**Object oriented Programming report**

Contents

[Class Diagram 2](#_Toc483006962)

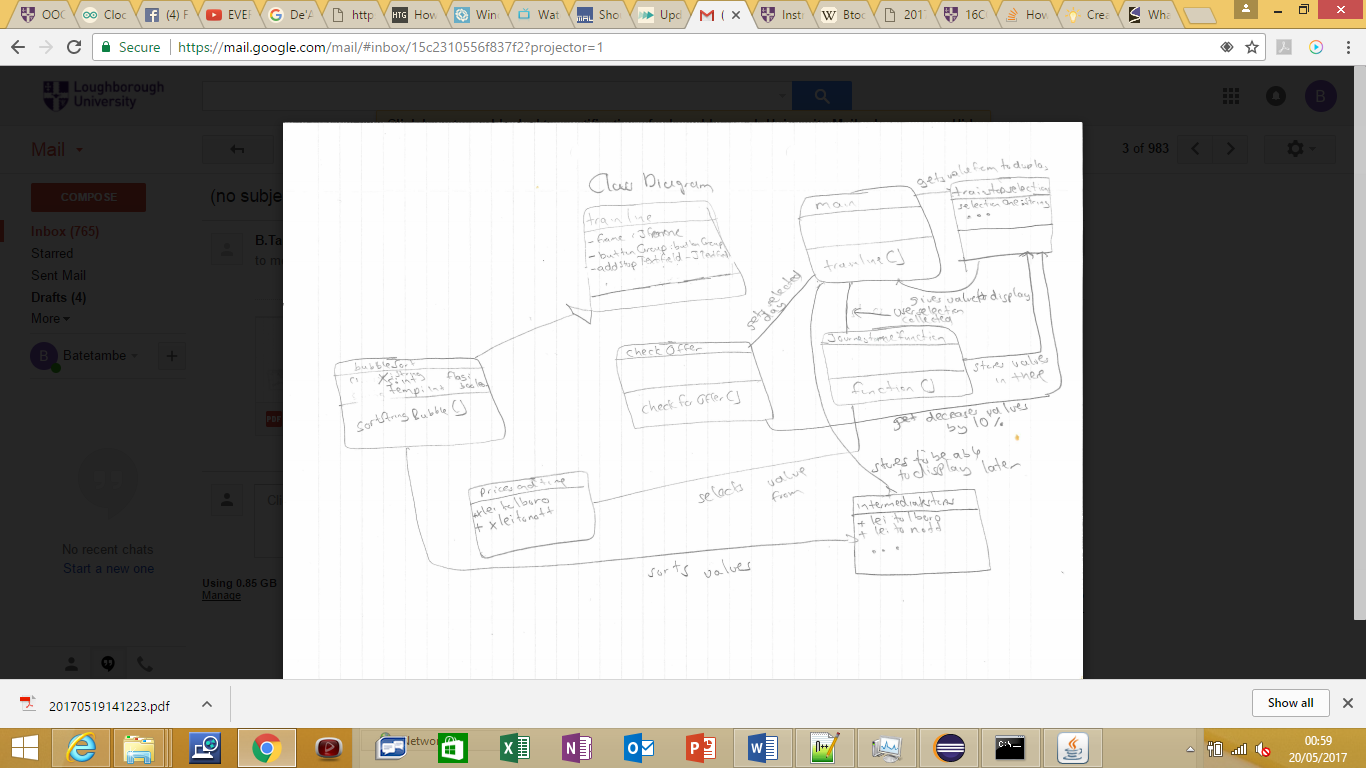
[Design 3](#_Toc483006963)

[User Guide 5](#_Toc483006964)

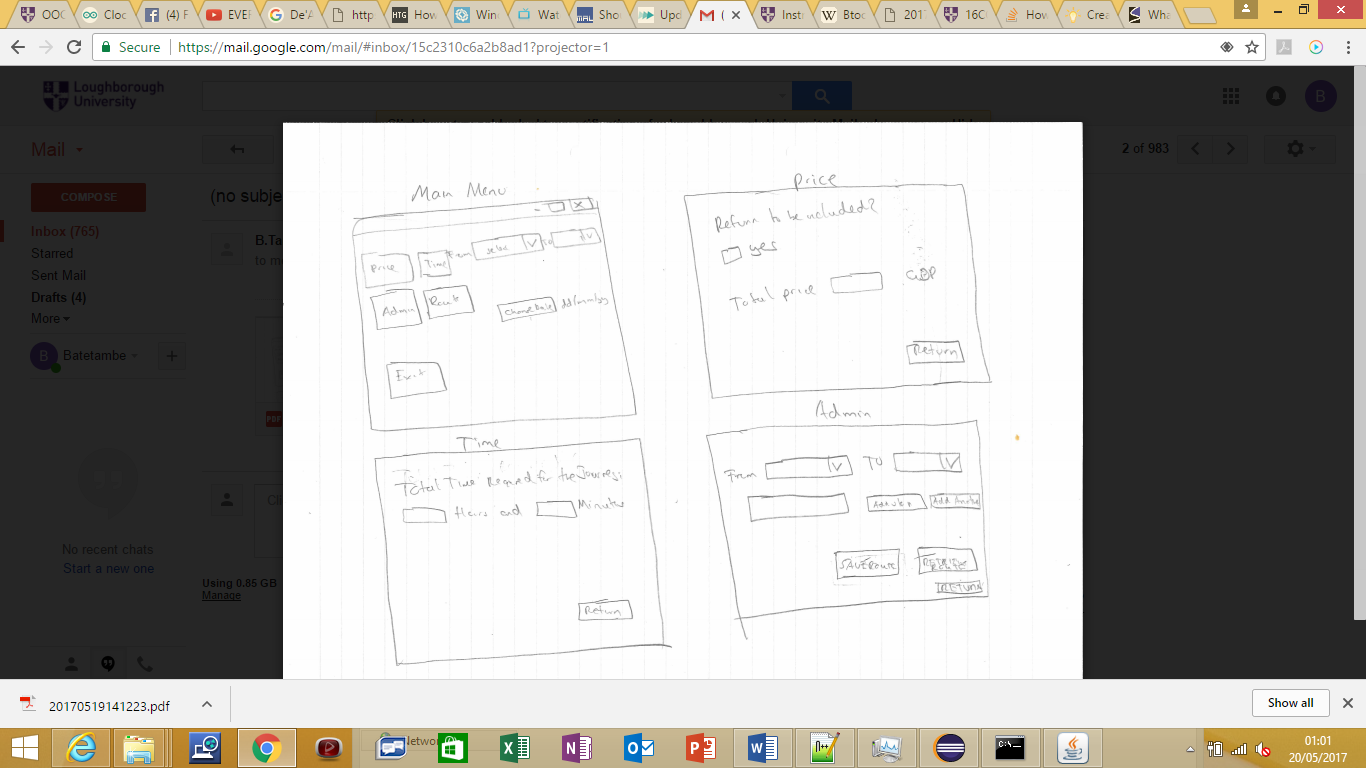
[Admin User Guide 7](#_Toc483006965)

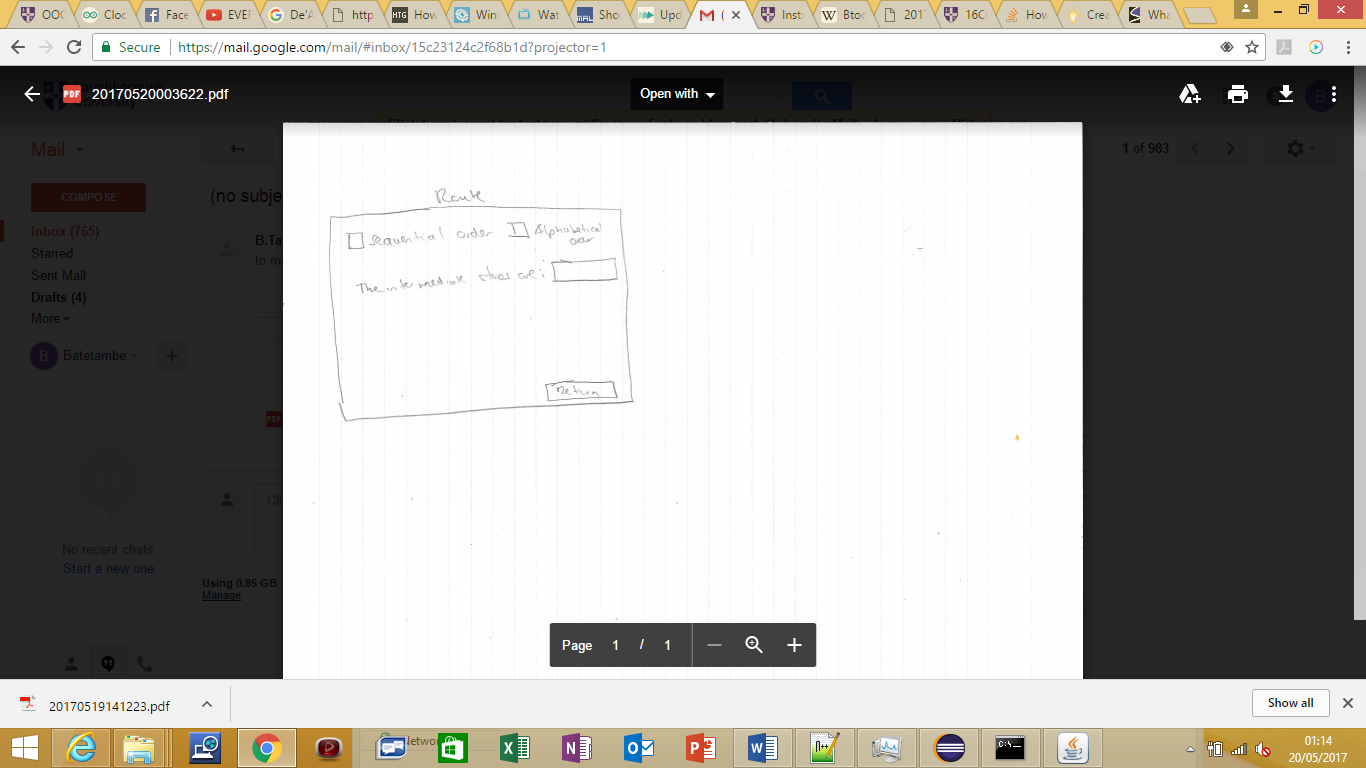
[Testing 8](#_Toc483006966)

# Class Diagram



# Design





The reason I chose to have a design where the price, time and route will be displayed on a different panel is to have the application navigate like a website which is a very common design with most users. I decided to remove the Input route option and instead give the user the option to choose the Route and add intermediate stops on the same panel to keep the solution simple. I also removed the SORT button and implemented the ability for the user to sort the routes on the Route panel. This also increases the simplicity of the solution.

# User Guide

**Running the trainline.java file**

Run Command Prompt

C:> cd \oopcoursework

This makes C:\mywork the current directory.

C:\oopcoursework> dir

This displays the directory contents. You should see filenamehere.java among the files.

C:\oopcoursework> set path=%path%;C:\Program Files\Java\jdk1.5.0\_09\bin

This tells the system where to find JDK programs.

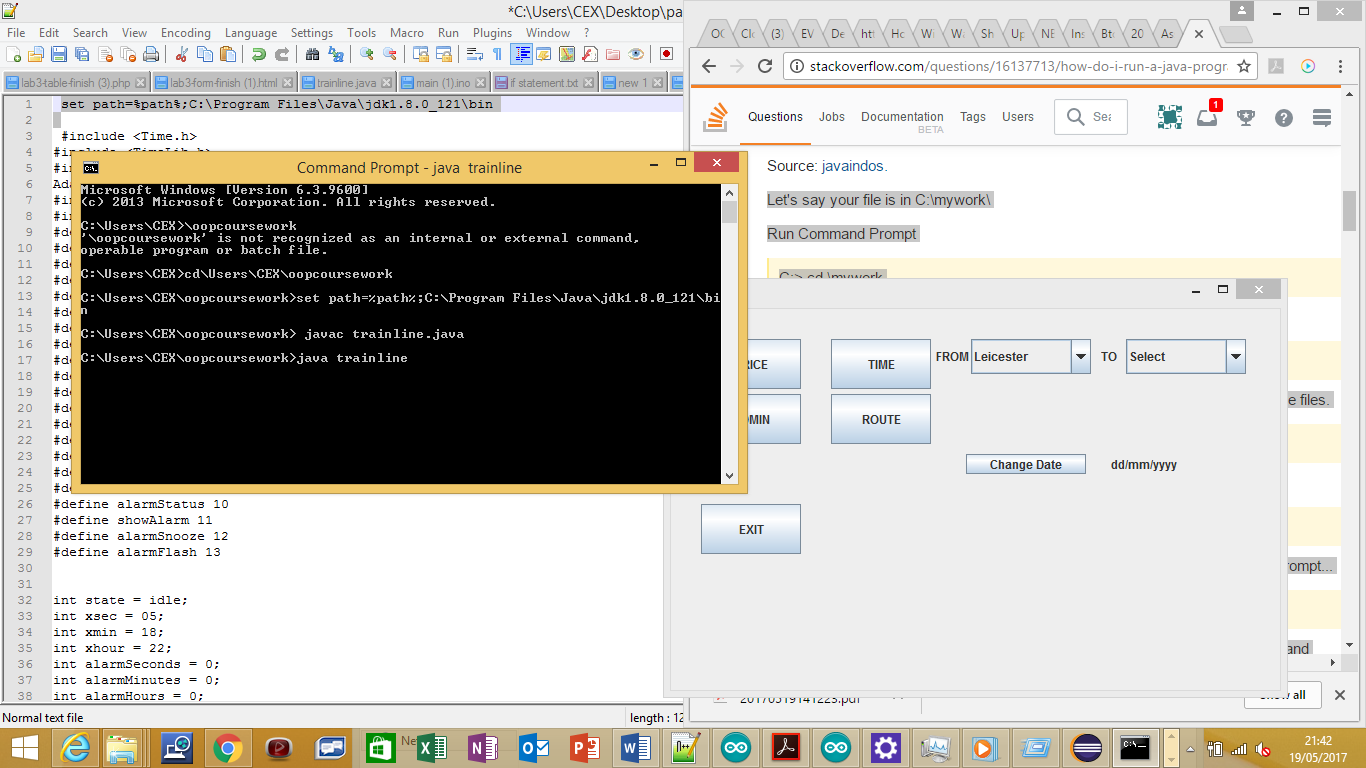
C:\oopcoursework> javac triainline.java

This runs javac.exe, the compiler. You should see nothing but the next system prompt...

C:\mywork> dir

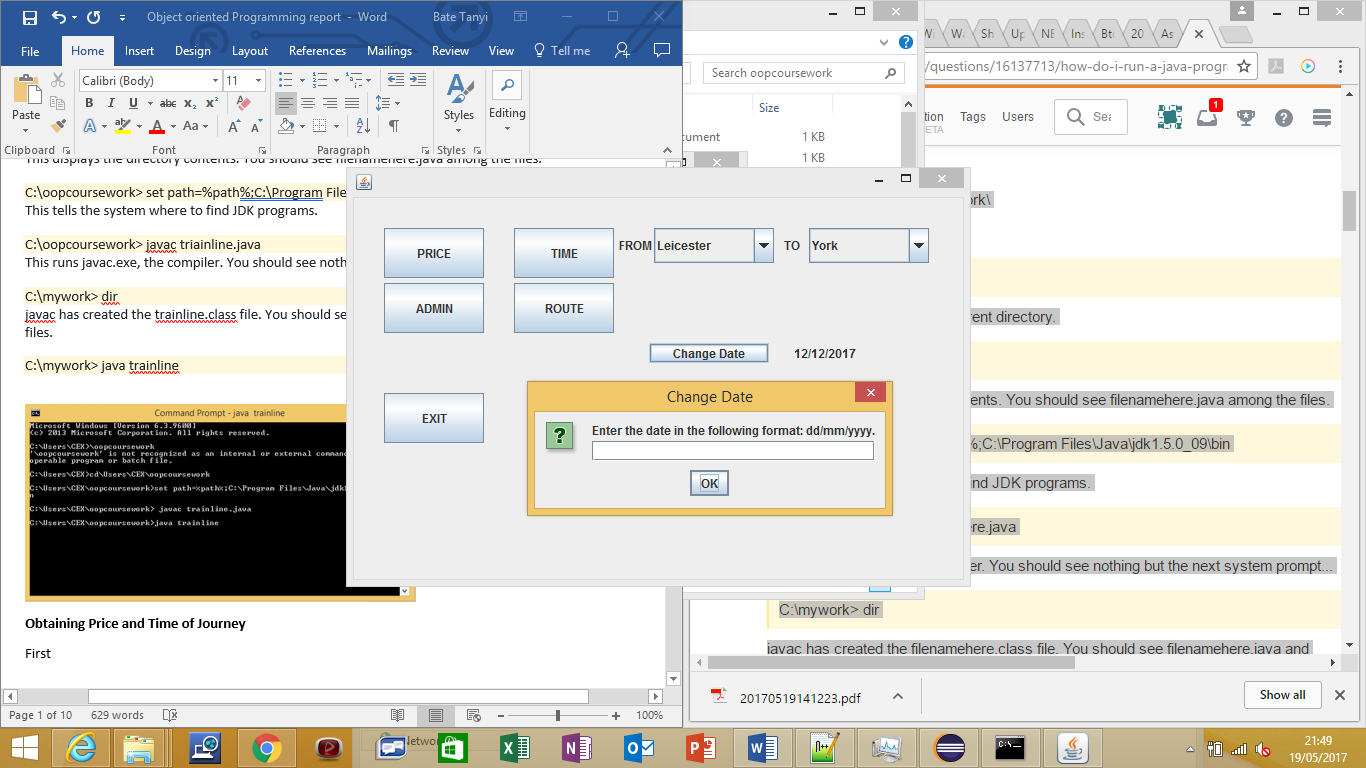
javac has created the trainline.class file. You should see trainline.java and trainline.class among the files.

C:\mywork> java trainline

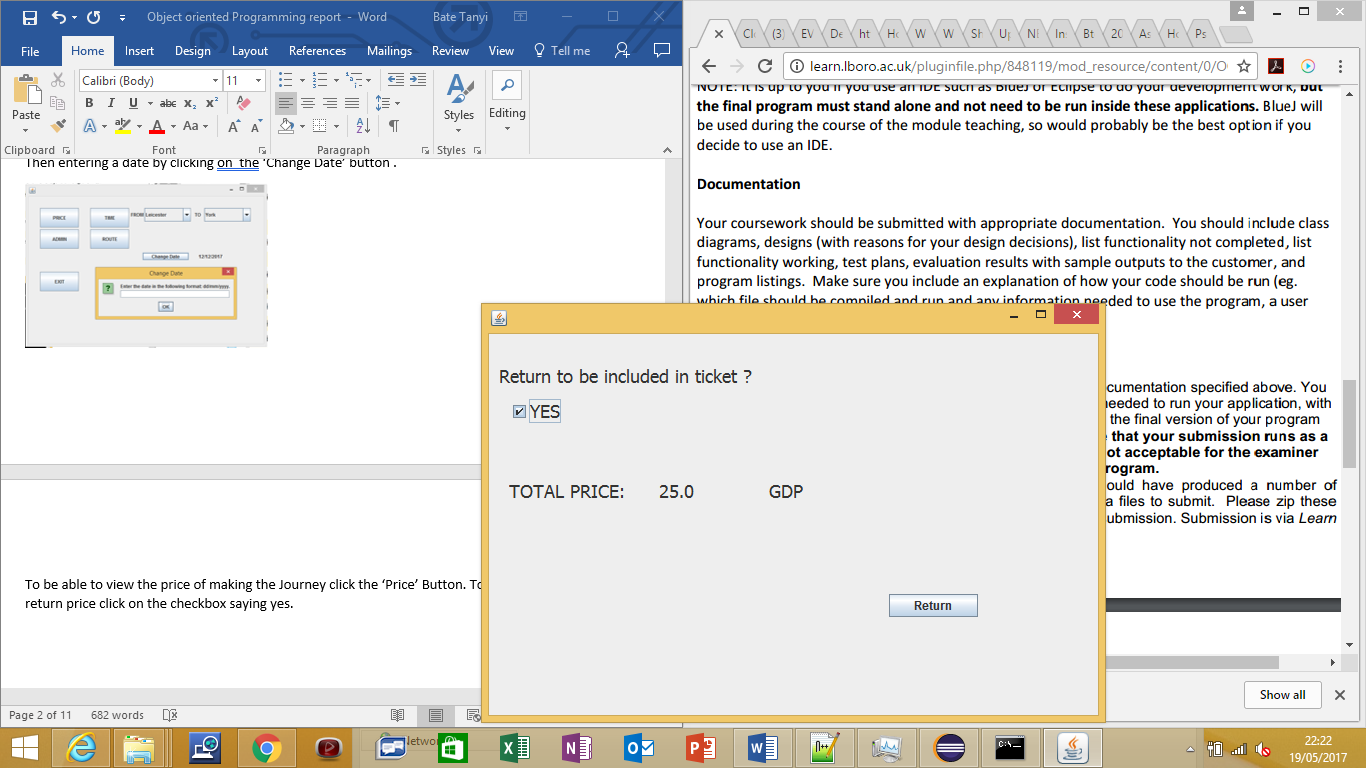


**Obtaining Price and Time of Journey and intermediate Stops**

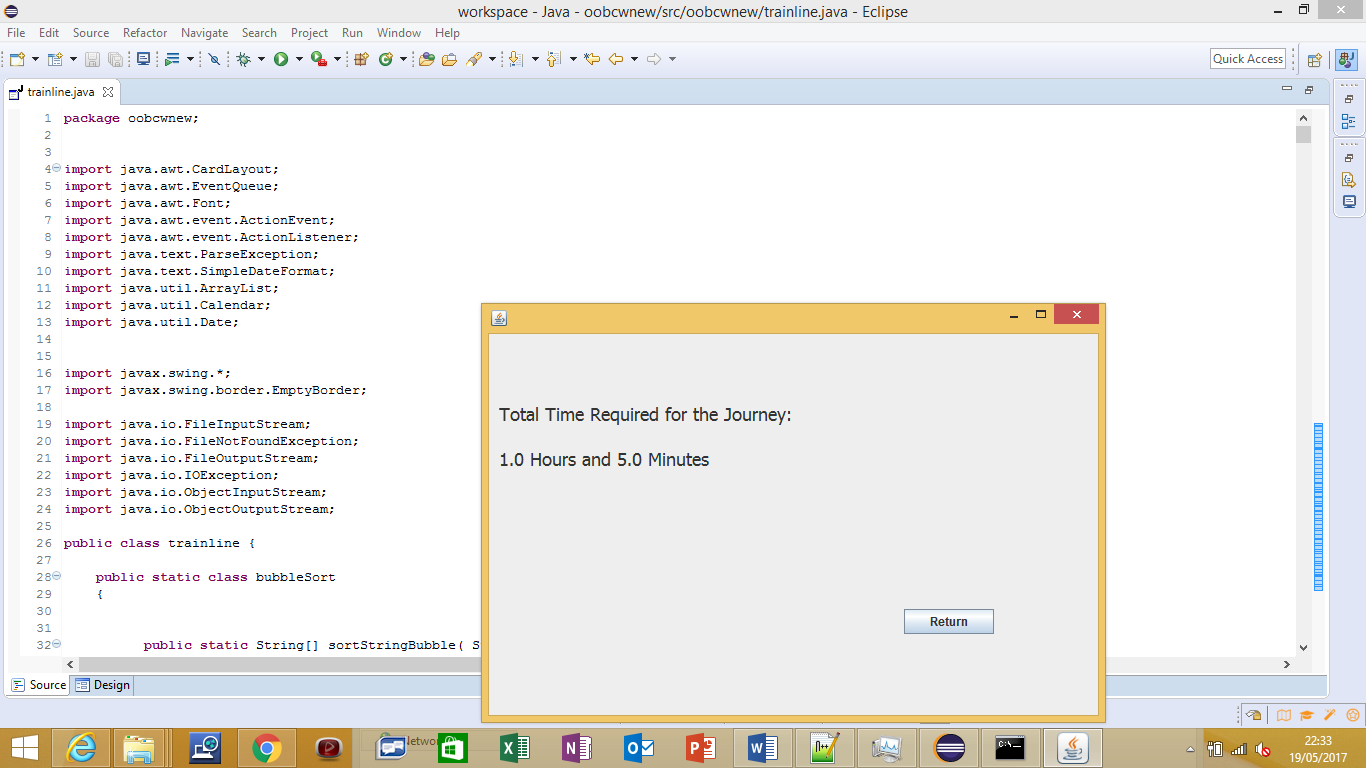
The Journey can be selected by selecting two different train stations using the drop down menu. Then entering a date by clicking on the ‘Change Date’ button .



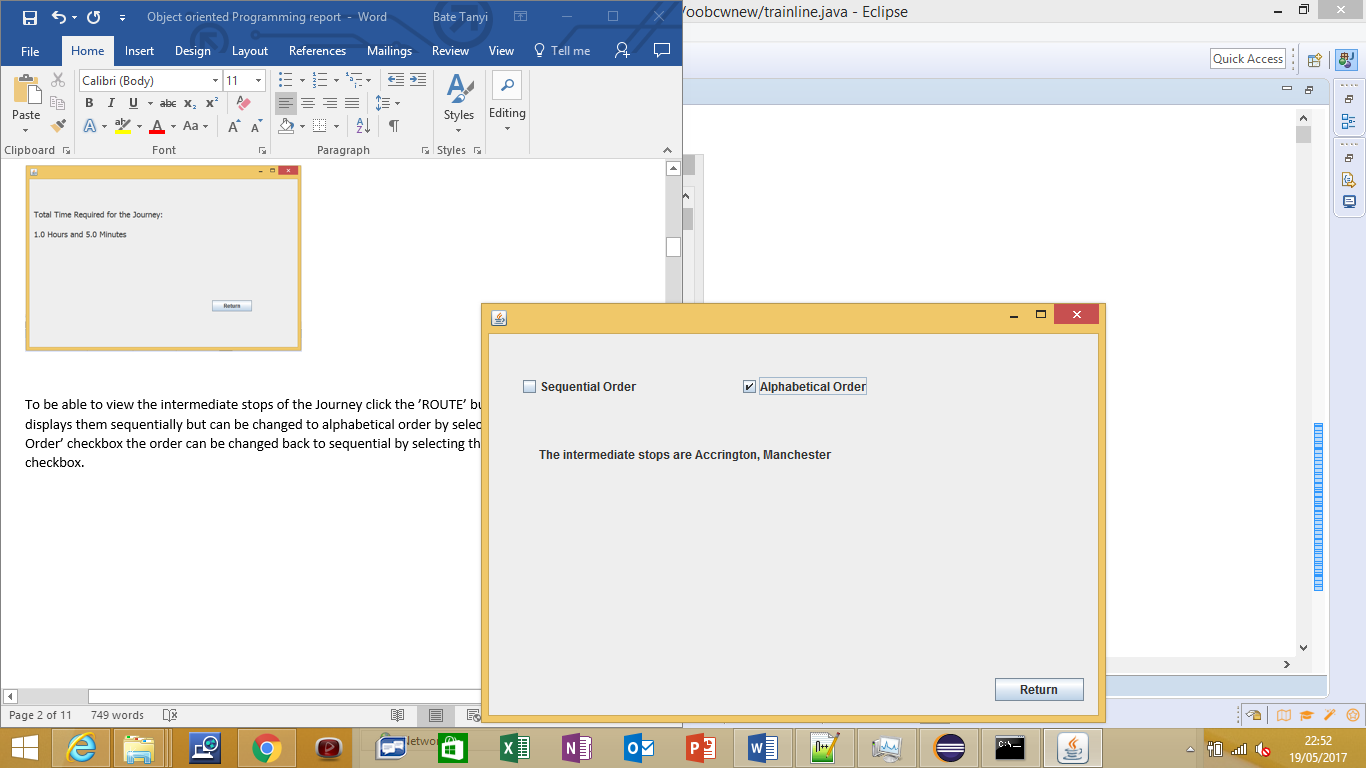
To be able to view the price of making the Journey click the ‘Price’ Button. To be able to see the return price click on the checkbox saying yes.



To be able to view the time required for the Journey click on the ‘TIME’ button.



To be able to view the intermediate stops of the Journey click the ’ROUTE’ button. It automatically displays them sequentially but can be changed to alphabetical order by selecting the ‘Alphabetical Order’ checkbox the order can be changed back to sequential by selecting the ‘Sequential Order’ checkbox.

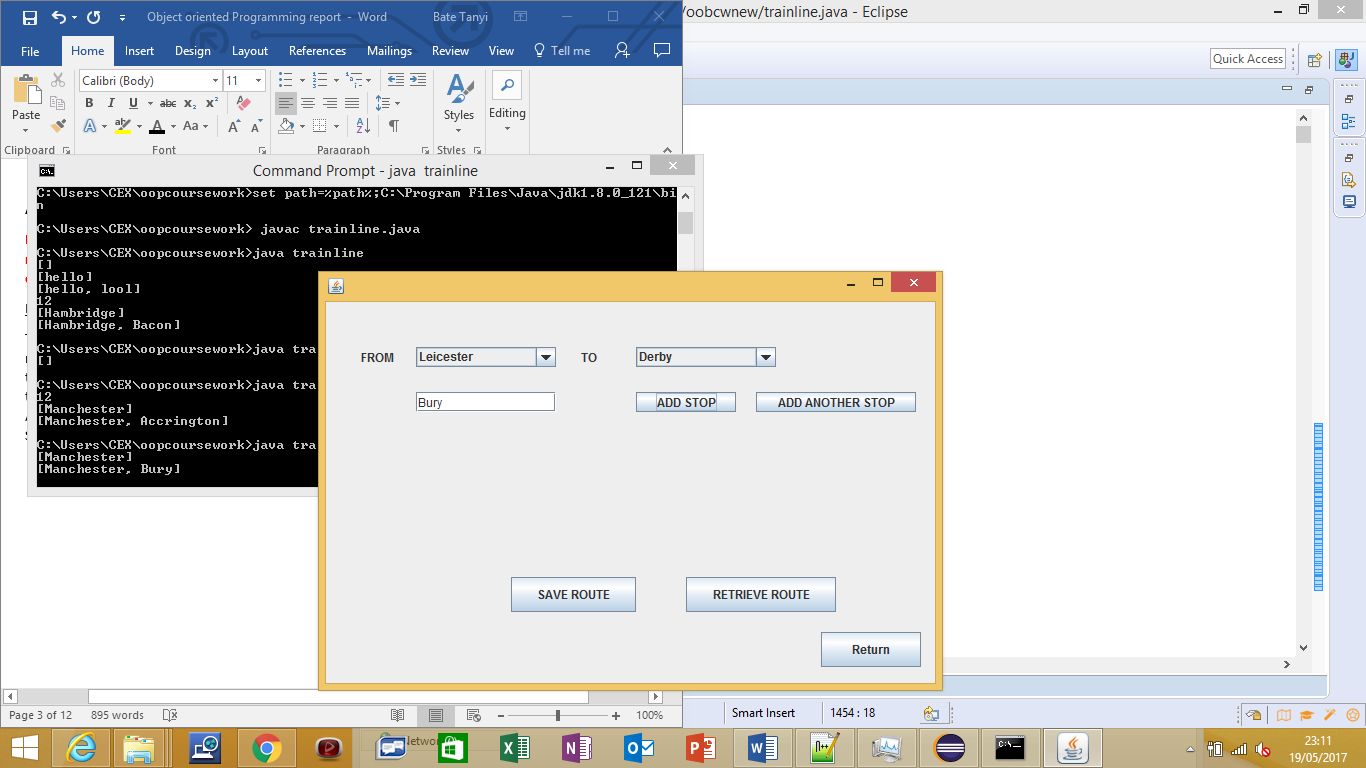


## Admin User Guide

**Important notice: Always when opening program always retrieve route before saving as saving the route before importing the previously saved route will overwrite the previously saved route completely. This will cause the previous intermediate stops to be deleted.**

How to add intermediate stops

To add intermediate stops first access the Admin panel by clicking the ‘ADMIN’ button on the main menu. Then select two different train stations to add intermediate stops. An intermediate stop can then added to that Route by typing the name of the intermediate stop into the textbox below it and then clicking the ‘ADD STOP’ button. To add another stop simply clear the textbox by clicking ‘ADD ANOTHER STOP’. To now add the new stop enter the stop into textbox and again click the ‘ADD STOP’ button. The current array of stops will automatically appear on the command line.



How to a save a Route

Once you have added all the immediate stops of the ROUTE simply click the ‘SAVE ROUTE’ button

How to retrieve a Route

To retrieve all routes so it will display to the user simply click the ‘RETRIEVE ROUTE’ button.

How to clear a Route

To clear a certain Route you will have to manually overwrite that Route. This can be done by closing the program and opening it again but this time do not retrieve the Routes. Select the starting and last train station of the route using the drop down menus and then save it. This will manually overwrite them. THEN RETRIEVE THE ROUTES TO PREVENT ANY OTHER ROUTES FROM BEING OVERWRITTEN.

# Testing

**Functions Completed (all were completed)**

Display journey time between any two stations in hours and minutes

Display the cost of travel (one-way or return) between any two stations

Dealing with the reduced cost on last day of any month

Administrator add and view intermediate stops on route (live program)

Save all routes to a file

Implement alphabetical sorting function for routes

Implement GUI

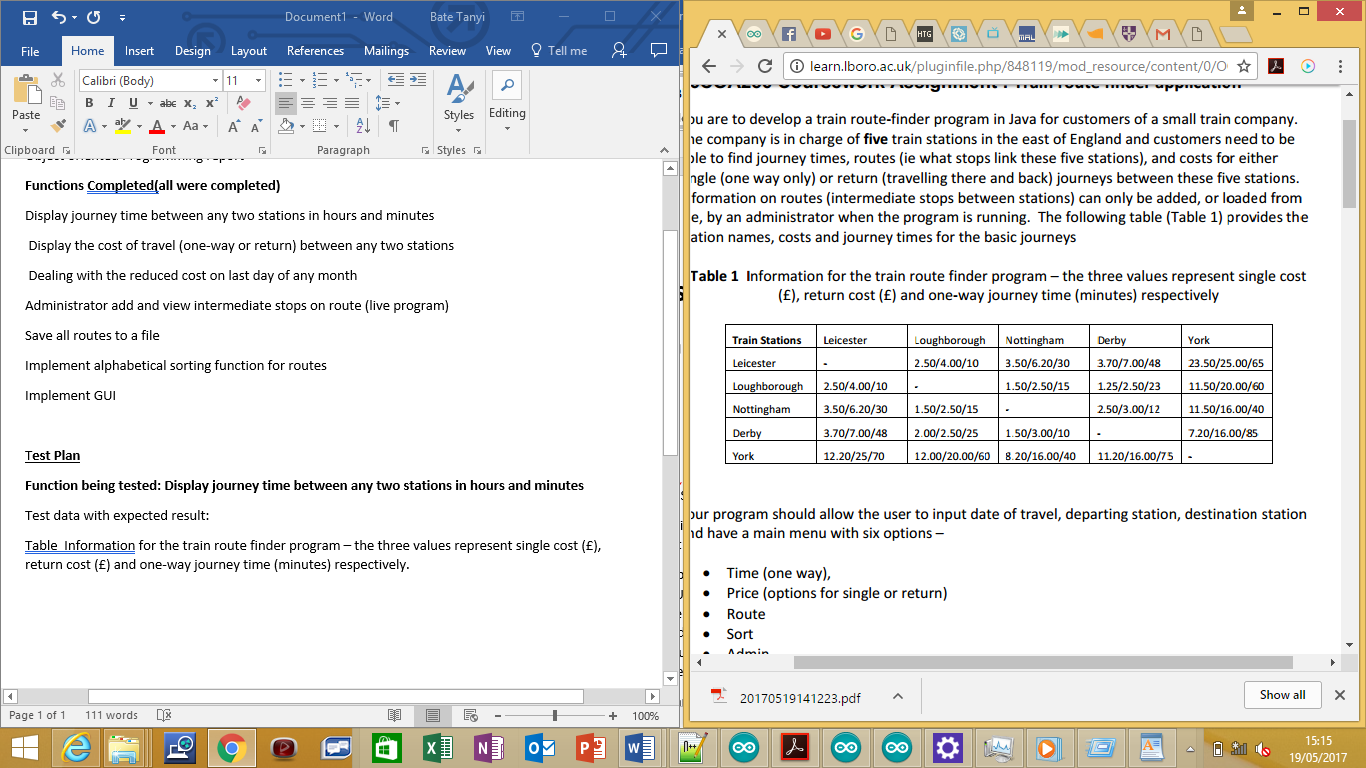
Error Trapping

T**est Plan**

**Function being tested: Display journey time between any two stations in hours and minutes**

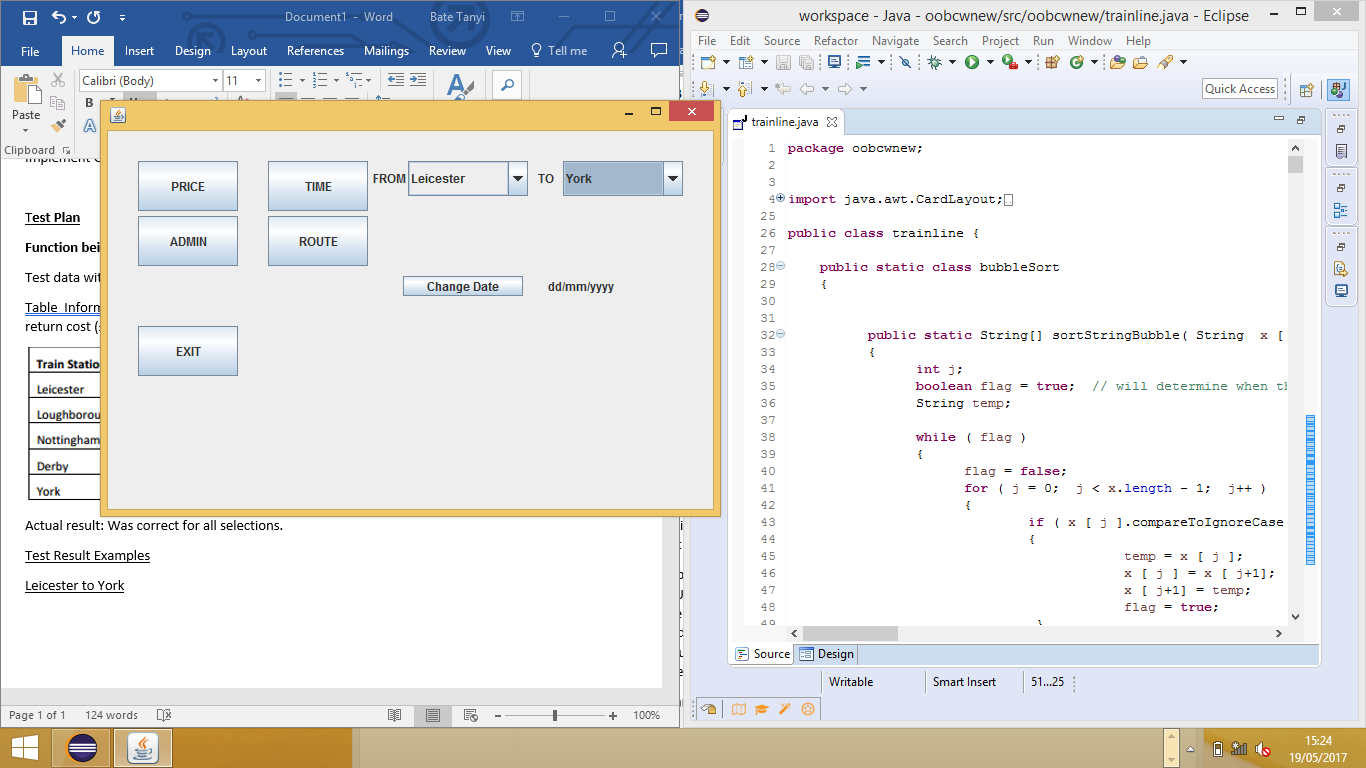
Test data with expected result:

Table Information for the train route finder program – the three values represent single cost (£), return cost (£) and one-way journey time (minutes) respectively.

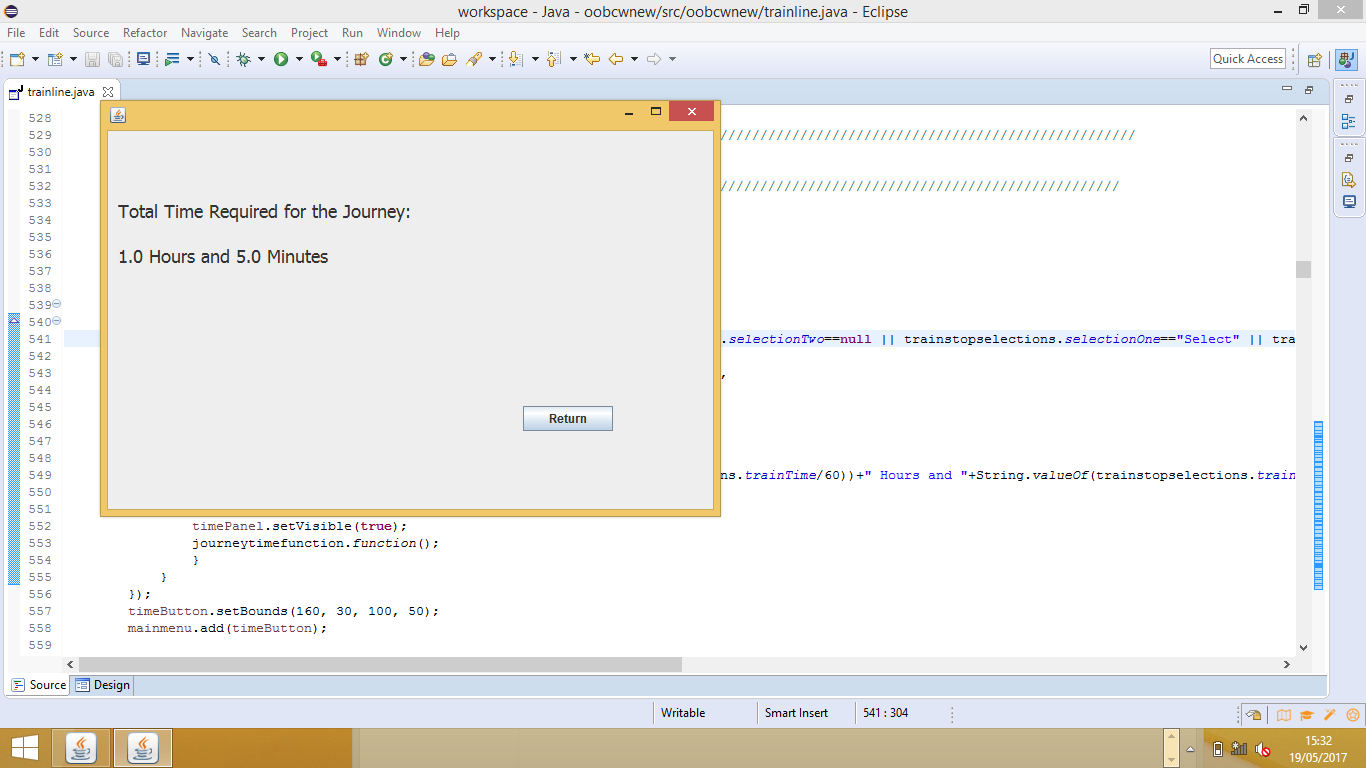


Actual result: Was correct for all selections.

Test Result Examples

Leicester to York

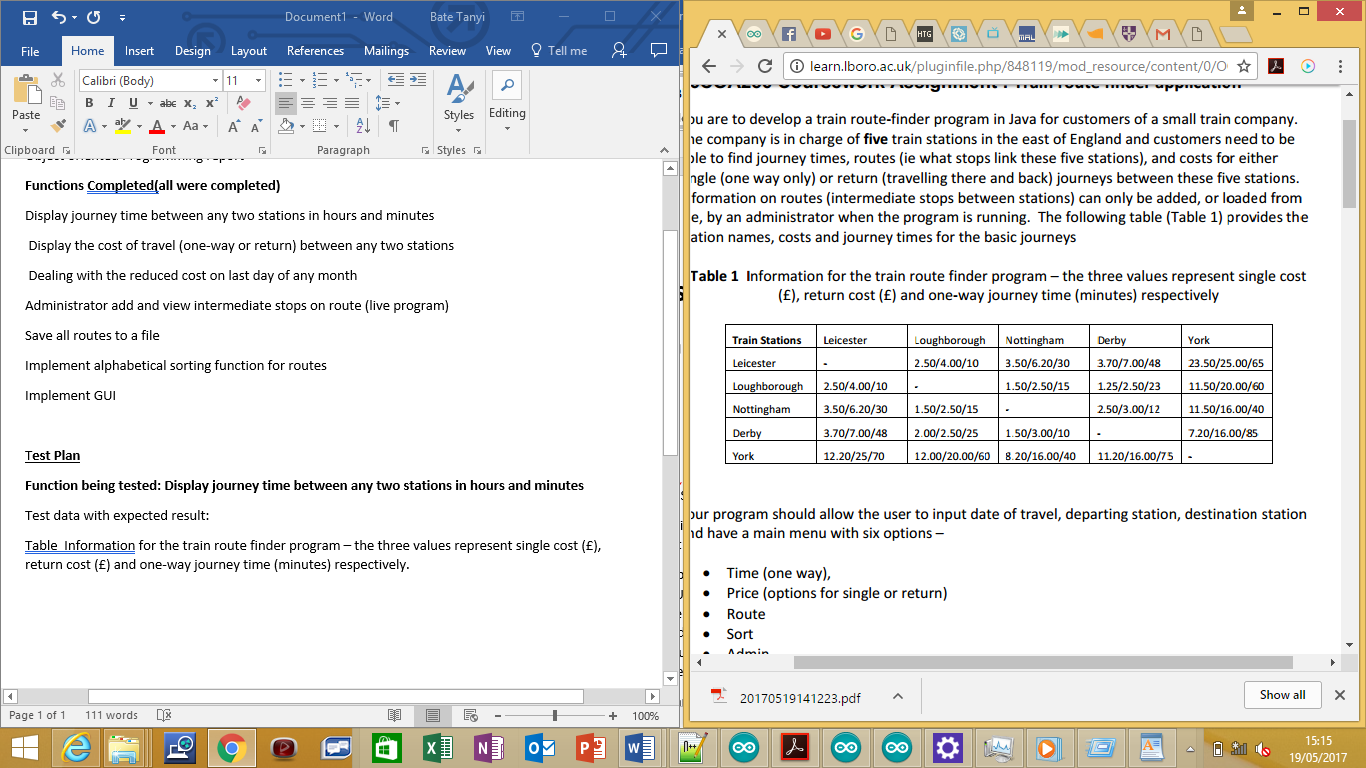
Result



**Function being tested:** Display the cost of travel (one-way or return) between any two stations

Test data with expected result:

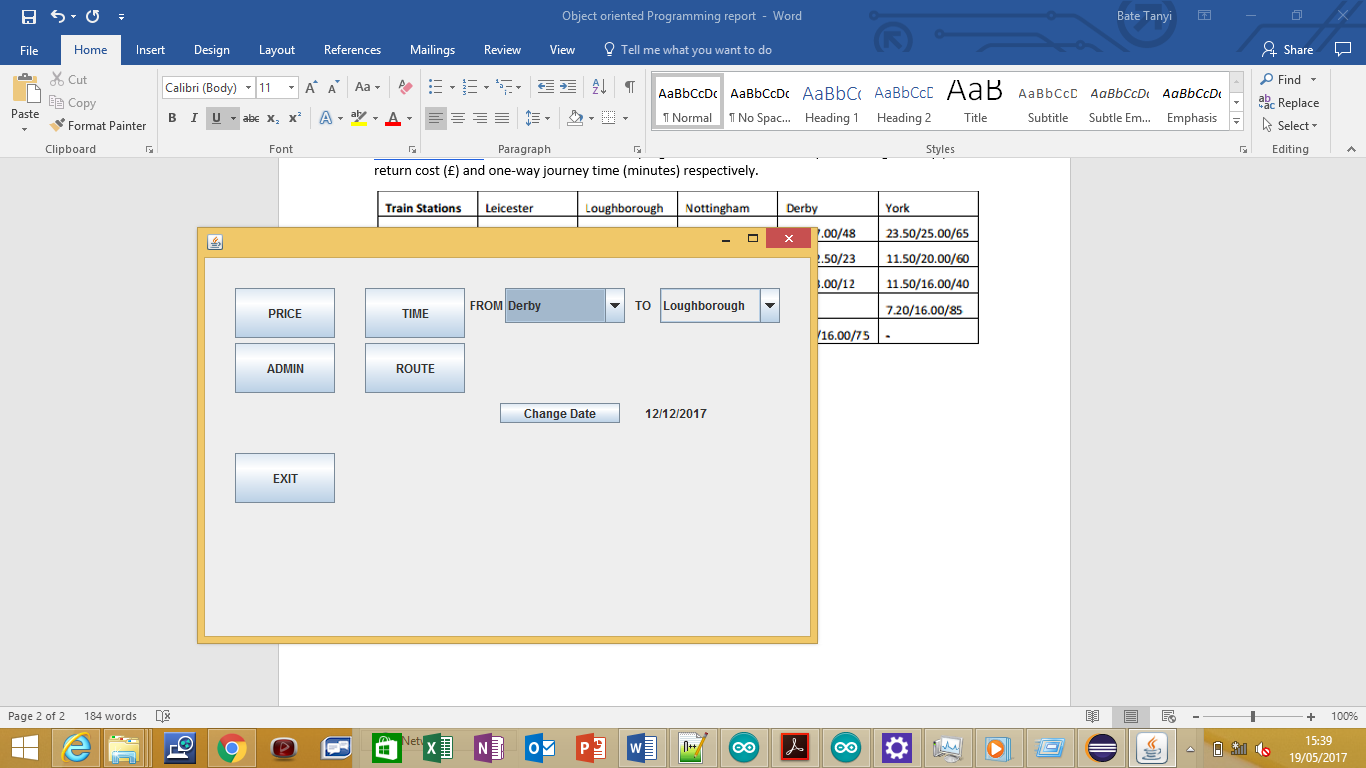
Table Information for the train route finder program – the three values represent single cost (£), return cost (£) and one-way journey time (minutes) respectively.



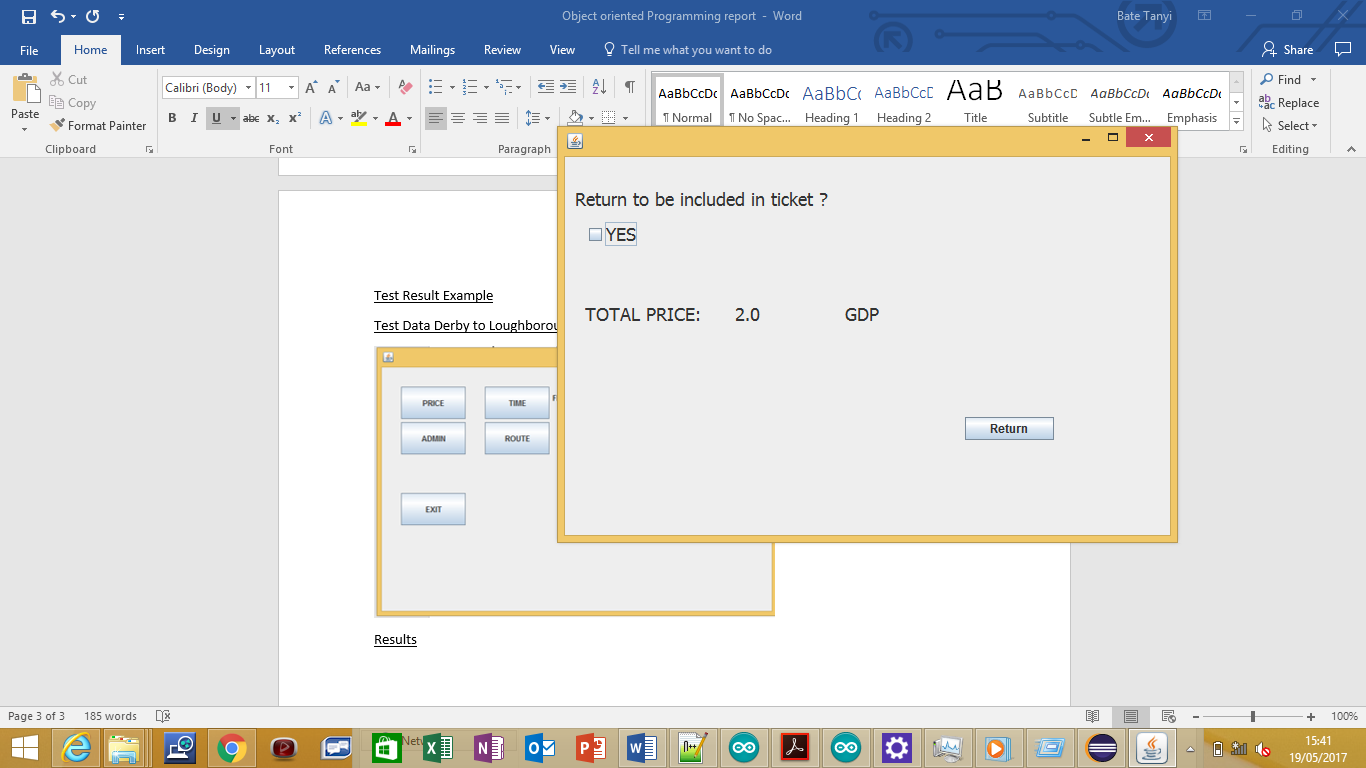
Actual result: Was correct for all selections

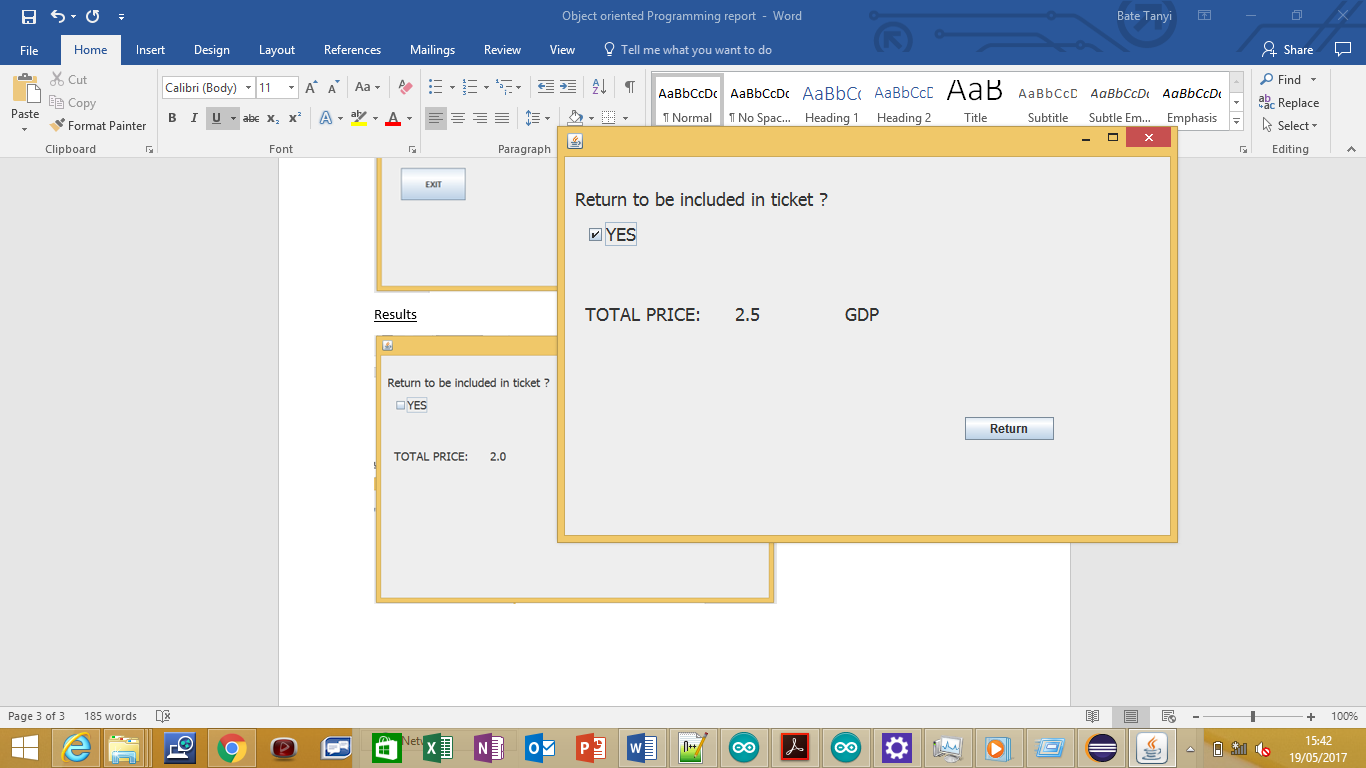
Test Result Example

Test Data Derby to Loughborough



Results





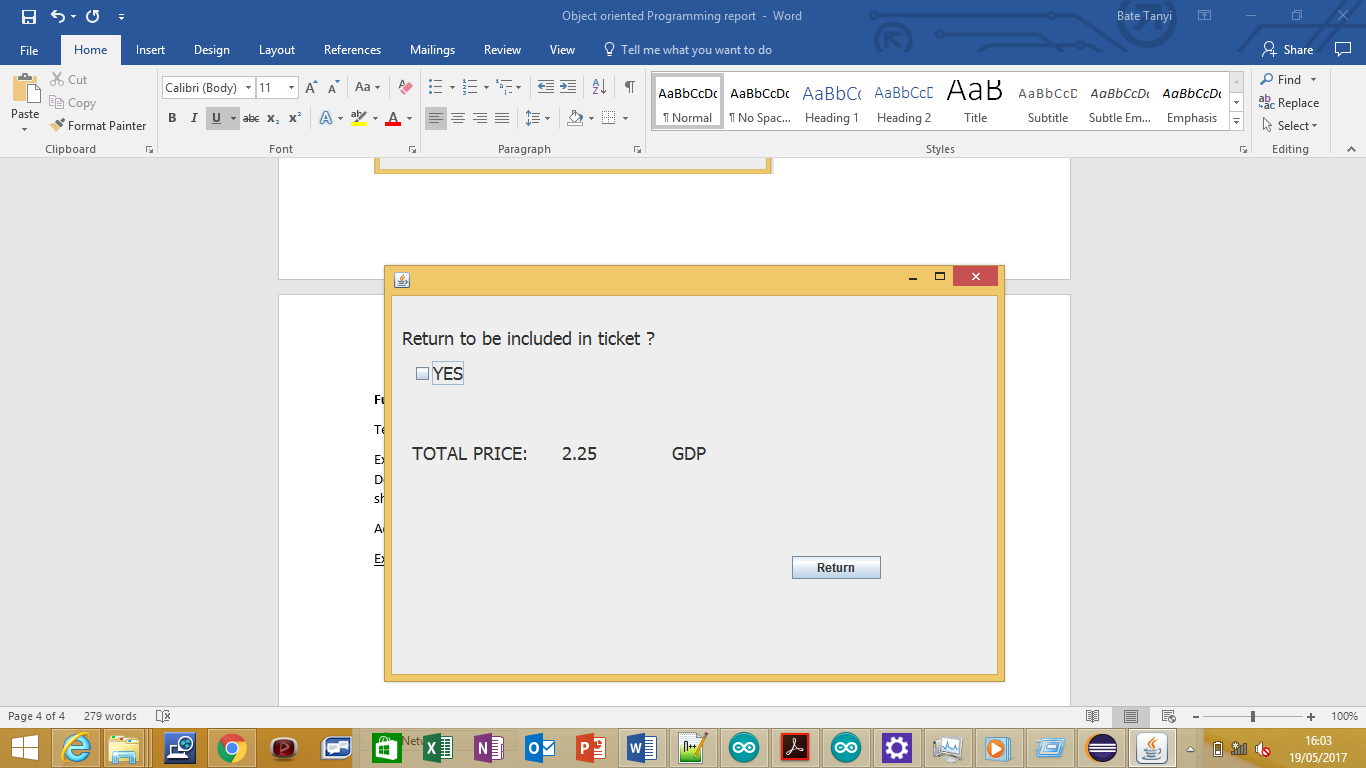
**Function being tested:** Dealing with the reduced cost on last day of any month

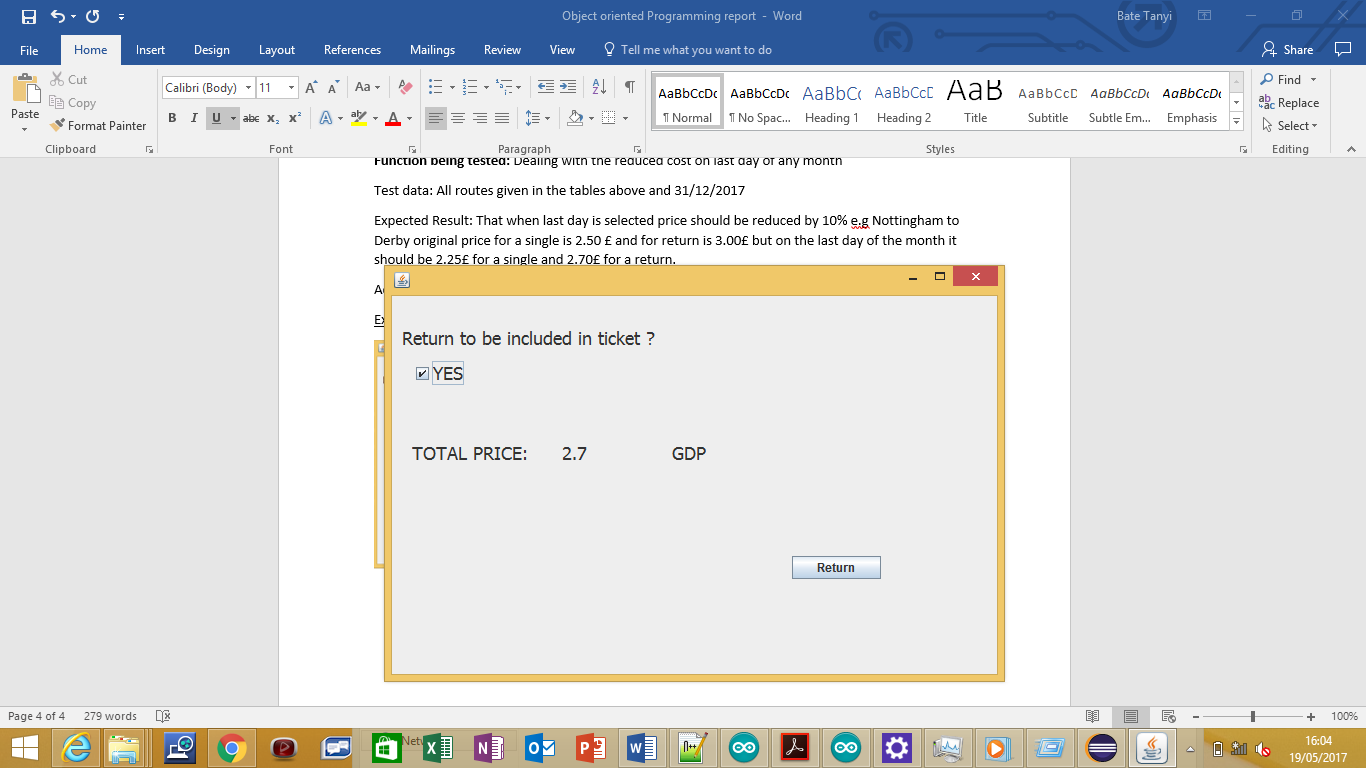
Test data: All routes given in the tables above and 31/12/2017

Expected Result: That when last day is selected price should be reduced by 10% e.g Nottingham to Derby original price for a single is 2.50 £ and for return is 3.00£ but on the last day of the month it should be 2.25£ for a single and 2.70£ for a return.

Actual Result: Single was 2.25 £ and 2.70£ for a return ticket.

Example Results using Nottingham to Derby

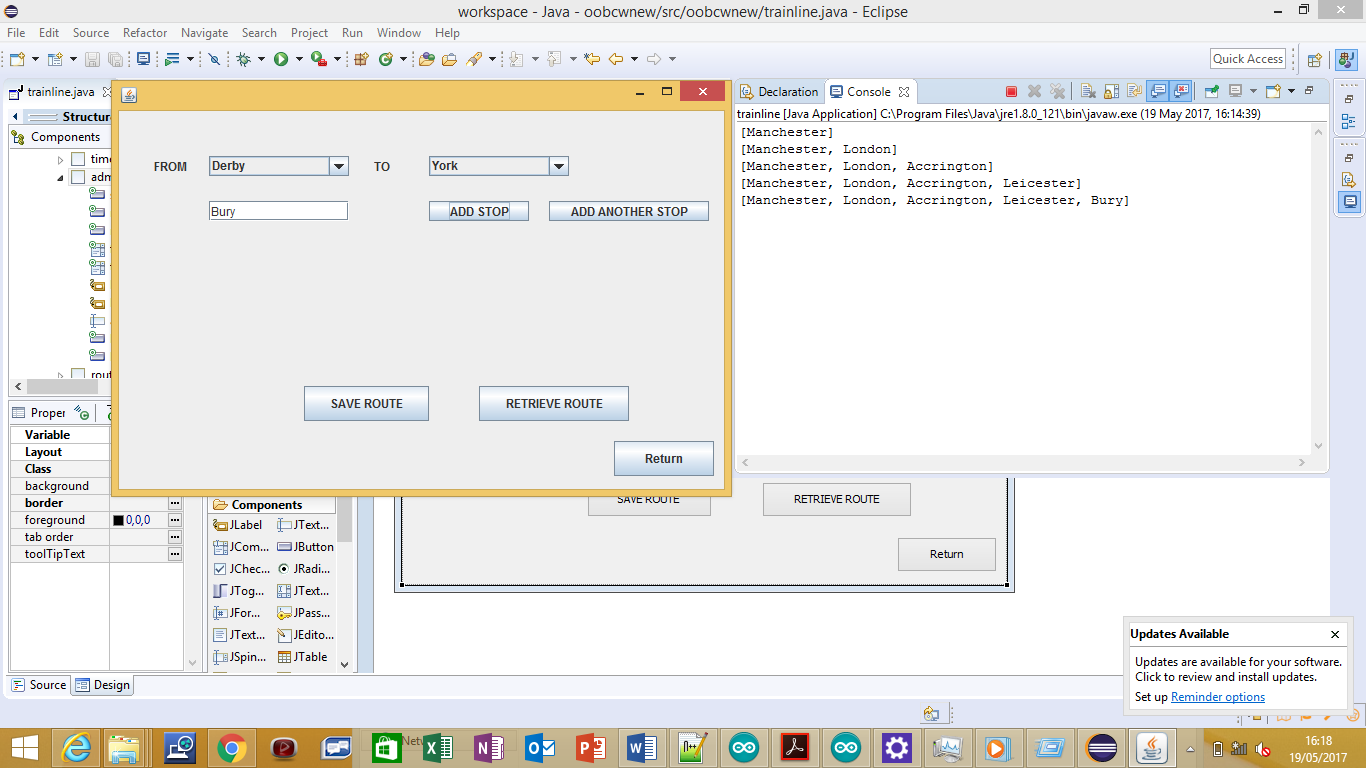


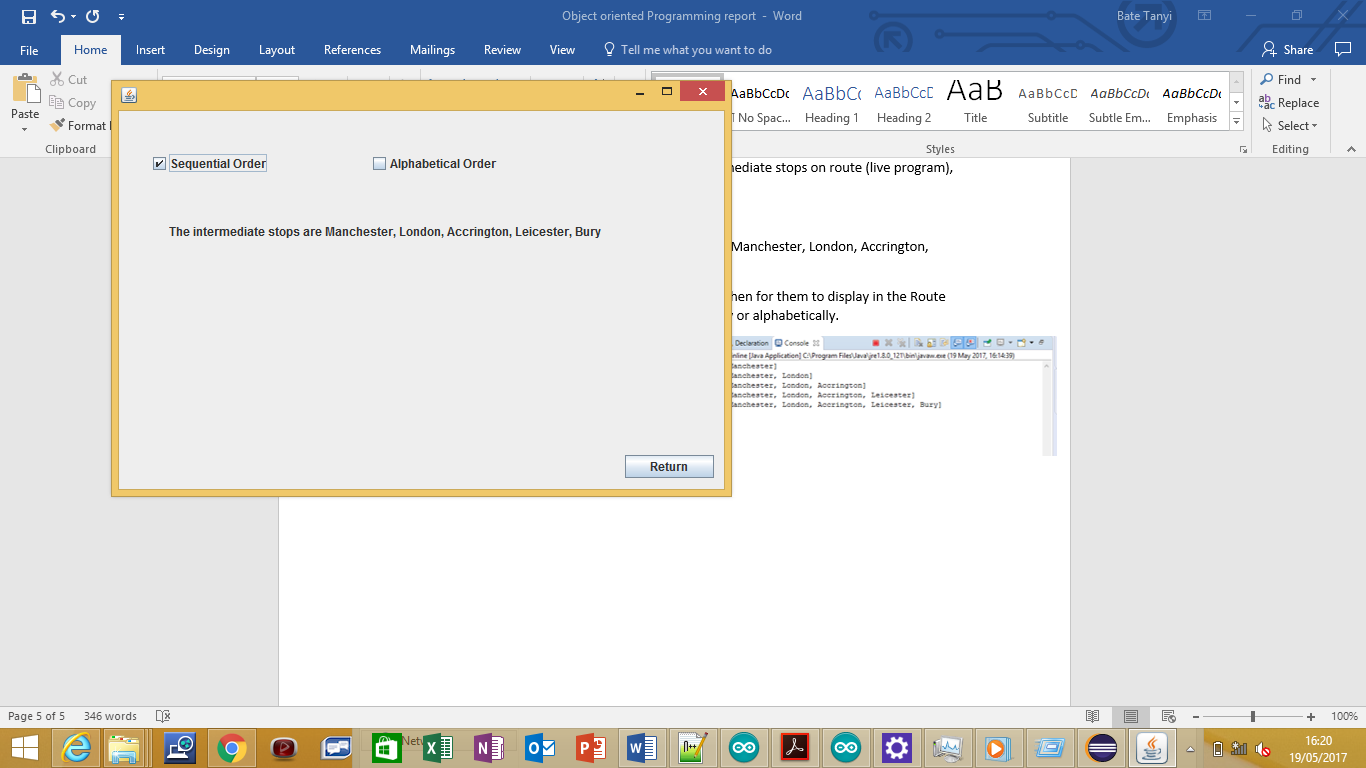


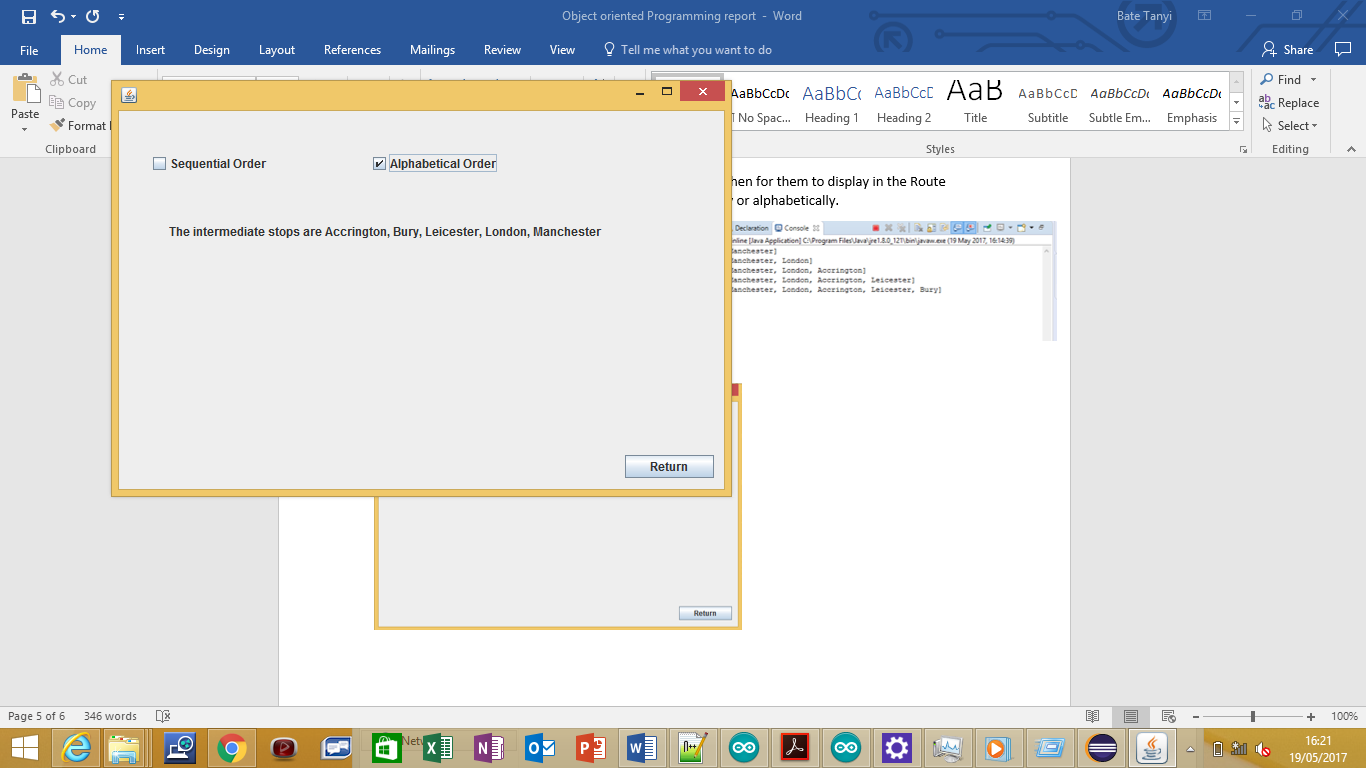
**Function/s being tested:** Administrator add and view intermediate stops on route (live program), Implement alphabetical sorting function for routes

Test Data: For Route Derby to York add intermediate stops: Manchester, London, Accrington, Leicester, Bury

Expected Result: To be able to add intermediate stops and then for them to display in the Route panel of the program. The route can be viewed sequentially or alphabetically.





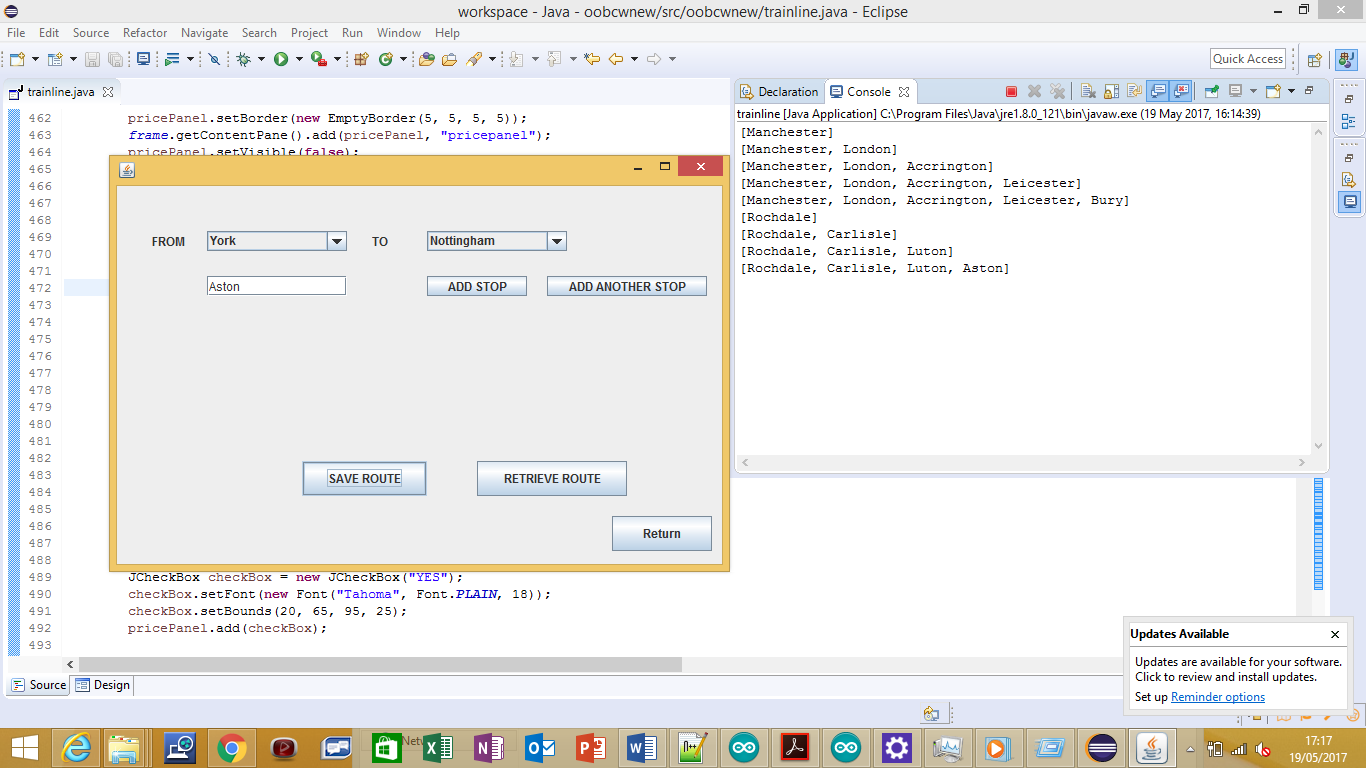


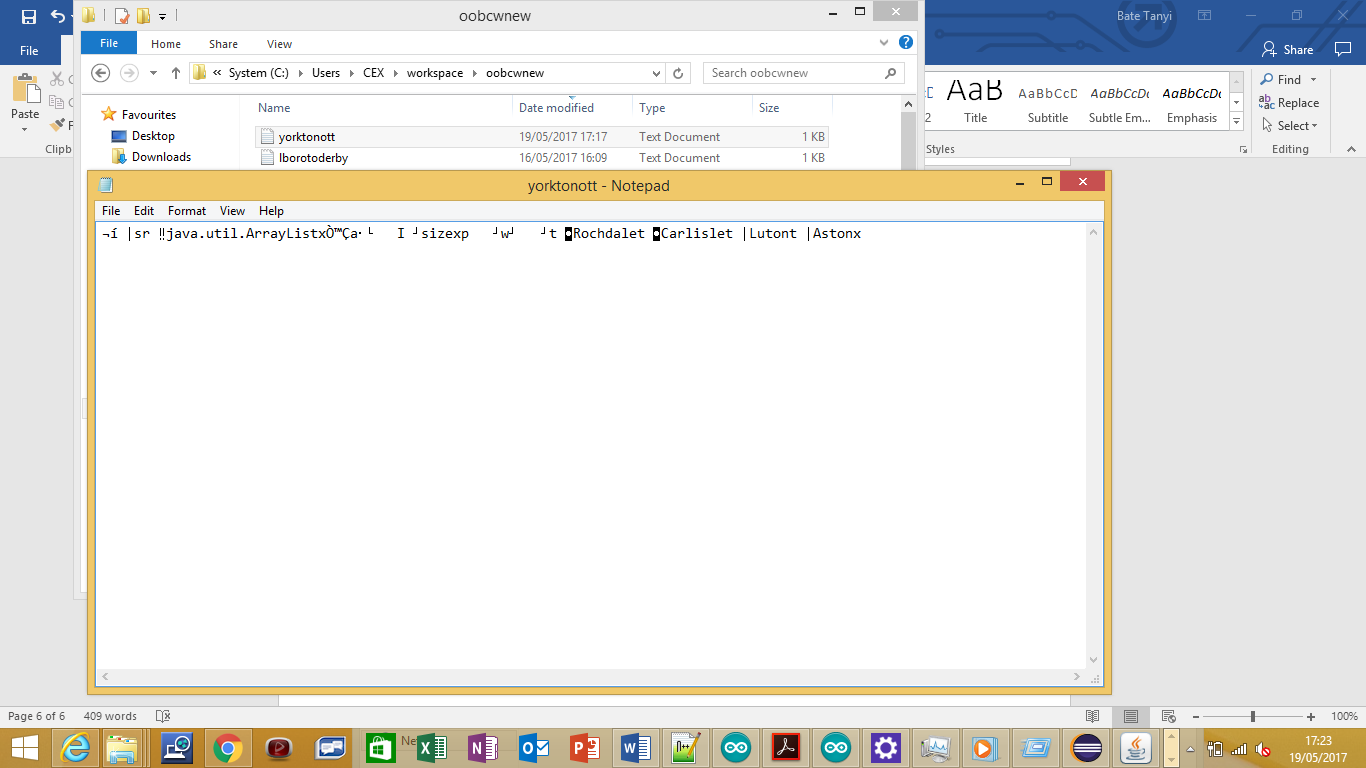
**Function/s being tested:** Save all Routes to a file and retrieve them

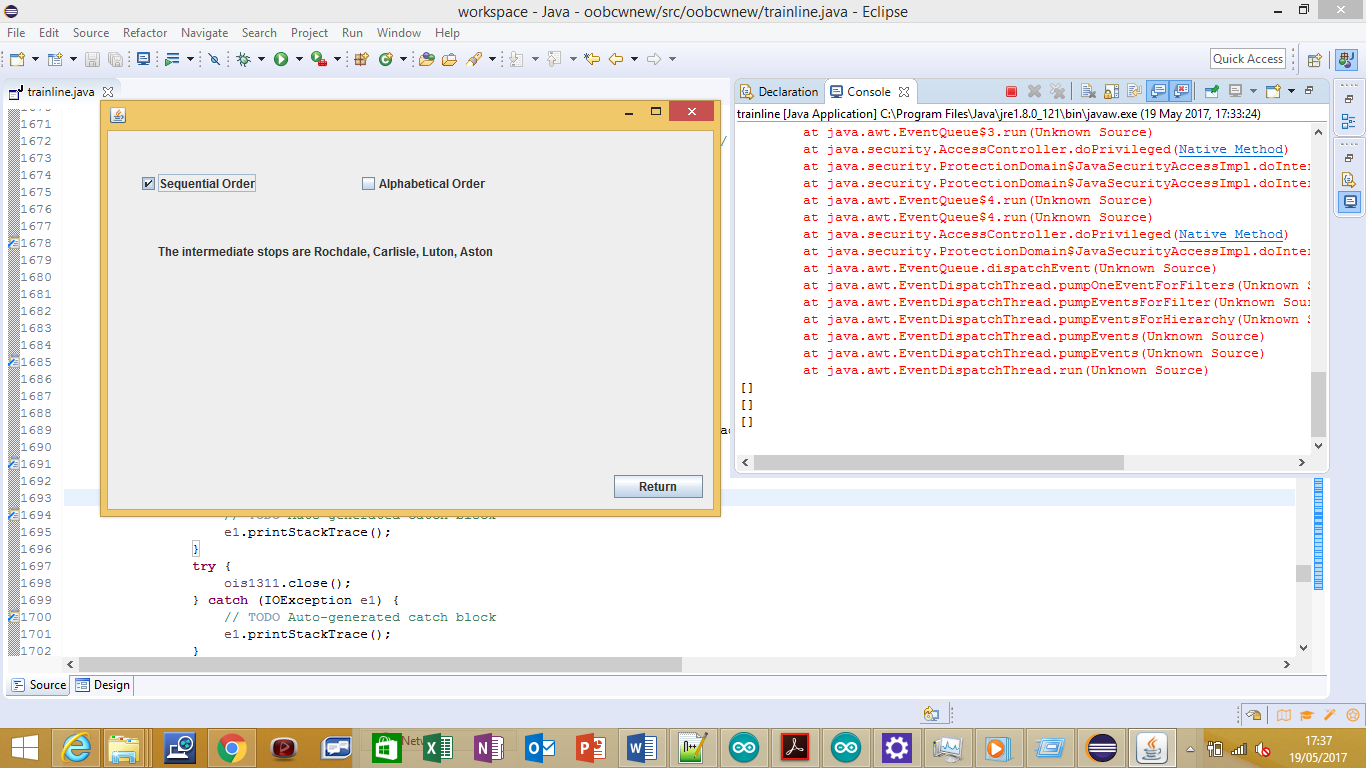
Test Data: For the route York to Nottingham add intermediate stops Rochdale, Carlisle, Luton, Aston

Expected Result: Save the route to a file and then retrieve it and display it.

Actual Result: File created containing Rochdale, Carlisle, Luton and Aston. Route was displayed when clicking on Route Button in main menu.







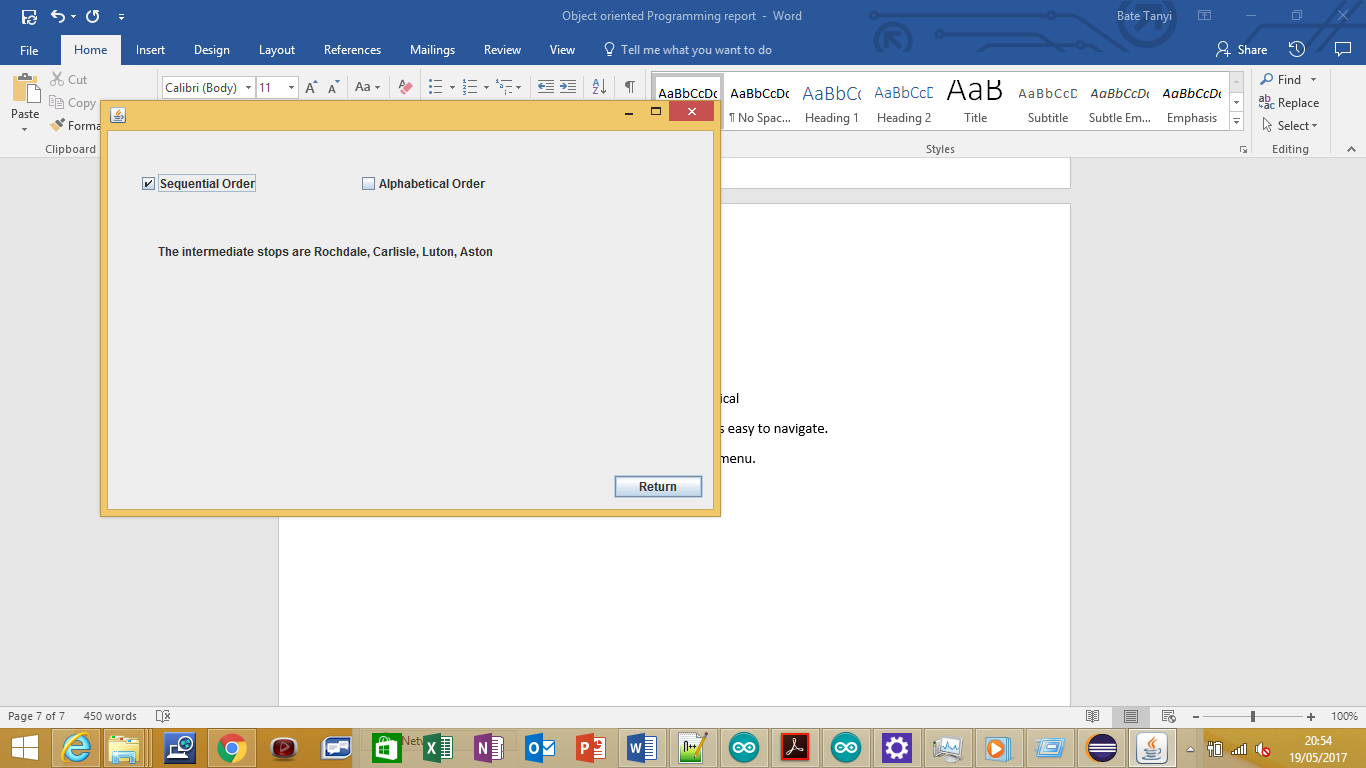
**Function/s being tested:** Implement GUI

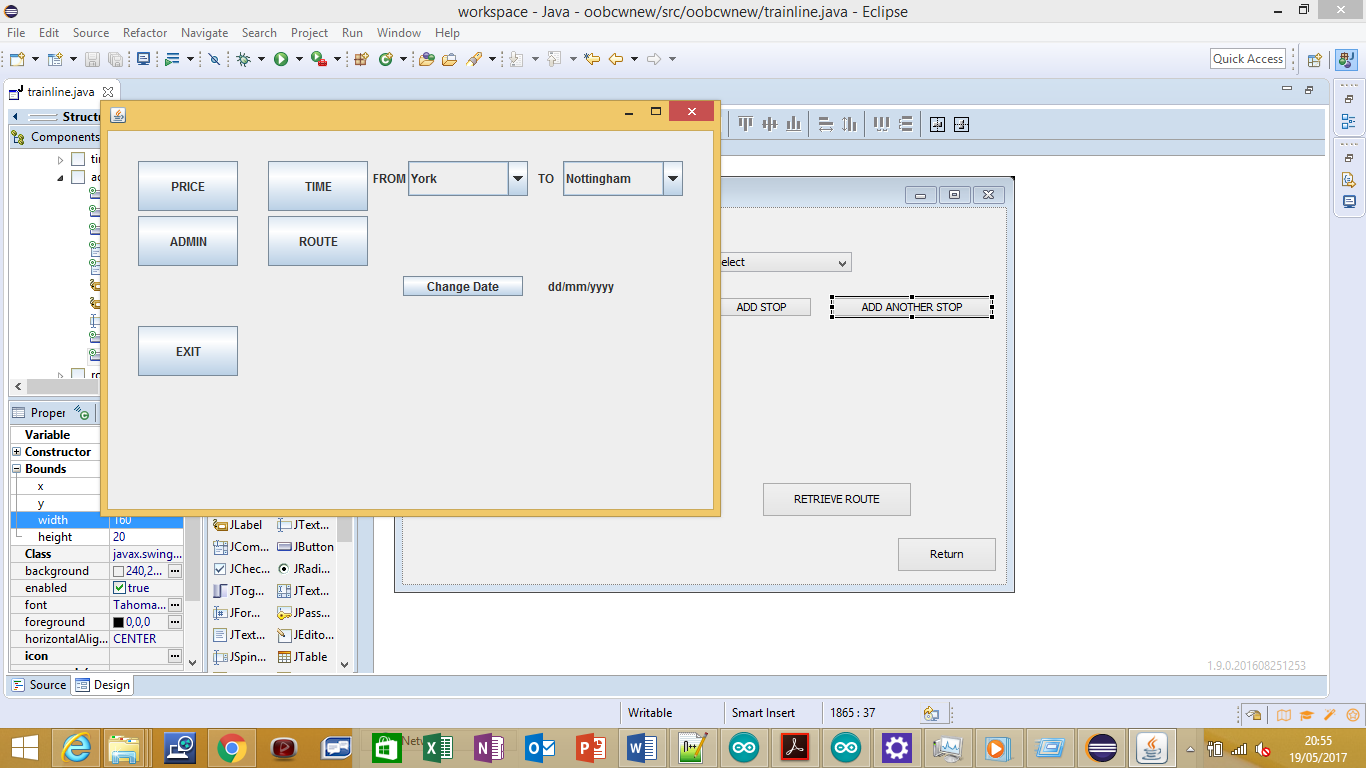
Test Data: Clicking Buttons

Expected Results: Interface can be navigated and is graphical

Actual Results: All Buttons were working and Interface was easy to navigate.

Example: when return button is clicked it returns to main menu.





**Function/s being tested:** Error trapping

Test data:

Leicester to Leicester as a Route, Date = 32/12/2017, Not selecting a Route at all. When on Admin menu select, the same train stops: Derby to Derby.

Expected Result: When Test Data entered, it will prevent a menu option from being able to be accessed. A JOption Pane will appear on the notifying the user of his error.

Actual Result: User was prevented from accessing the menu option they clicked on and a JOptionPane notifying them of their error appeared.

Example

