How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: mtpayne

Henderson Park Finder

Description

People of Henderson NV. We have many parks offering many activities. Have you ever wished you could find the exact park you were looking for? Want to play basketball and bbq afterwards? Which ones have a dog area?

Now there's an app for that! The Henderson Park Finder will allow you to search for the nearest park with the amenities you are looking for. Get details about the park and even view them in a map! Save your search preference so you can find the nearest park from wherever you are in Henderson.

Intended User

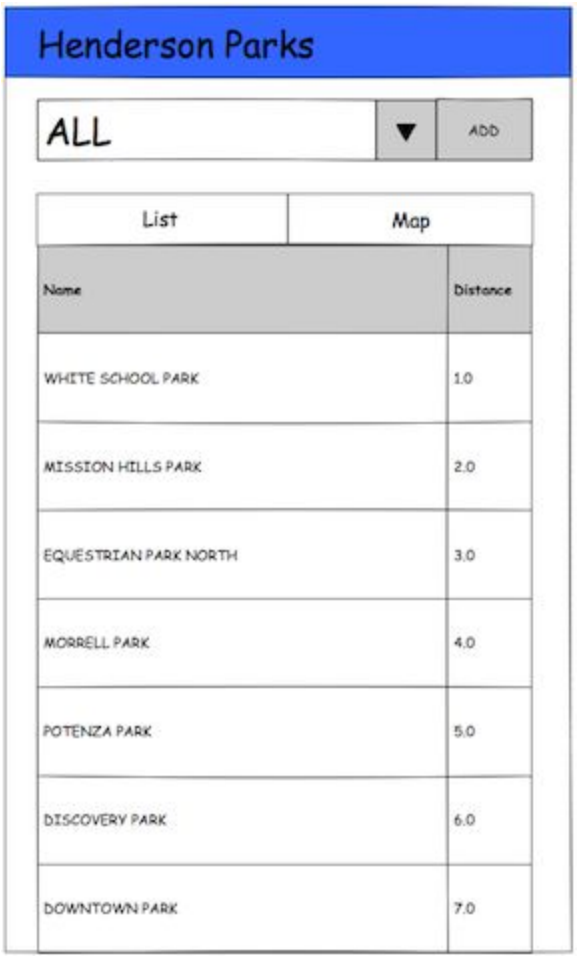
Henderson Park Finder is intended for park lovers in Henderson NV.

Features

- List parks.
- Filter for park amenities.
- Save park amenity filter.
- Display park details.
- Display parks in map.
- Display nearest parks.

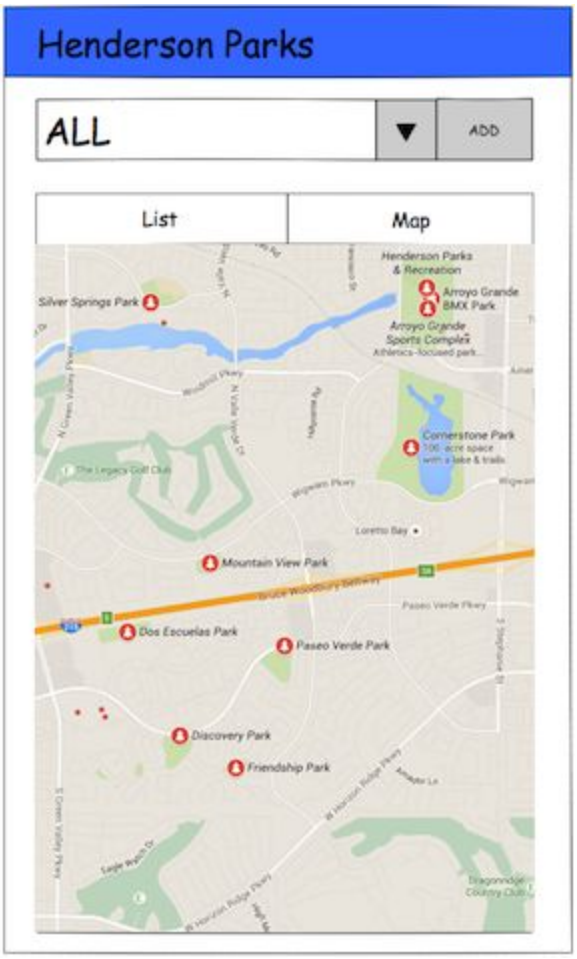
User Interface Mocks

Main Screen - List



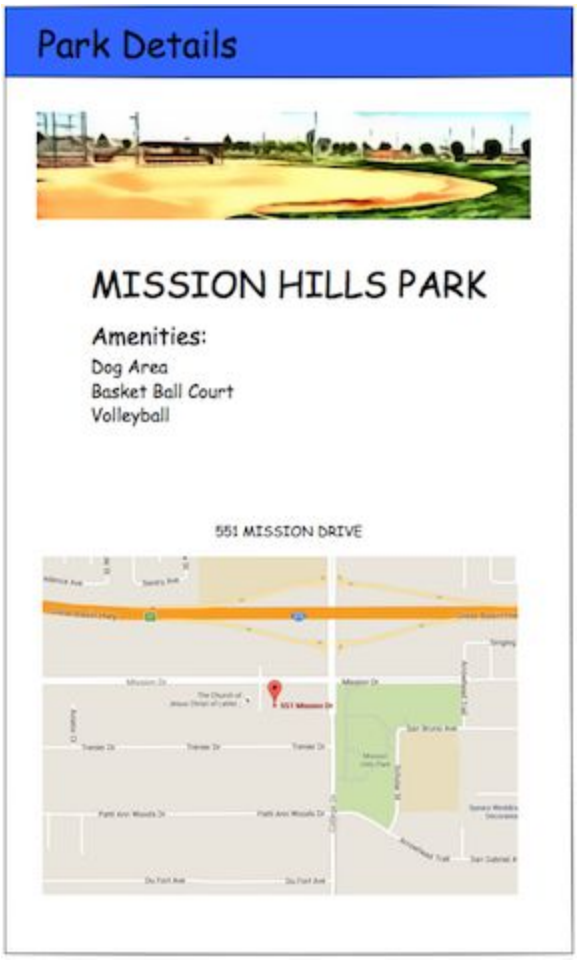
This is the main screen. By default it will show the list of parks by distance. You can filter the types of parks you want to see by selecting from the dropdown menu. Add new filters by tapping the ‘ADD’ button. You can view the parks in a map by tapping ‘Map’ tab. You can also view the park details by selecting a park from the list.

Main Screen - Map



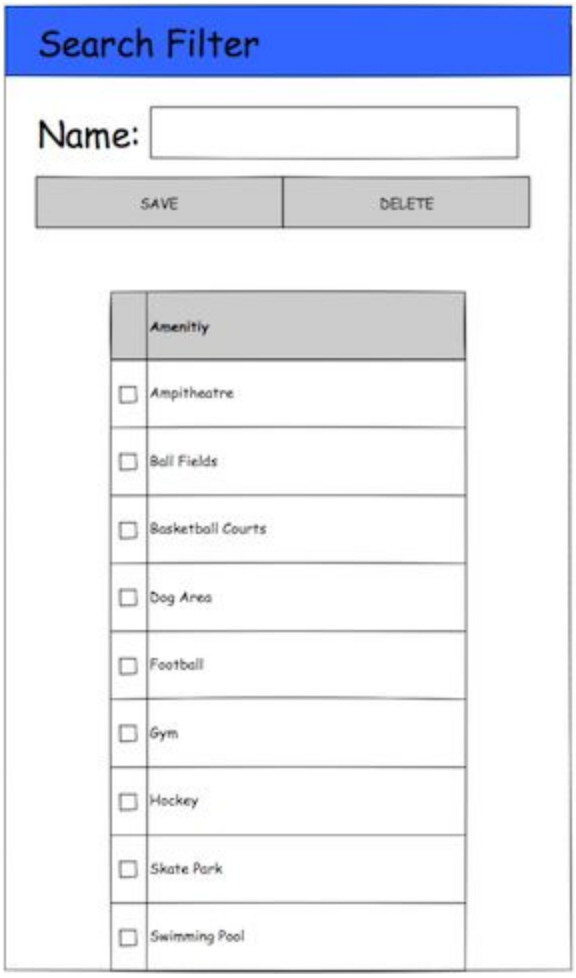
This is the main screen after selecting the ‘Map’ tab.

Detail Screen



This is the Detail screen after selecting a park from the ‘List’ tab in the Main screen. It contains an image, details and map address/location of the park.

Search Filter Screen



This is the Search Filter screen after tapping on the ‘ADD’ button from the Main screen. You can create/edit/delete a filter using the name field and save/delete buttons. You can check as many amenities you want for your filter and it will be saved along with the filter name.

Main Screen - Tablet

This is the main screen in tablet mode. It separates out the ‘List’ and ‘Map’ tabs from the phone mode and displays them at once. Otherwise the functionality remains the same.

Key Considerations

How will your app handle data persistence?

Persistence will be handled by a Content Provider storing data in sqlite database.

Describe any corner cases in the UX.

When the app is first launched there is no park data. It needs to retrieve the data and will require an internet connection. In the event there is no internet connection, a message will be displayed.

The app needs GPS access to make distance calculations. If GPS is not enabled a message will be displayed.

The app also needs internet connection to load park images and to use Google Maps. A message will be displayed if no connection is available. Placeholder images will be used when data is available in the database but no internet connection is available to retrieve park images.

A filter must be created before it can be used. By default a ‘ALL’ placeholder that returns all parks will be created and used. Since, filters can be deleted we will check to make sure at least 1 filter(extra) is available before deleting a filter. If there isn’t at least an extra filter to display data a message will be shown(informing user that they cannot delete the filter) if a user tries to delete the last filter.

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

Picasso for handling image loading.
Butter Knife for handing view(field and method) binding.
Instagram ig-json-parser for handling JSON data.
Google Play services for location and maps.
Android Support Library for backwards compatibility.
Android Design Support LIbrary for Material Design backwards compatibility.

Required Tasks

Task 1: Project Setup

Make sure Android Studio is up to date. Since we are incorporating Google Maps functionality. Follow the latest instructions on obtaining a Google Maps API key.

https://developers.google.com/maps/documentation/android-api/start#step_4_get_a_google_maps_api_key

If you followed the instructions above you now have a starter ‘Google Maps’ project and API key. Do the following:

- Remove unnecessary boilerplate code.
- Add libraries we will be using in project.
- Add permissions necessary for all functionality.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Activity
- Build UI for List Fragment
- Build UI for Map Fragment
- Build UI for Detail Activity
- Build UI for Detail Fragment
- Build UI for Filter Activity
- Build UI for Filter Fragment

Task 3: Implement Data Handling

- Build data schema and contract for park and filter.
- Build sql helper for park and filter data.
- Build content provider for park and filter data.
- Build service to load data in background. Data will come from Henderson Open Data site.

Task 4: Implement Each Activity and Fragment

- Build Main Activity incorporating List Fragment which will display park data retrieved from database and Map Fragment which will use Google Maps to display park data plots on a map. Add functionality to filter data and go to Filter page. Add functionality to view park Details page.
- Build Detail Activity incorporating Detail Fragment which will display park details retrieved from the database. Add Map of park location.
- Build Filter Activity incorporating Filter Fragment which will display park amenities that can be used to create, update and delete filters to database.

Task 4: Implement Tablet UI and Widget

- Create Tablet UI by separating out the List and Map to display at once.
- Create Widget that lists filters so user can quickly launch app with appropriate data.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"