# TEST SUITE

**for**

# Zulu - A Motor Part Shop Software

**Prepared by -**

**Ashutosh Kumar Singh (19CS30008)**

**Vanshita Garg (19CS10064)**

**Suhas Jain (19CS30048)**

**Indian Institute of Technology, Kharagpur**

**March 26, 2021**

# Contents

# 1 Introduction and Important Points

This is the Test Suite document for Zulu - a Motor Part Shop Software. Here, we list down test cases along with their expected golden output for each scenario described in sections 5 and 6 of the Test Plan document.

**Note :** At all places in this document, the numbers written in parentheses after the headings of a section or subsection denote the corresponding section number in the Test Plan Document.

# 2 Test Cases for the Frontend GUI Interface

## 2.1 Home Page (5.1)

This page cannot be tested because this is just a intermediate page to facilitate the loading of the software which redirects to the login page automatically. There are no functionalities present that can be tested here.

## 2.2 Login Page (5.2)

1. *Both Username and Password are correct*

   **INPUT :**
   Username - VASachcha
   Password - OkZulu!

   **GOLDEN OUTPUT / RESPONSE :**
   The owner successfully logs in to the system and the Dashboard is displayed.

2. *Username is correct but Password is incorrect*

   **INPUT :**
   Username - VASachcha
   Password - HiZulu:)

   **GOLDEN OUTPUT / RESPONSE :**
   The log in attempt is unsuccessful and an appropriate message is displayed in a new popup window.

3. *Username is incorrect but Password is correct*

   **INPUT :**
   Username - VASbura
   Password - OkZulu!

   **GOLDEN OUTPUT / RESPONSE :**
   The log in attempt is unsuccessful and an appropriate message is displayed in a new popup window.

4. *Both Username and Password are incorrect*

   **INPUT :**
   Username - VASbura
   Password - HiZulu:)

   **GOLDEN OUTPUT / RESPONSE :**
   The log in attempt is unsuccessful and an appropriate message is displayed in a new popup window.

## 2.3 Dashboard (5.3)

1. *Working of all buttons*

   **INPUT ACTION :**
   Click on all the buttons - Add an Item, Remove an Item, Report Sale, View Inventory, End Day, View Graph and Back, one by one.

   **GOLDEN OUTPUT / RESPONSE :**
   Each click should redirect to the appropriate window.

## 2.4 Add an Item (5.4)

**PRE - CONDITION :**
There are the following items in the inventory :

- Item Type - Tyre
  Manufacturer Name - MRF
  Vehicle Type - Motorbike
- Item Type - Windshield
  Manufacturer Name - 3M
  Vehicle Type - Sedan
- Item Type - Tyre
  Manufacturer Name - Michelin
  Vehicle Type - Sedan

1. *Working of all drop-down menus*

   **INPUT ACTION :**
   Look at the options available in the drop-down menus for Item Type, Manufacturer Name and Vehicle Type.

   **GOLDEN OUTPUT / RESPONSE :**
   The Item Type drop-down menu will have the following option(s) :

   - Tyre
   - Windshield

The Manufacturer Name drop-down menu will have the following option(s) :

- MRF
- 3M
- Michelin

The Vehicle Type drop-down menu will have the following option(s) :

- Motorbike
- Sedan

2. *Working of all text fields*

   **INPUT :**
   Item Type - Bumper
   Manufacturer Name - Tata
   Manufacturer Address - 7th Floor, Amar Business Park, Andheri, Mumbai
   Vehicle Type - SUV
   Price - 26850
   Initial Quantity - 4

   **GOLDEN OUTPUT / RESPONSE :**
   Whatever has been typed should be visible in the text boxes.

3. *Working of all buttons*

   **INPUT ACTION :**
   Click on the buttons - Add and Back, one by one.

   **GOLDEN OUTPUT / RESPONSE :**
   Each click should redirect to the appropriate window, and/or popup an appropriate message.

4. *The data entered in all the fields is valid / correct*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
   Vehicle Type - Motorbike
   Price - 3700
   Initial Quantity - 6

   **GOLDEN OUTPUT / RESPONSE :**
   This new item should be added in the inventory. This can be verified by using the View Inventory option available on the Dashboard.

5. *Item Type entered is invalid*

   **INPUT :**
   Item Type - T$*yre
   Manufacturer Name - MRF
   Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai

Vehicle Type - Motorbike
Price - 3700
Initial Quantity - 6

**GOLDEN OUTPUT / RESPONSE :**
An error message should be displayed telling that the format of the entered Item Type is invalid.

6. *Manufacturer Name entered is invalid*

**INPUT :**
Item Type - Tyre
Manufacturer Name - M*@RF
Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
Vehicle Type - Motorbike
Price - 3700
Initial Quantity - 6

**GOLDEN OUTPUT / RESPONSE :**
An error message should be displayed telling that the format of the entered Manufacturer Name is invalid.

7. *Manufacturer Address is invalid*

**INPUT :**
Item Type - Tyre
Manufacturer Name - MRF
Manufacturer Address - Devara;j Build@ing, Goregaon$ West, Mumbai
Vehicle Type - Motorbike
Price - 3700
Initial Quantity - 6

**GOLDEN OUTPUT / RESPONSE :**
An error message should be displayed telling that the format of the entered Manufacturer Address is invalid.

8. *Vehicle Type entered is invalid*

**INPUT :**
Item Type - Tyre
Manufacturer Name - MRF
Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
Vehicle Type - M%oto=*rbike
Price - 3700
Initial Quantity - 6

**GOLDEN OUTPUT / RESPONSE :**
An error message should be displayed telling that the format of the entered Vehicle Type is invalid.

9. *Price entered is not a number*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
   Vehicle Type - Motorbike
   Price - 3700xyz
   Initial Quantity - 6

   **GOLDEN OUTPUT / RESPONSE :**
   An error message should be displayed telling that the entered Price is not a number.

10. *Price of the item entered is zero*

    **INPUT :**
    Item Type - Tyre
    Manufacturer Name - MRF
    Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
    Vehicle Type - Motorbike
    Price - 0
    Initial Quantity - 6

    **GOLDEN OUTPUT / RESPONSE :**
    An error message should be displayed telling that the entered Price is zero.

11. *Price of the item entered is negative*

    **INPUT :**
    Item Type - Tyre
    Manufacturer Name - MRF
    Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
    Vehicle Type - Motorbike
    Price - -3700
    Initial Quantity - 6

    **GOLDEN OUTPUT / RESPONSE :**
    An error message should be displayed telling that the entered Price is negative.

12. *Initial Quantity entered is not a number*

    **INPUT :**
    Item Type - Tyre
    Manufacturer Name - MRF
    Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
    Vehicle Type - Motorbike
    Price - 3700
    Initial Quantity - 6abc

    **GOLDEN OUTPUT / RESPONSE :**
    An error message should be displayed telling that the entered Initial Quantity is not a number.

13. *Initial Quantity entered is zero*

    **INPUT :**
    Item Type - Tyre
    Manufacturer Name - MRF
    Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
    Vehicle Type - Motorbike
    Price - 3700
    Initial Quantity - 0

    **GOLDEN OUTPUT / RESPONSE :**
    An error message should be displayed telling that the entered Initial Quantity is zero.

14. *Initial Quantity entered is negative*

    **INPUT :**
    Item Type - Tyre
    Manufacturer Name - MRF
    Manufacturer Address - 9, Devaraj Building, Goregaon West, Mumbai
    Vehicle Type - Motorbike
    Price - 3700
    Initial Quantity - -6

    **GOLDEN OUTPUT / RESPONSE :**
    An error message should be displayed telling that the entered Initial Quantity is negative.

## 2.5 Remove an Item (5.5)

**PRE - CONDITION :**
There are the following items in the inventory :

- Item Type - Tyre
  Manufacturer Name - MRF
  Vehicle Type - Motorbike
- Item Type - Windshield
  Manufacturer Name - 3M
  Vehicle Type - Sedan
- Item Type - Tyre
  Manufacturer Name - Michelin
  Vehicle Type - Sedan

1. **INPUT ACTION :**
   Look at the options available in the drop-down menus for Item Type, Manufacturer Name and Vehicle Type.

**GOLDEN OUTPUT / RESPONSE :**
The Item Type drop-down menu will have the following option(s) :

- Tyre
- Windshield

Suppose now we select Tyre. The Manufacturer Name drop-down menu will now have the following option(s) :

- MRF
- Michelin

Suppose now we select MRF The Vehicle Type drop-down menu will now have the following option(s) :

- Motorbike

2. *Working of all buttons*

**INPUT ACTION :**
Click on the buttons - Remove and Back, one by one.

**GOLDEN OUTPUT / RESPONSE :**
Each click should redirect to the appropriate window, and/or popup an appropriate message.

3. *All the fields chosen - Item Type, Manufacturer Name and Vehicle Type are valid*

**INPUT :**
The selection is performed similar to that described in item 1 of *section 2.5* of this document.
After the series of selections we get :
Item Type - Tyre
Manufacturer Name - MRF
Vehicle Type - Motorbike
Then the Remove button is clicked.

**GOLDEN OUTPUT / RESPONSE :**
This item should be removed from the inventory. This can be verified by using the View Inventory option available on the Dashboard.

## 2.6 Report a Sale (5.6)

**PRE - CONDITION :**
There are the following items in the inventory :

- Item Type - Tyre
  Manufacturer Name - MRF
  Vehicle Type - Motorbike
  Quantity - 5
- Item Type - Windshield
  Manufacturer Name - 3M
  Vehicle Type - Sedan
  Quantity - 3

- Item Type - Tyre
  Manufacturer Name - Michelin
  Vehicle Type - Sedan
  Quantity - 6

1. *Working of all drop-down menus*

   Look at the options available in the drop-down menus for Item Type, Manufacturer Name and Vehicle Type.

   **GOLDEN OUTPUT / RESPONSE :**
   The Item Type drop-down menu will have the following option(s) :

   - Tyre
   - Windshield

   Suppose now we select Tyre. The Manufacturer Name drop-down menu will now have the following option(s) :

   - MRF
   - Michelin

   Suppose now we select MRF The Vehicle Type drop-down menu will now have the following option(s) :

   - Motorbike

2. *Working of all text fields*

   **INPUT :**
   Quantity - 4

   **GOLDEN OUTPUT / RESPONSE :**
   Whatever has been typed should be visible in the text box.

3. *Working of all buttons*

   **INPUT ACTION :**
   Click on the buttons - Report Sale and Back, one by one.

   **GOLDEN OUTPUT / RESPONSE :**
   Each click should redirect to the appropriate window, and/or popup an appropriate message.

4. *All the fields chosen and data entered are valid*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Vehicle Type - Motorbike
   Quantity - 4

   **GOLDEN OUTPUT / RESPONSE :**
   The sale should be completed, a success message should be displayed and the quantity of the item in the inventory should be appropriately adjusted.

5. *Quantity entered is not a number*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Vehicle Type - Motorbike
   Quantity - 4efg

   **GOLDEN OUTPUT / RESPONSE :**
   An error message should be displayed saying that the Quantity entered is not a number.

6. *Quantity entered is zero*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Vehicle Type - Motorbike
   Quantity - 0

   **GOLDEN OUTPUT / RESPONSE :**
   An error message should be displayed saying that the Quantity entered is zero.

7. *Quantity entered is negative*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Vehicle Type - Motorbike
   Quantity - -2

   **GOLDEN OUTPUT / RESPONSE :**
   An error message should be displayed saying that the Quantity entered is negative.

8. *Quantity entered is greater than the quantity in the inventory*

   **INPUT :**
   Item Type - Tyre
   Manufacturer Name - MRF
   Vehicle Type - Motorbike
   Quantity - 10

   **GOLDEN OUTPUT / RESPONSE :**
   An error message should be displayed saying that the Quantity entered is greater than the Quantity of the item in the inventory.

## 2.7 View Inventory (5.7)

1. *Working of the scrollable list of items*

   **INPUT ACTION :**
   Navigate down the list using the mouse wheel.

**GOLDEN OUTPUT / RESPONSE :**
A list of all the items in the inventory should be displayed in the form of a scrollable table / list. The list should show all the details for each item, like unique ID, Item Type, Manufacturer Name, Vehicle Type, Quantity, Price and In Stock or Not In Stock.

2. *Working of all buttons*

**INPUT ACTION :**
Click on the Back button.

**GOLDEN OUTPUT / RESPONSE :**
The click should redirect to the appropriate window.

## 2.8 Day - End Tasks (5.8)

1. *Computation of number of items to be ordered at the end of a day*

**INPUT ACTION :**
First, add a new item :
Item Type - Bumper
Manufacturer Name - Tata
Manufacturer Address - 7th Floor, Amar Business Park, Andheri, Mumbai
Vehicle Type - SUV
Price - 26850
Initial Quantity - 4

On Day 1, Report Sale :
Quantity - 2

On Day 2, Report Sale :
Quantity - 12

**GOLDEN OUTPUT :**
At the end of Day 1 :
Quantity to be Ordered - 12

At the end of Day 2 :
Quantity to be Ordered - 47

2. *Working of the generated order list, which will be a scrollable list*

**INPUT ACTION :**
Click on the appropriate option to generate the order list.

**GOLDEN OUTPUT / RESPONSE :**
A list of all the items to be ordered is generated. The list shows the unique ID, Item Type, Manufacturer Name, Manufacturer Address, Vehicle Type and Quantity to be ordered for each item.

3. *Working of all buttons*

   **INPUT ACTION :**
   Click on the View Daily Revenue and Back buttons one by one.

   **GOLDEN OUTPUT / RESPONSE :**
   Each click should redirect to an appropriate window.

## 2.9 Graph for Daily Sales of a Month (5.9)

1. *View the graph before the first month has ended*

   **INPUT ACTION :**
   Select the View Graph option.

   **GOLDEN OUTPUT / RESPONSE :**
   A message should be displayed saying that the first month has not yet been completed.

2. *View the graph on the day a month has ended*

   **INPUT ACTION :**
   Select the View Graph option.

   **GOLDEN OUTPUT / RESPONSE :**
   A graph showing the daily sales for the month which has just been completed should be displayed.

3. *View the graph in the middle of a month*

   **INPUT ACTION :**
   Select the View Graph option.

   **GOLDEN OUTPUT / RESPONSE :**
   A graph showing the daily sales for the previous completed month should be displayed.

# 3 Test Cases for Backend Classes and Database Management

## 3.1 Testing the `Owner` class (6.1)

### 3.1.1 Testing the `getName()` function (6.1.1)

1. *Retrieve and verify the* `name` *of the owner*

   **INPUT ACTION :**
   Call the `getName()` function.

   **GOLDEN OUTPUT :**
   `name` = Zulu Malik

#### 3.1.2 Testing the `setName(String name)` function (6.1.2)

1. *Set the* `name` *of the owner to a valid string*

   **INPUT :**
   `name` = Lulu Malik

   **GOLDEN OUTPUT :**
   No output as such, just verify that the `name` has changed.

2. *Set the* `name` *of the owner to an invalid string*

   **INPUT :**
   `name` = Lulu#@*Malik

   **GOLDEN OUTPUT :**
   An error message should be displayed.

#### 3.1.3 Testing the `getUsername()` function (6.1.3)

1. *Retrieve and verify the* `username` *of the owner*

   **INPUT ACTION :**
   Call the `getUsername()` function.

   **GOLDEN OUTPUT :**
   `username` = VASachcha

#### 3.1.4 Testing the `setUsername(String username)` function (6.1.4)

1. *Set the* `username` *of the owner to a valid string*

   **INPUT :**
   `username` = VASbura

   **GOLDEN OUTPUT :**
   No output as such, just verify that the `username` has changed.

2. *Set the* `username` *of the owner to an invalid string*

   **INPUT :**
   `name` = VAS*@#achcha

   **GOLDEN OUTPUT :**
   An error message should be displayed.

#### 3.1.5 Testing the `validate(String username, String password)` function (6.1.5)

1. *Both the* `username` *and* `password` *passed are the same as the actual* `username` *and* `password` *of the owner*

**INPUT :**
`username` = VASachcha
`password` = OkZulu!

**GOLDEN OUTPUT :**
true

2. `username` *is correct but* `password` *is incorrect*

   **INPUT :**
   `username` = VASachcha
   `password` = HiZulu:)

   **GOLDEN OUTPUT :**
   false

3. `username` *is incorrect but* `password` *is correct*

   **INPUT :**
   `username` = VASbura
   `password` = OkZulu!

   **GOLDEN OUTPUT / RESPONSE :**
   false

4. *Both* `username` *and* `password` *are incorrect*

   **INPUT :**
   `username` = VASbura
   `password` = HiZulu:)

   **GOLDEN OUTPUT / RESPONSE :**
   false

## 3.2 Testing the `Item` class (6.2)

### 3.2.1 Testing the Constructor `Item(String type, double price, int quantity, int manufacturerID, String vehicleType, Date startDate)` (6.2.1)

The constructor is called only after ensuring that all the parameters passed are valid. So, there is no need to test the constructor here.

**PRE - CONDITION for section 3.2.2 to 3.2.8 :**
Consider that no `Item` object has been created till now and suppose we create the following objects on the date 2021-04-03 (yyyy-mm-dd).
Create an `Item` object with the following attributes :
`type` = Suspension
`price` = 5400
`quantity` = 4
`manufacturerID` = 1
`vehicleType` = Hatchback

Now, create another object with the following attributes :
`type` = Mirror
`price` = 3600
`quantity` = 5
`manufacturerID` = 1
`vehicleType` = Minivan

### 3.2.2 Testing the `getUID()` function (6.2.2)

1. *Retrieve and verify the* `uID` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first and second object one by one.

   **GOLDEN OUTPUT :**
   `uID` = 1 (for the first object)
   `uID` = 2 (for the second object)

### 3.2.3 Testing the `getType()` function (6.2.3)

1. *Retrieve and verify the* `type` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first object.

   **GOLDEN OUTPUT :**
   `type` = Suspension

### 3.2.4 Testing the `getPrice()` function (6.2.4)

1. *Retrieve and verify the* `price` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first object.

   **GOLDEN OUTPUT :**
   `price` = 5400

### 3.2.5 Testing the `getQuantity()` function (6.2.5)

1. *Retrieve and verify the* `quantity` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first object.

   **GOLDEN OUTPUT :**
   `quantity` = 4

### 3.2.6 Testing the `getManufacturerID()` function (6.2.6)

1. *Retrieve and verify the* `manufacturerID` *of an* `Item` *object.*

**INPUT ACTION :**
Call the function on the first object.

**GOLDEN OUTPUT :**
`manufacturerID` = 1

### 3.2.7 Testing the `getVehicleType()` function (6.2.7)

1. *Retrieve and verify the* `vehicleType` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first object.

   **GOLDEN OUTPUT :**
   `vehicleType` = Hatchback

### 3.2.8 Testing the `getStartDate()` function (6.2.8)

1. *Retrieve and verify the* `startDate` *of an* `Item` *object.*

   **INPUT ACTION :**
   Call the function on the first object.

   **GOLDEN OUTPUT :**
   `startDate` = 2021-04-03

### 3.2.9 Testing the `save()` function (6.2.9)

1. *Insert a new item to the database when it is empty*

   **PRE CONDITION :**
   No item is present in the database.

   **INPUT :**
   Create a new object by calling the constructor with the following attributes, the constructor itself calls the `save()` method :
   `type` = Suspension
   `price` = 5400
   `quantity` = 4
   `manufacturerID` = 1
   `vehicleType` = Hatchback

   **GOLDEN OUTPUT / RESPONSE :**
   The item should be added to the database. It can be verified by querying the database to search for the item.

2. *Insert a new item to the database when it is not empty*

   **PRE CONDITION :**
   Ensure that one or more items are already present in the database.

**INPUT :**
Now, create a new object with the following attributes, the constructor itself calls the `save()` method :
`type` = Mirror
`price` = 3600
`quantity` = 5
`manufacturerID` = 1
`vehicleType` = Minivan

**GOLDEN OUTPUT / RESPONSE :**
The item should be added to the database. It can be verified by querying the database to search for the item.

### 3.2.10 Testing the `delete()` function (6.2.10)

1. *Delete a item when multiple items are present in the inventory database*

   **PRE CONDITION :**
   The following items are present in the database :

   - `uID` = 1
     `type` = Suspension
     `price` = 5400
     `quantity` = 4
     `manufacturerID` = 1
     `vehicleType` = Hatchback
   - `uID` = 2
     `type` = Mirror
     `price` = 3600
     `quantity` = 5
     `manufacturerID` = 1
     `vehicleType` = Minivan

   **INPUT ACTION :**
   Call the `delete()` method for the item with `uID` 2.

   **GOLDEN OUTPUT / RESPONSE :**
   The item should be deleted from the database. It can be verified by querying the database to see if it is present or not.

2. *Delete a item when only a single item is present in the inventory database*

   **PRE CONDITION :**
   Only a single item should be present in the database :

   - `uID` = 1
     `type` = Suspension
     `price` = 5400
     `quantity` = 4
     `manufacturerID` = 1
     `vehicleType` = Hatchback

**INPUT ACTION :**
Call the `delete()` method for the item with `uID` 1.

**GOLDEN OUTPUT / RESPONSE :**
The item should be deleted from the database, and so the database will not have any more items.

#### 3.2.11 Testing the `updateSale(int numSold)` function (6.2.11)

1. *Update the quantity of an item being sold*

   **PRE CONDITION :**
   Suppose the state of an item initially is as follows :
   `uID` = 1
   `type` = Suspension
   `price` = 5400
   `quantity` = 4
   `manufacturerID` = 1
   `vehicleType` = Hatchback

   **INPUT ACTION :**
   Call the `updateSale()` method for this item with `numSold` = 3.

   **GOLDEN OUTPUT / RESPONSE :**
   Now, in the new state, the item object will have `quantity` = 1.

## 3.3 Testing the `Manufacturer` class (6.3)

#### 3.3.1 Testing the Constructor `Manufacturer(String name, String address)` (6.3.1)

The constructor is called only after ensuring that all the parameters passed are valid. So, they will be tested either in the GUI testing or in the database testing. So, there is no need to test the constructor here.

**PRE - CONDITION for 3.3.2 to 3.3.5 :**
Suppose a `Manufacturer` has been added with the following attributes :
`uID` = 1
`name` = MRF
`address` = 9, Devaraj Building, Goregaon West, Mumbai
`itemCount` = 2

#### 3.3.2 Testing the `getUID()` function (6.3.2)

1. *Retrieve and verify the* `uID` *of a* `Manufacturer` *object.*

   **INPUT ACTION :**
   Call the function on the `Manufacturer` object.

   **GOLDEN OUTPUT :**
   `uID` = 1

### 3.3.3 Testing the `getName()` function (6.3.3)

1. *Retrieve and verify the* `name` *of a* `Manufacturer` *object.*

   **INPUT ACTION :**
   Call the function on the `Manufacturer` object.

   **GOLDEN OUTPUT :**
   `name` = MRF

### 3.3.4 Testing the `getAddress()` function (6.3.4)

1. *Retrieve and verify the* `address` *of a* `Manufacturer` *object.*

   **INPUT ACTION :**
   Call the function on the `Manufacturer` object.

   **GOLDEN OUTPUT :**
   `address` = 9, Devaraj Building, Goregaon West, Mumbai

### 3.3.5 Testing the `getItemCount()` function (6.3.5)

1. *Retrieve and verify the* `itemCount` *of a* `Manufacturer` *object.*

   **INPUT ACTION :**
   Call the function on the `Manufacturer` object.

   **GOLDEN OUTPUT :**
   `itemCount` = 2

### 3.3.6 Testing the `save()` function (6.3.6)

1. *Insert a new manufacturer to the database when it is empty*

   **PRE CONDITION :**
   No manufacturer is yet present in the database.

   **INPUT :**
   Create a new `Manufacturer` object by calling the constructor with the following attributes, the constructor itself calls the `save()` method :
   `name` = MRF
   `address` = 9, Devaraj Building, Goregaon West, Mumbai
   `itemCount` = 2

   **GOLDEN OUTPUT / RESPONSE :**
   The manufacturer should be added to the database. It can be verified by querying the database to search for the manufacturer.

2. *Insert a new manufacturer to the database when it is not empty*

   **PRE CONDITION :**
   Ensure that one or more manufacturers are already present in the database.

**INPUT :**
Now, create a new `Manufacturer` object with the following attributes, the constructor itself calls the `save()` method :
`name` = Tata
`address` = 7th Floor, Amar Business Park, Andheri, Mumbai
`itemCount` = 4

**GOLDEN OUTPUT / RESPONSE :**
The manufacturer should be added to the database. It can be verified by querying the database to search for the manufacturer.

### 3.3.7 Testing the `delete()` function (6.3.7)

1. *Delete a manufacturer when multiple manufacturers are present in the inventory database*

   **PRE CONDITION :**
   The following manufacturers are present in the database :

   - `uID` = 1
     `name` = MRF
     `address` = 9, Devaraj Building, Goregaon West, Mumbai
     `itemCount` = 2
   - `uID` = 2
     `name` = Tata
     `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
     `itemCount` = 4

   **INPUT ACTION :**
   Call the `delete()` method for the manufacturer with `uID` 2.

   **GOLDEN OUTPUT / RESPONSE :**
   The manufacturer should be deleted from the database. It can be verified by querying the database to see if it is present or not.

2. *Delete a manufacturer when only a single manufacturer is present in the inventory database*

   **PRE CONDITION :**
   Only a single item should be present in the database :

   - `uID` = 1
     `name` = MRF
     `address` = 9, Devaraj Building, Goregaon West, Mumbai
     `itemCount` = 2

   **INPUT ACTION :**
   Call the `delete()` method for the manufacturer with `uID` 1.

   **GOLDEN OUTPUT / RESPONSE :**
   The manufacturer should be deleted from the database, and so now the database will not contain any manufacturers.

## 3.4 Testing the `Inventory` class (6.4)

### 3.4.1 Testing the `retrieveData()` function (6.4.1)

1. *The inventory database is empty*

   **INPUT ACTION :**
   Call the `retrieveData()` method.

   **GOLDEN OUTPUT / RESPONSE :**
   The HashMaps `searchMap`, `itemsList` and `manufacturersList` should be empty.

2. *The inventory database is not empty*

   **PRE CONDITION :**
   List of items present in the database :

   - `uID` = 1
     `type` = Suspension
     `price` = 5400
     `quantity` = 4
     `manufacturerID` = 1
     `vehicleType` = Hatchback
   - `uID` = 2
     `type` = Mirror
     `price` = 3600
     `quantity` = 5
     `manufacturerID` = 1
     `vehicleType` = Minivan

   List of manufacturer(s) present in the database :

   - `uID` = 1
     `name` = Tata
     `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
     `itemCount` = 2

   **INPUT ACTION :**
   Call the `retrieveData()` method.

   **GOLDEN OUTPUT / RESPONSE :**
   `searchMap.get("Suspension").get(1).get("Hatchback").uID` = 1
   `searchMap.get("Suspension").get(1).get("Hatchback").type` = Suspension
   `searchMap.get("Suspension").get(1).get("Hatchback").price` = 5400
   `searchMap.get("Suspension").get(1).get("Hatchback").quantity` = 4
   `searchMap.get("Suspension").get(1).get("Hatchback").manufacturerID` = 1
   `searchMap.get("Suspension").get(1).get("Hatchback").vehicleType` = Hatchback

   `itemsList.get(2).uID` = 2
   `itemsList.get(2).type` = Mirror
   `itemsList.get(2).price` = 3600

itemsList.get(2).quantity = 5
itemsList.get(2).manufacturerID = 1
itemsList.get(2).vehicleType = Minivan

manufacturersList.get(1).uid = 1
manufacturersList.get(1).name = Tata
manufacturersList.get(1).address = 7th Floor, Amar Business Park, Andheri, Mumbai
manufacturersList.get(1).itemCount = 2

### 3.4.2 Testing the `removeItem(int itemID)` function (6.4.2)

1. *Remove an item when multiple items are present in the inventory database*

   **PRE CONDITION :**
   List of items present in the inventory :

   - uID = 1
     type = Suspension
     price = 5400
     quantity = 4
     manufacturerID = 1
     vehicleType = Hatchback
   - uID = 2
     type = Mirror
     price = 3600
     quantity = 5
     manufacturerID = 1
     vehicleType = Minivan

   **INPUT ACTION :**
   Call the `removeItem()` method with itemID = 2.

   **GOLDEN OUTPUT / RESPONSE :**
   The item should be removed from the inventory.
   New list of items in the inventory :

   - uID = 1
     type = Suspension
     price = 5400
     quantity = 4
     manufacturerID = 1
     vehicleType = Hatchback

2. *Remove an item when only a single item is present in the inventory*

   **PRE CONDITION :**
   Only one item is present in the inventory :

   - uID = 1
     type = Suspension
     price = 5400

`quantity` = 4
`manufacturerID` = 1
`vehicleType` = Hatchback

**INPUT ACTION :**
Call the `removeItem()` method with `itemID` = 1.

**GOLDEN OUTPUT / RESPONSE :**
The item should be removed from the inventory. Now the list of items in the inventory becomes empty.

3. *The manufacturer of the item being deleted makes some other item(s) too.*

**PRE CONDITION :**
List of items present in the inventory :

- `uID` = 1
  `type` = Suspension
  `price` = 5400
  `quantity` = 4
  `manufacturerID` = 1
  `vehicleType` = Hatchback
- `uID` = 2
  `type` = Mirror
  `price` = 3600
  `quantity` = 5
  `manufacturerID` = 1
  `vehicleType` = Minivan

List of manufacturer(s) :

- `uID` = 1
  `name` = Tata
  `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
  `itemCount` = 2

**INPUT ACTION :**
Call the `removeItem()` method with `itemID` = 2.

**GOLDEN OUTPUT / RESPONSE :**
The item should be removed from the inventory.
New list of items in the inventory :

- `uID` = 1
  `type` = Suspension
  `price` = 5400
  `quantity` = 4
  `manufacturerID` = 1
  `vehicleType` = Hatchback

New list of manufacturer(s) :

- `uID` = 1
  `name` = Tata
  `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
  `itemCount` = 1

4. *The manufacturer of the item being deleted does not make any other item.*

**PRE CONDITION :**
List of items present in the inventory :

- `uID` = 1
  `type` = Suspension
  `price` = 5400
  `quantity` = 4
  `manufacturerID` = 1
  `vehicleType` = Hatchback
- `uID` = 2
  `type` = Mirror
  `price` = 3600
  `quantity` = 5
  `manufacturerID` = 2
  `vehicleType` = Minivan

List of manufacturer(s) :

- `uID` = 1
  `name` = Tata
  `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
  `itemCount` = 1
- `uID` = 2
  `name` = Michelin
  `address` = 24, Chanu Road, Bandra, Mumbai
  `itemCount` = 1

**INPUT ACTION :**
Call the `removeItem()` method with `itemID` = 2.

**GOLDEN OUTPUT / RESPONSE :**
The item should be removed from the inventory.
New list of items in the inventory :

- `uID` = 1
  `type` = Suspension
  `price` = 5400
  `quantity` = 4
  `manufacturerID` = 1
  `vehicleType` = Hatchback

New list of manufacturer(s) :

- `uID` = 1
  `name` = Tata
  `address` = 7th Floor, Amar Business Park, Andheri, Mumbai
  `itemCount` = 1