
Paul F.J. Aranas

Capstone Stage 1

GitHub Username: Paul-FJ-Aranas

Oregon Garden Adventure

Description

The Oregon Garden is a lush 80 acre botanical garden located in Silverton, Oregon. This app allows you to easily find your way through the garden and to locate key destinations using an interactive GPS map. It provides details for every sub-garden. You can also get a space view of the garden and access necessary visitor information.

Intended User

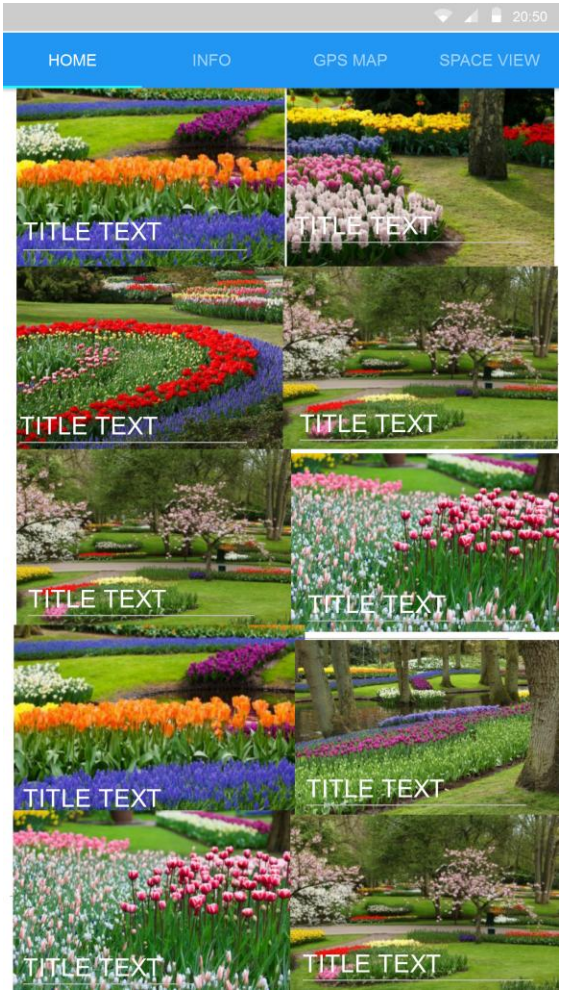
The intended user is a visitor to The Oregon Garden in Silverton, Oregon.

Features

- GPS map
- Sub-garden details
- Space View
- Visitor information
- Master-detail layout for tablets
- Material Design Theme

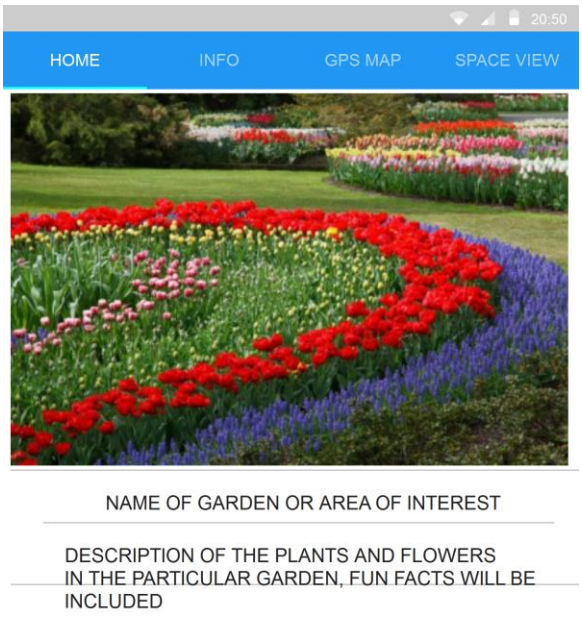
User Interface Mocks

Screen 1



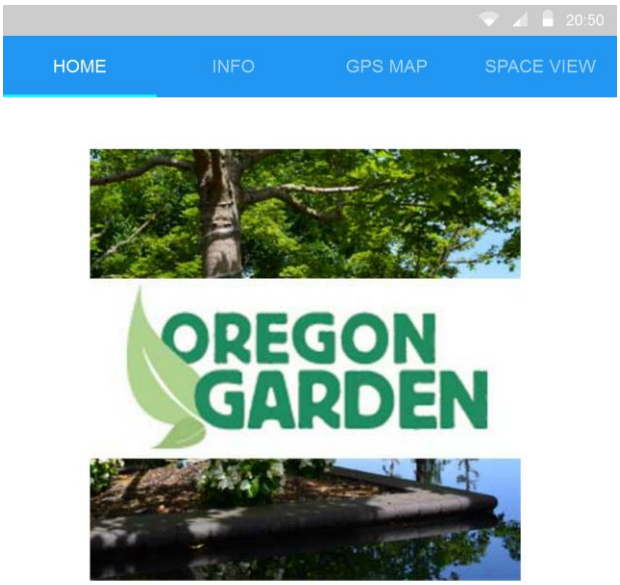
The launch screen is a Grid Layout + RecyclerView with images of each sub-garden. The title of each garden is over the bottom of the image using a scrim for clarity. A click on a garden image will take the user to a details screen. On tablets, a master-detail flow will be employed. Material design specs (margins, spacing, text size etc) will be employed.. The UI will be coded in a fragment.

Screen 2 (Garden Details Activity)



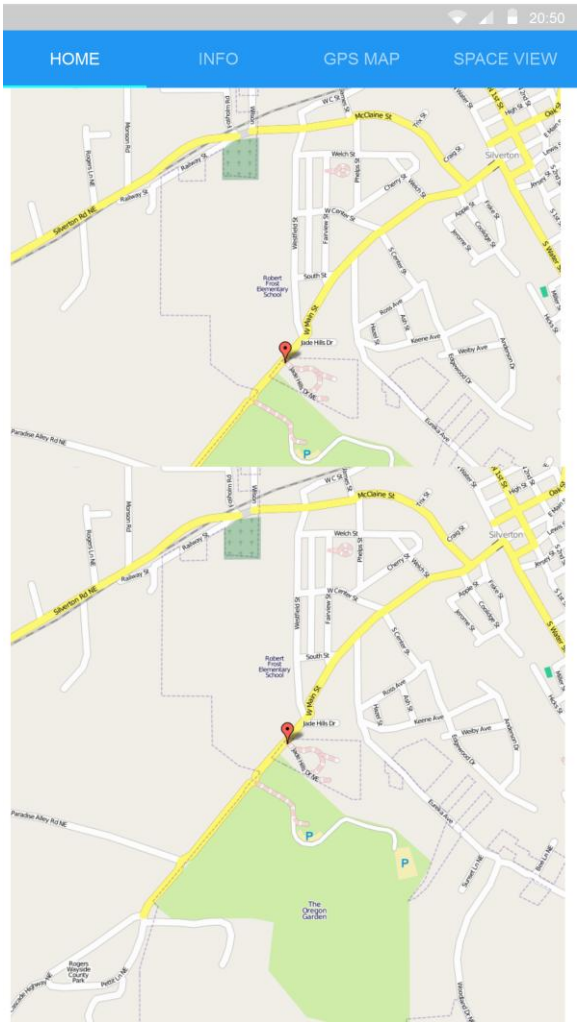
The garden details screen will give specific information on each sub-garden. There will be a shared element transition between the grid image in the launch screen and the main image in the details activity. The details information will come from existing literature as well as original in-person research. The UI will be coded in a fragment.

Screen 3



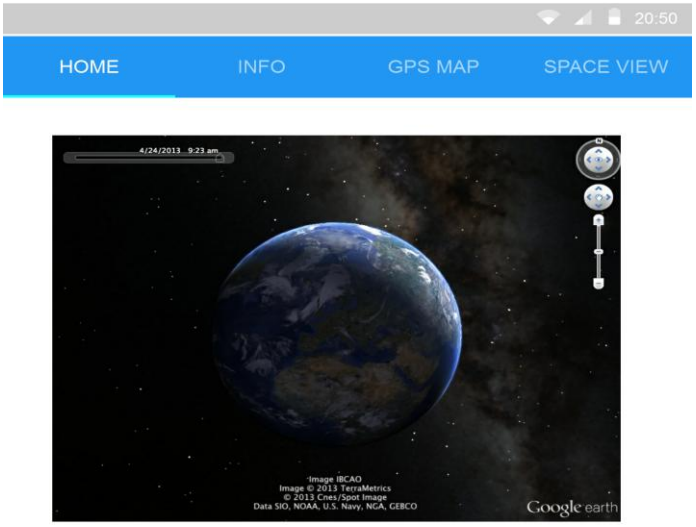
The Info Activity will contain a WebView with the official Oregon Garden Website. This will be useful for users to easily access important information without having to open a browser. This will provide information such as purchasing passes, directions, hours, history, and contact details.

Screen 4



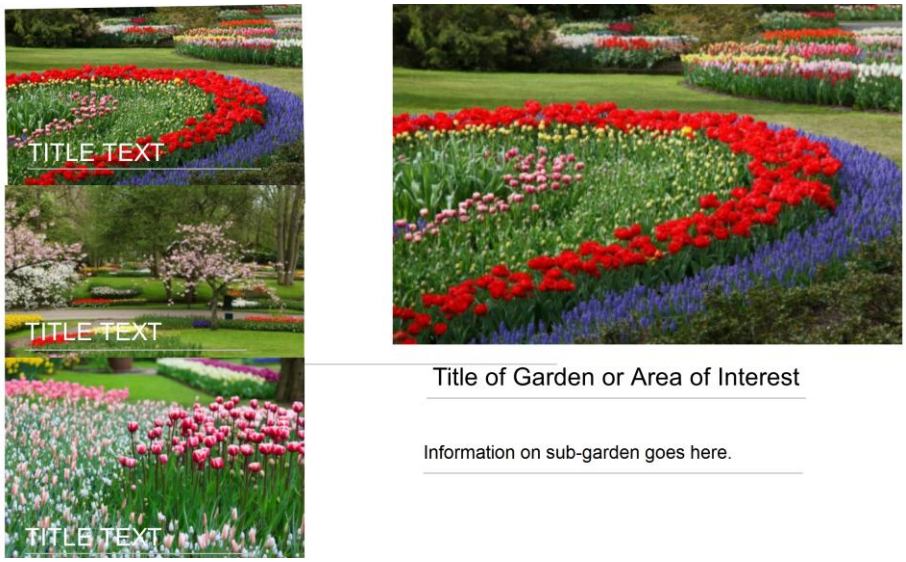
The GPS map activity is the center piece of the app. It will provide an accurate map of the garden using the Google Maps API and the current location of the user by implementing the Google Location API. Each sub-garden will be marked on the map using a clickable marker-- a circle will be drawn around the sub-garden using the radius. Upon clicking on the marker, some important information will be shown as well as the option to go to the garden details screen. The GPS coordinates for each garden will be recorded on-site to ensure precision. The location of the user will be marked on the map. The user will be able to zoom and adjust the map to view it with ease.

Screen 5



The Space View Activity will contain a VideoView of an mpeg-4 clip of a Google Earth Pro video which slowly zooms from space to the Silverton Garden. This feature will allow travelers to have a perspective of where Silverton, Oregon is located as well as view a cool and entertaining video clip. The video will be captured in avi and then converted to mpeg-4 to be used in the app.

Master-Detail Layout for Tablets (Screen 1 and 2)



The app will be optimized for tablets using a master-detail flow. The Grid Layout and the garden details fragment will be side by side when viewed in a tablet. The Info, GPS Map, and Space View activities will have their own activities on both mobile and tablet.

Key Considerations

How the app will handle data persistence?

I will implement a backend using Google App Engine and Cloud Endpoints. The data will be drawn from a java library. My app will use an SQLite database to store the data locally, providing for offline functionality. The database will be accessed via a Content Provider. Google Cloud Messaging will be used to contact devices when the data on the server has changed and needs to be synchronized with clients.

Corner cases in the UX

The app will have a tab bar for the following: the main garden Grid Layout, the GPS map, the info, and the space view activity. The up button will return a user from a garden details page to the main Grid Layout. When inside any of the other activities, the garden Grid Layout can be accessed by clicking on its corresponding tab. The space view video will be contained in a VideoView and embedded locally. The goal is to avoid corner cases.

Libraries used and reasons for including them

I will use Picasso to load garden photos. Picasso will take care of caching and threading during image loading. It will also provide methods for placeholders and resizing of the photos. I will also use Espresso to test the UI.

Next Steps: Required Tasks

Task 1: Project Setup

- Do Garden Research -- 23+ sub-gardens
- Record GPS Coordinates for each sub-garden
- Set up Google Maps API Account and Google Cloud Messaging (GCM)
- Get API Keys

Task 2: Create View from Space Video

- Capture view from space video
- Convert form avi to mpeg-4 format

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Details Activity
- Build UI for GPS MAP
- Build UI for Info Activity
- Build UI for Space View Activity

Task 4: Implement Map Functionality

- Customize map: mark each sub-garden based on recorded GPS coordinates
- Use the location API to mark current position of user within the garden
- Add relevant information to details box when marker is pressed
- Allow for offline access

Task 5: Implement Backend

- Create Java Library and populate it with garden data
- Implement Google App Engine -- Cloud Endpoints
- Use GCM for backend updates

Task 6: Implement SQLite Database and Content Provider

- Create database classes (helper, contract etc.)
- Create Content Provider

Task 7: Run Tests

- Run unit tests
- Run instrumentation tests (Espresso)
