GitHub Username: poph2

Price Cutz

Description

This app allows consumers and business owners connect through the available offers, coupons, discounts and deals.

This application provides a platform where business owners can create and manage the presentation of their deals to potential customers.

The app presents these coupons/deals to customers based on any combination of the following:
- User interests
- Locations based alerts
- Custom search

The web that supports the application will be a Google App Engine instance.

Features such as barcode scanning provide secure coupon redemption.

Intended User

This application is for anyone that is interested in maximizing his/her savings opportunity by optimally accessing available cost-saving deals.
Also, business owners who are interested in getting more patronage by offering potential customers deals.
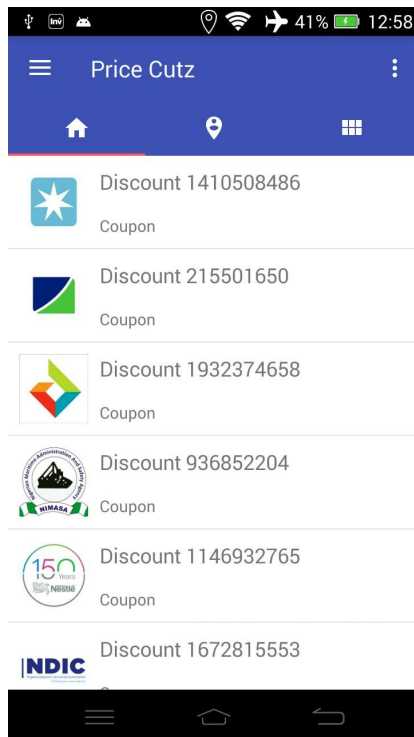
Features

List the main features of your app. For example:
- Deals by location
- Deals by filter
- Custom deal search
- Barcode scanning verification
- Coupon management for Business Owners

User Interface Mocks

Screen 1

Home Screen - This screen will display available deals based on popularity and/or preferences to the user.

Screen 2
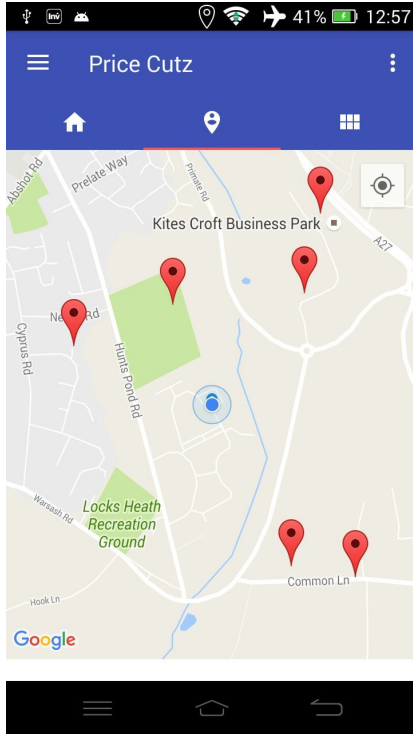
Near Me Screen - This screen shows coupons/deals based on their relative location to the user's location.

Screen 3
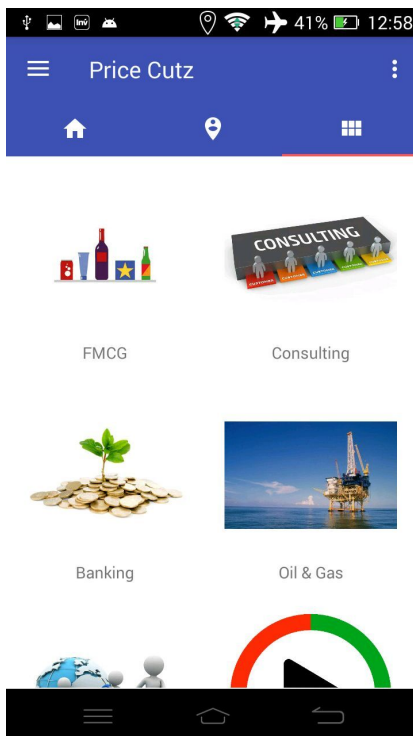
Categories Screen - This screen allows user to browse through the store by categories to view the discounts available for the these stores. This screen will also allow users perform a search on possible discounts.

Key Considerations

How will your app handle data persistence?

An SQLite library will be used to persist user preference data
However, standing data (coupons created) and transaction history (coupon usage) will be backed up on Google cloud (datastore) as well as on individual user devices.

Describe any corner cases in the UX.

There are 2 main corner cases;

- No location data: The application will be set to use GPS and well as Network location providers, however, if none of these are available, location feature will be disabled in the application without errors.
- No network access: If there are no network access during image requests, placeholder images will be used, however, Glide library also provide cache images when image is being reused

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

I will be using the following libraries
- Glide for image loading
- Crashlytics to monitor runtime errors on user devices
- Gson for easy serialization of objects to and from JSON
- Play Services for maps, location, ads and mobile vision (Barcode)
- Design Support Library
- Twitter SDK for Twitter login
- Facebook SDK for Facebook login

Describe how you will implement Google Play Services.

- Location & Maps - this will be used to visually represent discounts in the area
- Ads - the app will have a free and a paid version
- Mobile Vision - (some) discounts can be redeemed using QR codes generated for users

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

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:
- Configure required libraries
- Set up google cloud services for the app
- Obtain image resources

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity - This will have a DrawerLayout with three Fragments in a TabLayout.
    - HomeFragment
    - NearMeFragment
    - CategoriesFragment
- Other Activities will have regular layouts. They are as follows:
    - DiscountActivity - Displays list of discounts
    - CompanyActivity - Displays list of companies/stores
    - CategoryListActivity - Displays a grid of categories such as hotels, restaurants, retail stores, etc.
    - ProfileActivity - User area where deals can be created, used and managed
    - SettingsActivity - System setting area.

Task 3: Implement Data Persistence

A content provider will be implemented to store and retrieve local data within the app.

Task 4: Implement App Engine DataStore

Server side data storage of deals, companies, transactions etc. This will be set up as an App Engine Datastore.

Task 5: Implement App workflow

Here, specific application operations are implemented such as
- Loaders to display data on all views for the implemented activities and fragments
- Transaction (discount) creation and monitoring logic
- Location requests for nearby deals
- Barcode implementation
- AsyncTask to perform search in the CategoriesFragment by making web api requests.

Task 6: Accessibility and internationalization

- Implement accessibility for user with disabilities
- Support for multiple languages
- RTL support

Task 7: Implement Home Widget

Design a widget that shows nearby discounts from popular stores

Task 8: Design for Tablet

Update app design layouts for Tablets