ECS506U Software Engineering

Group Project 2017

**Group 30**

**Musab Adam**

**EC15210**

**150310350**

**Parts**

**Test and Contribution Report**

**Version 1.0**

**3 March 2017**

Prof. Martin Neil

Table of Contents

[Module Design 3](#_Toc478739434)

[Contribution Report 6](#_Toc478739435)

[Testing module 6](#_Toc478739436)

[Authentication 6](#_Toc478739437)

[4.1.1(Logon as existing system administrator-provide id and password) 6](#_Toc478739438)

[4.1.2(View list of existing day-to-day users) 6](#_Toc478739439)

[4.1.3(Change password of existing day-to-day user) 6](#_Toc478739440)

[4.1.4(Create a new day-to-day user) 7](#_Toc478739441)

[4.1.5(Logout existing administrator) 7](#_Toc478739442)

[4.1.6(Login as the new day-to-day user) 7](#_Toc478739443)

[4.1.7(Logout as new day-today user) 7](#_Toc478739444)

[4.1.8(Attempt login as new day-to-day user with incorrect authentication) 7](#_Toc478739445)

[4.1.9(Logon as existing system administrator) 7](#_Toc478739446)

[4.1.10(Delete newly created day-to-day user) 7](#_Toc478739447)

[4.1.11(Login as existing day-to-day user) 8](#_Toc478739448)

[Parts 8](#_Toc478739449)

[4.5.1(View list of parts used on selected vehicle under an existing “diagnosis and repair” booking) 9](#_Toc478739450)

[4.5.2(Select a part used and list details) 9](#_Toc478739451)

[4.5.3(Search for a part to add to a vehicle by name) 9](#_Toc478739452)

[4.5.4(Add a part to a repair of the selected vehicle. Show that stock level reduced) 9](#_Toc478739453)

[4.5.5(View current stock list) 9](#_Toc478739454)

[4.5.6(Add new stock item. Show stock of part increased 9](#_Toc478739455)

[4.5.7(Edit the record for a stock item) 9](#_Toc478739456)

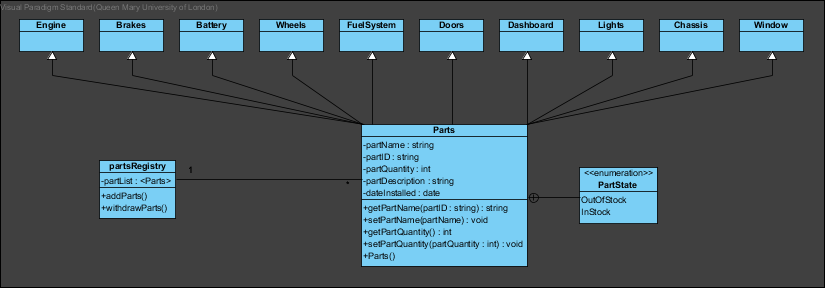
[4.5.8(Delete a stock item) 10](#_Toc478739457)

[4.5.9(Create a new part in the stock list and add items) 10](#_Toc478739458)

[4.5.10(Logout, close application and log back in. Show created record is persistently stored) 10](#_Toc478739459)

# Module Design

Module designed and built: Parts

The module is divided in to two sub modules which are parts stock and parts ordering. Initially the module was going to be in one super class that has many sub-classes, inheriting the super class with each sub-class being identified by a single stock part. However, this idea was not compatible with how we as a team wanted to have our GUI layout since it would be too much content to fit into one frame/window. Therefore, it was later separated into two classes. The “partRegistry” class was also removed in favour of SQLITE database.

OrderPartsController:

This sub-module is where the user can track/order parts for a specific booking slot.

Here the user can view customer data that currently exist in the database as well as the cars that are registered per customer and what booking slots have been assigned to the vehicle. When the user selects the customer->Vehicle->Booking slot, the program will list all the parts used/ordered during that booking slot. Here the user can add parts to the list which would update the stock to show that a part has been taken from the stock as well as adding the cost to the bill of that customer. The user can also edit parts records to allow mistakes, rectified. When the user edits the parts, the system will treat it as deleting the part from the installed parts table therefore refunding the money and then using the add parts method to add the correct part. The user can also delete the part from the vehicle which will remove the part and refund the money back to the customer as well as adding it back to the stock.

|  |  |
| --- | --- |
| refreshAll | Refreshes the Part Drop down menu to allow any part stock addition or removal |
| searchName | Calls the userListDopDown method and sending the first name as an argument |
| userListDropDown | Finds all the customer (or specified customer) and lists the customer in the customer drop down box |
| partListDropDown | Retrieves the current part stock from the database and lists in the drop-down box |
| changeBill | Retrieves the bill from the database and set the bill label with the bill from database, if the bill record does not exist, it will initiate a record in customerAccounts table |
| refreshAll(String vReg, int bID) | Displays all the parts linked to the specified Booking ID and creates instances of Parts for every record that exist under InstalledParts with that bookingID |
| addPart | Takes the specified part from the part dropdown box and adds it into the installedparts table whilst setting the data installed as today’s date and expired data as from 1 year from today. It will also update the bill of the bookingID and will reduce the stock of the specified part in the part stock |
| partWithdraw | Will reduce the specified part’s stock by one |
| partRefund | Will remove the value of the specified part from the customeraccounts bill |
| deletePart | Will remove the part from installed parts and then run the partRefund method |
| editPart | Will run the delete method and then run the add method |
| selectPart | Displays the selected part information |
| fillCustomerData | Display the selected Customer Information |
| fillVehicleData | Display the selected Vehicle Information |
| fillBookingData | Display the selected Booking Information |
| bookingItemDropDownBox | Lists all the booking slots under the specified vehicle registration |
| filterPart | Filters the part drop down menu to match the entered text |
| partDetails | Displays the part details |

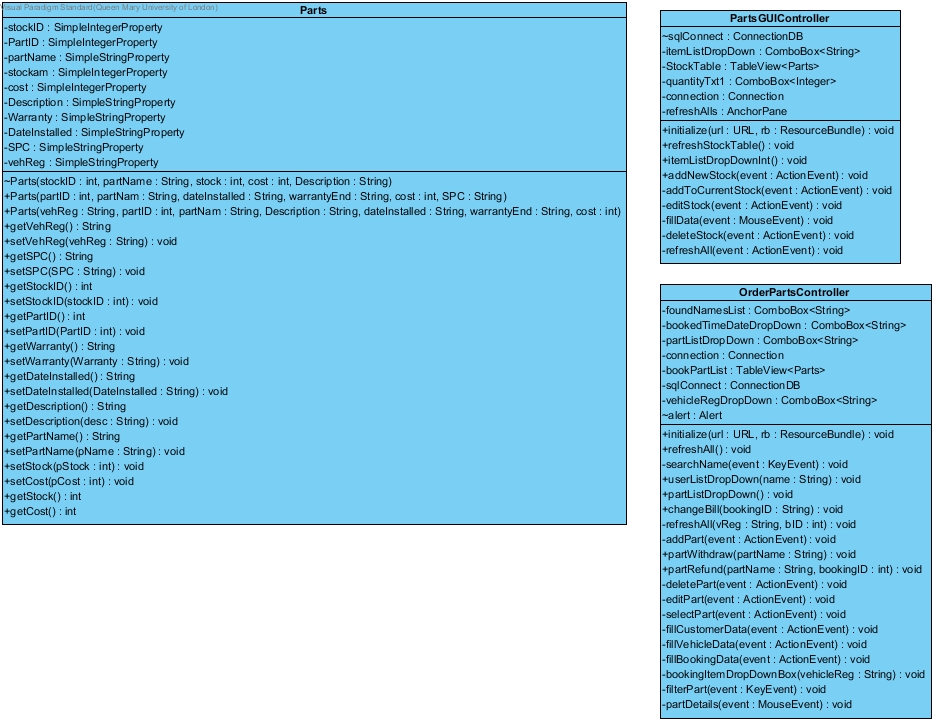
PartsGUIController:

This sub-module is where the programs keeps track on what parts are available to order, the amount of stock left and the cost of each part. In this sub-module, the user can add brand new stock items (Whilst specifying cost, stock and a short description) which would be added onto the data database(specifically the stockPart table) provided the new stock part was unique. The sub-module also allows room for incorrect inputs as an edit function has also been provided to correct any mistakes or to makes changes to the stock item (i.e.Cost). The user can also adjust the amount of stock per item with ease by simply selecting how many of the specific part they are adding to the stock.

Methods I have created

|  |  |
| --- | --- |
| refreshStockTable | Displays all the parts stock that currently exists in the partsStock table |
| itemListDropDownInt | Lists all the parts in the partsStock under the dropdown menu |
| addNewStock | Adds a brand-new stock item into the partStock table |
| addToCurrentStock | Updates the quantity of the specified part |
| editStock | Updates the data of the specified part |
| fillData | Displays the parts data of the selected part |
| deleteStock | Removes the part stock from the PartsStock table |
| refreshAll | Refreshes the table view of the display of parts stock |

The Classes do not directly communicate with each other, instead they communicate through the same database. This way both sub classes are always looking at the same database data. Each class will have a method that will link to a database class which returns the connection. This allows everyone to use the same type of connection (same database location) and therefore making it safer to integrate because if there were changes needed to be made to the location of the database, all that would need to be done is change one line of code in the ConnectionDB class. However, both the PartsGUIController and OrderPartsController create instances of Parts which will hold the data from the database in the instance.



# Testing module

## Authentication

### 4.1.1(Logon as existing system administrator-provide id and password)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Passed

### 4.1.2(View list of existing day-to-day users)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Step 4. Click on the Admin tab which is located on the top of the UI

Passed

### 4.1.3(Change password of existing day-to-day user)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Step 4. Click on the Admin tab which is located on the top of the UI

Step 5. Select the user “test test” in the table by clicking on the cell containing “test”

Step 6. Enter the current password “password” in the password text box

Step 7. Click on “Change Password” button

Step 8. Enter the new password “abc123” and click ok

Passed

### 4.1.4(Create a new day-to-day user)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Step 4. Click on the Admin tab which is located on the top of the UI

Step 5. Enter the name “tom” into the first name text field

Step 6. Enter the surname “harry” into the last name text field

Step 7. Enter the password “abc123” into the password text field

Step 8. Click on the add button

Step 9. Note the UserID that has been generated in the table view

Passed

### 4.1.5(Logout existing administrator)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Step 4. Click on Log out, located top-left

Passed

### 4.1.6(Login as the new day-to-day user)

Step 1. Enter “10008” (or the userID that you noted) into the user id textbox

Step 2. Enter “abc123” into the Password textbox

Step 3. Click on Log in

Passed

### 4.1.7(Logout as new day-today user)

Step 1. Enter “10008” (or the userID that you noted) into the user id textbox

Step 2. Enter “abc123” into the Password textbox

Step 3. Click on Log in

Step 4. Click on Log out, located top-left

Passed

### 4.1.8(Attempt login as new day-to-day user with incorrect authentication)

Step 1. Enter “10008” (or the userID that you noted) into the user id textbox

Step 2. Enter “83899” into the Password textbox

Step 3. Click on Log in

Passed

### 4.1.9(Logon as existing system administrator)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in button

Passed

### 4.1.10(Delete newly created day-to-day user)

Step 1. Enter “10000” into the user id textbox

Step 2. Enter “newblarg” into the Password textbox

Step 3. Click on Log in

Step 4. Click on the Admin tab which is located on the top of the UI

Step 5. Select the user “Tom Harry” in the table by clicking on the cell containing “Tom”

Step 6. Click on the delete button

Passed

### 4.1.11(Login as existing day-to-day user)

Step 1. Enter “10008” into the user id textbox

Step 2. Enter “password” into the Password textbox

Step 3. Click on Log in

Passed-The user “Test” was displayed near the log out button and was identified as a system user

## Parts

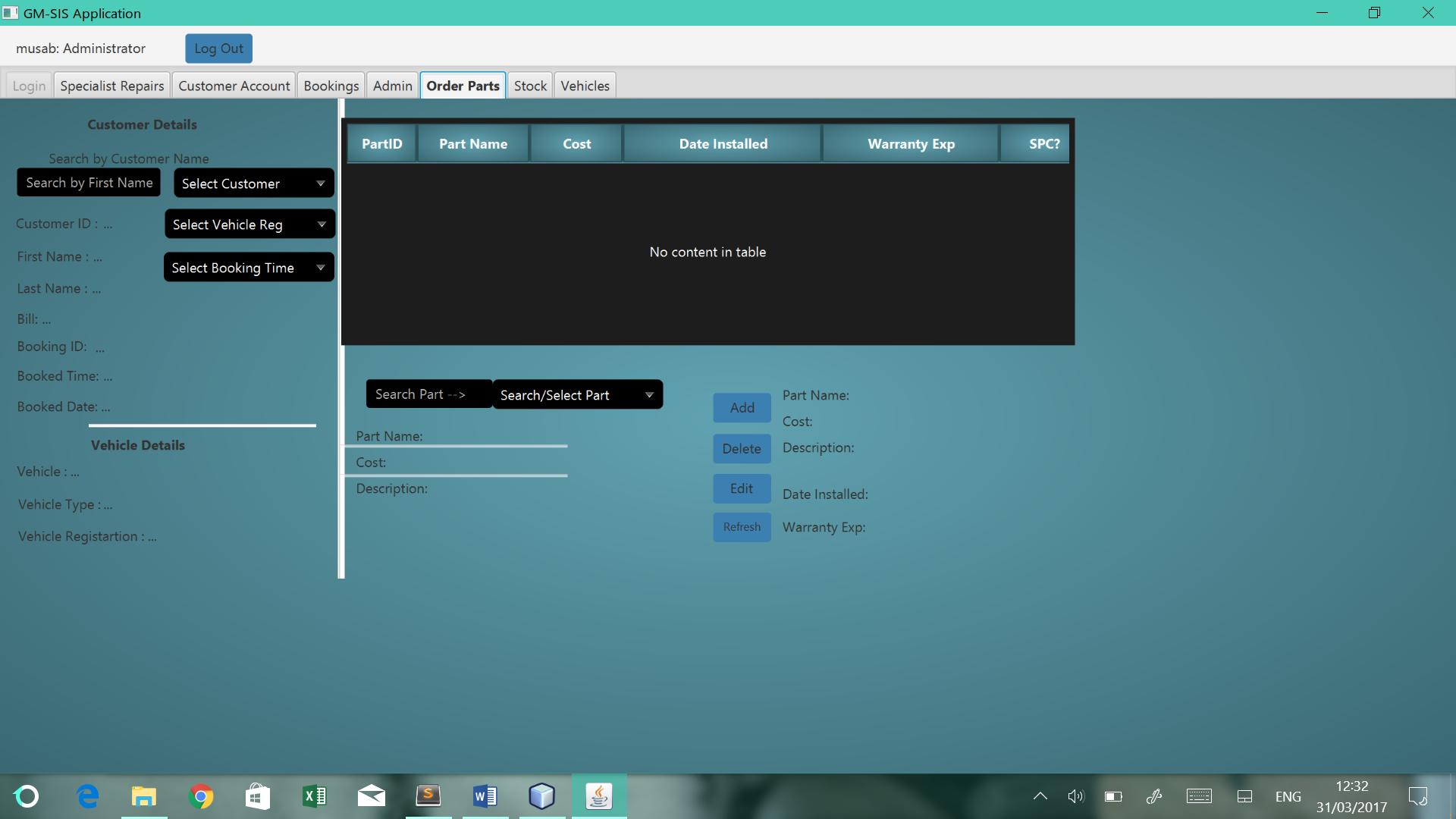


Table that will display the parts installed on a specific booking. Selecting the row will allow you to edit/delete

When a record is selected, the part details is listed here

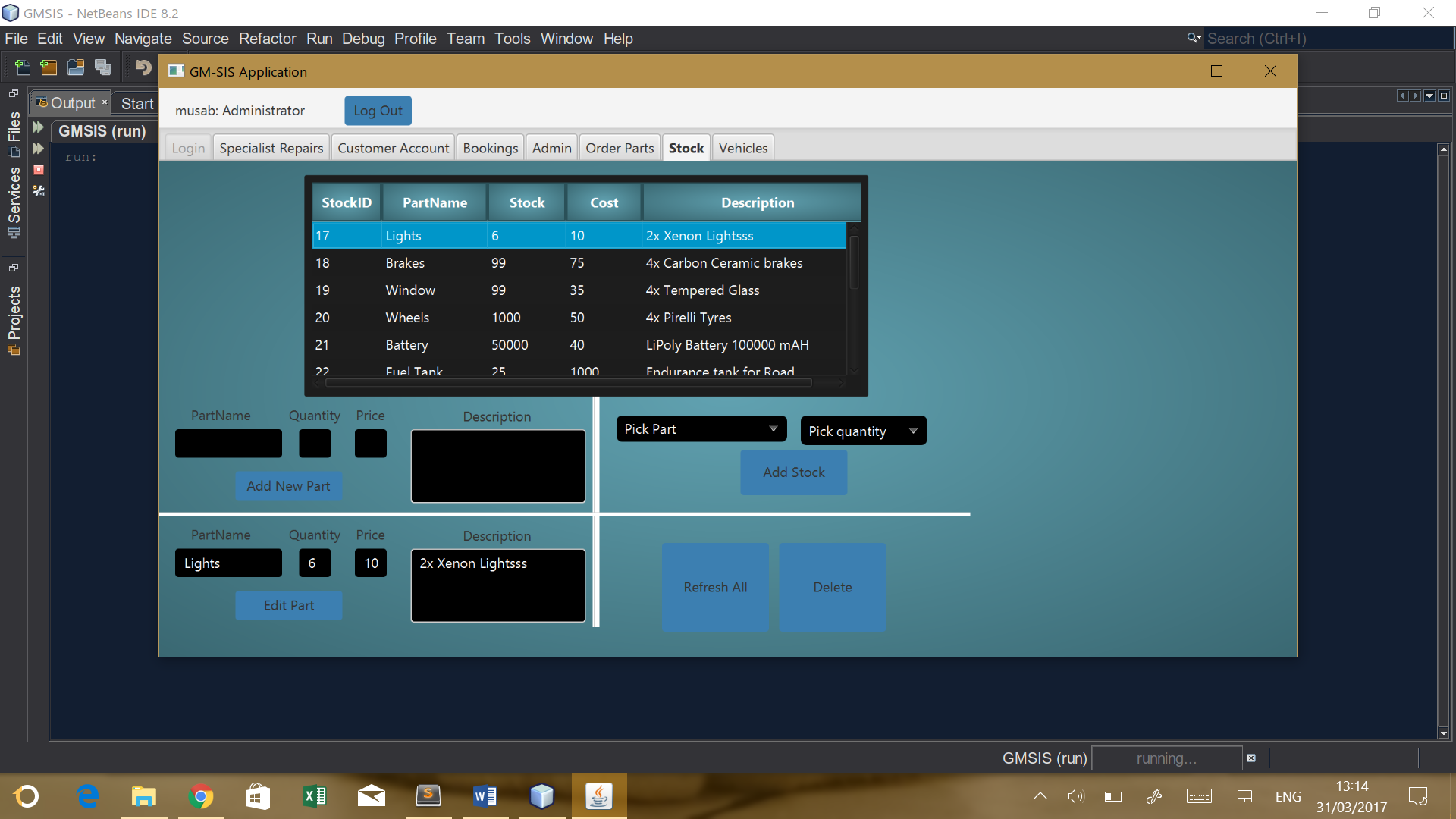
**Add** button that will install the selected part from the drop down,

**Delete** button will delete the selected record

**Edit** will replace the selected record with the part selected from dropdown

**Refresh** button will refresh all data to accommodate any changes to the database file

Select the customer then select the vehicle then select the booking slot



**Table view** of the current stock in the garage.

Records are **selectable** and once selected, can be **edited** or **deleted**

Pick part from drop down box and then select value from **quantity drop down box**, then hit the **add stock button**

Enter all the **fields** then hit **add new part** to enter a new part

**Select record from table**, which will automatically fill in the **fields**, edit what you wish and then hit **edit part button**

Assuming user is logged in as a system user

### 4.5.1(View list of parts used on selected vehicle under an existing “diagnosis and repair” booking)

Step 1. Click on Order Parts tab

Step 2. Click on the Customer Drop down menu and select “Test-Test”

Step 3. Click on the Vehicle Drop down menu and select “RU20 VEG”

Step 4. Click on the Booking Time Drop down menu and select “11:00…”

Passed

### 4.5.2(Select a part used and list details)

Step 1. Click on Order Parts tab

Step 2. Click on the Customer Drop down menu and select “Test-Test”

Step 3. Click on the Vehicle Drop down menu and select “RU20 VEG”

Step 4. Click on the Booking Time Drop down menu and select “11:00…”

Step 5. Click on the record that has a cell content of “Lights”

Passed

### 4.5.3(Search for a part to add to a vehicle by name)

Step 1. Click on Order Parts tab

Step 2. Click on the Customer Drop down menu and select “Test-Test”

Step 3. Click on the Vehicle Drop down menu and select “RU20 VEG”

Step 4. Click on the Booking Time Drop down menu and select “11:00…”

Step 5. Type “Ligh” into the search part text box

Step.6. Select Lights from parts drop down box

Step 7. Click on add button

Passed

### 4.5.4(Add a part to a repair of the selected vehicle. Show that stock level reduced)

Step 1. Click on Order Parts tab

Step 2. Click on the Customer Drop down menu and select “Test-Test”

Step 3. Click on the Vehicle Drop down menu and select “RU20 VEG”

Step 4. Click on the Booking Time Drop down menu and select “11:00…”

Step 5. Type “Ligh” into the search part text box

Step.6. Select Lights from parts drop down box

Step 7. Click on add button

Step 8. Click on Stock Tab

Step 9. Click on Refresh All button

Passed

### 4.5.5(View current stock list)

Step 1. Click on Stock Tab

Passed

### 4.5.6(Add new stock item. Show stock of part increased

Step 1. Click on Stock Tab

Step 2. Select part “Lights” from part drop down menu

Step 3. Select quantity “10” from quantity drop down box

Step 4. Click on add stock button

Passed

### 4.5.7(Edit the record for a stock item)

Step 1. Click on Stock tab

Step 2. Click on the record that has a cell content of “Lights”

Step 3. Change the cost in the cost text field (above edit button) to 5

Step 4. Click on edit button

Passed

### 4.5.8(Delete a stock item)

Step 1. Click on Stock tab

Step 2. Click on the record that has a cell content of “Lights”

Step 3. Click delete button

Passed

### 4.5.9(Create a new part in the stock list and add items)

Step 1. Click on Stock tab

Step 2. Enter “Lights” into the part name text box (above the add button)

Step 3. Enter “10” into the quantity text box (above the add button)

Step 4. Enter “50” into the price text box (above the add button)

Step 5. Enter “Xenon x2 Lights” into the Description text box (above the add button)

Passed

### 4.5.10(Logout, close application and log back in. Show created record is persistently stored)

Step 1. Click on Log out button

Step 2. Enter “10000” into the user id textbox

Step 3. Enter “newblarg” into the Password textbox

Step 4. Click on Log in button

Step 5. Click on Stock tab

Step 6. See that the record “light” exist

Passed