Campus-transaction

An Android Studio-based campus second-hand trading platform APP, providing convenient trading services for idle items to campus users.

Project Introduction

During graduation season, many graduates choose to sell their idle items. This project aims to provide a centralized second-hand trading platform for campus users, enabling buyers and sellers to complete transactions more efficiently and improve the circulation rate of idle items.

Features

User System

- User registration and login
- Personal information management (name, major, phone, QQ, address)
- Password modification
- Logout

Product Trading

- Product posting (supports images, title, price, description, contact information)
- Product browsing (displayed in chronological order)
- Product search (supports title and description search)
- Product category filtering (sports goods, daily necessities, electronics, study materials)
- Product detail viewing
- Product favorites function
- Product status management (for sale/sold)
- My posted products management
- Sold products viewing
- My favorites viewing

Social Interaction

- Product comment function
- Real-time chat messaging (private communication between buyer and seller)
- Chat list management
- Unread message reminders

Other Features

- Feedback (supports feature suggestions, bug reports, etc.)
- About Us

Tech Stack

Category	Technology
Programming Language	Java 17
Development Tool	Android Studio

Category	Technology
Build Tool	Gradle 8.7.3
Target SDK	Android 14 (API 34)
Minimum SDK	Android 5.0 (API 21)
Database	SQLite
UI Components	Material Design Components

Main Dependencies

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'androidx.core:core:1.12.0'
    implementation 'com.google.android.material:material:1.11.0'
    implementation 'androidx.cardview:cardview:1.0.0'
}
```

Project Structure

```
Campus-transaction/
├── app/
│   ├── src/main/
│   │   ├── java/page/page1/           # Java source code
│   │   │   ├── LoginMainActivity.java      # Login page
│   │   │   ├── RegisterMainActivity.java   # Registration page
│   │   │   ├── main_page.java              # Homepage (product list)
│   │   │   ├── item_info.java              # Product details
│   │   │   ├── AddItem.java                # Post product
│   │   │   ├── MyItems.java                # My posts
│   │   │   ├── MySoldItems.java            # Sold products
│   │   │   ├── kind_page1~4.java           # Category pages
│   │   │   ├── ChatActivity.java           # Chat page
│   │   │   ├── ChatListActivity.java       # Chat list
│   │   │   ├── MyselfActivity.java         # Personal center
│   │   │   ├── FeedbackActivity.java       # Feedback
│   │   │   ├── DatabaseHelper.java         # Database management
│   │   │   └── ...
│   │   ├── res/
│   │   │   ├── layout/                # Layout files
│   │   │   ├── drawable/              # Icons and background resources
│   │   │   ├── values/                # Strings, colors, styles
│   │   │   └── mipmap-*/              # App icons
│   │   └── AndroidManifest.xml        # App manifest
│   └── build.gradle                   # Module-level configuration
├── gradle/                            # Gradle Wrapper
├── build.gradle                       # Project-level configuration
└── settings.gradle                    # Gradle settings
```

Database Design

The project uses SQLite as the local database, with the current database version being **5**.

Table Structure

1. Users Table (users)

Field	Type	Description
userId	VARCHAR(20)	User account (Primary Key)
passWord	VARCHAR(20)	Password
name	VARCHAR(20)	Name
subject	VARCHAR(20)	Major
phone	VARCHAR(15)	Phone
qq	VARCHAR(15)	QQ number
address	VARCHAR(50)	Address

2. Product Information Table (iteminfo)

Field	Type	Description
id	INTEGER	Product ID (Primary Key, Auto-increment)
userId	VARCHAR(100)	Publisher account
title	VARCHAR(200)	Product title
kind	VARCHAR(100)	Product category
info	VARCHAR(1000)	Product description
price	VARCHAR(100)	Product price
image	BLOB	Product image
time	DATETIME	Publishing time
contact	VARCHAR(50)	Contact information
status	INTEGER	Status (0 for sale, 1 sold)

3. Comments Table (comments)

Field	Type	Description
userId	VARCHAR(100)	Commenter account
itemId	INTEGER	Product ID
comment	VARCHAR(1000)	Comment content
time	DATETIME	Comment time

4. Messages Table (messages)

Field	Type	Description
id	INTEGER	Message ID (Primary Key, Auto-increment)
senderId	VARCHAR(100)	Sender ID
receiverId	VARCHAR(100)	Receiver ID
content	VARCHAR(2000)	Message content
time	DATETIME	Sending time
isRead	INTEGER	Read status (0 unread, 1 read)

5. Conversations Table (conversations)

Field	Type	Description
id	INTEGER	Conversation ID (Primary Key, Auto-increment)
userId1	VARCHAR(100)	User 1
userId2	VARCHAR(100)	User 2
lastMessage	VARCHAR(500)	Last message
lastTime	DATETIME	Last message time
unreadCount	INTEGER	Unread message count

6. Feedback Table (feedback)

Field	Type	Description
id	INTEGER	Feedback ID (Primary Key, Auto-increment)
userId	VARCHAR(100)	User ID
type	VARCHAR(50)	Feedback type
content	VARCHAR(2000)	Feedback content
contact	VARCHAR(100)	Contact information
time	DATETIME	Submission time
status	INTEGER	Processing status (0 pending, 1 processed)

7. Favorites Table (favorites)

Field	Type	Description
id	INTEGER	Favorite ID (Primary Key, Auto-increment)
userId	VARCHAR(100)	User ID
itemId	INTEGER	Product ID
time	DATETIME	Favorite time
-	UNIQUE(userId, itemId)	Composite unique constraint

Installation and Running

Environment Requirements

- Android Studio Arctic Fox or higher
- JDK 17
- Android SDK 34
- Gradle 8.7.3

Running Steps

1. Clone the project locally

```
git clone https://github.com/your-username/Campus-transaction.git
```

2. Open the project with Android Studio

3. Wait for Gradle sync to complete

4. Connect Android device or start emulator

5. Click Run to run the project

Testing Results

Test Overview

Test Application	CampusSwap
System Platform	Android
Device Count	41
Unexecuted Devices	1
Test Pass Rate	77.5%

Detailed Test Results

Test Result	Test Terminals	Percentage
Installation Failure	9	22.5%
Startup Failure	0	0.0%
Monkey Failure	0	0.0%
Uninstall Failure	0	0.0%
Runtime Failure	0	0.0%
Passed	31	77.5%

Test Device List

Unexecuted (1 device)

Installation Failed (9 devices)

- realme X50 Pro
- OPPO Find X2 Pro
- Reno7 SE
- 一加 9 Pro
- OPPO A55
- 一加 8T
- OPPO Reno6 Pro
- OPPO A32
- vivo S5

Passed (31 devices)

- 荣耀Play6C
- HTC U11+ (全网通)
- 摩托罗拉 Z Play Droid
- Redmi Note 12 Turbo
- OPPO A1X
- 荣耀 50 SE
- 荣耀畅玩40C
- 华为 nova
- 荣耀 Magic3
- 华为 nova
- iQOO Neo5S
- 一加 6
- iQOO Neo5S
- realme V11
- 三星 Galaxy S6 Edge+
- 摩托罗拉 Z Play Droid
- 红米K60E
- OPPO A1X
- 努比亚 Z17 S
- OPPO Reno9 Pro
- 荣耀 50
- 魅蓝3
- 魅族 PRO 7
- 红米Note 11T Pro

Performance Metrics Overview

	Installation Time(s)	Startup Time(s)	CPU Usage(%)	Memory Usage(MB)	Battery Temperature(°C)	Network Traffic(MB)	FPS
Average	7.96	1.74	0.52	57.1	28.38	0.00	57.0
Peak	1.97	0.36	0.18	24.77	22.0	0.00	60.0
(Device Model)	Redmi Note 12 Turbo	魅族PRO 7 Plus	荣耀 Magic3	魅蓝3	OPPO Ace2	荣耀Play5T Pro	荣耀畅玩40C

Application Architecture

```
┌─────────────────────────────────────────────────────────┐
│                     UI Layer                            │
│  (Activities + XML Layouts + Adapters)                  │
├─────────────────────────────────────────────────────────┤
│                   Business Logic                        │
│  (Activity internal business logic processing)          │
├─────────────────────────────────────────────────────────┤
│                   Data Layer                            │
│  (DatabaseHelper + SQLite)                              │
└─────────────────────────────────────────────────────────┘
```

Core Function Implementation

User State Management

Using static variable `LoginMainActivity.post_userid` to store the currently logged-in user, implementing simple session management.

Image Storage

Product images are stored directly in the SQLite database in BLOB format, simplifying file management.

Real-time Chat

Implementing "real-time" message updates through database polling with a 3-second interval timer.

Page Navigation

Using Intent for page navigation, passing necessary data parameters through `putExtra()`.

Product Categories

Category	Description
Sports Goods	Sports equipment, fitness devices, etc.
Daily Goods	Daily necessities, household items, etc.
Electronics	Phones, computers, digital devices, etc.
Study Materials	Books, stationery, study materials, etc.

Future Plans

- Introduce backend server for truly real-time communication
- Add product search functionality
- Support multi-image upload
- Add favorites functionality
- Introduce user rating system
- Add message push notifications
- Support WeChat/QQ third-party login