Software Requirements Specification (SRS)

Project: SkillSwap Mobile Application (Frontend MVP Only)

Version: 1.0

Date: 25 Sep 2025

# 1. Purpose

SkillSwap enables students to trade skills by posting offers. This frontend-only MVP focuses on login/signup, home feed, profile, and create post screens.

# 2. Scope

In scope (Frontend MVP):  
- Email/password signup and login  
- User profiles with name, bio, avatar  
- Skill Offers: create, browse in feed  
  
Out of scope (Frontend MVP):  
- Bookings, reviews, ratings  
- Admin console  
- Notifications  
- Payments, chat, video, calendar sync

# 3. Stakeholders and Users

- Student user: can sign up, log in, create posts, and manage a profile.  
- Instructor/Developer: evaluates and extends MVP.

# 4. Assumptions and Dependencies

- React Native frontend only; backend not implemented.  
- Mock data used for feed and posts.  
- iOS 15+ and Android 10+ supported.

# 5. References

Course assignment brief and deliverables.

# 6. High-level System Context

The app consists of four screens:  
- Login/Signup: account creation and access  
- Home: feed of posts (static or mock data)  
- Create Post: form to create a new skill post  
- Profile: view/edit personal details

# 7. User Stories

- As a new user, I want to sign up so I can start using the app.  
- As a returning user, I want to log in so I can access my account.  
- As a student, I want to view a home feed of posts from other users.  
- As a student, I want to create my own post so I can share what I offer.  
- As a student, I want to edit my profile so it reflects my information.

# 8. Functional Requirements

Authentication  
- Register with email + password  
- Login and logout flows  
  
Profile  
- View/edit profile details (name, bio, avatar)  
  
Home Feed  
- Display list of posts (static/mock data)  
  
Create Post  
- Add a new post with title, description, and tags

# 9. Non-functional Requirements

- Usability: simple navigation between screens  
- Performance: load screens quickly with mock data  
- Reliability: stable operation without crashes  
- Compatibility: React Native 0.7x+, iOS 15+, Android 10+

# 10. Data Model

Users  
- \_id, email, password (mock), name, bio, avatarUrl  
  
Posts  
- \_id, title, description, tags, createdBy  
  
(No bookings, reviews, or admin features in this frontend MVP)

# 11. Business Rules

- A post must include a title and description.  
- Users can only edit their own profile and posts.  
- No duplicate emails allowed in signup (mock constraint).

# 12. Error Handling

- Invalid input shows error messages (e.g., empty fields).  
- Incorrect login shows 'invalid credentials' (mock).  
- Navigation errors redirect to home screen.

# 13. Security Controls

- Basic validation on signup/login fields.  
- Passwords stored in mock state only; no real backend security in MVP.

# 14. Analytics

- Optional: track screen navigation and post creation events (mock).

# 15. Acceptance Criteria

- User can sign up and log in successfully (mock auth).  
- User can navigate between login, home, create post, and profile screens.  
- User can create a post and see it in the feed.  
- User can edit profile details and see updates reflected.

# 16. Risks & Mitigations

- No backend: data not persistent → mitigate by using local mock storage.  
- Limited functionality: focus only on UI → acceptable for MVP.  
- Security minimal → clearly marked as mock-only.

# 17. Future Work

- Backend integration with API and database  
- Booking, reviews, notifications  
- Admin tools and moderation  
- Payments, chat, advanced search