# SkillSwap Mobile Application

**Software Requirements Specification (SRS)**

## 1. Introduction

### 1.1 Purpose

This document defines the requirements for the **SkillSwap mobile application**. The purpose of this system is to enable peer-to-peer skill exchange among students and professionals. Instead of monetary payment, users can **offer skills** they have and **learn skills** they want. This SRS provides details of functionality, system constraints, and design requirements.

### 1.2 Scope

**In Scope (MVP):**

User registration and login

Create and manage user profiles

Create skill offers and requests

Search/filter skill offers

Book sessions between learners and tutors

Ratings and reviews

**Out of Scope (for initial release):**

In-app payments

Video calling or chat integration

Advanced AI-based recommendations

### 1.3 Definitions & Acronyms

**Tutor**: A user offering a skill.

**Learner**: A user seeking a skill.

**SkillSwap**: Trading skills without money.

**CRUD**: Create, Read, Update, Delete.

**MVP**: Minimum Viable Product.

**UML**: Unified Modeling Language.

## 2. Overall Description

### 2.1 User Roles

| **Role** | **Permissions & Goals** |
| --- | --- |
| **Student (Tutor/Learner)** | Can switch roles, create/edit profile, post offers, search skills, book sessions, review tutors. |
| **Admin** | Monitor user activity, remove inappropriate content, suspend/delete accounts. |

### 2.2 User Stories

As a learner, I want to filter tutors by skill & rating so I can find the best match.

As a tutor, I want to set availability so learners can only book me when free.

As a student, I want to see my past/upcoming sessions to manage commitments.

As an admin, I want to remove inappropriate offers to keep the community safe.

## 3. Functional Requirements

**FR1:** Register with email & password.

**FR2:** Secure login/logout.

**FR3:** Create & edit profile (name, bio, picture).

**FR4:** Post new skill offers (title, description, category, duration).

**FR5:** Search/filter skill offers by keyword, category, or location.

**FR6:** View list of available skill offers with tutor info.

**FR7:** Book sessions with tutors.

**FR8:** Notify both users when booking is confirmed.

**FR9:** Rate & review other users after a session.

**FR10:** Show average rating on profiles.

**FR11:** Users can delete offers or cancel sessions.

**FR12:** Admin can remove offers/accounts.

**FR13:** Persistent storage in database.

**FR14:** Show number of sessions a tutor can still offer.

**FR15:** Allow reporting inappropriate content.

## 4. Non-Functional Requirements

**Usability**: Simple interface, tasks ≤3 taps.

**Performance**: Screen loads ≤2 sec on Wi-Fi.

**Security**: Passwords hashed & salted; HTTPS/SSL for all traffic.

**Reliability**: 99.9% uptime; regular backups.

**Scalability**: Should support thousands of concurrent users.

## 5. Database Schema

| **Collection** | **Fields** | **Description** |
| --- | --- | --- |
| **Users** | \_id, email, passwordHash, name, bio, profilePic, skills[], avgRating | Stores user info. |
| **Offers** | \_id, title, description, category, createdBy, createdAt | Skill offers posted. |
| **Sessions** | \_id, offerId, tutorId, learnerId, scheduledTime, status | Booking records. |
| **Reviews** | \_id, fromUser, toUser, rating, comment, createdAt | Ratings and feedback. |

## 6. UML Diagrams

### 6.1 Use Case Diagram

Actors: **Student, Admin**  
Use cases: Login, Create Profile, Post Offer, Search Skills, Book Session, Review, Manage Users.

👉 (Insert diagram image here – can be made in Draw.io/Lucidchart)

### 6.2 Class Diagram

Classes: **User, Offer, Session, Review, Admin**  
Relationships:

User ↔ Offer (1-to-many)

User ↔ Session (many-to-many)

User ↔ Review (1-to-many)

👉 (Insert diagram here)

## 7. MVP Frontend (React Native)

### 7.1 Login/Signup Screen

Fields: Email, Password

Button: Login → goes to Home

### 7.2 Home Screen

List of dummy offers: Python, Guitar, Drawing, Yoga

### 7.3 Create Post Screen

Input fields for skill title + description

Button: Post → logs to console

### 7.4 Profile Screen

Shows static user info (name, skills, bio)

## 8. Deliverables

**SRS Document** (this file)

**UML Diagrams** (export as PNG/JPEG and include in /A1 folder)

**MVP React Native app** (in /skillswap-mvp folder, linked to GitHub)