**GitHub Username**: burcakDemircioglu

WannaBeer

# Description

This app will provide detailed information about the beers. It will also have daily beer suggestion notifications and a list of beers that user tried and liked or disliked before. User will be able to search the beer with the name of the beer and see a list of the beers which are in the same category.

# Intended User

University Students, Travelers, Young Adults

# Features

* Provides beer info
* Gives beer suggestions (notification)
* Save the rate info of the beers which are rated by the user

# User Interface Mocks

## ../Desktop/Nexus5Wireframing.pngScreen 1

This is the mock of the main activity. All of the beers are listed with the view of their logos which are clickable to go to their detail page. There is a menu button at the top left of the screen. There is explode animation when a beer card is clicked. Also there is a find button at the bottom of the screen which is linked to another activity to search for beers by their name.

**Alternative:** User can choose between the card view and list view. If she/he chooses list view that this screen will contain list view rather than card view. List view will be very similar with the Screen 5’s list view.



## Screen 2

This mockup is for the navigation menu to list the beers by their category or to go to the user’s liked beers’ list or to activate any other future feature of the app. When the user hits back button application turns to the screen where the menu is clicked open.

## 



## Screen 3

This mockup is for the beer’s detail page. On the page there is information about the bear and there are 2 buttons for like or dislike options. If like button is pushed then the beer is added to the liked list of the user. The buttons are animated with color filters. When like button is pushed a green filter appears and similarly a red filter appears when the dislike button is pushed. Also the image is collapsible in this page when the user scrolls to read the remaining information. Also there is a share button at the bottom of the screen which makes user able to share this beer with anybody by using any social platform.

## ../Desktop/Nexus5Wireframing2.png

## Screen 4

This mockup is for the beer search page. The items in the list are clickable which lead to the beer details page of the clicked item.

## ../Desktop/Nexus5Wireframing2.png

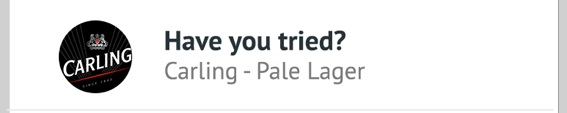
## Screen 5

This mockup is for the beer categories page. The items are clickable which leads to another list page of the beers which are in the same clicked category.

## ../Desktop/Nexus5Wireframing2.png

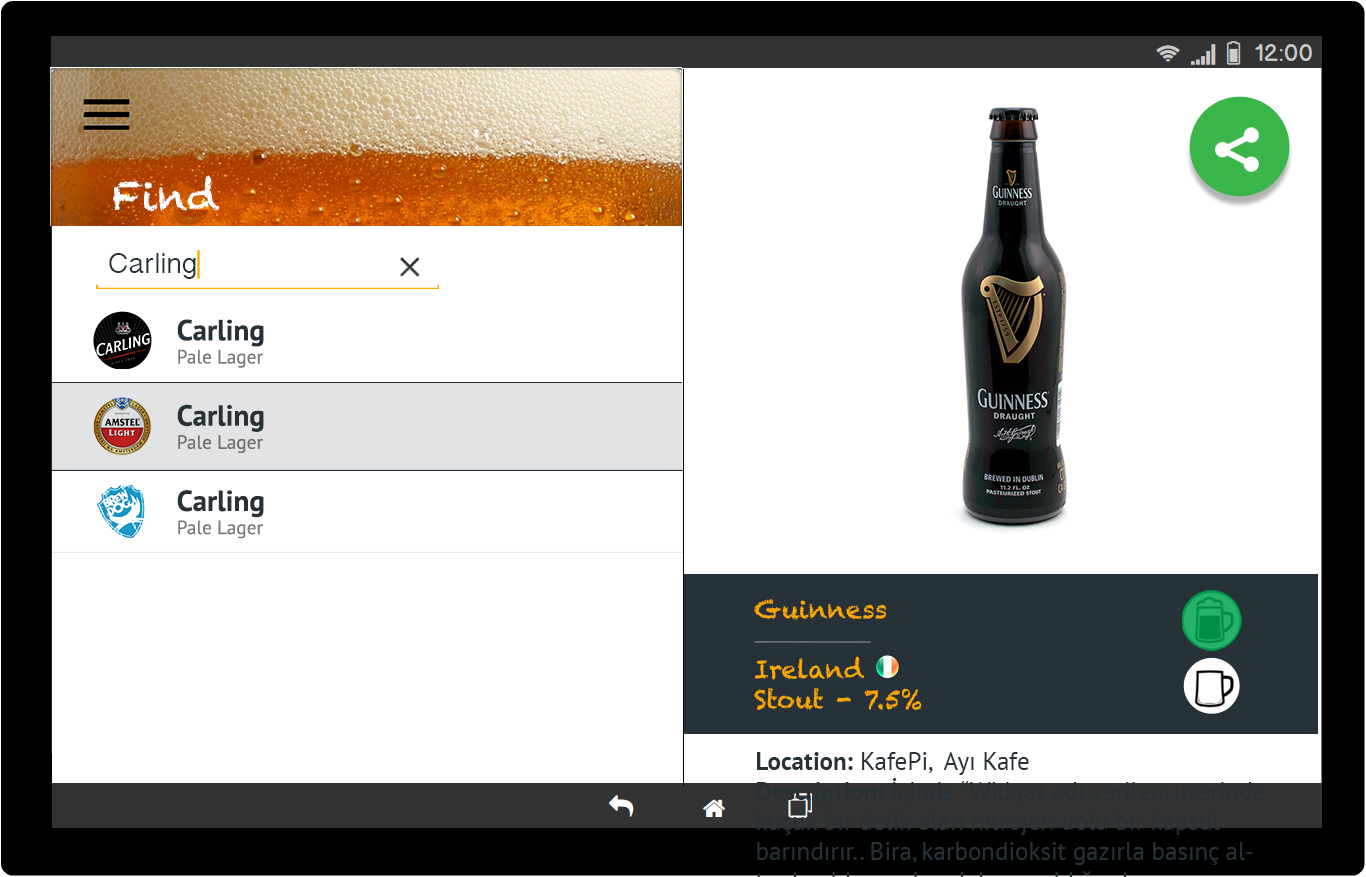
## Screen 6

This mockup is for the “my beers” page. It shows the beers that the user liked and disliked before. The items are clickable and linked to their details page. Liked and disliked lists are separate as panes and user can change the lists by clicking on the numbers of the related section



## Notification Screen

This is the notification’s mockup which gives beer suggestions to the user at approximately 5pm =)



# Tablet Screen

This mockup is for tablet screens which shows two pane view including a list view and details page. This two pane screen will be valid for all list pages for the tablet UI. At this screen the image in details pane will be collapsible by scroll.

# Key Considerations

### Does you app require any permissions?

Yes, It will require following permissions:

<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.BROADCAST\_STICKY" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.READ\_SYNC\_SETTINGS" />  
<uses-permission android:name="android.permission.WRITE\_SYNC\_SETTINGS" />  
<uses-permission android:name="android.permission.AUTHENTICATE\_ACCOUNTS" /><uses-permission android:name="android.permission.WAKE\_LOCK" />  
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

### How will your app handle data persistence?

I will build a Content Provider. The text data will be in dropbox as a json file which will be created by me. Also the images will be in the dropbox to, which will be reachable with the url that is provided in the json file. Also the like and dislike info of the user will be saved in shared preferences of an sqlite database will be created in order to save the info.

### Describe any corner cases in the UX.

There will be transition animations between the main activity and detail activity. Also there will be transition between the detail activities.

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

I will use Glide to handle the loading and caching of images. Because according to my research from Internet Picasso is slower on loading big images from Internet into ListView and since I have a lot of ListViews with images this downside is very important for me. Also among others Glide is easy to use, have small size of library and cache and not freezes the UI. Therefore, I am decided on to use Glide for the images.

# Next Steps: Required Tasks

## Task 1: Project Setup

* Configure libraries
* Create JSON file which includes beer info. FIREBASE e bak bunun yerine
* Find and save the bottle photos and logos of all of the listed beers.
* Find and save the country flag images.
* Fetch data from dropbox
* Create the sqlite database or the shared preferences

## Task 2: Implement UI for Each Activity and Fragment

* Build UI for MainActivity
* Build UI for Navigation Menu
* Build UI for Beer details
* Build UI for Beer Search Page
* Build UI for Beer Category List and Beer List within Categories
* Build UI for My Beers List

## Task 3: Connect Backend to Frontend

* Connect the data coming from the fetching to the UI
* Test user like list
* Handle error cases

## Task 4: Make the Notification

* Make the Notification UI
* Automatize the Notification for every day.

## Task 5: Build and Test

* Test all of the features of the app
* Handle error cases