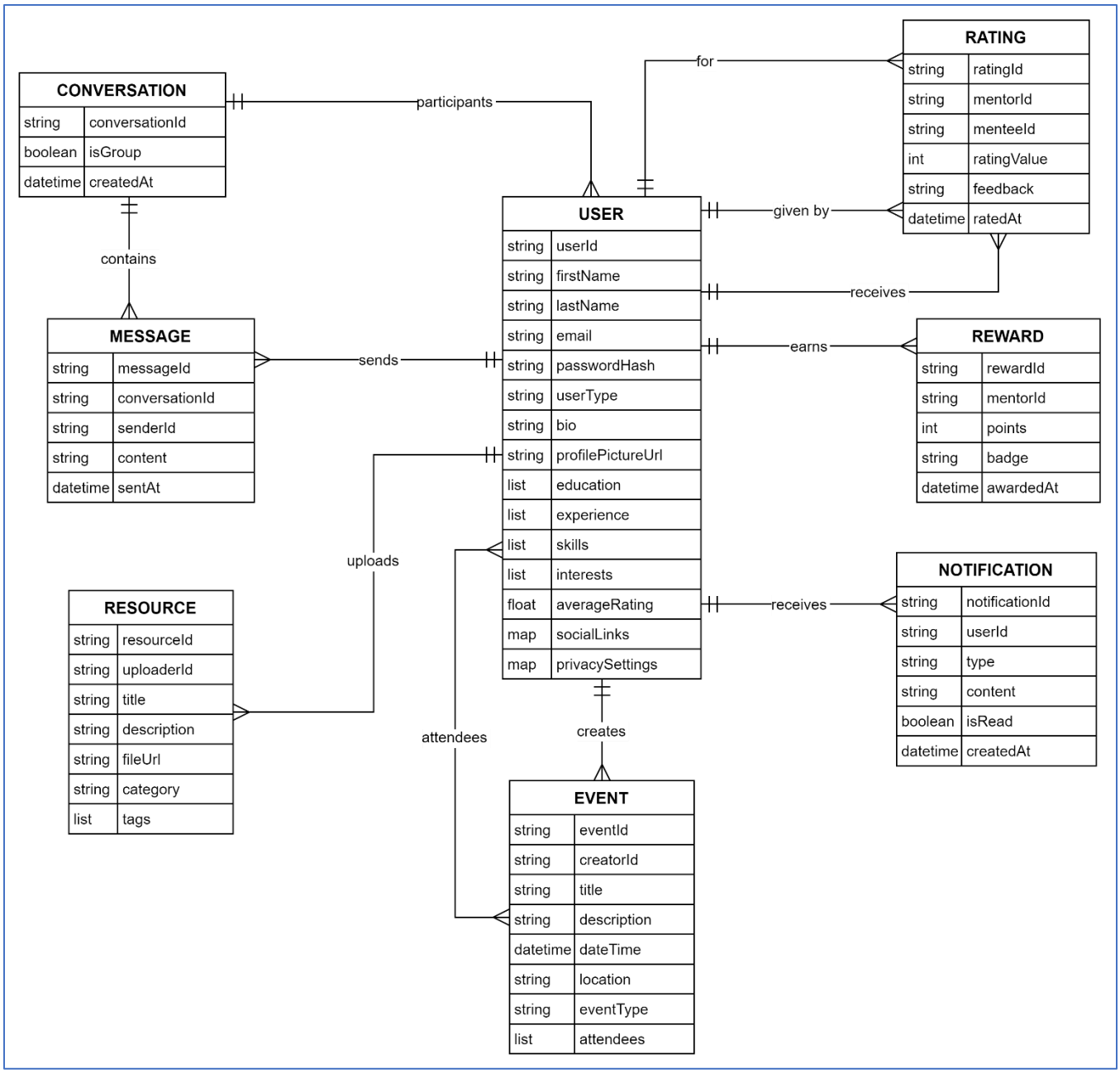




7.3 Database Design

The database design for PmuMentor involves several entities to store user data, events, resources, messages, and other relevant information. The database can be implemented using Firebase Firestore.

The following figure showing the ER Diagram for the PMU-Mentor Database:



Explanation of Entities:

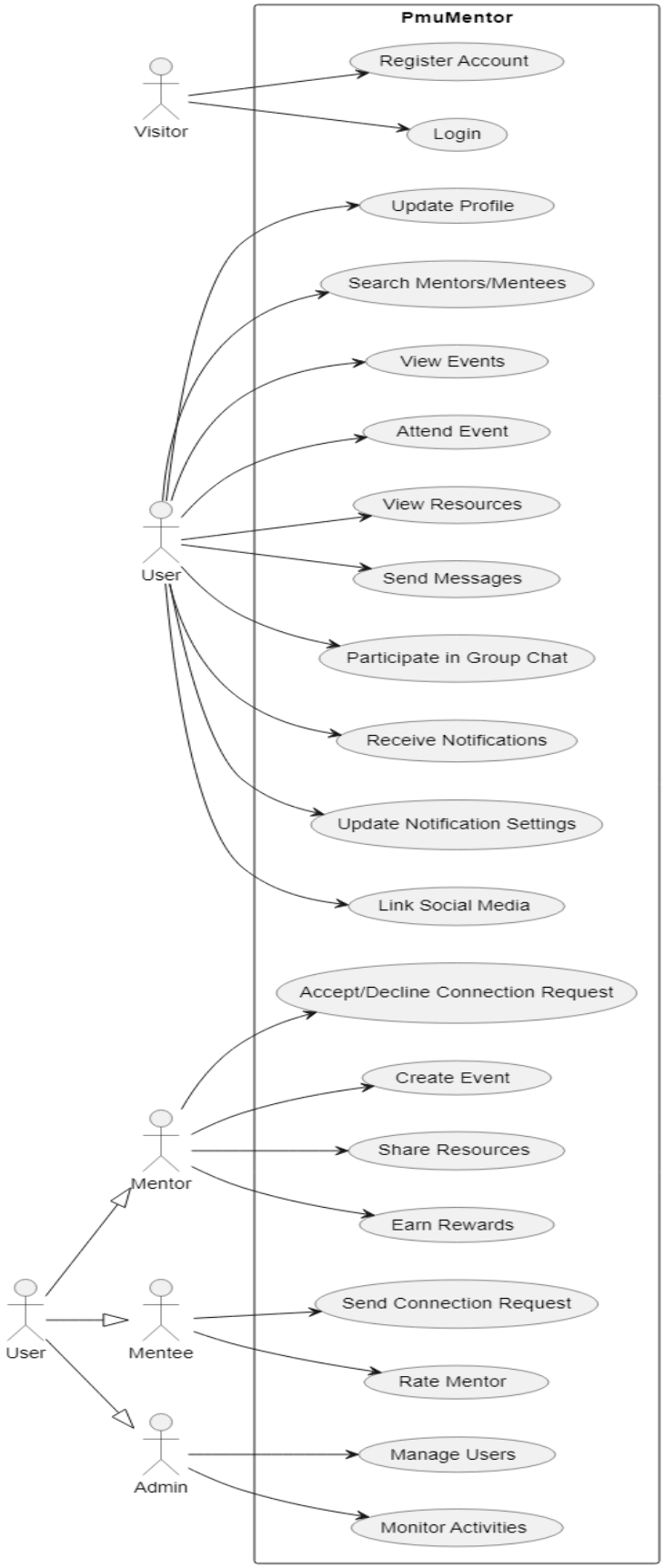
- **USER**: Stores user information.
- **EVENT**: Contains details about events created by mentors.
- **RESOURCE**: Holds information about educational resources shared by mentors.
- **MESSAGE**: Represents individual messages in conversations.
- **CONVERSATION**: Represents chat sessions, either individual or group.
- **RATING**: Stores feedback and ratings given by mentees to mentors.
- **NOTIFICATION**: Manages notifications sent to users.
- **REWARD**: Tracks rewards earned by mentors.

7.4 Program Design (UML Diagrams)

The program design includes UML diagrams such as Use Case Diagram, Class Diagram, and Sequence Diagrams to illustrate the system's functionality and structure.

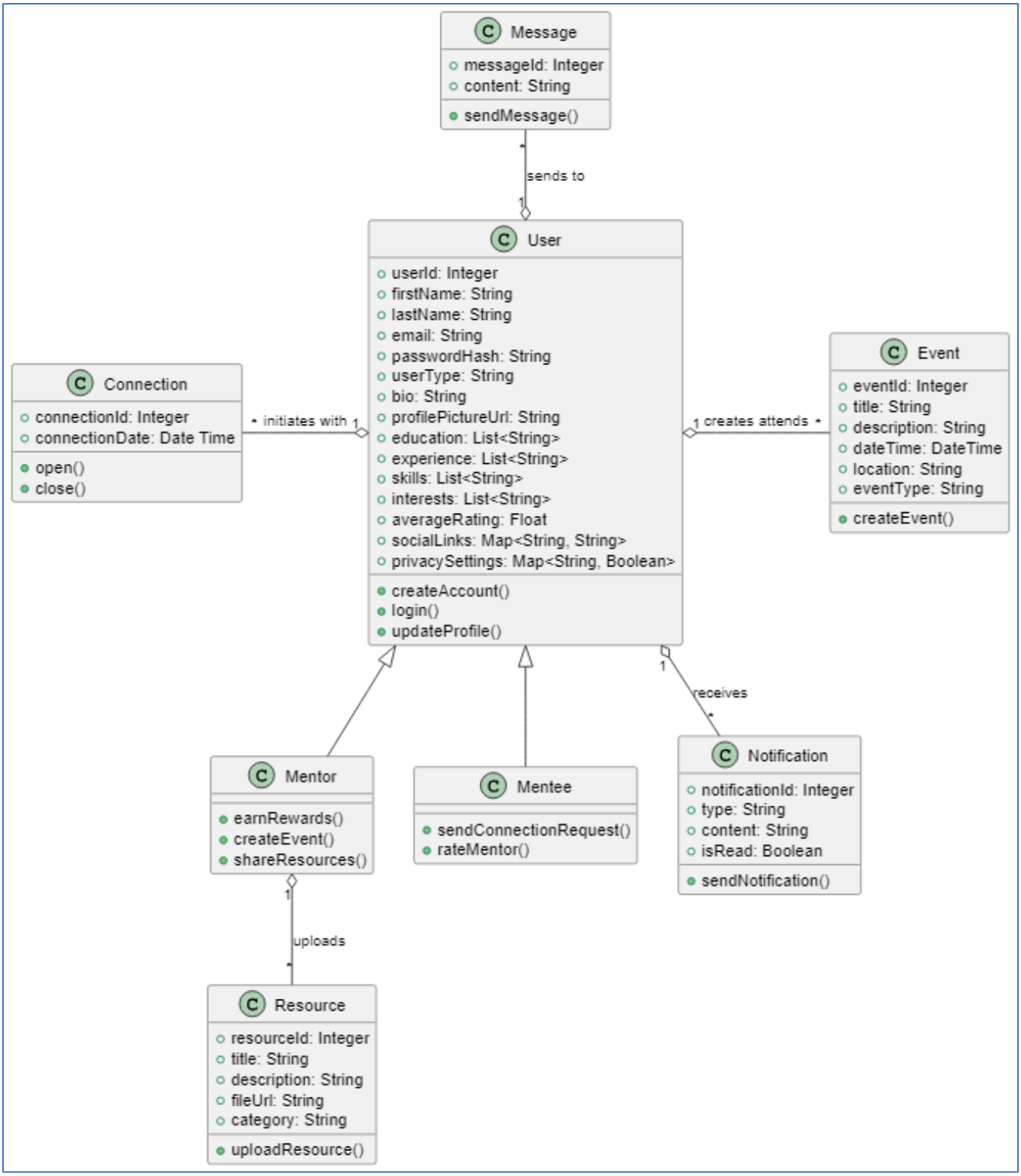
7.4.1 Use Case Diagram

The Use Case Diagram represents the interactions between users (actors) and the system.


PmuMentor
Visitor
Register Account
Login
User
Update Profile
Search Mentors/Mentees
View Events
Attend Event
View Resources
Send Messages
Participate in Group Chat
Receive Notifications
Update Notification Settings
Link Social Media
Mentor
Accept/Decline Connection Request
Create Event
Share Resources
Earn Rewards
User
Mentee
Send Connection Request
Rate Mentor
Admin
Manage Users
Monitor Activities

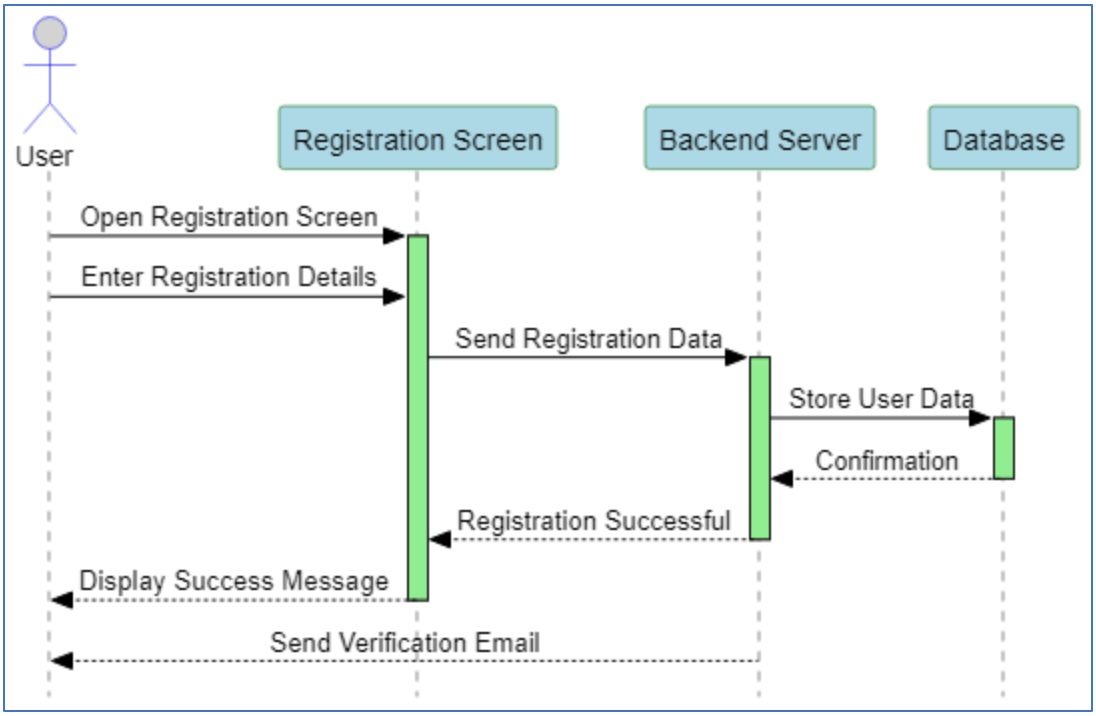
7.4.2 Class Diagram

The Class Diagram illustrates the classes, their attributes, methods, and relationships.

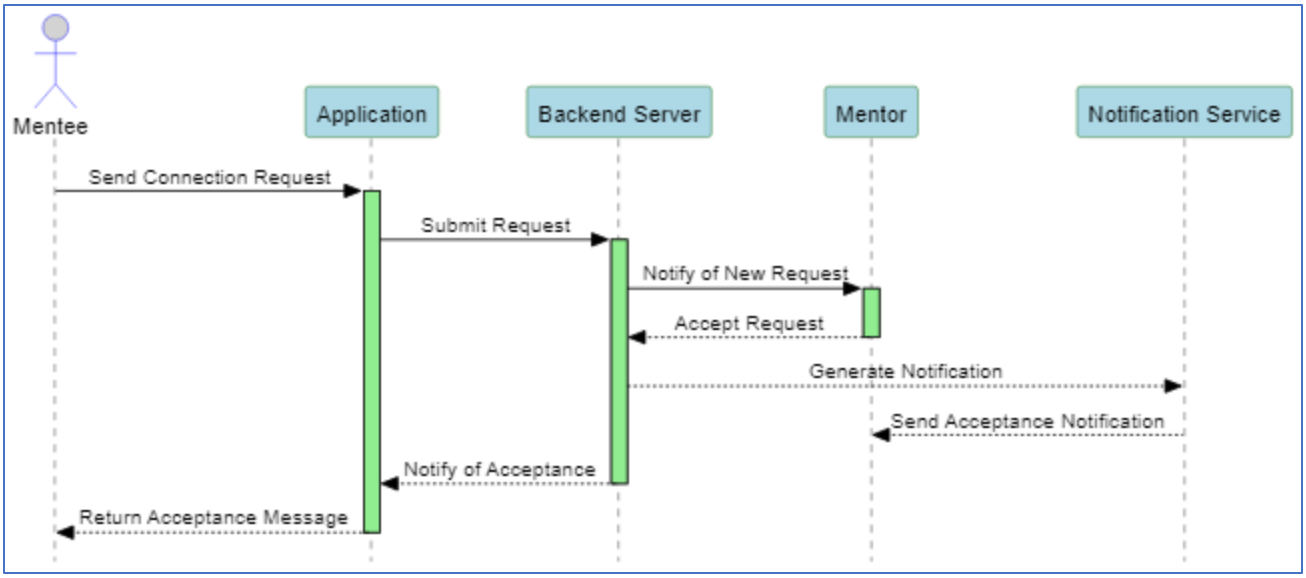


7.4.3 Sequence Diagrams

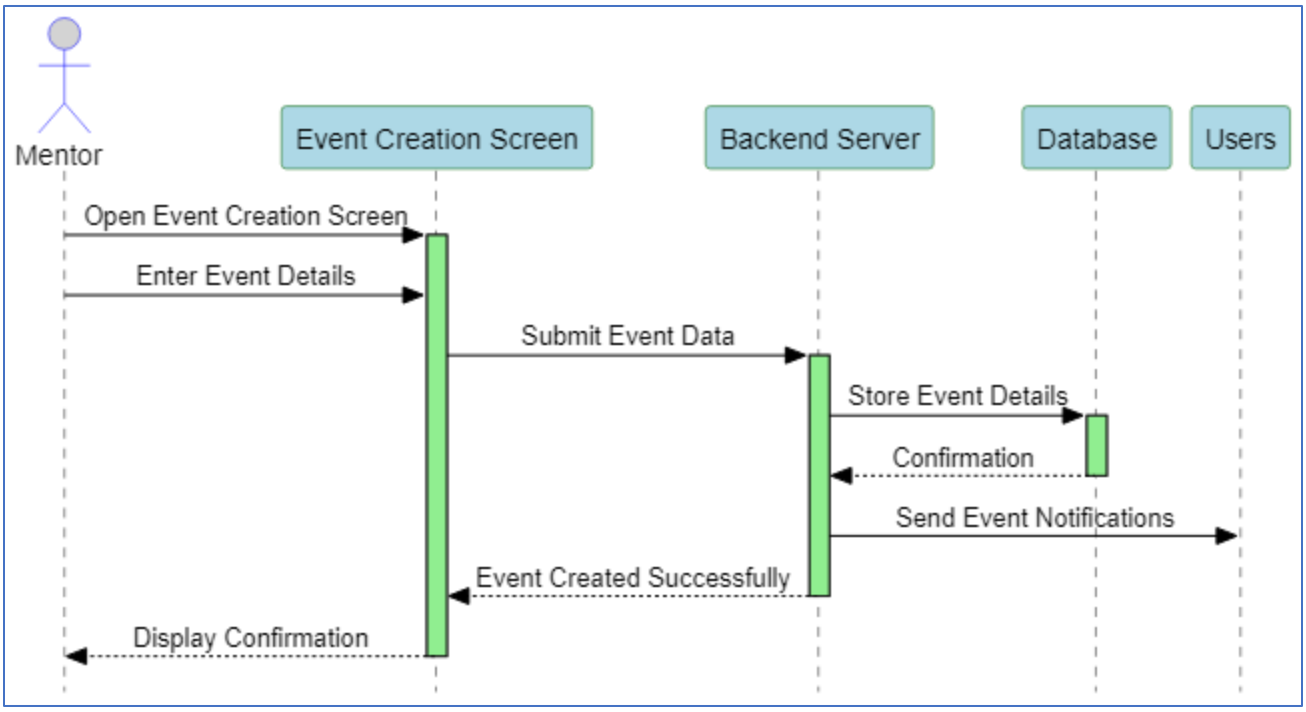
Sequence Diagram 1: User Registration



Sequence Diagram 2: Mentor Accepts Connection Request



Sequence Diagram 3: Event Creation



8 Project Management

Effective project management is crucial for the successful completion of the PmuMentor application. The team adopted the Agile methodology, specifically the Scrum framework, to manage the project efficiently and adapt to changes quickly.

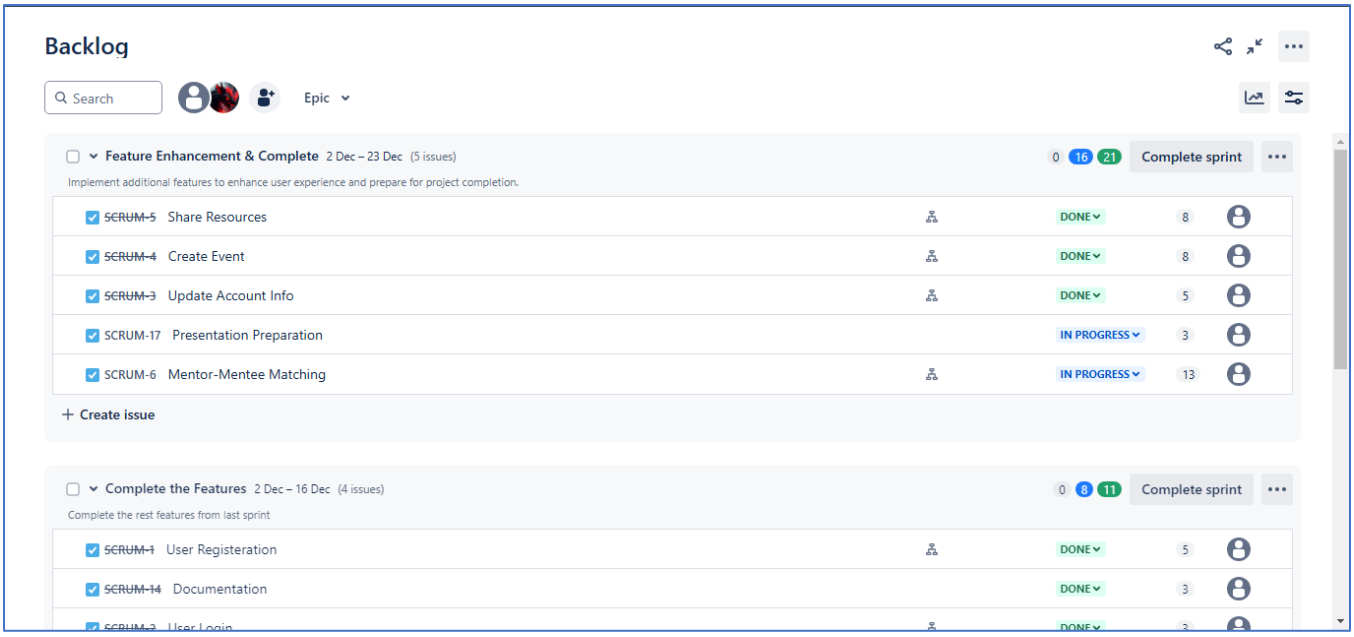
8.1 Agile Methodology (Scrum)

Scrum Framework Overview

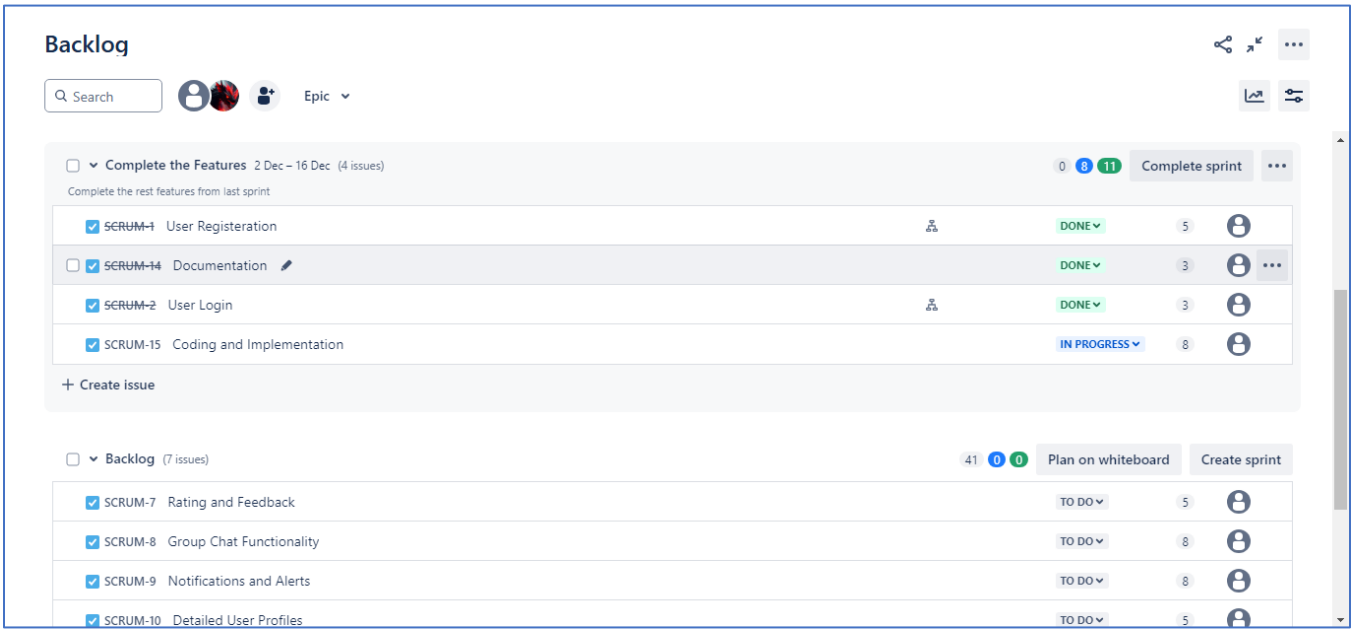
Scrum is an Agile process framework used for managing complex projects. It emphasizes teamwork, collaboration, and iterative progress toward a well-defined goal.

8.2 Jira Screenshots

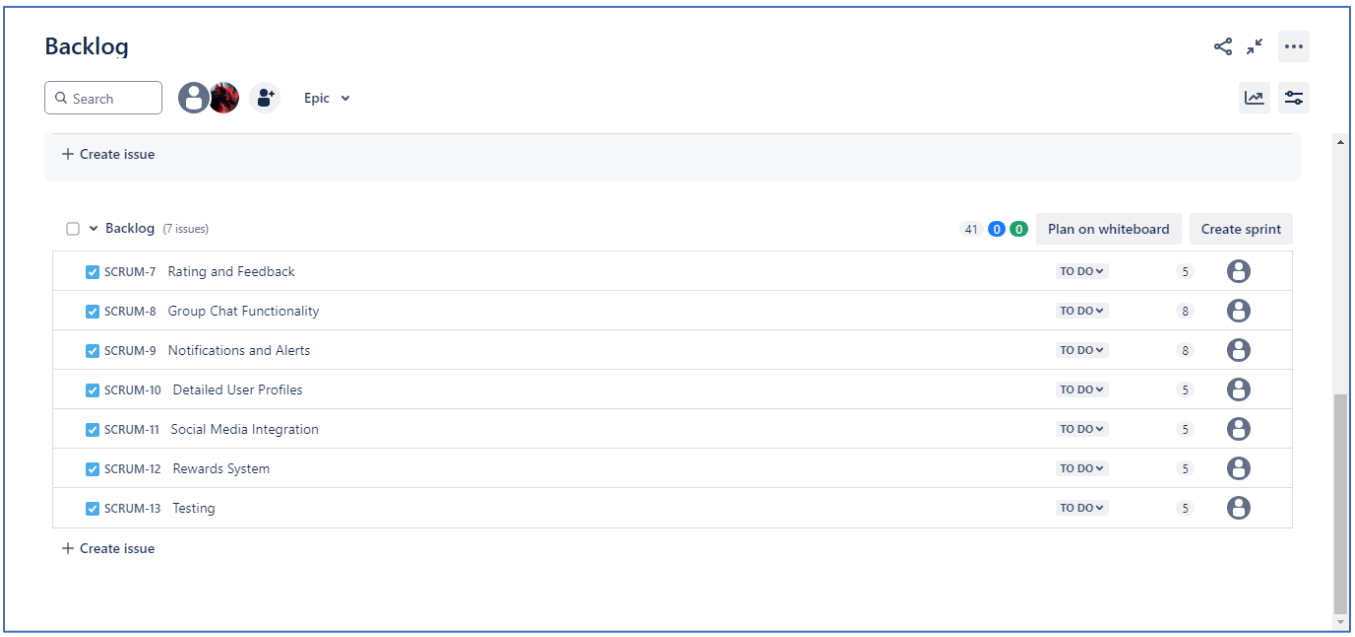
Sprint 1:



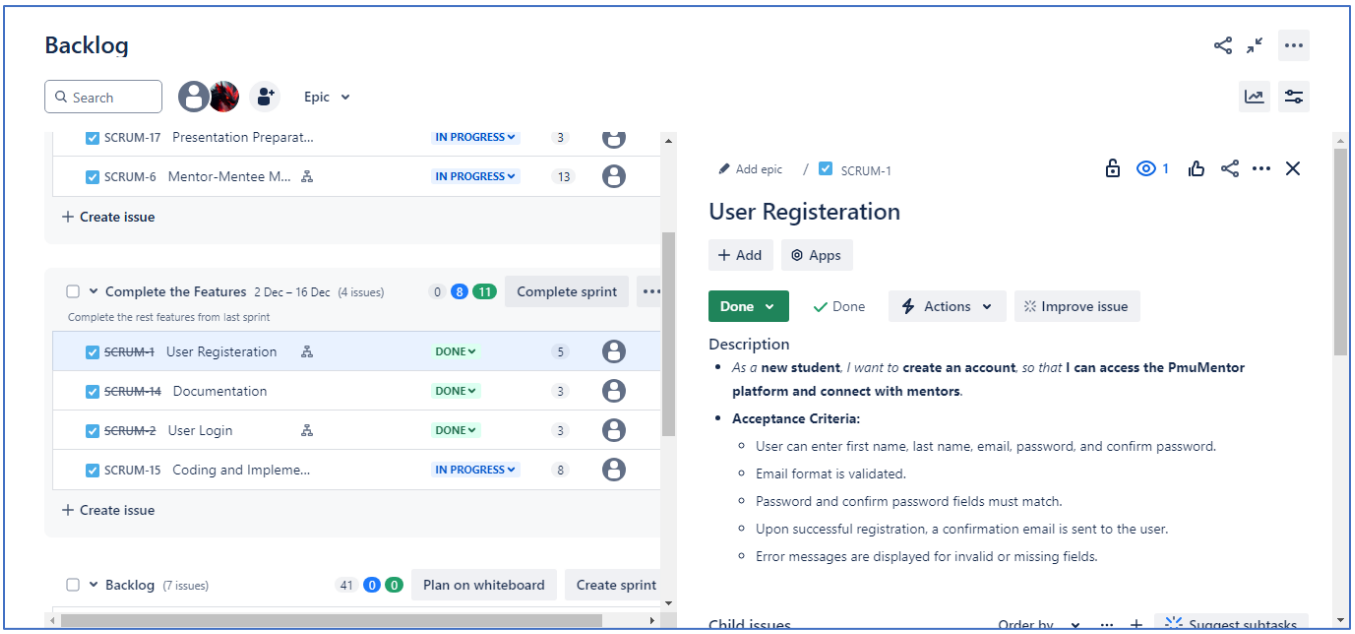
Sprint 2:



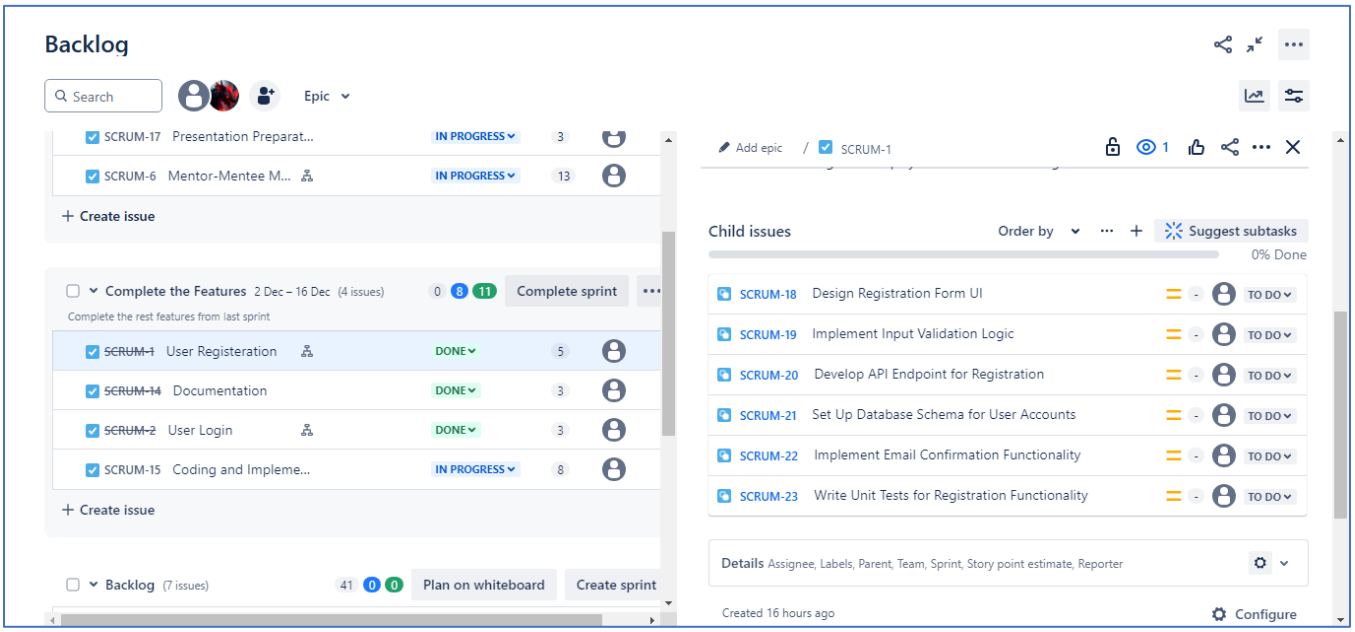
Backlog:



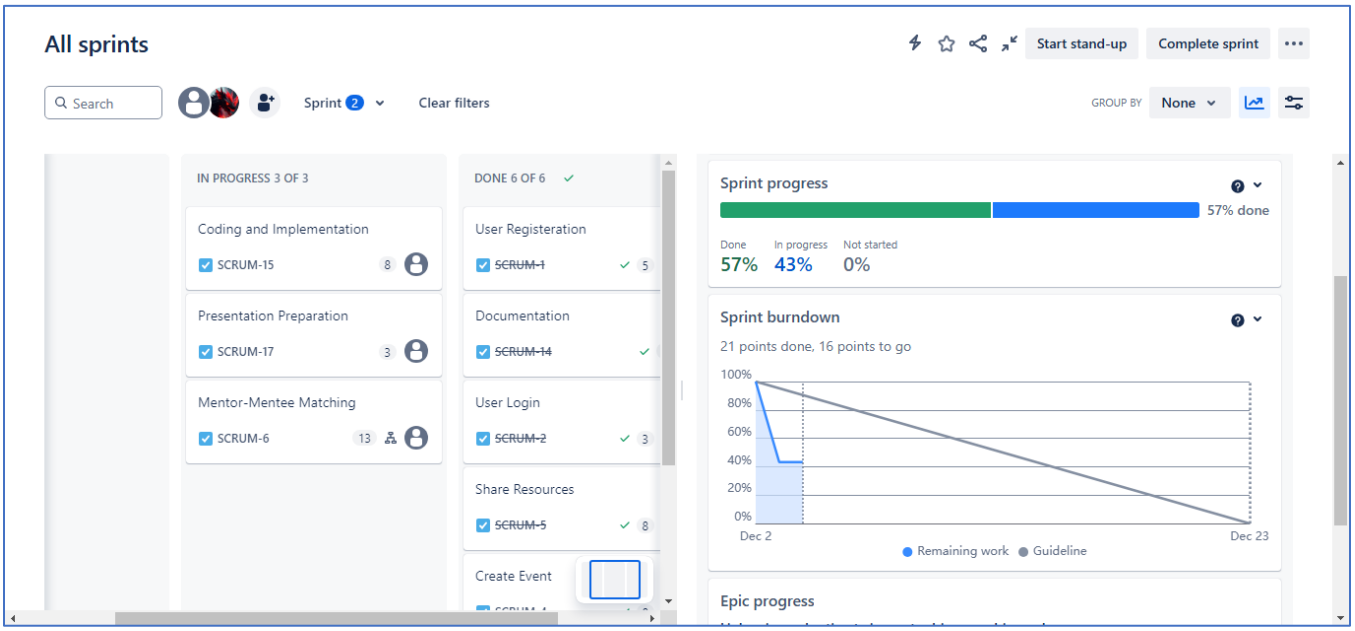
User Stories Details:



User Stories Sub-tasks:



Jira Board:



8.3 GitHub Repository:

Repository for this Project is available on GitHub at https://github.com/raghadhk/PMUMentor

9 Test

- **User Story: User Registration**
  - **Test Case 1: Successful User Registration**
    - **Given**: A new user provides valid input data.
    - **When:** The user submits the registration form.
    - **Then:** The system should create a new account and send a verification email.
  - **Test Case 2: Missing Required Fields**
    - **Given**: A user submits the registration form with missing required fields.
    - **When**: The form is submitted.
    - **Then**: The system should display an error message indicating the missing fields.
  - **Test Case 3: Email Already Registered**
    - **Given**: A user tries to register with an email that is already in use.
    - **When**: The form is submitted.
    - **Then**: The system should inform the user that the email is already registered.
- **User Story: Password Recovery**
  - **Test Case 4: Successful Password Recovery**
    - **Given**: A user requests a password reset using a valid email.
    - **When**: The user clicks the password reset link and sets a new password.
    - **Then**: The system should update the password and allow the user to log in with the new password.
  - **Test Case 5: Invalid Email for Password Reset**
    - **Given**: A user enters an invalid email during password recovery.
    - **When**: The user submits the request.
    - **Then**: The system should inform the user that the email is not registered.
- **User Story: Mentor-Mentee Matching**
  - **Test Case 6: Search for Mentor by Field of Study**
    - **Given**: A mentee searches for mentors by selecting a specific field of study.
    - **When**: The search is performed.

        - **Then**: The system should return a list of mentors matching the selected criteria.
    - **Test Case 7: Mentor Accepts Connection Request**
        - **Given**: A mentee sends a connection request to a mentor.
        - **When**: The mentor accepts the request.
        - **Then**: The mentee and mentor should be notified, and the connection established.

- **User Story: Event Creation**
    - **Test Case 8: Create a Virtual Event**
        - **Given**: A mentor provides valid event details.
        - **When**: The event is created.
        - **Then**: The system should display the event on the events calendar and send notifications to relevant users.
    - **Test Case 9: Maximum Participants Reached**
        - **Given**: A mentor sets a maximum participant limit for an event.
        - **When**: The limit is reached.
        - **Then**: The system should disable further registrations and inform users that the event is full.

- **User Story: Resource Sharing**
    - **Test Case 10: Upload a Study Guide**
        - **Given**: A mentor uploads a valid study guide file with a title and description.
        - **When**: The upload is completed.
        - **Then**: The system should add the resource to the library and make it searchable for mentees.

10 Conclusion

The **PmuMentor** app addresses a significant need within the PMU community by providing a central platform for mentoring, resource sharing, and community engagement. Through careful planning, agile project management, and a focus on user-centered design, the development team created a solution that enhances communication and collaboration among students.

By leveraging modern technologies such as Flutter for cross-platform mobile development and Firebase for back-end services, the app ensures scalability, reliability, and a seamless user experience. Core

functions such as user authentication, mentor-mentee matching, event management, and real-time communication have been carefully implemented to meet the diverse needs of users.

Adopting the Scrum framework facilitated efficient project execution, allowing the team to adapt to challenges and continually improve the product. Tools such as Jira and Confluence enhanced collaboration and transparency throughout the development process, ensuring that all team members were aligned with project goals.

In conclusion, PmuMentor not only achieves its initial goals, but also lays the foundation for future improvements and scalability. This app embodies a commitment to enriching the university experience by empowering students to connect, learn, and grow together. The app is a testament to the power of technology to enhance academic success and personal development within learning communities.

11 References

1. **draw.io** - Online Diagram Software & Flowchart Maker. *draw.io*. Available at: https://www.draw.io
2. **Figma** - Collaborative Interface Design Tool. *Figma Inc.* Available at: https://www.figma.com
3. **Jira Software** - Project Management Software. *Atlassian*. Available at: https://www.atlassian.com/software/jira
4. **Flutter Documentation** - Google Developers. *Flutter.dev*. Available at: https://flutter.dev/docs
5. **Firebase Documentation** - Guides and Reference Materials. *Firebase by Google*. Available at: https://firebase.google.com/docs
6. **Agile Project Management with Scrum -** Schwaber, K. (2004). Microsoft Press. ISBN: 978-0735619937
7. **Material Design Guidelines** - Principles and Best Practices. *Google Design*. Available at: https://material.io/design
8. **Prince Mohammad Bin Fahd University (PMU)** - Official University Website. Available at: https://www.pmu.edu.sa
9. **Atlassian Confluence** - Team Collaboration Software. *Atlassian*. Available at: https://www.atlassian.com/software/confluence