
GitHub Username: https://github.com/ssowens/Capstone-Project

Home For Now

Description

This app will allow you to select, save and share vacation rentals. The vacation rentals are separated into categories: most popular and top rated. For each vacation rental you will be able to make it your favorite. Once you make a vacation rental your favorite, you will have the option to view all of your favorite vacation rentals from the menu option.

Intended User

Travelers are the intended users for this app.

Features

MAIN FEATURES:

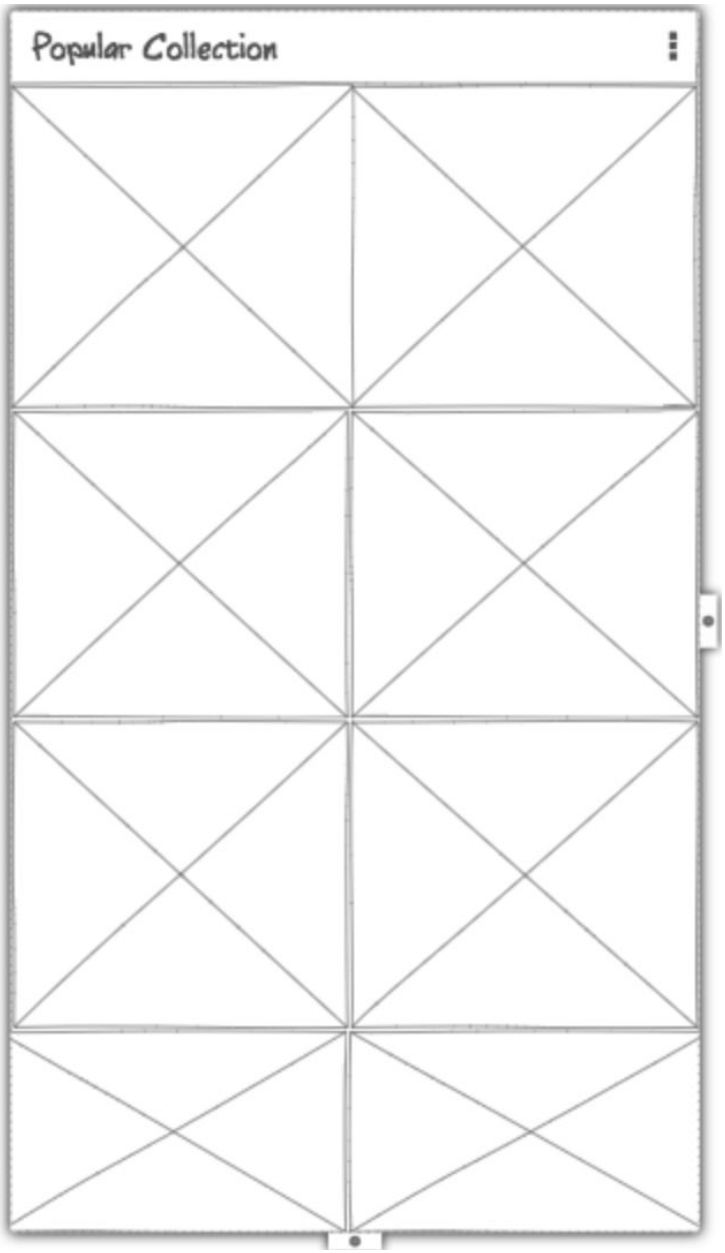
- Sorts vacation rentals by categories: Popular Vacation Rentals, Top Rated Vacation Rentals and Favorite Vacation Rentals
- Allows the user the option to select and save favorite vacation rental
- User may access favorite vacation rentals at any time, even when WiFi is not available.
- If the user chooses to purchase the app, they will not see ads

PERKS: FIND THE PERFECT SPOT FOR YOU AND YOUR FAMILY

- Peruse popular and top-rated vacation rentals in one app
- View houses, condos and rooms that meet your needs
- Save and share your vacation spots with family and friends
- See photos of your vacation spots at a glance

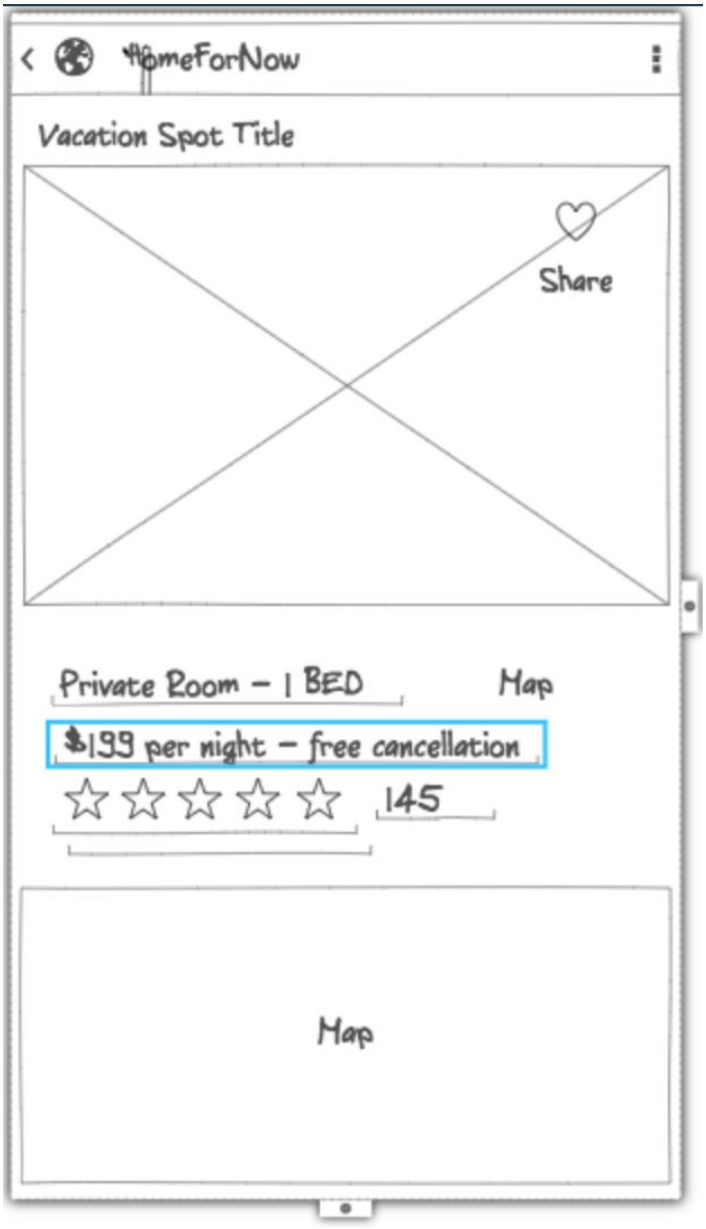
User Interface Mocks

Screen 1 - Main Screen



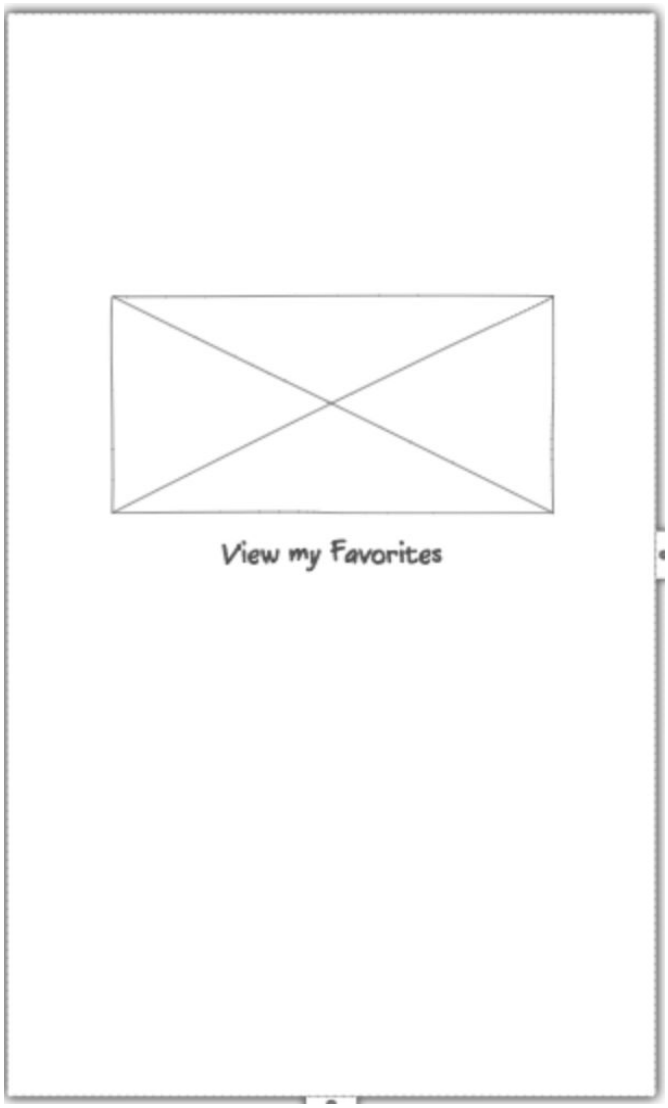
This screen will initially present the most popular collection of vacation rentals. There will be a menu option that will allow the user to select between most popular vacation spots and most highly rated. After the user has selected favorites, they will have a third option from the menu to select their favorites. If the user, clicks on a vacation rental, they will be taken to a detail screen about the vacation rental.

Screen 2 - Detail Screen



This screen will show the detail of a selected vacation spot.The detail will include the number of bedrooms, the number of bathrooms, the ratings, the number of people who have rated the rental location, an option to make this location a favorite, the option to share with family and friends will be an option and a map to show where the location rental is located. To make the location a favorite, the user will be able to tap on the heart which will save the vacation rental detail information. Saving the vacation rental, will make it available from the menu.

Screen 3 - Widget



Picture that will be used on the widget.

Key Considerations

The App will be written solely in the Java Programming Language.
Android Studio 3.1.3 will be the IDE that will be used to build the app.
The version of Gradle that will be used is 4.5.1 or higher (stable).

How will your app handle data persistence?

For the data persistence, I will use Room which will provide database access. Room will be used to save the vacation rental favorites. Room will also provide caching of the data. Users will have access to their favorites whether they have internet access or not.

Describe any edge or corner cases in the UX.

The user will be able to select a vacation rental and transition to a screen which provides detail about the rental. On the detail screen the user will be able to return back to the main selection screen. From the main selection screen, the user will be able to select most popular, top rated and favorite rentals. Favorite rentals will only be available if the user has selected any rentals as favorites.

There will be a back button on the detail view, so the user can return back to the main view.

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

1. I will use Glide to handle the loading and caching of images. Glide loads faster than Picasso and Picasso will cache only full-size images, where Glide will cache separate files for each size of an imageView. Glide: [Version `glide:4.7.1`]

2. I will use Data Binding to bind UI components. Data Binding will improve performance because the view hierarchy for "findViewById" is only traversed once for the binding class, fonts can be set directly in the layout XML, MVVM (model-view-viewModel) approach is enabled which allows separating presentation logic from business logic, has callbacks and eliminates creating adapters. [Data Binding: Version 2]

3. I will use ButterKnife to simplify code which includes "findById" methods used to find views on a View, Activity or Dialog, if needed. Butterknife: [Version `butterknife:8.8.1`]

4. I will use Timber to help with debugging and to enhance logging. Timber: [Version `timber:4.7.1`]

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

Describe which Services you will use and how.

- I will use Google Ads
- I will use Google Maps to show the location of the rental property

Next Steps: Required Tasks

Task 1: Project Setup and requirements

- Create structure for project
- Create project to handle activities/fragments
- Create a menu
- Create a strings.xml file. All strings will be stored in strings.xml.
- The app will include support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.
- If the app regularly pulls or sends data to/from a web service or API, the app will update the data in its cache at regular intervals using a SyncAdapter or JobDispatcher.

-OR-

- If the app needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), the app will use an IntentService to do so.
  -OR-
- If the app performs short duration, on-demand requests(such as search), the app will use an AsyncTask.
- Configure libraries
    - Butterknife
    - Data Binding
    - Timber
    - Glide
    - Ads
    - Realm

Task 2: Implement UI for Each Activity and Fragment

- The app will enable RTL layout switching on all layouts.
- Build UI for MainActivity/fragment
    - Use grid layout for main screen
- Build UI for DetailActivitiy/fragment
    - Reference Screen layout (above) for detail view
- Build Settings Activity
- Use Material Design Guide
    - Select App Colors
    - Select App Font
- Add toolbars
- Check for Internet access availability
- Include support for accessibility

Task 3: Implement Build Variants

- Create a build variant for paid and free
- Set up Strings.xml for paid
- Set up Strings.xml for free
- Add placeholder to layout screen for the detail view

Task 4: Set-up Room Database

- Room will be used for the database. [`room_version = "1.1.1"`]
- Save necessary information when heart is tapped on detail screen to db

- Favorites will be saved to a list for display from the menu option
- Implement LiveData and ViewModel for Room (required)

Task 5: Get data from Travel API

Reference this site for a travel API: github.com - public-apis or https://www.programmableweb.com/api/sygic-travel

- Set up models for data from API
- Get data from API

Task 6: Add a widget

- Implement the widget
- If the internet is not available, display the favorites selection when going to the app
- A toast message will be displayed if no favorites are available and/or no internet is available.

Task 7: Implement Google Play Services

- Implement Google maps (Maps/Places)
- Implement Google ads
