**Activity: User Profile**

This is the activity where the user can set their profile information and create and manage their registered vehicles. This is accessible by clicking “Profile”(R.id.nav\_profile) in the navigation drawer on the Main Activity.

Despite the two tabs that are anchored to the top of the screen, this activity does NOT use a Page Viewer to display the different layouts as Fragments. Instead, this Activity is managed by three different classes. ProfileMain.java manages the entire Activity, while ProfilePersonal.java manages the Fragment for updating the user’s name and contact information. ProfileVehicle displays the user’s list of cars, and allows the user to create, modify, or delete existing cars.

**Manifest Details**

This activity is implemented with ProfileMain.java in the Android Manifest. Like most of the Activities, its screenOrientation is restricted to “portrait.” This Activity inherits the app’s theme.

Android:windowSoftInputMode is used here, and is set to “adjustPan”. This prevents the saveButton on the ProfilePersonal fragment from being pushed up by the virtual keyboard.

The Intent filter provides standard access when accessed by name.

**Layout Files**

activity\_profile\_main.xml

* The container layout for the personal information and vehicle layouts. It only contains two Views to change the layout, but fragment\_profile\_personal is inflated first by default.
* personalTab & vehicleTab (ConstraintLayout)
  + While these look like buttons, they are actually ConstraintLayouts with a drawable as a background. Click events are sent in the ProfileMain.java code behind, but these buttons swap the current layout.

fragment\_profile\_personal.xml

* Retrieves and displays the user’s profile photo, name, contact information, and ratings when the Fragment starts. The user can tap the saveButton to send a PUT request to the server, updating their information.
* userImageEdit (TextView):
  + Click event is set in the ProfilePersonal.java code behind, but this allows the user to choose an image from their phone’s Gallery app to use as a profilePhoto. The image is assigned to driverImage, and is immediately sent to the server as a request.
* ratingsLayout (LinearLayout):
  + A layout to display the user’s rating. The ratingBar will display the user’s average rating, but the bar is not intractable on this screen.
  + ratingsTotal (TextView):
    - Displays the number of ratings the user has received.
    - It’s click event is handled in the code behind, but if the user has at least one rating, this number can be clicked to view all of their reviews.
* saveButton
  + The onClick event is handled in the ProfilePersonal.java code behind. When this button is clicked, it will validate the text in the Input Layouts, and send a PUT request to update the user’s profile information.

fragment\_profile\_vehicle.xml

* Operates similarly to the Ride List on the Main Activity. This Fragment displays all of the user’s registered vehicles.
* vehiclesListView (ListView)
  + A ListView that displays a series of item\_profile\_vehicle layouts through the Fragment’s ProfileVehicle.java code-behind.
* createCarButton (Button)
  + The onClick event is handled in the ProfileVehicle.java code behind, but this button opens the CreateCar Activity

Item\_profile\_vehicle.xml

* Represents a user’s registered vehicle. This displays a vehicle’s nickname,make, model, year, number of seats, and the vehicle’s color.
* vehicleNameEdit (TextView)
  + The green “Edit” text opens the Create Car activity, but modifies the activity to allow the user to edit or delete the chosen vehicle.

|  |  |
| --- | --- |
|  |  |
|  | |

**Class Files**

ProfileMain.java

* The main class of the User Profile Activity. This layout displays either the Profile or Vehicle Fragments, and provides a control to swap between them, using Android’s FragmentManager.
* Variables
  + -currentView: String – The current fragment that is displayed
  + -fragmentManager: FragmentManager – Provides functionality to change the currently displayed Fragment.
  + -personalTab: ConstraintLayout – The button-like tab that displays the Personal Fragment when clicked.
  + -vehicleTab: ConstraintLayout – the button-like tab that displays the Vehicle Fragment when clicked.
* Methods
  + \*onCreate(savedInstanceState): void
    - Creates and displays the empty activity\_profile\_main, and creates a Back button in the app’s toolbar.
    - Creates an Instance of the FragmentTransaction class, creates an instance of the ProfilePersonal class, calls the commit() method to display the Personal Fragment within the main layout, and sets currentView to “personal”
    - Finds personalTab and vehicleTab and set their click events to showPersonal() and showVehicles()
  + +onOptionsItemSelected(item): Boolean
    - Go back to the Main Activity, or Ride list, when the back button is clicked in the toolbar.
  + -showPersonal(): void
    - Do nothing, if the currentView is “personal”
    - Otherwise, show the Profile Personal Fragment and set the currentView to “personal”
  + -showVehicles(): void
    - Do nothing, if the currentView is “vehicles”
    - Otherwise, show the Profile Vehicle Fragment and set the currentView to “vehicles”
  + -changeTabColor(): void ---UNUSED---
    - Swap the tab color between green and purple, but this was disabled because the current implementation doesn’t work on older Android devices, and fixing it was low priority…

ProfilePersonal.java

* The Profile Fragment of the User Profile Activity. This is where the user can change their profile picture and personal information. The Edit button under the profile picture sends a request to change the avatar as soon as the user selects an image from their gallery. Meanwhile, the saveButton will send a request to a separate API for updating the user’s personal information, but it does not include the Profile Picture when doing so.
* Variables
  + -firstNameInput: TextInputEditText
  + -lastNameInput: TextInputEditText
  + -addressInput: TextInputEditText
  + -zipInput: TextInputEditText
  + -phoneInput: TextInputEditText
  + -profileImage: ImageView
  + -ratingBar: RatingBar – displays the user’s average rating
  + -ratingTotal: TextView
  + –queue: RequestQueue – Holds JSON requests
  + –prefs: SharedPreferences – The devices preferences, used to retrieve the access token and userId.
  + –userId: String
  + username: String
* Methods
  + +onCreateView(inflater, container, savedInstanceState): View
    - Inflates and displays fragment\_profile\_personal and calls a method to initialize the inputs and rating bar.
    - Creates a variable for userImageEdit and sets its onClickListener to call getProfilePic()
    - Creates a variable for saveButon and sets its onClickListener to call updateUser()
    - Initializes queue and prefs, and retrieves the userId from prefs.
    - Finally, it returns the view.
  + +onStart(): void
    - Calls getUserInfo()
  + +getProfilePic(): void
    - Creates an Intent to open Android’s Galler to select an image. When an image is selected, the Gallery is closed, and onActivityResult() is immediately called
  + +onActivityResult(requestCode, resultCode, data): void
    - This method is called automatically when the user selects an image.
    - The image is compressed and converted into a Base64 String, and is passed as a parameter to upload the new picture.
  + –uplloadNewProfilePic(encoded): void
    - Takes the Base64 string and sends a JSON Object POST request to the server, uploading the new profile picture.
  + –updateUser(): void
    - Validates the text in the input fields, and if every field passes the check, the text is stored in a HashMap
    - The HashMap is then sent in a JSON Object PUT request
  + –getUserInfo(): void
    - Sends a JSON Object GET request to pull a user’s information, based on the stored userId.
    - All values are then set to the InputFields in the layout.
    - This will also calculate the number of ratings the user has received, and their average rating received. If there is at least one rating available, the ratingsTotal’s onClickListener is set to open the Reviews activity.
  + –setProfileLabels(v): void
    - Initializes all the InputText variables, the rating bar and text, and the profile image.
  + –round(value, places): double
    - Used by getUserInfo() to round the average rating to the nearest fifth.

ProfileVehicle.java

* The Vehicles section of the User Profile Activity. This Fragment mainly consists of a ListView that will display a series of layouts to represent individual vehicles. This class contains an internal Class for handling vehicles.
* Variables
  + There are no variables for the scope of this class.
* Methods and Internal Classes
  + +onCreateView(inflater, container, savedInstanceState): View
    - Displays the fragment\_profile\_vehicle layout
    - Finds and assigns the createCarButton, and sets its onClickListener to start the Create A Car Activity
  + +onStart(): void
    - Call the method to retrieve all of a user’s cars.
  + –getAllCars(): void
    - Creates a new ArrayList<HashMap> to hold each car
    - Create a SharedPreferences instance and retrieve the user’s ID
    - Sent a JSON Array GET request to retrieve all vehicles
    - Loop through each vehicle. For each iteration, create a new HashMap that accepts a vehicle’s values, then add the HashMap to the ArrayList.
    - Pass the ArrayList into the ListAdapater’s constructo
  + VehicleAdapter
    - Internal Class that creates each vehicle’s layout, and displays them on the ListView.
    - Variables
      * -vehicles:ArrayList<HashMap> -- Stores a copy of the list of Vehicles.
    - Methods
      * +VehicleAdapter(context, vehicles)
        + Store a copy of the list of vehicles
      * +getView(position, convertView, parent): View
        + Inflate a single item\_profile\_vehicle layout and retrieve a HashMap from the vehicles ArrayList
        + Set an onClickListener for the editLabel TextView on the layout. This will make the Edit Button open the Create A Car Activity, but it will send an Intent that changes the Activity to where they can edit the selected car.
        + Assign Values to the layouts other labels and return the view.