Kooroky Social Media Platform

- Mehdi Maleki | Systems Design Document | 2024

Table of Contents

Overview

Scenario

Stakeholders and Users

Methodology and Team Structure

Tech Stack (Used Tools)

Implementation of the Scrum Process on Jira

Future Goals

UML Diagrams

Conclusion

Overview

Kooroky is a **developer-centric social media platform** designed to address the unique needs of developers and tech enthusiasts. It provides a space for sharing code snippets, discussing technical topics, and fostering professional networking. The platform integrates **real-time updates** and **AI-powered chat assistance** to enhance user interaction and collaboration.

Scenario

The platform allows users to:

1. **Create posts** with text, images, and code snippets.
2. **Comment** on and **like** posts.
3. **Manage profiles** with social links and personal information.
4. Receive **real-time updates** on new posts and interactions.
5. Use an **AI chat assistant** for instant technical support and collaboration.

1. Purpose and Problem to Solve

The purpose of Kooroky is to provide a professional social networking platform where developers can:

- Share code snippets with syntax highlighting.
- Create technical posts.
- Network with other developers.
- Get real-time updates.
- Chat with AI assistance.
- Manage professional profiles with social links.

The main problem Kooroky aims to solve is the lack of a dedicated social platform for developers that integrates code sharing and professional networking in a seamless manner.

2. Project Details

1. **User Management**

    - **Authentication & Authorization**: Secure login and registration using JWT.
    - **Profile Management**: Users can create and update their profiles with personal and professional information.
    - **Social Links Integration**: Users can add links to their social media profiles (Twitter, LinkedIn, Instagram and Facebook.).
    - **Friend System**: Users can add and remove friends, view friends' profiles, and interact with their posts.

2. **Content Management**

    - **Post Creation/Editing**: Users can create and edit posts with text, images, and code snippets.
    - **Code Snippet Sharing**: Users can share code snippets with syntax highlighting.
    - **Syntax Highlighting**: Integrated with PrismJS for highlighting code in various programming languages.
    - **Image Upload**: Users can upload images to their posts.

3. **Real-time Features**

    - **Live Updates**: Real-time updates for posts, likes, and comments using WebSocket.
    - **Instant Messaging**: Real-time AI assistant chat functionality.

4. **Search & Feed**
    - **Search**: Users can search for posts and other users using a search bar. Also, they can select posts and users from search results.
    - **Feed**: Users can view a feed of posts from friends and explore posts from the community.

3. Project Limitations or Challenges

- **General Concerns:**

- **Scalability**: Ensuring the platform can handle a large number of users and real-time interactions.
- **Security**: Protecting user data and preventing unauthorized access.
- **Performance**: Optimizing the platform for fast loading times and smooth user experience.
- **Data Privacy**: Ensuring compliance with data protection regulations.

- **Major Concerns:**

- **Security**: Implementing robust authentication and authorization mechanisms, securing data transmission, and protecting against common vulnerabilities.
- **Speed**: Optimizing database queries, implementing caching, and using a CDN for static assets.
- **Requirements**: Ensuring the platform meets the needs of developers, including features like code sharing, real-time updates, and professional networking.

4. User Interaction with the System in Details

1. **Registration and Login**

    - The user visits the Kooroky website and registers by providing their email, password, and other required information.
    - The user receives a confirmation email and verifies their account. (In real scenario)
    - The user logs in using their email and password.

2. **Profile Setup**

    - The user sets up their profile by adding personal and professional information, including social links and a profile picture.
    - The user can also add a bio, specify their location, occupation and unique ID.

3. **Creating and Sharing Posts**

- The user creates a new post by entering a description, uploading an image, and adding a code snippet (support 4 programming language) with syntax highlighting.
- The user shares the post, which appears in their feed and the feeds of their friends.

4. **Interacting with Posts**

   - The user can like, comment on, and share posts from other users.

5. **Searching and Networking**

   - The user searches for other developers and posts using the search bar.
   - The user sends friend requests to connect with other developers.
   - The user can view and interact with the profiles and posts of their friends OR global posts of all users.

6. **Real-time Chat (AI Assistance)**

   - The user interacts with the AI chat assistant for coding queries and other assistances.

Stakeholders and Users

5. Stakeholders

1. **Project Manager**: Oversees project progress and ensures goals are met. (Hypothetical)
2. **Client**: Provides requirements and feedback to align the platform with business objectives.
3. **Development Team**: Implements the features and functionalities of the platform. (Hypothetical)
4. **QA Team**: Ensures the quality of the product by testing for bugs, performance issues, and usability problems. (Family Members in my case)

6. Interests of Stakeholders

1. **Project Manager**: Focuses on delivering the project on time and within budget.
2. **Client**: Aims to meet business goals and ensure user satisfaction.

3. **Development Team**: Strives for an efficient and smooth development process.
4. **QA Team**: Ensures the product is high-quality and free of defects.

7. End Users

1. **Target Users**: Software developers, IT professionals, computer science students (especially QIAU students), and tech enthusiasts.
2. **Demographics**: Primarily young professionals and students who are tech-savvy and active in the developer community.
3. **User Needs**: Easy code sharing, real-time collaboration, professional networking, and AI assistance.

8. Beneficiaries' Desires

To better understand user needs, I conducted interviews with potential users:

1. **Mehdi Maleki (Myself)**:
2. **Question**: What features would make a social media platform most useful for you as a developer?
3. **My Mentality**: "I love a platform that supports **syntax highlighting** for code snippets and allows **real-time collaboration** on technical posts. An AI assistant for debugging would be a game-changer."
4. **Pouria Omrani (C# Developer – from QIAU)**:
5. **Question**: What challenges do you face with existing platforms when sharing code?
6. **Answer**: "Most platforms don't support **C# syntax highlighting** well, and I'd like to see **version control integration** for shared code snippets."
7. **Hossein GolMohammadi (Scrum Master - from ALPHA CO.**):
8. **Question**: How can a social media platform support agile teams?
9. **Answer**: "A platform with **real-time updates** and **AI-powered task management** would help teams stay aligned and productive."
10. **Simin Badri (C++ Developer - from ALPHA CO.**):
11. **Question**: What features would you prioritize in a developer-focused platform?
12. **Answer**: "I like **C++ syntax highlighting**, **offline access** to posts, and a **community-driven Q&A section**."

9. Practical Example of Project Implementation

1. **User Registration and Login**

    - The user visits the Kooroky website and clicks on the "Register" button.
    - The user fills out the registration form with their first name, last name, email, password, location, and occupation.
    - The user uploads a profile picture and clicks "Submit."
    - The user receives a confirmation email and clicks on the verification link. (In applied real server.)
    - The user logs in using their email and password.

2. **Creating a Post**

    - The user navigates to the homepage and clicks on the "Create Post" button.
    - The user enters a description of the post, uploads an image, and adds a code snippet (for example, a simple JavaScript program) with syntax highlighting.
    - The user clicks "Post," and the post appears in their feed and the feeds of their friends.

3. **Interacting with Posts**

    - The user scrolls through their feed and likes a post by clicking the "Like" button.
    - The user comments on a post by typing a comment and clicking "Submit."
    - The user shares a post by clicking the "Share" button and selecting the friends to share it with.

4. **Searching for Users and Posts**

    - The user types a search query into the search bar and clicks "Search."
    - The user views the search results, which include users and posts matching the query.
    - The user clicks on a user's profile to view their posts and send a friend request.

5. **Real-time Chat and AI Assistance**

- The user clicks on the chat icon to open the chat window.
- The user asks the AI chat assistant a coding question and receives an instant response. (In any language)

Methodology and Team Structure

10. Methodology

For my project, which involves developing a social media platform with real-time features, user management, and content management, the Scrum methodology would be a suitable choice. Here are the reasons why Scrum is beneficial for my project:

1. **Iterative Development**: Scrum allows for iterative development, which means you can develop the platform in small, manageable increments. This is particularly useful for complex projects like a social media platform, where requirements may evolve over time.

2. **Flexibility and Adaptability**: Scrum is designed to be flexible and adaptable to changing requirements. As you gather feedback from users and stakeholders, you can easily adjust your priorities and development focus.

3. **Regular Feedback**: Scrum emphasizes regular feedback through sprint reviews and retrospectives. This ensures that you are continuously improving the product and addressing any issues promptly.

4. **Transparency and Communication**: Daily standups and sprint planning meetings promote transparency and communication within the team. This helps in identifying and resolving blockers quickly, ensuring smooth progress.

5. **Focus on Delivering Value**: Scrum prioritizes delivering working features at the end of each sprint. This ensures that you are consistently providing value to your users and stakeholders.

6. **Team Collaboration**: Scrum encourages collaboration among team members, fostering a sense of ownership and accountability. This is crucial for a project that requires coordination between frontend developers, backend developers, UI/UX

By adopting Scrum, I can ensure that my project remains on track, adapts to changes effectively, and delivers a high-quality social media platform that meets the needs of my users.

So, **Scrum** is chosen for its **iterative development**, **flexibility**, and **continuous feedback**.

11. Team Structure

1. **Core Team**: Product Owner, Scrum Master, Frontend Developers, Backend Developers, UI/UX Designers, QA Engineers. (Hypothetical)
2. **Extended Team**: DevOps Engineer, Database Administrator, Security Specialist. (Hypothetical)

12. Team Interaction

1. **Daily Standups**: Short meetings to discuss progress and blockers.
2. **Sprint Planning**: Define backlog items and prioritize features.
3. **Sprint Review**: Demonstrate completed features to stakeholders.
4. **Sprint Retrospective**: Reflect on the sprint and identify improvements.

13. Systems Design Practices

1. **Scalability**: The platform is designed to handle a growing number of users and data. Techniques like **horizontal scaling** (adding more servers) and **vertical scaling** (upgrading server resources) are considered.
2. **Modularity**: The system is divided into independent modules (e.g., frontend (client), backend (server and mongo database), real-time features, AI chat) to ensure flexibility and ease of maintenance.
3. **Performance Optimization**: Techniques like **caching**, **lazy loading**, and **optimized database queries** are used to ensure fast response times.
4. **Security**: **JWT Authentication** and **secure data storage** practices are implemented to protect user data.
5. **Real-time Communication**: **WebSocket** is used for real-time updates, ensuring users receive live notifications without performance degradation.

Tech Stack

14. Core Components

1. **Frontend:**

- **React.js**: For building the user interface.
- **Redux**: For state management.
- **Material-UI**: For responsive and visually appealing design.
- **PrismJS**: For syntax highlighting in code snippets.

2. **Backend**:

- **Node.js**: For server-side development.
- **Express.js**: For building RESTful APIs.
- **MongoDB**: For data storage.
- **JWT Authentication**: For secure user authentication.

3. **Real-time Features**:

- **WebSocket**: For live updates and notifications.

4. **AI Integration**:

- **Groq API**: For AI-powered chat assistance.

15. Algorithms

1. **Search Algorithms**: For finding posts and users efficiently => pattern matching algorithm using regular expression.
2. **Real-time Update Algorithms**: Using WebSocket's Best Practices for live notifications.
3. **AI Chat Integration**: Using Groq API and traditional API Calls for natural language processing.

16. Time and Date Handling

1. **Moment.js**: For formatting dates and times.

17. Data Handling

1. **File Formats**: JSON for API communication, images for user profiles and posts.
2. **Database Integration**: MongoDB with Mongoose for data persistence.

18. Performance Optimization

1. **Lazy Loading**: For images and posts to reduce initial load time.
2. **Caching**: For frequently accessed data to improve speed.
3. **Optimized Database Queries**: To ensure fast data retrieval.
4. **CDN**: For delivering static assets like images and CSS files.

19. Development Tools

1. **Build Systems**: Webpack for bundling frontend assets.
2. **Documentation**: Swagger for API documentation, user guides, and developer documentation.
3. **Version Control**: Git for source code management, GitHub for repository hosting.

Implementation of the Scrum Process on Jira

20. Configure Board: Scrum board with columns for To Do, In Progress, and Done.

21. Defining the Backlog

1. **Task List**: User stories, tasks, and bugs.
2. **Details**: Descriptions, acceptance criteria, and estimates.

22. Setting Priorities

1. **High Priority**: Critical features and urgent bugs.
2. **Low Priority**: Minor enhancements and non-urgent tasks.

23. Explaining Sprints

1. **Sprint Planning**: Tasks from the backlog for the sprint.
2. **Timeframes**: Typically, 2-week sprints.

24. Reviewing Completed Tasks

1. **Sprint Review**: Demonstrate completed tasks.
2. **Progress Reporting**: Using Jira reports to track sprint progress and team performance.

25. Practical Implementation on Jira

Future Goals

26. Notifications

1. **Real-time Notifications:** Implement real-time notifications for likes, comments, and new followers.
2. **Email Notifications:** Send email notifications for important updates and activities.

27. Files

1. **More File Types**: Allow users to upload audio, video and documents.

28. Direct Chat

1. **User-to-User Chat:** Enable direct messaging between users.
2. **Group Chat:** Allow users to create and join group chats for discussions.

29. Enhanced AI Assistance

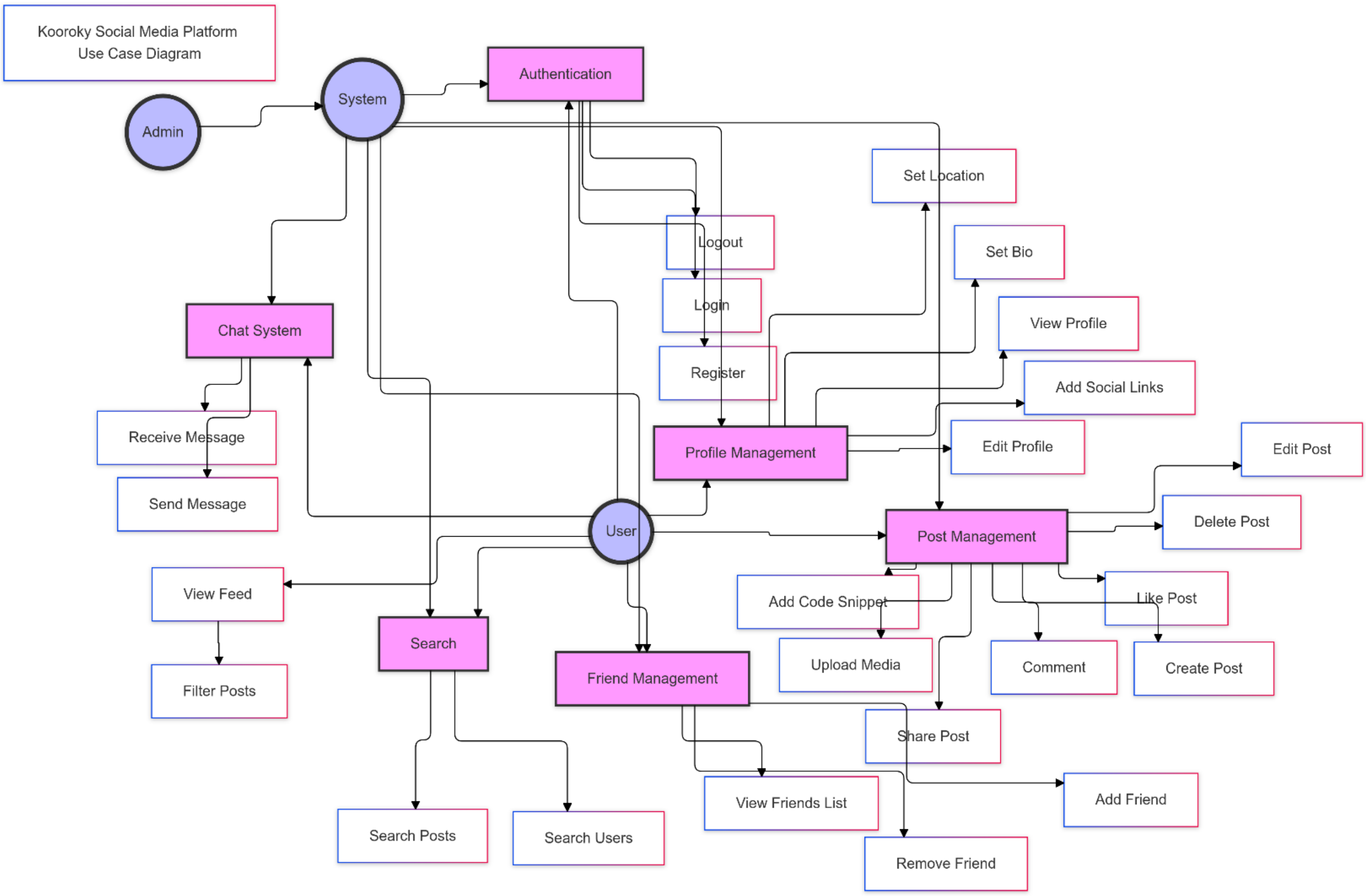
1. **Advanced AI Chat:** Improve the AI chat assistant to provide more accurate and context-aware responses and write code to posts.
2. **Code Review Assistance:** Integrate AI-powered code review and suggestions.

30. Additional Features

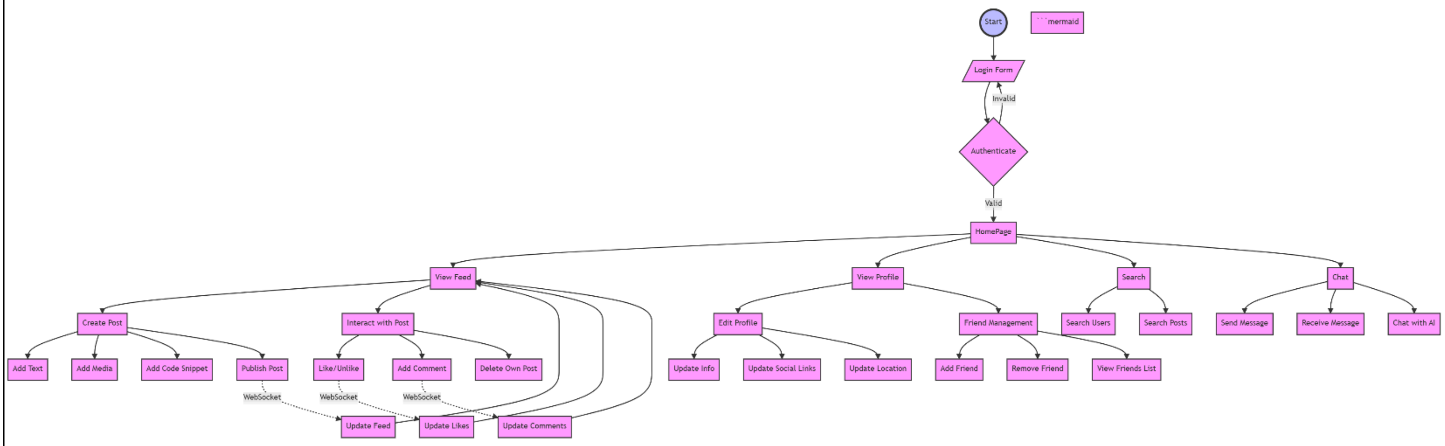
1. **Post Scheduling:** Allow users to schedule posts for future publication.
2. **Analytics Dashboard:** Provide users with analytics on their posts' performance and engagement.
3. **Customizable Themes:** Enable users to customize the appearance of their profiles and posts.

UML Diagrams

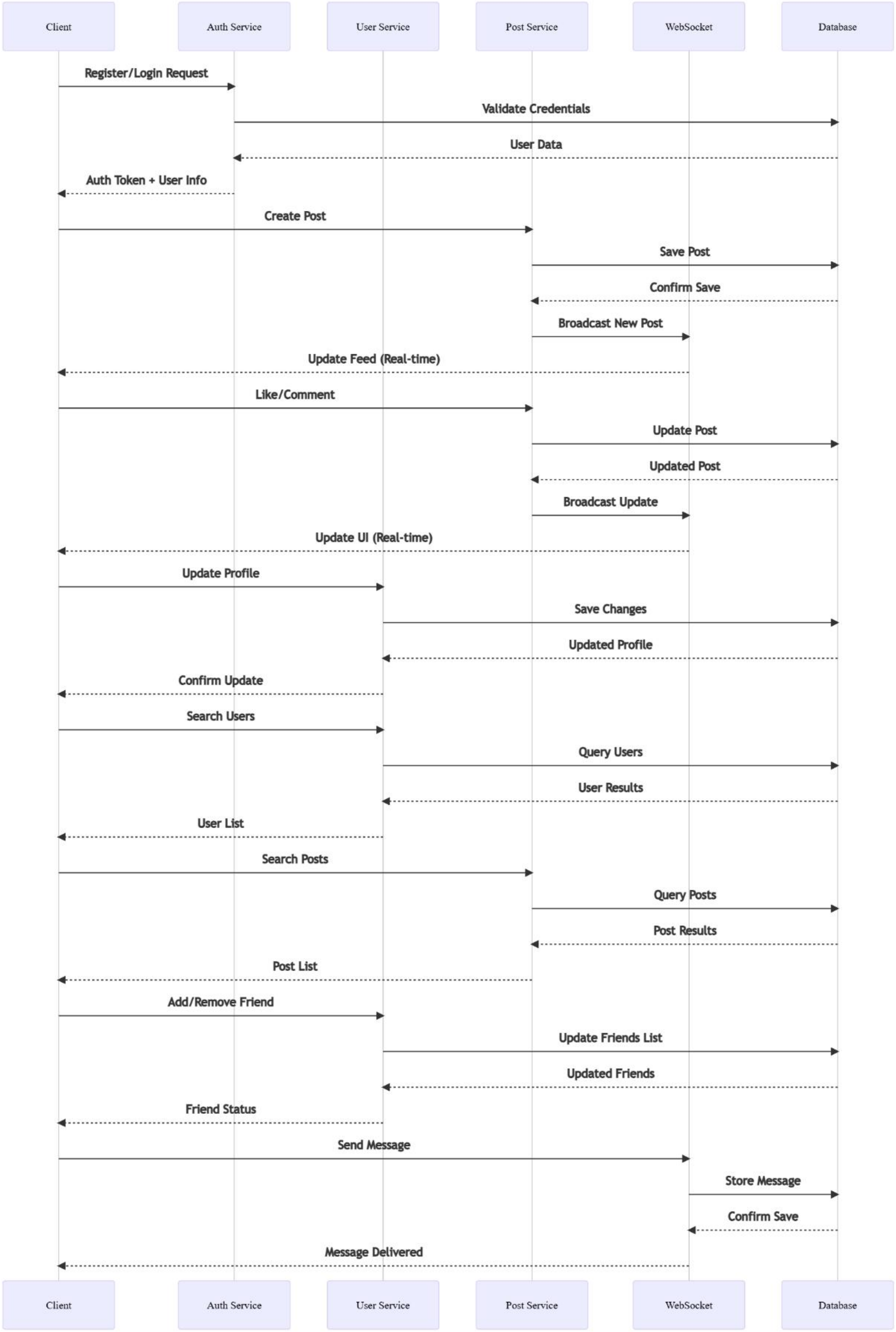
31. Use Case Diagram



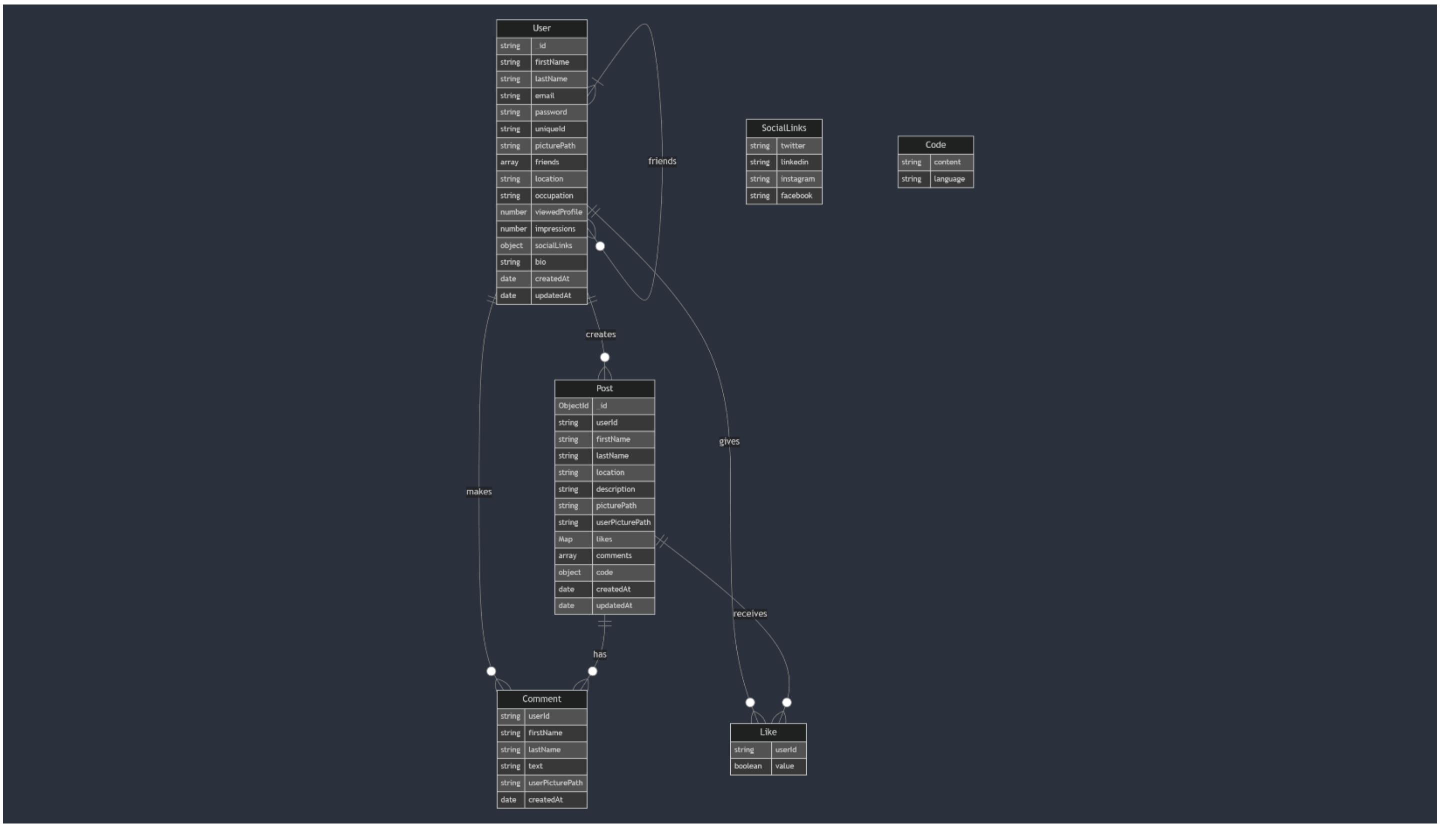
32. Activity Diagram



33. Swimlane Diagram


Client
Auth Service
User Service
Post Service
WebSocket
Database
Register/Login Request
Validate Credentials
User Data
Auth Token + User Info
Create Post
Save Post
Confirm Save
Broadcast New Post
Update Feed (Real-time)
Like/Comment
Update Post
Updated Post
Broadcast Update
Update UI (Real-time)
Update Profile
Save Changes
Updated Profile
Confirm Update
Search Users
Query Users
User Results
User List
Search Posts
Query Posts
Post Results
Post List
Add/Remove Friend
Update Friends List
Updated Friends
Friend Status
Send Message
Store Message
Confirm Save
Message Delivered
Client
Auth Service
User Service
Post Service
WebSocket
Database

33. ER Diagram



Conclusion

By following this structured approach and incorporating **systems design practices**, the **Kooroky** Social Media Platform aims to deliver a **high-quality, developer-focused social networking experience**. The platform addresses the unique needs of developers, fosters collaboration, and integrates advanced features like real-time updates and AI-powered assistance. By gathering feedback from beneficiaries like Mehdi Maleki (Me), Pouria Omrani, Hossein GolMohammadi, and Simin Badri, the platform ensures it meets the desires and expectations of its target users.