GitHub Username: GerardoDRM

Beer Route

Description

Beer Route is the app for beer lovers and also for those newbies in the craft beer world. This app allows you to find the best route for drinking beers. Choose one beer in the catalogue, then check which are the best options to continue drinking or stop the party. Improve your experience when drinking craft beer.

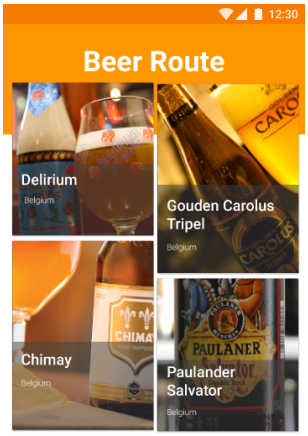
Intended User

This app is for people who know about craft beer and also for those newbies in the craft beer world.

Features

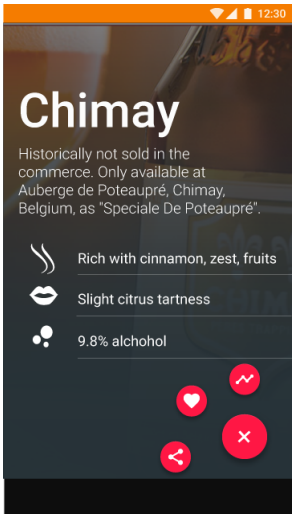
- Display craft beers
- Save favorite beers, in order to get them when offline.
- Display the best route for drinking beers.
- Display favorite beers on a widget.
- Share content about a beer.

User Interface Mocks



Screen 1

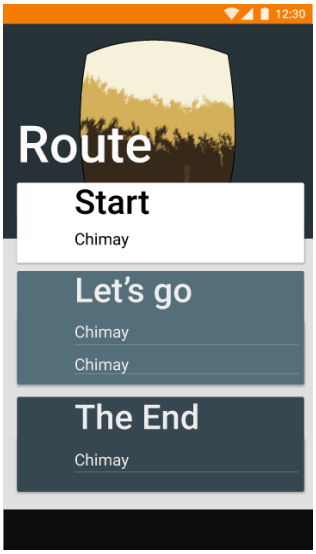
This is the main screen, the app display the beer catalogue. This screen structure is also used when user add a beer to favorites.



Screen 2

This screen show the beer description, it also has three options.

1. Share beer description with external app.
2. Save beer in favorites.
3. Continue to check the beer route.



Screen 3

This screen show the beer route. It has three cards.

- The first card shows the first beer that user decided at the beginning.
- The second card shows 3 possible beers if user wants to continue drinking. Those beers are selected to improve drinking experience.
- The third card shows 2 possible options, each option represent the last beer that user can drink after first card. This card improve beer taste.

Key Considerations

How will your app handle data persistence?

The app will use a Content Provider using schematics library and SQLite in backend.

Describe any corner cases in the UX.

- The user will return to catalogue screen with back button.
- The catalogue screen will have a menu on right corner with the options of sort by alcohol grade and show favorites.

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

- Picasso to handle the loading and caching of images.
- Retrofit to get data from server.
- Design Support library to improve material design on the app.
- Schematic to handle data persistence.
- Palette to extract colors from images.
- ButterKnife to write less code.

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

- Steup libraries
- Create project structure by folders (data, utils, models, api).

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for RouteActivity.
- Build Tablet UI for all activities.

Task 3: Implement Google Play Services

On this task I decided to implement Admob and Google Analytics

- Create Banner Admob on MainActivity.
- Track crashes on the app.
- Track shared content.

Task 4: Create Content Provider

This task will require Schematic Library.

- Create database model
- Use library in order to create content provider.
- The app needs to store favorite beers.
- Create a Loader for accessing to data.

Task 5: Handle Error Cases

- The app will prevent crashes in case of wrong data.
- Validate data from server.

Task 6: Create Widget

The app will have a widget which has the favorite user beers listed.

Task 8: ShareActionProvider (Optional)

The app will share a beer description with external application.

Task 9: Material Design (Optional)

The app will have shared element transitions and also it will have a parallax effect on RouteActivity.