

Software Analysis Software Design

for

Motor Part Shop Software

By:

Abhishek Kalyan 12CS30001

M Srinivas Reddy 12CS30019

Table of Contents

Feasibility Study	9
The Problem and its Scope	9
The Problem	9
Storing the Data:.....	9
Defining access privileges:.....	9
The Stakeholders	10
Shop Owner :	10
Description	10
Design Concerns	10
Accessibility Rights:	10
Relationships	11
Customer	11
Description:	11
Design Concerns:	11
Accessibility Rights:	11
Relationships	11
Unusual Circumstances	12
Unavailability of Product:	12
Unexpected response from DB:	12
Dependencies alter:	13
USE CASES	14
LOGIN	14
Name/Short Description:.....	14
Preconditions:.....	14
Post condition:.....	14
Failure Situations:	14

Post condition in case of failure:	14
Actors:.....	14
Trigger:.....	14
Main Success Scenario:	14
Extensions/Variations:	14
CHANGE PASSWORD	15
Name/Short Description:.....	15
Preconditions:.....	15
Post condition:.....	15
Failure Situation:.....	15
Post condition in case of failure:	15
Actors:.....	15
Trigger:.....	15
Main Success Scenario	15
Extensions/Variations:	15
Insert Product	16
Name/Short Description:.....	16
Preconditions:.....	16
Post condition:.....	16
Failure Situations:	16
Post condition in case of failure:	16
Actors:.....	16
Trigger:.....	16
Main Success Scenario:	16
Extensions/Variations:	16
Delete Product.....	17
Name/Short Description:.....	17
Preconditions:.....	17
Post condition:.....	17
Failure Situations:	17
Post condition in case of failure:	17

Actors:.....	17
Trigger:.....	17
Main Success Scenario:	17
Extensions/Variations:	17
Update Product Information.....	18
Name/Short Description:.....	18
Preconditions:.....	18
Post condition:.....	18
Failure Situations:	18
Actors:.....	18
Trigger:.....	18
Main Success Scenario:	18
Check Sales Information.....	19
Name/Short Description:.....	19
Preconditions:.....	19
Post condition:.....	19
Failure Situations:	19
Actors:.....	19
Trigger:.....	19
Main Success Scenario:	19
Print Requirements and order selected items to vendor.....	20
Name/Short Description:.....	20
Preconditions:.....	20
Post condition:.....	20
Failure Situations:	20
Actors:.....	20
Trigger:.....	20
Main Success Scenario:	20
User Validation:.....	21

Name/Short Description:.....	21
Preconditions:.....	21
Post condition:.....	21
Failure Situations:	21
Actors:.....	21
Trigger:.....	21
Main Success Scenario:	21
Change Password	22
Name/Short Description:.....	22
Preconditions:.....	22
Post condition:.....	22
Failure Situations:	22
Actors:.....	22
Trigger:.....	22
Main Success Scenario:	22
Search Products and Add to Cart	23
Name/Short Description:.....	23
Preconditions:.....	23
Post condition:.....	23
Failure Situations:	23
Actors:.....	23
Trigger:.....	23
Main Success Scenario:	23
Compare Product	24
Name/Short Description:.....	24
Preconditions:.....	24
Post condition:.....	24

Failure Situations:	24
Post condition in case of failure:	24
Actors:.....	24
Trigger:.....	24
Main Success Scenario:	24
Edit Cart	25
Name/Short Description:.....	25
Preconditions:.....	25
Post condition:.....	25
Failure Situations:	25
Actors:.....	25
Trigger:.....	25
Main Success Scenario:	25
Checkout	26
Name/Short Description:.....	26
Preconditions:.....	26
Post condition:.....	26
Failure Situations:	26
Actors:.....	26
Trigger:.....	26
Main Success Scenario:	26
Detailed Design	27
I/O Procedures and User Interfaces.....	27
User Interfaces.....	27
Class Design.....	29

This Page is Intentionally left Blank

Feasibility Study

1.1 The Problem and its Scope:

This software address the needs of the **Motor Part Shop** , it helps in the sale of the products(parts) , managing the inventory, sustaining the sale through JIT (Just In Time) , Accounts, Sale Statistics of the Shop.

1.2 The Problem

The Motor Part Shop Software is mainly a database problem, it pertains to efficient storing of data. The problem may be written down as the following sub-problems

1.2.1 Storing the Data

A model for the data is required which prevents irrelevant entries to be added and at the same time speedy recovery of data is to be maintained

1.2.2 Identifying The Stakeholders

The key users who shall interact with the software belong to this group shall be identified by login credentials in the welcome screen.

1.2.3 Defining Access Privileges

Different stack people have different levels of access to different data. This design model should ensure that it sticks to it.

The Stakeholders

Shop Owner

2.1.1 Description

The Shop Owner is the sole authority for Inserting new product in the stock or database or deleting it, he is also the sole authority to check the statistics and order new products based on the threshold value and he handles various managements.

2.1.2 Design Concerns

In this Project we have assumed that for a Shop a *single owner* exists. The owner is allotted a predefined username and password required to access his page. This data is stored in the database. However he can change his password in his login page

2.1.3 Accessibility Rights

- **Login** : Owner has his own valid login credentials which give him administrator privileges.
- **Insert Product**: He can insert a new product in the database.
- **Managing Sales** : Responsible for properly allocating selected product to the customer and checking the checking the statistics.
- **Manage customers** : View the details of any customer, delete an existing customer.
- **Manage Inventory** : View the Inventory in the shop. Insert, Delete, Update the attributes of Product(change their cost etc.) in the Inventory.
- **Calculate Threshold and order**: He can check the threshold value for all the items in the inventory and order the items to the vendor accordingly.
- **Review Statistics** : View the revenue for each day and at the end of the each month, statistics graph showing the sales for each day of month.
- **Manage Invoices / Transaction** : Can manage invoices and Transactions.
- He can read the details of the customers but he can't change any information about them. Only editing feature is removing the customer.

2.1.4 Relationships

There exists a one-many relationship between the owner and the customer as is described in the Class Diagram

Customer

2.2.1 Description

Any new customer can register themselves by entering the details after registering the Customer can purchase any product in the Inventory , a functionality called "Compare product" is available which helps them in selecting their desired product. they can add products to cart and they can also edit the products in their existing cart.

2.2.2 Design Concerns

In this Project we have assumed that for a Shop, *any number of Customers* may exist. a customer can have only *single cart* the items in their cart can be purchased by selecting the "check out" function in the login window. a unique username, secure password should be selected by the user at the time of registration can be used to login. However he can change the password on his login page.

2.2.3 Accessibility Rights

- **Login** : Customers must have valid login id to enter in the site.
- He can't access the Customer Database. The logging in tasks is to managed by separate module. However he can see the purchases made by him.
- **Register** : New customers can register themselves, they can change their details later on their login page.
- **Edit/ Add to cart** : he can add or remove items or increase or decrease the quantity of items in his cart.
- **Purchase new Parts** : any registered customer can purchase items in his cart.
- **Compare Products** : Can view all available products in a category and can compare them to make best choice for the purchase.

2.2.4 Relationships

There exists a many-one relationship between the customer and the owner as is described in the Class Diagram.

Alternatives, Evaluations

- We can also make some algorithm for estimating which product must be given more priority in sales by analysing the statistics.
 - We can also make accounts for dialy customers, store all their transactions for giving them some incentives.
 - We can increase the security by adding more features like face detection.
 - We can also develop this software to run on a web server through which this can be online shopping portal
-

Unusual Circumstances

4.1 Unavailability of Product

The Product requested by the customer may not be available due to unforeseen circumstances, Accounting for those entails customer cannot make the purchase.

4.2 Unexpected Response from DB

There are various associated problems with this. The database of stakeholder's machine might not respond thus bringing his associated tasks to standstill. The software might go down unexpectedly.

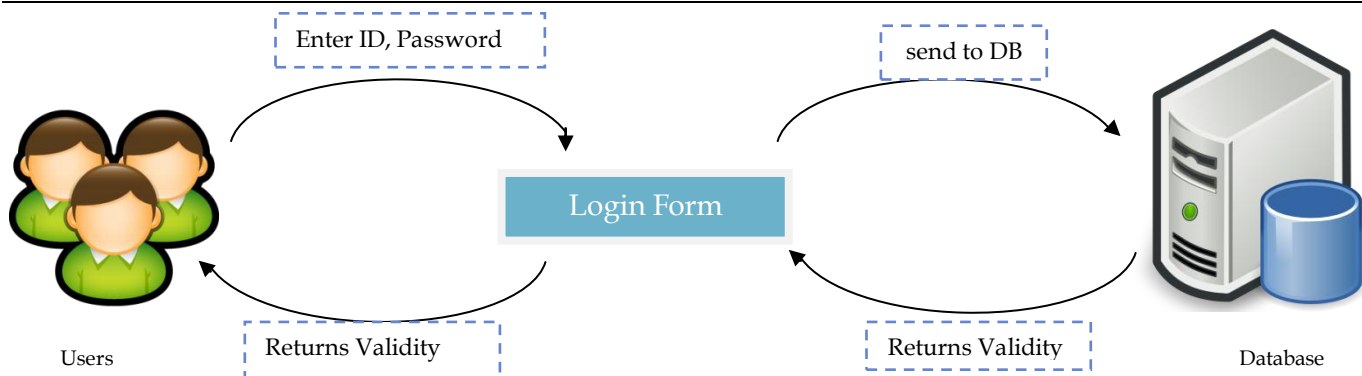
4.3 Dependencies Alter

MPSS is dependent upon external software environment. These may change at any time. If their compatibility with compatibility with software changes, this would make the software non functional.

Use Cases

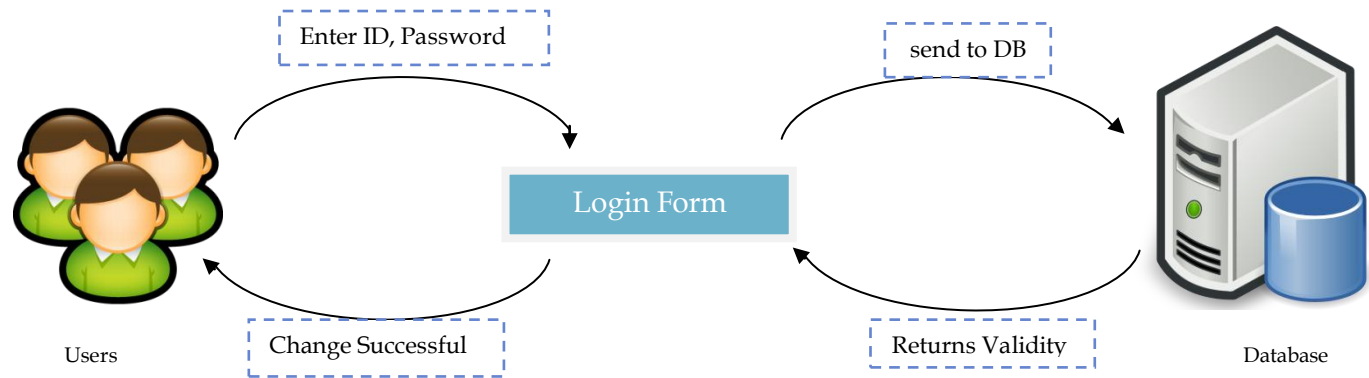
5.1 All Stakeholders Tasks

5.1.1 Login



- **Name/ Short Description :** Allow the users of this software to login.
- **Pre conditions :** User data is stored in the database.
- **Post condition :** Users are led to their screen.
- **Failure Situations :** Wrong Password or User ID is entered.
- **Post condition in case of Failure :** Error message is displayed.
- **Actors :** all users of the System.
- **Trigger :** Login is clicked.
- **Main success scenario :**
 1. Login is clicked
 2. Data is matched with password in Database
 3. Employee is Led to his Screen

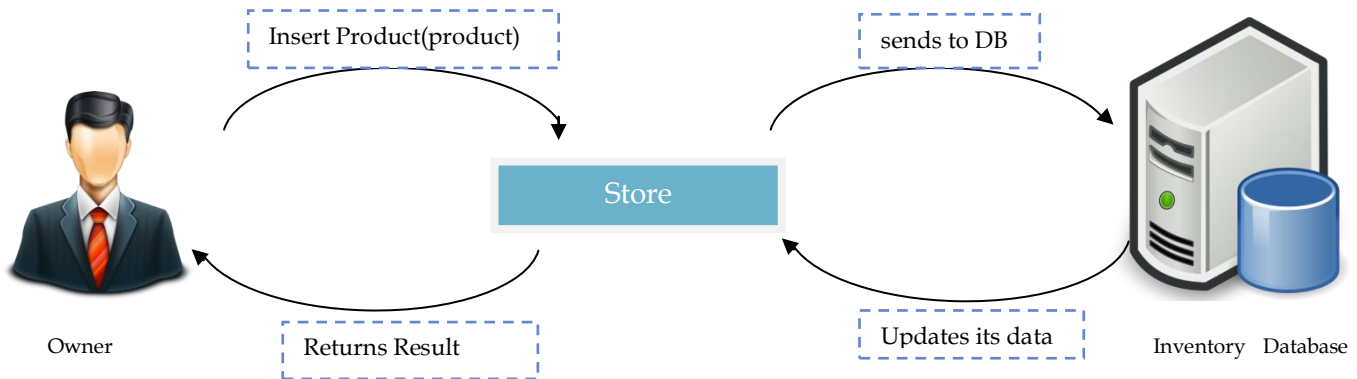
5.1.2 Change Password



- **Name/ Short Description :** Allow the users of this software to change password.
- **Pre conditions :** User data is stored in the database.
- **Post condition :** Password of the user is changed and message is displayed
- **Failure Situations :** Password mismatch.
- **Post condition in case of Failure :** Error message is displayed.
- **Actors :** all users of the System.
- **Trigger :** Login is clicked.
- **Main success scenario :**
 1. Change password is clicked.
 2. Data is matched with password in Database.
 3. Database is updated.
 4. Message is displayed that the password is changed.
- **Extensions/ Variations :** Error message is displayed.

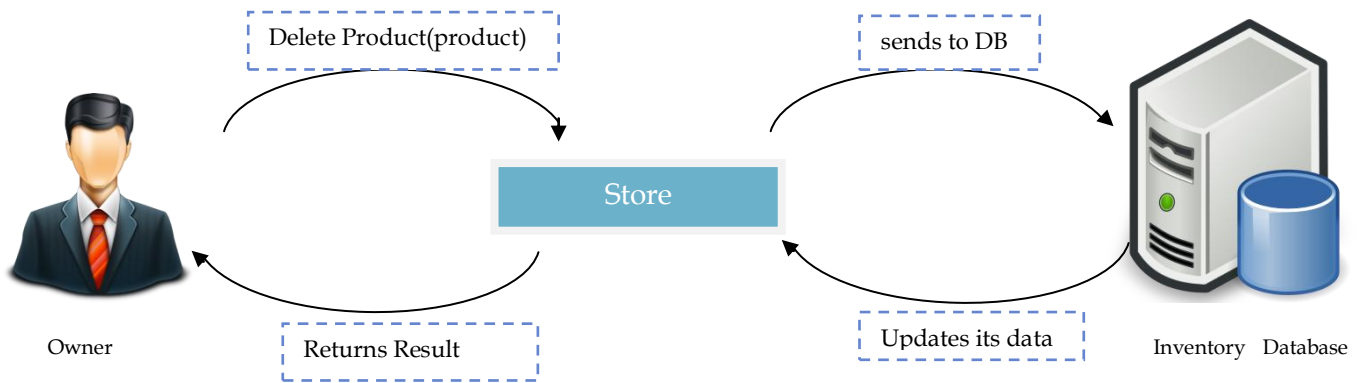
5.2 Owner's Tasks

5.2.1 Insert Product



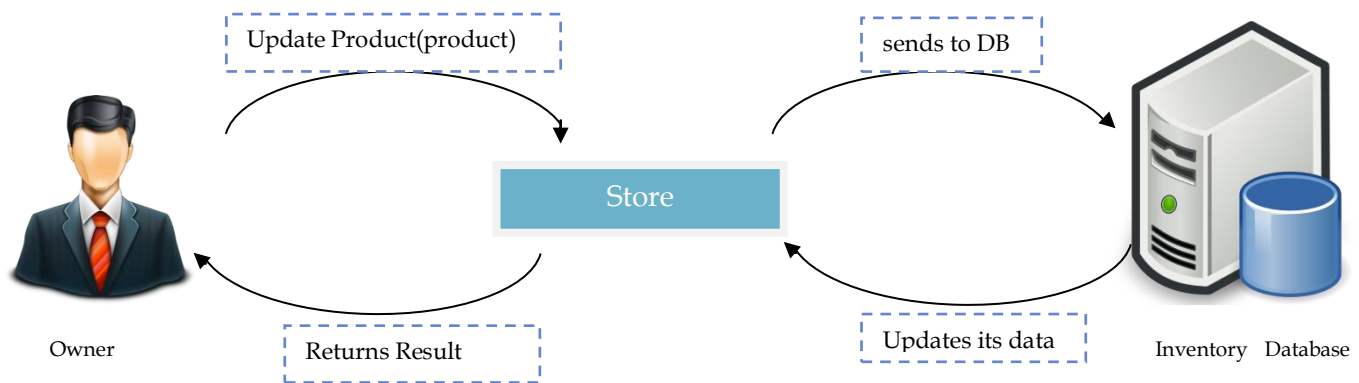
- **Name/ Short Description :** Allow the Owner of this shop to Insert Product in the Inventory.
- **Pre conditions :** Inventory data is stored in the database.
- **Post condition :** New Product is added to the database and it displays a message
- **Failure Situations :**
 1. Any detail of the product is not specified
 2. The product with the given id is already present in the database.
- **Post condition in case of Failure :** Error message is displayed.
- **Actors :** Owner of the shop.
- **Trigger :** Insert product button is clicked.
- **Main success scenario :**
 1. Insert product is clicked.
 2. Data is added in Database.
 3. Database is updated.
 4. Message is displayed that the product is added.
- **Extensions/ Variations :** Error message is displayed.

5.2.2 Delete Product



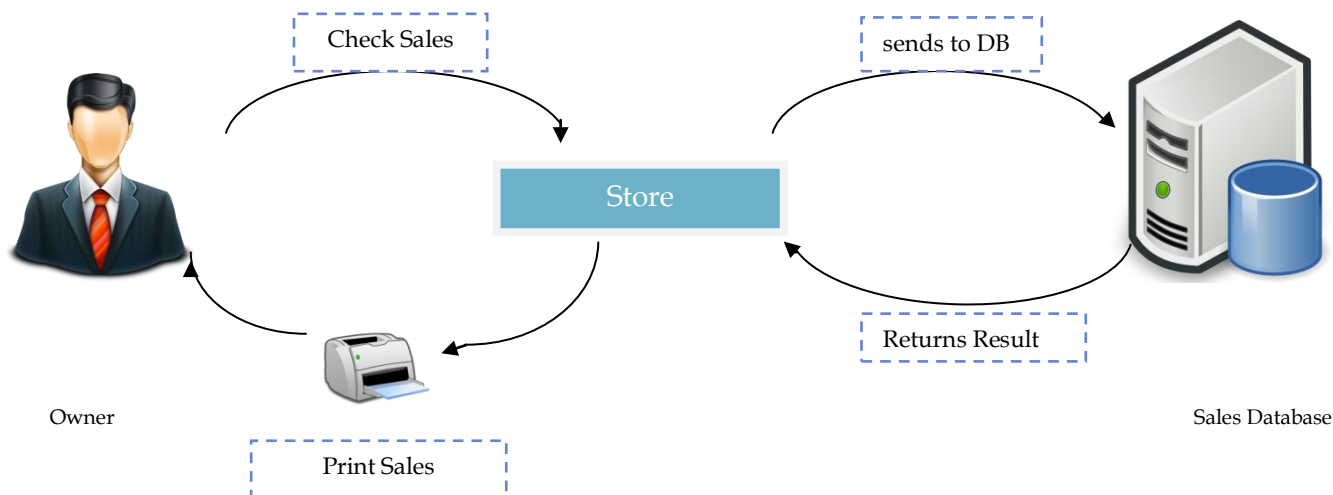
- **Name/ Short Description** : Allow the Owner of this shop to Delete Product in the Inventory.
- **Pre conditions** : Inventory data is stored in the database.
- **Post condition** : Product given is deleted in the Inventory database and it displays a message
- **Failure Situations** : Product given is not present in the Database
- **Post condition in case of Failure** : Error message is displayed.
- **Actors** : Owner of the shop.
- **Trigger** : Delete product button is clicked.
- **Main success scenario** :
 1. Delete product is clicked.
 2. Data is deleted in Database.
 3. Database is updated.
 4. Message is displayed that the product is Deleted.
- **Extensions/ Variations** : Error message is displayed.

5.2.3 Update Product Information



- **Name/ Short Description :** Allow the Owner of this shop to Update Product in the Inventory. update product info includes changing the price of the product or changing the vendor details or size of the part.
- **Pre conditions :** Inventory data is stored in the database.
- **Post condition :** Product given is Updated in the Inventory database and it displays a message
- **Failure Situations :** Product given is not present in the Database
- **Post condition in case of Failure :** Error message is displayed.
- **Actors :** Owner of the shop.
- **Trigger :** Update product button is clicked.
- **Main success scenario :**
 1. Delete product is clicked, the product details can be edited there.
 2. Data is updated in Database.
 3. Database is updated.
 4. Message is displayed that the product is Updated.
- **Extensions/ Variations :** Error message is displayed.

5.2.4 Check Sales Information

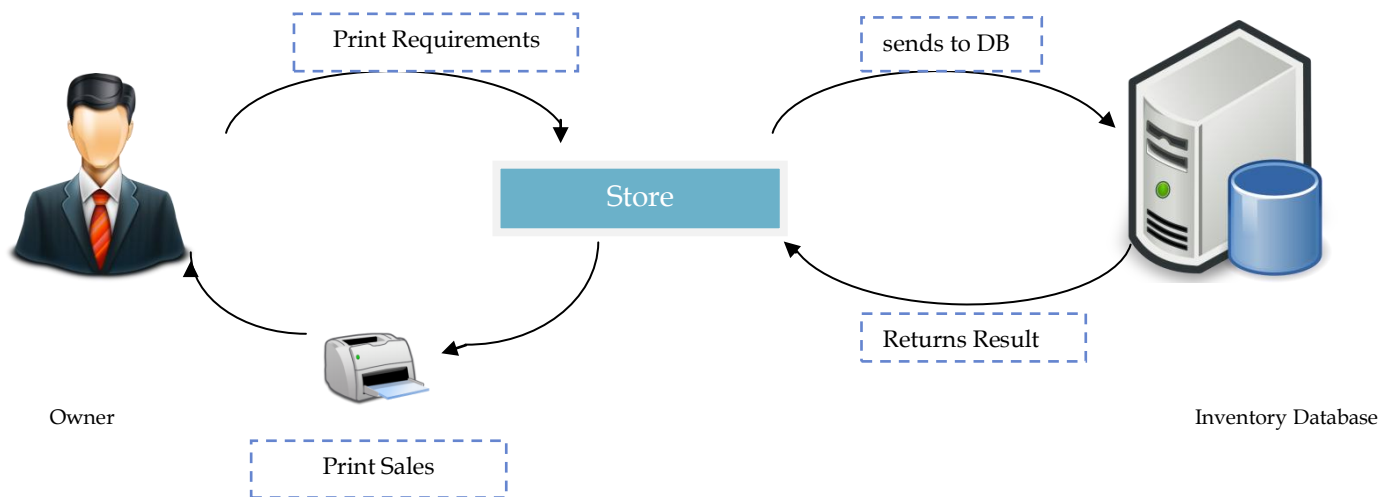


- **Name/ Short Description :** Allow the Owner of this shop to Print the sales report of the shop for a given month.
- **Pre conditions :** Sales data is stored in the database.
- **Post condition :** Sales report is printed.
- **Failure Situations :** No failure situations, if no sales take place in a given month it will print the same.
- **Post condition in case of Failure :** -
- **Actors :** Owner of the shop.
- **Trigger :** Sales report button is clicked and month is selected.

Main success scenario :

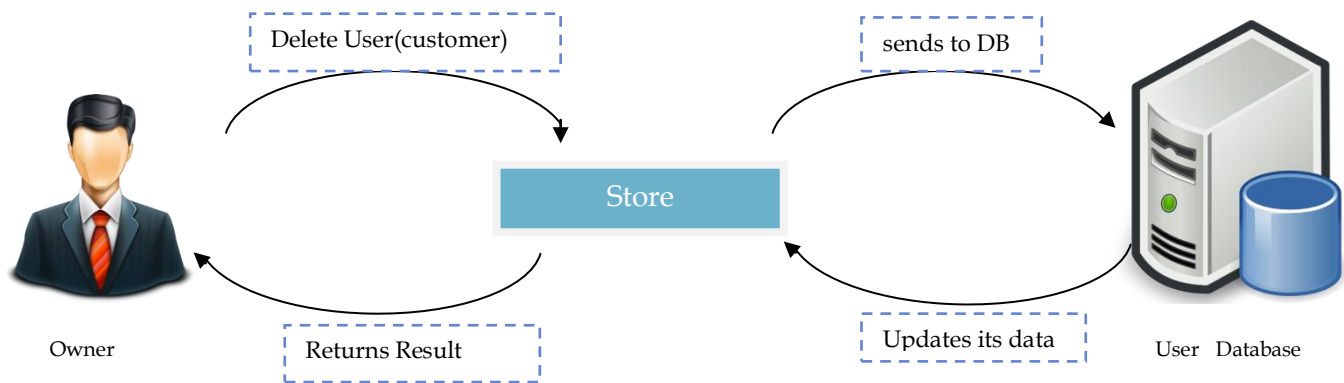
1. Sales report button is clicked and month is selected.
2. Data is printed.

5.2.5 Print Requirements and order selected items to Vendor



- **Name/ Short Description :** Allow the Owner of this shop to Print the requirement list to sustain sale for a week using threshold value.
- **Pre conditions :** Inventory data is stored in the database.
- **Post condition :** Requirement list is printed or mailed to the vendors.
- **Failure Situations :** No failure situations.
- **Post condition in case of Failure :** -
- **Actors :** Owner of the shop.
- **Trigger :** Get Requirement button is clicked from the owner selection form.
- **Main success scenario :**
 1. Get Requirement button is clicked from the owner selection form.
 2. Data is printed.
 3. Then owner manually selects the items to be ordered to vendor.
 4. Selected Items are ordered

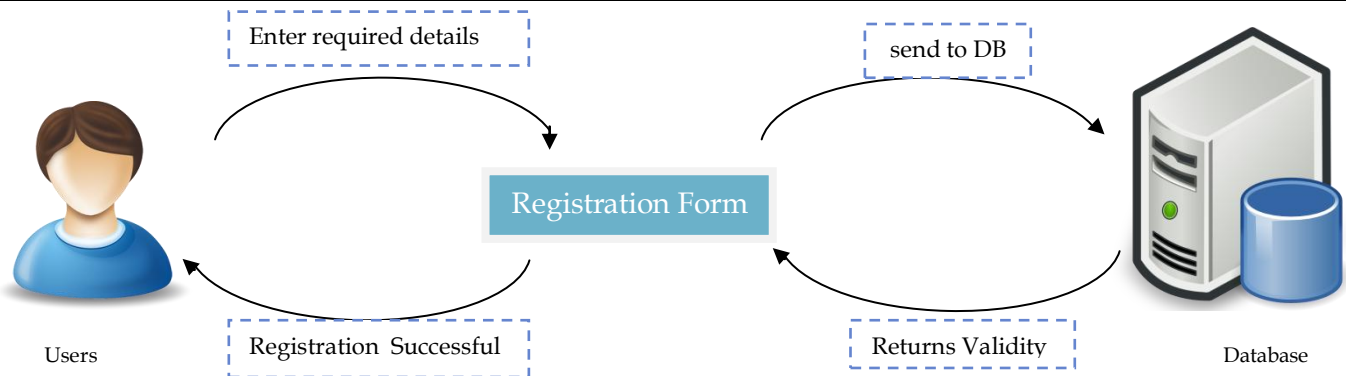
5.2.6 User Validation



- **Name/ Short Description :** This function is a special function for Administrative functions it allows the Owner of this shop to Delete Customer from the DB.
- **Pre conditions :** User data is stored in the database.
- **Post condition :** Customer given is deleted from the database and it displays a message
- **Failure Situations :** Customer given is not present in the Database
- **Post condition in case of Failure :** Error message is displayed.
- **Actors :** Owner of the shop.
- **Trigger :** User Validation button is clicked.
- **Main success scenario :**
 1. User Validation is clicked.
 2. Data is deleted in Database.
 3. Database is updated.
 4. Message is displayed.
- **Extensions/ Variations :** Error message is displayed.

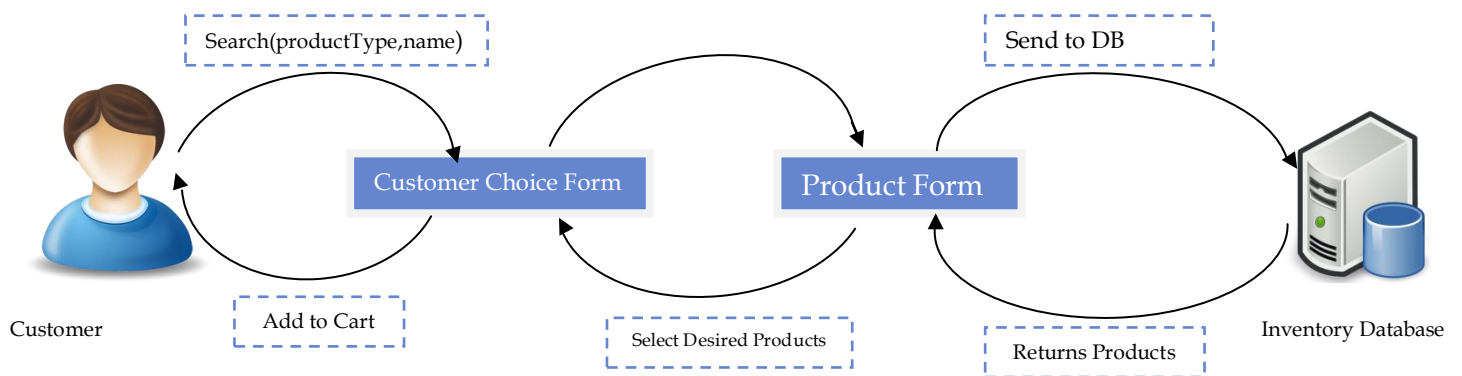
5.3 Customer's Tasks

5.3.1 Change Password



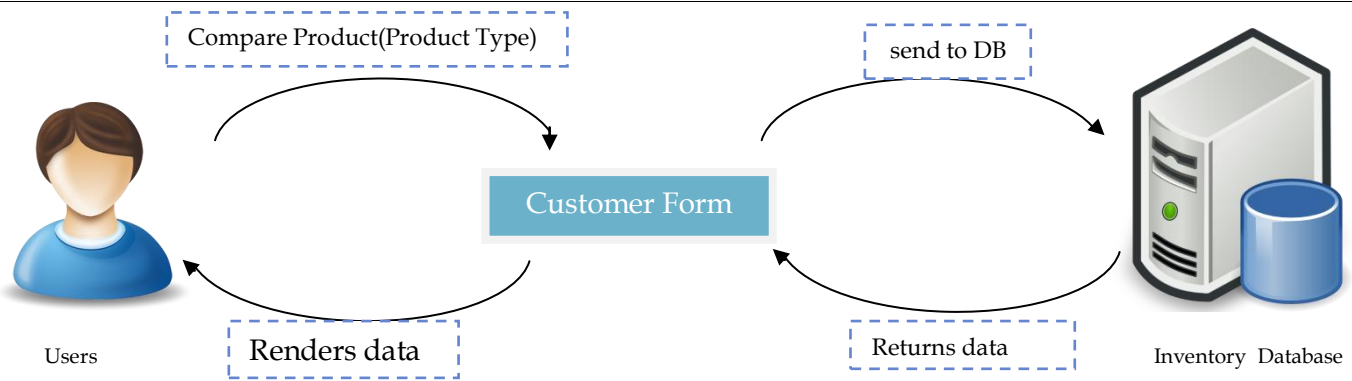
- **Name/ Short Description :** Allows the New customer to Register.
- **Pre conditions :** -.
- **Post condition :** User is registered and message is displayed
- **Failure Situations :** Form filled is incomplete.
- **Post condition in case of Failure :** Error message is displayed and it is prompted to fill it again.
- **Actors :** New customers.
- **Trigger :** Register is clicked.
- **Main success scenario :**
 1. Register is clicked.
 2. The registration form is filled.
 3. Database is updated.
 4. Message is displayed that the Customer is registered.
- **Extensions/ Variations :** Error message is displayed.

5.3.2 Search Product and Add to Cart



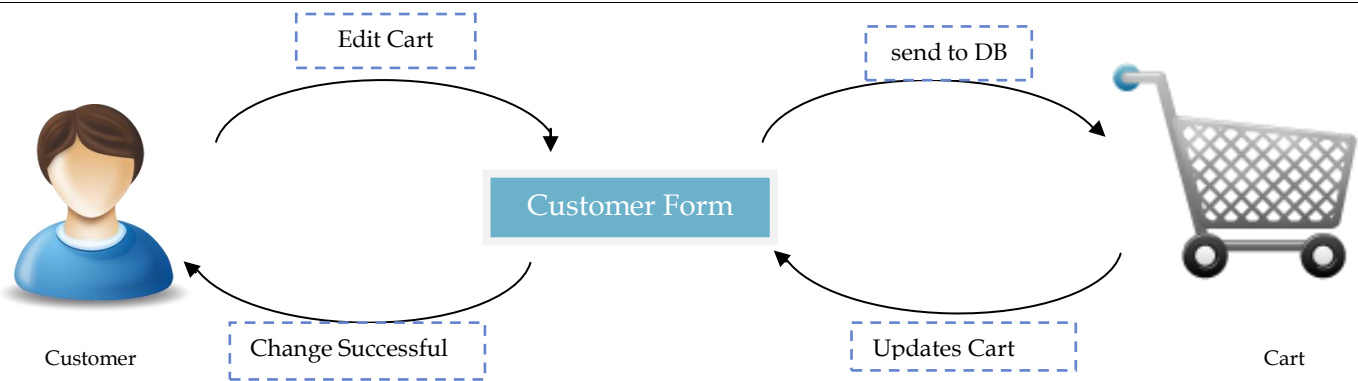
- **Name/ Short Description :** Allows the Customer to search Product in the Inventory and add their desired Product to the cart. search can be filtered by product type, product id, name.
- **Pre conditions :** Inventory data is stored in the database.
- **Post condition :** New Product is added to the cart and it displays a message
- **Failure Situations :** Product Not Found, if the entered details of the product could not match any product.
- **Post condition in case of Failure :** Error message is displayed, and it is prompted to search it again.
- **Actors :** Customer.
- **Trigger :** Product details can be entered in the customer form and selected products are added to cart.
- **Main success scenario :**
 1. Product details can be entered in the customer form.
 2. The related products are returned .
 3. user selects products to add in cart.
 4. Message is displayed that the product is added to cart.
- **Extensions/ Variations :** Error message is displayed.

5.3.3 Compare Products



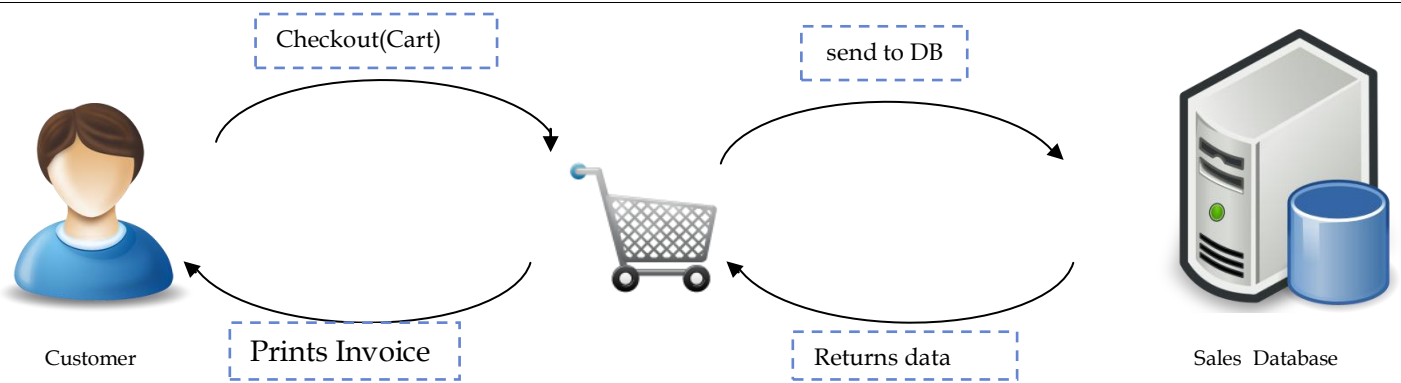
- **Name/ Short Description :** Allows the Customer to Compare different products of the same type for comfortable selection.
- **Pre conditions :** Inventory data is stored in the database..
- **Post condition :** Product list is returned.
- **Failure Situations :** Entered product type is not available in the database.
- **Post condition in case of Failure :** Error message is displayed and it is prompted to fill it again.
- **Actors :** Customers.
- **Trigger :** Product Type is entered in the customer form and Compare Products button is clicked.
- **Main success scenario :**
 1. Compare Products is clicked.
 2. Product Type is Entered.
 3. Database returns the Product list.
- **Extensions/ Variations :** Error message is displayed and user is prompted to enter again.

5.3.4 Edit Cart



- **Name/ Short Description :** Allow the customers of Shop to Edit Cart.
- **Post condition :** Cart is updated.
- **Failure Situations :** No Failure cases.
- **Post condition in case of Failure :** -.
- **Actors :** Customers.
- **Trigger :** Edit cart is clicked .
- **Main success scenario :**
 1. Edit cart is clicked.
 2. The cart is returned and the items to be deleted or the quantity is changed .
 3. Cart is updated.

5.3.5 Checkout



- **Name/ Short Description :** Allows the Customer to Checkout and purchase products.
- **Pre conditions :** Cart should have at least one item.
- **Post condition :** Product list is returned.
- **Failure Situations :** Number of items in the cart is 0.
- **Post condition in case of Failure :** Error message is displayed .
- **Actors :** Customers.
- **Trigger :** Checkout button is clicked.
- **Main success scenario :**
 1. Checkout is clicked.
 2. The sales database is updated.
 3. Invoice is printed.
- **Extensions/ Variations :** Error message is displayed and user is prompted to enter again.

6. Detailed Design

6.1 Software Architecture

The overall software is best modelled in Object-oriented architecture. In this way, the individual methods may be implemented and encapsulated to prevent interference. It also provides abstraction so two sets of entities may interact with each other with ease. Also, the software must broadly an event-based design and therefore it is best to use a language, which supports event handling. For this and other purposes described above, Java is one of the best candidates to implement the software.

6.2 I/O Procedures and UI

The Screens are Described below:

6.2.1 Login Screen:

It has 2 text fields for Employee ID along with Password and a Login Button and Exit Button. It also has a button "New Customer? Register" for new Customers.

6.2.2 Change Password Screen:

This screen has text fields for confirming the old password and then entering the new password and then another password field to confirm the new password along with a button to change the password and another to go back

6.2.3 Customer Welcome Screen / Home Screen :

The Various choices given in the customer form are given below, It has a button link to different screens :

- Search Item : Takes it to the Products Form Screen where user is prompted to select different Products.
User is given various options to filter his search for the products. like he has to select product type and key in the item name and click search.
- Edit Cart : Takes it to the Existing cart screen (If Exists).
- Compare Products : Takes it to another product Form Screen where the user can see different products available under the selected product type category.
- Edit Customer Details : He is prompted to another screen where he can change his details (Name, address, password, tel No).

6.2.4 Product Form Screen:

Displays the various products in the requested category. all the products are given and the user is asked to select the desired product and there is an option of "Add to Cart" which adds the products into the cart.

The products that are in the cart are displayed in the *cart panel* which is to the left of the screen. There is button named "Checkout" in the cart panel of the product Form Screen, which takes the user to a Checkout Screen where the user can purchase the products.

6.2.5 Cart Screen:

It is similar to the Checkout Screen, you can edit the quantity of items ordered or delete the item in cart.

6.2.6 Checkout Screen:

It Displays the list of items in the cart and user can confirm and make the payment so that he can purchase the products

6.2.7 Owner Welcome Screen:

The Various choices given in the Owner form are given below, It has a button link to different screens :

- Insert/Update/Delete Product : If the owner clicks any of these buttons it takes owner to *Store Functions Screen*.
- Check Sales : It takes Owner to *Sales Screen*
- Get Requirements : it takes Owner to *Requirements Page*.
- Validate User : Takes Owner to *User Validation Page*.

6.2.8 Store Functions Screen:

For Insert Product it asks the user to enter the details of new product, on clicking update/ delete it asks the user to enter the unique product id which is either updated or deleted.

6.2.9 Sales Screen:

It Displays the list of purchases made in the selected month also shows the graph representation of products brought from different categories

6.2.10 Requirements Screen:

It Displays the list of items required to be ordered by the store in order to sustain sales for a week, owner can select items which he likes to order and order them.

6.2.11 User Validation Screen:

It Displays the list of customers registered in the shop sorted by their customer id, the owner can select the customer whom he want to delete.

7 Class Design

The class diagram would further be detailed in the UML Diagram in the next stage. The following are broad entity types, which are the building blocks for the class diagram.

- Owner
- Store
- Customer
- Product
- Product type
- Cart

Please refer the UML and SRS for complete list of functions.