# Dog Bork

## Description

Do you ever have problems of finding a dog off-leash park, or a dog-friendly hiking trail that you can spend time outdoor with your lovely furry friend? Or you come to a new place and you don’t know where you should take your dogs for a walk. With Dog Bork mobile app, you will find that is so easy to find an outdoor activity location for you and your dogs. And checking with the detail information of the searched location, you can decide when and where you walk your dog. Besides parks and hiking trails, Dog Bork is able to search any pet stores close to you location, and provides information about the stores. Once you have found your location, you can favorite the location as well.

Moreover, Dog Bork allows you to add and edit your dog’s profile with a nice user interface. If you have more than one dog, you just need to add profile for each of your dog under your account. Also, the profile will track the size and the weight of your dog. With these tracked histories, you can see how well and healthy your lovely pets are growing while you are taking them to exercise and meet friends in a nice outdoor park.

## Intended User

This is an app for dog owners.

## Features

The main features of Dog Bork are:

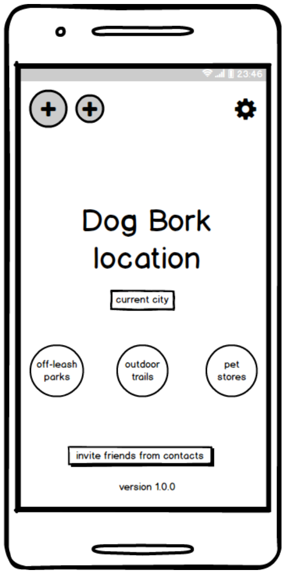
- Search the nearby outdoor activity locations or pet stores for your dogs

- Favorite your locations
- Manage your dog’s profile

## User Interface Mocks

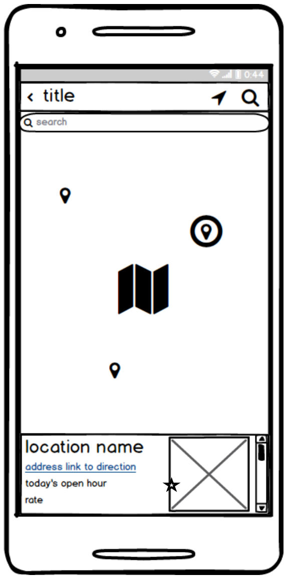
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

### Screen 1



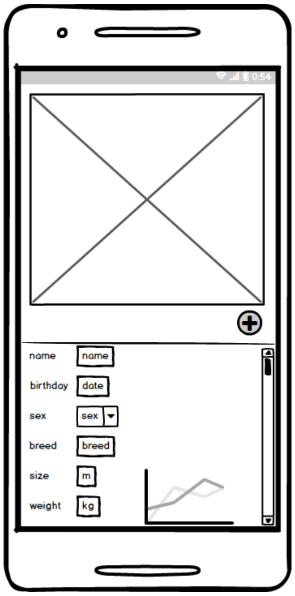
Home page is the first page to show user after the app is launched.

### Screen 2



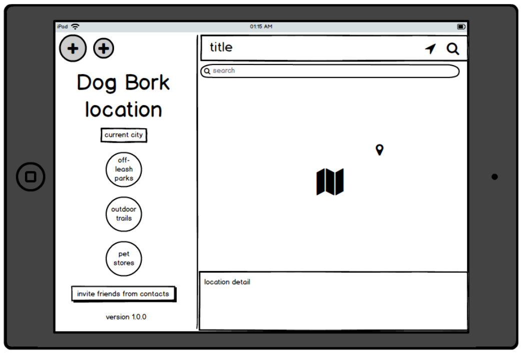
Map page is the page opened from clicking on the location category icon. The main part of the page is the map

## Screen 3



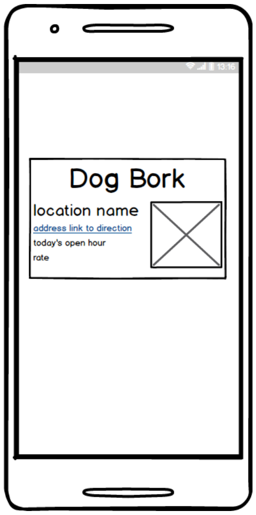
Dog profile page is the page to edit and read the dog profile and information.

## Screen 4



Tablet layout is to show the app appearance on a larger screen.

## Screen 5



Widget layout is to show the app widget on device home screen.

# Key Considerations

### How will your app handle data persistence?

I will Firebase Real-time Database to handle data persistence. More specifically, I will upload the user account information as well as his/her dog profile information to the Firebase Real-time Database. So once I launch the app, the app will load data from the database and will show on the UI.
Besides, I will use content provider to store any favorite locations that user favorited before. So the favorite location can be shown on the favorite screen with or without internet connection.

### Describe any edge or corner cases in the UX.

If there is no user account and dog profile information, user is still able to search the locations but will be prompted to add information before going to the map searching.

If there is no internet connection or the GPS signal is not strong enough, and it results in failing loading data, the user account page, profile page and the map page will be gray out and will show the error pop up dialog.

User can change the map search range, but there is a maximum radius range of 50 km which should be enough to cover a city area.

User does a location search and tap to open the location detail, if user goes from foreground to background and then comes back, the location screen should be persisted, i.e., the location should be still selected and the location detail should be shown. But user leaves the map search and goes to other screen, or does a new search, the data should be refreshed.

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

I will use Picasso library to handle the loading and caching of images because it is a powerful and common image library to be used.
I will also use Butterknife library to handle the UI element mapping in the code because it will keep the code clean and organized.
I will use Android material design library to design the app and improve the app UI.
I will use Google Play Services to support location search and Firebase Database to support data persistence.

### Describe how you will implement Google Play Services or other external services.

I will also use Firebase Real-time Database and Storage to store my users and their dogs' information once the user input the information and click on "Add" button. The database responses to requests fast and it is not difficult to implement in terms of coding.

Because this app mostly provides location search service, I will use Google Map, Location, Google Place APIs to support my location search features.

## Next Steps: Required Tasks

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

### Task 1: Project Setup

Create a project in Android Studio and open blank empty activity.

- Configure libraries – in app level build.gradle file add require libraries in the dependencies block.
- Configure manifest – update the manifest file for the require libraries including google-service.
- Configure Firebase – Create an account for Firebase service, and attach the project to the Firebase.
- Configure Google Map – Create an account for Google Play service and get the API key for using Google Map.
- Configure Admob – Create an account for Admob.

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

Use the mock-ups as reference to build the UI

- Build UI for MainActivity, i.e. create a layout as home page.
- Build UI for UserAccountActivity that allow user to input his/her information.
- Build UI for DogProfileActivity that allow user to input dog information.
- Build UI for LocationSearchActivity that contains the map fragment and location details.
- Build UI for MapFragment.
- Build UI for FavoriteLocationActivity that shows the favorite location list.
- Build UI for SettingAcitivity page.
- Build UI for Widget of the app, which will show the detail information of the location user previous clicked on the map.

### Task 3: Implement MainActivity

Basing on the requirements, build the MainActivity.java:

- Create layout.
- Implement the elements according to the layout file.

### Task 4: Use AsyncTaskLoader to load user data

Use AsyncTaskLoader to do the following steps:

- Connect to Firebase Database, and use loader to fetch the user and dog information if the data exists.

## Task 5: Use SyncAdapter to regularly sync location

Use SyncAdapter to do the following steps:

- Use Google Location service to get the device current location in terms of city at regular intervals.

## Task 6: Implement UserAccountActivity

Basing on the requirements, build the UserAccountActivity.java:

- If user account information exists, then populate the information.
- Create layout.
- Implement the elements according to the layout file.
- Connect to Firebase Database to add/update user information.

## Task 7: Implement DogProfileActivity

Basing on the requirements, build the DogProfileActivity.java:

- If dog information exists, then populate the information.
- Create layout.
- Implement the elements according to the layout file.
- Connect to Firebase Database to add/update dog information.

## Task 8: Implement LocationSearchActivity

Basing on the requirements, build the LocationSearchActivity.java:

- Create layout.
- Implement the elements according to the layout file.
- Handle the location detail section if the MapFragment return information.

## Task 9: Use IntentService to do Google Place search

Once user open the LocationSearchActivity, or user do a new search on the LocationSearchActivity, the app use IntentService to get data:

- Use the current location to locate the center of the map and then show the map with a set radius.
- Use Google Place API to do a search based on place type, current location and radius.

## Task 10: Implement MapFragment

Basing on the requirements, build the MapFragment.java:

- Create layout.
- Implement the elements according to the layout file.
- Filter the search result and fetch the desire information.
- Display the result on the map with markers.
- Enable onClick listener to handle user-clicking event.

## Task 11: Implement FavoriteLocationActivity

Basing on the requirements, build the FavoriteLocationActivity.java:

- Create layout.
- Implement the elements according to the layout file.
- Create content provider to store the favorite location list.
- Use a RecyclerView to display the favorite location list.

## Task 12: Implement SettingActivity

Basing on the requirements, build the SettingAcitivity.java:

- Create layout.
- Implement the elements according to the layout file.
- Apply the change on the app after resume the app.

## Task 13: Implement widget

Basing on the requirements, build the widget.java:

- Create layout.
- Implement the elements according to the layout file, so it shows the selected location information on device home screen.

## Task 14: Add Admob on the LocationSearchActivity

- Enable Admob service on the LocationSearchAcitivity.

## Task 15: Improve the app

- Create support for accessibility, eg., be able to adjust the size of the text.
- Use Material Design guideline to improve the app user experience.
- Create Gradle task to build and deploy the app.
- Implement notification when user arrives at one of his/her favorite location.
- Implement sharing functionality in “Invite friends from contacts” feature.

Add as many tasks as you need to complete your app.

---

**Submission Instructions**

- After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”