TU 914-4

FINAL YEAR PROJECT PROPOSAL

Martin Cepanec

C21498806

Save4U

Project title: Save4U

Github: https://github.com/MartinCep1?tab=repositories

Project Overview:

Save4U is a mobile app designed to help users save money by optimizing spending and managing subscriptions. It tracks income and expenses via linked credit/debit cards, providing AI-driven financial insights, personalized savings goals, and cash flow predictions. Save4U also finds cheaper deals through web scraping and API integrations, allowing users to switch services and cancel subscriptions directly. With real-time notifications for price drops and gamified savings progress, Save4U ensures users get the best value for their money, making it the ultimate financial management tool.

Business Case:

With the rise of online subscriptions and digital spending, users often lose track of their finances. Unlike cash, digital transactions can give a false sense of available funds, making it harder to manage budgets effectively. In today's fluctuating economy, controlling spending and optimizing savings is more important than ever. Save4U addresses this by providing AI-driven financial insights, cost-saving recommendations, and predictive alerts, helping users take control of their finances and make smarter, more informed decisions.

Technology Overview:

Language: Kotlin for Android

Frontend: Jetpack Compose for UI/UX

Backend: Django for managing the server-side logic

Database: Firebase for real-time data sync

API's: Google Shopping API

Notification: Firebase Cloud Messaging

Payment and Security: Stripe

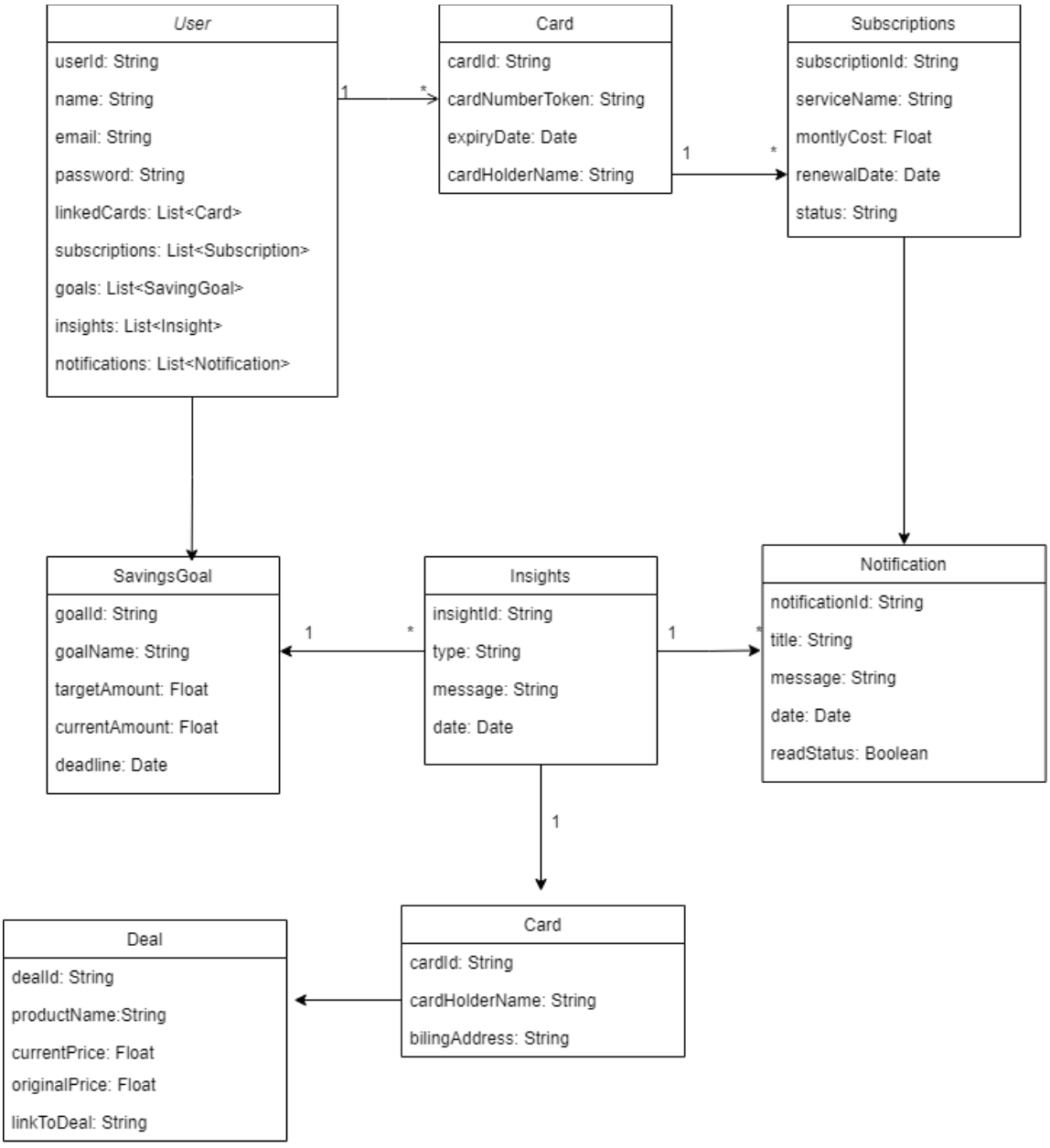
Machine Learning: TensorFlow for AI features

Project Plan:

1. Design the UI/UX using Jetpack Compose in Android Studio, focusing on creating user-friendly screens for financial insights, savings goals, and deal tracking.
2. Integrate APIs like Google Shopping and Skyscanner for dynamic deal tracking and price notifications, while implementing an AI helper for web scraping and recommendations.
3. Set up the backend using Django to manage user data, handle subscriptions, and provide AI-driven insights for personalized savings and spending analysis.
4. Integrate Firebase Firestore for real-time data storage and syncing, and implement Stripe for secure payment processing and subscription management.
5. Conduct comprehensive testing of the AI modules, web scraping functionality, and notification systems, ensuring seamless operation before the final deployment of the application.

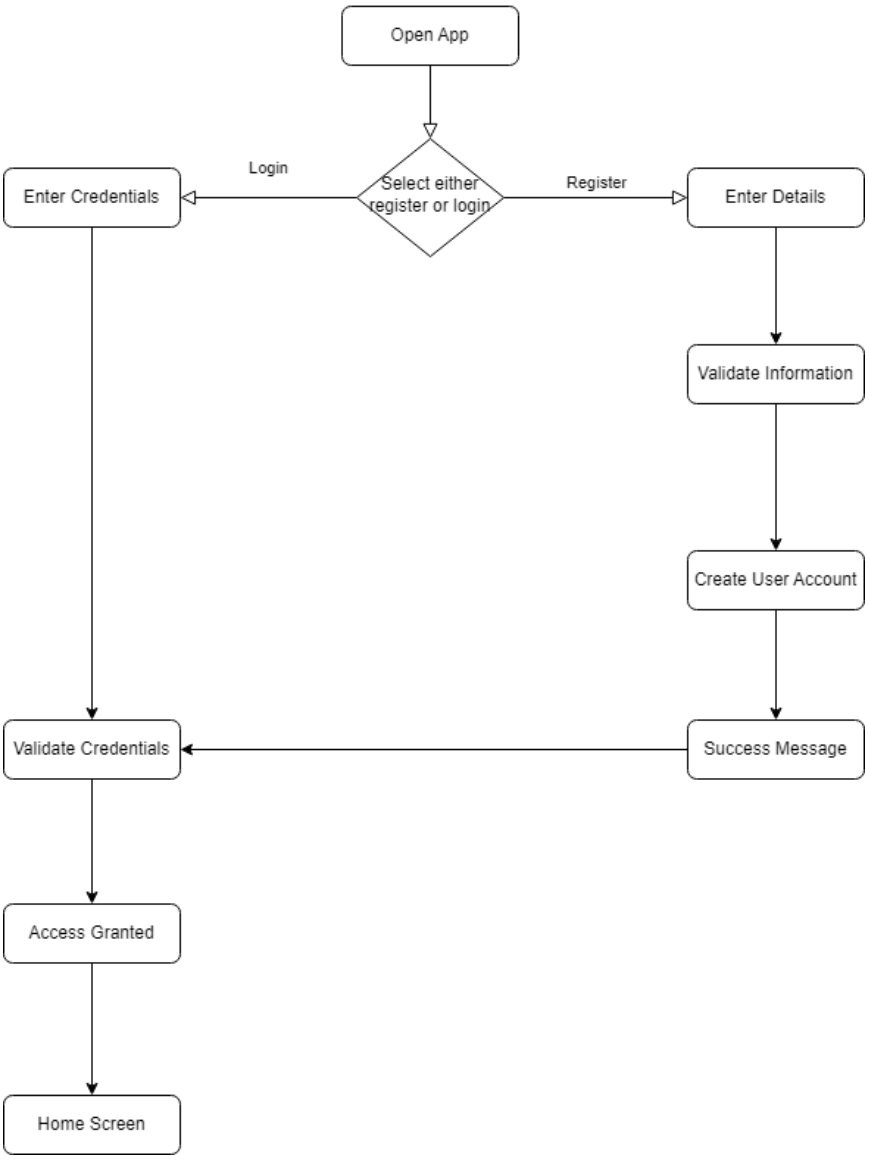
Diagrams:

Class Diagram

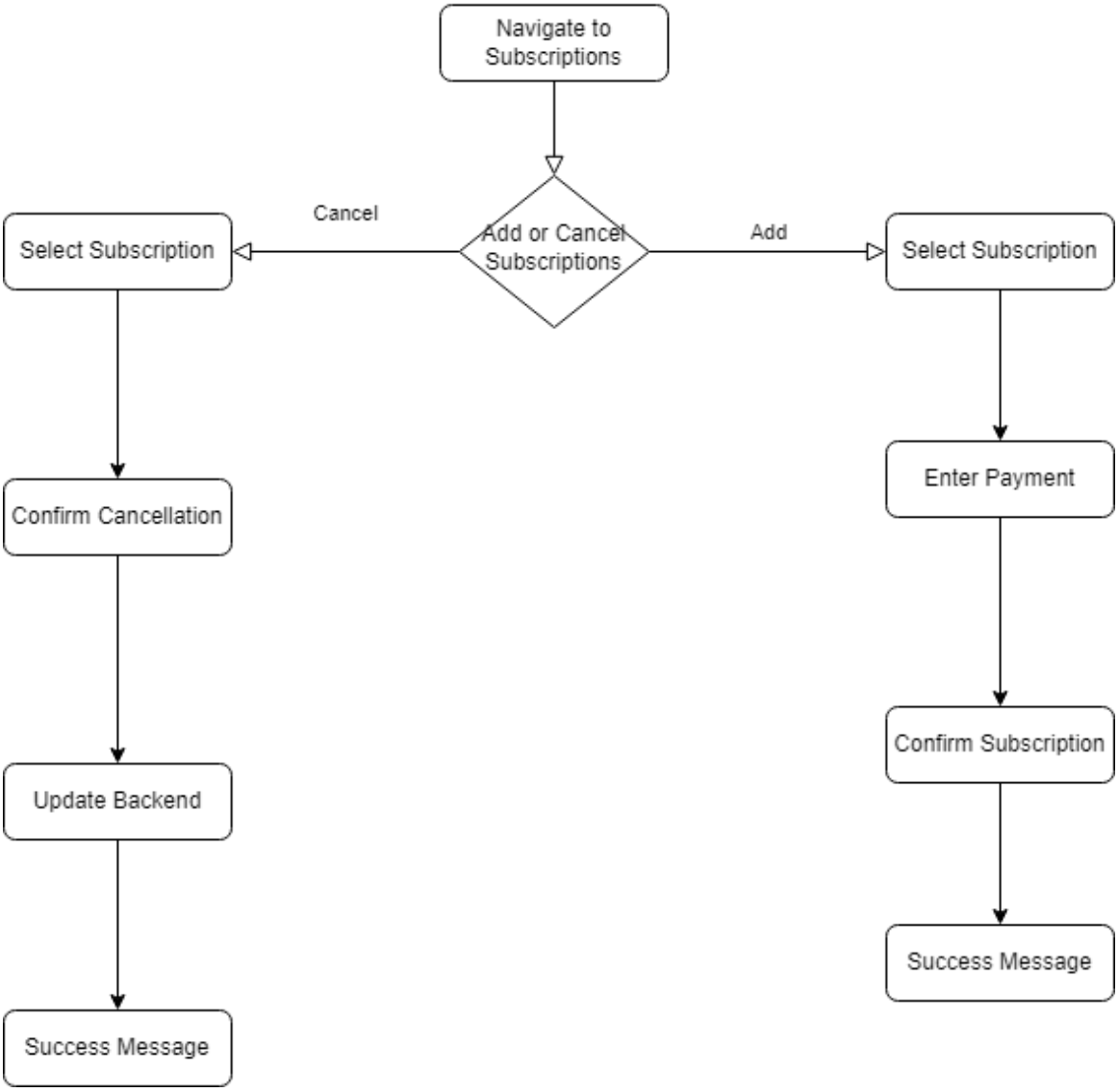


Activity Diagram

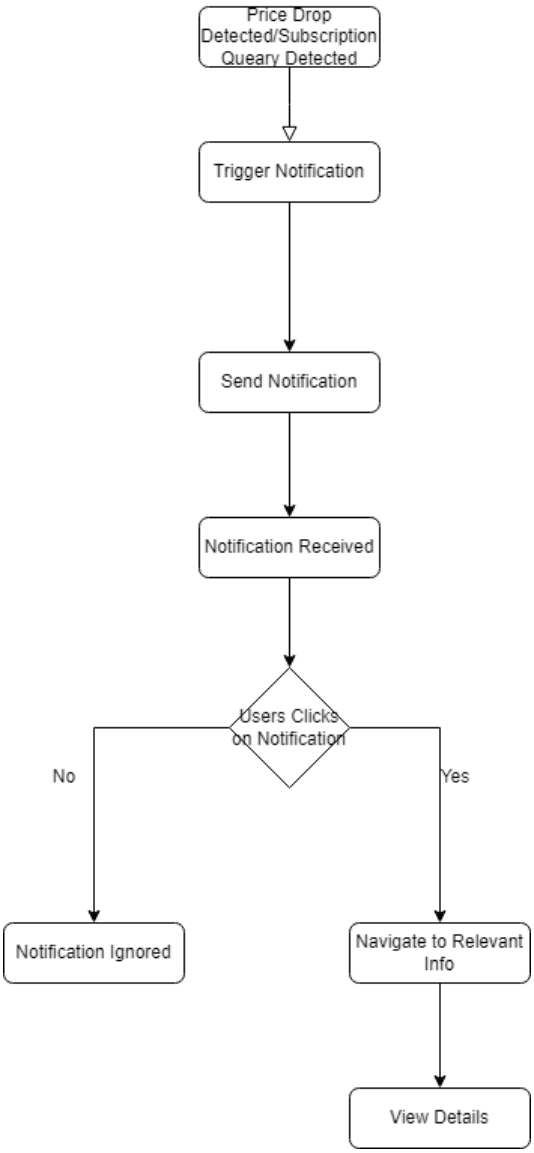
User Registration and Login

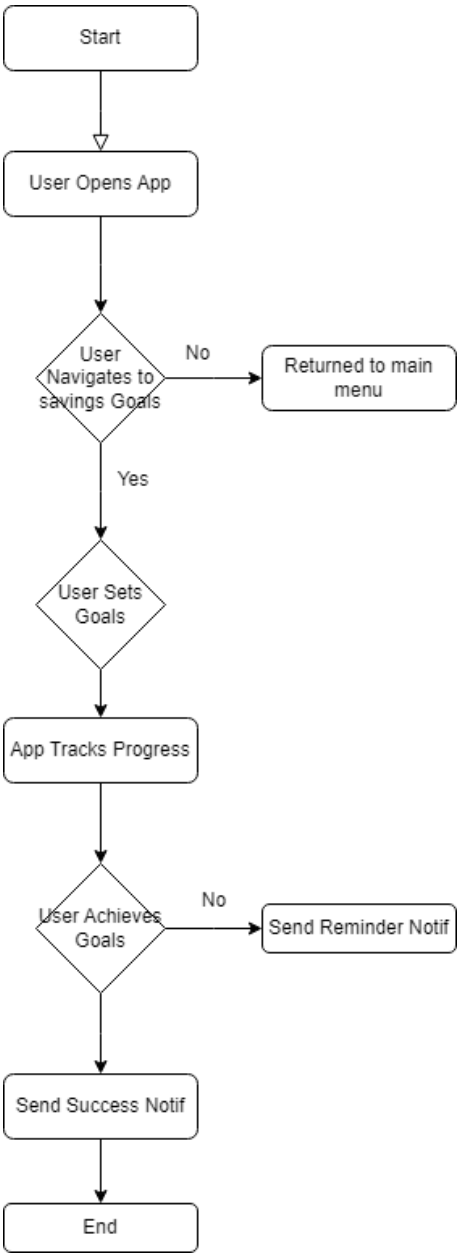


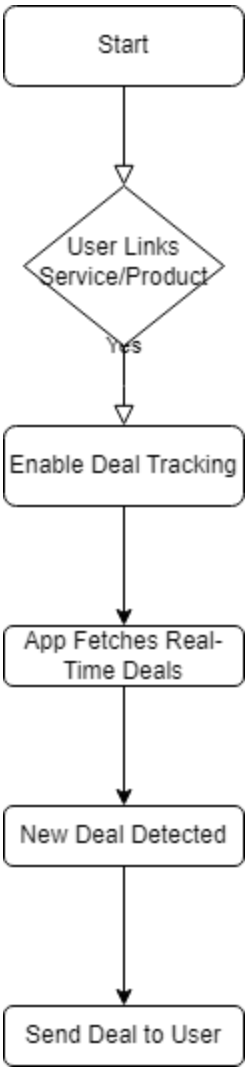
Subscriptions



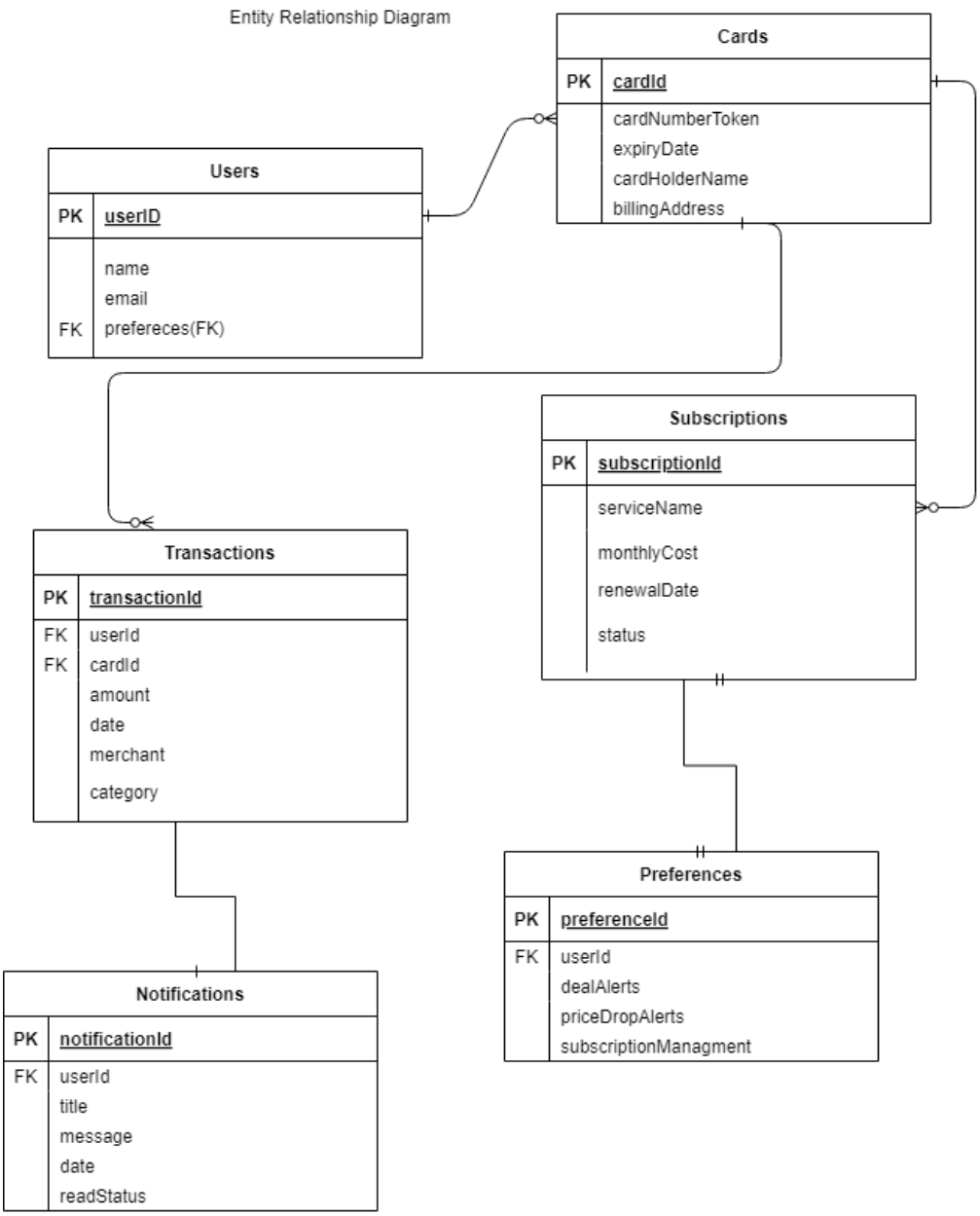
Notifications




Start
User Opens App
User Navigates to savings Goals
No
Returned to main menu
Yes
User Sets Goals
App Tracks Progress
User Achieves Goals
No
Send Reminder Notif
Send Success Notif
End


Start
User Links Service/Product
Yes
Enable Deal Tracking
App Fetches Real-Time Deals
New Deal Detected
Send Deal to User

ER Diagram



Constraints and Assumptions:

1. API Availability: Google Shopping and other APIs are assumed to remain available during development.
2. User Security: Users expect secure handling of financial data and payment processing.
3. Skill Level: Beginner-level proficiency in Kotlin and Jetpack Compose may slow development.
4. API/AI Integration: Successful integration of APIs and AI is assumed but may introduce technical challenges.

5. Platform Limitations: Firebase scalability could be an issue as the user base grows.
6. Timeline: Smooth integration of frontend, backend, and AI features is essential for meeting the project timeline.

Risks and Contingencies:

1. Data Privacy: Potential security breaches; ensure encryption and compliance with PCI-DSS.
2. API Dependency: Outages or changes in APIs may disrupt functionality.
3. Scalability: Increased user load could strain Firebase; need for optimization.
4. AI Accuracy: Inaccurate insights if models aren't updated regularly.
5. User Engagement: Users may disengage without effective notifications and rewards.
6. Network Issues: Poor connectivity may slow app performance.