Campus Social App 📱

Campus Social App is an innovative mobile application for modern university campus ecology, which provides a cross-lingual and high-efficiency campus information exchange platform for students via NFC fast authentication, real-time Chinese/English translation and intelligent content recommendation.

This app not only solves the problem of information fragmentation, but also helps students to access learning resources, discover academic activities and share learning experiences, as well as provides practical information on life such as lost and found, campus food and accommodation, and provides students with channels for socialising and communication. For schools, this app can help with better news broadcasting, events promotion and enhancing students' sense of belonging and engagement on campus, making it particularly suitable for modern universities with a high degree of internationalisation and a diverse student body.

This project demonstrates the potential of mobile technologies in creating connected & interactive campus environments by providing an innovative platform for student communication and content discovery.

Link to GitHub Repository

GitHub Repository - https://github.com/ucl-casa-ce/casa0015-mobile-assessment

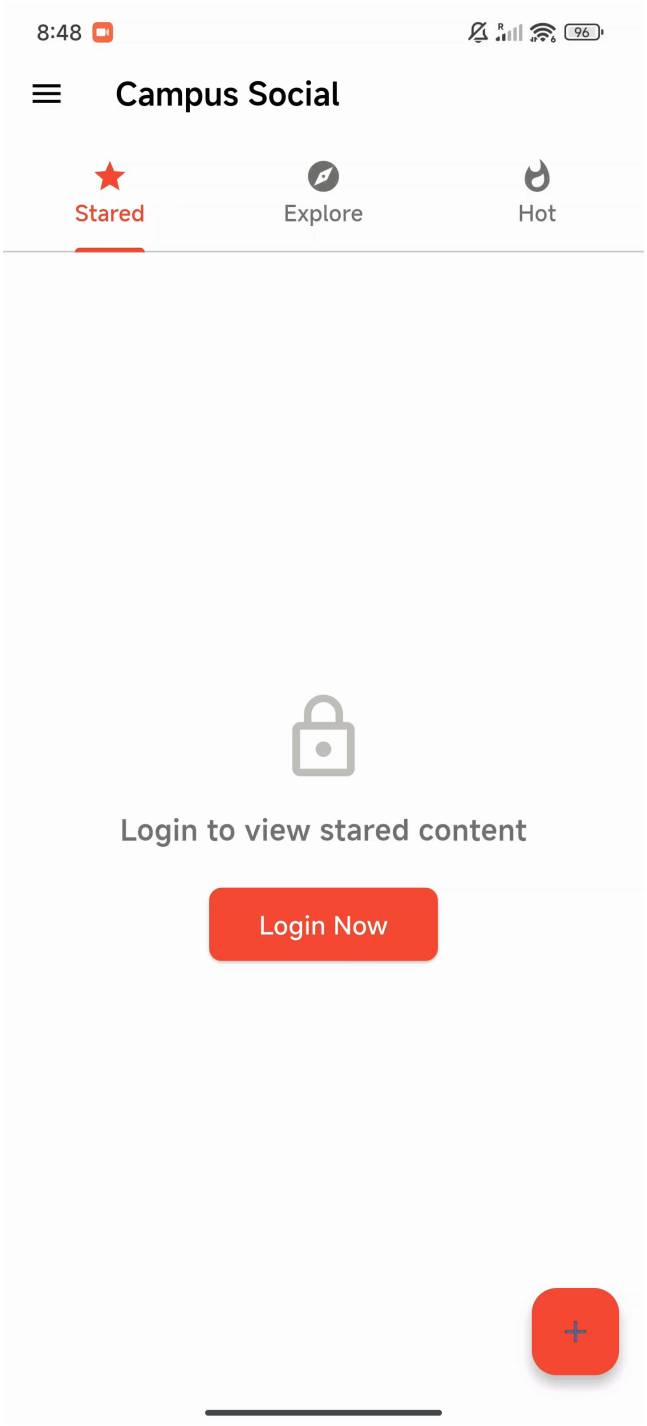
✨ Key Features

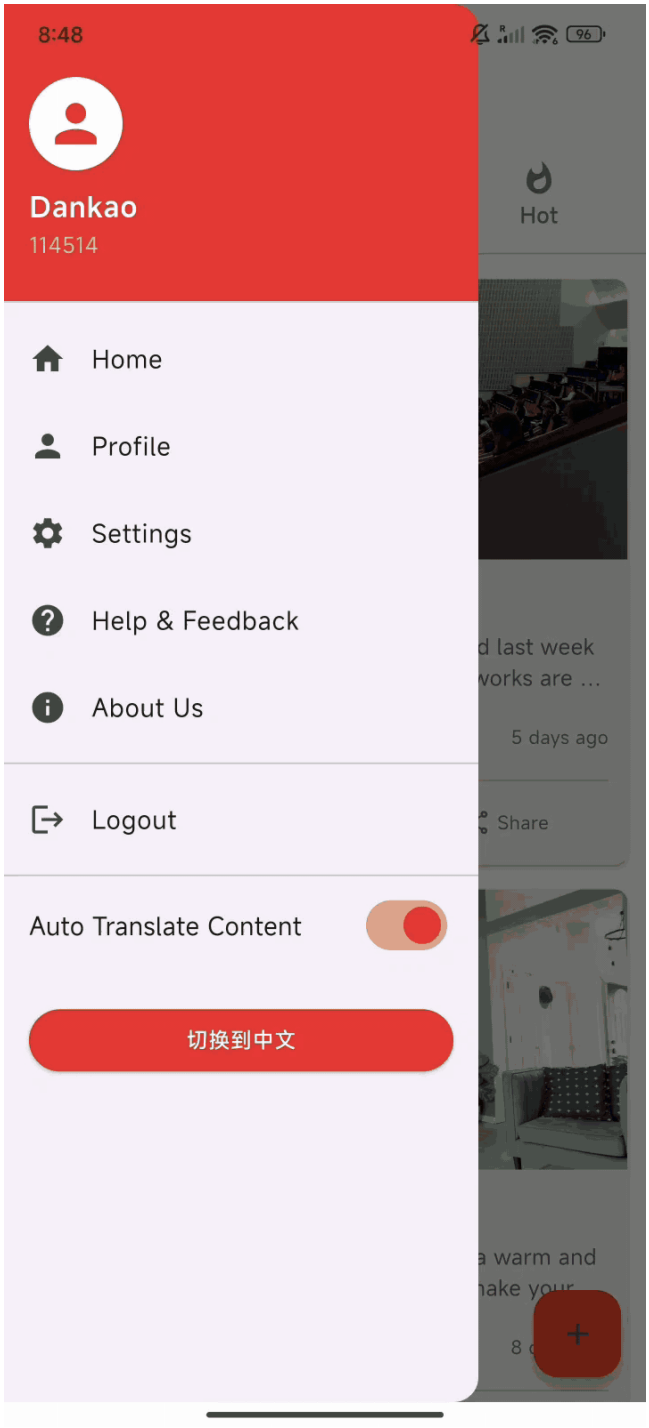
- **NFC Authentication**: Secure login using campus NFC cards
- **Bilingual Support**: Automatic translation between Chinese and English
- **Content Categories**: Organized posts by topics (Study, Activities, Lost & Found, Food, Accommodation)
- **Shake-to-Refresh**: Unique content update gesture
- **Social Interactions**: Like, comment, and share posts within the community

📱 Screenshots

The screenshots offer a clear and intuitive glimpse of the key features:

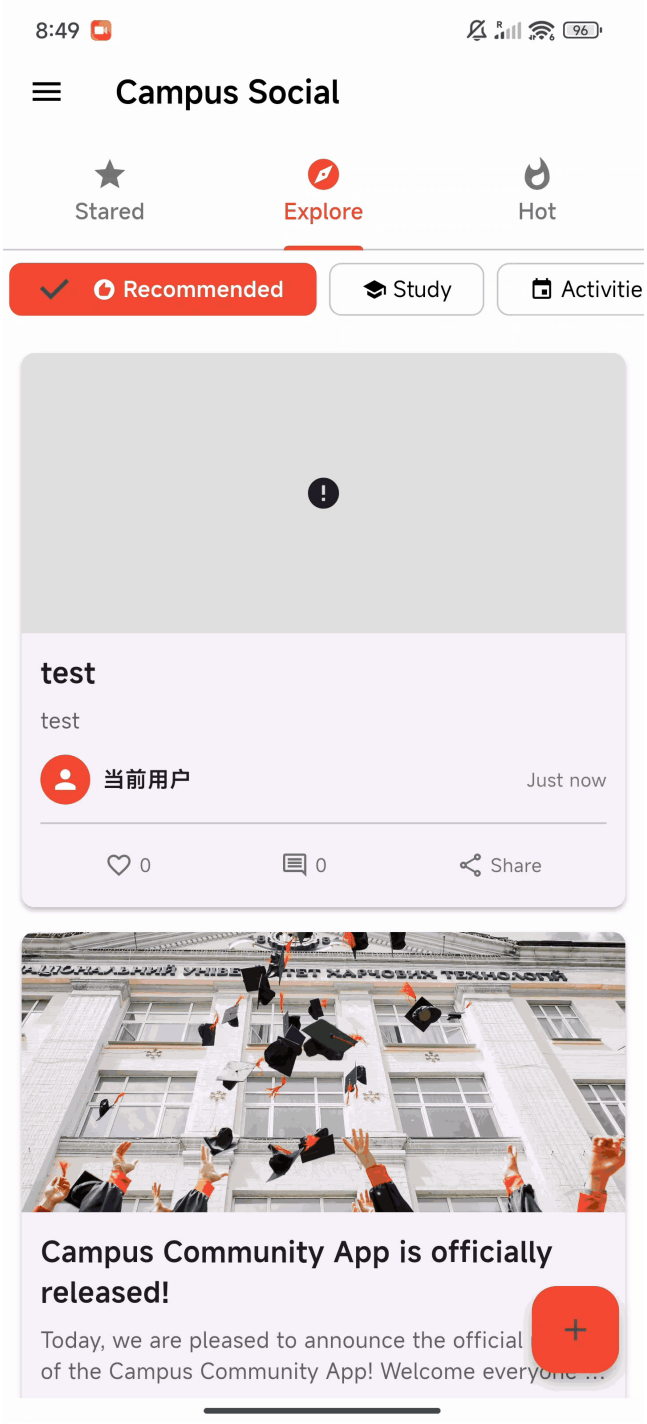
NFC Quick Login	Smart Bilingual Support

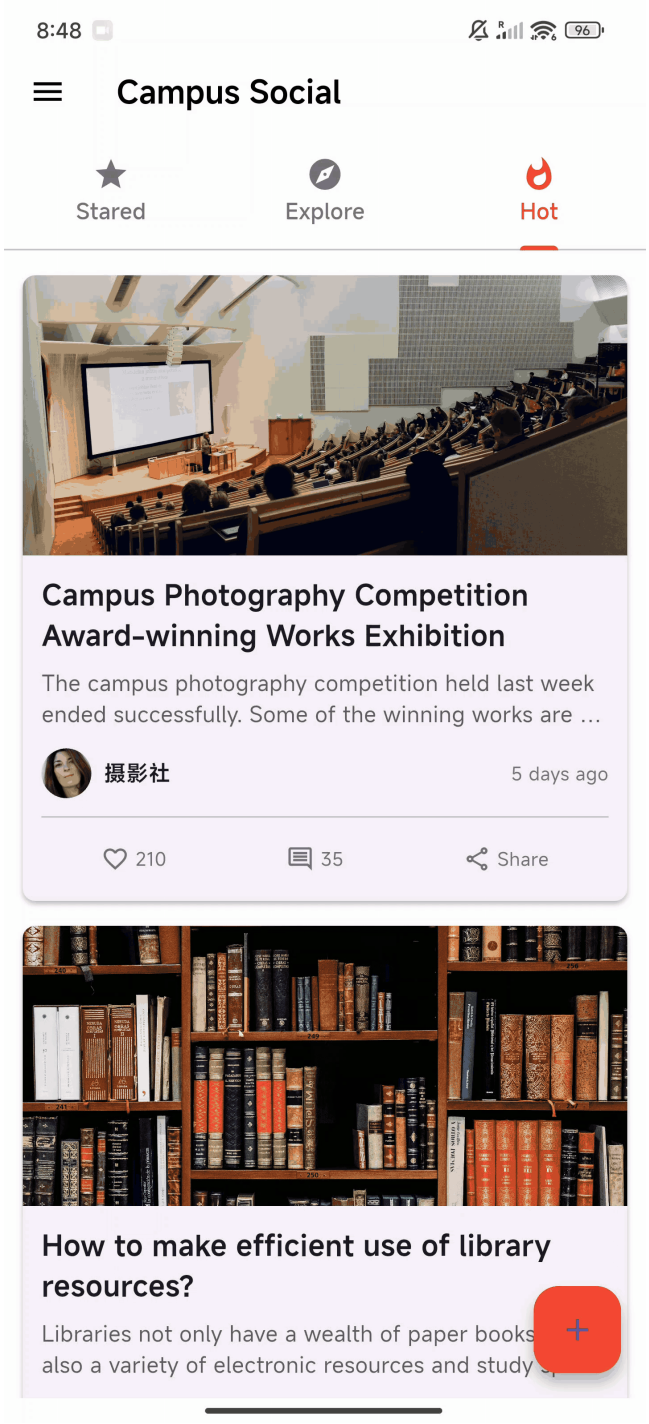




- Campus Card Verification
- One-Tap Login
- Fast and Seamless
- Real-Time Translation
- Instant Language Switching
- Smart Adaptation

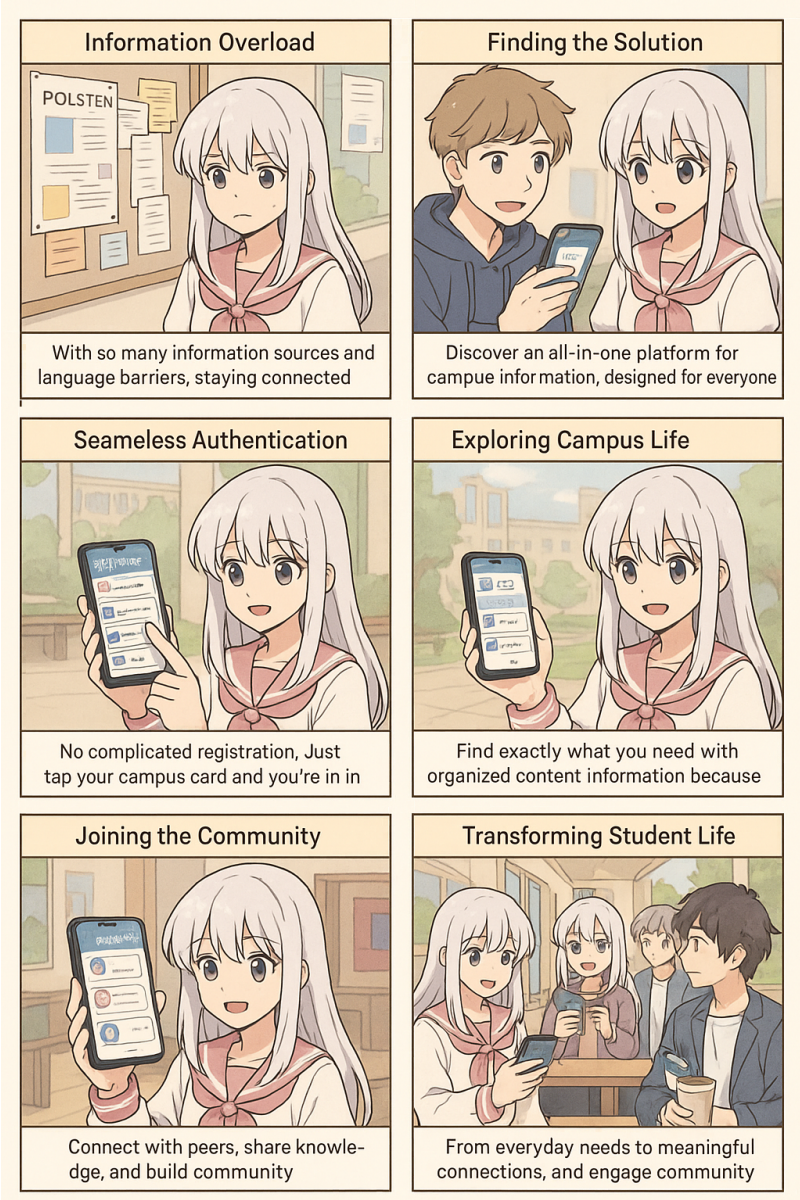
Shake & Pull to Refresh	Content Publishing Test





- 🔶 Shake to Refresh
- 🔶 Pull Down to Update
- 🔶 Instant Feedback
- 🔶 Quick Post Creation
- 🔶 Rich Media Support
- 🔶 Interactive Engagement

📖 Story Board



📽️ Demo Video

Watch Demo Video

🌐 Internationalization

The app supports Chinese and English with automatic content translation:

- UI elements translated via `.arb` files
- Content translation powered by Google Cloud Translation API
- Language preference saved locally

🔐 Authentication

The app uses NFC-based authentication:

1. User taps their campus NFC card
2. App reads the card UID

3. Authentication service validates the card
4. User session is created

🛠️ Technology Stack

- **Framework**: Flutter 3.6.1+
- **State Management**: Provider
- **Authentication**: NFC Manager
- **Networking**: HTTP/Dio
- **Local Storage**: Shared Preferences
- **UI Components**: Material Design, Cached Network Image
- **Internationalization**: Flutter Localizations
- **Image Handling**: Image Picker, Permission Handler

📦 Development Guidelines

Our project adheres to modern Flutter development best practices with a focus on maintainability, performance, and user experience. This section provides comprehensive guidance for setting up the development environment, understanding our project architecture, and contributing to the codebase. Whether you're joining as a new developer or maintaining existing features, these guidelines ensure consistent code quality and seamless collaboration across the team. The following subsections cover installation procedures, project structure details, and contribution workflows to help you get started efficiently.

🚀 Installation

The latest release can be found at build/app/outputs/flutter-apk/app-release.apk

Prerequisites

- Flutter SDK: >=3.6.1
- Dart SDK: >=3.0.0
- Android Studio / VS Code with Flutter extensions
- Android SDK 21 or higher (for Android)
- iOS 12.0 or higher (for iOS)

Setup Instructions (Android Studio Suggested)

1. **Clone the repository**

```
git clone https://github.com/Reikimen/casa0015-ucl-community
```

2. **Install dependencies**

```
flutter pub get
```

3. **Set up environment variables**

   - Copy the

     ```
     .env.example
     ```

     file to

     ```
     .env
     ```

     ```
     cp .env.example .env
     ```

   - Edit the

     ```
     .env
     ```

     file and add your API keys:

     ```
     GOOGLE_TRANSLATION_API_KEY=your_actual_api_key_here
     ```

4. **Run the app**

```
flutter run
```

Environment Configuration

1. **Google Cloud Translation API**
   - Get an API key from Google Cloud Console
   - Enable Google Cloud Translation API in your project
   - Add your API key to the `.env` file (see Setup Instructions)
2. **NFC Configuration**
   - Ensure your device supports NFC
   - Grant NFC permissions when prompted

- Test with supported NFC cards (see docs for compatible card types)

3. **Security Note**
   - Never commit your `.env` file to version control
   - The `.gitignore` file is configured to exclude sensitive files
   - For production deployment, set environment variables directly in your hosting platform

📁 Project Structure

The application follows a clean architecture pattern with a clear separation of concerns. This structure promotes maintainability, scalability, and easy testing.

The app follows a clean architecture pattern with:

- **Providers**: State management for app, user, and post data
- **Services**: API and authentication services
- **Models**: Data models for posts, comments, and users
- **Screens**: UI components organized by feature
- **Widgets**: Reusable UI components

```
lib/
├── main.dart                    # Application entry point
├── app.dart                     # Application configuration and theme
setup
├── routes/                      # Navigation and routing
│   └── app_router.dart         # Route generation and navigation logic
├── providers/                   # State management using Provider pattern
│   ├── app_provider.dart       # Application-wide state (language,
translation)
│   ├── user_provider.dart      # User authentication and session state
│   └── post_provider.dart      # Posts and content management
├── models/                      # Data models
│   ├── user_model.dart         # User data structure
│   ├── post_model.dart         # Post content model
│   ├── comment_model.dart      # Comment data structure
│   └── auth_result.dart        # Authentication result model
├── services/                    # Business logic and external services
│   ├── api_service.dart        # API communication layer
│   ├── auth_service.dart       # Authentication service
│   └── storage_service.dart    # Local storage management
├── screens/                     # UI pages
│   ├── home/                   # Home screen and tabs
│   │   ├── home_screen.dart    # Main home container
│   │   ├── stared_tab.dart     # Following/starred content tab
│   │   ├── explore_tab.dart    # Discovery tab with categories
│   │   └── hot_tab.dart        # Trending posts tab
│   ├── profile/                # User profile
│   │   └── profile_screen.dart # User profile display
│   ├── post/                   # Post-related screens
│   │   ├── post_detail.dart    # Individual post view
│   │   └── post_create.dart    # Create new post screen
│   └── auth/                   # Authentication screens
```

```
│       ├── login_screen.dart   # Login options screen
│       └── nfc_login.dart      # NFC card authentication
├── widgets/                     # Reusable UI components
│   ├── app_drawer.dart         # Navigation drawer
│   ├── post_card.dart          # Post display card
│   ├── subcategory_bar.dart    # Category selection bar
│   └── shake_detector.dart     # Shake gesture detector
├── utils/                       # Utilities and helpers
│   ├── constants.dart          # Application constants
│   ├── helpers.dart            # Helper functions
│   └── config.dart             # Environment configuration
└── l10n/                        # Localization resources
    ├── app_en.arb              # English translations
    └── app_zh.arb              # Chinese translations
```

Providers (State Management)

The application uses the Provider pattern for state management, with three main providers:

- **AppProvider**: Manages application-wide state such as language preferences, auto-translation settings, and theme
- **UserProvider**: Handles user authentication state, NFC login, and user session management
- **PostProvider**: Manages posts, categories, and content interaction logic

Services

Services layer handles external API calls, local storage, and business logic:

- **ApiService**: Centralizes all HTTP requests and API communications
- **AuthService**: Manages authentication flows, including NFC card login
- **StorageService**: Handles persistent local data storage using SharedPreferences

Screens

The screens are organized by feature:

- **Home**: Contains the main navigation structure with three tabs (Stared, Explore, Hot)
- **Profile**: User profile and settings
- **Post**: Post creation, viewing, and interaction
- **Auth**: Login methods including NFC authentication

Internationalization

The app supports both English and Chinese languages:

- Translation files are located in the `l10n` directory
- Uses ARB files for managing translations
- Includes auto-translation feature using Google Translate API

Environment Configuration

The project uses dotenv for environment configuration:

- API keys are stored in `.env` file (not included in the repository)
- Configuration is accessed through the `Config` class in `utils/config.dart`

Configuration Files

- **pubspec.yaml**: Defines dependencies and assets
- **.env**: Stores environment variables (API keys, etc.)
- **android/app/src/main/AndroidManifest.xml**: Android permissions and metadata

🤝 Contributing

We welcome contributions! Please follow these steps:

1. Fork the repository
2. Create your feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

📞 Contact

Project Lead: Dankao Chen

Email: zczqdc2@ucl.ac.uk

GitHub: @Dankao

🙏 Acknowledgments

- UCL CASA for project support
- Flutter community for excellent packages
- Google Cloud for translation services
- All contributors and testers

Declaration of Authorship

Me, Dankao Chen HERE, confirm that the work presented in this assessment is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

29/04/2025

Made with ❤️ by UCL CASA