Aim : Do requirement analysis and develop software requirement specific sheet for you specific area

**Video Sharing App System Requirements Specification**

**1. Introduction**

**1.1 Purpose**

The System Requirements Specification (SRS) for the video sharing app serves multiple essential purposes in the software development process. Firstly, it defines the project's objectives in detail, clarifying what the app aims to achieve, such as enabling users to share and interact with video content and fostering a community of users. This clear articulation of objectives ensures that all project stakeholders share a common understanding of the app's purpose, thereby reducing misunderstandings.

Secondly, the SRS establishes a common vision among cross-functional teams. In complex projects, alignment among developers, designers, testers, and project managers is paramount. The SRS acts as a unifying document that outlines functionality and design specifications, minimizing miscommunication and ensuring everyone is on the same page.

Thirdly, it serves as a development guidebook by providing detailed specifications for each app feature and functionality. Developers rely on the SRS to build the software systematically, meeting intended goals and user expectations.Moreover, the SRS fosters collaboration. It creates a common reference point for discussions and decisions, allowing stakeholders to engage in informed conversations about design, functionality, and technical requirements, promoting a collaborative and productive development environment.Furthermore, it aids in managing expectations by clearly outlining the app's scope, preventing scope creep, and unrealistic expectations. This proactive approach to expectation management contributes to a smoother development process and minimizes conflicts.

Lastly, the SRS supports quality assurance and decision-making. It helps QA teams create test plans to ensure the app functions correctly according to requirements and facilitates informed decisions on design, features, and technical implementations.

In essence, the SRS is a foundational document that not only guides development but also fosters collaboration, manages expectations, supports quality assurance, and aids decision-making. Its importance lies in its ability to ensure that the video sharing app meets its objectives and delivers value to its users..

**1.2 Scope**

The video sharing app will:

* Enable user registration and authentication, supporting email, password, and third-party options.
* Allow users to upload, categorize, tag, and manage videos with customizable details.
* Facilitate video viewing, interaction (likes, comments, sharing), and reporting of inappropriate content.
* Provide user profile pages displaying uploaded videos, followers, and followings.
* Offer powerful search and video recommendation features.
* Deliver real-time notifications for user activity.
* Be accessible on web browsers, iOS, and Android devices.
* Support multiple languages and regions for a global user base.

In essence, the app aims to provide a versatile, engaging, and cross-platform video-sharing experience for users, ensuring international accessibility and user interaction..

**1.3 Definitions, Acronyms, and Abbreviations**

* **User:** An individual who accesses and uses the video sharing app.
* **Video:** Multimedia content in the form of recorded videos and live stream
* **SRS:** System Requirements Specification.
* **Live Stream:** the live feed of the video from the users device

**2. System Overview**

**2.1 System Description**

The video sharing app will provide users with the following features:

* User registration and authentication on the platform .
* Video upload and sharing capabilities.
* Video live Stream capabilities .
* Auto video quality scaling according to the network quality .
* Video categorization and tagging using classification.
* User profile management.
* Video searching and recommendation .
* Social interaction features (liking, commenting, sharing, liking ,watch count etc ).
* Notification to the user using image generation

**2.2 System Architecture**

Expanding on the system architecture of the video sharing app provides a more detailed understanding of how the various components work together to deliver its functionalities:

**2.2 System Architecture**

The system architecture of the video sharing app comprises several interconnected components, each with specific roles and responsibilities:

**2.2.1 Client-Side**

**1. Web Application**: The web application serves as the primary interface for users accessing the app from desktop and mobile web browsers. It provides the user interface (UI) for video browsing, uploading, and interaction.

**2. Mobile Applications:** Native mobile applications for iOS and Android platforms offer a consistent and optimized user experience. These apps are designed to leverage device-specific features and ensure smooth performance.

**3. User Devices:** The video sharing app is accessible on a variety of user devices, including smartphones, tablets, laptops, and desktop computers. It is responsive and compatible with different screen sizes and resolutions.

**2.2.2 Server-Side**

**4. Backend Server:** The backend server is the central component responsible for processing user requests, managing user accounts, and storing video metadata and user-generated content. It handles user authentication, video uploads, video recommendations, and notifications.

**5. Database:** A robust and scalable database system stores user data, video information, user interactions (likes, comments, follows), and system logs. This data is crucial for content management, analytics, and user recommendations.

**6. Content Delivery Network (CDN):** The CDN plays a vital role in optimizing video delivery. It caches and serves video content to users worldwide, reducing latency and ensuring smooth video streaming regardless of geographical location.

**2.2.3 Third-Party Services**

**7. Authentication Providers:** Third-party authentication services like Google and Facebook are integrated to provide users with convenient and secure login options.

**8. Push Notification Service:** A push notification service facilitates the delivery of real-time notifications to users' devices, enhancing user engagement.

**2.2.4 Application Programming Interfaces (APIs)**

**9. API Layer:** To enable communication between the client-side and server-side components, a set of APIs is implemented. These APIs facilitate data exchange, user interactions, and video content retrieval.

The system architecture is designed to be scalable and resilient to accommodate high user traffic and ensure minimal downtime. It also emphasizes data security, efficient video streaming, and a responsive user experience across various devices and platforms. Collaboration between the client-side, server-side, and third-party services is essential for the seamless operation of the video sharing app.

**3. Functional Requirements**

**3.1 User Registration and Authentication**

* Registration of the User into the database
* Allowing the user to register using the other accounts such as Facebook , Google etc

**3.2 Video Upload and Management**

* Users can upload videos.
* User can livestream video
* Categorization and Tagging of videos
* Users can edit video details (title, description, privacy settings).
* Videos can be deleted or made private/public by the user.

**3.3 Video Viewing and Interaction**

* Users can view videos.
* Users can like, comment on, and share videos.
* Videos can be reported for inappropriate content.
* Videos can be embedded on external websites.

**3.4 User Profile**

* Users can edit their profiles (avatar, username, bio).
* Users can follow other users.
* Users can view their own and others' uploaded videos.

**3.5 Video Discovery**

* Users can search for videos by keywords, categories, or tags.
* The app provides video recommendations based on user activity.

**3.6 Notifications**

* Users receive notifications for likes, comments, and new followers.
* Users can manage notification preferences.

**4. Non-Functional Requirements**

**4.1 Performance**

* The app must support a high volume of concurrent users.
* Videos should load quickly and without buffering.
* The system should have a response time of under 2 seconds for most user actions.

**4.2 Security**

* User data must be securely stored and transmitted.
* Authentication and authorization mechanisms must be robust.
* Inappropriate content should be filtered and moderated.

**4.3 Scalability**

* The system should be designed to scale horizontally to accommodate increased user traffic.

**4.4 Usability**

* The user interface should be intuitive and user-friendly.
* The app should be accessible to users with disabilities.

**4.5 Compatibility**

* The app should work on a variety of web browsers and mobile devices.

**5. Constraints**

* Development must adhere to budget and time constraints.
* The app must comply with relevant data privacy and copyright laws.

**6. Assumptions and Dependencies**

* The app assumes users have internet connectivity.
* Third-party services (e.g., cloud hosting, CDN) are available and reliable.

**7. Appendices**

Include any additional information, diagrams, or mockups that support the SRS.

**8. Revision History**

Document revision history, including dates and changes made.

This template provides a structured framework for creating an SRS for your video sharing app. You can further tailor it to your specific project's needs by adding more details or custom requirements as necessary. Collaboration with your development team and stakeholders is crucial to ensure that the SRS accurately represents the project's objectives and constraints.

Top of Form