

# Skill-Swap Mobile Application

## Software Requirements Specification (SRS)

### 1. Introduction

#### 1.1 Purpose

This document serves as the official Software Requirements Specification (SRS) for the *SkillSwap Mobile Application*. The purpose of this application is to provide a structured platform where individuals can share their knowledge, abilities, and expertise with others while learning new skills in return.

Unlike traditional learning platforms that involve monetary payments, SkillSwap focuses on a barter system of skills. For example, one user may offer guitar lessons, while another user may provide coding tutorials. Through SkillSwap, these individuals can connect and exchange their abilities without involving financial transactions.

This SRS describes in detail the features, roles, user stories, and technical requirements necessary to build the application, ensuring that both developers and stakeholders have a clear understanding of the project's objectives.

#### 1.2 Scope

The SkillSwap application is designed to function as a peer-to-peer skill-sharing network. Its scope includes the following:

- **Core Features:**
  - User profile creation and editing.
  - Posting and browsing skill listings.
  - A booking system that allows learners to schedule sessions with tutors.
  - Ratings and review functionality to build trust and maintain quality.
- **Out of Scope (Not included in MVP):**
  - In-app monetary payments.
  - Direct video calling features.
  - Instant messaging or chat-based communication.

The focus will be on building an MVP (Minimum Viable Product) that provides essential functionality for users to connect, share, and learn skills in a simple and effective way.

#### 1.3 Definitions and Acronyms

To ensure clarity, the following terms are used throughout this document:

- **Tutor** – A user who is offering to teach a specific skill.

- **Learner** – A user who wants to learn or book a skill session.
- **SkillSwap** – The act of exchanging skills between two or more users without money.
- **CRUD** – Standard operations for data: Create, Read, Update, Delete.
- **MVP** – Minimum Viable Product, the initial version of the application with only the most essential features.
- **UML** – Unified Modeling Language, used for creating system diagrams.

## 2. Overall Description

### 2.1 User Roles

The SkillSwap application supports two main roles:

#### 1. Student (Tutor/Learner):

- A single user may act as both a tutor and a learner, depending on the situation.
- They can create and edit their personal profiles.
- They may post skill offers, browse listings, and book sessions.
- They can also leave reviews after sessions.

#### 2. Admin:

- Has elevated privileges and can view all data within the system.
- Responsible for removing inappropriate content, suspending, or deleting accounts when necessary.

### 2.2 User Stories

To understand the needs of users, the following user stories highlight real-world use cases:

- **Learner Story:** A learner filters tutors by skill type and rating before booking.
- **Tutor Story:** A tutor can set their available time slots for learners to choose from.
- **Student Story:** A student can view their history of past learning and teaching sessions.
- **Admin Story:** An administrator can flag or remove inappropriate or offensive offers.

## 3. Functional Requirements

The functional requirements (FR) describe the expected features of the application:

1. **FR1:** Users must be able to register accounts using an email and password.
2. **FR2:** Secure login and logout functionality must be implemented.
3. **FR3:** Users can create and edit profiles, including name, bio, and profile picture.
4. **FR4:** Tutors can create new skill offers with details such as title, description, category, and duration.

5. **FR5:** Learners can search for skill offers by keyword, category, or location.
6. **FR6:** Offers must display relevant tutor profiles for transparency.
7. **FR7:** Learners can book sessions with tutors.
8. **FR8:** Both tutor and learner receive notifications when a session is booked.
9. **FR9:** Users can rate and review one another after completing a session.
10. **FR10:** The application will display an average rating for each user.
11. **FR11:** Users can delete skill offers or cancel upcoming sessions.
12. **FR12:** Admins have the ability to delete any offer or user profile.
13. **FR13:** All data will be stored persistently using a secure database.
14. **FR14:** Tutors can show how many available sessions are left.
15. **FR15:** Users can report inappropriate content to the admin.

## 4. Non-Functional Requirements

To ensure quality and reliability, the system must also meet the following non-functional requirements:

- **Usability:** The user interface should be simple, intuitive, and easy to navigate.
- **Performance:** Application screens should load in less than 2 seconds when on a Wi-Fi connection.
- **Security:** User passwords must be encrypted, and all communication should occur over HTTPS/SSL.
- **Reliability:** The system should maintain at least 99.9% uptime and provide regular data backups.

## 5. Database Schema

The SkillSwap application will rely on a database with the following main entities:

- **Users Table:**
  - Fields: `_id`, `email`, `passwordHash`, `name`, `bio`, `profilePic`, `skills[]`, `avgRating`.
- **Offers Table:**
  - Fields: `_id`, `title`, `description`, `category`, `createdBy`, `createdAt`.
- **Sessions Table:**
  - Fields: `_id`, `offerId`, `tutorId`, `learnerId`, `scheduledTime`, `status`.
- **Reviews Table:**
  - Fields: `_id`, `fromUser`, `toUser`, `rating`, `comment`, `createdAt`.

## 6. MVP Frontend Development (React Native)

The MVP version of the application will focus on building a **mobile-first user interface** using React Native. The following screens will be included:

### 1. Login/Signup Screen

- Provides simple forms for user login and account creation.
- Dummy data for testing:
  - Email: `test@student.com`
  - Password: `12345`

### 2. Home Feed Screen

- Displays a scrollable list of skill offer cards.
- Example offers for testing: *Python Tutoring, Guitar Lessons, Drawing Basics, Yoga & Meditation.*

### 3. Create Post Screen

- Allows tutors to create and publish new skill offers by filling in a form.

### 4. Profile Screen

- Displays a user's personal information, skills, and biography.