Software Requirements Specification (SRS)

For SkillSwap Mobile Application

1. Introducti

1.1 **Purpose**

The purpose of this document is to define the functional and non-functional requirements of the SkillSwap mobile application. This app provides a peer-to-peer skill-sharing platform where university students can exchange skills without monetary transactions. The SRS serves as a reference for developers, testers, and stakeholders, ensuring alignment on the system’s features, behavior, and constraints.

1.2 **Scope**

SkillSwap is a cross-platform mobile app (Android & iOS, built with React Native) with a Node.js/Express backend and a MongoDB database.

The system enables:

Student registration and profile management.

Posting and browsing skill offers.

Requesting, accepting, or rejecting skill swap sessions.

In-app messaging for coordination.

Notifications and reminders.

Admin oversight for content moderation and user management.

The MVP (Minimum Viable Product) will include account creation, profile setup, skill posting, browsing, swap requests, and basic chat/notifications. Future versions may add calendar scheduling, group sessions, gamification (points/rewards), or integration with third-party tools like Zoom/Google Calendar.

1.3 Definitions, Acronyms, Abbreviations

SkillSwap: The application/platform for swapping skills.

Student/User: A normal app user who can offer and request skills.

Admin: A privileged user who manages and moderates the platform.

Skill Offer: A post where a user advertises a skill they can teach.

Skill Swap Session: A confirmed agreement between two users to exchange skills.

MVP: Minimum Viable Product.

SRS: Software Requirements Specification.

---

2. Overall Description

2.1 Product Perspective

Standalone mobile application.

Client-server architecture (React Native frontend + REST API backend).

Cloud-hosted database (MongoDB).

Notifications via push or in-app UI indicators.

2.2 User Roles

Student (Normal User):

Create/manage profiles.

Offer and request skills.

Chat with peers.

Leave reviews after sessions.

Admin:

Manage user accounts.

Moderate content (posts, reviews).

Handle reports of abuse.

2.3 User Characteristics

Target: University/college students (18–30).

Familiar with mobile apps/social media.

Expect quick onboarding and easy navigation.

2.4 Constraints

Cross-platform support (Android 10+ and iOS 13+).

Secure data handling (password hashing, HTTPS).

Limited real-time features in MVP (basic chat, simple notifications).

---

3. User Stories

As a student, I want to register easily, so I can use the app quickly.

As a student, I want to list my skills and learning needs, so others know what I can offer.

As a student, I want to browse and search skills, so I can find what interests me.

As a student, I want to send and receive skill swap requests, so I can exchange skills.

As a student, I want to chat with my partner after a swap request is accepted, so we can coordinate details.

As a student, I want to leave ratings/reviews, so trust can be built on the platform.

As an admin, I want to moderate users and content, so the community stays safe.

---

4. Functional Requirements

1. User Registration & Authentication

Register via email/password.

Login/logout securely.

Passwords hashed and stored securely.

2. Profile Management

Edit bio, skills offered, and skills wanted.

View other users’ profiles.

3. Skill Offer Management

Create/edit/delete skill offers.

Browse skill offers in a feed.

Search/filter skill offers by category/tags.

View offer details.

4. Skill Swap Requests

Send swap request with a proposed skill in return.

Accept/reject incoming requests.

Mark a session as completed.

5. Communication

In-app chat between matched users.

Notifications for new requests, messages, acceptances.

6. Feedback & Reviews

Rate and review users after completed sessions.

Ratings update user reputation.

7. Admin Functions

Manage user accounts (ban/deactivate).

Moderate posts and reviews.

Handle abuse reports.

---

5. Non-Functional Requirements

Usability: Simple, intuitive UI with onboarding tutorial.

Performance: Feed loads in <2s, search <1s.

Scalability: Support up to 10k users in early stages.

Security: HTTPS, hashed passwords, access control.

Reliability: Graceful error handling, atomic operations.

Maintainability: Modular code, well-documented APIs.

Privacy: Minimal data collection, only necessary info shared.

Extensibility: Future support for points system, scheduling, or group sessions.

---

6. Database Schema (MongoDB)

Users Collection

{

"\_id": ObjectId("..."),

"name": "Alice Doe",

"email": "alice@example.com",

"passwordHash": "<hash>",

"role": "student",

"bio": "I love learning languages.",

"skillsOffered": ["French", "Guitar"],

"skillsWanted": ["JavaScript"],

"rating": 4.8,

"numReviews": 5,

"createdAt": ISODate("2025-09-01T...")

}

Offers Collection

{

"\_id": ObjectId("..."),

"userId": ObjectId("..."),

"title": "Guitar Lessons",

"description": "Basics of acoustic guitar.",

"category": "Music",

"tags": ["music", "guitar"],

"location": "Campus",

"availability": "Weekends",

"status": "active",

"createdAt": ISODate("2025-09-05T...")

}

Sessions Collection

{

"\_id": ObjectId("..."),

"userAId": ObjectId("..."),

"userBId": ObjectId("..."),

"skillFromA": "Spanish",

"skillFromB": "Guitar",

"status": "completed",

"initiatedAt": ISODate("2025-09-10T..."),

"acceptedAt": ISODate("2025-09-11T..."),

"completedAt": ISODate("2025-09-20T...")

}

Reviews Collection

{

"\_id": ObjectId("..."),

"sessionId": ObjectId("..."),

"reviewerId": ObjectId("..."),

"revieweeId": ObjectId("..."),

"rating": 5,

"comment": "Great teacher!",

"createdAt": ISODate("2025-09-21T...")

}

Messages Collection (NEW – for chat)

{

"\_id": ObjectId("..."),

"threadId": ObjectId("..."),

"fromUserId": ObjectId("..."),

"toUserId": ObjectId("..."),

"text": "Hi, when are you available?",

"timestamp": ISODate("2025-09-22T...")

}

Reports Collection (NEW – for abuse reporting)

{

"\_id": ObjectId("..."),

"reportedById": ObjectId("..."),

"targetUserId": ObjectId("..."),

"reason": "Inappropriate content",

"status": "pending",

"createdAt": ISODate("2025-09-23T...")

}

---

7. Future Enhancements

Gamification: Points, badges for active contributors.

Calendar Integration: Schedule sessions directly in-app.

Video/Voice Chat: Built-in online sessions.

AI Matchmaking: Suggesting skill partners based on interests.