

Software Design Document

SkillSwap Hub — Building a Local Mutual-Aid Economy

1. Introduction

1.1 Background and Motivation

SkillSwap Hub is a community-oriented application that helps people exchange skills locally without money. Many students and young professionals want to learn practical skills or receive help in daily life, but paid courses are often expensive and inflexible. At the same time, people also have skills they can share, yet informal skill-exchange channels (such as group chats or forum posts) are typically unstructured, inefficient, and difficult to trust. Our project aims to provide a structured solution that makes local skill exchange easier, fairer, and safer.

A key design idea in SkillSwap Hub is to use time as a standardized unit of value. The app introduces “Time Coins,” where one hour of service is treated as one unit. This reduces negotiation and avoids pricing conflicts. Combined with location-based matching, users can discover relevant offers nearby and build long-term trust through transaction records and mutual ratings. The overall product logic follows a clear loop: users publish skills, earn Time Coins by helping others, spend Time Coins to learn new skills, and gradually build reputation through repeated exchanges.

1.2 Project Goals and Scope

The main goal of this project is to design and implement a functional Android application that supports the core “skill exchange” workflow with a cloud backend and a basic trust mechanism. The project also includes a lightweight web component (HTML) as a supporting interface (for example, presenting basic information, documentation pages, or supplementary display). The backend is implemented using Firebase services to support authentication, data storage, and real-time updates.

The scope of the project focuses on the essential functions required to complete a full exchange cycle: skill posting, local discovery/matching, chat/communication, transaction completion, Time Coin updates, and rating/reputation. In addition, we include basic testing and user feedback analysis through small-scale user testing and a third-party cloud testing platform (Firebase Test Lab).

1.3 Intended Audience

This document is written for instructors, teaching assistants, and reviewers who want to understand the system requirements, architecture, interface design, key technologies, and testing results. It can also serve as a reference for future maintenance or feature expansion.

2. Requirements Specification

2.1 User Requirements

SkillSwap Hub is designed primarily for people aged roughly 20–40, especially students and young professionals in urban environments. These users typically have limited budgets but strong motivation for skill improvement and social interaction. Based on our product positioning, we consider three representative user roles in the system: (1) users who mainly want to learn new skills (“Explorers”), (2) users who mainly want to teach and earn Time Coins (“Sharers”), and (3) users who help organize community activities and encourage offline participation (“Connectors”).

From a user perspective, the app should meet several practical expectations: it should be easy to publish skills, easy to find relevant matches nearby, and safe enough to support communication before meeting offline. Users should also be able to view their Time Coin balance and reputation, so they feel the exchange system is fair and reliable.

2.2 Functional Requirements

The functional requirements below describe what the system must do. The wording is intentionally concrete, because these items are directly testable in the implemented system.

FR1 — User Registration and Login: The system shall allow users to register and log in using a secure authentication mechanism.

FR2 — User Profile Management: The system shall allow users to create and update a profile including basic information, skill tags, and personal introduction.

FR3 — Skill Posting: The system shall allow users to publish skill offers with category/tags, description, and availability.

FR4 — Local Discovery and Matching: The system shall allow users to browse/search posts and prioritize items geographically close, with ranking based on distance, tags, and reputation.

FR5 — Chat and Communication: The system shall provide an in-app chat channel for users to communicate before confirming an exchange.

FR6 — Exchange Confirmation and Completion: The system shall allow users to confirm an exchange and mark it as completed.

FR7 — Time Coin Accounting: The system shall maintain a Time Coin balance and update it consistently after completion.

FR8 — Rating and Reputation: After each exchange, both sides shall be able to rate each other and the system shall summarize reputation.

FR9 — Basic Safety and Reporting (Optional): The system should allow users to block or report problematic accounts.

2.3 Non-Functional Requirements

Usability: The main user flows should be easy to learn and consistent.

Performance: Common operations such as loading nearby posts or opening a chat should respond quickly under typical network conditions.

Reliability: The system should avoid inconsistent states under repeated taps or network latency, especially during coin updates.

Security and Privacy: User data must be protected by authentication, access control rules, and secure transport.

Maintainability: The codebase should be modular with clear separation between UI, business logic, and data access.

3. Overall Design

3.1 System Architecture Overview

SkillSwap Hub follows a client–cloud structure. The Android app is the primary client. Firebase provides backend services for authentication, data storage, and real-time synchronization. A lightweight web component (HTML) is used as a supporting interface.

At a conceptual level, the app is organized using an MVC-style approach: the View layer contains Android UI screens; the Controller layer orchestrates user actions; and the Model layer represents domain entities and communicates with Firebase through a centralized data access module. This structure keeps UI logic separate from data logic and improves maintainability.

3.2 Main Modules and Responsibilities

The system is implemented around several major modules. The User & Profile Module handles registration, login, session persistence, and profile editing. The Skill Marketplace Module supports posting and browsing/searching. The Matching Module ranks or filters posts using distance, tags, and reputation. The Chat Module supports user-to-user communication, and the Exchange & Time Coin Module manages confirmation, completion, and accounting updates. Finally, the Rating & Trust Module stores ratings and computes a simple reputation score displayed on profiles.

3.3 Data Design

The data model is designed around several core entities: User (profile, coin balance, reputation), SkillPost (offer details and location), Exchange (participants, status, duration), ChatThread/Message (conversation records), and Rating (score and optional comment).

A practical consideration is transactional consistency for Time Coin accounting. When an exchange is marked completed, updates should be handled carefully to avoid double counting. Firebase supports atomic update patterns and transactions, and our design processes completion as a controlled workflow to minimize inconsistent states.

3.4 Key Workflows

Workflow A: Skill Posting — a logged-in user creates a post, the app validates input, and data is saved to Firebase.

Workflow B: Local Discovery and Matching — the app requests location permission, loads posts in the same area, and ranks them using a deterministic scoring function.

Workflow C: Exchange and Time Coin Update — users chat, confirm an exchange, mark it as completed, then the system updates coins and enables rating submission.

4. User Interface Design

4.1 Design Principles

The UI design focuses on clarity, low learning cost, and fast navigation across the core flows. Users should quickly understand what is available nearby, how Time Coins work, and how to communicate safely before offline meetings. The interface emphasizes consistent layouts, clear action buttons, and minimal steps for common tasks.

4.2 Main Screens

The main screens include Home/Discovery (nearby posts and filters), Skill Detail (full description and chat entry), Post Skill (create/edit offers), Chat (messaging with context), Profile/Dashboard (coin balance and reputation), and Exchange History (status and rating actions). Each core function is reachable within a small number of taps.

5. Key Technologies

5.1 Android Platform and Client Implementation

The primary client is an Android application. Android provides the UI framework, lifecycle management, and access to device features such as location services. The app structure separates UI rendering from business logic and data access to improve maintainability.

5.2 Firebase Backend Services

Firebase is used for authentication, database storage (Firestore or Realtime Database), real-time synchronization, and security rules. This approach reduces the need for a custom server while still providing a realistic client–cloud architecture.

5.3 Location-Based Services (LBS)

Local discovery depends on device location and distance filtering. The Android client uses location APIs to obtain approximate position (with user permission), and matching logic ranks posts by proximity and relevance.

5.4 Technical Challenges and Solutions

A common challenge is maintaining consistent state under asynchronous updates. For example, repeated taps on “Complete Exchange” could trigger duplicate writes and corrupt coin balances.

To mitigate this, we use controlled status transitions (requested → confirmed → completed), bind coin updates to a completed exchange record, temporarily disable critical UI buttons during processing, and use atomic/transaction-like update patterns where available.

6. Testing and User Experience Analysis

6.1 Testing Strategy

Testing was conducted in two layers. First, internal functional testing covered login, posting, discovery, chat, completion, and rating. Second, small-scale user testing with classmates and friends observed whether new users could complete tasks without extensive guidance. Additionally, Firebase Test Lab was used to evaluate stability across different Android devices and configurations.

6.2 Cloud Platform Testing (Firebase Test Lab)

Firebase Test Lab was used to validate basic startup, navigation across major screens, and typical interactions under different device environments. Issues found were addressed through lifecycle handling improvements, UI adjustments, and more robust edge-case checks.

6.3 User Testing and UX Feedback

User testing was task-based and qualitative. Participants generally understood the Time Coin concept and local discovery quickly. Some usability friction points were observed, such as needing clearer guidance on when to confirm an exchange and expecting more visible indicators for reputation and safety. Based on feedback, we refined button wording, improved in-flow hints, and adjusted navigation to reduce unnecessary steps.

7. Conclusion

SkillSwap Hub demonstrates a practical approach to building a local skill exchange platform without money. By combining a time-based virtual currency, location-based discovery, and a trust mechanism, the app supports a complete exchange loop from posting to completion.

Technical challenges related to asynchronous updates and data consistency were addressed through controlled state transitions and careful backend update patterns. Testing included both real user feedback and cloud-based validation via Firebase Test Lab.

For future improvements, the project could add richer verification and safety features, more advanced matching strategies, improved onboarding, and tools for community organizers.