GitHub Username: pestonio

Charted

Description

Have you ever wondered about the history of all those things you see on holiday? Who built this? Who painted that? When did this become a thing? *Charted* is supposed to help you answer those questions!
Without trying to reinvent the wheel, *Charted* will present you with a list of a location's best sights and do it's very best to tell you about them, rather than just show you where they are.

Intended User

This is aimed at travellers. More specifically, travellers who are interested in finding answers about all the things they're taking photos of.

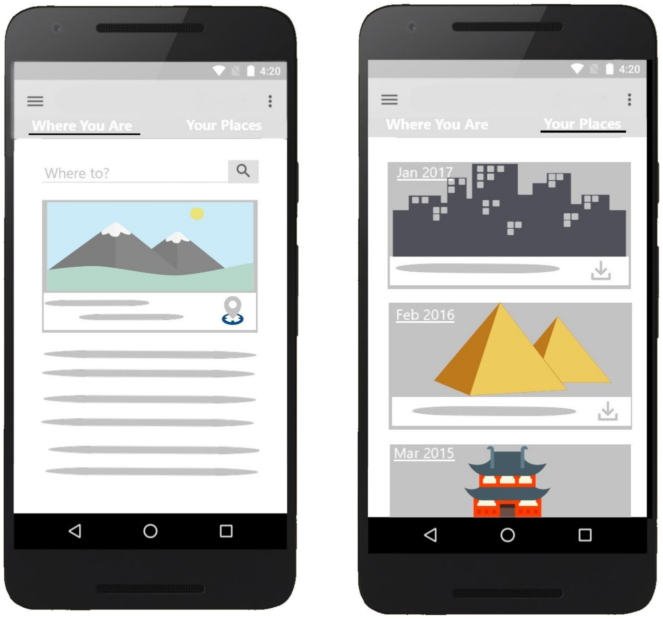
Features

- Gets user's location or allows user to search for one, and plan their trip.
- Displays top attractions and allows users to read detailed information about them.
- Allows users to delve into the history of the places they're visiting.

- Uses Google Places to display a list of attractions and info.
- Uses Google Maps to aid navigation and highlight POIs.
- Uses MediaWiki API to provide detailed information to display under locale and POI.
- Widget displaying current location and random attraction/historical fact.

User Interface Mocks

Screen 1 + 2

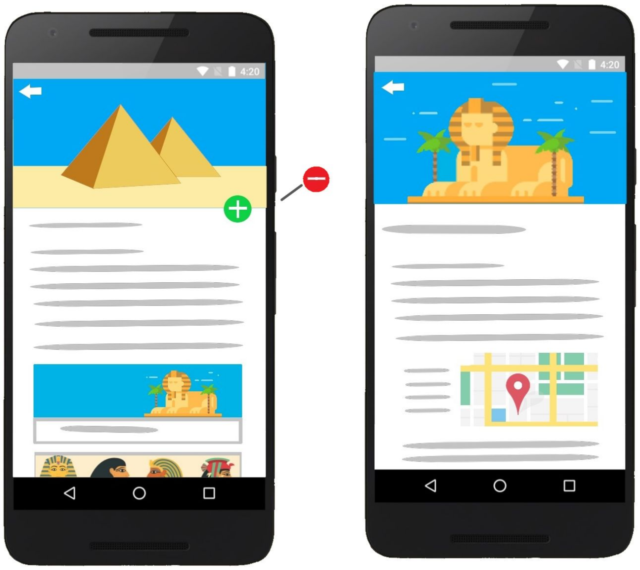


The opening screen will be a two pane viewpager. The ‘home’ page will show the user’s current location, and information related to that. It will also contain a search option to allow users to look up locations.

The second pane, will show the user’s saved places in a recycler view. These will be populated from database references, if a user has opted to save a location. It also allows a user to download place location to enable offline viewing.

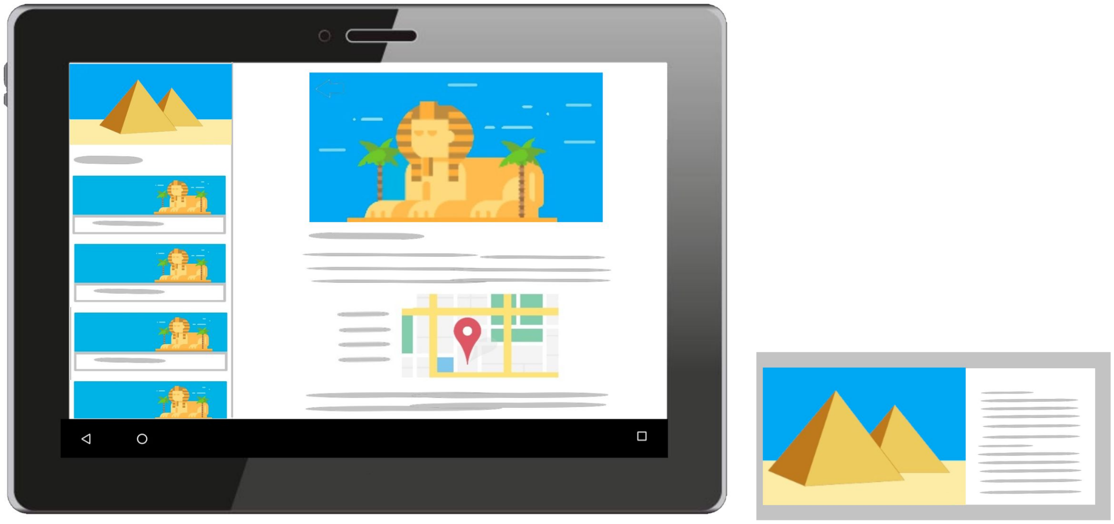
(The settings and navigation icons at the top are placeholder, and may be added later)

Screen 3 + 4



When a user taps on a place, it will display as detailed in the left image. This will show an image, up navigation, an FAB for adding to a user's saved places, an initial blurb about the location followed by a list of local POIs.
If a user taps on a POI, it will display the right image: the detail view of that specific POI. This will contain an image, segmented information, a map view and place data including name, address, website and contact details.

Screen 5 + 6 (Tablet Landscape + Homescreen Widget)



Lastly is the tablet layout. For now this will simply use fragments to show the same info from screens 3 + 4 but side-by-side. Default display on the right will be the location info. POI display will replace it if a user selects a POI from the list.

The homescreen widget (right) will simply show an image, and some selective information about their current location. Tapping it will bring the user to the location details activity/screen 3.

Key Considerations

How will your app handle data persistence?

Data persistence will be handled with an SQLite Database and an original Content Provider. Additional data handling by Google Play Services may be added later using Google Sign In.

Describe any corner cases in the UX.

All pages will provide ‘Up’ navigation to return to the parent activity.

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

Picasso and/or Glide for image loading, Butterknife for findViewById calls.

Describe how you will implement Google Play Services.

Google Maps for displaying and navigating current / user specified location.
Google Places for listing POIs and their respective details.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Add dependencies for Picasso/Glide and ButterKnife libraries.
- Specify targeted and minimum SDKs (enable / disable features as required).
- Ensure Google Play Services are implemented and available.
- Create empty templates for all required activities, ready to code.
- Import any required persistent resources (images and icons etc).
- Specify themes/colours/styles to be used throughout app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Base Activity.
- Build UI for Location Activity.
- Build UI for POIs Activity.
- Build UI for home page Widget.
- Create alternate layout for large / tablet screens.

Task 3: Your Next Task

- Implement functional code for all activities and ensure smooth transitions between activities with placeholder location data.
- Create networking code for Google API Client, Maps and Places.
- Create AsyncTask networking code and JSON parsing for Wikipedia information about locations and POIs.
- Ensure all data is validated and potential crashes handled.

Task 4: Your Next Task

- Strategically test all complex tasks and network calls.
- Create automated tests for all UI behaviour.
- Replace placeholder locations and POIs with live data and test for behaviour.
- Test online and offline functionality, caching.
- Bug hunting, display size / orientation testing, content descriptions, accessibility options.

Add as many tasks as you need to complete your app.

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**"