**Software Requirements Specification (SRS)**

**For SkillSwap Mobile Application**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the functional and non-functional requirements of the **SkillSwap Mobile Application**. This mobile app allows users to **exchange skills** by creating and managing swap requests, chatting with other users, and maintaining personal profiles.

**1.2 Scope**

The **SkillSwap App** is a cross-platform mobile application built using **React Native, Expo, and React Navigation**. The application provides the following key features:

* **User Authentication**: Signup and login functionality.
* **Skill Listings**: Browse available skills offered by other users.
* **Skill Swaps**: Create, view, and manage skill swap requests.
* **Conversations**: Chat with other users to arrange skill exchanges.
* **Profile Management**: View and edit personal information, skills, and profile picture.

The app will support both Android and iOS platforms.

**1.3 Definitions, Acronyms, and Abbreviations**

* **App** – Mobile application
* **SkillSwap** – The name of the system/application
* **Swap** – An exchange between two users offering and requesting skills

**2. Overall Description**

**2.1 Product Perspective**

The application is a **standalone mobile app**. It has a modular structure using navigation stacks and bottom tabs for easy user flow. Data persistence (e.g., storing users, swaps) will later be integrated with backend services (e.g., Firebase, REST API).

**2.2 Product Functions**

* **Login & Signup**: Users authenticate using email and password.
* **Skill Browsing**: Users can view a list of available skills.
* **Skill Swap Management**: Users can send and track swap requests.
* **Conversations**: Users can communicate with others.
* **Profile Editing**: Users can update their name, email, skills, and profile picture.

**2.3 User Classes and Characteristics**

* **Learners** – Users who want to learn new skills.
* **Tutors** – Users who want to share their skills.
* **General Users** – Can be both learners and tutors simultaneously.

**2.4 Operating Environment**

* Platform: **Android (API 23+)**, **iOS 12+**
* Development Framework: **React Native (Expo managed workflow)**
* UI Components: **React Native Paper, Ionicons, Material Icons**

**2.5 Design and Implementation Constraints**

* Built with Expo (requires camera/gallery permissions).
* Requires stable internet for profile image uploads and real-time communication (future backend integration).

**2.6 Assumptions and Dependencies**

* Users have an active internet connection.
* Device supports image picker and storage access.
* Backend server/API will be developed for authentication and data storage.

**3. Specific Requirements**

**3.1 Functional Requirements**

**FR-1: User Authentication**

* **FR-1.1**: User shall be able to log in using email and password.
* **FR-1.2**: User shall be able to create a new account via signup.
* **FR-1.3**: System shall navigate user to **MainTabs** after successful login/signup.

**FR-2: Skill Browsing**

* **FR-2.1**: System shall display a list of available skills.
* **FR-2.2**: Each skill card shall include the skill name, provider name, and a **Request Swap** button.

**FR-3: Swap Management**

* **FR-3.1**: System shall categorize swaps as **Pending**, **Active**, or **Completed**.
* **FR-3.2**: User shall be able to view swap details including offered and requested skills.
* **FR-3.3**: A Floating Action Button (FAB) shall allow users to initiate new swap requests.

**FR-4: Conversations**

* **FR-4.1**: System shall allow users to view and start conversations.
* **FR-4.2**: If no conversation exists, an empty state message shall be shown.
* **FR-4.3**: A FAB shall allow users to start new conversations.

**FR-5: Profile Management**

* **FR-5.1**: User shall be able to view profile information (name, email, skills, profile picture).
* **FR-5.2**: User shall be able to edit and save profile information.
* **FR-5.3**: User shall be able to upload a profile picture from the gallery.

**3.2 Non-Functional Requirements**

**Performance Requirements**

* The app should load each screen within **2 seconds**.
* Navigation transitions should be smooth and responsive.

**Security Requirements**

* User credentials must be securely handled (integration with backend authentication in future).
* Profile image access requires user permission.

**Usability Requirements**

* App must have an intuitive UI with bottom tabs for navigation.
* Buttons and input fields should be clearly visible and responsive.

**Reliability Requirements**

* The app should handle image upload failures gracefully.
* The system should maintain navigation flow even if some data is unavailable.

**4. System Features**

* **Login/Signup**: Authenticate users.
* **Skill Swap Listings**: Browse available skills.
* **My Swaps**: Track and manage swap requests.
* **Conversations**: Chat functionality.
* **Profile**: View and update personal details.

**5. Future Enhancements**

* Integration with **Firebase/Backend API** for real-time data.
* Push notifications for new swap requests and messages.
* In-app chat with file sharing.
* Search and filter skills by category.
* Rating system for users after swap completion.