Loughborough University

Students must complete this cover sheet to accompany each piece of assessed coursework submitted.

Student Name: Zack Pollard

ID Number: B424338

Degree Programme: Computer Science

Part:

Module Title: OO Programming and Algorithms

Module Code: COA256

Session: 2014/2015

Semester: 2

Title/Description: Java assignment

Part: 2 of 2

This coursework represents 40 % of the module assessment

Staff Member responsible: Gerald Schaefer

Date set: 1/5/2015

Date to be handed in: 20/5/2015 Before 4pm

Specification: As on learn.
Submission via learn.

Method(s) of Presentation:

Assessment Guidelines:

If this coursework was part of a group activity, enter the Group No (if relevant):

and list the names of the other group members:

DECLARATION:

I certify that the attached coursework is my own work, except that anything which is copied from or based upon the work of others has its source clearly acknowledged.

Signature:

No. of pages following: 10

Date: 20/05/2015

No. of discs included:

Object Oriented Programming and Algorithms - Train Route Finder

Zack Pollard

May 20, 2015

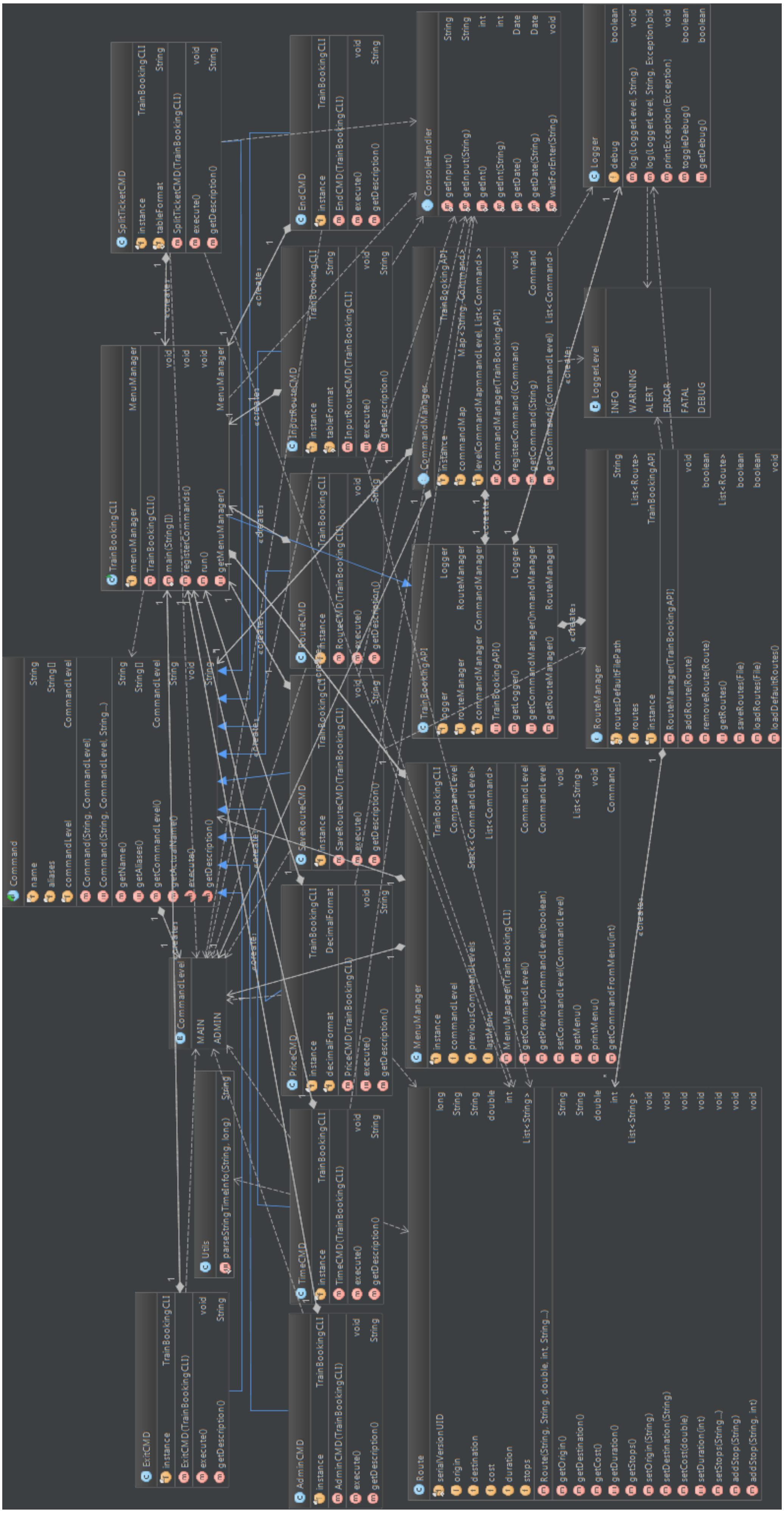
1 Design

I wrote most of my code inside an api package as the code in there had nothing to do with being a CLI. I decided to have the api package as I wasn't sure whether or not I would write a GUI or a CLI at the time, or maybe both. If I had written a GUI as well as a CLI interface, then the API would have ensured that I didn't re-write the generic code. I used an abstract class called Command in order to base all my commands off of this class, this allowed me to have a CommandManager that stored all the commands and I could do lookups and get the objects that I wanted. The class containing the main method for running the CLI is called TrainBookingCLI and it extends TrainBookingAPI in order to inherit it's methods, constructor and also the variables it has.

2 File format

The files can be saved with whatever extension is desired by the user. The default file name is main.routes which is created when the program first runs and contains all of the default routes specified in the specification. The routes data is stored using serialisation which Java provides functionality for without any external libraries. Serialisation takes all of the variables in a class and changes them into a format that can be stored to disk. In my case, I serialise the List¡Route¿ of all the routes that are currently loaded. The List is serializable by default and I have made my Route class serializable and so Java can serialize all of the data. This data is not human readable once serialized, but it is stored in such a way that when it is loaded the program can check that the stored data has the same types as the class it is being loaded into, if this is not the case, it will throw an error.

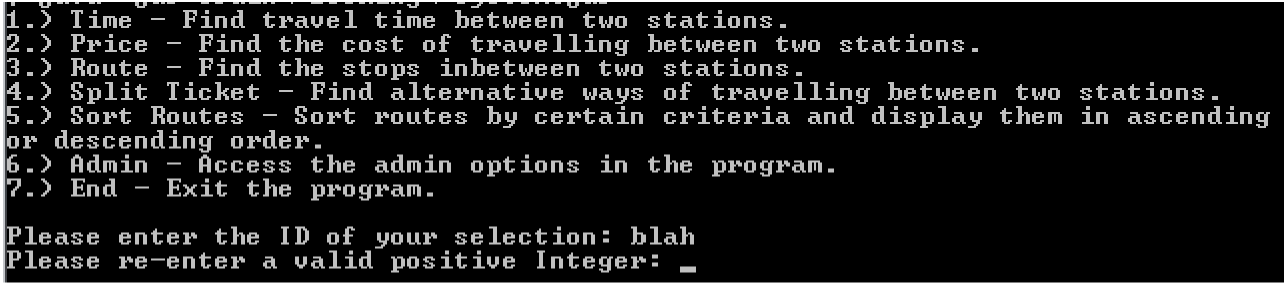
3 UML Diagram for Classes



4 Program Testing and Evidence

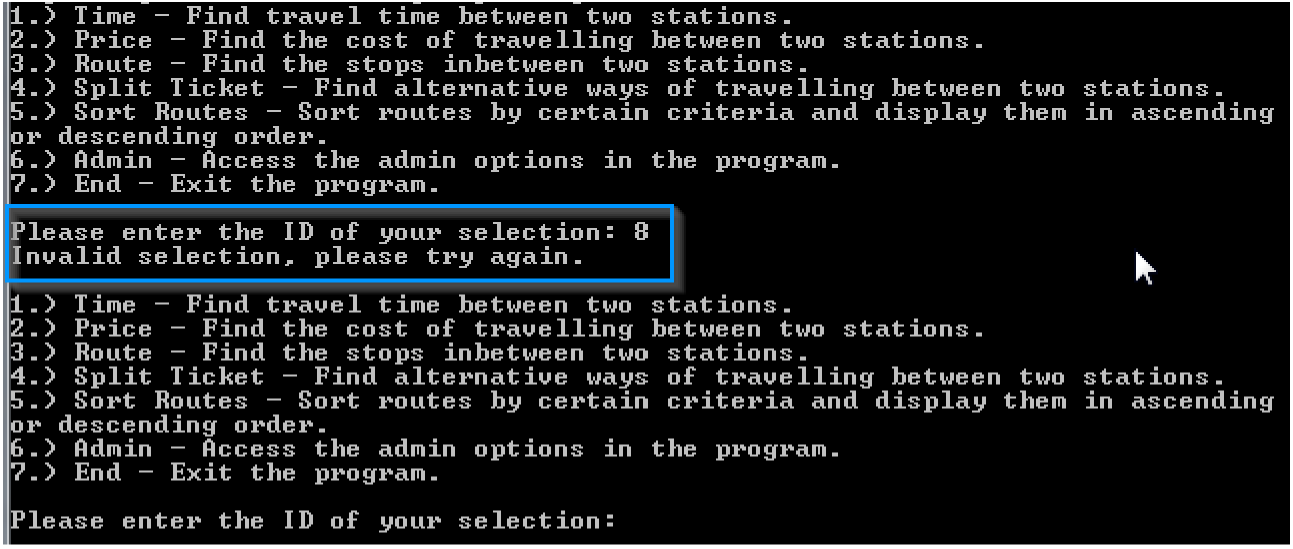
4.1 Input Data Validation

Here I will show screenshots of me testing the validation checks on data input by the user of my program.



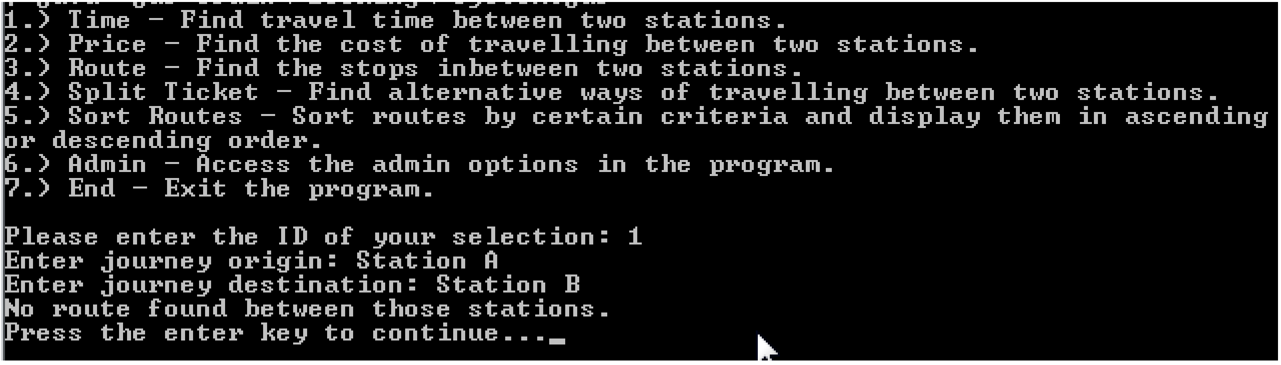
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: blah
Please re-enter a valid positive Integer: _
```

The screenshot above shows that the program checks to determine if an integer was entered on the main menu.



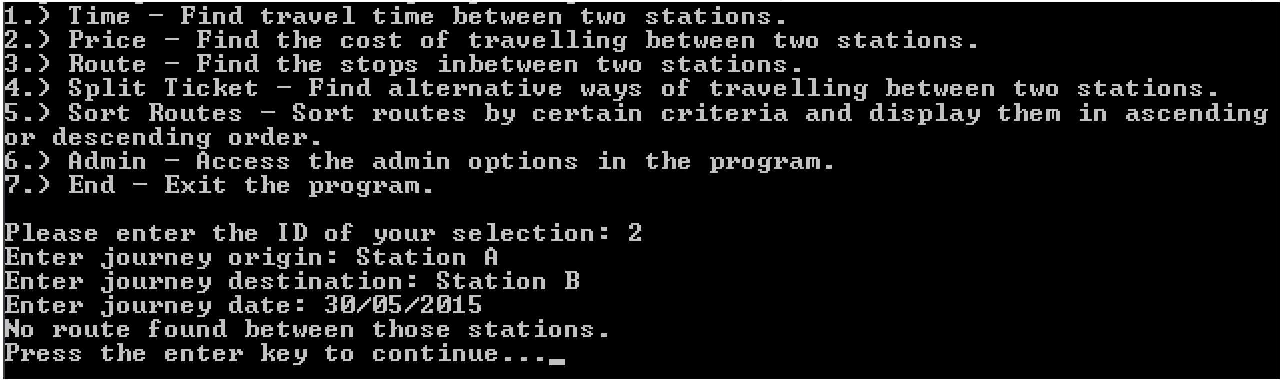
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 8
Invalid selection, please try again.

1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection:
```

The screenshot above shows that the program checks to determine if the selection made was a valid option on the main menu.



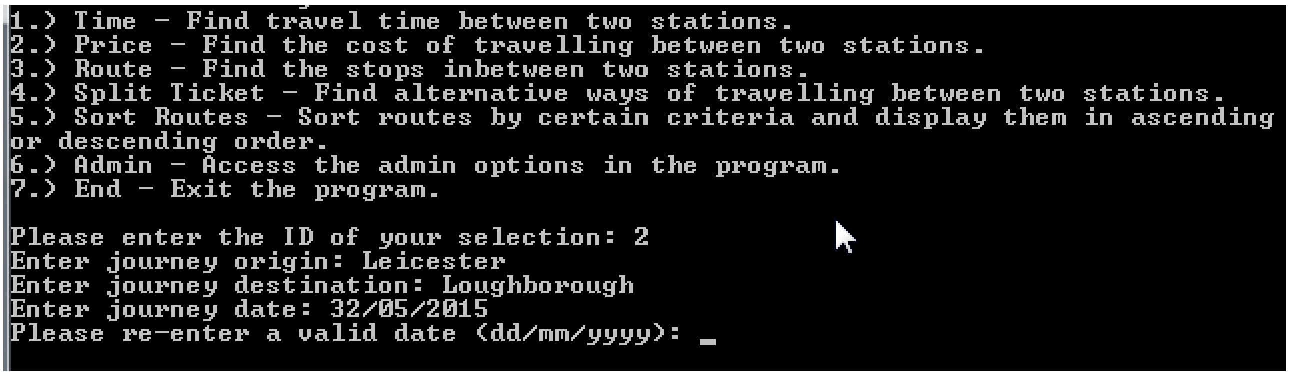
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 1
Enter journey origin: Station A
Enter journey destination: Station B
No route found between those stations.
Press the enter key to continue..._
```

The screenshot above shows that the program checks to determine if valid stations were entered in the time function.



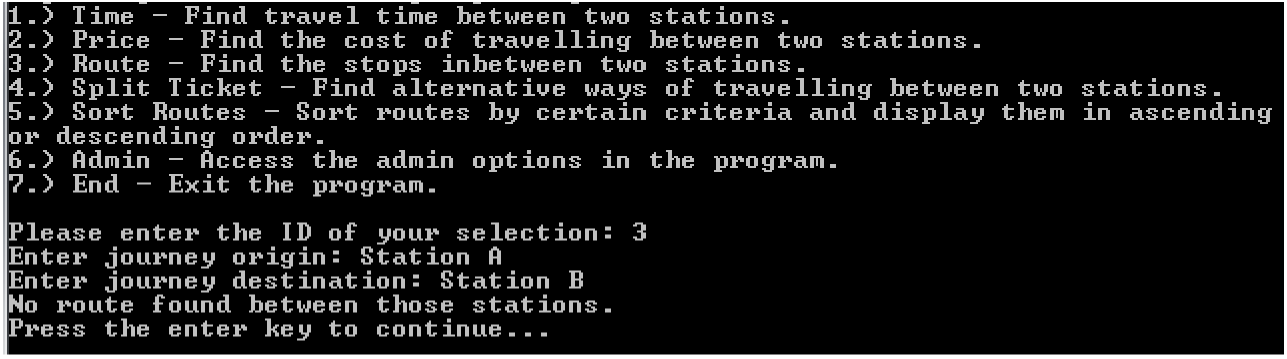
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 2
Enter journey origin: Station A
Enter journey destination: Station B
Enter journey date: 30/05/2015
No route found between those stations.
Press the enter key to continue..._
```

The screenshot above shows that the program checks to determine if valid stations were entered in the price function.



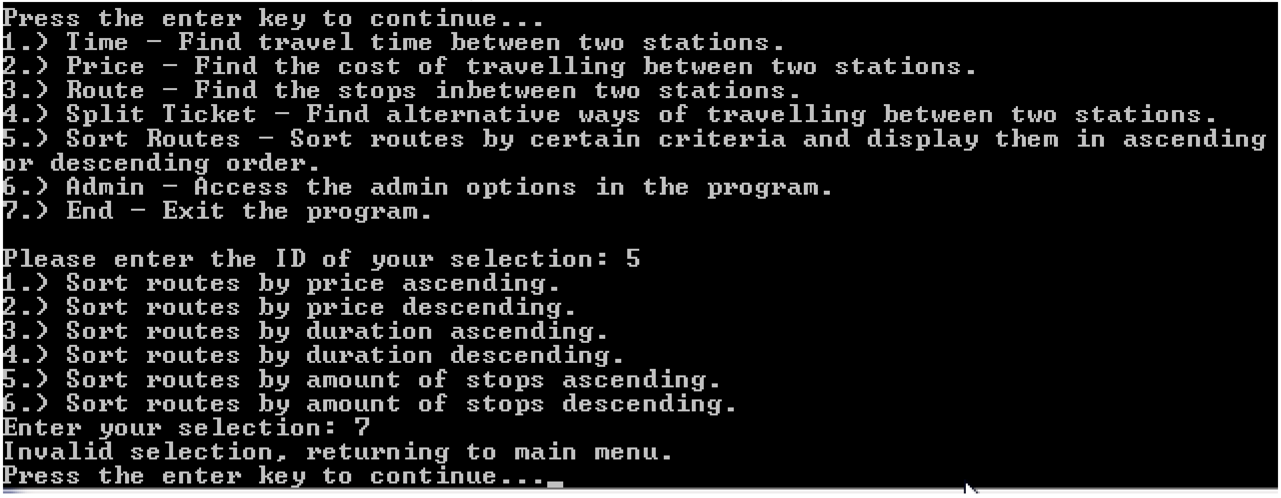
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 2
Enter journey origin: Leicester
Enter journey destination: Loughborough
Enter journey date: 32/05/2015
Please re-enter a valid date (dd/mm/yyyy): _
```

The screenshot above shows that the program checks to determine if a valid date was entered in the price function.



```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 3
Enter journey origin: Station A
Enter journey destination: Station B
No route found between those stations.
Press the enter key to continue...
```

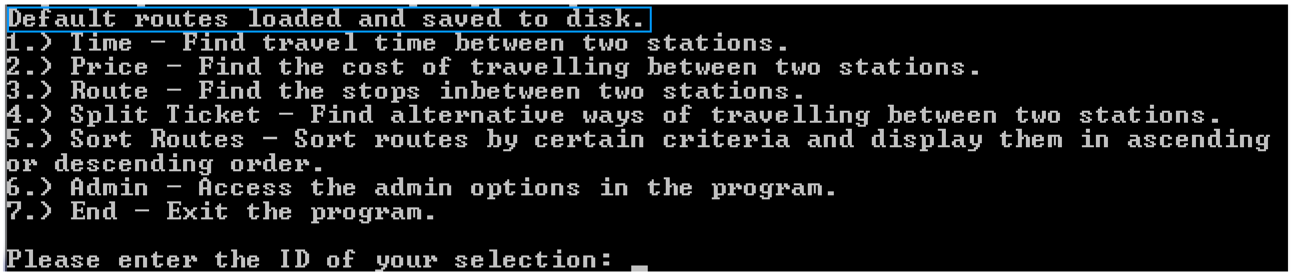
The screenshot above shows that the program checks to determine if valid stations were entered in the route function.



```
Press the enter key to continue...
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 5
1.) Sort routes by price ascending.
2.) Sort routes by price descending.
3.) Sort routes by duration ascending.
4.) Sort routes by duration descending.
5.) Sort routes by amount of stops ascending.
6.) Sort routes by amount of stops descending.
Enter your selection: 7
Invalid selection, returning to main menu.
Press the enter key to continue..._
```

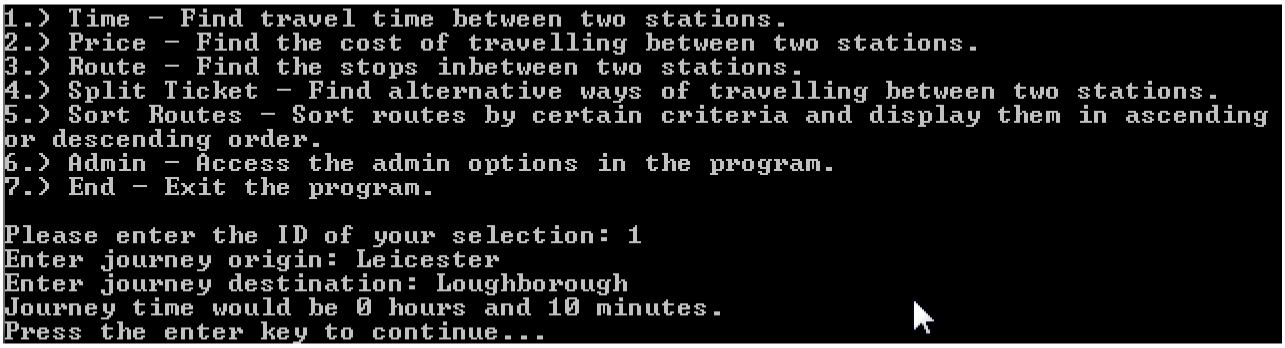
The screenshot above shows that the program checks to determine if the selection made was a valid option on the sorting selection menu.

4.2 Functionality Testing



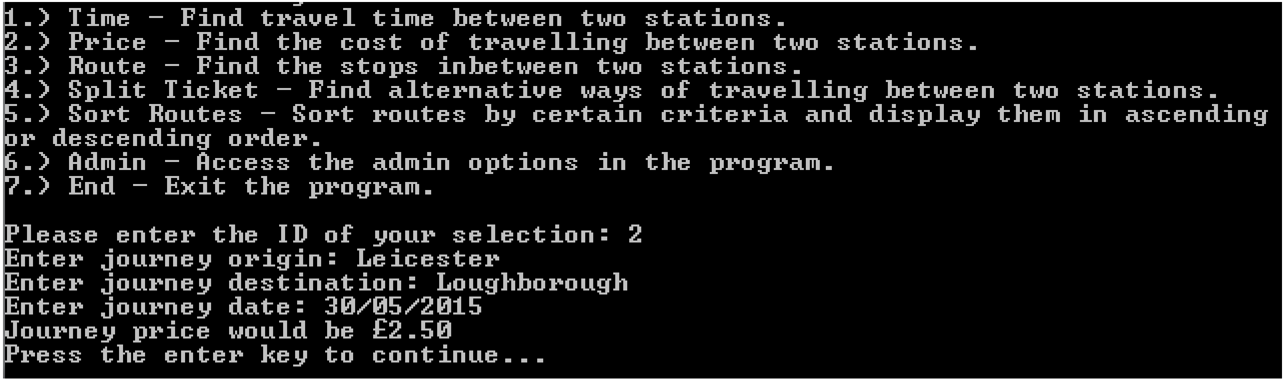
```
Default routes loaded and saved to disk.
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: _
```

The screenshot above shows that the default routes are loaded and saved to the default file location when no routes file exists on startup.



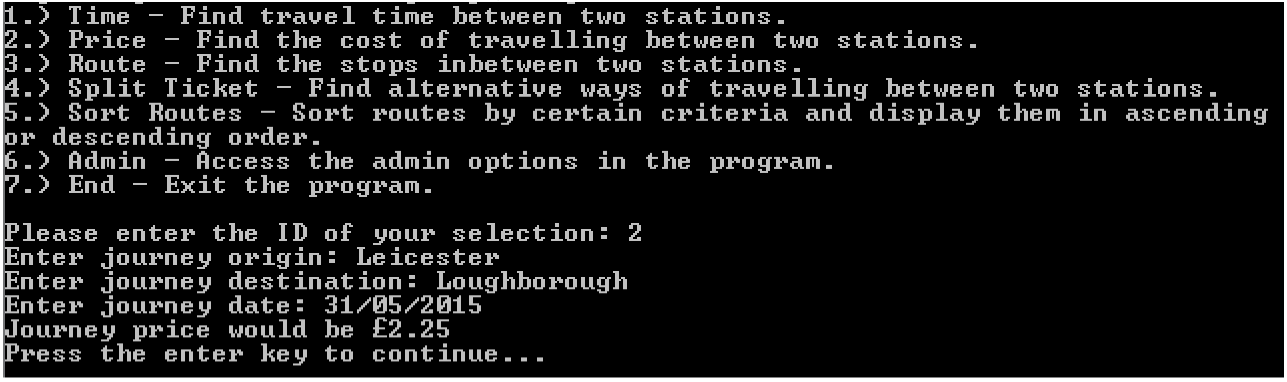
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 1
Enter journey origin: Leicester
Enter journey destination: Loughborough
Journey time would be 0 hours and 10 minutes.
Press the enter key to continue...
```

The screenshot above shows that the time function works as expected.



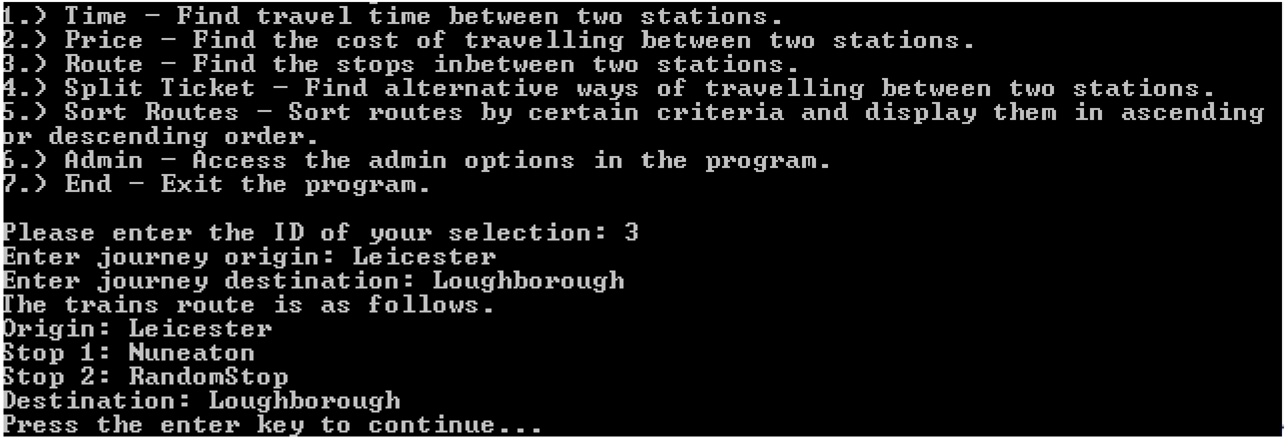
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 2
Enter journey origin: Leicester
Enter journey destination: Loughborough
Enter journey date: 30/05/2015
Journey price would be £2.50
Press the enter key to continue...
```

The screenshot above shows that the price function works as expected.



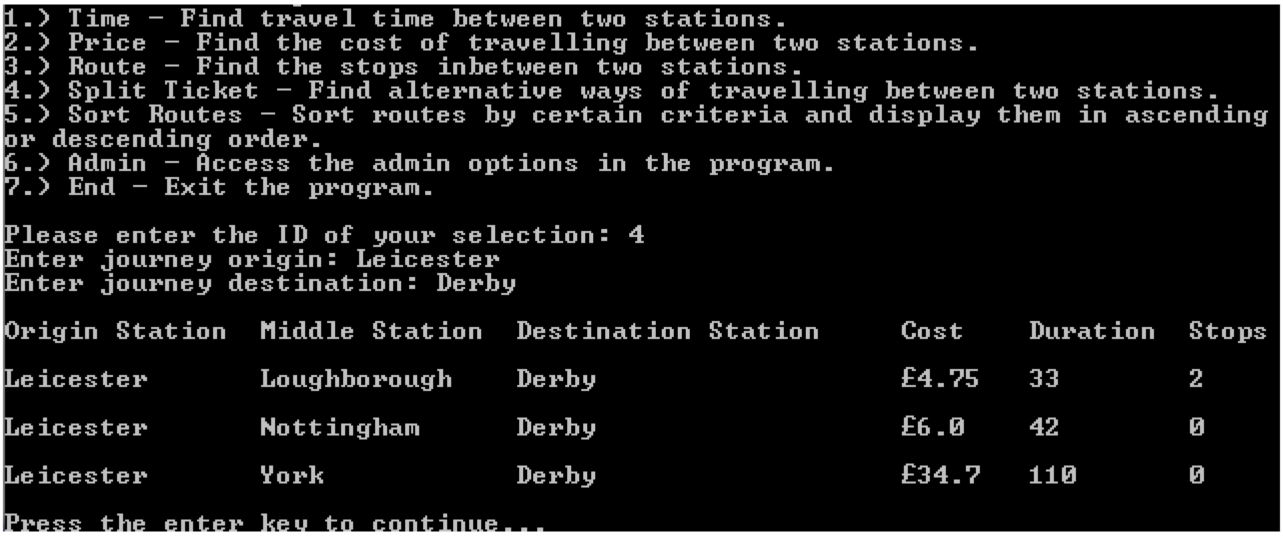
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 2
Enter journey origin: Leicester
Enter journey destination: Loughborough
Enter journey date: 31/05/2015
Journey price would be £2.25
Press the enter key to continue...
```

The screenshot above shows that the price function takes into account the last day of the month and offers a discount.



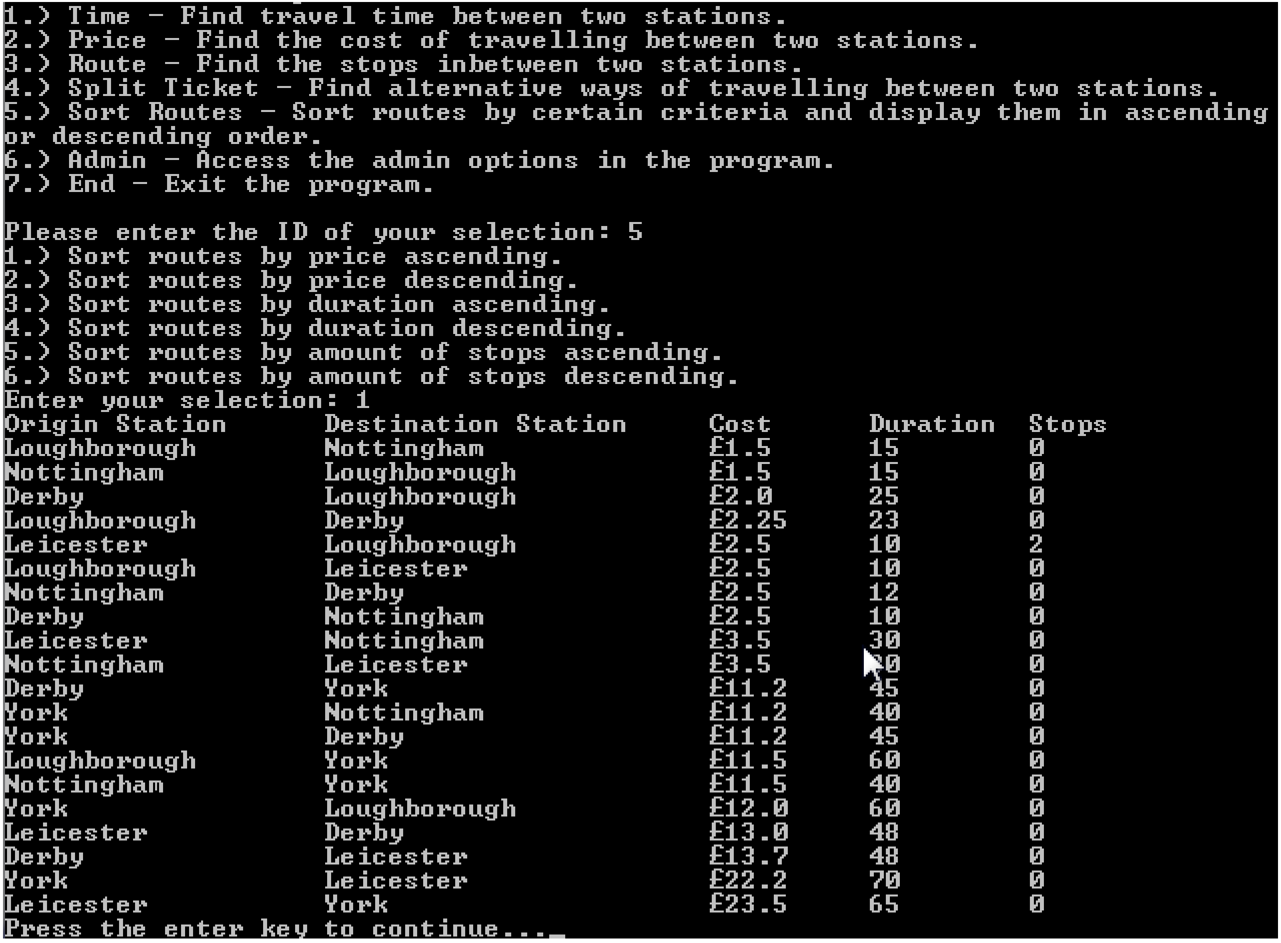
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 3
Enter journey origin: Leicester
Enter journey destination: Loughborough
The trains route is as follows.
Origin: Leicester
Stop 1: Nuneaton
Stop 2: RandomStop
Destination: Loughborough
Press the enter key to continue...
```

The screenshot above shows that the route function works as expected.



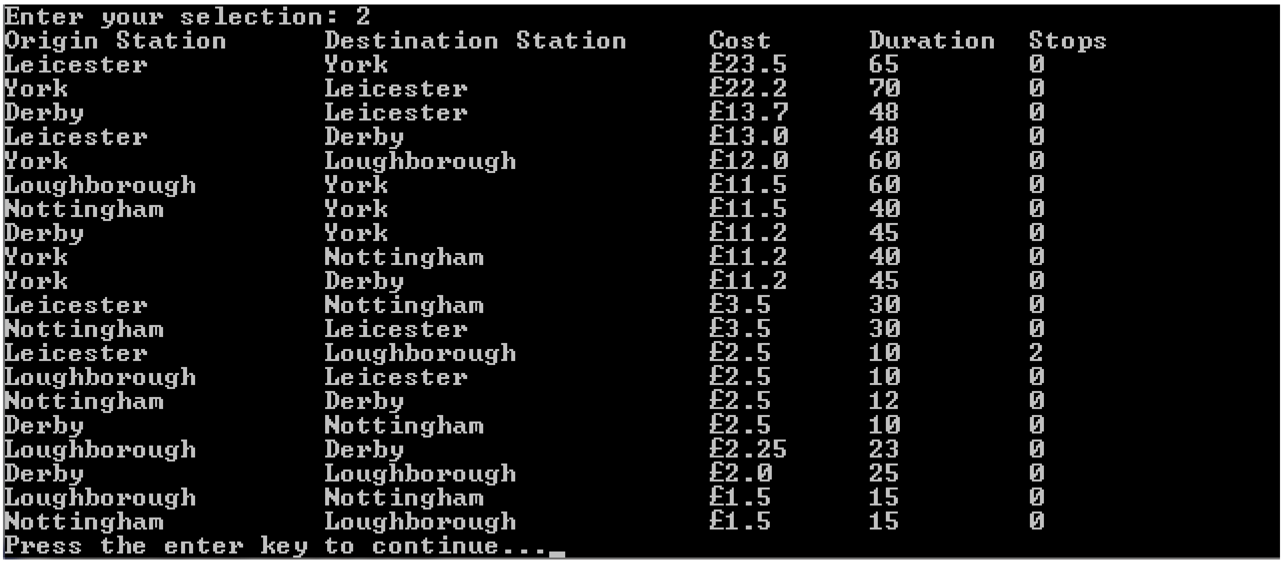
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 4
Enter journey origin: Leicester
Enter journey destination: Derby

Origin Station   Middle Station   Destination Station     Cost    Duration   Stops

Leicester        Loughborough     Derby                   £4.75   33         2

Leicester        Nottingham       Derby                   £6.0    42         0

Leicester        York             Derby                   £34.7   110        0

Press the enter key to continue...
```

The screenshot above shows that the split ticket function works as expected.



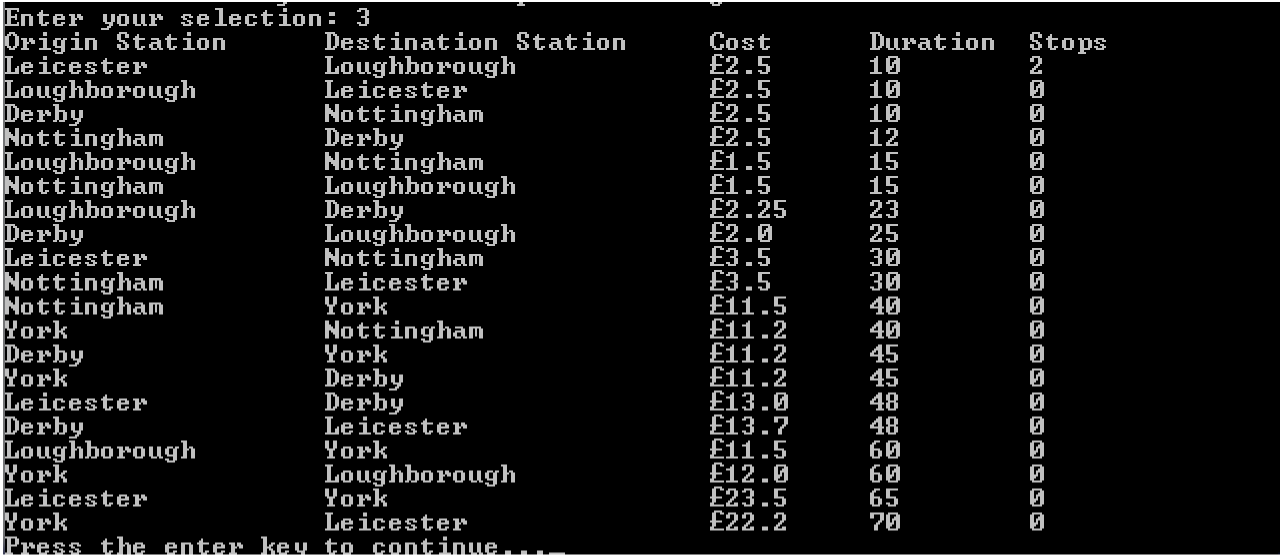
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 5
1.) Sort routes by price ascending.
2.) Sort routes by price descending.
3.) Sort routes by duration ascending.
4.) Sort routes by duration descending.
5.) Sort routes by amount of stops ascending.
6.) Sort routes by amount of stops descending.
Enter your selection: 1
Origin Station      Destination Station     Cost      Duration  Stops
Loughborough        Nottingham              £1.5      15        0
Nottingham          Loughborough            £1.5      15        0
Derby               Loughborough            £2.0      25        0
Loughborough        Derby                   £2.25     23        0
Leicester           Loughborough            £2.5      10        2
Loughborough        Leicester               £2.5      10        0
Nottingham          Derby                   £2.5      12        0
Derby               Nottingham              £2.5      10        0
Leicester           Nottingham              £3.5      30        0
Nottingham          Leicester               £3.5      30        0
Derby               York                    £11.2     45        0
York                Nottingham              £11.2     40        0
York                Derby                   £11.2     45        0
Loughborough        York                    £11.5     60        0
Nottingham          York                    £11.5     40        0
York                Loughborough            £12.0     60        0
Leicester           Derby                   £13.0     48        0
Derby               Leicester               £13.7     48        0
York                Leicester               £22.2     70        0
Leicester           York                    £23.5     65        0
Press the enter key to continue..._
```

The screenshot above shows that the sort routes function works as expected for price ascending.



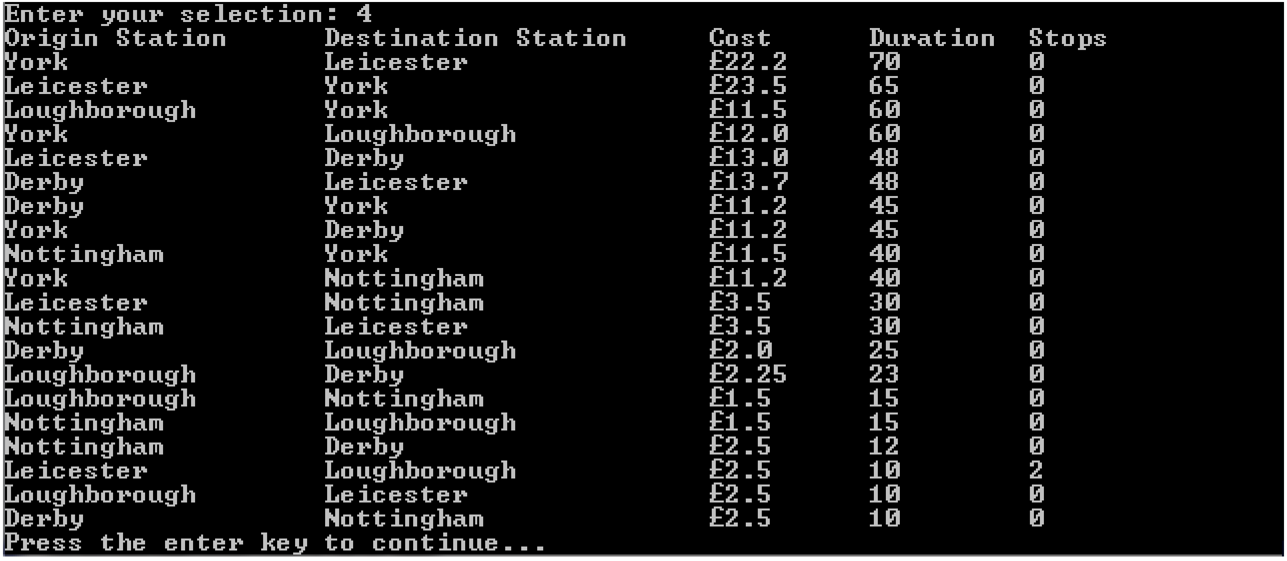
```
Enter your selection: 2
Origin Station      Destination Station     Cost      Duration  Stops
Leicester           York                    £23.5     65        0
York                Leicester               £22.2     70        0
Derby               Leicester               £13.7     48        0
Leicester           Derby                   £13.0     48        0
York                Loughborough            £12.0     60        0
Loughborough        York                    £11.5     60        0
Nottingham          York                    £11.5     40        0
Derby               York                    £11.2     45        0
York                Nottingham              £11.2     40        0
York                Derby                   £11.2     45        0
Leicester           Nottingham              £3.5      30        0
Nottingham          Leicester               £3.5      30        0
Leicester           Loughborough            £2.5      10        2
Loughborough        Leicester               £2.5      10        0
Nottingham          Derby                   £2.5      12        0
Derby               Nottingham              £2.5      10        0
Loughborough        Derby                   £2.25     23        0
Derby               Loughborough            £2.0      25        0
Loughborough        Nottingham              £1.5      15        0
Nottingham          Loughborough            £1.5      15        0
Press the enter key to continue...
```

The screenshot above shows that the sort routes function works as expected for price descending.



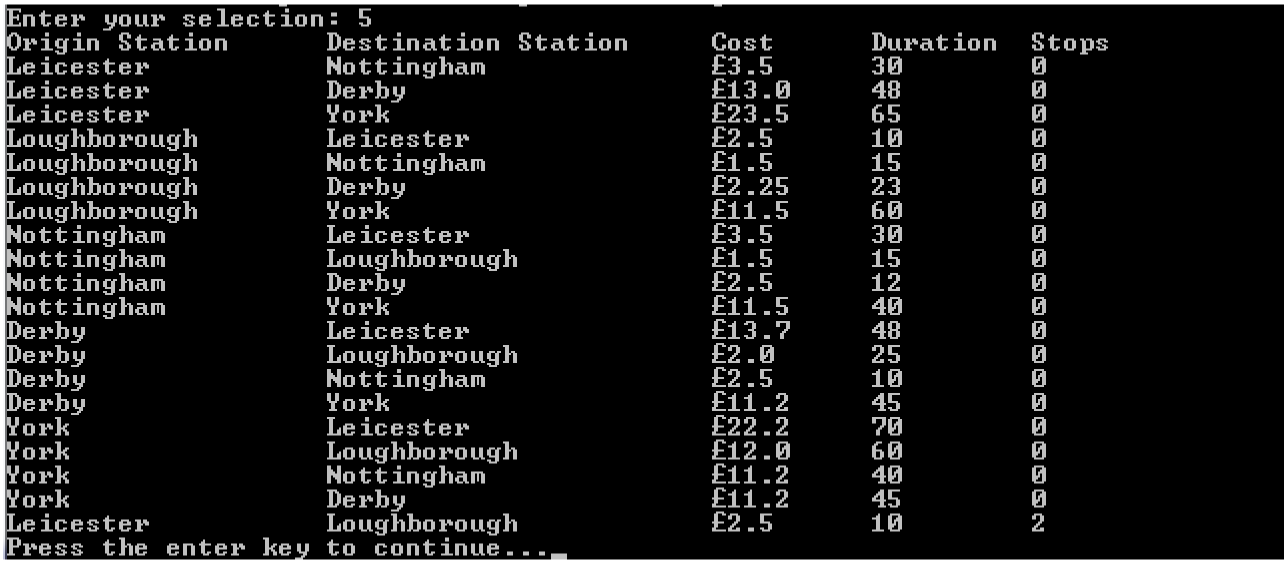
```
Enter your selection: 3
Origin Station      Destination Station     Cost      Duration  Stops
Leicester           Loughborough            £2.5      10        2
Loughborough        Leicester               £2.5      10        0
Derby               Nottingham              £2.5      10        0
Nottingham          Derby                   £2.5      12        0
Loughborough        Nottingham              £1.5      15        0
Nottingham          Loughborough            £1.5      15        0
Loughborough        Derby                   £2.25     23        0
Derby               Loughborough            £2.0      25        0
Leicester           Nottingham              £3.5      30        0
Nottingham          Leicester               £3.5      30        0
Nottingham          York                    £11.5     40        0
York                Nottingham              £11.2     40        0
Derby               York                    £11.2     45        0
York                Derby                   £11.2     45        0
Leicester           Derby                   £13.0     48        0
Derby               Leicester               £13.7     48        0
Loughborough        York                    £11.5     60        0
York                Loughborough            £12.0     60        0
Leicester           York                    £23.5     65        0
York                Leicester               £22.2     70        0
Press the enter key to continue...
```

The screenshot above shows that the sort routes function works as expected for duration ascending.



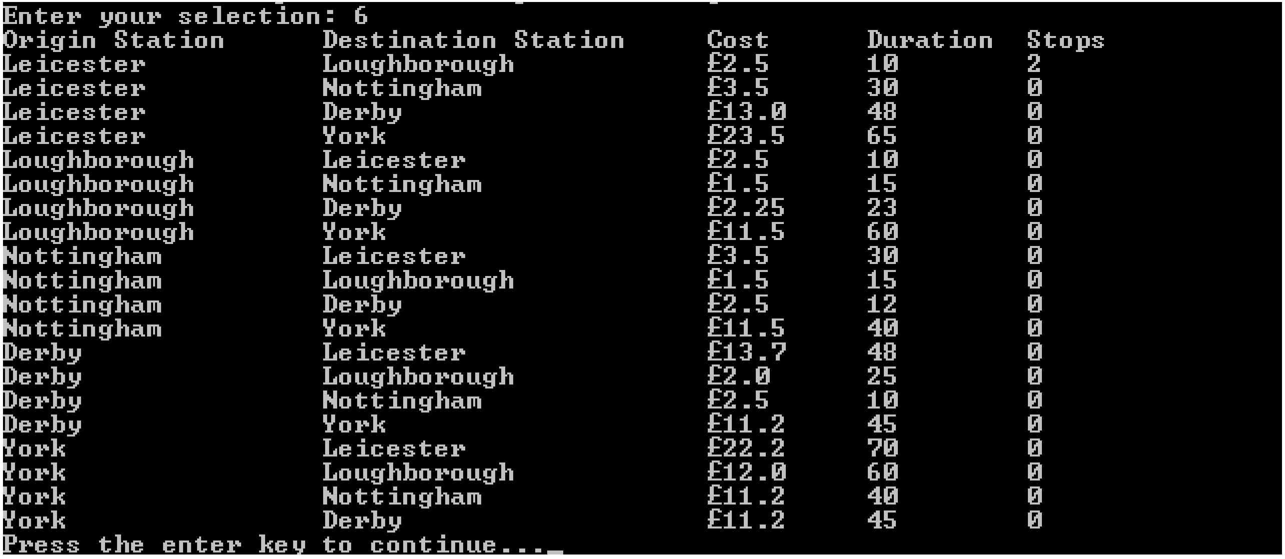
```
Enter your selection: 4
Origin Station      Destination Station     Cost      Duration  Stops
York                Leicester               £22.2     70        0
Leicester           York                    £23.5     65        0
Loughborough        York                    £11.5     60        0
York                Loughborough            £12.0     60        0
Leicester           Derby                   £13.0     48        0
Derby               Leicester               £13.7     48        0
Derby               York                    £11.2     45        0
York                Derby                   £11.2     45        0
Nottingham          York                    £11.5     40        0
York                Nottingham              £11.2     40        0
Leicester           Nottingham              £3.5      30        0
Nottingham          Leicester               £3.5      30        0
Derby               Loughborough            £2.0      25        0
Loughborough        Derby                   £2.25     23        0
Loughborough        Nottingham              £1.5      15        0
Nottingham          Loughborough            £1.5      15        0
Nottingham          Derby                   £2.5      12        0
Leicester           Loughborough            £2.5      10        2
Loughborough        Leicester               £2.5      10        0
Derby               Nottingham              £2.5      10        0
Press the enter key to continue...
```

The screenshot above shows that the sort routes function works as expected for duration descending.



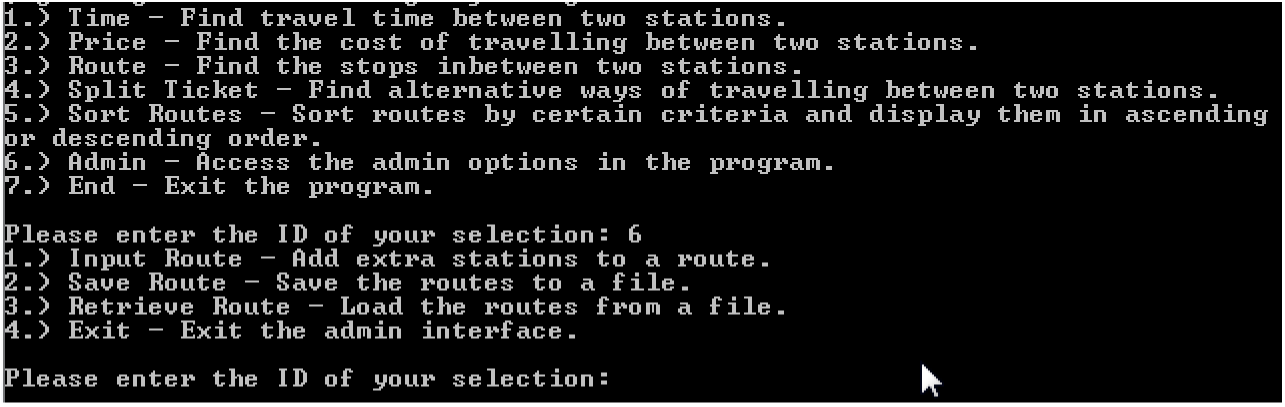
```
Enter your selection: 5
Origin Station      Destination Station     Cost      Duration  Stops
Leicester           Nottingham              £3.5      30        0
Leicester           Derby                   £13.0     48        0
Leicester           York                    £23.5     65        0
Loughborough        Leicester               £2.5      10        0
Loughborough        Nottingham              £1.5      15        0
Loughborough        Derby                   £2.25     23        0
Loughborough        York                    £11.5     60        0
Nottingham          Leicester               £3.5      30        0
Nottingham          Loughborough            £1.5      15        0
Nottingham          Derby                   £2.5      12        0
Nottingham          York                    £11.5     40        0
Derby               Leicester               £13.7     48        0
Derby               Loughborough            £2.0      25        0
Derby               Nottingham              £2.5      10        0
Derby               York                    £11.2     45        0
York                Leicester               £22.2     70        0
York                Loughborough            £12.0     60        0
York                Nottingham              £11.2     40        0
York                Derby                   £11.2     45        0
Leicester           Loughborough            £2.5      10        2
Press the enter key to continue...
```

The screenshot above shows that the sort routes function works as expected for routes ascending.



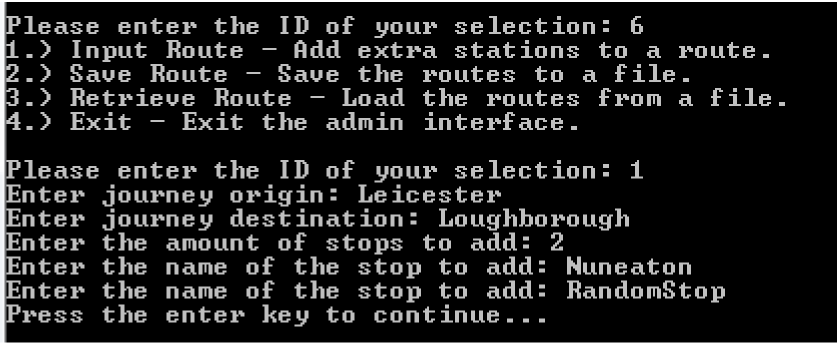
```
Enter your selection: 6
Origin Station      Destination Station     Cost      Duration  Stops
Leicester           Loughborough            £2.5      10        2
Leicester           Nottingham              £3.5      30        0
Leicester           Derby                   £13.0     48        0
Leicester           York                    £23.5     65        0
Loughborough        Leicester               £2.5      10        0
Loughborough        Nottingham              £1.5      15        0
Loughborough        Derby                   £2.25     23        0
Loughborough        York                    £11.5     60        0
Nottingham          Leicester               £3.5      30        0
Nottingham          Loughborough            £1.5      15        0
Nottingham          Derby                   £2.5      12        0
Nottingham          York                    £11.5     40        0
Derby               Leicester               £13.7     48        0
Derby               Loughborough            £2.0      25        0
Derby               Nottingham              £2.5      10        0
Derby               York                    £11.2     45        0
York                Leicester               £22.2     70        0
York                Loughborough            £12.0     60        0
York                Nottingham              £11.2     40        0
York                Derby                   £11.2     45        0
Press the enter key to continue...
```

The screenshot above shows that the sort routes function works as expected for routes descending.



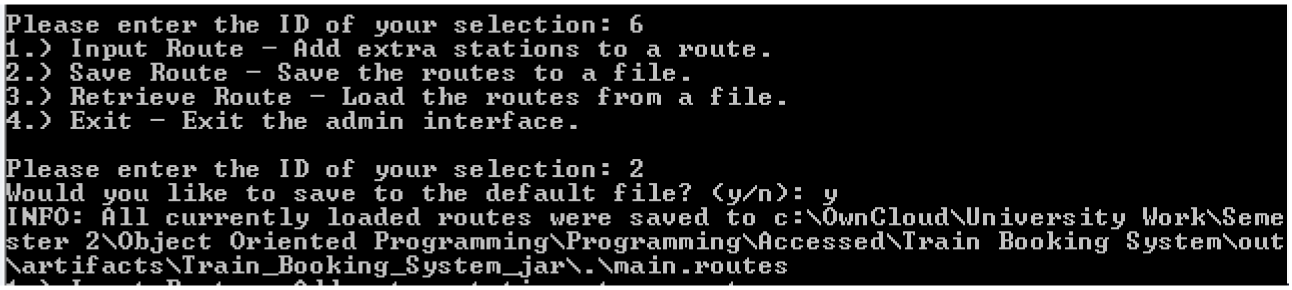
```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 6
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection:
```

The screenshot above shows that the admin menu loads when you select it from the main menu.



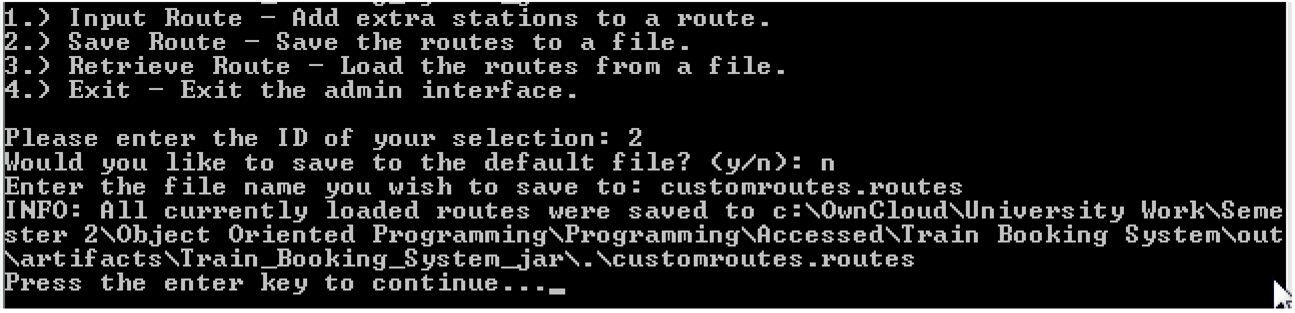
```
Please enter the ID of your selection: 6
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection: 1
Enter journey origin: Leicester
Enter journey destination: Loughborough
Enter the amount of stops to add: 2
Enter the name of the stop to add: Nuneaton
Enter the name of the stop to add: RandomStop
Press the enter key to continue...
```

The screenshot above shows that the input route function works as expected.



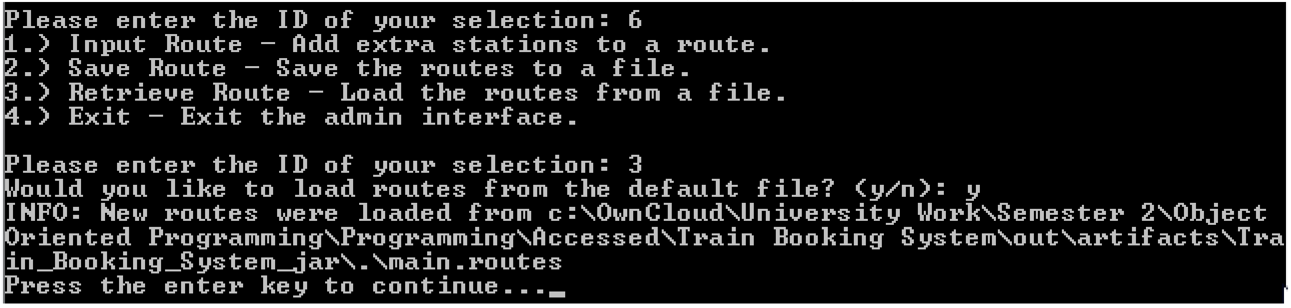
```
Please enter the ID of your selection: 6
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection: 2
Would you like to save to the default file? (y/n): y
INFO: All currently loaded routes were saved to c:\OwnCloud\University Work\Seme
ster 2\Object Oriented Programming\Programming\Accessed\Train Booking System\out
\artifacts\Train_Booking_System_jar\.\main.routes
```

The screenshot above shows that the save routes function works for saving to the default file.



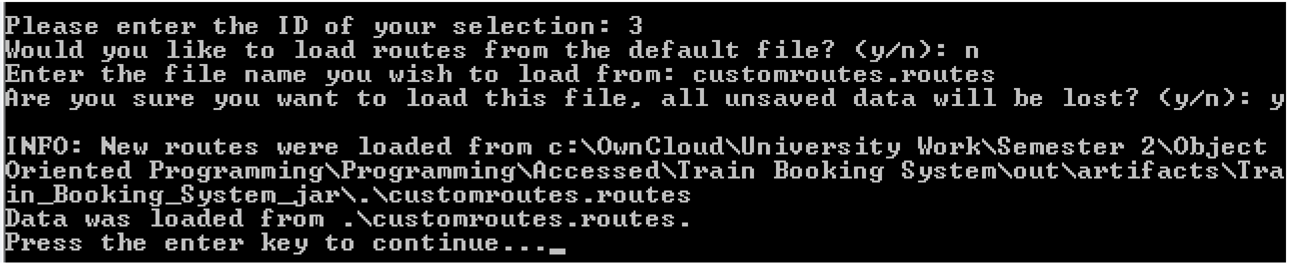
```
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection: 2
Would you like to save to the default file? (y/n): n
Enter the file name you wish to save to: customroutes.routes
INFO: All currently loaded routes were saved to c:\OwnCloud\University Work\Seme
ster 2\Object Oriented Programming\Programming\Accessed\Train Booking System\out
\artifacts\Train_Booking_System_jar\.\customroutes.routes
Press the enter key to continue..._
```

The screenshot above shows that the save routes function works for saving to a custom file.



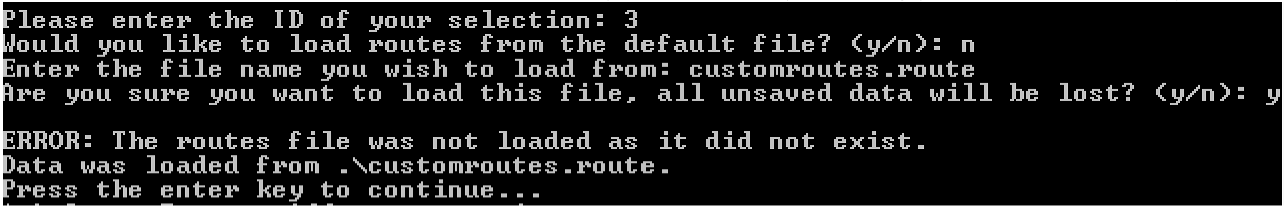
```
Please enter the ID of your selection: 6
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection: 3
Would you like to load routes from the default file? (y/n): y
INFO: New routes were loaded from c:\OwnCloud\University Work\Semester 2\Object
Oriented Programming\Programming\Accessed\Train Booking System\out\artifacts\Tra
in_Booking_System_jar\.\main.routes
Press the enter key to continue..._
```

The screenshot above shows that the retrieve routes function works for loading from the default file.



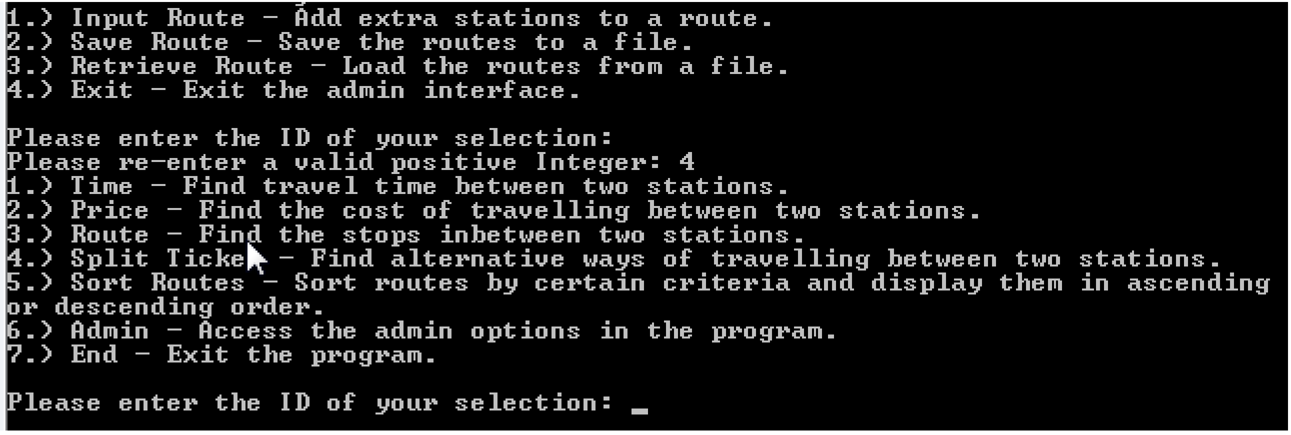
```
Please enter the ID of your selection: 3
Would you like to load routes from the default file? (y/n): n
Enter the file name you wish to load from: customroutes.routes
Are you sure you want to load this file, all unsaved data will be lost? (y/n): y

INFO: New routes were loaded from c:\OwnCloud\University Work\Semester 2\Object
Oriented Programming\Programming\Accessed\Train Booking System\out\artifacts\Tra
in_Booking_System_jar\.\customroutes.routes
Data was loaded from .\customroutes.routes.
Press the enter key to continue..._
```

The screenshot above shows that the retrieve routes function works for loading from a custom routes file.



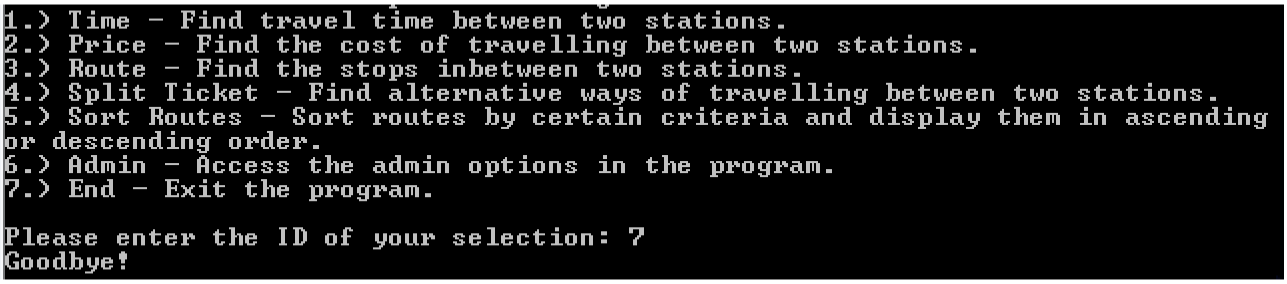
```
Please enter the ID of your selection: 3
Would you like to load routes from the default file? (y/n): n
Enter the file name you wish to load from: customroutes.route
Are you sure you want to load this file, all unsaved data will be lost? (y/n): y

ERROR: The routes file was not loaded as it did not exist.
Data was loaded from .\customroutes.route.
Press the enter key to continue...
```

The screenshot above shows that the retrieve routes function detects if the file does not exist when requested for loading.



```
1.) Input Route - Add extra stations to a route.
2.) Save Route - Save the routes to a file.
3.) Retrieve Route - Load the routes from a file.
4.) Exit - Exit the admin interface.

Please enter the ID of your selection:
Please re-enter a valid positive Integer: 4
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: _
```

The screenshot above shows that the exit option works and returns to the previous menu.



```
1.) Time - Find travel time between two stations.
2.) Price - Find the cost of travelling between two stations.
3.) Route - Find the stops inbetween two stations.
4.) Split Ticket - Find alternative ways of travelling between two stations.
5.) Sort Routes - Sort routes by certain criteria and display them in ascending
or descending order.
6.) Admin - Access the admin options in the program.
7.) End - Exit the program.

Please enter the ID of your selection: 7
Goodbye!
```

The screenshot above shows that the end option works and exits the program.

5 Functionality

Functionality	**Y(Complete)**
	P(Partial)
	N(None)
		Comments (e.g. more details on what is not working etc.)

Search for price	Y	
Search for travel time	Y	
Display route	Y	
Split ticket	Y	Displayed a list of all available split routes as was unsure on what was required.
Sort route data	Y	I provided the option of sorting it ascending and descending as I wasn't sure if you wanted one or both.
Input route	Y	
Load route (from file)	Y	
Save route (to file)	Y	
Handling dates correctly	Y	
Error handling	Y	All errors that could be a problem are handled by the Logger and will be printed out to console with both the stacktrace and an informative message as to what went wrong.