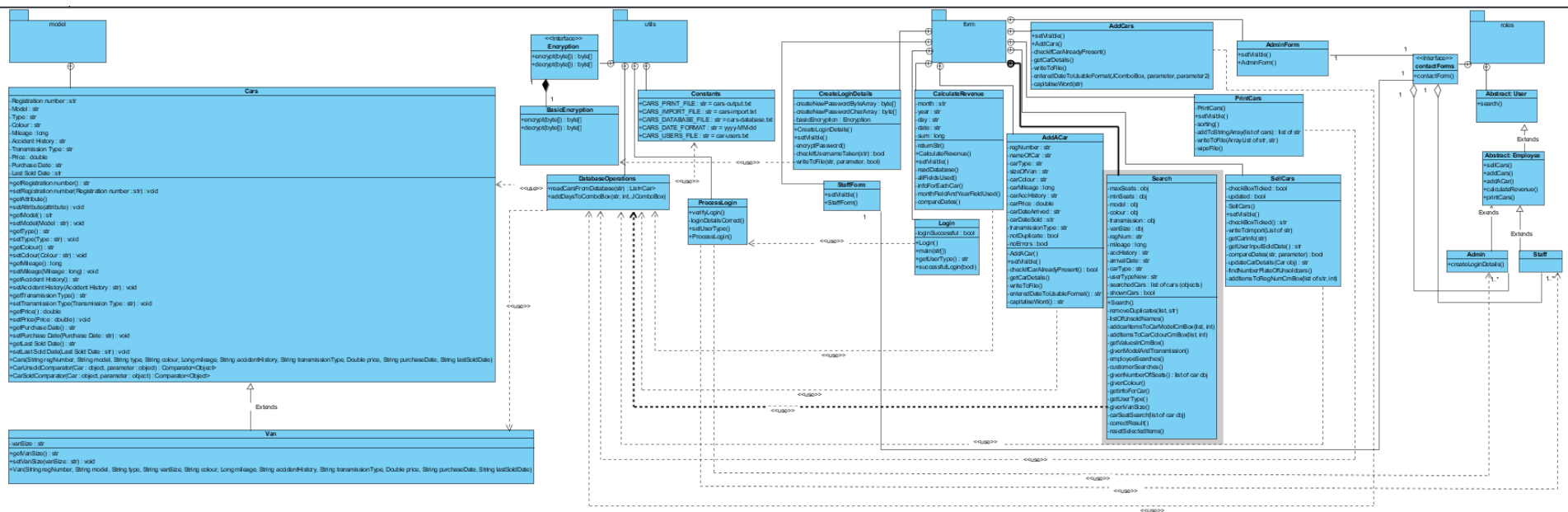


Object Oriented Programming COA256

CARDEALER.JAR

(s) Anmol Dadlani
B830342



I have designed this project with 4 packages:

- **Model** – contains a car class and a van class. The van class extends the car class as the car class contains all the attributes the van does but does not include van size. The classes in this package contain getters and setters for every attribute. This would make it easier to obtain the value of the attributes of each car and also set them to certain values.
- **Form** – contains all the GUI windows. Every function has its own window as the way in which IntelliJ makes Graphical User Interfaces is by linking one GUI Form with one Java class. Therefore, I had to make each function into a class. Majority of the classes use the DatabaseOperations class in the utils package to get an array list of objects. These objects are cars.
- **Roles** – contains all classes related to the users. It contains an interface that is called upon to contact files in the Form folder. This means that the staff is always calling a handler in order to make the windows appear on the screen. The user is an abstract class which means that methods are declared without the need of implementation of the methods. However, the user only contains the search as this is the only function in common. There is another abstract class in this package which is called Employee. This contains all of the functions except search and create login credentials. Staff and Admin extend this abstract class. Aggregation arrows are used when the user classes are contacting interface. This is because, the user will still exist if the interface does not.
- **Utils** – this contains the classes that are known as Helper classes and only contain static methods so they cannot be instantiated. The files that are contained in this are the Encryption files, basic Database operations file, Login processing file and constants file. The encryption interface has a composition arrow from Basic Encryption to itself, this means that the Basic Encryption file cannot exist without the Encryption interface. The interface will contain two methods, encrypt and decrypt; this will be used for the passwords entered by the user. Database Operations has a function that reads the cars-database.txt file and turns each car into an object and returns an ArrayList of Cars. This can then be easily used when getting the values for each car in the functions.

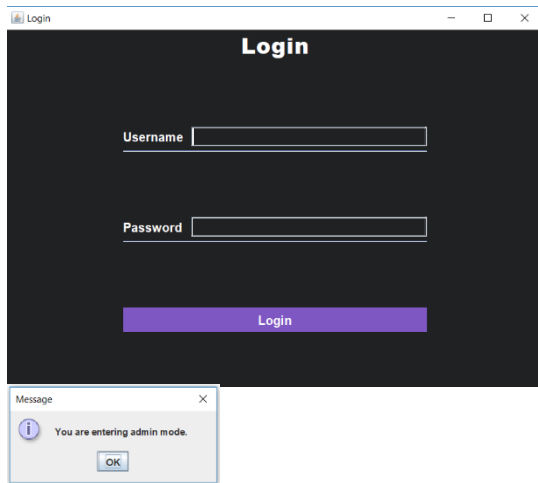
The Visual Paradigm file is in the folder if you require access to it to view it in greater depth.

Documentation – User Instructions

The jar file is in the out\artifacts\cardealer_jar filepath. You can either double click it from there or open it in command line.

On the day of submission, I tried to run my jar file in command line, the login page loads but when the details are entered, in command line, a file not found exception is thrown. However, if you double click the jar file in the folder, it works perfectly. If you open cars-database.txt in Notepad ++, you will be able to see each car on a new line. Also, I have used “|” instead of commas. In addition, I have added “N/A” at the end of the car in the database when it has not been sold.

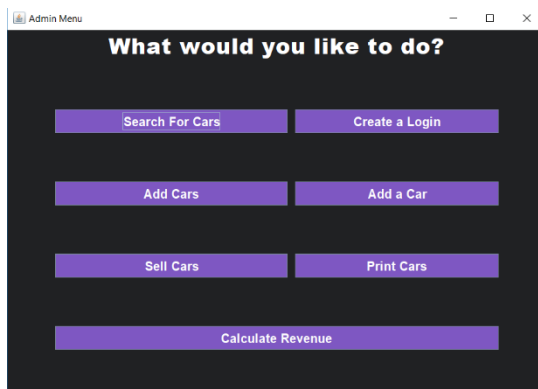
Start-up



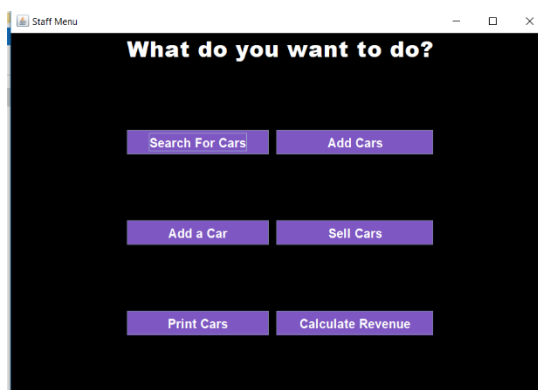
This is the start-up GUI window for the project. Here you enter the login details and depending on your username, the system figures out which menu to show. For example:

- Admin details are used, a message pops up informing the user they are entering admin mode
- Staff details are used, a staff message pops up
- Customer details are used, a customer message pops up.

Menus



The GUI window on the left is the window that opens when the user presses 'OK' on the message message dialog.



The GUI window on the left is the window that opens when the user presses 'OK' on the message message dialog.

This menu does not include Create a Login as an admin is the only one that is allowed to do it.

The GUI window on the left is the window that opens when the user presses 'OK' on the message message dialog.

The customer can only access the search function so this is the only window that opens.

Search

This is the Search Menu to find all cars. The car model combo box reads the car database file and for every different car model, it adds it as an element. The same happens for colour.

There are checkboxes at the start and depending on which one has been selected, the correct options are set to visible and show the comboBoxes.

This is an example of a search result. A picture of the car shows along with all of the details of the car.

The result depends on which user has logged in. The first one has been accessed via a customer login. Therefore, the accident history has not been revealed.

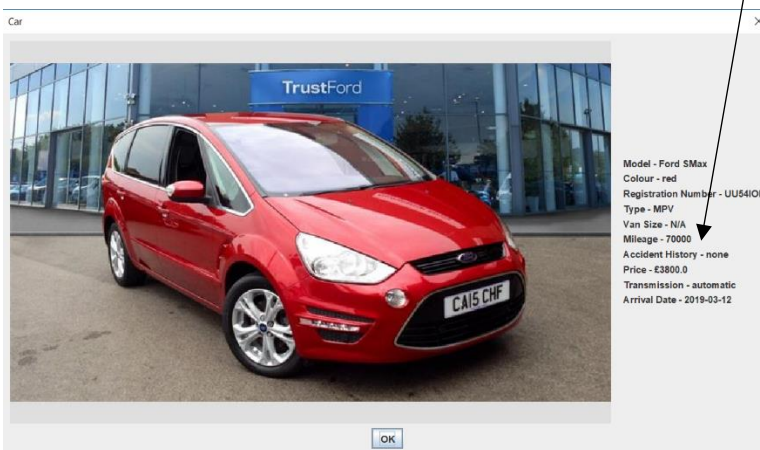
The second one has been accessed via a staff or admin login. Thus, the accident history details have been revealed.

User needs to press the OK button to view the next car. Also there is only a finite list of car pictures so if a new car is added to the database, a picture will not load up. The admin or staff will need to add a picture of the car to the carPhotos folder.

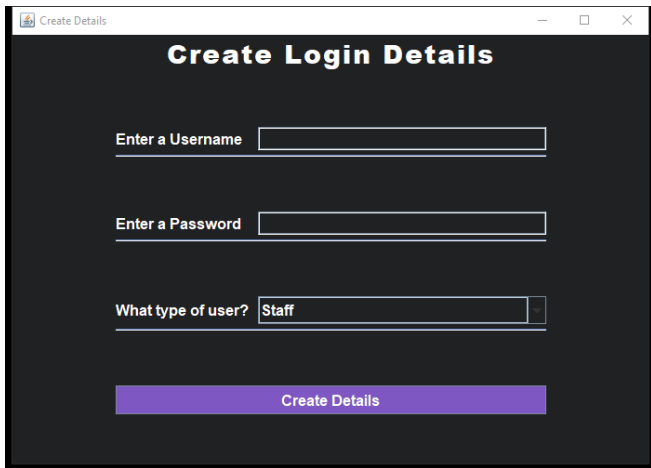
Everytime the user searches something and there are no cars, no Message Dialog box will appear.

After the user searches for a car, the values in the comboBox reset so that the user can search for another car with a different criteria.

NOTE: PICTURES OF CARS HAVE BEEN REMOVED DUE TO COPYRIGHTING ISSUES.



Create Login Details



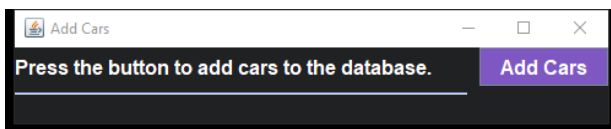
When the Create Login Details button is pressed on the Admin Menu, this window pops up and then it tells the admin to enter the details.

If any of the fields are left blank, an error message pops up.

You cannot make a new admin credential so the JComboBox is restricted to Staff and Customer.

When the login details have been created, a message pops up saying it has been created.

Add Cars



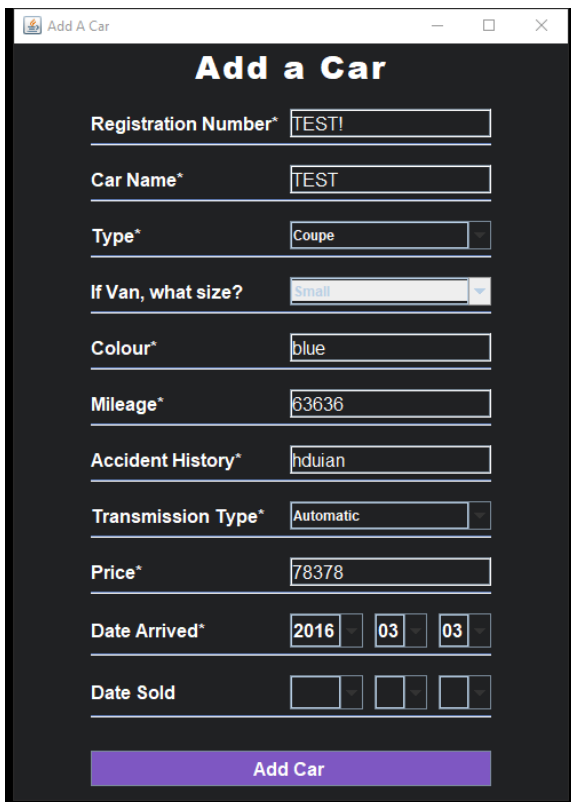
This window pops up when the Add Cars button is pressed.

Once the Add Cars button is pressed, a message pops up saying all the cars have been added to the database.

Add cars appends cars that are not in the database already.

NOTE: In my code I have user “|” (bars) to split the details in my cars-import file so any other character will not work with this code.

Add A Car



Add A Car button loads up this window with the fields the admin or staff need to fill in, in order to add a Car to the database.

When the Type is set to van, only then does the van size Combo Box enables, however if the Type is changed after being set to Van, the combo box does not disable again. This does not affect the details of the car as the code later recognises if the type is not “Van” then the van size is blank.

Once they have filled in the details, the car is added.

Depending on the year and month chosen, the day combo box updates itself.

Sell Car

Which car did you sell?

☐ Tick this if the car was sold today

If not, when was it sold?

Year

Month

Day

Update Details

This consists of the initial comboBox that has a list of all the Cars that are available. The user continues to fill in the details and a message pops up when the car has been updated.

Print Cars

Press the button to print the cars in the database

```
Sold Cars:
Y01808A|Honda|Jazz|Hatchback|[grey|none|manual|8000.0|2018-09-01|2018-10-04
N0170FG|Lexus|IS|Saloon|[dark blue|none|manual|26100.0|2018-10-04|2018-10-14
V010088|Vauxhall|Movano|Van|large|white|rear windscreen damage|manual|5000.0|2019-01-10|2019-03-01
P014EP0|Mercedes|EClass|Coupe|[yellow|none|automatic|18400.0|2018-03-08|2019-03-02
P013PZF|Vauxhall|Insignia|Saloon|[green|none|automatic|10800.0|2018-08-10|2019-03-05
H011PM|BMW|1|Saloon|[black|door damage|manual|19000.0|2019-03-01|2019-03-05

Unsold Cars:
H019P0H|Jaguar|FType|Coupe|[silver|none|automatic|52000.0|2018-03-01|N/A
H067TTP|Mercedes|Citan|Van|small|white|front windscreen damage|automatic|10000.0|2018-08-04|N/A
P013PZF|Vauxhall|Insignia|Saloon|[red|none|manual|18000.0|2018-08-10|N/A
A013QE|Ford|Transit|Van|small|black|none|automatic|7300.0|2018-11-01|N/A
H067TTP|Mercedes|Citan|Van|small|white|front damage|manual|9500.0|2018-11-04|N/A
B062N8|Lexus|IS|Saloon|[white|indent on passenger door|automatic|18100.0|2019-03-02|N/A
P064N0|Vauxhall|Movano|Van|large|silver|none|automatic|8800.0|2019-03-11|N/A
F18AT0H|Fiat|Bathobile|Coupe|[black|none|automatic|99999.0|2019-03-11|N/A
T156A0A|Lexus|RX|SUV|[dark blue|none|automatic|11800.0|2019-03-12|N/A
H05410P|Ford|SMax|MPV|[red|none|automatic|3800.0|2019-03-12|N/A
H05410I|Ford|SMax|MPV|[red|none|manual|11000.0|2019-05-02|N/A
```

When pressed, the cars-ouput.txt file gets updated with the list of all the cars in two separate lists (as shown).

Calculate Revenue

Which year's revenue do you want?

Which month's revenue do you want?

Which day's revenue do you want?

Calculate Revenue

This contains the dates that change with the year and month selected. A message pops up showing the revenue for that month or day.

Packages

- Form
- Model
- Utils
- Roles

Description of Object-Oriented Design

There are four key object-oriented programming concepts that are required to be implemented:

1. Encapsulation.

Each class has a set of private variables and few public methods. This means that each of my classes contain attributes but also methods for it to work. Majority of my methods are private and void but there are many examples that return Boolean, list of objects, list of arrays, strings, and integers.

2. Abstraction.

Abstract classes contain methods that do not need to be implemented and contain only the function names. This means that when the function is used in the extended classes, they can have unique methods.

In my program, I have abstract User and Employee classes which are extended by specialised Users and/or Employees. For example, Customer and Employee extend the User class and the Employee class is extended by the staff and admin.

The User class only contains the search function as this is the only function all users have in common. The Employee class contains the remaining functions minus the create login credentials function as the only type of user able to create them is the admin.

3. Inheritance.

The Staff and Admin classes extend the Employee classes. This means that they inherit all the methods in the Employee abstract class.

The van class is an extension of the car class as it uses the same characteristics as a car but is an instantiation of it. Therefore, it has one more property, which is van size.

The van class also implements Serializable which translates the object (in this case, a car or a van) into a format that can be reconstructed later.

4. Polymorphism.

For example, the staff class extends the employee searches class that contains the basic method names with no code. This means that even though the methods in User and Admin are the same, they contain different sets of coding.

Every time the database operations class file is read, it uses the getters and setters in the car class to return an ArrayList of objects. This is made possible by the Serializable interface, mentioned earlier. This interface makes it easier to access the details of the vehicles. Database Operations also includes a function that uses the car objects to find all different entries. For example, in search, the database is read, and all the different car models are found along with the colours and placed as items in the comboBox.

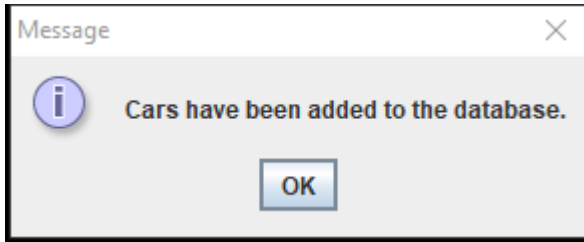
There is also a constants file that makes it easier to call final strings, so the names do not need to be remembered, they are saved.

Test Plans

Functionality Name	Test Data	Outcome	Was the outcome expected?	Image Ref.
AddCars	Press the button on the GUI window.	Displays 'cars have been added to the database' message.	Yes	1
Add A Car	Enter the details of the car correctly.	Van size combobox should not become enabled. Message should be outputted saying it has been added to the database.	Yes	2
Add A Car	Enter details incorrectly.	Error message should be shown saying that the details such as price and mileage are not double or long numbers, respectively.	Yes	3
Calculate Revenue	Enter nothing in the comboBoxes.	Outputs a message telling the user to fill in the starred details.	Yes	4
Calculate Revenue	Enter month and year.	Shows what the revenue is for that time frame.	Yes	5
Create Login Details	Have the username blank but password filled.	Shows an error message.	Yes	6
Create Login Details	Enter details correctly.	Shows a message saying it has been created.	Yes	7
Print Cars	Press the button	Shows message that the list can now be printed.	Yes	8
Search Cars	Depending on the Check box ticked, the corresponding comboBoxes appear.	Correct comboBoxes appear.	Yes	9
Search Cars	Search for colour and each car pops up with its details.	Correct car appears.	Yes	10
Sell Cars	Press radioButton for today's date.	ComboBoxes should disappear.	Yes	11

Images

Ref. 1

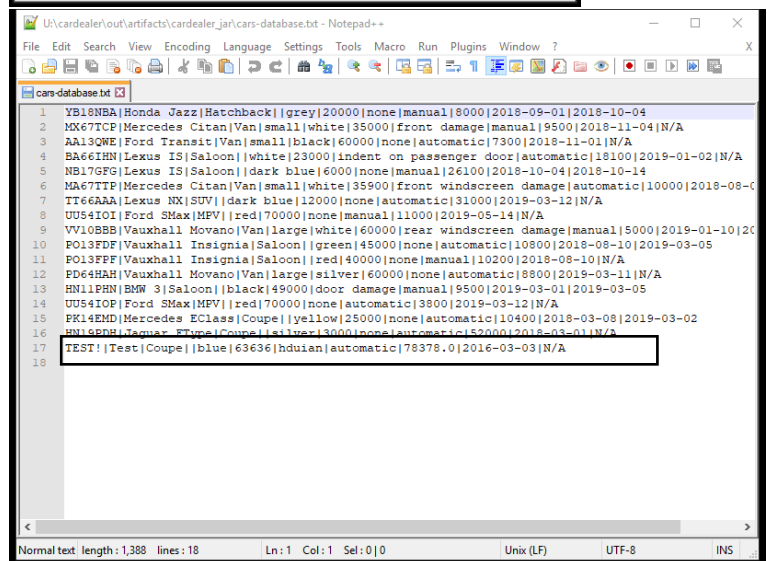
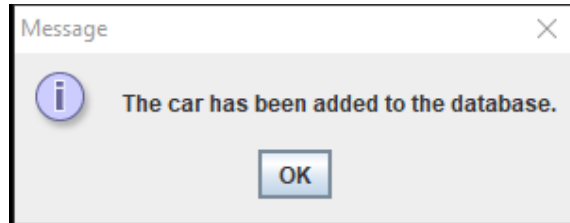


Ref. 2

A form titled 'Add a Car' with the following fields and values:

- Registration Number*: TEST!
- Car Name*: TEST
- Type*: Coupe
- If Van, what size?: Small
- Colour*: blue
- Mileage*: 63636
- Accident History*: hduian
- Transmission Type*: Automatic
- Price*: 78378
- Date Arrived*: 2016 03 03
- Date Sold: (empty)

At the bottom is a purple 'Add Car' button.

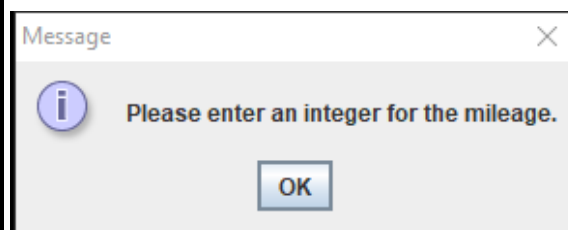


Ref. 3

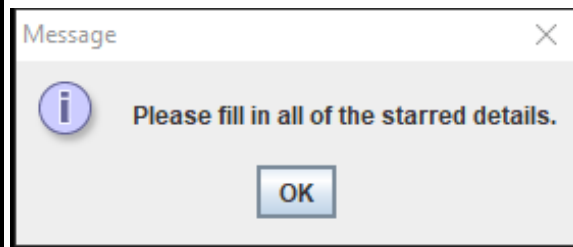
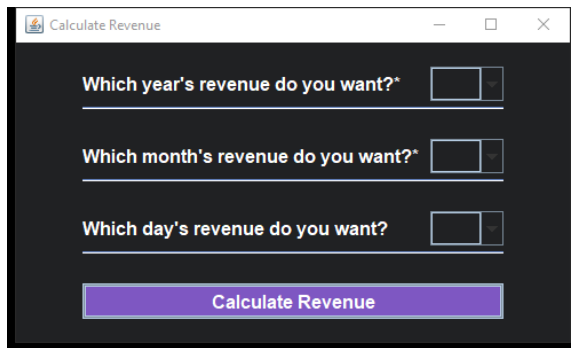
A form titled 'Add a Car' with the following fields and values:

- Registration Number*: (empty)
- Car Name*: (empty)
- Type*: Coupe
- If Van, what size?: Small
- Colour*: (empty)
- Mileage*: (empty)
- Accident History*: (empty)
- Transmission Type*: Automatic
- Price*: (empty)
- Date Arrived*: (empty)
- Date Sold: (empty)

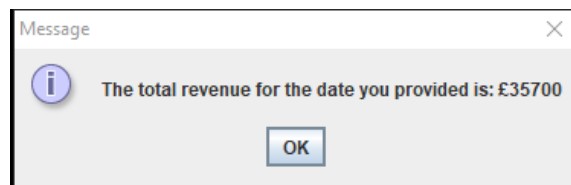
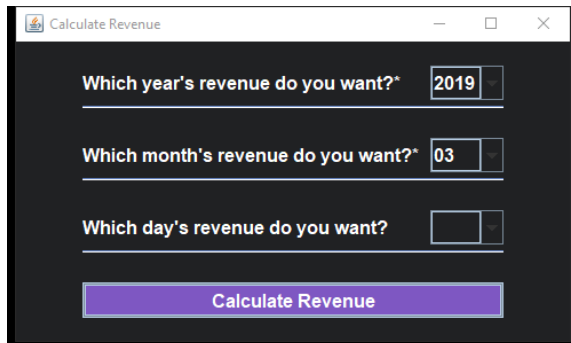
At the bottom is a purple 'Add Car' button.



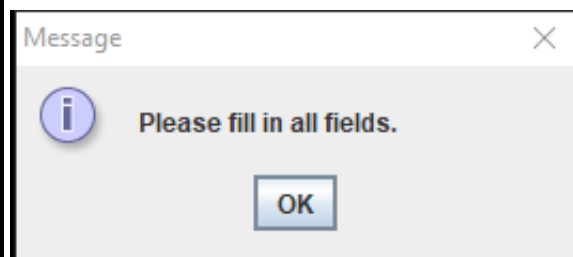
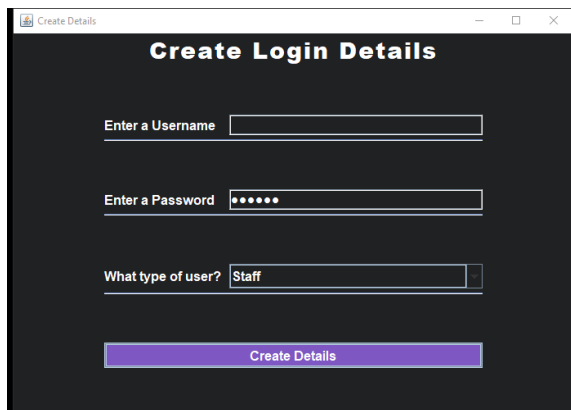
Ref. 4



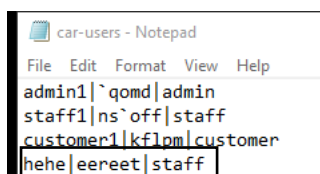
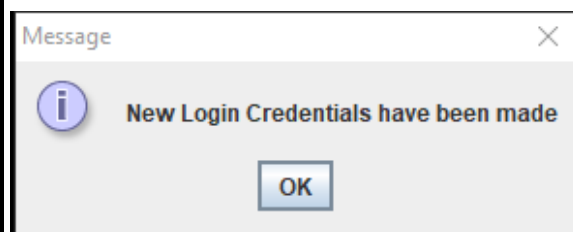
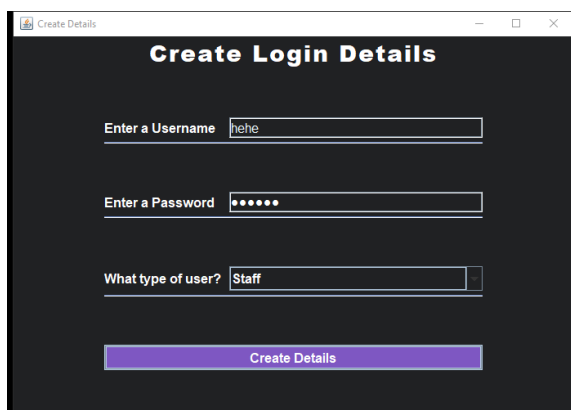
Ref. 5



Ref.6



Ref. 7



```
car-users - Notepad
File Edit Format View Help
admin1|`qomd|admin
staff1|ns`off|staff
customer1|kf1pm|customer
hehe|eereet|staff
```

Ref. 8

Calculate Revenue

Press the button to print the cars in the database

Print

Message

All of the vehicles have been sorted and can now be printed.

OK

cars-output - Notepad

File Edit Format View Help

Sold Cars:
YB180B4|Honda Jazz|Hatchback||grey|none>manual|8000.0|2018-09-01|2018-10-04
NB17GFG|Lexus IS|Saloon||dark blue|none>manual|26100.0|2018-10-04|2018-10-14
VW1088B|Vauxhall Movano|Van|large|white|rear windscreen damage>manual|5000.0|2019-01-10|2019-03-01
PK14EMD|Mercedes EClass|Coupe||yellow|none|automatic|10400.0|2018-03-08|2019-03-02
P013FDF|Vauxhall Insignia|Saloon||green|none|automatic|10800.0|2018-08-10|2019-03-05
HN11PHH|BMW 3|Saloon||black|door damage>manual|9500.0|2019-03-01|2019-03-05

Unsold Cars:
TEST1|Test|Coupe||blue|hdulan|automatic|78378.0|2016-03-03|N/A
HN19PDH|Jaguar FType|Coupe||silver|none|automatic|52000.0|2018-03-01|N/A
MA57TTP|Mercedes Citan|Van|small|white|front windscreen damage|automatic|10000.0|2018-08-04|N/A
P013FPF|Vauxhall Insignia|Saloon||red|none>manual|10200.0|2018-08-10|N/A
AA13QNE|Ford Transit|Van|small|black|none|automatic|7300.0|2018-11-01|N/A
MX67TCP|Mercedes Citan|Van|small|white|front damage>manual|9500.0|2018-11-04|N/A
BA66JHH|Lexus IS|Saloon||white|indent on passenger door|automatic|18100.0|2019-01-02|N/A
P0649HH|Vauxhall Movano|Van|large|silver|none|automatic|8800.0|2019-03-11|N/A
TT66AAA|Lexus NX|SUV||dark blue|none|automatic|31000.0|2019-03-12|N/A
U05410P|Ford SMax|MPV||red|none|automatic|3800.0|2019-03-12|N/A
U05410I|Ford SMax|MPV||red|none>manual|11000.0|2019-05-14|N/A

Ref. 9

Search

Search for Cars

☒ Colour

☐ Transmission Type and Model

☐ Van Size

☐ Number of Seats

Colour

Search

Search

Search for Cars

☐ Colour

☒ Transmission Type and Model

☐ Van Size

☐ Number of Seats

Model

Transmission Type

Search

Search

Search for Cars

☐ Colour

☐ Transmission Type and Model

☒ Van Size

☐ Number of Seats

Van Size

Search

Search

Search for Cars

☐ Colour

☐ Transmission Type and Model

☐ Van Size

☒ Number of Seats

Min Number of Seats

Max Number of Seats

Search

Ref. 10

Search

Search for Cars

☒ Colour

☐ Transmission Type and Model

☐ Van Size

☐ Number of Seats

Colour

dark blue

Search

Car

Model - Lexus NX

Colour - dark blue

Registration Number - TT66AAA

Type - SUV

Van Size - N/A

Mileage - 12000

Accident History - none

Price - £31000.00

Transmission - automatic

Arrival Date - 2019-03-12

OK

Ref. 11

Sell Cars

Which car did you sell?

TT66AAA

Tick this if the car was sold today

If not, when was it sold?

Update Details