Antonio Scalfaro

CMSC 215 / 6381

Assignment 3 – Trip Cost Estimator

09/23/2023

**UML Diagram**

|  |
| --- |
| **TripCost** |
| days:int  gasolineCost:double  gasMileage:double  hotelCost:double  foodCost:double  attractions:double  distance:double  df:DecimalFormat |
| TripCost()  gasCost()  totalTripCost() |

|  |
| --- |
| **Project3 extends Application(JavaFX)** |
| grid:GridPane  scene:Scene  tc:TripCost |
| start()  createNumericTextField()  isValidInput()  main() |

|  |
| --- |
| **Project3\_UnitTest (junit)** |
|  |
| testGasCost()  testTotalTripCost() |

**Test Plan:**

Testing the Trip Cost Estimator program takes both a junit test (for the TripCost class methods) and manual testing (for the GUI). There are two junit tests:

* **testGasCost():** tests the gasCost() with an instance of TripCost(1000, 3.95, 30, 2, 100, 100, 30)
* **testTotalTripCost():** tests the totalTripCost() with an instance of TripCost(1000, 3.95, 30, 2, 100, 100, 30).

These tests both pass when run. Creating junit tests for a GUI were a bit too complicated for this week so manual testing is used. The TextFields are created so only the proper format is used (standard notation for decimal numbers), no special characters beside the single decimal can be used to include (-) which means negative numbers are not accepted. The total cost of the trip is displayed in a TextField that does not accept any input and only displays the total once the Calculate Button is pressed. The GUI passed all manual tests.

**Lessons:**

There were many valuable learning experiences during this week’s assignment. First, learning the basics of JavaFX was enjoyable and made easier by my experience in React, but I still had to learn the syntax and structure. Second, I had to learn how to structure the files correctly to get the program working correctly in eclipse while using javaFX. Another lesson learned was using junit. This is my first week running junit tests, just learning the basics was a great experience, unfortunately I need more experience before creating junit tests for a GUI. Lastly, while I made the GUI difficult to accept invalid user input, I provided no protection to the TripCost class in the case of an injection of invalid data. As a final tidbit, I tried to create listeners for the ComboBoxes so that if one changed they would all change, however this did not go well and I removed the functionality.