# TEST CASES

for

# A Motor Part Shop Software

Prepared by -

**Jatin Gupta** (20CS10087)

**Gopal** (20CS30021)

**Jay Kumar Thakur** (20CS30024)

Indian Institute of Technology, Kharagpur

March 23, 2022

# Contents

# 1 Introduction and Important Points

This is the Test Suite document for the Motor Part Shop Software. Here, we list down test cases along with their expected output for each scenario described in sections 10 and 11 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

This page cannot be tested because this is just an 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

- Both Username and Password are correct

  **INPUT :**
  Username = Jaypal
  Password = Jatin@123

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

- Username is correct but Password is incorrect

  **INPUT :**
  Username = Jaypal
  Password = Jatin@456

  **EXPECTED OUTPUT :**
  The login attempt is unsuccessful and "*Invalid Login Please Try Again!!*" message is displayed.

- Username is incorrect but Password is correct

  **INPUT :**
  Username = Rampal
  Password = Jatin@123

  **EXPECTED OUTPUT :**
  The login attempt is unsuccessful and "*Invalid Login Please Try Again!!*" message is displayed.

- Both Username and Password are incorrect

  **INPUT :**
  Username = Rampal
  Password = Jatin@456

**EXPECTED OUTPUT :**
The login attempt is unsuccessful and “*Invalid Login Please Try Again!!*” message is displayed.

## 2.3 Dashboard

- Testing of all buttons

**INPUT :**
Click on all the buttons - Modify/Add Item, Modify/Add Manufacturer, Buy an Item, Sell an Item, View Inventory, END Day and Statistics, one by one.

**EXPECTED OUTPUT :**
Each click should redirect to the appropriate window.

## 2.4 Modify the Item Database

**Pre-Condition :**
There are the following items in the inventory :

➔ Item Type = Breaks
Manufacturer(s) = Sibre Brakes, Brembo Brake
Vehicle Type = Motorbike

➔ Item Type = Tyre
Manufacturer(s) = MRF, Apollo
Vehicle Type = 4 Wheeler

➔ Item Type = Shock Absorber
Manufacturer(s) = Magneti Marelli Motherson
Vehicle Type = Motorbike

### 2.4.1 Testing “Add New Item” Button

□ After clicking on “Add New Item” Button we move on to the next window that contains necessary input fields for adding a new item.

- Testing of input string name of the item

**INPUT :**
Item Type = Tyre
Vehicle Type = Motorbike
Price = 3000
Initial Quantity = 10
Rack Number = A1

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

- The input stock amount is a positive integer

**INPUT :**
Item Type = Tyre
Vehicle Type = Motorbike
Price = 3000

Initial Quantity = 10
Rack Number = A1

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

- The input stock amount is 0

  **INPUT :**
  Item Type = Tyre
  Vehicle Type = Motorbike
  Price = 3000
  Initial Quantity = 0
  Rack Number = A1

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered Quantity is invaild.

- The input stock amount is a string

  **INPUT :**
  Item Type = Tyre
  Vehicle Type = Motorbike
  Price = jgj
  Initial Quantity = 10
  Rack Number = A1

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered Price is invaild.

- Input vehicle type is a valid string

  **INPUT :**
  Item Type = Tyre
  Vehicle Type = Motorbike
  Price = 3000
  Initial Quantity = 10
  Rack Number = A1

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

- Input vehicle type is not valid

  **INPUT :**
  Item Type = Tyre
  Vehicle Type = Mot$%^orbike
  Price = 3000
  Initial Quantity = 0
  Rack Number = A1

  **EXPECTED OUTPUT :**

An error message should be displayed telling that the entered Vehicle Type is invaild.

- Input rack is valid

  **INPUT :**
  Item Type = Tyre
  Vehicle Type = Motorbike
  Price = 3000
  Initial Quantity = 0
  Rack Number = A1

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

### 2.4.2 Testing “Item’s Manufacturer” Button

□ After selecting an item from the list and clicking on “Item’s Manufacturer” Button we move on to the next window that contains the item's manufacturer list.

- Checking the Add button

□ After clicking on “Add” Button we move on to the next window that displays all manufacturers then we will select the manufacturer from the list and enter its cost price.

  - The input cost price should be a positive number

    **INPUT :**
    Cost Price = 3000

    **EXPECTED OUTPUT :**
    Manufacturer is Added to the item’s manufacturer list.

  - The input cost price is 0 or negative

    **INPUT :**
    Cost Price = 0/-3000

    **EXPECTED OUTPUT :**
    An error message should be displayed telling that the entered Cost is invaild.

  - The input cost price is string

    **INPUT :**
    Cost Price = abc

    **EXPECTED OUTPUT :**
    An error message should be displayed telling that the entered Cost is invaild.

- Checking the Remove button

**INPUT :**
Selecting a manufacturer from the item’s manufacturer list and then clicking on the “Remove” Button.

**EXPECTED OUTPUT :**
Manufacturer is Removed from the item's manufacturer list.

## 2.5 Modify the Manufacturer Database

**Pre-Condition :**
There are the following items in the inventory :

- ➔ Manufacturer Name = MRF
  Address = C-211, JCB, IIT KGP
  Phone Number = 123456789
  Item(s) = Tyre, Breaks
- ➔ Manufacturer Name = Magneti Marelli Motherson
  Address = C-103, LBS, IIT KGP
  Phone Number = 987654321
  Item(s) = Shock Absorber, Breaks
- ➔ Manufacturer Name = Apollo
  Address = C-210, LBS, IIT KGP
  Phone Number = 9993454321
  Item(s) = Tyre

### 2.5.1 Testing “Add New Manufacturer” Button

□ After clicking on “Add New Manufacturer” Button we move on to the next window that contains necessary input fields for adding a new manufacturer.

- Testing of the input manufacturer name

  **INPUT :**
  Name = MRF
  Address = C-211, JCB, IIT KGP
  Phone Number = 123456789

  **EXPECTED OUTPUT :**
  This new manufacturer should be added to the manufacturer's list.

- Testing of the input address of the manufacturer

  **INPUT :**
  Name = MRF
  Address = C-211, JCB, IIT KGP
  Phone Number = 123456789

  **EXPECTED OUTPUT :**
  This new manufacturer should be added to the manufacturer's list.

- Testing of the correctness of contact of the manufacturer

  **INPUT :**
  Name = MRF

Address = C-211, JCB, IIT KGP
Phone Number = 123456789

**EXPECTED OUTPUT :**
This new manufacturer should be added to the manufacturer's list.

### 2.5.2 Testing “View Manufactured Items” Button

**INPUT :**
Selecting a manufacturer from the list and clicking on “View Manufacturer Items”.

**EXPECTED OUTPUT :**
List of manufactured items should be displayed.

## 2.6 Buy an Item

□ After clicking on the “Buy an Item” Button we move on to the next window that contains the item list and the user will select an item from that list.

- The input given in the quantity field is a positive integer

  **INPUT :**
  Quantity = 100

  **EXPECTED OUTPUT :**
  Quantity of the item in the inventory list should be increased by 100.

- The input given in the quantity field is zero

  **INPUT :**
  Quantity = 0

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is invaild.

- The input given in the quantity field is negative integer

  **INPUT :**
  Quantity = -34

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is invaild.

- The input given in the quantity field is string

  **INPUT :**
  Quantity = abc

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is invaild.

## 2.7 Sell an Item

☐ After clicking on “Sell an Item” Button we move on to the new window that contains the item list and the user will select an item from that list.

- The input given in the quantity field is zero

  **INPUT :**
  Quantity = 0

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is invaild.

- The input given in the quantity field is negative integer

  **INPUT :**
  Quantity = -300

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered Quantity is invaild.

- The input given in the quantity field is string

  **INPUT :**
  Quantity = abc

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is invaild.

- The input given in the quantity field is greater than Stock

  **INPUT :**
  Quantity = 1000 (Greater than Stock quantity)

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered quantity is greater than stock quantity.

- The input given in the price field is negative number

  **INPUT :**
  Price = -6543.35

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered price is invalid.

- The input given in the price field is string

  **INPUT :**
  Price = abc

  **EXPECTED OUTPUT :**
  An error message should be displayed telling that the entered price is invalid.

## 2.8 View Inventory

- Working of the scrollable list of items

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

  **EXPECTED OUTPUT :**
  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, Vehicle Type, Quantity, Rack number.

## 2.9 END Day

- Display of Revenue for the day

  **EXPECTED OUTPUT:**
  Revenue for the day should be displayed.

- Working of the generated order list

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

## 2.10 Statistics

- View the graph on the day a month has ended

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

- View the graph before the first month has ended

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

- View the graph in the middle of a month

  **EXPECTED OUTPUT :**
  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

### 3.1.1 Testing the getUsername() function

**INPUT :**
Call the getUsername() function

**EXPECTED OUTPUT :**
Returned Name = Jay Kumar Thakur

### 3.1.2 Testing the setUsername(String username) function

**INPUT :**
name = Jay Kumar Thakur

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
name = Jay#$Kumar$#Thakur

**EXPECTED OUTPUT:**
NOT ACCEPTED!

### 3.1.3 Testing the getPassword() function

**INPUT :**
Call the getPassword() function

**EXPECTED OUTPUT :**
Returned password = **********

### 3.1.4 Testing the setPassword(String password) function

**INPUT :**
password = sDf#@5^(K

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
password = $5FD

**EXPECTED OUTPUT :**
NOT ACCEPTED!

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

**Pre-condition :**

- ➔ Target username = Jay Kumar Thakur
- ➔ Target password = tJCfUXGsvt8fA

**INPUT :**
username = Jay Kumar Thakur
password = tJCfUXGsvt8fA

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
username -=Jay Kumar Thakur
password = hj#@%67

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**INPUT :**
username = Ajay Kumar Thakur
password = hj#@%67

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**INPUT :**
username = Ajay Kumar Thakur
password = tJCfUXGsvt8fA

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**Pre-Condition** for Testing the classes :

We create 2 items on the date 2021-04-03 (yyyy-mm-dd) :

➔ **Item 1**
type = Suspension
price = 5400
quantity = 4
itemID = 1
rack_name = 01
manufacturerID_list = {{1, 4800}, {2, 4815}}
vehicleType = Hatchback

➔ **Item 2**
type = Mirror
price = 3600
quantity = 5
itemID = 2
rack_name = 02
manufacturerID_list = {{2, 3250}}
vehicleType = Minivan

We create 3 manufacturers on the date 2021-04-03 (yyyy-mm-dd) :

➔ **Manufacturer 1**
manufacturerID = 1
name = JK Warehouse

address = Shamim Market, Bamrauli, Prayagraj, UP 211012
phone_no = 7764074914
itemID_list = {1}

➔ **Manufacturer 2**
manufacturerID = 2
name = Aaryan Motor Works
address = Block D, Near Clock Tower, Bhuvneshwar, Odisha 148705
phone_no = 8319662744
itemID_list = {2}

➔ **Manufacturer 3**
manufacturerID = 3
aame = Jatin Motor Works
address = C-103 Near Kalam Street Hospital, Jammu, 236143
phone_no = 3479645217
itemID_list = {3}

## 3.2 Testing the Item class

### 3.2.1 Testing the getPart_ID() function

**INPUT :**
Call the getPart_ID() function of each object one by one

**EXPECTED OUTPUT :**
Item ID = 1 {for 1st object}
Item ID = 2 {for 2nd object}

### 3.2.2 Testing the getType() function

**INPUT :**
Call the getType() function of each object one by one

**EXPECTED OUTPUT :**
Type = Suspension
Type = mirror

### 3.2.3 Testing the getPrice() function

**INPUT :**
Call the getPrice() function of each object one by one

**EXPECTED OUTPUT :**
Price = 5400
Price = 3600

### 3.2.4 Testing the getQuantity() function

**INPUT :**
Call the getQuantity() function of each object one by one

**EXPECTED OUTPUT :**

Quantity = 4
Quantity = 5

### 3.2.5 Testing the getVehicleType() function

**INPUT :**
Call the getVehicleType() function of each object one by one

**EXPECTED OUTPUT :**
Vehicle Type = Hatchback
Vehicle Type = Minivan

### 3.2.6 Testing the getRack_Name() function

**INPUT :**
Call the getRack_name() function of each object one by one

**EXPECTED OUTPUT :**
Rack Name = 01
Rack Name = 02

### 3.2.7 Testing the getManufacturer_list() function

**INPUT :**
Call the getManufacturer_list() function of each object one by one

**EXPECTED OUTPUT :**
Manufacturer ID = 1
Name = JK Warehouse
Address = Shamim Market, Bamrauli, Prayagraj, UP 211012
Phone Number = 7764074914

Manufacturer ID = 2
Name = Aaryan Motor Works
Address = Block D, Near Clock Tower, Bhuvneshwar, Odisha 148705
Phone Number = 8319662744

Manufacturer ID = 3
Name = Aaryan Motor Works
Address = Block D, Near Clock Tower, Bhuvneshwar, Odisha 148705
Phone Number = 8319662744

### 3.2.8 Testing the setType(string type) function

**INPUT :**
Type = Tyre

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
Type = Ty$re

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.9 Testing the setPrice(Float price) function

**INPUT :**
Price = 1400

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
Price = 12$56

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.10 Testing the setQuantity(Int quantity) function

**INPUT :**
Quantity = 23

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
Quantity : 23|$

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.11 Testing the setVehicleType(string type) function

**INPUT :**
Type = SUV

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT:**
Type = SU$V

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.12 Testing the setRack_Name() function

**INPUT :**
Rackname = 06-$

**EXPECTED OUTPUT :**
ACCEPTED!

### 3.2.13 Testing the save() function

**INPUT :**
**ITEM 3**
type = Tyre
price = 7400
itemID = 3
rack_name = 06
vehicleType = SUV

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
**ITEM 4**
type = Mirror
price = 3600
itemID = 2
rack_name = 02
vehicleType = Minivan

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.14 Testing the delete() function

**INPUT :**
itemID = 2

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
itemID = 4

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.2.15 Testing the addManufacturer(Manufacturer a, float cost_price) function

**INPUT :**
Call the addManufacturer(Manufacturer a, float cost_price) function of each object one by one
**ITEM 3.**addManufacturer(Manufacturer 2, 4545)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
**ITEM 3.**addManufacturer(Manufacturer 2, 4546)

**EXPECTED OUTPUT :**

ACCEPTED (cost price of that manufacturer will be updated)

### 3.2.16 Testing the updateStock(Int quantity) function

**INPUT :**
Call the addManufacturer(Manufacturer a, float cost_price) function of each object one by one

**EXPECTED OUTPUT :**
ACCEPTED!

## 3.3 Testing the Manufacturer class

### 3.3.1 Testing the getmanufacturer_ID() function

**INPUT :**
Call the getmanufacturer_ID() function of each object one by one

**EXPECTED OUTPUT :**
Manufacturer ID = 1 {for 1st object}
Manufacturer ID = 2 {for 2nd object}
Manufacturer ID = 3 {for 3rd object}

### 3.3.2 Testing the getName() function

**INPUT :**
Call the getName() function of each object one by one

**EXPECTED OUTPUT :**
Name = JK Warehouse
Name = Aaryan Motor Works
Name = Jatin Motor Works

### 3.3.3 Testing the getAddress() function

**INPUT :**
Call the getAddress() function of each object one by one

**EXPECTED OUTPUT :**
Address = Shamim Market, Bamrauli, Prayagraj, UP 211012
Address = Block D, Near Clock Tower, Bhuvneshwar, Odisha 148705
Address = C-103 Near Kalam Street Hospital, Jammu, 236143

### 3.3.4 Testing the getPhone_no() function

**INPUT :**
Call the getPhone_no() function of each object one by one

**EXPECTED OUTPUT :**
Phone Number = 7764074914
Phone Number = 8319662744
Phone Number = 3479645217

### 3.3.5 Testing the setName(string name) function

**INPUT :**
name = JKT Warehouse

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
name = JKT$$Warehouse

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.3.6 Testing the setAddress(string address) function

**INPUT :**
address = JK colony @ search tower 4 * 5 Pentagonal Square

**EXPECTED OUTPUT :**
ACCEPTED!

### 3.3.7 Testing the setPhone_no(Int phone_no) function

**INPUT :**
phone_no = 9580650313

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
phone_no = +919580650313

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**INPUT :**
phone_no = 9569721

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**INPUT :**
phone_no = 95806503$3

**EXPECTED OUTPUT :**
NOT ACCEPTED!

**INPUT :**
phone_no = 345843759348

**EXPECTED OUTPUT :**

NOT ACCEPTED!

### 3.3.8 Testing the save() function

**INPUT :**
**Manufacturer 4**
manufacturerID = 4
name = HJB Motor Works
address = C-103 Near Kalam Street Hospital, Jammu, 236143
phone_no = 3479645217

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
**Manufacturer 5**
manufacturerID = 3
name = Jatin Motor Works
address = C-103 Near Kalam Street Hospital, Jammu, 236143
phone_no = 3479645217

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.3.9 Testing the delete() function

**INPUT :**
manufacturerID = 2

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
manufacturerID = 2

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.3.10 Testing the addItem(Item a) function

**INPUT :**
Call the addItem(Item a) function of each object one by one
**ITEM 3.**addItem(Item 2)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
**ITEM 3.**addItem(Item 2)

**EXPECTED OUTPUT :**
NOT ACCEPTED!

## 3.4 Testing the Invoice class

### 3.4.1 Testing the addItem(Item a) function

**INPUT :**
itemID = 2

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
itemID = 2

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.4.2 Testing the getRevenue() function

**INPUT :**
User calls the getRevenue function

**EXPECTED OUTPUT :**
Total Revenue of the day = 148700

## 3.5 Testing the Database class

### 3.5.1 Testing the retrieveItem() function

**INPUT :**
Database -> { }
User calls the retrieveItem()

**EXPECTED OUTPUT :**
{ } (Empty List)

**INPUT :**
Database as specified earlier
User calls the retrieveItem()

**EXPECTED OUTPUT :**
**ITEM 1**
Type = Suspension
Price = 5400
Quantity = 4
Item ID = 1
Rack Name = 01
Manufacturer ID List = {{1, 4800}, {2, 4815}}
Vehicle Type = Hatchback

**ITEM 2**
Type = Mirror
Price = 3600
Quantity = 5
Item ID = 2

Rack Name = 02
Manufacturer ID List = {{2, 3250}}
VehicleType = Minivan

### 3.5.2 Testing the retrieveManufacturer() function

**INPUT :**
Database -> { }
User calls the retrieveManufacturer()

**EXPECTED OUTPUT :**
{ } (Empty List)

**INPUT :**
Database as specified earlier
User calls the retrieveManufacturer()

**EXPECTED OUTPUT :**
**Manufacturer 1**
Manufacturer ID = 1
Name = JK Warehouse
Address = Shamim Market, Bamrauli, Prayagraj, UP 211012
Phone Number = 7764074914
Item ID List = {1}

**Manufacturer 2**
Manufacturer ID = 2
Name = Aaryan Motor Works
Address = Block D, Near Clock Tower, Bhuvneshwar, Odisha 148705
Phone Number = 8319662744
Item ID List = {2}

**Manufacturer 3**
Manufacturer ID = 3
Name = Jatin Motor Works
Address = C-103 Near Kalam Street Hospital, Jammu, 236143
Phone Number = 3479645217
Item ID List = {2}

### 3.5.3 Testing the retrieveInvoice() function

**INPUT :**
Database as specified earlier
User calls the retrieveInvoice()

**EXPECTED OUTPUT :**
Total Revenue of 3 days = {14800, 15400, 13000}

### 3.5.4 Testing the addItem(Item a) function

**INPUT :**
addItem(Item ITEM_1)

**EXPECTED OUTPUT :**

ACCEPTED!

**INPUT :**
addItem(item ITEM_1)

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.5.5 Testing the addManufacturer (Manufacturer a) function

**INPUT :**
addManufacturer(Manufacturer MANUFACTURER_1)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
addManufacturer(Manufacturer MANUFACTURER_1)

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.5.6 Testing the addInvoice(Invoice invoice) function

**INPUT :**
addInvoice(Invoice INVOICE_1)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
addInvoice(Invoice INVOICE_1)

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.5.7 Testing the deleteItem(Item a) function

**INPUT :**
deleteItem(Item a)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
deleteItem(Item a)

**EXPECTED OUTPUT :**
NOT ACCEPTED!

### 3.5.8 Testing the deleteManufacturer(Manufacturer a) function

**INPUT :**

deleteManufacturer(Manufacturer MANUFACTURER_1)

**EXPECTED OUTPUT :**
ACCEPTED!

**INPUT :**
deleteManufacturer(Manufacturer MANUFACTURER_1)

**EXPECTED OUTPUT :**
NOT ACCEPTED!