**Software Requirements Specification (SRS) for Artistry Hub App**

**1. Introduction**

**1.1 Purpose**

This document outlines the Software Requirements Specification (SRS) for the **Freelancer Marketplace App**, which connects artists (both technical and non-technical) with customers who are interested in aesthetic and handcrafted products. The app will allow artists to showcase their talents, manage bookings, and organize events. Customers will be able to view artists' profiles, book services, and attend events.

**1.2 Scope**

The application is a platform for artists and customers to interact with features including user registration, profile management, service management, booking, time slot management, event management, and real-time notifications. The platform will also support both in-person and virtual events.

**1.3 Definitions, Acronyms, and Abbreviations**

* **Artist**: A person offering artistic services (e.g., mehendi artist, canvas painter).
* **Customer**: A person purchasing services or attending events.
* **Admin**: A user with administrative privileges to manage platform content and users.
* **SRS**: Software Requirements Specification.

**1.4 References**

Any external resources, standards, or frameworks referenced in this document.

**2. Overall Description**

**2.1 Product Perspective**

The **Freelancer Marketplace App** serves as a platform connecting artists and customers. The app supports features like profile creation, service management, booking, event management, and feedback. It will be built using **Spring Boot (Java)** for backend services, **ReactJS** for the frontend, and **MySQL** for database storage.

**2.2 Product Features**

* **User Registration**: Artists and customers can register using social logins (Google, Facebook) or email.
* **Service Management**: Artists can list their services with detailed descriptions, pricing, and availability.
* **Time Slot Management**: Artists can define their working hours and time slots.
* **Event Management**: Artists can organize and promote events.
* **Review System**: Customers can review services and events.
* **Real-Time Notifications**: Users will receive updates for bookings, cancellations, and event reminders.
* **Admin Dashboard**: Admins will manage users, events, complaints, and featured content.

**2.3 User Classes and Characteristics**

* **Artist**: Creates and manages profiles, services, availability, and events.
* **Customer**: Browses services, books appointments, and attends events.
* **Admin**: Manages platform content, user accounts, and events.

**2.4 Operating Environment**

* **Frontend**: ReactJS-based web application.
* **Backend**: Spring Boot-based REST API.
* **Database**: MySQL.

**3. System Features**

**3.1 User Registration and Profile Management**

* **Description**: Users will register and manage their profiles.
* **Functional Requirements**:
  + Social login (Google, Facebook) or email/password authentication.
  + Profile creation with details for artists (bio, portfolio, services, and availability).
  + Profile updates by users.

**3.2 Service Management**

* **Description**: Artists can create, list, and manage their services.
* **Functional Requirements**:
  + Artists can create services with descriptions, prices, and durations.
  + Services can be edited or removed.

**3.3 Availability and Time Slot Management**

* **Description**: Artists define available time slots for booking.
* **Functional Requirements**:
  + Artists can set available hours (e.g., 10 AM - 6 PM).
  + Time slot configurations (e.g., max clients per slot, slot duration).
  + Customers can book services based on available time slots.

**3.4 Event Management (Buzz Tab)**

* **Description**: Artists can organize events and exhibitions.
* **Functional Requirements**:
  + Artists can create events with details (name, date, location, tickets).
  + Customers can view event details and book tickets.
  + Artists and customers can leave event reviews.

**3.5 Notification System**

* **Description**: Users will receive real-time notifications for various actions.
* **Functional Requirements**:
  + Booking confirmation, cancellations, event reminders, and feedback notifications.

**4. External Interface Requirements**

**4.1 User Interfaces**

* **Frontend**: Web-based user interface developed in **ReactJS**.
  + The homepage will display featured artists and services.
  + Artists will have a dedicated profile page with their services and availability.
  + Admins will have a dashboard to manage users, services, events, and complaints.

**4.2 Hardware Interfaces**

* The app will be hosted on cloud servers such as AWS or Google Cloud, with access via modern web browsers.

**4.3 Software Interfaces**

* **Frontend**: ReactJS, HTML, CSS, JavaScript.
* **Backend**: Spring Boot, REST API for communication.
* **Database**: MySQL for data storage.

**5. System Attributes**

**5.1 Performance Requirements**

* Response times for user actions (booking, searching) should be within 2-3 seconds.
* The app should handle up to 1,000 concurrent users.

**5.2 Security Requirements**

* **Authentication**: Email/password or social login (Google, Facebook).
* **Data Encryption**: All sensitive data, including passwords, will be encrypted.
* **Authorization**: Admins will have access to user management and event approval.

**6. Other Requirements**

**6.1 Backup and Recovery**

* Regular database backups to be stored securely.
* Recovery from system failures will take no more than 24 hours.

**6.2 Legal and Regulatory Requirements**

* The app will comply with **GDPR** guidelines for user data protection.
* Payment transactions will be secured through services like **Stripe** or **PayPal**.

**7. Appendices**

**7.1 Glossary of Terms**

* Definitions of terms such as **Artist**, **Customer**, **Admin**, **Service**, **Event**.

**7.2 References**

* External tools or resources used to develop this SRS document (e.g., **Lucidchart** for diagrams, **MySQL Workbench** for schema design).

**Document Version**

* **Version 1.0**
* **Date**: [Insert Date]