ARUBA FLORA & FAUNA

Description

Intended Users

Features

User Interface Mocks

    Screen 1

    Screen 2

    Screen 3

Key Considerations

        How will your app handle data persistence?

        Describe any edge or corner cases in the UX.

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

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

Next Steps: Required Tasks

    Task 1: Project Setup

    Task 2: Rest API

    Task 3: Data persistence app

    Task 4: Handle authentication

    Task 4: Post data to api

GitHub Username: abdullah10182

Aruba Flora & Fauna

Description

The app is intended as an educational tool to help the user learn more about the diverse nature of the island of Aruba. Aruba is a pretty small island and it is therefore difficult to find information about the flora and fauna of the island on the internet.

The app will consist of two main categories for now, namely plants (flora) and animals (fauna). Under each category the users can select species of plants or animals to find out more information.

The idea is that in the future along with the app there will be also a website that displays all this information. The website will initially be build but only as a rest server to feed and collect data to and from the app.

Intended Users
- Nature enthusiasts
- Scientists
- Students
- Tourists
- Kids

Features
- View info about a species
- View the local name of the species
- See if the species is endemic or invasive to the island
- User can create account and send corrections, questions or feedback on a species or request to add a species
- App will not need RTL support because it will only be English and is mostly intended for people from Aruba
- All strings will be stored in xml files in their corresponding folders

Nice to have
- Image classification of plants
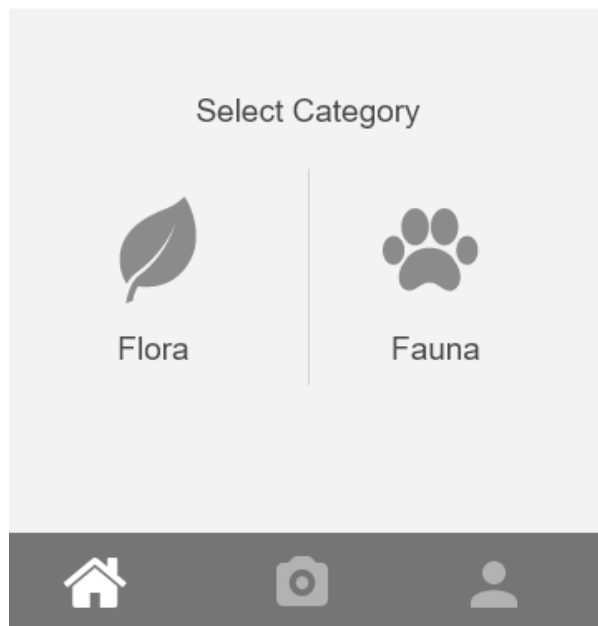- Geo tag where the image of the plant was taken

Technologies
- Programming language: Native Android app written in Java
- IDE: Android Studio
- Rest API: Drupal 8.x.x as headless CMS
- Gradle 6.x.x
- Retrofit 2.6.x
- Glide 4.10.x
- Butterknife 10.2.x
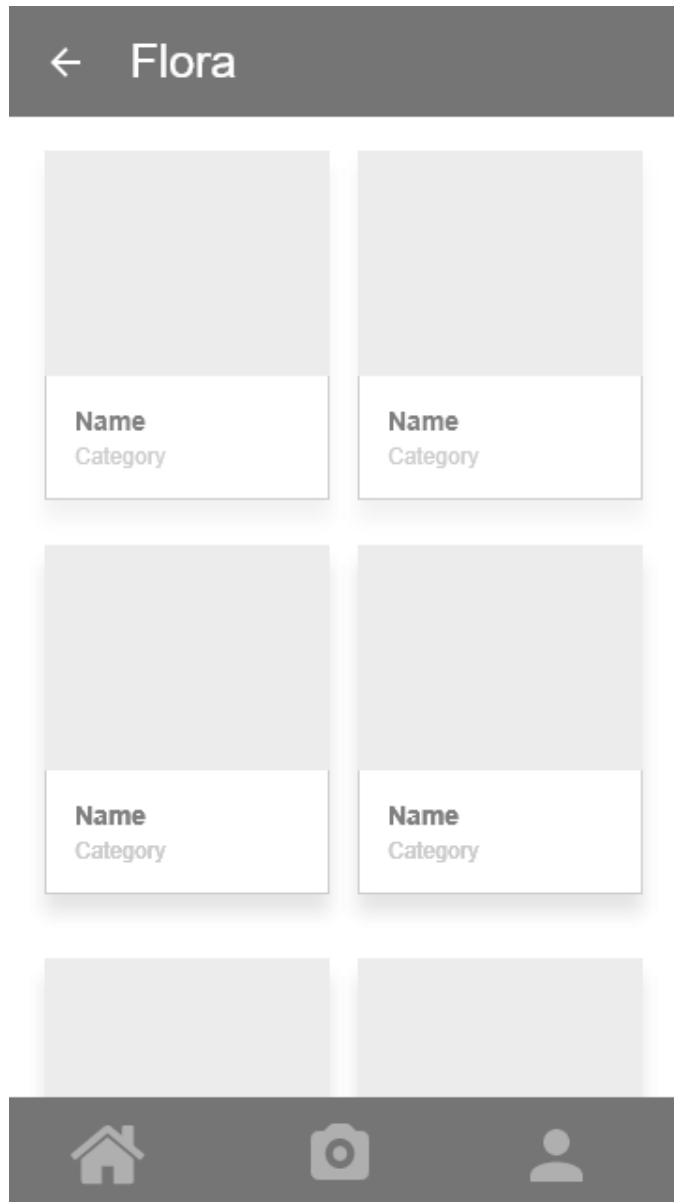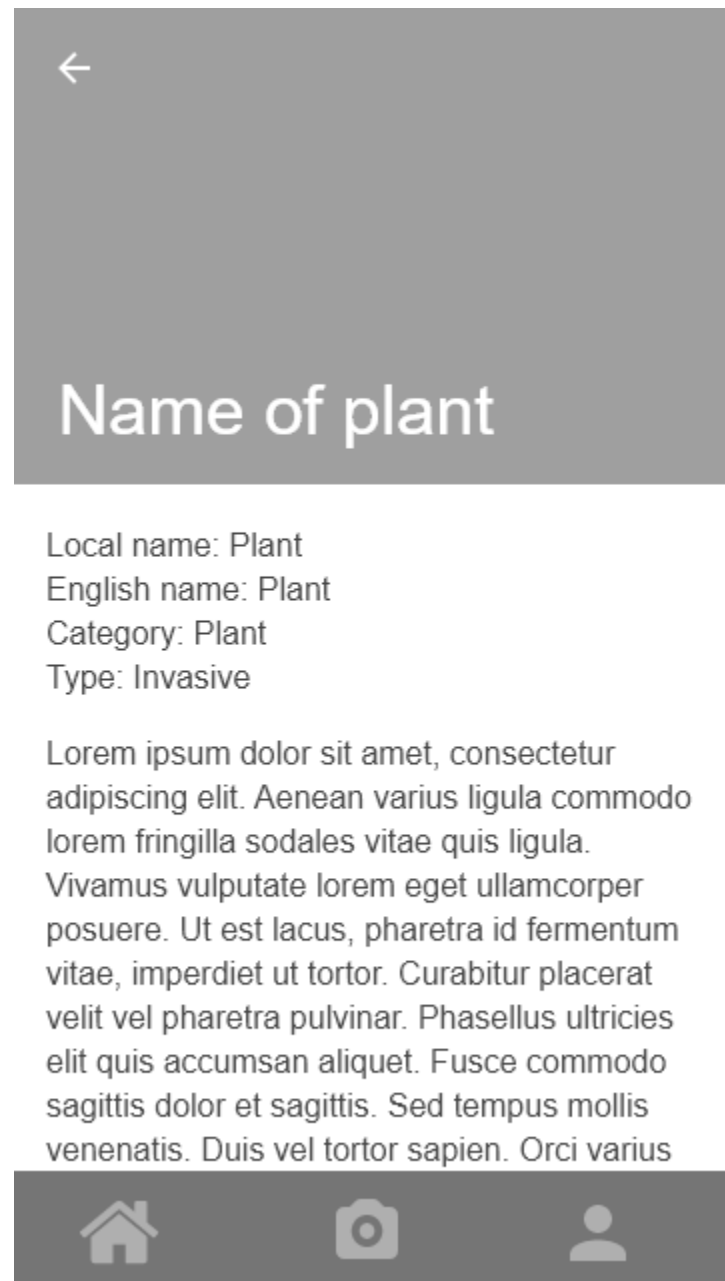- Material design 28.x.x

User Interface Mocks

Screen 1

Logo

Aruba Flora & Fauna

Select Category

Flora   Fauna

Screen 2

← Flora

Name
Category

Name
Category

Name
Category

Name
Category

Screen 3

Local name: Plant
English name: Plant
Category: Plant
Type: Invasive

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean varius ligula commodo lorem fringilla sodales vitae quis ligula. Vivamus vulputate lorem eget ullamcorper posuere. Ut est lacus, pharetra id fermentum vitae, imperdiet ut tortor. Curabitur placerat velit vel pharetra pulvinar. Phasellus ultricies elit quis accumsan aliquet. Fusce commodo sagittis dolor et sagittis. Sed tempus mollis venenatis. Duis vel tortor sapien. Orci varius

Screen 4 (widget)

Key Considerations

How will your app handle data persistence?
I will use Drupal as a headless CMS as a rest API. And for the app I will use room to persist visited data.

Describe any edge or corner cases in the UX.
The app will only work in portrait mode.

Describe any libraries you'll be using and share your reasoning for including them.
Retrofit to get and post data. Glide to handle images and the material design library for the design. These are mainly what I used in the course, so I already have experience with these.

Describe how you will implement Google Play Services or other external services.
Will use try to use google maps to geo tag where image was taken.

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
- Start blank project in AS
- Import all needed libraries
- Create all activities and fragments

Task 2: Rest API
- Set up drupal as headless cms
- Create all fields
- Set up rest server

Task 3: Data persistence app
- Set up room
- Connect to api
- Display data

Task 4: Handle authentication

- Create user registration form and make it work through either a webview or through an api with JWT
- Create user sign up form and make it work with api with JWT
- All calls must me authorized with JWT
- Try to implement SSO with Google and Facebook login

Task 4: Post data to api

- Create user upload suggestions or feedback form
- Create camera activity
- Post picture to api