CS 501 Course Project – JustMoveIn

Team: Haixi Zhang, Jiayin Fan

Work distribution is noted by the words’ background as above.

Architecture and Design

1. Welcome Activity (WelcomeActivity.kt & activity_welcome.xml)

A starter page of the application. When a user open the application, they will see an introduction and choices for logging in or signing up.

2. Log in Activity and Sign up Activity (LogInActivity.kt & activity_log_in.xml, SignUpActivity.kt & activity_sign_up.xml)

Activities for logging in and signing up an user account, which can be navigated to each other. Currently, signing up by Email and Password is supported. User account information is stored in Firebase. Users can self-select whether or not to display their passwords.

3. Main Activity (MainActivity.kt & activity_main.xml)

The main page of the application, including a bottom navigation bar that can navigate to fragments(Housing, Trading and Profile).

4. Trading Features (Housing and Item, Fragments)

1) A recycler view for active trading items (TradeMainFragment.kt, ItemAdaper.kt & fragment_trade_main.xml, item_grid_layout.xml)
    a) A search bar allowing for searching by title. Matching by words.
    b) A filter for categories.
    c) Including two tabs: “Browse” tab for all active items ordered by newest listed first; “Liked” tab for listings liked by the current user.
2) Item/housing detail page (ItemDetailActivity.kt & activity_item_detail.xml)
    a) A “like” button for the user to mark the item as “liked”.
    b) A contact button for showing the information of the seller (ContactSellerActivitty & activity_contact_seller.xml)
        i) For the ContactSeller page, the contact button will navigate the user to the email application. The seller’s email address and the subject line will be pre-filled.
    c) A Share button for sharing the item’s name, price and description by a sharing intent.
    d) If the item is listed by the current user, the user can see a popup menu allowing them to activate/deactivate or delete the listing/
3) Listing a new item/housing page (ListItemActivity.xml & activity_list_item.xml)
    a) Allow for add at most 3 photos; allowing for reselecting photos.
4) Item/Housing data object (item.kt)
    a) likedBy: a mutable map consists a user’s id and a Boolean indicating whether the user likes the item.

   b) sellerId: a string which is the id of the user who lists the item.

5. Profile and Settings

1) A profile main page including user account information and settings (ProfileActivity.kt & activity_profile.xml)
   a) Allowing for uploading user icon.
   b) Allowing for editing and saving the user's information including name, email address and zip code. (ProfilerEditActivity)
2) Your Listings
   a) The current user can see the items listed by them and activate/deactivate or delete the item.
3) Night Mode
4) Notifications
   a) Manage the system's permission with JustMoveIn's notifications.
5) Security and Privacy
6) Language
   a) Language options: English (default) and Japanese
7) User data object (user.kt)

Challenges

1. Real-time Data Synchronization

Ensuring that the user interface reflects real-time data from the database posed synchronization challenges:

- Challenge: It was crucial to keep the UI in sync with the database updates both upstream (when data is sent to the database) and downstream (when data is retrieved).
- Approach: We utilized Firebase Realtime Database's event-driven model, which updates UI components in real-time as data changes occur, ensuring that users always see the most up-to-date information. However, sometimes the UI will only update if the user manually switches the tabs. We have using a lot of debugging logs and still cannot figure this issue.

2. Night Mode Setting

Implementing a uniform night mode across the application presented several challenges:

- Challenge: Integrating day and night modes required careful handling of color schemes to ensure consistency and usability in various lighting conditions.
- Approach: To address this, we adjusted color schemes and refined UI elements specifically for night mode. This involved customizing different colors and themes for both modes to enhance user experience and visual accessibility.

3. Managing Multiple Images for Listings

Adding functionality to upload and display multiple images per listing involved complex image management and UI development:

- Challenge: We needed a robust solution to allow users to upload multiple images for an item, and display these images in a user-friendly, scrollable view.
- Approach: We implemented a dynamic image handling system that stores images in Firebase Storage and retrieves them for display in a ViewPager2, ensuring smooth user interactions and efficient data handling.

4. Associating User Information with Listings

Linking user profiles with their listings required careful design to maintain data integrity:

- Challenge: Each listing needed to be associated with a user's profile without compromising on security and data privacy.
- Approach: We implemented a secure system where each listing stores a reference to the user's ID. This allows us to fetch and display user-related data such as name and contact information only when necessary, using Firebase's security rules to protect user data.

Future Works

1. **Privacy Policy Enhancement**

Review and improve the privacy policy to ensure it complies with the latest data protection regulations, enhancing user trust and security.

2. Further Personalization Features

Develop more personalized user experiences based on user behavior and preferences to enhance engagement and satisfaction.

- "Near You" Feature: Introduce a "Near You" feature to help users find listings based on geographic proximity, enhancing the app's utility for local transactions.
- More Useful Filters: Expand the filtering options to include more specific attributes, allowing users to more efficiently find the listings that meet their needs.

3. Refining the App's Performance and Stability

Continuously monitor and optimize the app's performance to handle larger user loads and improve overall stability.

- Real-time Data Synchronization: Address issues related to real-time data synchronization to ensure that the app remains responsive and up-to-date across all user interfaces.

4. Include Additional Services

- In-Application Messaging: Implement a direct messaging feature within the app to facilitate communication between buyers and sellers without leaving the platform.

- Auto-Generating Screenshots for Sharing: Develop functionality to automatically generate screenshots of listings for easy sharing on social media or other platforms.