

BDIC Intelligent Second-hand Trading Platform BETA

GROUP2: Cao Xinghang, Liu haopu, Qian Hongyao

December 5, 2025



1 Project Background and Overall Objectives

1.1 Motivation and Background

- **Target users:** Students of the Beijing University of Technology – BDIC (Sino-foreign joint program).
- **Real-world pain points:**
 - At the beginning of each semester, second-hand textbook information is scattered across group chats and social media, making matching and searching inefficient.
 - Freshmen are not familiar with course arrangements and textbook editions, which often leads to buying the wrong books or purchasing unnecessary ones.
 - Prices of second-hand textbooks are not transparent and there is no unified platform to aggregate supply and demand.
- **Design goals:**
 - Provide a one-stop platform for textbook trading and exchange.

- Offer intelligent textbook recommendations tailored to BDIC course timetables and CET exam seasons.
- Lower the threshold for posting and searching textbooks, and improve matching efficiency.

1.2 Overall Objectives and Key Features

- **Overall objective:** Build an Android-based second-hand textbook trading platform that integrates posting, browsing, searching, intelligent recommendation, chat and exchange matching.
- **Key features:**
 - **Intelligent recommendation:** Parse student ID to infer entry year and major, then combine this with current month and timetable to generate explainable textbook recommendations.
 - **AI camera recognition:** Use CameraX + ML Kit / Baidu AI to recognize item categories and estimate value, reducing friction when posting items.
 - **Bilingual UI:** All texts are centralized in an `AppStrings` structure to support both Chinese and English interfaces.
 - **Extended features:** Achievement system, price alerts, chat system and more to increase user engagement.



2 System Architecture and Technology Stack

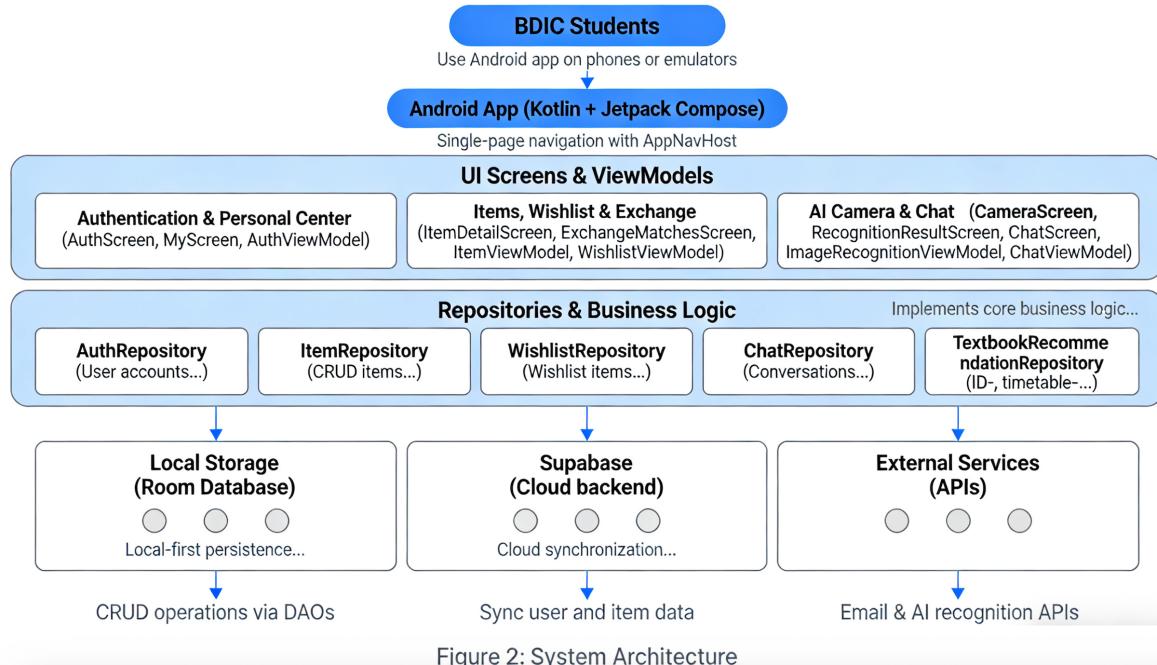
2.1 Overall Architecture

- **Client:** Android application implemented in Kotlin with Jetpack Compose for UI and interaction logic.
- **Local storage:** Room is used to store items, wishlists, timetable courses, chat messages, user achievements and other data.
- **Cloud services:** Supabase is used to store user data, item data and item images, enabling synchronization across devices.
- **External services:**
 - Email service: `EmailVerificationService` sends verification codes for user registration.
 - AI recognition: CameraX provides camera preview; local ML Kit and Baidu AI are used for image recognition.

2.2 Main Module Decomposition

- **Authentication and user module:**
 - `AuthRepository`: Handles register, login, logout, email verification, student ID and simulated month.
 - `AuthScreen`: UI for login, registration and verification code input.
- **Item and posting module:**
 - `ItemRepository`: CRUD operations for items, image upload and Supabase synchronization.
 - `ItemListScreen`, `ItemDetailScreen`, and the post-item screen.
- **Wishlist and exchange matching module:**
 - `WishlistRepository`, `WishlistViewModel`, `WishlistDao`.
 - `ExchangeMatchesScreen`: Shows matching pairs between “what I want” and “what others are selling”.
- **Intelligent recommendation module:**
 - `StudentIdParser`: Parses entry year and major from BJUT student IDs.

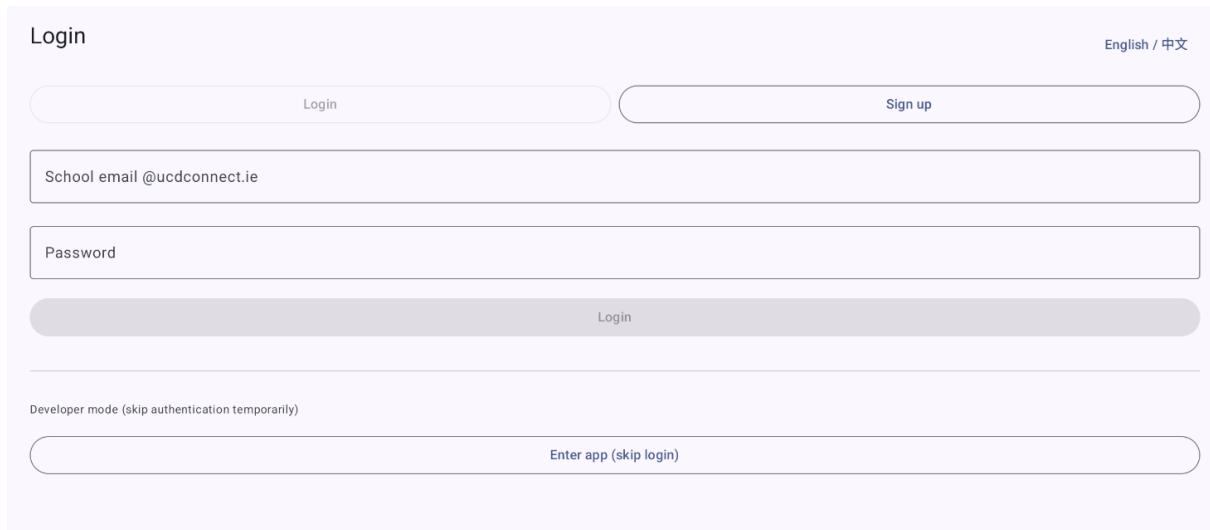
- **TimetableRepository:** Provides BDIC timetable-based course data.
- **TextbookRecommendationRepository:** Implements ID-, time- and wishlist-based textbook recommendation.
- **Chat and messaging module:**
 - **ChatRepository, ChatViewModel, ChatMessageDao.**
 - **ChatScreen:** One-to-one chat; the “My Messages” tab in MyScreen shows recent conversations.
- **Achievements and price alert module:**
 - **AchievementRepository, AchievementScreen, UserAchievementDao.**
 - **PriceAlertService:** Checks wishlist prices on app startup and sends local notifications if conditions are met.



3 Core Business Features I: Users and Items

3.1 Login, Registration and Email Verification

- **Account system:** Supports email-based registration, login and logout, with mandatory email verification.
- **Verification code workflow:**
 - During registration, `EmailVerificationService` generates a 6-digit code that is valid for 30 minutes.
 - The code is sent to the user's mailbox via a real email service API.
 - `AuthRepository` stores pending user info and checks the verification code and its expiry.
 - Users can request resending the code; expired codes are rejected with clear error messages.
- **Bilingual login UI:** All text labels are centralized in `AppStrings`, which supports both Chinese and English.



3.2 Personal Center and Settings (MyScreen)

- **Entry:** The “My” tab in the bottom navigation.
- **Main content:**
 - Tabs include: My Sold Items, My Bought Items, My Wishlist, My Messages, Settings.

- In Settings, users can change their password, configure student ID, set a developer simulated month and switch language.
- Student ID and simulated month are stored by `AuthRepository` in Shared Preferences and are used by the recommendation engine.

3.3 Item Management: CRUD and Exchange Matching

- **Posting items:**
 - From the item list screen, users can navigate to a post-item screen.
 - They fill in title, description, price, category and upload item images.
 - `ItemRepository` writes data into the local Room database and uploads images to Supabase.
- **Browsing and management:**
 - The item list screen shows available textbooks; users can open item details, add to wishlist or contact the seller.
 - Sellers manage their own listings under the “My Sold Items” tab (view, delete or update).
- **Exchange matching:**
 - `ExchangeMatchesScreen` computes a match score between the user’s wishlist and others’ items.
 - Matching pairs (“what I want” vs. “what the other user sells”) are displayed, and users can start a chat from there to negotiate a swap.

Post item

Title*

Price* (number)

Item category (optional)

Electronics

Clothing & accessories

Books & stationery

Furniture & appliances

Sports & fitness

Beauty & skincare

Food & beverages

Toys & models

Description

Item story (optional)

Phone number*

4 Core Business Features II: Intelligent Recommendation and AI Camera

4.1 ID- and Time-driven Textbook Recommendation

- **Student ID parsing:** `StudentIdParser` extracts entry year and major code (e.g., SE, FIN) from an 8-digit student ID.
- **Timetable data:** `TimetableRepository` provides BDIC timetable-based course data for the current and next term.
- **Time and weighting:**
 - Based on the current month, the system determines which term is the “current term” and which is the “preview/next term”, and assigns different weights.
 - During CET preparation seasons (Feb–May and Aug–Nov), CET-related textbooks are given higher weights.
- **Wishlist personalization:**
 - The engine loads the current user’s wishlist and extracts both direct item ID matches and keyword features from titles, categories and descriptions.
 - `TextbookRecommendationRepository` combines timetable/CET scores with personalization scores to compute a final ranking.
- **Explainable recommendation:**
 - Each result is wrapped in a `RecommendedItem` that carries one or more reason types (current term, preview term, CET season).
 - On the home screen, recommendations are grouped and labeled by these reasons, making it clear why each item is recommended.



4.2 Home Screen Presentation and Study Task Card

- **Grouped presentation:**

- In HomeTab, recommended items are grouped into three horizontal lists (LazyRows): current term textbooks, preview/next-term preparation and CET exam zone.
- Users can horizontally scroll within each group and tap items to open details.

- **Study task summary card:**

- A “This month’s study focus” card is displayed above the recommendation blocks.
- Depending on whether current-term, preview and CET recommendations exist, the card shows corresponding English (and Chinese) guidance texts.

- **Interaction design:**

- A “Refresh recommendations” button allows users to re-load recommendations on demand.
- When there are no recommendations, the UI shows a hint and guides the user to fill in their student ID or complete their wishlist.

4.3 AI Camera Recognition and Recommendation

- **Camera functionality:**

- **CameraScreen** uses CameraX to provide real-time preview and photo capture.
 - Users can open the camera from the item list or personal center to quickly capture an item.
- **Multiple recognition backends:**
 - The app supports both local ML Kit and cloud-based Baidu AI as recognition backends, selectable from the UI.
 - When the cloud API keys are not configured, the app automatically falls back to on-device recognition for robustness.
- **Recognition results and follow-up recommendation:**
 - **RecognitionResultScreen** displays the captured image, recognized labels/categories and confidence scores.
 - Based on recognition results, the app recommends similar items and allows the user to either view existing listings or create a new listing directly.



5 Extended Features, Technical Highlights and Summary

5.1 Extended Features: Achievements, Price Alerts and Chat

- **Achievement system:**
 - `AchievementRepository` tracks user behaviors such as number of posts and wishlist interactions, and unlocks achievements accordingly.
 - `AchievementScreen` displays unlocked and locked achievements; some achievements trigger local notifications as rewards.
- **Wishlist price alerts:**
 - `PriceAlertService` runs on app startup, reading the user's wishlist from Room.
 - When an item's price falls below a configured threshold, the service sends a system notification to remind the user.
- **Chat system:**
 - `ChatRepository`, `ChatViewModel` and `ChatMessageDao` provide persistent one-to-one chat.
 - `ChatScreen` supports conversations initiated from an item detail page or from the conversation list.
 - The “My Messages” tab in `MyScreen` shows recent conversations and allows users to quickly continue chatting.

5.2 Technical Highlights and Engineering Practices

- **Explainable recommendation:** Each recommended item is tagged as “current term”, “preview/next term” or “CET season”, which makes the reasoning behind recommendations transparent to users and instructors.
- **Multi-source data fusion:** The engine combines student ID → entry year and major, timetable courses, current month and wishlist data to build BDIC-specific personalized textbook recommendations.
- **Multiple AI backends:** The app supports both on-device ML Kit and cloud-based Baidu AI for image recognition, balancing offline usability and accuracy.
- **Local-first with cloud sync:** Room is used as the primary local data store, while Supabase serves as a cloud backup and sharing layer.

- **Modular and extensible design:** Authentication, items, wishlist, chat, achievements and recommendation modules are decoupled, making the system easier to maintain and extend.

5.3 Problems and Solutions

- **Problems Encountered During Development:**
 - The Gradle JDK path is incompatible across different operating systems, causing build failures. The issue was resolved by removing the Windows-specific path and reconfiguring the local JDK.
 - Some Compose components (such as LazyRow) had missing imports, resulting in compilation errors. Later, the IDE prompted for completion and unified management of imports was implemented.
 - The initial sample data covered a limited number of grades and courses. Subsequently, by extending `SampleItemSeeder`, the mainstream courses of BDIC and CET-related textbooks were covered.