## **Kooroky Social Media Platform**

* Mehdi Maleki | Systems Design Document | 2024

## **Table of Contents**

### ****Scenario****

### **Detailed Description**

### **Systems Design Practices**

### ****Stakeholders and Users****

### **Stakeholders**

### **Interests of Stakeholders**

### **End Users**

### **Beneficiaries' Desires**

### ****Methodology and Team Structure****

### **Methodology**

### **Team Structure**

### **Team Interaction**

### ****Tech Stack (Used Tools)****

### **Core Components**

### **Algorithms**

### **Time and Date Handling**

### **Data Handling**

### **Performance Optimization**

### **Development Tools**

### ****Implementation of the Scrum Process on Jira****

### **Creating a Scrum Project in Jira**

### **Defining the Backlog**

### **Setting Priorities**

### **Explaining Sprints**

### **Reviewing Completed Tasks**

### ****Future Goals****

### ****Conclusion****

## **Scenario**

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.

## **Detailed Description**

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.

## **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, backend, 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.

## **Stakeholders and Users**

##### ****Stakeholders****

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

##### ****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.

##### ****End Users****

1. **Target Users**: Software developers, IT professionals, computer science 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.

## **Beneficiaries' Desires**

To better understand user needs, we 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. **Answer**: "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)**:
5. **Question**: What challenges do you face with existing platforms when sharing code?
6. **Answer**: "Most platforms donâ€™t support **C# syntax highlighting** well, and Iâ€™d like to see **version control integration** for shared code snippets."
7. **Hossein GolMohammadi (Scrum Master)**:
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)**:
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**."

## **Methodology and Team Structure**

##### ****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**.

##### ****Team Structure****

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

##### ****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.

## **Tech Stack**

##### ****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.

1. **Backend**:

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

1. **Real-time Features**:

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

1. **AI Integration**:

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

##### ****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.

##### ****Time and Date Handling****

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

##### ****Data Handling****

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

##### ****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.

##### ****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**

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

##### ****Defining the Backlog****

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

##### ****Setting Priorities****

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

##### ****Explaining Sprints****

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

##### ****Reviewing Completed Tasks****

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

## Future Goals

**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.

**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.

**Enhanced AI Assistance**

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

**Mobile Application**

1. **Mobile App Development:** Develop a mobile application for iOS and Android to provide a seamless experience on mobile devices.

**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.

## **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, Pouria Omrani, Hossein GolMohammadi, and Simin Badri, the platform ensures it meets the desires and expectations of its target users.