CS501 Project Proposal — DealTracker

GitHub: https://github.com/buChloeCLY/CS501-Project-DealTracker

App Concept

DealTracker is an intelligent shopping comparison application designed to help users make informed and cost-effective purchasing decisions. By integrating with multiple e-commerce platforms, the app provides real-time price comparisons, price alerts, and personalized product recommendations. DealTracker consolidates the fragmented online shopping experience into a single, efficient, and user-friendly interface, ultimately saving users time and money.

Primary Use Case

1. **Discovery & Search:** The user opens the DealTracker app and searches for a specific product (e.g., Bluetooth headphones).
2. **Comparison & Analysis:** DealTracker fetches real-time prices and availability data from multiple e-commerce platforms (e.g., Amazon, Walmart) via APIs, and presents the results in a clear, sortable list by price, rating, or availability.
3. **Tracking & Notification:** The user adds a desired product to their "Wishlist" and sets a target price. DealTracker automatically monitors the product and sends a push notification the moment the price drops to the desired level.
4. **Personalized Recommendation:** DealTracker analyzes Wishlist and browsing history to suggest relevant products aligned with the user's interests.

Target Users

DealTracker is designed for:

- **Budget-Conscious Shoppers** who frequently compare prices before making a purchase.
- **Efficient Online Shoppers** who shop across multiple platforms and want an efficient way to track deals.

Problem Being Solved

The price comparison experience of current online shopping is fragmented and inefficient, leading to several key problems for consumers:

1. Prices for the same product often vary across platforms, manually comparing prices is time-consuming and inefficient.
2. Users easily miss limited-time deals.
3. Generic deal platforms pushes irrelevant discounts to users, lacking personalization.

DealTracker addresses these pain points by automatic price comparisons, timely deal alerts, and personalized recommendation systems, thereby improving the efficiency of online shopping and helping users save money and time with minimal effort.

Core Features

1. Home Page – Search / Category Browsing / Today's Deals

Functions

1. Global product search (by keyword, brand).
2. Category browsing (electronics, beauty, home goods, food, fashion, etc.).
3. “Deals of the Day” section sorted by 24-hour price drop or historical low.
4. Filters: price range, platform, free shipping, seller type, rating, stock availability.

APIs and Sensors

- Product search and metadata
  - Option A – Official APIs: Amazon Product Advertising API, Best Buy Developer API, Walmart Developer API
  - Option B – Aggregator APIs: Rainforest API / SerpAPI / DataForSEO (for multi-platform search results)
- Historical low-price / trend data: Maintain our own price history database and update it periodically.
- Sensors: Microphone for voice-based product search.

2. Real-Time Cross-Platform Price Comparison

Functions

1. Unified product details (specifications, brand, attributes, gallery).
2. Real-time prices across multiple e-commerce sites.
3. Direct “Buy Now” deep links to each platform.
4. Historical price chart showing daily price changes and lowest points.

APIs and Sensors

- Product details and pricing aggregation
  - Use Amazon / Best Buy / Walmart APIs or aggregators like SerpAPI.
  - Keepa API for Amazon price-history timelines.
- Tax estimation: Avalara Tax API (or simplified built-in rules).
- Sensors: GPS for location-based tax and shipping estimates.

3. Price Alerts and Wishlist

Functions

1. Wishlist: Collect items that you hope will be discounted.
2. For each product, users can set a target price.
3. Receive in-app or push notifications when a price falls below the target.
4. Provides “monitoring frequency” (high/medium/low) and “price anomaly filtering” to exclude clearance and unreliable third-party sellers.

APIs and Sensors

- Push notifications: Firebase Cloud Messaging (FCM) / OneSignal.

4. User Profile – Settings / Browsing History

Functions

1. Account and Preferences: Includes user’s state/city (used for tax), and frequently used e-commerce platforms.
2. Activity Data: Keeps track of browsing history, subscribed price-drop alerts, and a history of triggered notifications.

APIs and Sensors

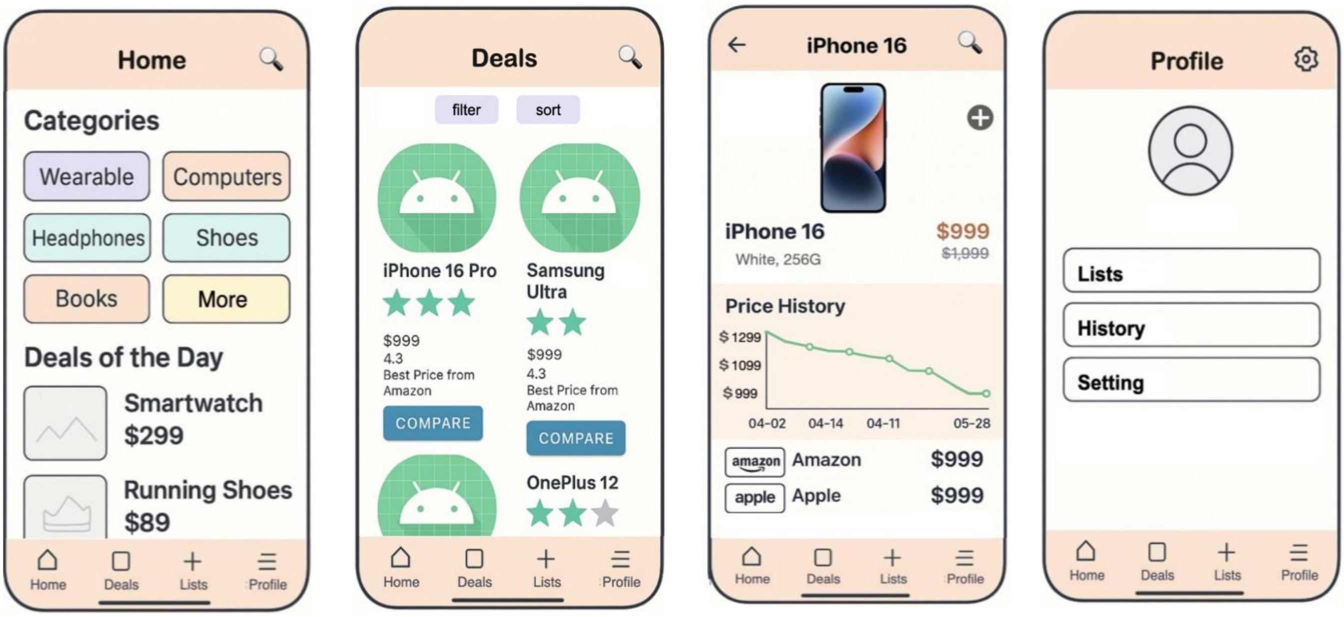
- Account / Authentication: Firebase Authentication or OAuth.

- Storage: Cloud storage via Firestore; local caching via Room Database.

Stretch Goals

1. **Smart Coupon Integration:** Automatically collect and recommend discount codes or coupons based on user browsing and shopping patterns using machine learning and weighted relevance scoring.
2. **Social Sharing:** Allow users to share good deals directly to social media or messaging apps using Android's Intent-based sharing API.

Rough Navigation Map



Team Member Roles and Responsibilities

Team Member	**Primary Responsibilities**
Leyan Chen	Responsible for project coordination, core application logic, and database management, ensuring consistency between modules.
Yijia Chen	Responsible for integrating and managing APIs and sensors, testing applications to ensure availability and compatibility.
Juling Fan	Responsible for UI and UX design, application navigation structure, and ensuring adherence to Material Design guidelines.