GitHub Username: williamspete321

Trail Finder

Description

Trail Finder finds a hiking trail near you! Just hit "Find Me A Trail!" and the app will randomly select a nearby hiking trail based on your location. Let Trail Finder surprise you!

Want to choose yourself? No problem! Just select "Let Me Choose" and you can select from a list of nearby hiking trails to explore.

Intended User

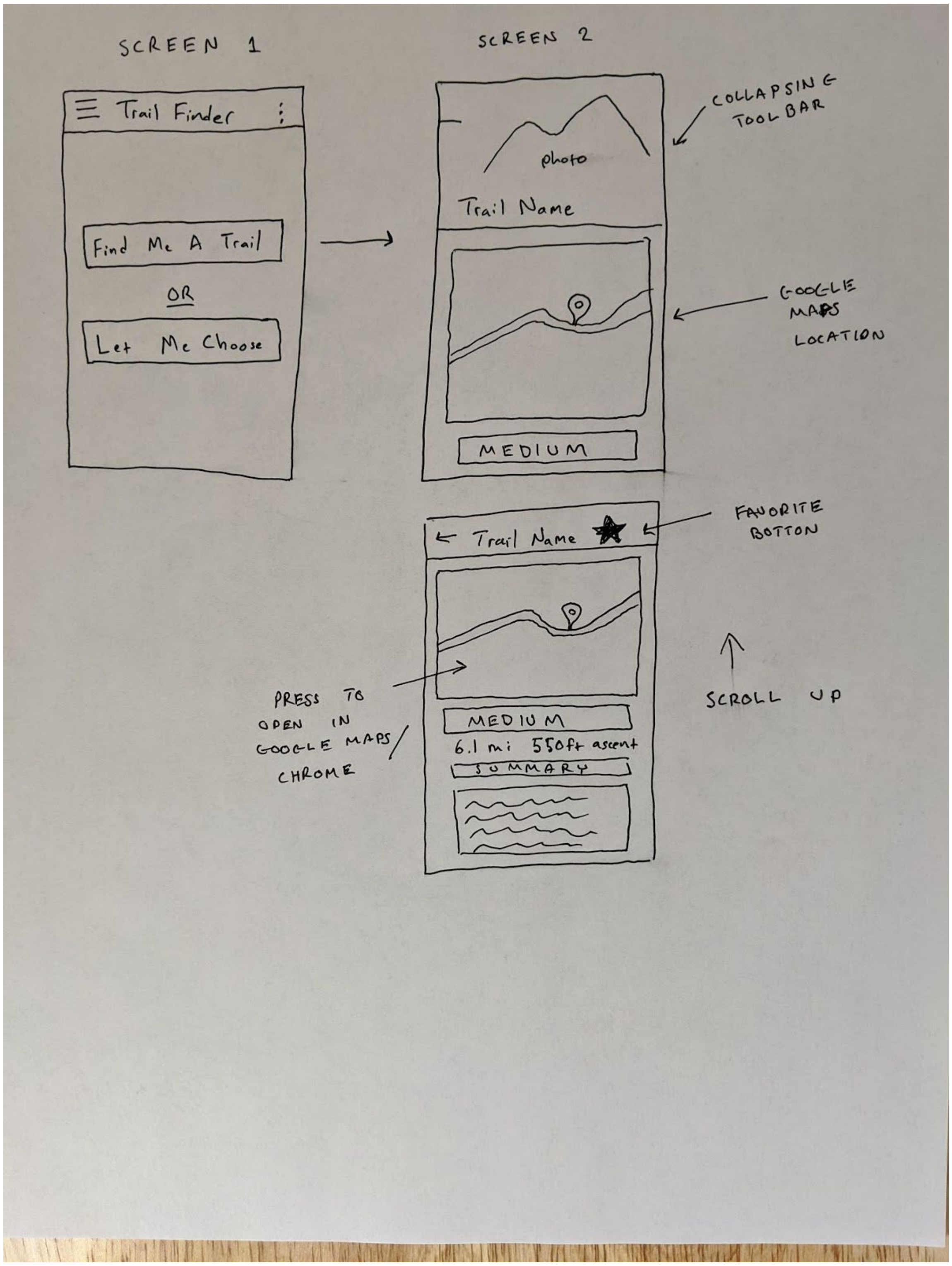
Outdoor enthusiasts. Explorers. Adults. Students. Children.

Features

- Finds nearby hiking trails based on your location.
- Adjust settings to find trails from 30 miles up to 200 miles away.
- Randomly select a trail or view all available trails.
- View the selected trail's information and start point on Google Maps.
- View recently loaded trails offline.

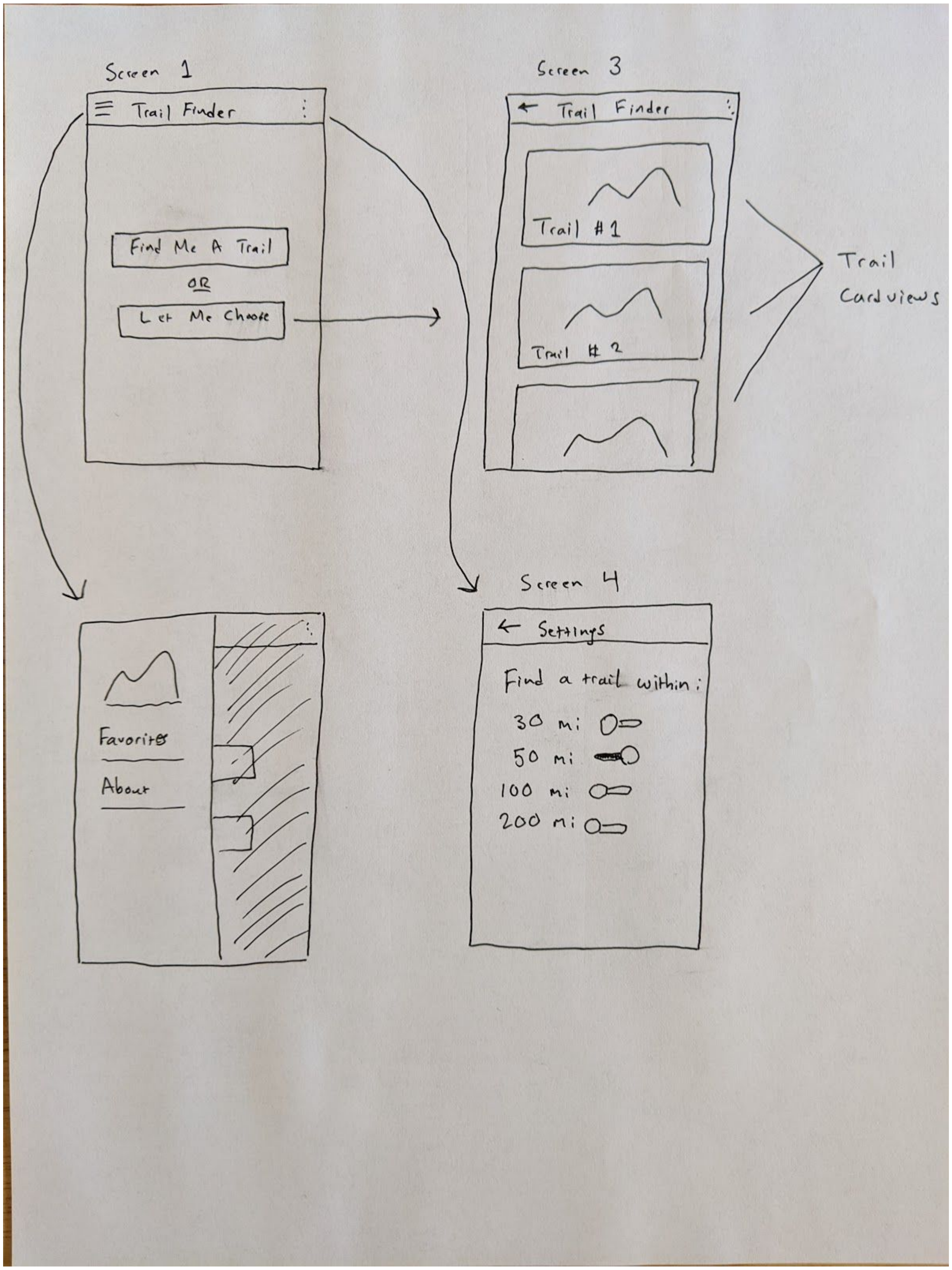
User Interface Mocks

Screen 1 + 2



The image above shows the Main Screen, and the second Trail Description Screen, after the user selects the “Find Me A Trail” button. The Trail Description Screen has a collapsing toolbar that shows a photo of the trail. When the user scrolls up the trail photo collapses. The Trail Description Screen displays the location of the trail on Google Maps. The user can select the google maps image to open the location in Google Maps. Below the map, is the level of the trail’s difficulty (Easy, Medium, or Hard). Below the difficulty level are more descriptions of the trail and a brief summary (“An easy to moderate out and back that culminates at views of Emerald Lake.") The user can also select the star at the top of the toolbar to favorite the trail.

Screen 1 + 3, 4


Screen 1
Trail Finder
Find Me A Trail
OR
Let Me Choose
Screen 3
Trail Finder
Trail #1
Trail # 2
Trail Cardviews
Favorites
About
Screen 4
Settings
Find a trail within:
30 mi
50 mi
100 mi
200 mi

The image above shows the Main Screen and the screen that appears if the user selects “Let Me Choose.” This next List Screen (Screen 3) displays a card view of 5-10 trails that are nearby the user, allowing the user to choose one themselves. After the user selects one of the cardviews, the Trail Description Screen is brought up.

Screen 4 shows what the Settings Screen will look like. There are slider buttons that let the user select how much of a distance (up to 200 mi) away from the user’s location that the app will find a trail for. The user can only select one option, and selecting that slider button to On will turn Off the other slider buttons.

Finally, a view of the Navigation Drawer is visible. The Navigation Drawer will allow the user to select *Favorites* or *About*, etc. *Favorites* will display on the List Screen or a similar looking screen.

Key Considerations

How will your app handle data persistence?

This application will use the Room Persistence library to store trail data. Trails near the user’s location will be collected from the Hiking Project Data API.

Describe any edge or corner cases in the UX.

The application will have a navigation drawer button in the top left corner of the Main Screen toolbar. This navigation drawer will have an option for “Favorites”, and “About”. The top right corner of the Main Screen will have a settings icon that will allow the user to select the Settings Screen.

The Trail Description Screen (Screen 2) will have a back navigation arrow in the top left corner of the toolbar. This back arrow will let the user return to the Main Screen.

The Trail List Screen (Screen 3) will also have a back navigation arrow in the top left corner of the toolbar. This back arrow will let the user return to the Main Screen.

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

Butterknife will be used for data binding.
Retrofit will be used to make network calls.
Picasso will be used to help load images.

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

Google's Maps SDK, Places SDK, and Google Play services will be used to find the user's Android device location.

The Hiking Project Data API will be used to gather nearby hiking trails based on the user's location that is collected from Google services.

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

- Create and include Google API key, and Hiking Data Project API Key.
- Add dependencies: RecyclerView, CardView, Google Play services, Retrofit, etc.
- Grant required permissions.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity and MainFragment.
    - Build UI for NavigationDrawer.
    - Build UI for SettingsMenu.
- Build UI for TrailDescriptionActivity and TrailDescriptionFragment.
    - Building UI for CollapsingToolbar and AppBar layout.
    - Include a Favorite star or similar button in toolbar.
- Build UI for TrailListActivity and TrailListFragment.
- Build UI for SettingsActivity and SettingsFragment.

Task 3: Create Remote Network Package

- Define interface and util classes for Retrofit network calls.

Task 4: Create Local Database Package

- Build database package using MVVM architecture.
    - Create Trail DAO.

    - Create database.
    - Create ViewModel and LiveData classes.
    - Create Repository class to retrieve local data from database or remote data from network.
- Create Executor, AsyncTask, or similar class to use for background thread work.

Task 5: Implement Google Maps SDK and Google Places SDK

- Implement methods to find device's location in MainActivity or helper classes.

Task 6: Implement Communication Between Fragments

- Implement communication between MainFragment/TrailListFragment and TrailDescriptionFragment using ViewModel.

Task 7: Create Widget

- Create TrailWidgetProvider class for home screen widget.

Task 8: Complete Finishing Touches

- Complete SettingsFragment to correctly store user's preferences.
- Fill in NavigationDrawer with Favorite section, About section, and any other section.
- Implement Favorite button in toolbar that saves the current.
- Implement widget functionality; home screen widget should display hiking logo and open the last viewed trail when selected.
