SkillSwap — Detailed Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

This document provides a detailed specification for SkillSwap, a mobile application prototype that enables students to exchange skills and knowledge. The SRS defines system objectives, functional and non-functional requirements, data models, UML diagrams, and acceptance criteria.

1.2 Scope

SkillSwap facilitates peer-to-peer skill sharing where students can act as tutors or learners. The MVP scope includes user authentication, posting and browsing skill offers, booking sessions, profile management, and leaving reviews. Future enhancements may include chat, payments, persistent storage, and admin moderation.

1.3 Definitions

• Tutor: A student offering a skill. • Learner: A student requesting or booking a skill. • Offer: A skill listing created by a Tutor. • Session: A scheduled booking of an offer. • Review: Feedback and rating left by a learner after a session.

2. Overall Description

2.1 Product Perspective

The app is designed as a standalone React Native prototype running in Expo. It uses dummy authentication and local state to simulate real-world interactions. It is portable across Android, iOS, and web.

2.2 Product Functions

Key product functions include user registration/login, profile editing, skill offer posting, browsing offers, booking sessions, and submitting reviews. The MVP avoids backend complexity by keeping all logic local.

2.3 User Roles

• Student: Can login, create offers, browse offers, book sessions, manage profile, and leave reviews. • Admin (future scope): Can moderate content and manage users.

3. Functional Requirements

FR1: Users shall register/login with email and password.
FR2: Users shall create and update their profile.
FR3: Users shall create skill offers (title, description, category).
FR4: Users shall browse skill offers in a feed.
FR5: Users shall book a skill session.
FR6: Users shall leave a review after a session.
FR7: The system shall display average ratings for each user.

4. Non-Functional Requirements

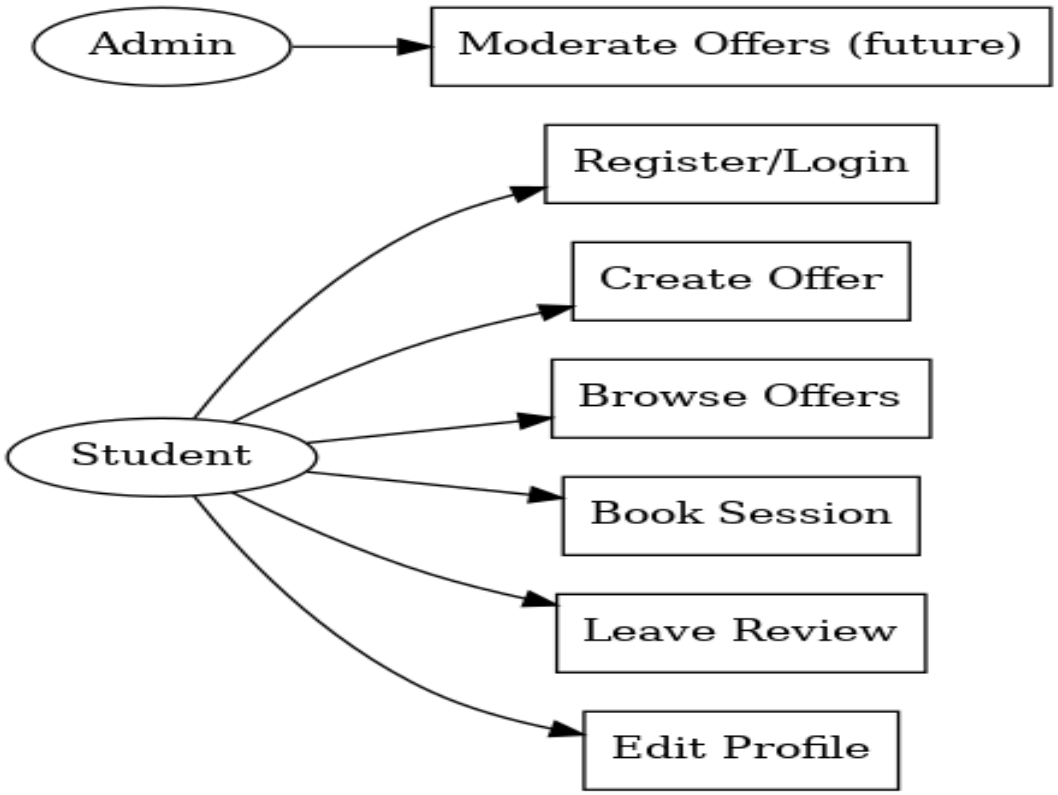
Usability: The app should be simple and intuitive, suitable for students.
Performance: All screens should load within 2 seconds under normal conditions.
Security: Credentials will be verified locally for MVP; production must use secure encryption.
Portability: App runs across Android, iOS, and web (via Expo).
Scalability: Architecture is modular, supporting future backend integration.

5. Data Model

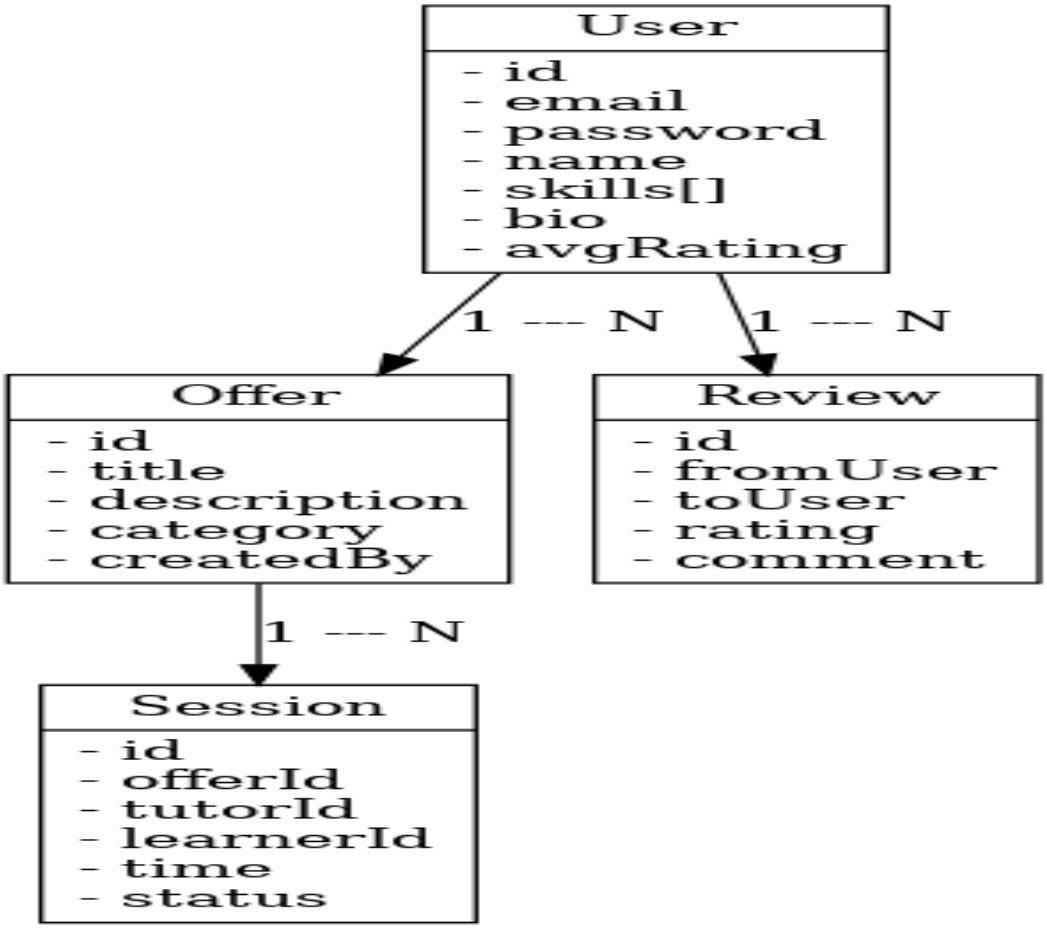
• User: id, email, password, name, skills[], bio, avgRating • Offer: id, title, description, category, createdBy • Session: id, offerId, tutorId, learnerId, time, status • Review: id, fromUser, toUser, rating, comment

6. UML Diagrams

6.1 Use Case Diagram



6.2 Class Diagram



7. Acceptance Criteria

AC1: A user can log in with dummy credentials (test@student.com / 12345).
AC2: Users can create at least one skill offer and see it in the feed.
AC3: Users can book a skill session with confirmation.
AC4: Users can view and edit profile details.
AC5: Reviews can be added and displayed locally.

8. README Instructions

• Setup: Open https://snack.expo.dev, create a new project, replace App.js and add the navigation/screens/components files. • Run: Click Run on Web or scan QR code in Expo Go. • Prototype Login: test@student.com / 12345

9. Conclusion

This detailed SRS defines the MVP for SkillSwap. It ensures students can exchange skills easily while keeping the design modular for future enhancements like backend APIs, messaging, and payment integration.