**Service: Location Tracking**

The Location Tracking Service is a class that is used when the drivers and passengers are in an active ride. The Service pings for the phone’s geographic location after a set duration and posts the coordinates to the Knight Rider server.

An instance of the Service is created on the Ride Details Activity. For the driver, the Service begins when the driver taps the “Start Ride” button within the activity, and ends if the user taps “Cancel Ride.” For the passenger, it begins when they tap the “Join Ride” button, and ends if they tap the “Leave Ride” button.

In the current implementation, the Service will note the user’s current location every 10 seconds, constantly checking if the user has moved within a certain radius of the destination (300 kilometers at the moment). When the user has arrived, a notification is sent to the driver, telling them they have arrived at the campus. Passengers are also sent a notification, but they have the option to tap the notification to open an Activity that allows them to leave a rating a review of the ride.

This service’s StartCommand returns START\_STICKY when the app launches, which makes the Service restart if the device runs out of RAM, but this seems to cause the Service to carry over if the user logs out and logs in with another user while active.

**Manifest Details**

Services like this one are given a <service> tag within the Android Manifest. The Location Tracking service is implemented by the TripLocationService.java class. Its “enabled” property is set to “true,” which allows the Service to be activated, and its “exported” property is set to false, which prevents other apps from having access to it.

**Class File(s) and Internal Classes**

TripLocationService.java

* This class implements the Location Tracking Service, but most of the class’s functionality resides in an internal class “TripLocationListener.” The Listener was given its own description below.
* The outer class’s main purpose is to handle the Intent sent from the Ride Details activity and to destroy the Location Listener when the Service is stopped.
* Variables
  + +isActive: Boolean – A static variable that tracks if the Service is active. This is used to determine if the user is part of an active ride or not.
  + –TAG: String – Unique tag for the Service
  + –mLocationManager: LocationManager – used to get the GPS Provider for the LocationListener
  + –LOCATION\_INTERVAL: int – How long the Service will wait before pinging the user’s location. In this case, it’s set to 10 seconds (10,000 milliseconds)
  + –LOCATION\_DISTANCE: float – How far the user must travel before the pinging the user’s location. I set it to 0 meters.
  + –DESTINATION\_RANGE: float – How close the user must be to the destination before the service will send a notification and end itself. This value is set to 400 meters.
  + –prefs: SharedPreferences – Get the user’s Id and access token for requests
  + –queue: RequestQueue – Sends JSON Requests
  + –userId: String
  + –userLat, userLng, userSpeed: double
  + –destName: String
  + –destLat, destLng: double
  + –tripId: String
  + –driverId: String
  + –DESTINATION\_NOTIFICATION\_ID: int – Notifications require ID numbers to determine if they should be updated
  + –mLocationListener: TripLocaitonListener – Instance of the internal Location Listener class that passes the Location Manager’s GPS\_PROVIDER into the constructor.
* Methods
  + +onStartCommand(intent, flags, startId): int
    - This method is called whenever the startService method is used.
    - Prefs and queue are initialized,
    - destName, destLat, destLng, tripId, and driverId are pulled from the Intent’s Extras.
    - userId is pulled from the phone
    - isActive is set to true,
    - The method returns START\_STICKY, which will automatically restart the activity if the device runs out of memory.
  + +onCreate(): void
    - Sets the location manager to request Location updates, using the gps provider, interval, distance, and TripLocationListener as parameters
  + +onDestroy(): void
    - This is called whenever the Service is *explicitly* told to stop. Because START\_STICKY is in effect, this method will never run when the phone runs out of memory.
    - This tells TripLocationListener to stop receiving location updates, and it sets “isActive” to false, allowing the user and driver to join more rides.
  + +onBind: IBinder(intent)
    - Simply returns null. This was a required method.

-TripLocationListener

* The internal class that actually deals with how the user’s location is handled, and what is done with the user’s location.
* Variables
  + There are no internal variables.
* Methods
  + +onLocationChanged(location): void
    - Retrieves the user’s longitude, latitude, and speed
    - It seems the app creates a new Location called “destLocation” and sets its longitude and latitude, and calculates the distance to the location in a float variable.
    - An if statement checks if the distance to the location is under the DESTINATION\_RANGE constant. If so, call the CompleteTrip() and stopSelf() methods to send a request to the server and stop the service.
    - Call postLocationToServer using the userLat, userLng, and userSpeed as parameters.
    - **Note about the second point:** Looking back at this, The Location object could be a variable. This method is called multiple times, so basically the same object is being created, and having its distance calculated every 10 seconds.
  + **–**postLocationToServer(lat, lng, speed): void
    - Build the request url using the user’s id,
    - Create a new HashMap of parameters, using the user’s Id, latitude, longitude, and speed
    - Lastly, send a JSON Object POST Request to the server.
  + –completeTrip(): void
    - This method sends a complete request to the server, but this can only be done by a ride’s driver.
    - Calls the displayNotificaiton() method
    - If the userId does NOT match the driverId, end this method.
    - Else, build the complete trip request url, and send a JSON Object POST Request with no parameters.
  + -displayNotifcation(): void
    - Create display a notification on the phone. This shows two different notifications for the driver and passenger.
    - If the driver id and user id match, the notification will simply display a “You have arrived!” message.
    - If they DON’T match, a new Intent to start the Rate Ride Activity is created, an a clickable notification is created, appending a “Tap to leave a review” message.
  + **These last few methods are unused, and were required as part of the LocationListener Interface**
    - +onProviderDisabled(provider): void
    - +onProviderEnabled(provider): void
    - +onStatusChanged(provider, status, extras): void