Software Requirements Specification (SRS)

Project: SkillSwap Mobile Application

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the requirements for the SkillSwap mobile application. The app provides a peer-to-peer platform where users can exchange skills instead of money. Users can act as learners or tutors, book sessions, and rate each other’s contributions.

## 1.2 Scope

SkillSwap will enable:  
- User Profiles: Manage profile information and listed skills.  
- Skill Listings: Create, search, and browse skills.  
- Booking System: Schedule and manage skill exchange sessions.  
- Ratings/Reviews: Review and rate tutors/learners.  
  
Out of scope (initial version):  
- In-app payments.  
- Real-time chat or video calls.

## 1.3 Definitions & Acronyms

Tutor: A user offering skills.  
Learner: A user seeking skills.  
SkillSwap: Exchange of 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) | Create profile, list skills, request skills, book sessions, and leave reviews. |
| Admin | Monitor content, delete inappropriate posts, suspend/delete accounts. |

## 2.2 User Stories

- As a learner, I want to filter tutors by skill/rating to find reliable helpers.  
- As a tutor, I want to set available time slots to control bookings.  
- As a student, I want to view session history for tracking.  
- As an admin, I want to remove inappropriate content to maintain safety.

# 3. Functional Requirements

FR1: User registration with email/password.  
FR2: Secure login/logout.  
FR3: Profile creation/editing (name, bio, profile pic).  
FR4: Post skill offers (title, description, category, duration).  
FR5: Search skills by keywords, category, or location.  
FR6: Display skill offers with tutor profile.  
FR7: Book session with time slot.  
FR8: Notify both tutor & learner after booking.  
FR9: Allow ratings/reviews after sessions.  
FR10: Calculate/display average rating.  
FR11: Cancel/delete skill offers or sessions.  
FR12: Admin deletes offers/accounts.  
FR13: Persistent storage of user data.  
FR14: Show available tutor sessions.  
FR15: Report inappropriate content.

# 4. Non-Functional Requirements

Usability: Intuitive UI (post skill in ≤3 taps).  
Performance: Pages load ≤2s on Wi-Fi.  
Security: Password hashing + HTTPS.  
Reliability: 99.9% uptime; regular backups.

# 5. Database Schema

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

# 6. UML Diagrams (Placeholder)

6.1 Use Case Diagram: Actors (Student, Admin), Use cases (Login, Post Offer, Book Session).  
6.2 Class Diagram: User, Offer, Session, Review with attributes & relationships.

# 7. MVP Frontend Development (React Native)

7.1 Login/Signup Screen: Simple form with dummy credentials.  
7.2 Home Feed Screen: Scrollable skill offers list.  
7.3 Create Post Screen: Form to add new skill offers.  
7.4 Profile Screen: User info, skills, and bio.