**Software Requirements Specification (SRS)**

**Stranger Connection**

**1. Introduction**

**1.1 Purpose**

This document outlines the functional, non-functional, and technical requirements for the "Stranger Connection" application. Stranger Connection is a mobile application designed for anonymous communication with strangers, providing video calling functionality through PeerJS and WebRTC, along with user authentication, a points system, and Firebase integration for backend services.

**1.2 Scope**

Stranger Connection enables:

1. User authentication via Google Sign-In using Firebase Authentication.
2. Random matching of users for anonymous text or video communication.
3. Video calls powered by PeerJS and WebRTC.
4. A points system where:
   * Each new user starts with 100 points.
   * Each video call costs 5 points.
   * Points can be recharged by watching ads.
5. Data storage and synchronization via Firebase Realtime Database.
6. Tracking of user interactions and history.

This application targets Android users seeking random anonymous communication, with potential future scalability to iOS and other platforms.

**1.3 Definitions, Acronyms, and Abbreviations**

* **Firebase**: Backend-as-a-Service platform for database and authentication.
* **PeerJS**: A simple WebRTC library for creating peer-to-peer connections.
* **WebRTC**: Web Real-Time Communication for audio, video, and data exchange.
* **Points System**: Virtual currency mechanism to regulate calls.
* **Google AdMob**: Service for displaying ads and generating revenue.

**2. Overall Description**

**2.1 Product Perspective**

Stranger Connection integrates multiple technologies to provide a seamless experience:

1. Firebase for user authentication, data storage, and synchronization.
2. PeerJS and WebRTC for video calling.
3. Google AdMob for ad-based point recharging.

**2.2 Product Features**

1. **User Authentication**:
   * Login via Google Sign-In using Firebase Authentication.
   * Store user profiles in Firebase Realtime Database.
2. **Random Matching**:
   * Randomly pair users for text or video communication.
   * Provide anonymity during communication.
3. **Video Calling**:
   * Enable video calls using WebRTC and PeerJS.
   * Dynamically establish peer-to-peer connections.
4. **Points System**:
   * Deduct 5 points for each call.
   * Prevent calls when points are insufficient.
   * Allow users to recharge points by watching ads.
5. **Ad Integration**:
   * Earn points by watching ads integrated via Google AdMob.
6. **Data Management**:
   * Store user profiles, points, and call history in Firebase Realtime Database.

**2.3 User Characteristics**

* Users require an Android device running API level 24 (Android 7.0) or higher.
* Basic familiarity with app navigation and Google Sign-In.

**2.4 Constraints**

* **Internet Dependency**: Stable internet is essential for authentication, video calling, and Firebase access.
* **Platform**: Initially limited to Android devices.
* **Privacy**: No personally identifiable information is shared during communication.

**3. Functional Requirements**

**3.1 Authentication**

* Log in using Google Sign-In via Firebase Authentication.
* Store user profiles in Firebase Realtime Database.

**3.2 Video Calling**

* Establish peer-to-peer video connections using PeerJS and WebRTC.
* Provide fallback for disconnection or poor connection quality.

**3.3 Random Matching**

* Match users randomly for text or video communication.
* Maintain anonymity for both participants.

**3.4 Points System**

* Start new users with 100 points.
* Deduct 5 points for every call initiated.
* Prevent calls when the user has insufficient points.

**3.5 Ad-Based Recharge**

* Allow users to earn points (e.g., 10 points per ad) by watching Google AdMob ads.

**3.6 Data Storage**

* Use Firebase Realtime Database to store:
  + User profiles
  + Call history
  + Points balance

**4. Non-Functional Requirements**

**4.1 Performance**

* Video calls should establish within 2 seconds.
* Firebase Realtime Database operations should have low latency (<1 second).

**4.2 Security**

* Secure authentication via Firebase Authentication.
* Enforce end-to-end encryption for video calls using WebRTC.

**4.3 Usability**

* Simple and intuitive UI with clear navigation.
* Provide detailed error messages for failed actions (e.g., insufficient points).

**4.4 Scalability**

* Support up to 10,000 concurrent users.
* Optimize PeerJS signaling server for scaling.

**5. System Architecture**

**5.1 Overview**

* **Client**: Android app implemented using the following:
  + **Firebase Authentication**: For user login and session management.
  + **Firebase Realtime Database**: For storing user data.
  + **PeerJS and WebRTC**: For establishing peer-to-peer video connections.
  + **Google AdMob**: For serving ads and point recharging.

**5.2 Modules**

1. **Authentication Module**:
   * Integrate Google Sign-In via Firebase Authentication.
   * Store user profiles securely in Firebase Realtime Database.
2. **Matchmaking Module**:
   * Randomly connect users for text or video communication.
3. **Video Calling Module**:
   * Use PeerJS to establish WebRTC connections.
   * Ensure secure and low-latency video calling.
4. **Points Management Module**:
   * Deduct points for calls.
   * Prevent actions if points are insufficient.
5. **Ads Integration Module**:
   * Use Google AdMob to serve ads.
   * Reward points for each ad watched.

**6. External Interface Requirements**

**6.1 User Interfaces**

* **Onboarding Screen:** It is a starting screen where user will start the application**.**
* **Login Screen**: Google Sign-In button.
* **Home Screen**:
  + Display available points.
  + Option to request points by watching ads.
  + Buttons for "Start Call" and "Video Call".
* **Video Call Screen**:
  + Peer-to-peer video connection with mute/unmute and disconnect options.
* **Ads Screen:** Here user can watch online advertisement to earn points which is used for calling.
* **Waiting Screen:**

Users waits for other users to connect with each other, so that they can talk in call.

**6.2 Hardware Interfaces**

* Android device with:
  + Camera
  + Microphone
  + Stable internet connection

**6.3 Software Interfaces**

* Google Sign-In API
* Firebase Authentication
* Firebase Realtime Database
* PeerJS WebRTC integration
* Google AdMob SDK

**8. Future Enhancements**

* Add user reporting and blocking features.
* Introduce text-only chat rooms.
* Expand to iOS and web platforms.