How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: MercyMutuku

Cats App

Description

Get to know much about the different cat breeds as well as categories we have in the world and browse through amazing photos of each breed / category. The App provides detailed information on each breed, e.g. origin, life span, temperament etc.

Intended User

From cat lovers to everyone interested in knowing more about cats. There are no limitations on who can use the app

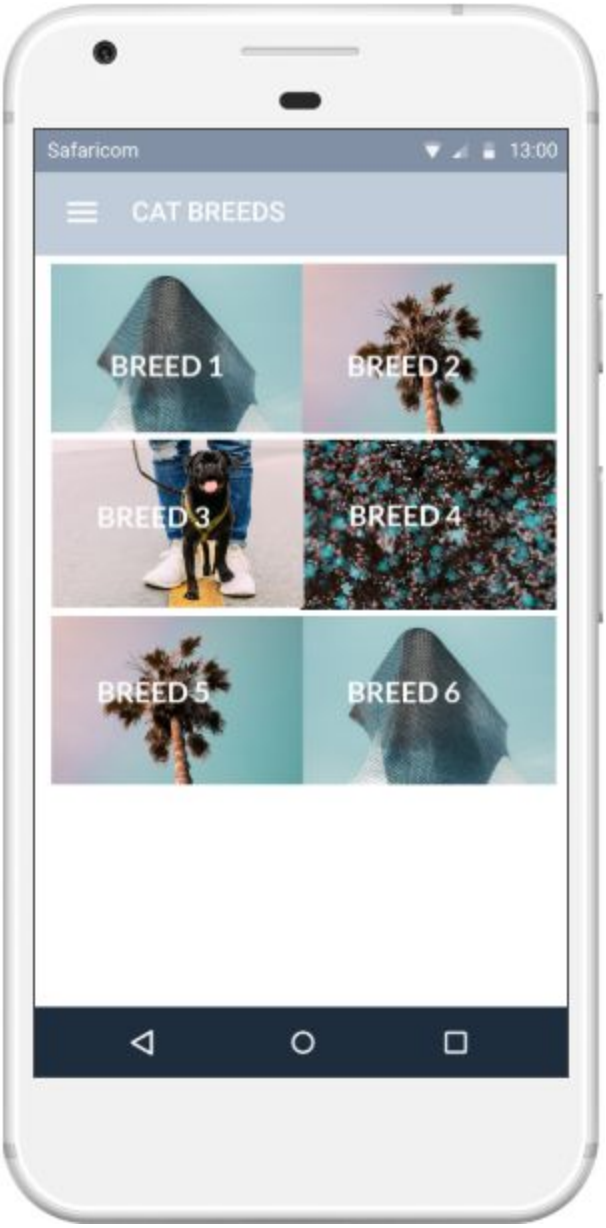
Features

- Gets all cat breeds and categories
- Provides detailed information on each cat breed
- Provides images for the different breeds as well as the categories

User Interface Mocks

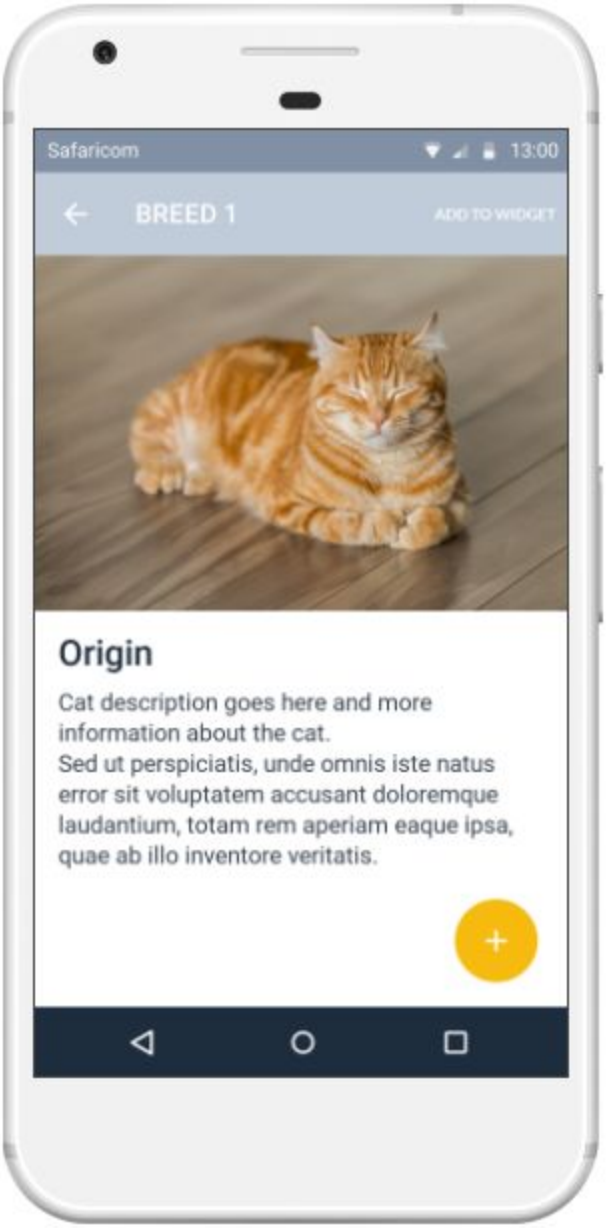
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

HOME SCREEN



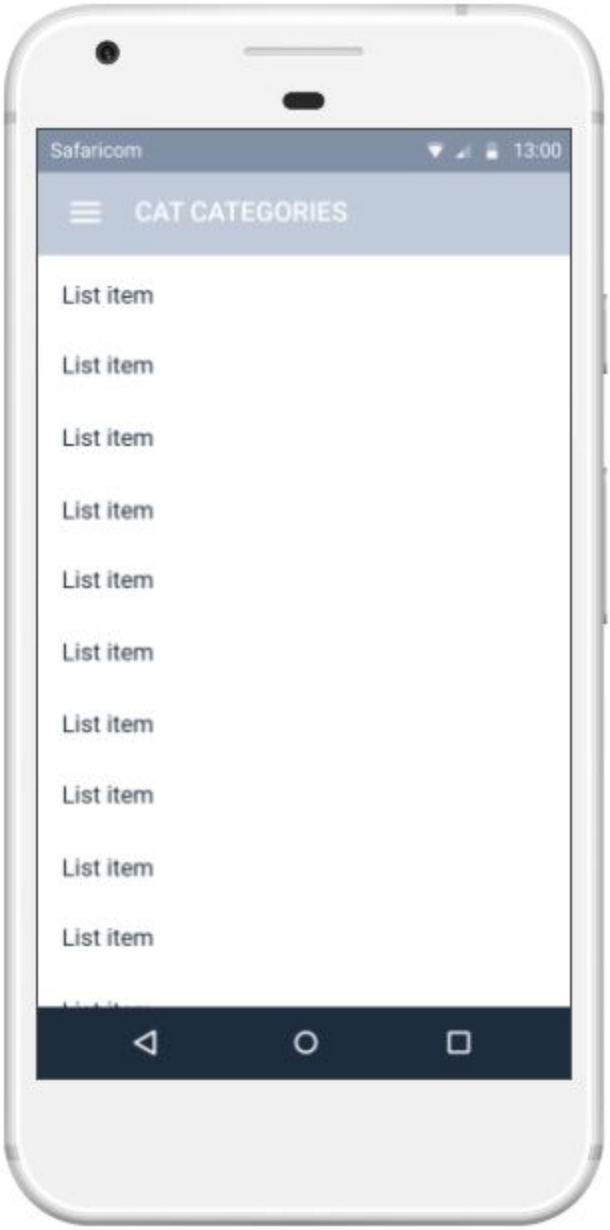
This is the default screen upon launching the app. This screen shows a list of the different breeds of cats. **Clicking on a breed launches the Breed Details Screen.** On the top left corner is an icon for the navigation drawer.

CAT BREED DETAILS SCREEN



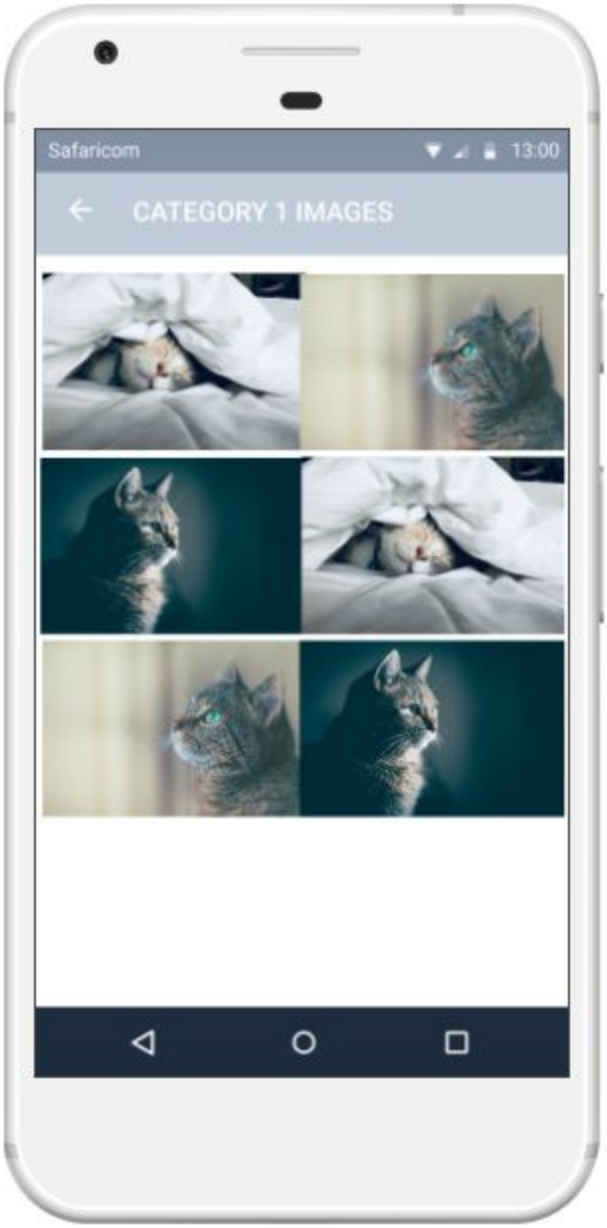
This screen provides detailed information on the selected breed. The floating action bar allows a user to mark a breed as favorite. In the App bar, a user can add a breed to the widget by clicking on "**ADD TO WIDGET**".

CAT CATEGORIES LIST SCREEN



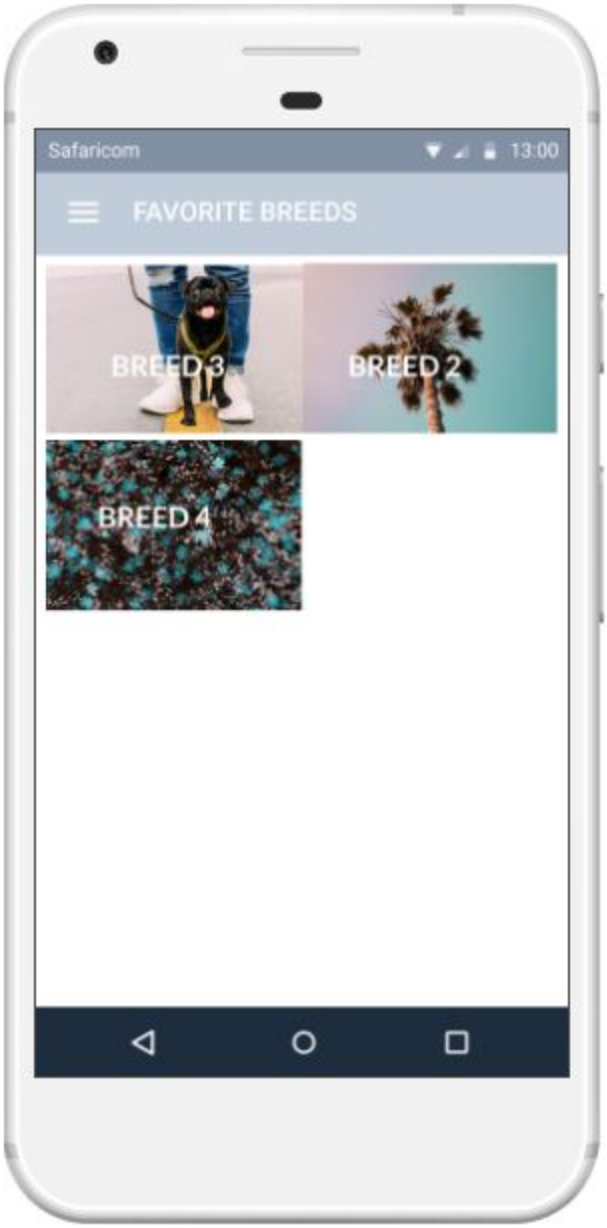
This screen provides a list of cat categories. Clicking on a category launches a screen with images for the selected category.

CATEGORY IMAGES SCREEN



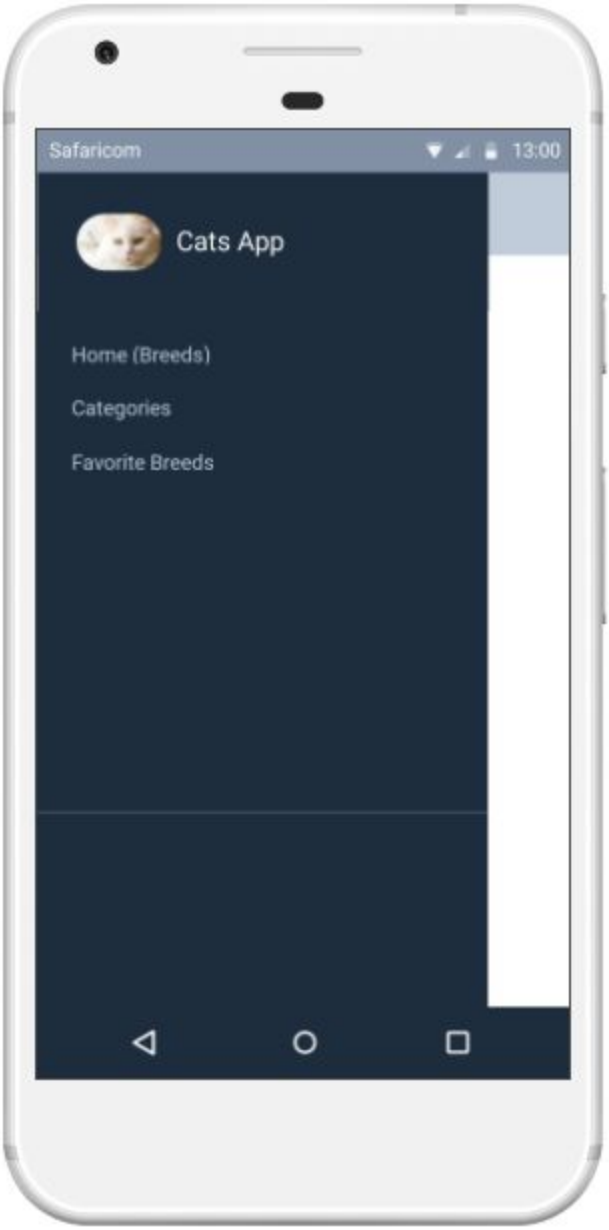
This screen displays images of the selected category. A user can click on an image to view the full image.

FAVORITE BREEDS SCREEN



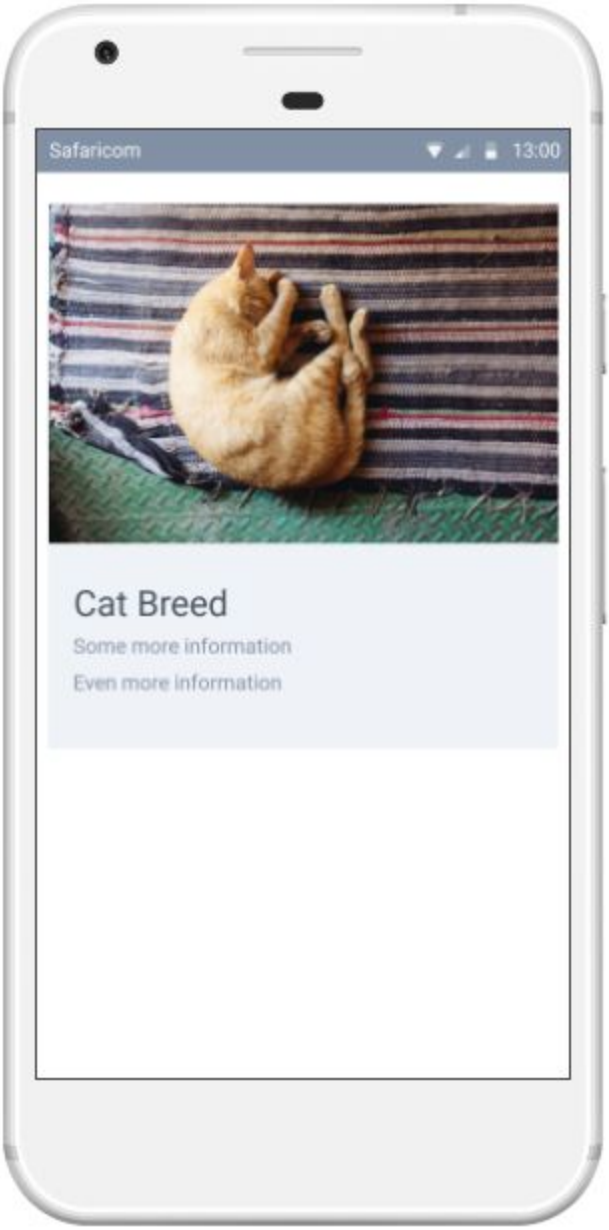
This screen lists the breeds that a user has marked a favorite. **Clicking on a breed launches the Breed Details Screen just like in the home screen.**

NAVIGATION DRAWER



Allows the user to switch between screens. It is launched by clicking on the Hamburger Icon.

WIDGET



This is the App widget. It shows the image and brief information about the breed that a user has added to the widget.

Key Considerations

How will your app handle data persistence?

Data will be acquired from THE CAT API. I will save both information about a breed that has been marked as favorite and a breed that has been added to widget to Room DB using AsyncTask.

Describe any edge or corner cases in the UX.

Activities providing details will be correctly mapped to their parent activities, meaning, hitting the back button will lead the user to the appropriate activity. When there's no internet connection, a friendly message will be displayed to inform the user about the network state. Breeds marked as favorite will be available offline but the images will be replaced with placeholder images.

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

Picasso (v 2.5.2) - To handle loading of images.
Jake Wharton's Butterknife (v 8.8.1) - to enhance code readability and easier views and resources lookups
Android design library (v 28.0.0) - for material design components and other design features
Android support library (v 28.8.0) - for backward compatibility
Room (v 1.1.1) - for storing data from api to db for offline access
Google mobile ads (v 17.1.2) - for monetization of the app
Google analytics (v 4) - for usage analytics
Timber (v 4.7.1) - for logging
Retrofit (v 2.3.0) - for api work
Gson (v 2.3.0 - for creating TypeConverters

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

Google mobile ads - for displaying ads on the app
Google analytics - for usage analytics

Other Considerations

App will be written solely in the Java Programming Language
App will utilize stable release versions of all libraries, Gradle, and Android Studio
App will keep all strings in a strings.xml file and will enable RTL layout switching on all layouts
All images will have content descriptions
The UI mocks will definitely need more polishing

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

- Set up project, set name, build tools version, min and target sdk versions
- Implement all external libraries
- Setup permissions
- Setup vcs for project

Task 2: Implement UI for Each Activity, Fragment and Widget

- Build UI for MainActivity
- Build UI for all fragments
- Build UI for the other activities
- Build UI for alternative screen sizes
- Build UI for the widget

Task 3: Database and API Setup

- Create Room db and classes
- Create Retrofit api methods

Task 4: Create RecyclerView Adapters

- Create adapters for recycler views, taking data from Room
- Set Adapters for recycler views

Task 5: Implement Ads

- Implement Ads in the appropriate flavor

Task 6: Polish UI

- Create layouts for different screen sizes
- Implement material design guidelines

Task 7: Implement Analytics

- Implement google analytics

Task 8: Testing

- Test connection to DB
- Test UI
- Test app on devices with different screen sizes

Task 9: Clean code and submit

- Clean the code
- Check for and delete unused resources
- Submit for review

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 "**Cap stone_Stage1.pdf**"