**Activity: Creating a Car**

This Activity allows the user to register a vehicle to their account. It is accessible when clicking the createCarButton from the Profile Vehicle Fragment. On this page, the user can enter the car’s Make, Model, Capacity, and Color.

But this Activity has another function. If the user clicks the Edit text on a vehicle’s list item, the Activity changes from Create a Car, to Edit A Car. The layout stays roughly the same, except the text boxes will contain the current vehicle’s information. The createCarButton’s onClick function changes to an Update function, and a previously hidden deleteCarButton becomes visible.

**Note:** As of writing this (11/12/17),in the request for creating the car, the server isn’t actually set to handle color changes. The user will have to edit the car later to add a color.

**Manifest Details**

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

The Intent filter provides standard access when accessed by name.

**Layout File**

Activity\_create\_car.xml

* The layout for the Create Car activity. Because this activity changes depending on the Intent, the layout is slightly different.
* All the TextInput fields are empty when creating a car, but they will contain values if the user is editing an existing car.
* createCarButton (button)
  + onClick: addCar – Calls the Add Car function in the code behind.
  + The default text is “Add Car”, but when editing a car, the text is changed to “Save Changes”. The onClickListener is also set to call the updateCar function instead.
* deleteCarButton (button)
  + The onClickListener is set in the code behind to call the deleteCar function.
  + When the Activity is set to Create a car, this button’s visibility property is set to GONE. This means that it is both hidden from view, and it doesn’t take up space. When editing a car, the visibility is set to VISIBLE.

|  |  |
| --- | --- |
|  | C:\Users\Allan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Create Car 1.png |

**Class Files**

CreateCar.java

* This class implements the Create Car activity.
* Variables
  + –queue: RequestQueue – Sends JSON Requests to the server
  + –prefs: SharedPreferences – Obtains the userId stored on the phone
  + –userId: String – the current user’s id
  + –carMake: TextInputEditText
  + –carModel: TextInputEditText
  + –carCapacity: TextInputEditText
  + –carColor: TextInputEditText
  + –deleteCarButton: Button
* Methods
  + \*onCreate(savedInstanceState): void
    - Displays the activity\_create\_car layout
    - Initializes all the variables
    - Creates a back button in the app’s toolbar
    - Checks the Intent’s Boolean Extra. If the extra is “true”, then a method call is sent to modify the Activity for Editing a car. If the extra is “false”, then the deleteCarButton is hidden from the app’s layout.
  + +onOptionsItemSelected(item): Boolean
    - Returns the app to the previous Activity
  + –createEditCarActivity(): void
    - Changes the Create Car activity to an Edit Car activity.
    - It renames the Activity in the title bar, it changes the createCarButton’s onClickListener to call the updateCar() method, and renames the Button’s text to “Save Changes”.
    - It also modifies the left and right margins of the createCarButton and deleteCarButton, respectively. These are not set in the layout by default because when the deleteCarButton is hidden, the createCarButton is centered and stretched. The unnecessary margin would make the button appear off-center
    - Calls the getCarValues() method to set the default text in each TextInput
  + –getCarValues(): void
    - Retrieves a car’s ID number from the Intent, and appends it to the url for a JSON Object GET Request.
    - A successful response will set each TextInput to the appropriate values, while a failed response will exit the Activity.
  + +addCar(view): void
    - The default method called by the createCarButon.
    - It validates the text stored in each input box, and stores them into a HashMap if Validation is successful.
    - The HashMap is converted into a JSON Object and sent in a JSON Object POST request. A successful response will display a message and exit the Activity. An failed response will display an error message and exit the Activity.
  + –updateCar():void
    - Similar to addCar(), except the HashMap is sent as a JSON Object PUT request.
  + –deleteCar(view): void
    - Sent by the deleteCarButton in the layout.
    - Retrieves the carId sent through the Intent, and sends a JSON Object DELETE Request to delete a car on the server.