[Description](#_Toc523151039)

[Intended User](#_Toc523151040)

[Features](#_Toc523151041)

[User Interface Mocks](#_Toc523151042)

[Main screen](#_Toc523151043)

[Detail screen](#_Toc523151044)

[Widget](#_Toc523151045)

[Key Considerations](#_Toc523151046)

[How will your app handle data persistence?](#_Toc523151047)

[Describe any edge or corner cases in the UX.](#_Toc523151048)

[Describe how you will implement Google Play Services or other external services.](#_Toc523151049)

[Next Steps: Required Tasks](#_Toc523151050)

[Task 1: Project Setup](#_Toc523151051)

[Task 2: Implement UI for Each Activity and Fragment](#_Toc523151052)

[Task 3: Database](#_Toc523151053)

[Task 4: Widget](#_Toc523151054)

[Task 5: UI Tests](#_Toc523151055)

**GitHub Username**: ghutchins56

Access Guide

# Description

Google Maps contains accessibility information such as wheelchair access for places, but this is not the information most prominently displayed for a place selected by a user. This app will prominently display accessibility information for places and will allow the user to use Google Maps for other information about the selected place, such as directions. Java, Android Studio version 3.2 and Gradle version 3.2 will be used for development. This app will keep all strings in resource XML files to support language translation and layout direction switching.

# Intended User

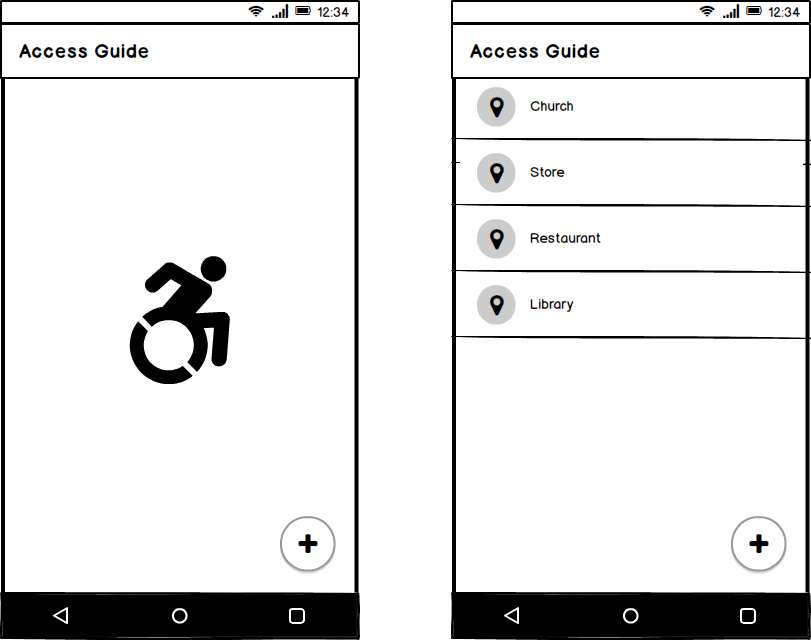
The intended user of this app is anyone with physical mobility challenges, or a friend or family member, who needs ready access to accessibility information for places.

# Features

* Saves a list of places that can be modified by the user (adding or removing places)
* Allows the user to create, edit or delete an accessibility review for a place
* Displays a map, address, accessibility information and user accessibility reviews for a place
* Allows the user to search for a place or select a place on a map

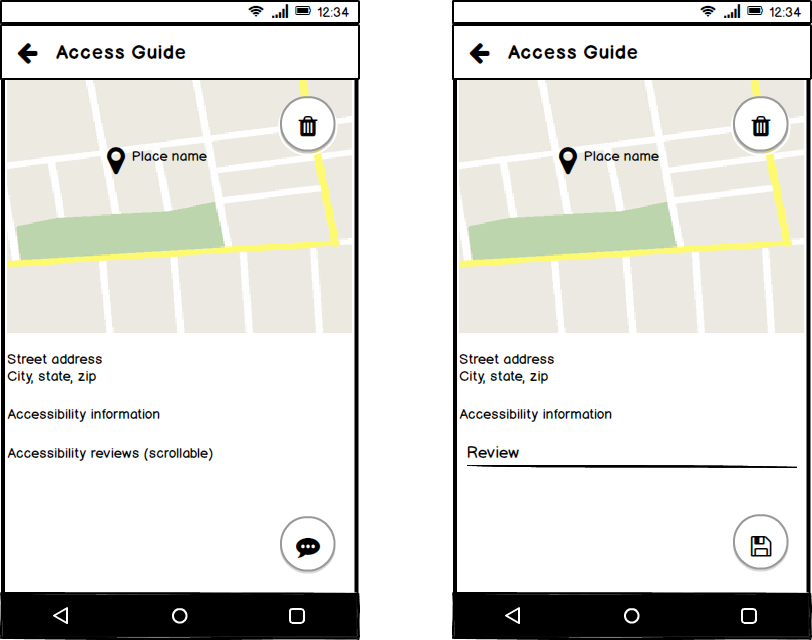
# User Interface Mocks

## Main screen



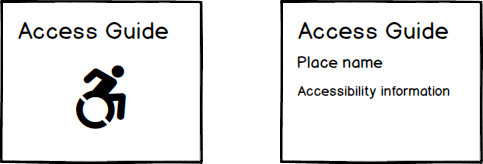
When the app starts, a FirebaseUI sign-in screen will be displayed if the user is not signed in to Firebase. The screen on the left will be displayed if the user is signed in but has not yet saved any places or has deleted all saved places. The screen on the right will display the list of saved places. These screens will be displayed by activity MainActivity.

## Detail screen



The screen on the left will displayed in response to the user selecting a place in the list on the main screen. In response to the user clicking the floating action button in the lower right corner of this screen, the screen on the right will be displayed. In either screen, the user will be able to delete the place shown in the map from the list of saved places using the floating action button in the upper right corner of the map. In the screen on the right, the user will be able to create a new accessibility review or edit (or delete by submitting an empty text field) an accessibility review previously submitted. Each user will not be able to edit or delete reviews submitted by other users. These screens will be displayed by activity DetailActivity.

## Widget



The widget on the left will be displayed after installation or after the user has deleted all the places from the list of saved places. The widget on the right will be displayed for the place most recently selected by the user.

# Key Considerations

### How will your app handle data persistence?

This app will use Firebase Realtime Database version 16.0.2 to store each user’s list of saved places and to store accessibility reviews submitted by users for places. Each list of saved places will only be accessible by its user. Reviews will be accessible by all users.

### Describe any edge or corner cases in the UX.

If one or more of the places in the user’s list of saved places is no longer available in Google Places, the user will be notified when the main screen is displayed, and each such place will be deleted from the list.

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

Google Maps API version 15.0.1 will be used to display a map with a marker for the selected place. Wheelmap API version 2.4.4 will be used to obtain accessibility information for a place. Google Places API version 15.0.1 and PlacePicker will be used to allow the user to search for a place or select it on a map to add it to the list of saved places in response to the floating action button on Screen 1. Firebase Authorization version 16.0.3 and FirebaseUIAuth version 4.1.0 will be used for user sign-in. FirebaseUIDatabase version 4.2.0 will be used for the place list on the main screen and the review list on the detail screen. Volley version 1.1.1 will be used for reading JSON data from Wheelmap.

# Next Steps: Required Tasks

## Task 1: Project Setup

* Add dependencies and permissions for Google Maps API
* Add dependencies and permissions for Google Places API
* Add dependencies and permissions for Wheelmap API
* Add dependencies and permissions for Firebase Realtime Database
* Add dependencies and permissions for Firebase Authorization
* Add dependencies for FirebaseUI
* Add dependencies for Volley

## Task 2: Implement UI for Each Activity and Fragment

* Build UI for MainActivity (layout and Java)
* Build UI for DetailActivity (layout and Java)

## Task 3: Database

* Retrieve JSON data from Wheelmap using Volley
* Use AsyncTask to parse data retrieved by Volley

## Task 4: Widget

* Create AppWidgetProvider class
* Create AppWidgetProvider XML file

## Task 5: UI Tests

* Create instrumented tests for MainActivity UI
* Create instrumented tests for DetailActivity UI