Criterion B: Design

Design of the Solution

There will be two objects:

- Driving main
- Day

Driving Main has the following properties:

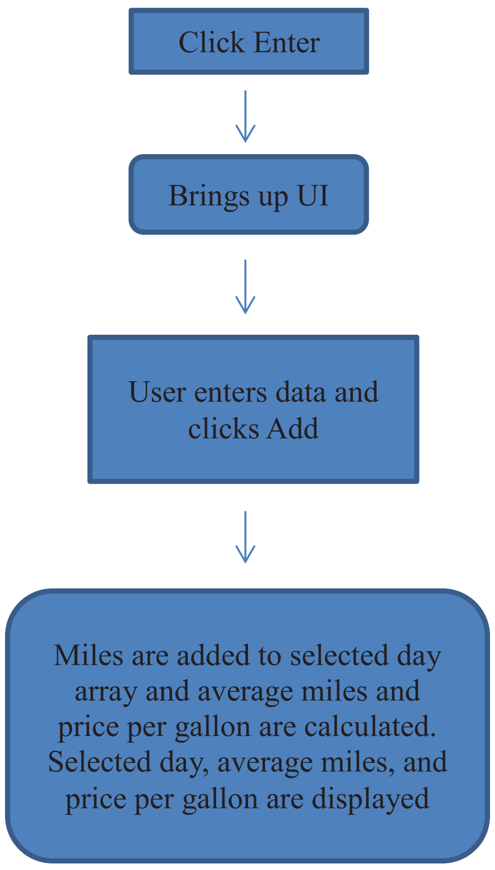
-Enters day of week
-Enters miles driven
-Displays and can shuffle through days with average miles driven
-Enters dollars spent at pump
-Enters miles driven since last fill up
-Updates list box of average dollars per mile
-Receives destination distance, trip, and number of people
-Calculates and displays price per rider

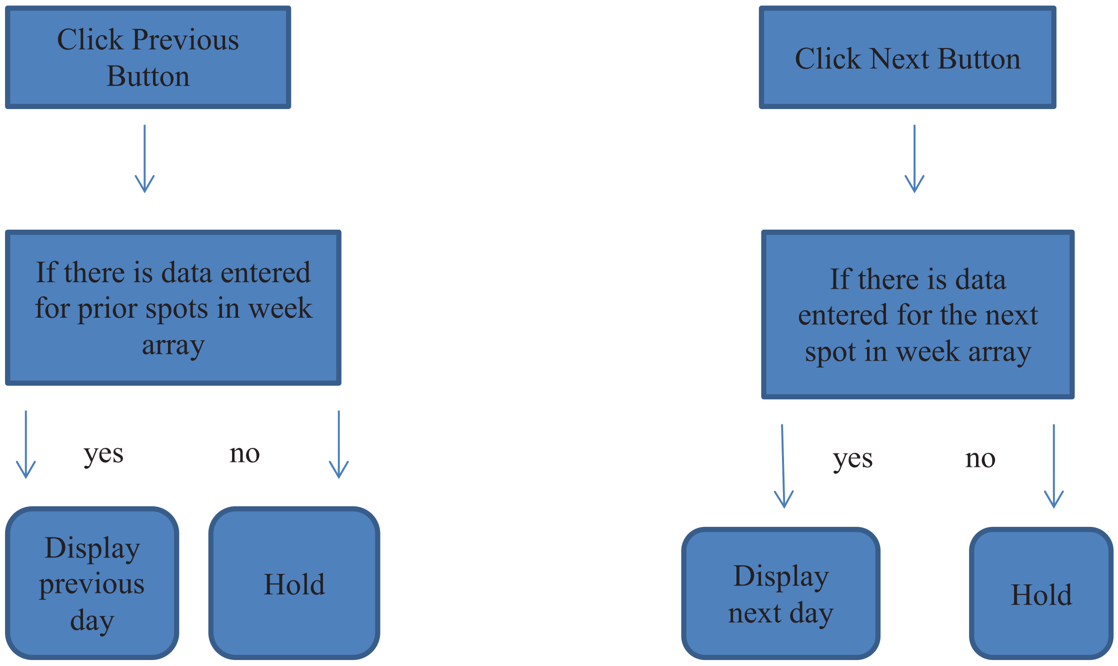
Day has the following properties:

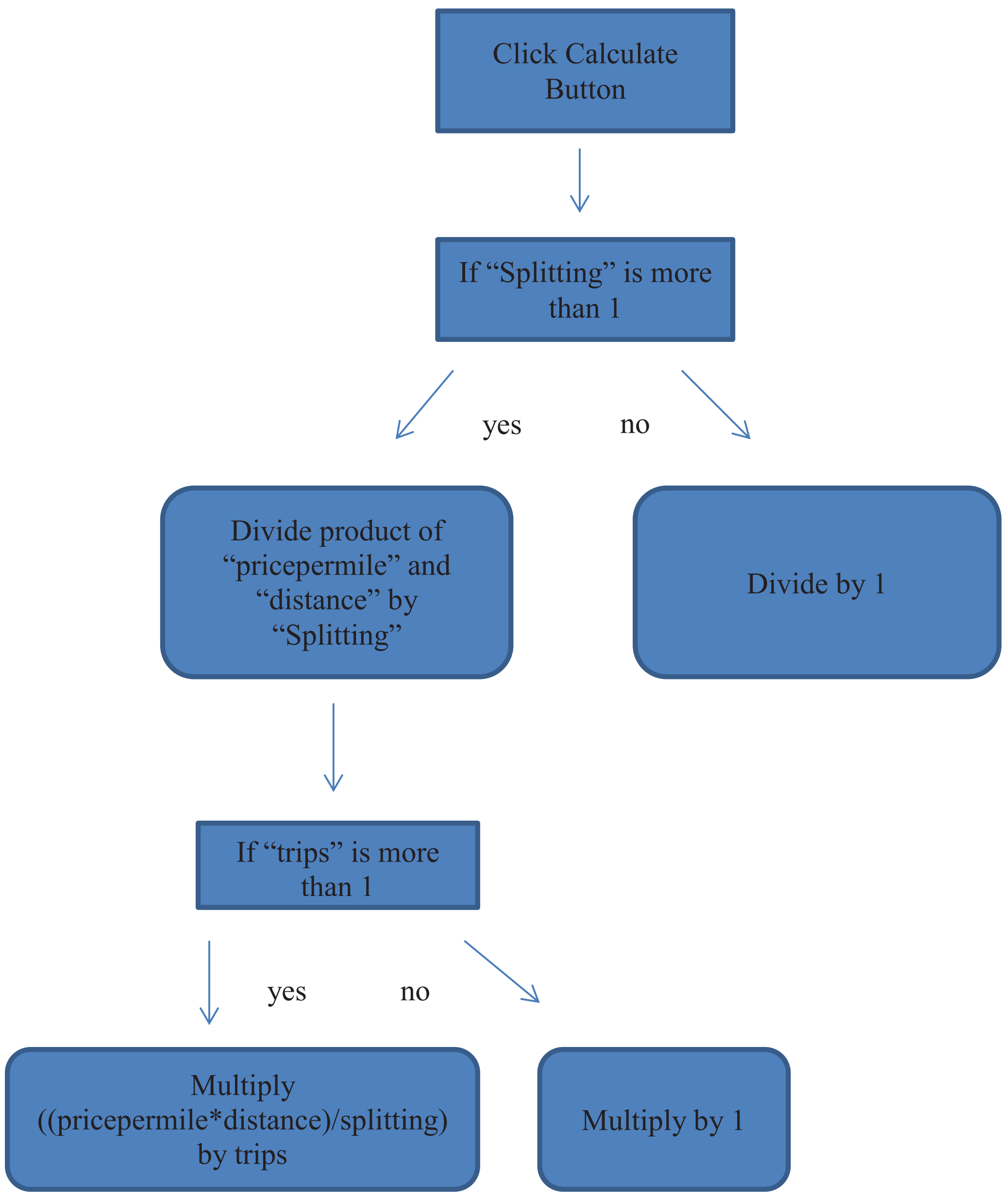
-Get and set methods for all entered data
-Array list for miles
-Add miles method to add to the miles array list
-Find total miles method to add the values in the array
-Calculate average method to find the average for that day

Actions

Getters and Setters - retrieves and sets all inputted data
ReadFromFile - retrieves data from text file
WriteToFile – writes data from UI into text file
AddMiles – adds miles to the array
FindTotalMiles – adds miles within the array
CalculateAverage – divides total miles by array size to find average


Click Enter
Brings up UI
User enters data and clicks Add
Miles are added to selected day array and average miles and price per gallon are calculated. Selected day, average miles, and price per gallon are displayed

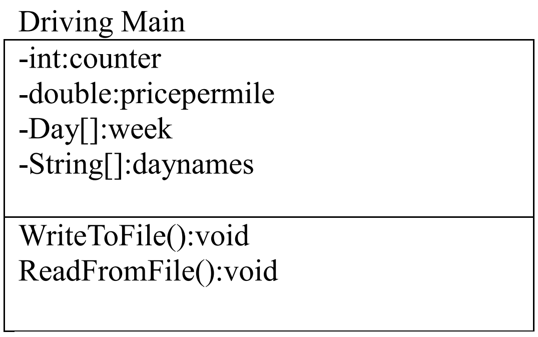

Click Previous Button
If there is data entered for prior spots in week array
yes
no
Display previous day
Hold
Click Next Button
If there is data entered for the next spot in week array
yes
no
Display next day
Hold


Click Calculate Button
If "Splitting" is more than 1
yes
no
Divide product of "pricepermile" and "distance" by "Splitting"
Divide by 1
If "trips" is more than 1
yes
no
Multiply ((pricepermile*distance)/splitting) by trips
Multiply by 1

Actions to test:

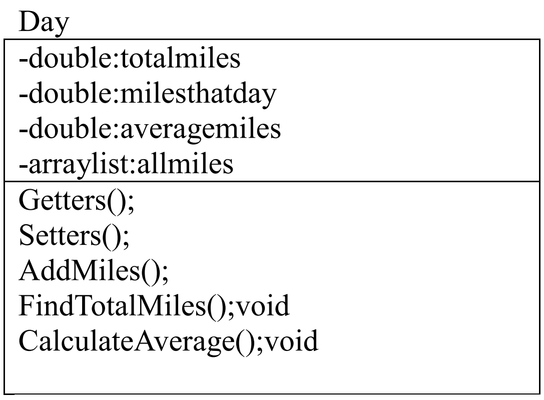
Action to Test	**Method of Testing - Result**
1st section averages miles that are on the same day of the week	Enter different mile values on same days of the week and the resulting value should be an average
1st section keeps day values and corresponding mile averages in correct order while also being able to navigate through said values	Days of the week should stay in order and navigate without blank returns
Clicking the enter button should successfully bring up the UI	Run the program and click the enter button
In 2nd section, response entered in trips textbox should successfully alter the total distance and eventually the price per rider	Enter different values in the Trips text box – the resulting price per riding should change accordingly
In 2nd section, add button should divide total distance by riders and update price per rider list box	Clicking the Add button should result in a price per rider that is the product of the Distance and the Trips, divided by number of riders

UML Diagrams


Driving Main

-int:counter
-double:pricepermile
-Day[]:week
-String[]:daynames

WriteToFile():void
ReadFromFile():void


Day

-double:totalmiles
-double:milesthatday
-double:averagemiles
-arraylist:allmiles

Getters();
Setters();
AddMiles();
FindTotalMiles();void
CalculateAverage();void