**Activity: Finding a Ride**

The Find Ride Activity implements Google Maps to display several markers, indicating Rides from other drivers. This can be accessed by tapping “Find Ride” on the Main Activity (or Ride List).

When the activity loads, it will display the map, and send a JSON Array request to the server to get all of the available rides. The request will check each ride in the Array if the “completed” field has been set to “true.” If the ride has been completed, its iteration is skipped. Once a ride has been processed, its Marker is added to the map with a click event that opens an InfoWindow, and clicking the InfoWindow sends an Intent with the ride’s ID number, and the user’s current coordinate to display the RideDetails Activity.

At the top of the screen is a Spinner that presents a drop-down list of all campuses. The user can select a campus as their destination, and another request is sent, filtering out all campuses that don’t match the search criteria.

At the time of writing this, making the request filter out rides whose departure times have passed is desirable. I also kind of cheesed this Activity by making Markers appear over each campuses coordinates. A the moment, Markers will overlap each other, but I’m looking into Android’s Marker clustering to display markers within a general area, but never overlap.

I don’t think there’s anything left to be done for the Map itself, but if you need a reference, I followed this tutorial to get it up and running: <https://www.youtube.com/watch?v=NHXa96-r8TY>

**Manifest Details**

FindRide.java implements this activity in the Android Manifest. Like most of the Activities, its screenOrientation is restricted to “portrait.” This activity inherits the app’s theme.

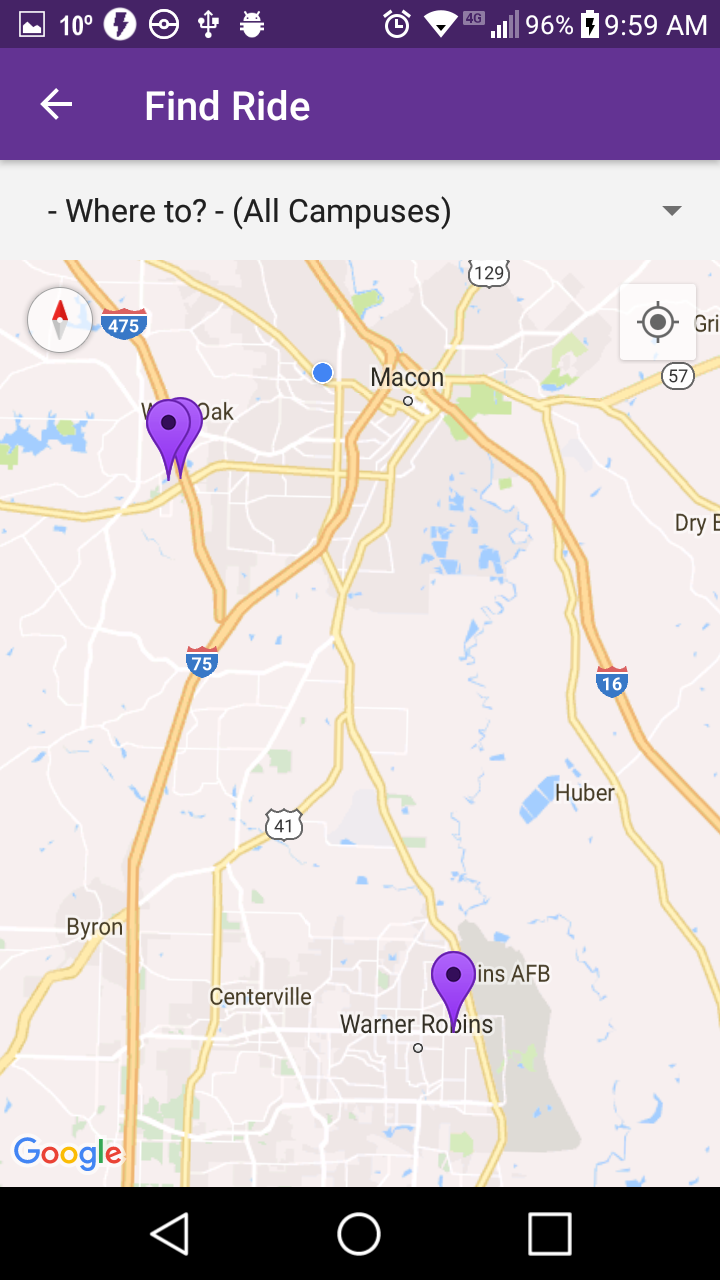
Above FindRide’s <activity> entry, there is a <meta-data> tag that references the Google Maps API Key stored in google\_maps\_key.xml.

The Intent filter provides standard access when referenced by name.

**Layout File**

Activity\_find\_ride.xml

* This one is…weird. The entire activity consists of two objects: a Spinner to provide the dropdown list (Stored inside a TextInputLayout), and an empty Fragment. The Fragment is where the Map is displayed, but it is created by the SupportMapFragment object that is instantiated in the Activity’s onCreate() function.
* In the .xml’s text mode, there is a commented line of code for a TextInputEditText. Back when the app was developed for the Code for Atlanta competition, the app had to be modified outside of school use, so it needed to accept other destinations. This search bar allowed the user to enter a building or an address, and the code behind would Geocode the appropriate address, and display any matching markers.
* map(Fragment)
  + Displays the map
* searchBarInput(Spinner)
  + A drop down list to create campuses. This is populated by the setUpSearchBar() method called onResume().
* ---searBarInput(TextInputEditText)--- UNUSED
  + A text bar that would allow the user to type in the address or building they would want to travel to. This has been disabled in favor of the campus selector.



**Class File**

* FindRide.java implements the Finding a Ride Activity. Lines 309-453 contain a massive chunk of code that has been commented. It was supposed to support a TextInputEditText view that would Geocode the user’s input to determine which markers should be displayed. When the Spinner was implemented, this code was abandoned.
* Variables
  + –queue: RequestQueue – Sends JSON Requests to the server
  + –prefs: SharedPreferences – Retrieves the user’s access token from the phone
  + –mMap: GoogleMap – the Map itself. Provides access to functions like moving the camera and placing Markers
  + –client: GoogleApiClient – I think this contains an instance of the client using the app?
  + –locationRequest: LocationRequest – Grabs the user’s location
  + -currentLocationMarker: Marker – Displays the user’s location as a blue dot on the map
  + +RQUESTION\_LOCATION\_CODE: int – Access code to request location
  + –coordinatesList: ArrayList<double[]> -- Stores used coordinates to prevent Markers from overlapping.
  + –MARKER\_OFFSET: double – the amount to shift a Marker by if it shares another Marker’s coordinates.
  + –userLat, userLng: double – Saves the user’s coordinates to pass during an Intent
  + –markers: HashMap<Marker, HashMap<String, String>> -- Stores Markers and their ride information
  + –markerValues: HashMap<String, String> -- A Marker’s ride information that is displayed in their info window.
  + –selectedCampus: String – Campus selected from the dropdown list.
  + –searchBarInput: TextInputEditText ---UNUSED--- Text box to store a user-typed address
  + –searchInput: String ---UNUSED--- stores searchBarInput’s text.
  + –storedAddress: String ---UNUSED--- stores the geocoded address from searchInput.
* Methods
  + \*onCreate(savedInstanceState) : void
    - Displays the maps activity layout, and initializes prefs and queues.
    - Asks the user permission to access their location, and then it turns the map Fragment into a Google Map.
    - Lastly it creates the back button in the app’s toolbar.
    - The next code would have set up the Search Bar if the activity is using a TextInputEditText. It would have also set a listener for whenever the user taps the “done” button on the keyboard, causing the app to display markers with the desired location.
  + \*onResume(): void
    - Calls the setUpSearchBar() method, which automatically fires the findAllRides() method. Since this is onResume, it is called whenever the focus returns to the FindRideActiity.
  + –clearMarkers(): void
    - Clears all markers on the map. This is called any time findAllRides() is called.
  + –setUpSearchBar: void
    - Grabs the campus\_array from the campus\_coordinates.xml file and populates their values in the searchBarInput spinner.
    - Sets an ItemSelectedListener, where selecting the first option (“Where to?”) will display all rides, and selecting any other option will store the desired campus in selected campus. Both cases will call findAllRides, but case 0 always runs as soon as the Spinner is created.
  + –findAllRides(campus): void
    - Clears all markers, then sends a JSON Array Request to the server to get all the rides.
    - Loops through each iteration of the ride. If a ride is marked as completed, then the iteration is skipped. If the ride’s departure time has passed, this iteration is skipped. If the campus parameter does not contain the campus city’s name, the iteration is skipped.
    - The request stores the ride’s latitude, longitude into double variables, and the trip id, origin city, destination city, the available seats, and profile picture are stored into a new markerValues HashMap. A marker is created using the longitude and latitude, and the marker is stored into the markers HashMap, using the current Marker as the key, and the markerValues as the value.
    - Next, markers are given a click event that displays their InfoWindow.
    - InfoWindows are set up using the Map’s setInfoWindowAdapter method. The view is created in getInfoWindow contents, and it will retrieve a marker’s value from the marker’s HashMap.
    - Lastly, the InfoWindow is given a click event that will open the Ride Details activity, sending the tripID, user latitude, and user longitude as Extras.
    - If there is a server error when accessing rides, the activity is reset.
  + –setMarkerOffset(coordinates): double
    - When retrieving a marker’s coordinates, first check if the pair of coordinates has been used in the coordinatesList. If the coordinates were used previously, give them a random offset, and check again.
    - If the coordinates weren’t used, return the coordinates + the offset, and store them in the coordinates list.
  + –GeocodeAddressTask ---UNUSED---
    - An AsyncTask that generates the correct address, based on the user’s search input, this immediately calls the FindMatchingRidesTask, which performs the same function as the findAllRides() method.
    - This class is unused because input is handled through a Spinner, rather than a text box.
  + +onOptionsItemSelected(item): Boolean
    - Sets the back button in the app’s toolbar to return to the previous activity.
  + +checkLocationPermission(): boolean
    - Requests the user’s permission to use their location.
  + +onRequestPermissionsResult(requestCode, permission, grantResults): void
    - Decides whether to display the map, based on checkLocationPermission’s result.
  + +onMapReady(googleMap): void
    - Creates the googleAPI client if permission is granted, and also enables the location tracker
  + \*bulidGoogleApiClient(): void
    - Creates an instance of the API client, and connects it to Google Maps
  + +onLocationChanged(location): void
    - Updates userLat and UserLng as the user moves.
    - It also creates a blue dot to represent the user’s current location.
  + +onConnected(bundle): void
    - Sets the interval on which to update the user’s location
  + +onConnectionSuspended(i)
    - Unused. Required for implementing the GoogleApiClient.ConnectionCallbacks interface
  + +onConnectionFailed(connectionResult)
    - Unused. Required for implementing the GoogleApiClient.ConnectionCallbacks interface

**Other Files**

campus\_coordinates.xml

* Contains two string arrays, one with each campus, and another with each campus’ address.

google\_maps\_api.xml

* Contains the Google Maps API Key for this activity. Do not use this key anywhere else or it will be invalidated, and a new one will need to be obtained from Google.