GitHub Username: asalfo

Wiugli

Description

Wiugli learns from the users and recommend them Products and Services that may be of interest to them. That is, products and services recommender to help users find what they need. Users can browse the catalogue, rate and review products and services.

Intended User

Customers who love online shopping.

Features

- Rate products and services
- List Products and services
- Manage wishlist
- Recommendations (For you)
- Login/Create account
- Edit Profile
- Nearest products and services (based on user current location)
- Notifications of new recommendation
- App preferences

User Interface Mocks

SIGN IN / REGISTER

ITEMS LIST / DETAIL

Key Considerations

How will your app handle data persistence?

App uses a RESTful API to retrieve information of products and services. A content provider is used to persist the users favorite items, users wishlits localy.

Describe any corner cases in the UX.

Notifications are shown when the user is nearby a wishlist item or new recommendations are ready for her/she.
Everything will be stored in the save instance when onPause is called, so when the user come back to the apps, it will be at the same place where they left out. If the user closed the app, it will go back to main screen when they open it again.

Describe any libraries you'll be using and share your reasoning for including them.

Butterknife to find and automatically cast the corresponding view in your layout.
Cardview, RecyclerView and Design support for the material design.
Picasso to handle the loading and caching of images.
Retrofit for consuming APIs

Describe how you will implement Google Play Services.

This app will use Google map to show items locations to the user

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create a git repo on Github
- Configure libraries
- Configure API endpoint access credentials

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for LoginActivity
- Build UI for RegisterActiviy
- Build UI for ItemDetailActivity
- Build Ui for ForYouActivity
- Build UI for Preferences

Task 3: Create a Content Provider for persistence

Task 4: Configure Retrofit.

Task 5: Configure Google Play Services for Maps

Task 6: Handle Error Cases, control every that can make the app crash.

Task 7 Integration Testing Bug Fixes and Refactor

- Test all functionalities
- Fix bugs
- Refactor code (if necessary)

Task 8: Build, Clean and Sign app

- Create Gradle tasks to build and clean the install Release flavour
- Digitally sign the APK.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"