Ahmedabad University

Software Engineering

Project Report

Inventory Management System

Team - 3

Prepared by:

- Shaili Gandhi(AU1841012)
- Kashish Shah(AU1841014)
- Harsh Patel(AU1841015)
- Hriday Nagrani(AU1841042)
- Shivam Lakhtariya (AU1841084)
- Mohammadrazin Mansuri(AU1841097)
- Jainesh Patel(AU1841101)
- Vardhan Shah (AU1841138)

Inventory Management System

1. Project Overview

1.1 Introduction:

- ➔ Inventory is the stock of physical items such as materials, components, work-in-progress, finished goods, etc., held at a specific location at a specific time.
- ➔ Inventory Control is a technique of maintaining and monitoring the size of the inventory at appropriate level, so that the production and distributions take place effectively.

1.2 Domain:

Our domain is a supermarket store like D-mart which has a large number of products from different categories.Our goal is to manage the inventory of the products which are stored at different sections in the warehouse. Our software targets to manage employees ,customers and products.

1.3 Scope:

A supermarket stores and sells products from different categories. It is difficult to manage all these products, their sufficient stocks, sales records, analyzing sales and reordering from time to time. An inventory management system would help in fulfilling the above mentioned tasks.

The system would give appropriate alerts, for reordering the quantity, and help maintain a safe stock. It will also assist in the billing process. Authorized users would be able to analyze the sales of various products, which would help in adjusting the purchase and the sales strategies, thus leading to an increase in profit.

1.4 Advantages:

- ➔ Accuracy in database handling
- ➔ Increase efficiency and reliability of the system
- ➔ User friendly interface
- ➔ Save time, manpower, paperwork
- ➔ Eliminate risk of overstocking and stockout.

2. Software Management Methodology

● Selected Model:

Incremental: This model focuses on the delivery of an operational product with each increment. Early increments are stripped-down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user.

Our major requirements in the project are almost fixed but there are some supplementary requirements that may come into picture during the development of the project. The incremental gives us this flexibility to deal with additional requirements that are not covered in the first increment.

➢ Why incremental and not others?

Waterfall: This is a very basic model in which phases are completed in a linear sequence. If any errors are present, they would not be detected until the testing stage.

Iterative: This model requires user engagement throughout the process, which can become an unnecessary obligation on their part. It also increases the risk of feature risk.

Agile: This model is suitable for projects in which there is possibility of more number of changes in the needs of the end user. Because of that, there is less focus on the initial design and documentation part. In our project major requirements are almost fixed.

Spiral: This model is suitable for projects in which the requirements are complex and can change significantly over the time.

RAD: This is a good model as it takes care of user feedback and also can be completed in a time period of 2-3 months. But it has high dependency on the modelling and needs resources with high business knowledge.

V-Shaped: This model focuses on verification and validation of the components during the development of the project. It is the least flexible model. It is based on complete pre planning and all the requirements should be specified apriori. If any changes happen midway, then the test documents along with requirement documents have to be updated. No feedback can be taken once the final model is developed.

3.Requirements Analysis:

3.1 Stakeholders:

- ➔ Warehouse owner
- ➔ Warehouse Manager
- ➔ Product billing operator
- ➔ Product distribution operator
- ➔ Accountant
- ➔ Customers
- ➔ Supplier

3.2 Functional Requirements:

- The Users should authentically login to the system.
- The System can add or remove quantity of products.
- The System should have a function to Assign a product to its respective Category.
- The System Should have function to fetch the information for different products from the generated barcodes.
- The System allows users to Update product stock information and can change the price of a product from reports generated.
- The System should have functionality to address the location of products in the warehouse.
- The user can add or update customer details.
- The user can Add or update vendor details and can place stock orders of products.
- The System must provide functionality to Add or update employee details.
- The system should have a function to Generate the reports of underperforming stocks.
- Users can generate bills or invoices for products sold, and based on that, the system deducts the corresponding amount of product quantity from the inventory.
- Users can make payment transactions in different methods.
- The system must allow the user to create orders for the products that are above threshold.
- The System Must provide functionalities to generate reports of employees,products and customers.

3.3 Nonfunctional Requirements:

- We divide this section into three parts
  - Usability
  - Reliability
  - Performance
  - Supportability
  - Packaging

Usability :

- The system must be easy to use by users of all hierarchy (i.g. From operator to manager and owner) so that they don't need any special manual or training
- The system must be intuitive and simple in the way that it presents all relevant data and relationship
- The menus of the system must be efficiently navigable by the users with keys that can be easy to understand

Reliability :

- The System must provide correct inventory status to the user continuously. Any errors are taken care through the regular establishment of the actual levels with the levels displayed in the system.
- The System must successfully add any product, vendors, or specific incidents given by the customers and present estimations and inventory status in significance with the newly updated items.
- The system shouldn't update the database for any failed processes
- The system should give the user updates on the fulfillment of processes and if the requested processes fail then it should provide the customer the reason for the failure.

Performance :

- All the functions of the system must be available to the user each time the system starts.
- The calculations executed through the system must comply with all the norms set by the user and should not vary unless explicitly changed by the user.
- The system must complete updating the databases, adding products, sellers, and occasions successfully each time the customer requests such a process.

Supportability :

- The software is created such that it operates even on systems having the minimum configuration.

  - The device is adaptable even supposing new plugins or modules are delivered at a later point.

    Packaging:

  - The software must incorporate a license key authentication process.
  - The system must be able to run on all OS like windows,linux etc. In case of Windows it should be capable with WindowsXP, and also able to compile on future releases such as the upcoming Windows.

3.4 Stakeholders characteristics:

- Business Owner:
  - The Business Owner can create a warehouse and update information about the warehouse.The Business owner has functionalities to hire employees for the warehouse.The Business Owner can demand the reports of products ,employees and customers from the manager.The Business owner can set or change the price of product.

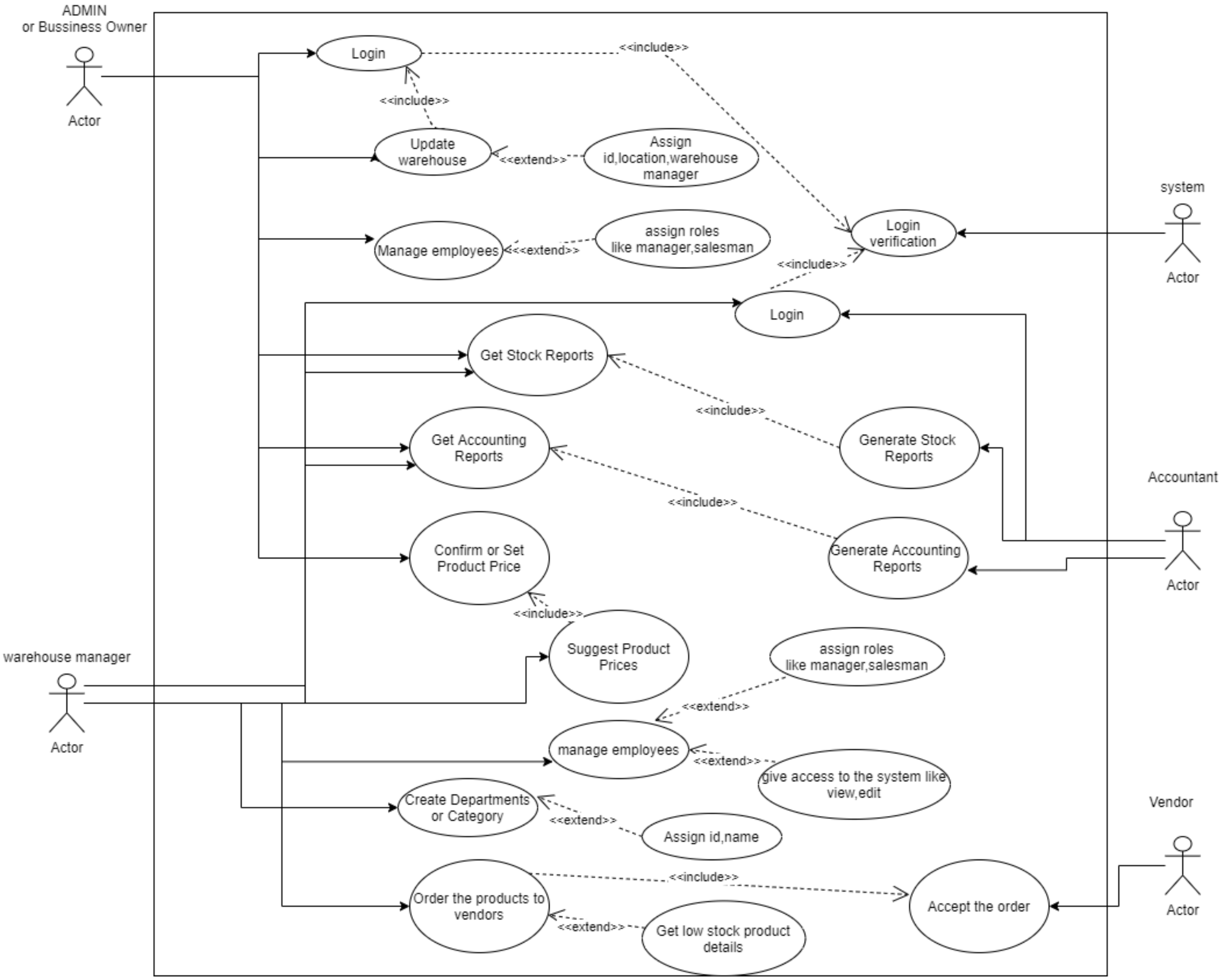
- Warehouse manager:
  - The Warehouse manager using a system can assign different roles to different employees.The manager is able to generate reports on products,customers and employees and can make subtle decisions.

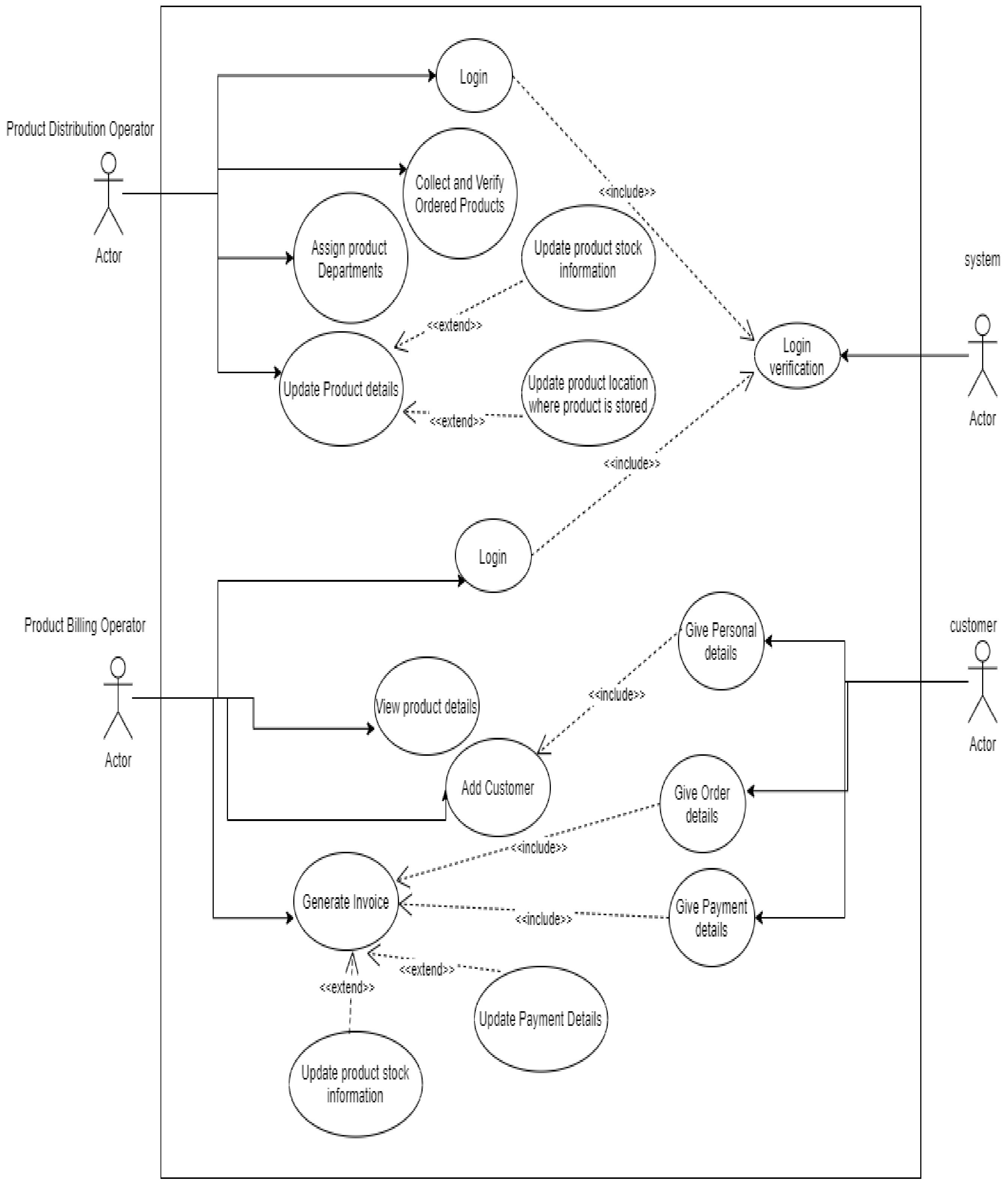
- Product Distribution operator :
  - The Product distribution operator collects the ordered stock of products from vendors and verifies the quantity.The product distribution operator can generate barcodes of the product to get the details of the product instantly and put products in a rack and can update the stock of the products.

- Product billing operator:
  - Users can take the customer's information and order information.User check the availability of the products and view the prices of the products.Based on order, generate the invoice,update payment details and update product stock information.

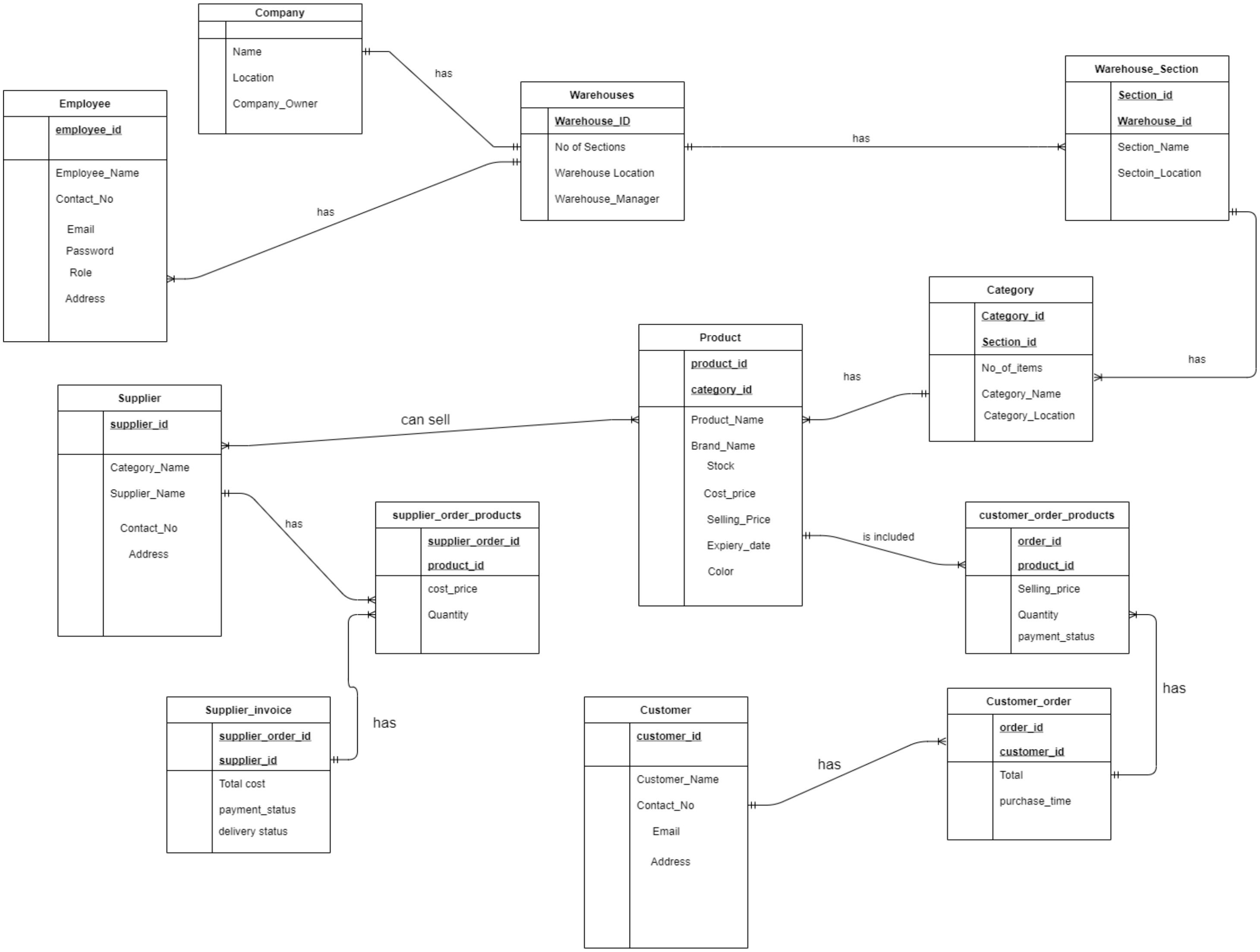
- Accountant:
  - The Employee has to perform visualization and derive analytics based on the present database.The Accountant collects invoices product stock purchased,product sold invoice,employee salary and turnover of company.

3.5 Use case diagram:



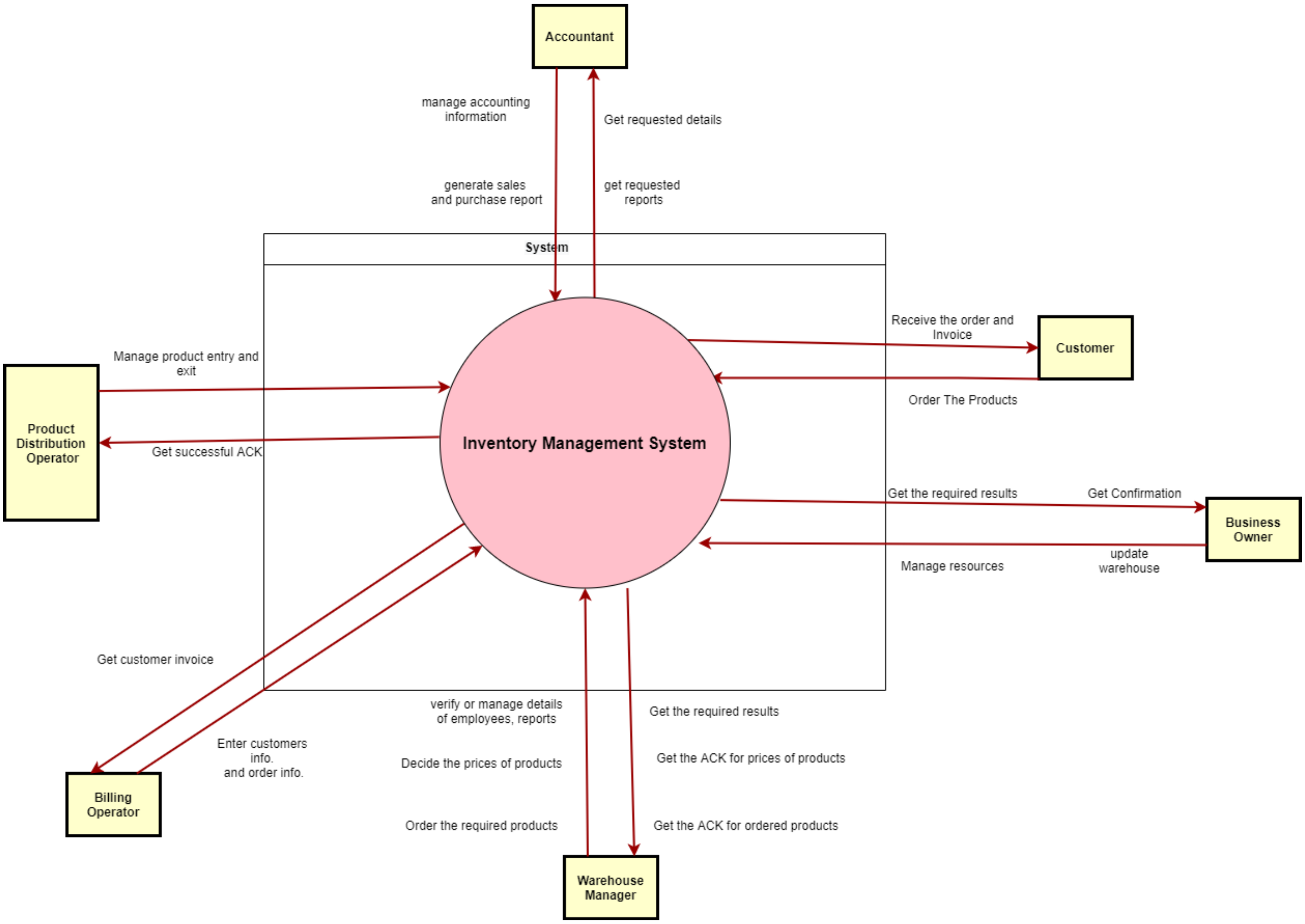

Product Distribution Operator
Actor
Login
Collect and Verify Ordered Products
Assign product Departments
Update product stock information
Update Product details
Update product location where product is stored
<<include>>
<<extend>>
<<extend>>
system
Login verification
Actor
<<include>>
Login
Product Billing Operator
Actor
Give Personal details
customer
Actor
<<include>>
View product details
Add Customer
Give Order details
<<include>>
Generate Invoice
<<include>>
Give Payment details
<<extend>>
<<extend>>
Update Payment Details
Update product stock information

3.6 ER diagram:

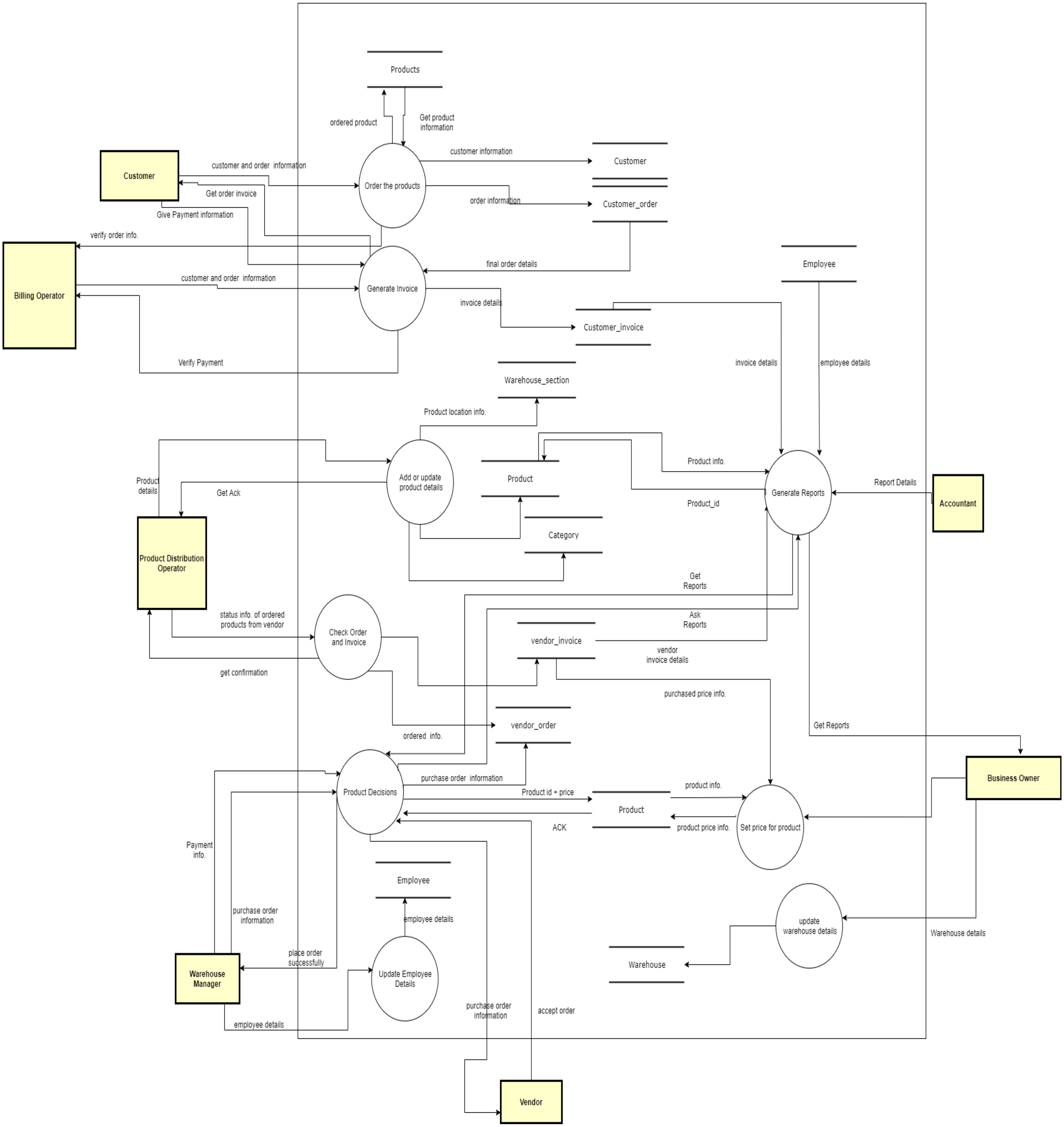


3.7 Data flow diagram:

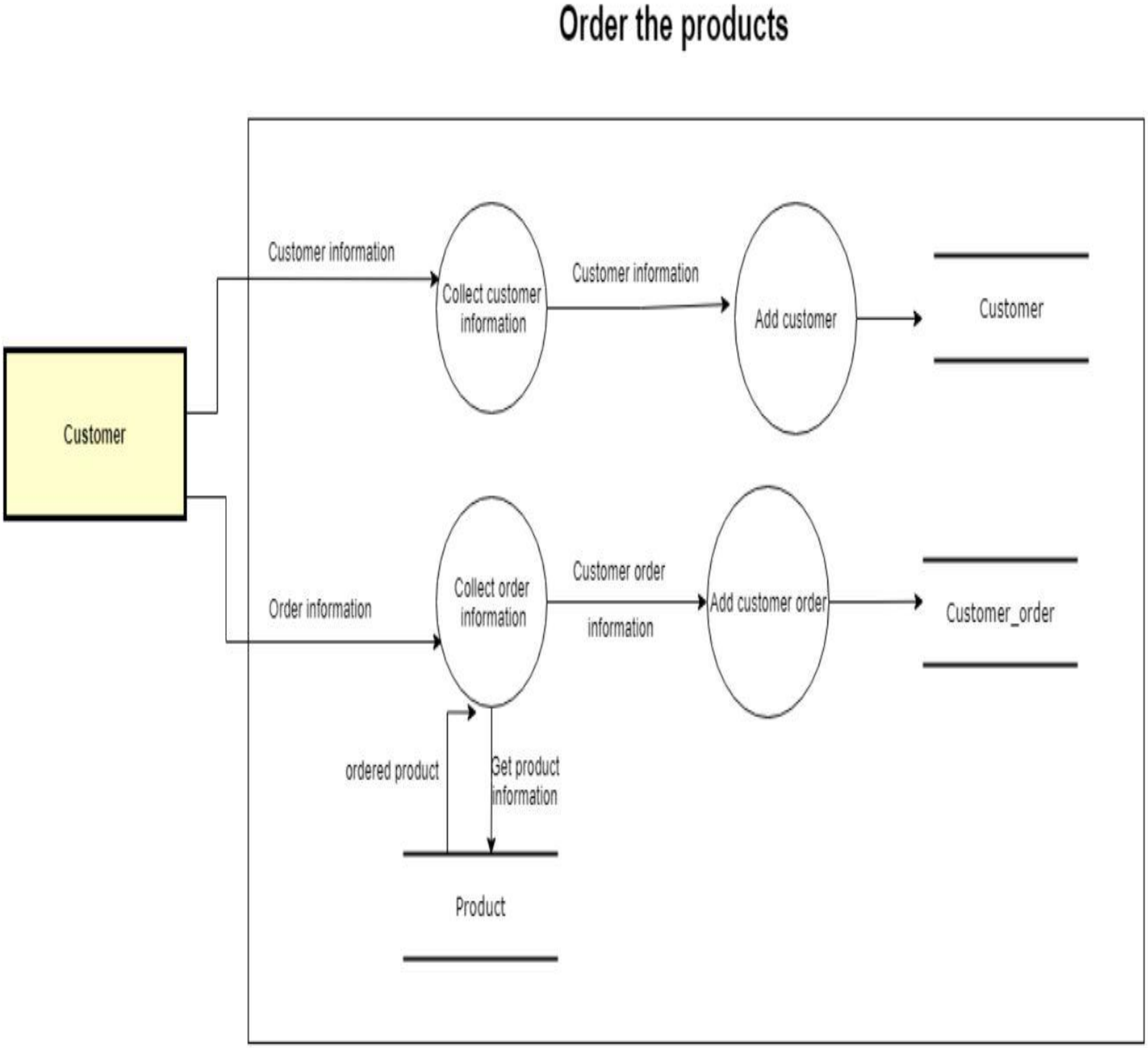
Level 0 :


Accountant
manage accounting information
generate sales and purchase report
Get requested details
get requested reports
System
Inventory Management System
Product Distribution Operator
Manage product entry and exit
Get successful ACK
Customer
Receive the order and Invoice
Order The Products
Business Owner
Get the required results
Get Confirmation
Manage resources
update warehouse
Billing Operator
Get customer invoice
Enter customers info. and order info.
Warehouse Manager
verify or manage details of employees, reports
Decide the prices of products
Order the required products
Get the required results
Get the ACK for prices of products
Get the ACK for ordered products

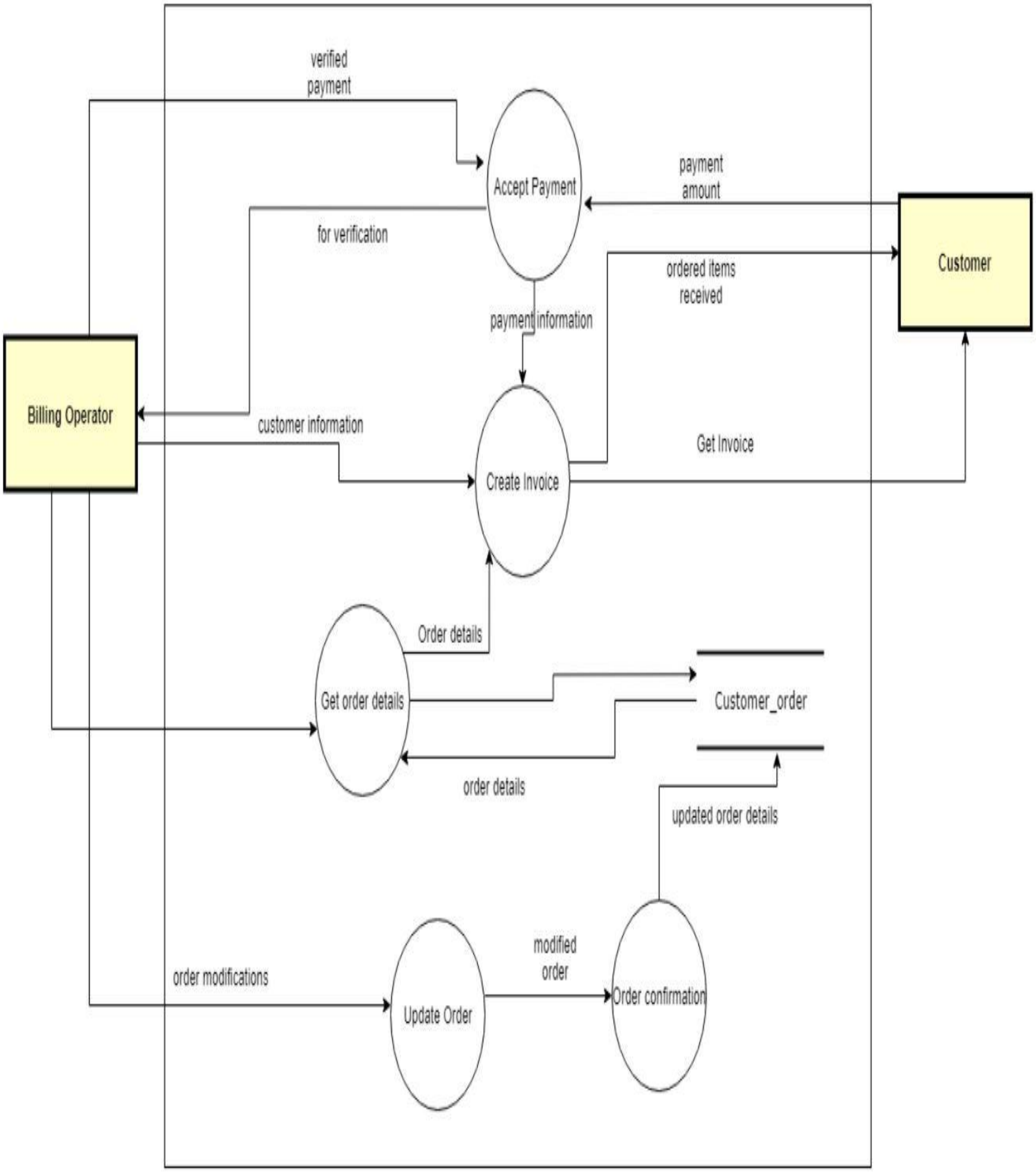
Level 1 :

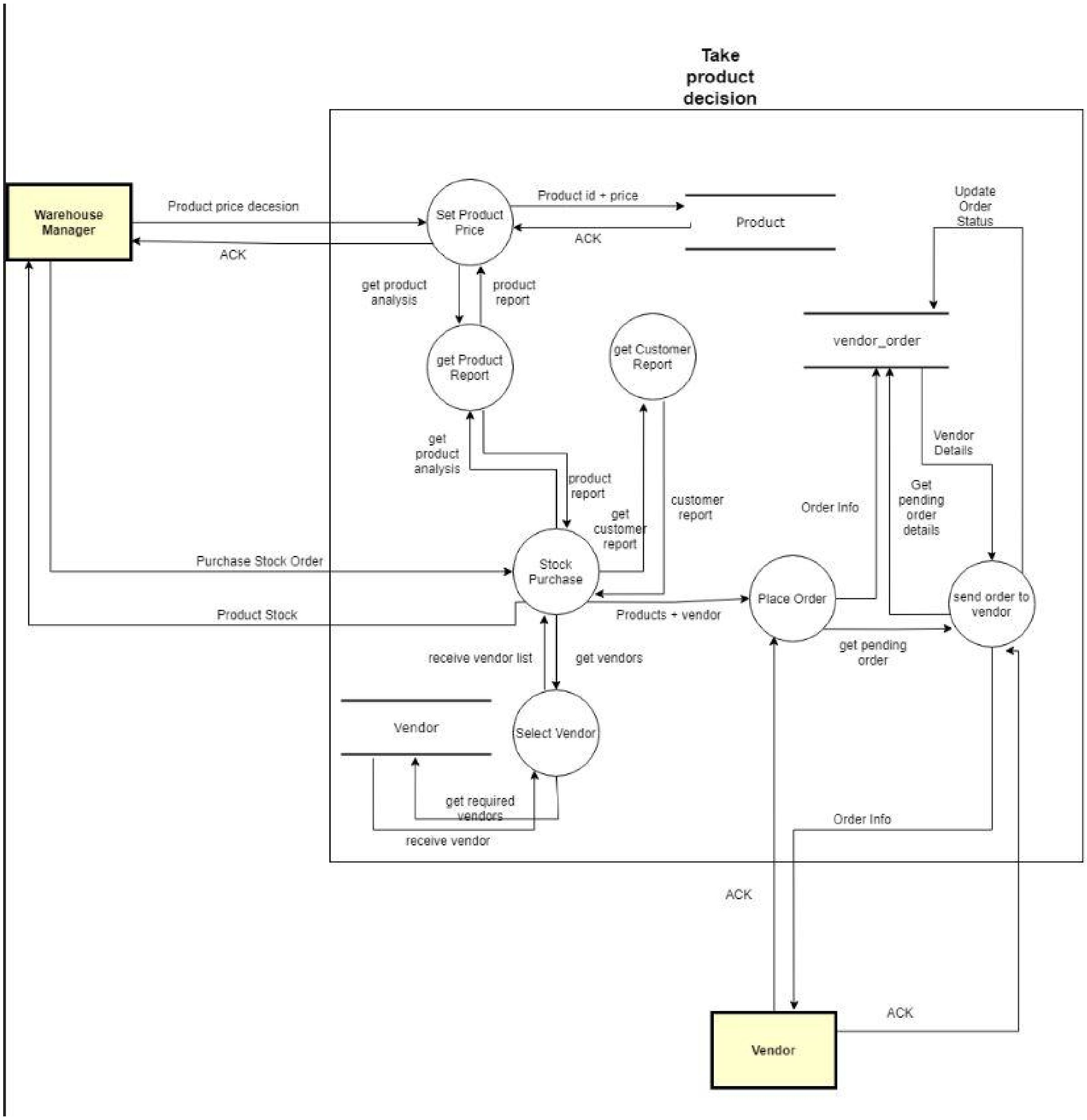

Products
ordered product
Get product information
Customer
customer and order information
Get order invoice
Give Payment information
Order the products
customer information
Customer
order information
Customer_order
verify order info.
Billing Operator
customer and order information
Generate Invoice
final order details
Employee
invoice details
Customer_invoice
Verify Payment
invoice details
employee details
Warehouse_section
Product location info.
Product info.
Product details
Get Ack
Add or update product details
Product
Generate Reports
Report Details
Accountant
Product_id
Category
Product Distribution Operator
Get Reports
status info. of ordered products from vendor
Check Order and Invoice
Ask Reports
vendor_invoice
vendor invoice details
get confirmation
purchased price info.
vendor_order
Get Reports
ordered info.
Business Owner
purchase order information
Product Decisions
Product id + price
product info.
Product
Payment info.
ACK
product price info.
Set price for product
Employee
employee details
purchase order information
update warehouse details
Warehouse details
Warehouse
place order successfully
Warehouse Manager
Update Employee Details
purchase order information
accept order
employee details
Vendor

Level2 :

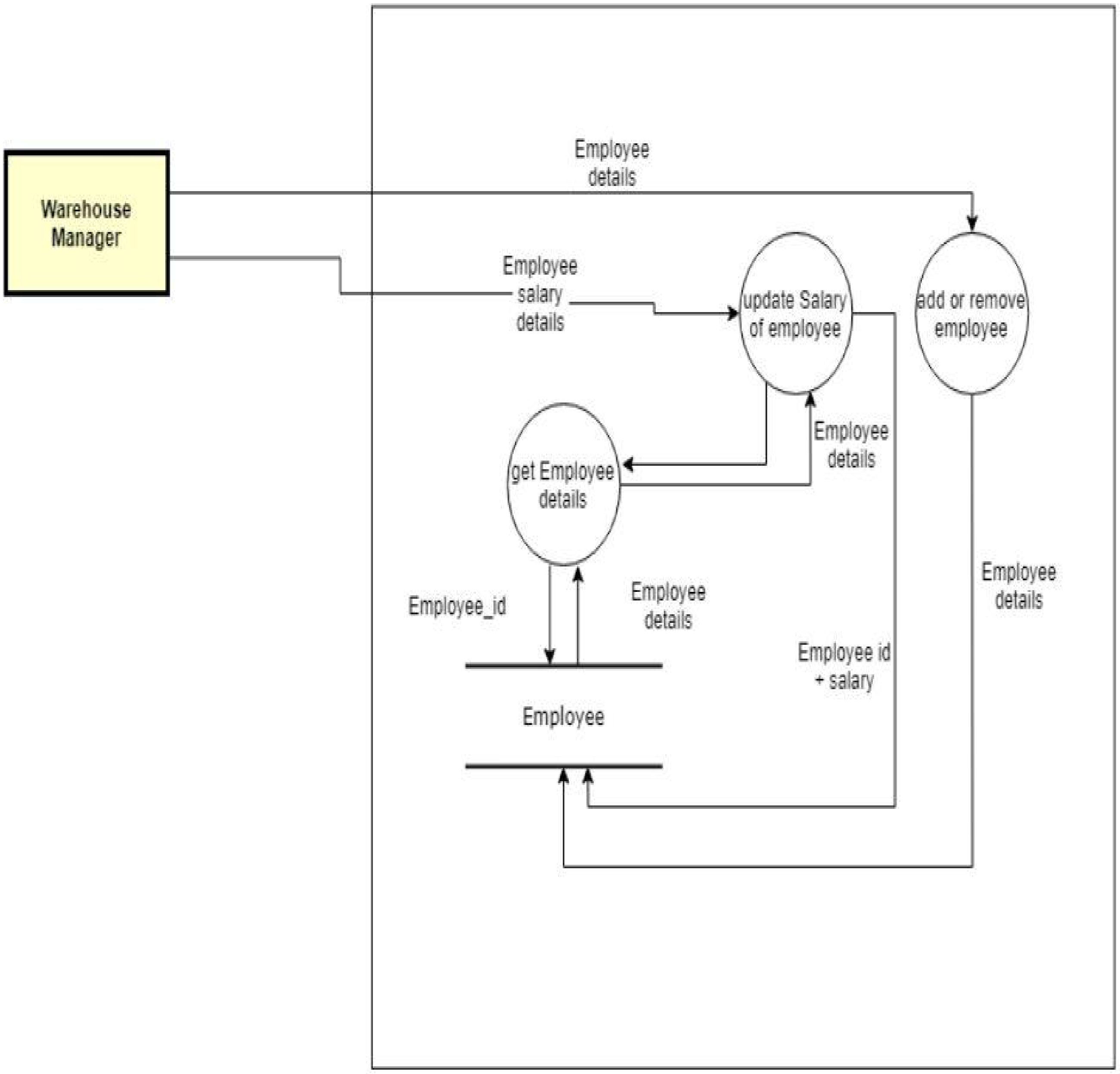

Order the products
Customer information
Collect customer information
Customer information
Add customer
Customer
Customer
Order information
Collect order information
Customer order information
Add customer order
Customer_order
ordered product
Get product information
Product

Generate Invoice

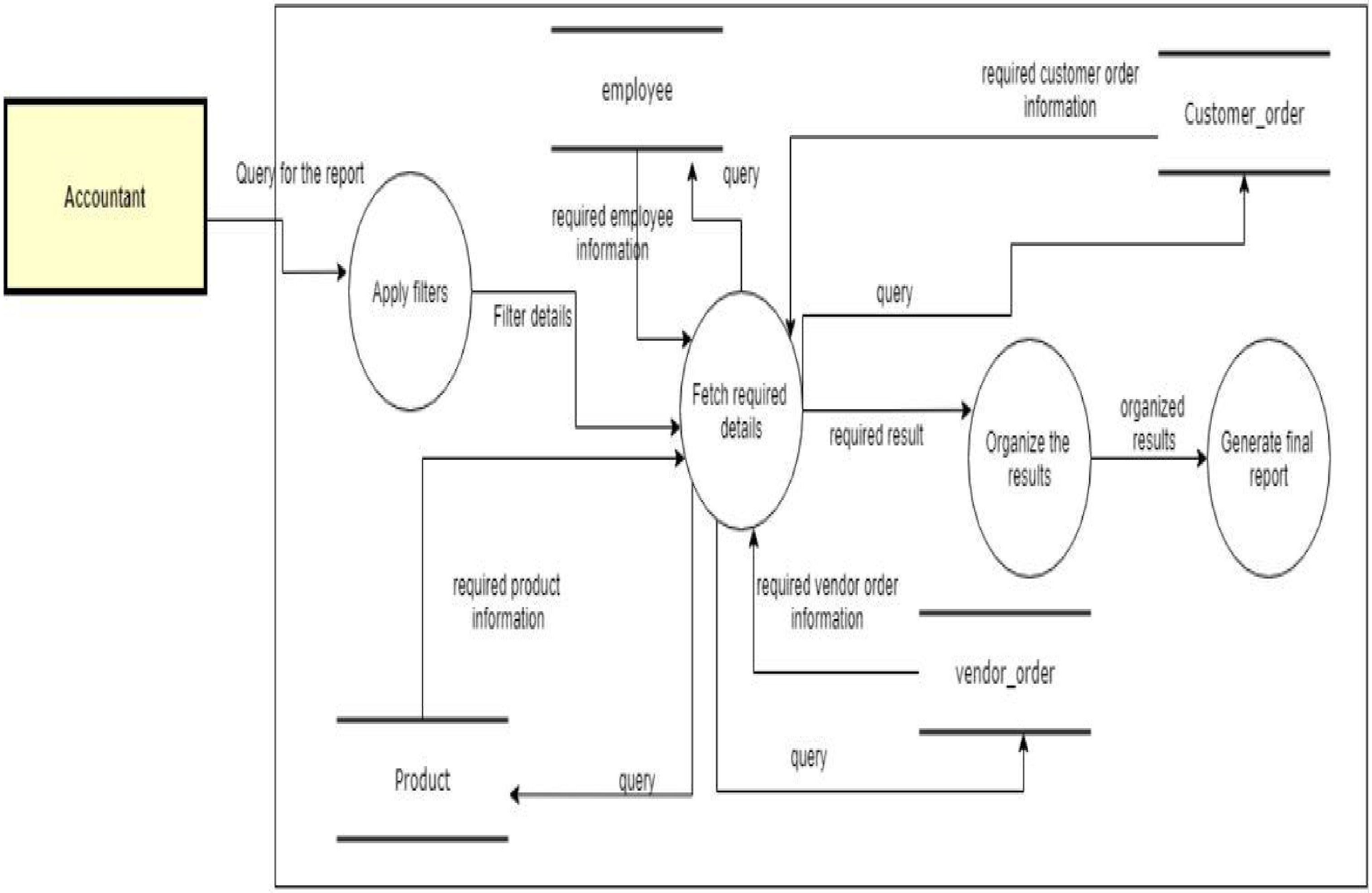



Take product decision
Warehouse Manager
Product price decesion
ACK
Set Product Price
Product id + price
ACK
Product
Update Order Status
get product analysis
product report
get Product Report
get Customer Report
vendor_order
get product analysis
product report
get customer report
customer report
Vendor Details
Get pending order details
Order Info
Purchase Stock Order
Stock Purchase
Place Order
send order to vendor
Product Stock
Products + vendor
get pending order
receive vendor list
get vendors
Vendor
Select Vendor
get required vendors
receive vendor
Order Info
ACK
ACK
Vendor

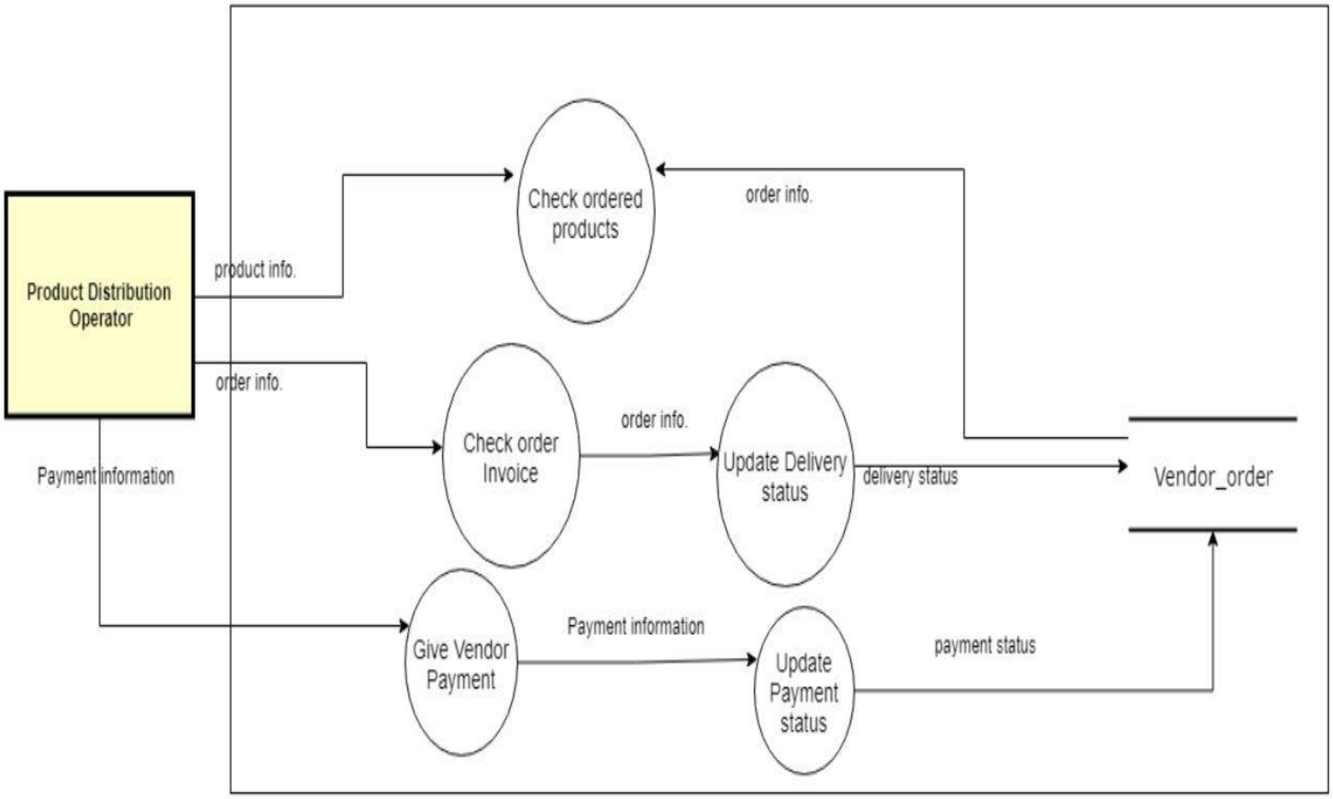
Update Employee Details



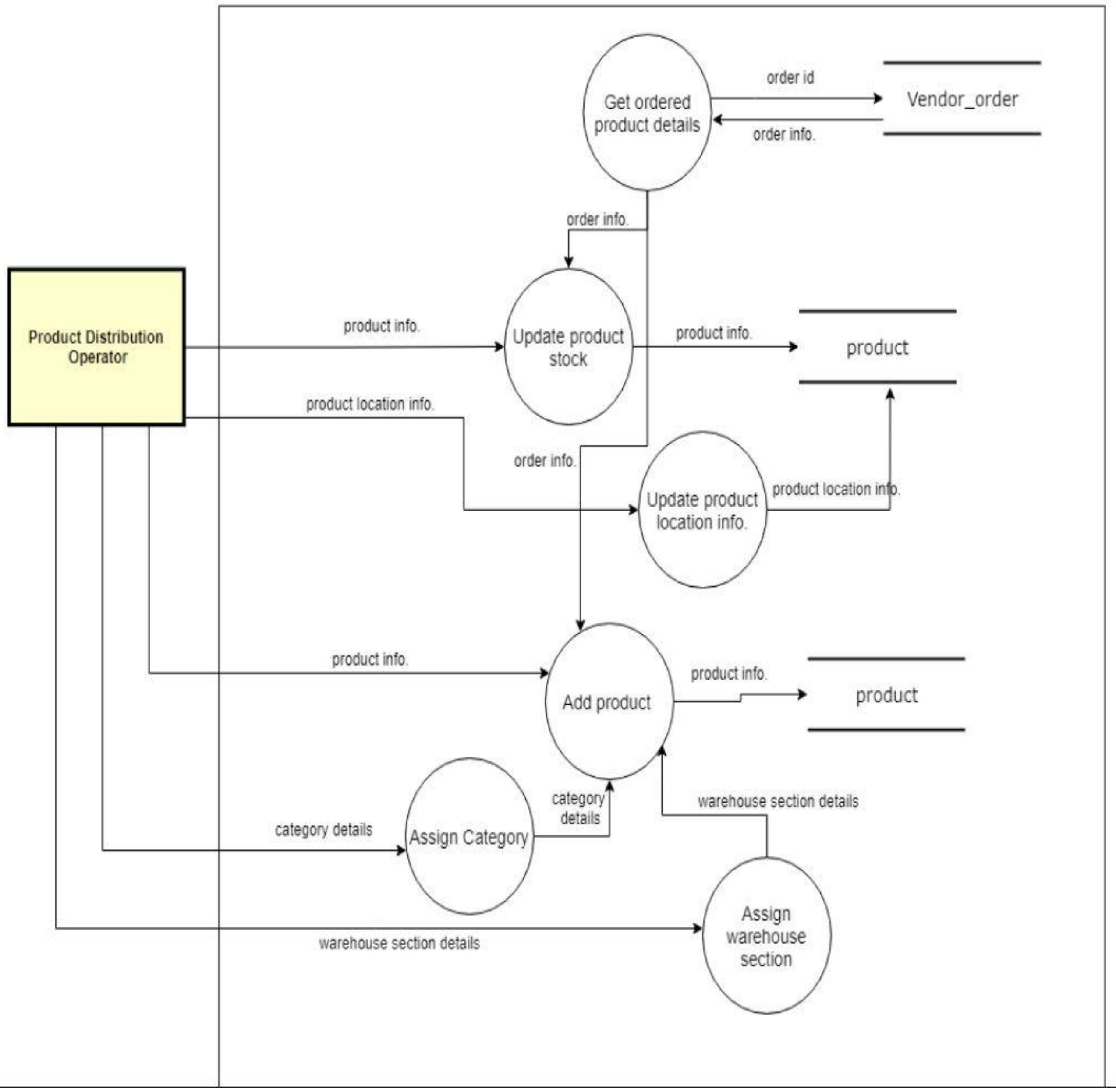
Generate Reports



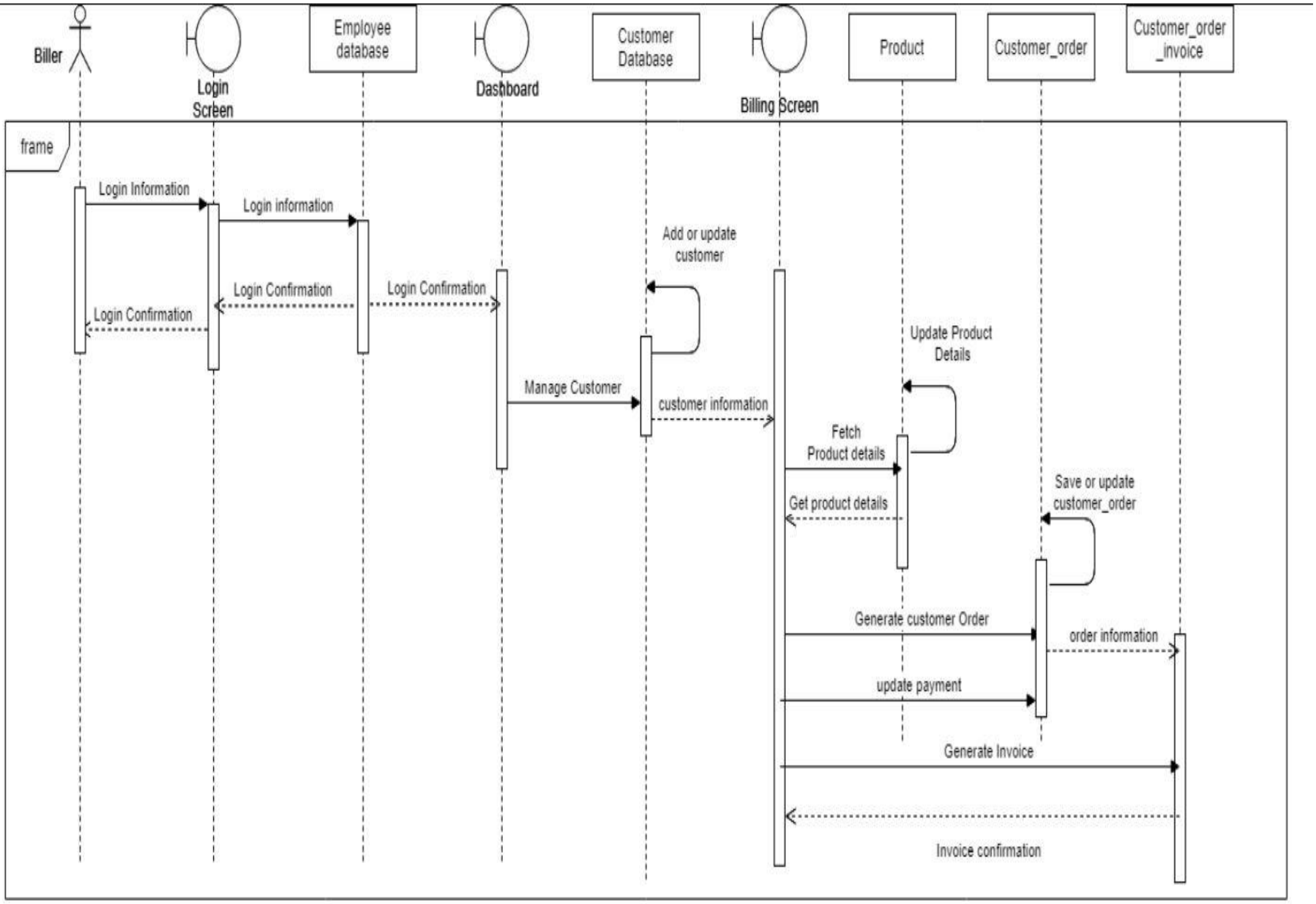
Check order and Invoice

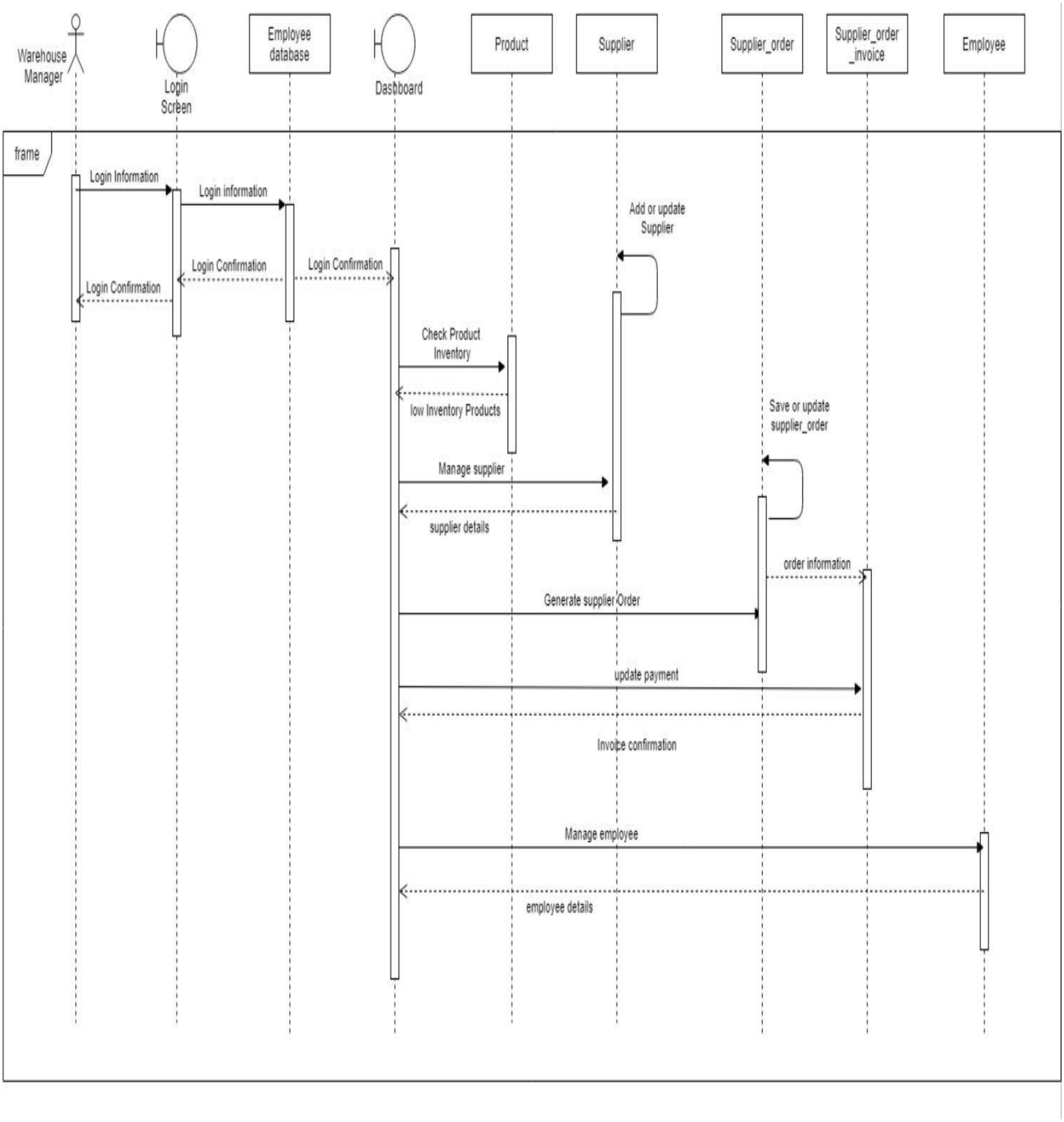


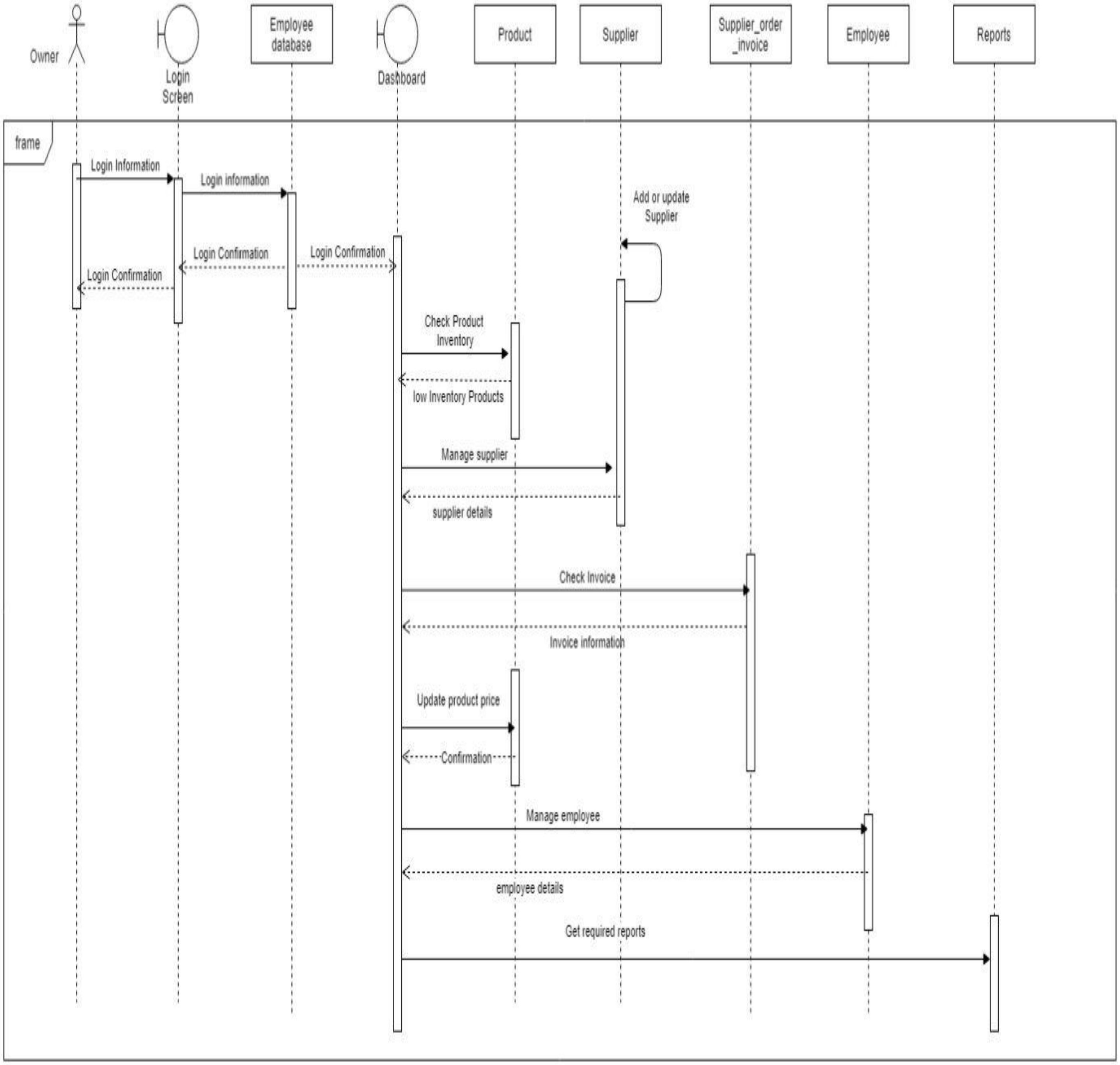
add/update products

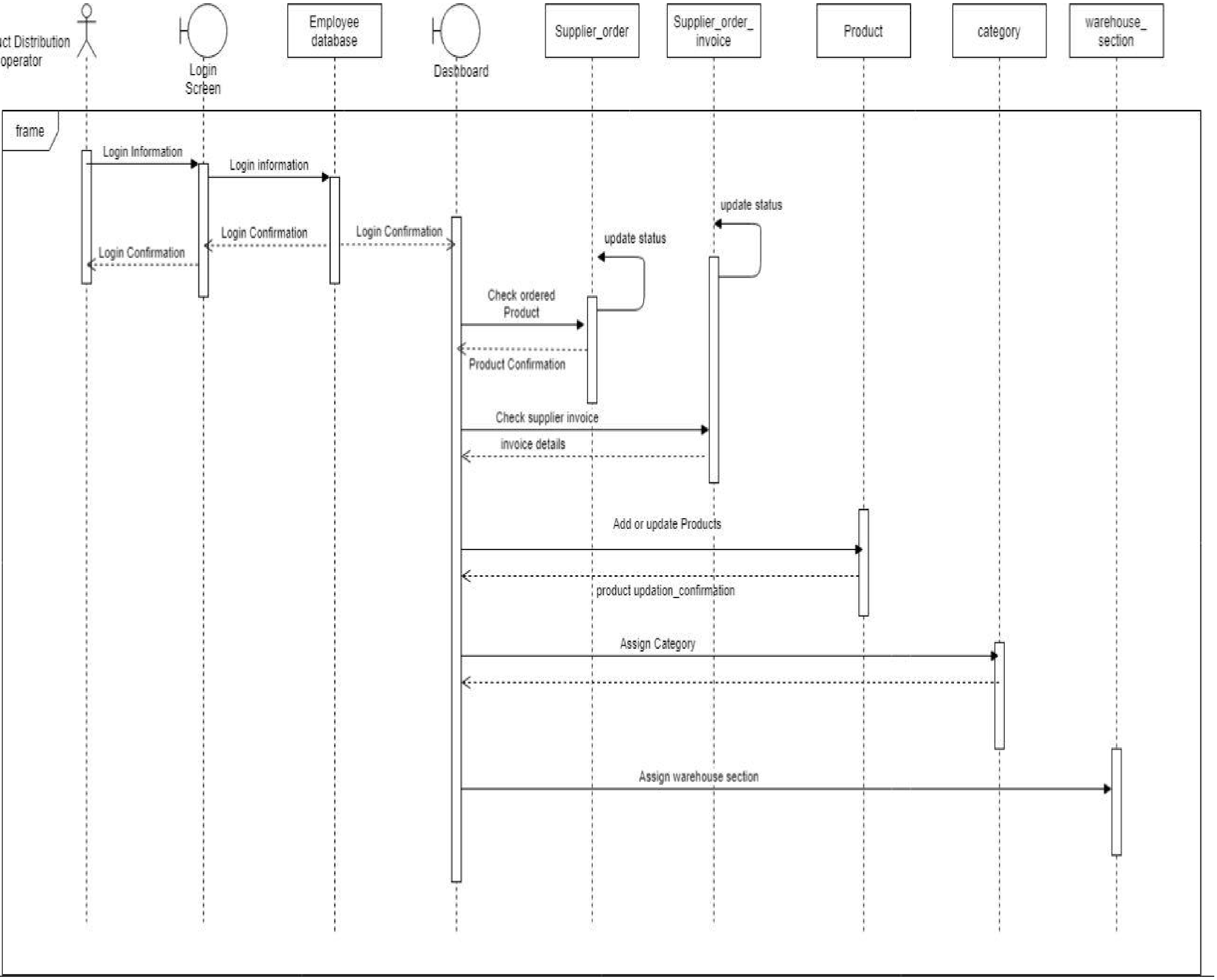


3.8 Sequence diagram:




Warehouse Manager
Login Screen
Employee database
Dashboard
Product
Supplier
Supplier_order
Supplier_order _invoice
Employee
frame
Login Information
Login information
Add or update Supplier
Login Confirmation
Login Confirmation
Login Confirmation
Check Product Inventory
low Inventory Products
Save or update supplier_order
Manage supplier
supplier details
order information
Generate supplier Order
update payment
Invoice confirmation
Manage employee
employee details


Owner
Login Screen
Employee database
Dashboard
Product
Supplier
Supplier_order _invoice
Employee
Reports
frame
Login Information
Login information
Add or update Supplier
Login Confirmation
Login Confirmation
Login Confirmation
Check Product Inventory
low Inventory Products
Manage supplier
supplier details
Check Invoice
Invoice information
Update product price
Confirmation
Manage employee
employee details
Get required reports


ct Distribution
operator
Login
Screen
Employee
database
Dashboard
Supplier_order
Supplier_order_
invoice
Product
category
warehouse_
section
frame
Login Information
Login information
update status
Login Confirmation
Login Confirmation
update status
Login Confirmation
Check ordered
Product
Product Confirmation
Check supplier invoice
invoice details
Add or update Products
product updation_confirmation
Assign Category
Assign warehouse section

4 Design Documentation

4.1 Purpose of document

The Purpose of this document is to provide a description of design of systems which provide help for software development and to carry forward with an understanding of what to build and how to build. This helps in getting a clear understanding of the project requirements and streamlining the design implementation.

4.2 Why website?

- ➔ A website gives a better reach and greater accessibility because it does not require any special installation needs and can be easily accessed by one's phone or desktop browser.
- ➔ It is easy to fix the bugs and updates in a website. Also, the users would not have to notice the update process, which is a plus point.

4.3 language and tools:

- **Languages:**

➢ Javascript
➢ Html
➢ CSS

- **Tools and Libraries:**

➢ Visual Studio Code
➢ React Js
➢ Node Js
➢ Draw.io
➢ Figma
➢ route
➢ react-router-dom
➢ Material UI(css)
➢ Bootstrap react-bootstrap(css)
➢ reactstrap(css)
➢ font awesome icons and fonts(css)

- **Databases:**

➢ Google Firebase

4.4 Tools and Technologies and its description.

Sr.no	Software	Description
1	Node JS	Back-end runtime Javascript environment.
2	React JS	Frontend Javascript Library to make User interfaces components.
3	Firebase	Real time database.
4	Github	Version Control.
5	react-router-dom	Handles routes in webapp.
6	Bootstrap react-bootstrap	User interface framework.
7	route	Standard routing library.
8	reactstrap	Bootstrap Components.
9	font awesome icons and fonts(css)	Structuring web pages.
10	Material UI	User interface framework.

		Description
Programming language	Java Script	JavaScript to program the behavior of web pages.It is used for creating web pages dynamically.
Frontend framework	React JS	React js is frontend javascript which helps to design the user interfaces.
Backend Framework	Node JS	Node is Backend Javascript runtime environment Which runs javascript on the server side to produce dynamic applications.
Database	FireBase	The Firebase is a Realtime Database which is a cloud-hosted database. Data is stored as JSON format and synchronized in realtime to every connected client.

4.5 Design Consideration

- Usability - How easily can a specific user in defined conditions be able to use the User interface correctly.

- Performance - The software must be able to perform in bounded time and using limited memory.

- Modularity - The software should contain well defined individual components to perform testing in isolation before being integrated.

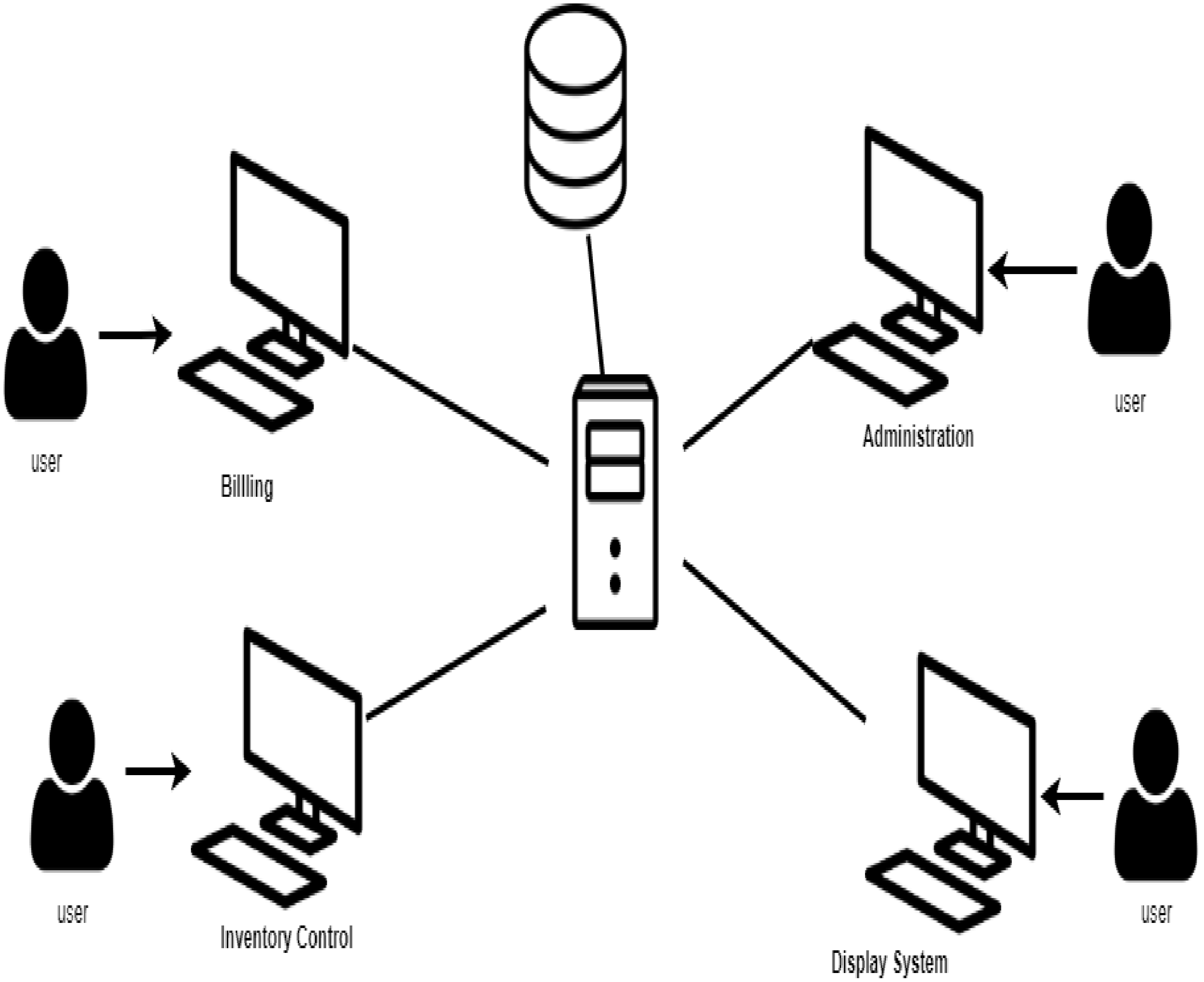
- Extensibility - The software must be able to add new changes in it without changing the structure.

- Correctness :Software design has to be correct as per requirement document.

- Maintainability: The design should be so simple that other designers can understand and maintain it easily.

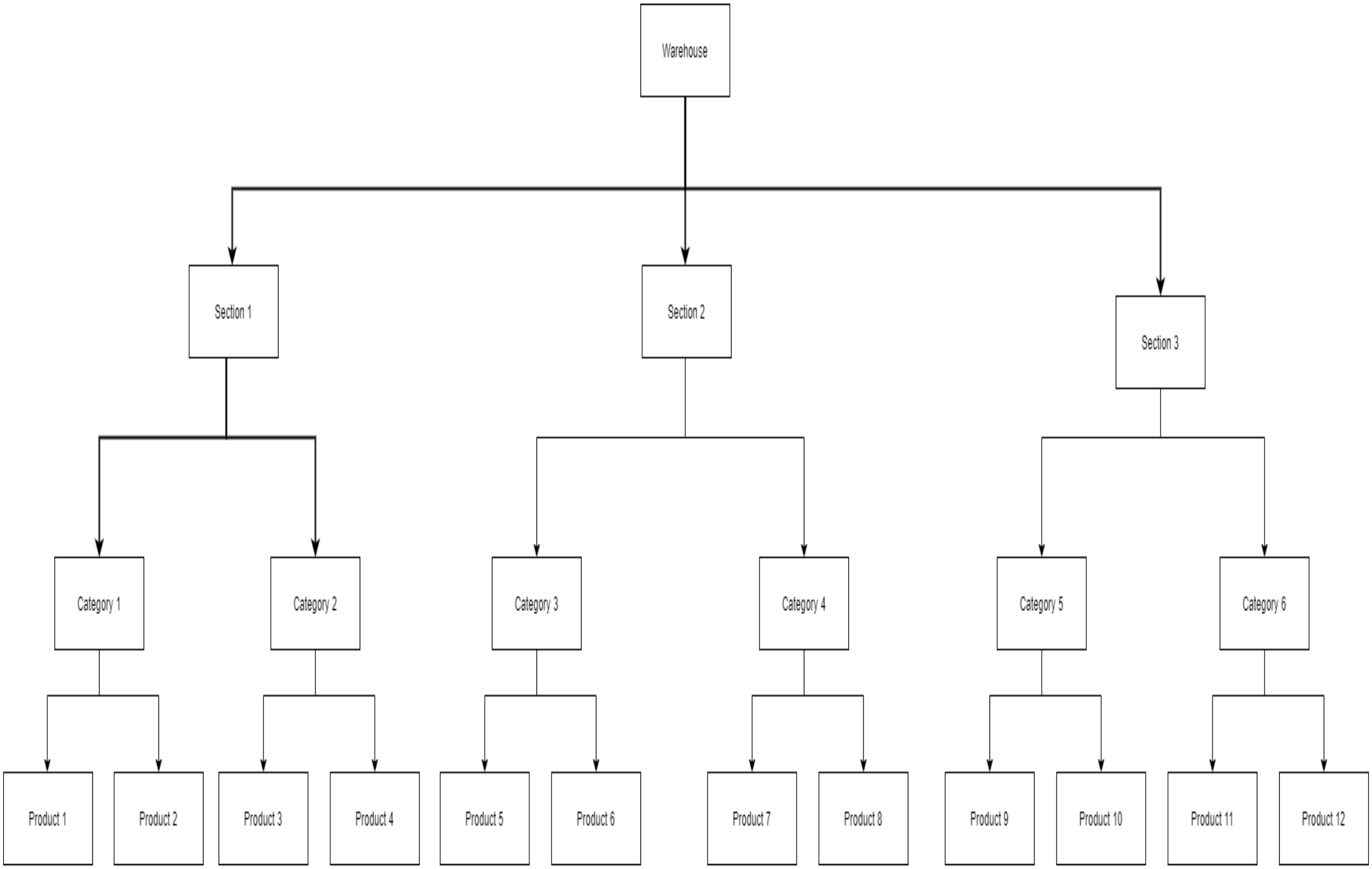
4.6 System Architecture and Design

- Network Architecture Diagram



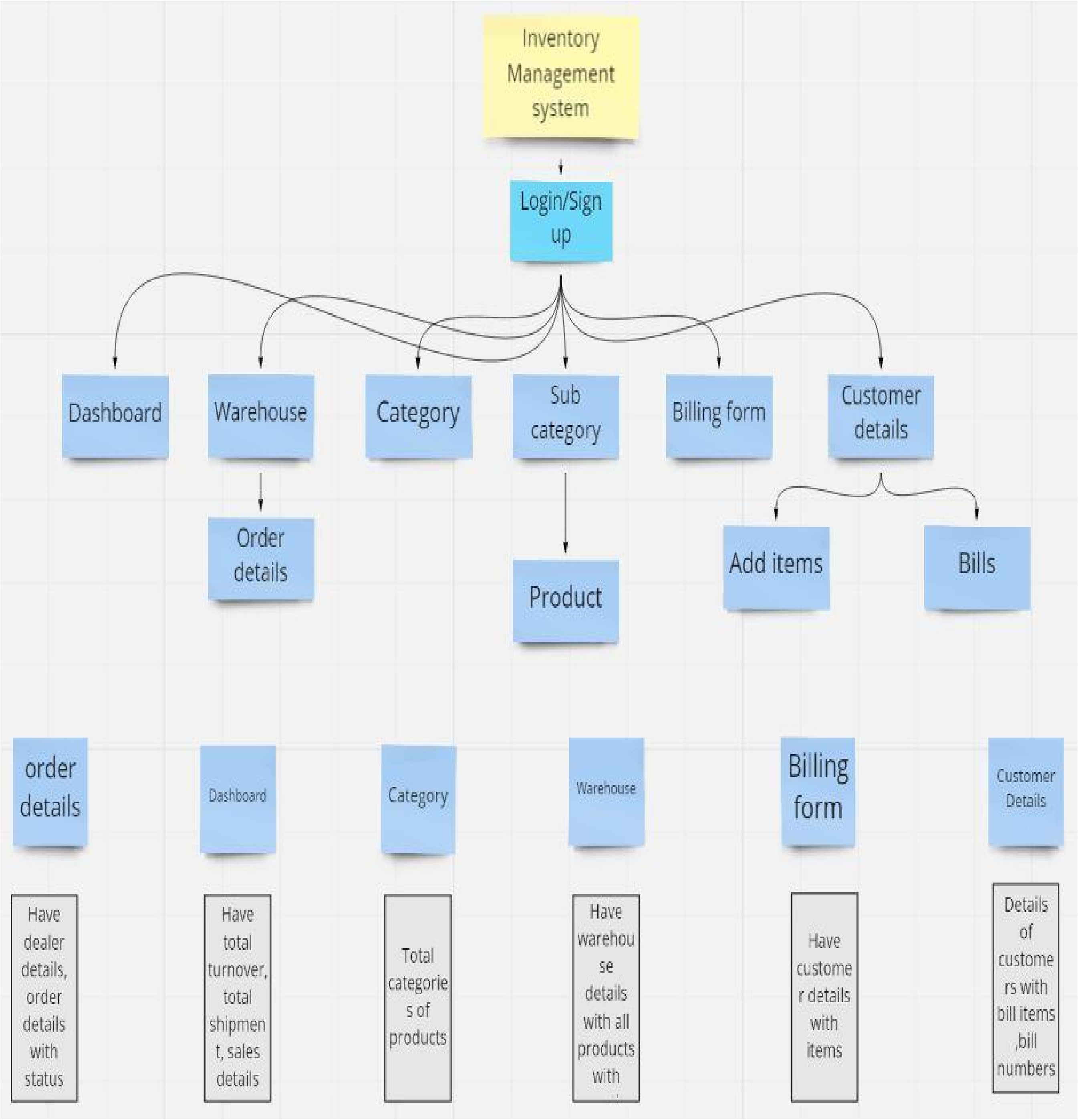
The system has been designed to be used by different users simultaneously. The system has been deployed on the server and users can access it through url.Database is used to store the information and provide data to the server for functionalities.

- **Warehouse Architecture**

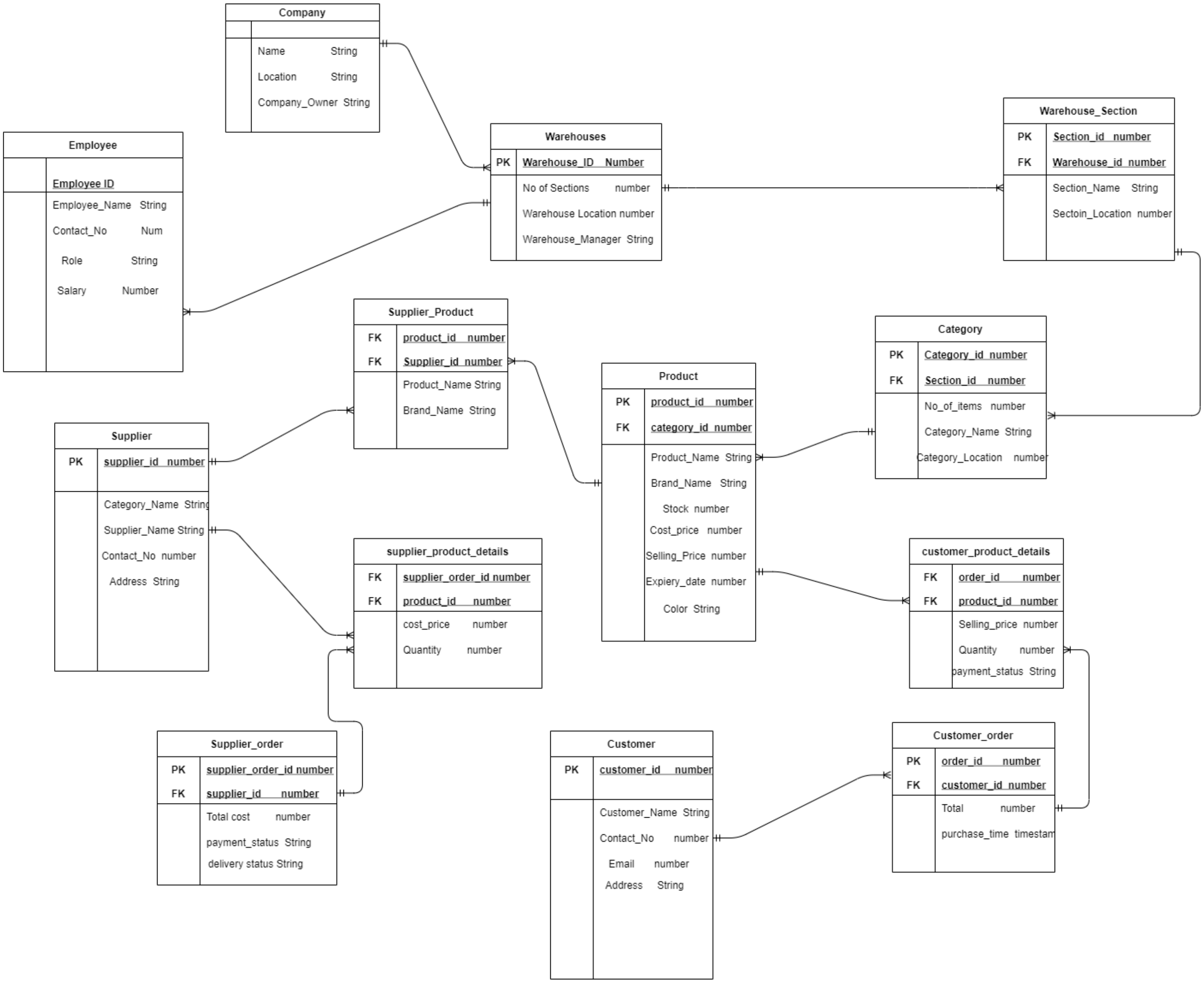


We have proposed a system to access and track the location of products easily.we have derived a warehouse architecture that a particular warehouse has different sections for category and each category has many sub categories.each subcategory assigned to section where products are placed and grouped on the basis of category.

- **Website architecture design**


Inventory Management system
Login/Sign up
Dashboard
Warehouse
Category
Sub category
Billing form
Customer details
Order details
Product
Add items
Bills
order details
Dashboard
Category
Warehouse
Billing form
Customer Details
Have dealer details, order details with status
Have total turnover, total shipment, sales details
Total categories of products
Have warehouse details with all products with
Have customer details with items
Details of customers with bill items ,bill numbers

4.7 System Design:



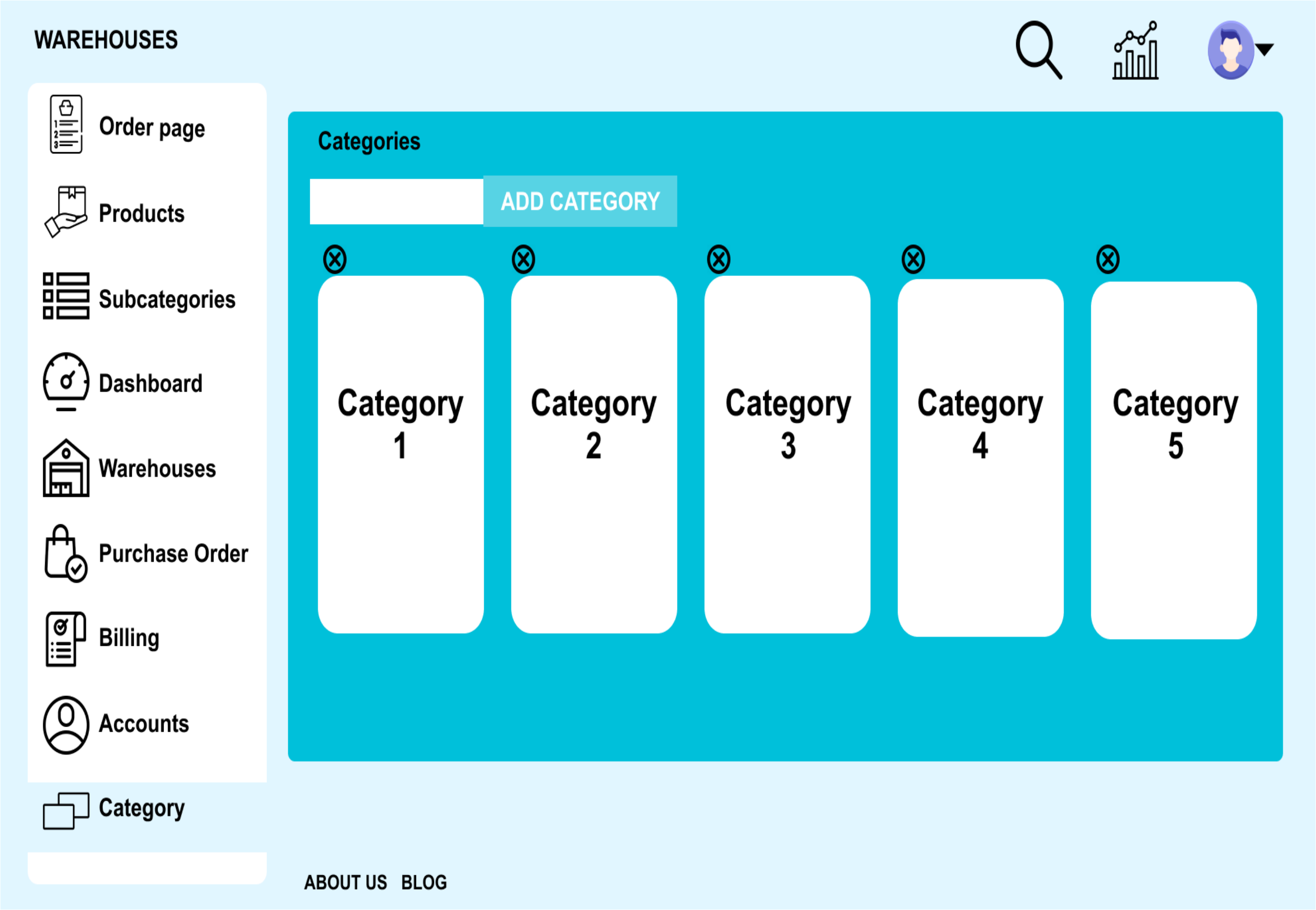
Cloud Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in documents, which are organized into collections.

Each document contains a set of key-value pairs. Cloud Firestore is optimized for storing large collections of small documents.

All documents must be stored in collections. Documents can contain subcollections and nested objects, both of which can include primitive fields like strings or complex objects like list

4.8 UI screen design:

(1)Display Category Screen



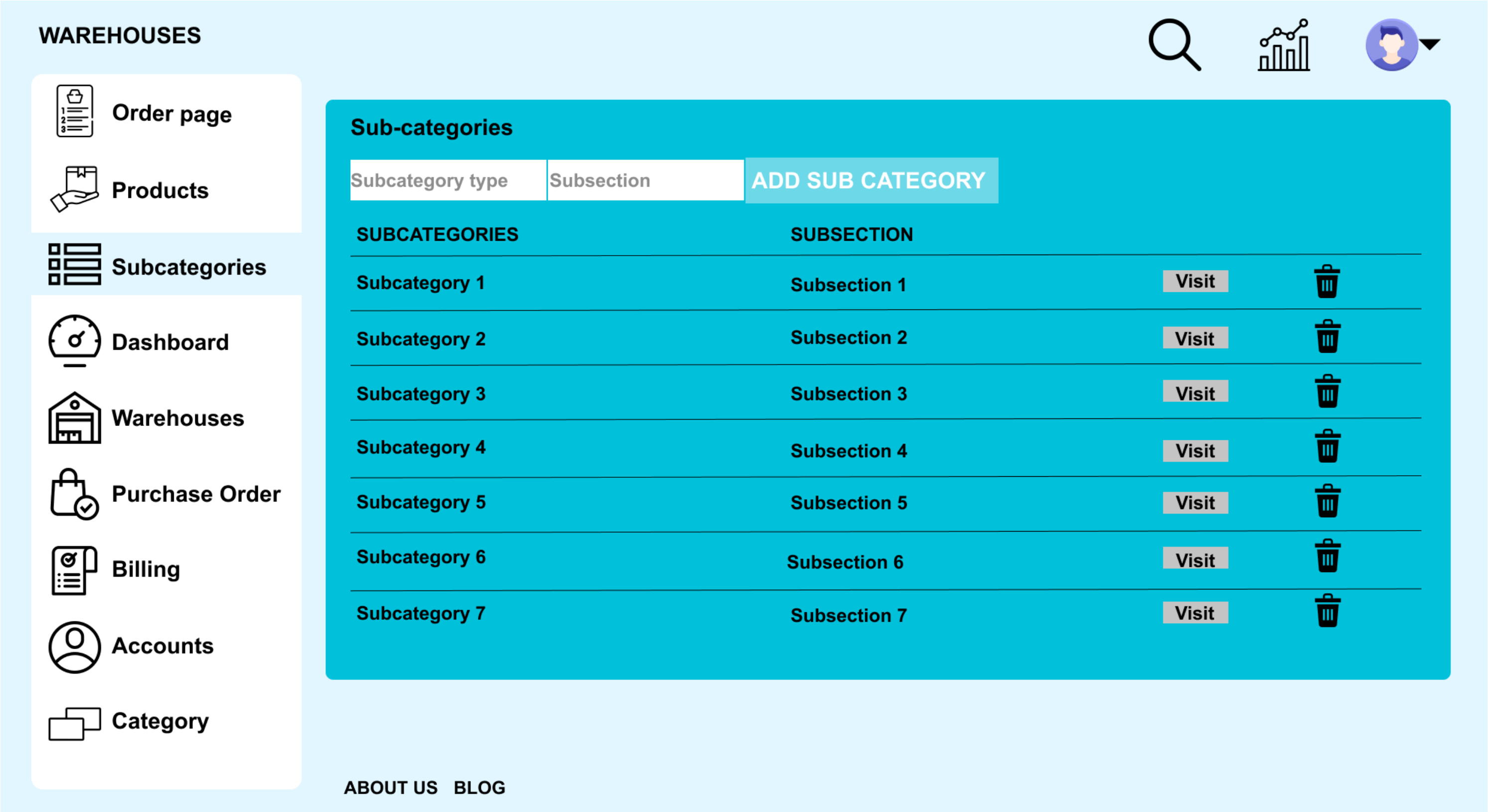
Add category:
When the new products arrive at the warehouse which do not belong to predefined categories then add a new category to the system for those products.

Delete Category :

When a particular category has no products available in the warehouse then that category will not be useful so we can delete it.

View Category:
One category has many sub categories(i.g. Electronic category has mobile,TV as a sub categories), so by view category we can see the sub-categories and their section id where the subcategories products are located.

(2) Show list of Subcategories and subsection.



SUBCATEGORIES	SUBSECTION		
Subcategory 1	Subsection 1	Visit	🗑
Subcategory 2	Subsection 2	Visit	🗑
Subcategory 3	Subsection 3	Visit	🗑
Subcategory 4	Subsection 4	Visit	🗑
Subcategory 5	Subsection 5	Visit	🗑
Subcategory 6	Subsection 6	Visit	🗑
Subcategory 7	Subsection 7	Visit	🗑

One category has many subcategories and it has section id which gives the information about which sub category product is located.

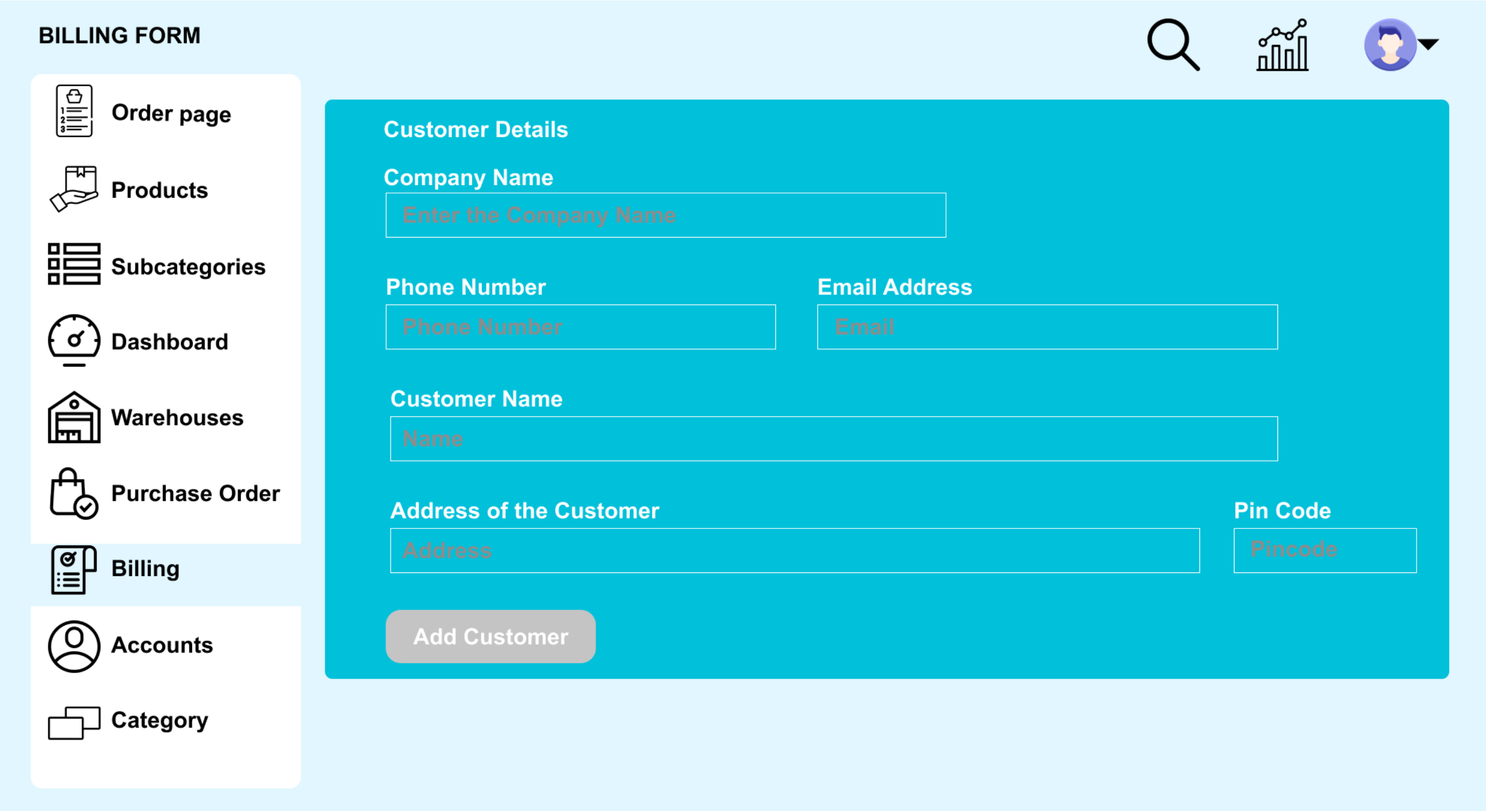
Add Subcategory:

When the new products arrive at the warehouse which do not belong to predefined sub categories then add a new sub category to the system for those products.

Delete Category :
When a particular sub category has no products available in the warehouse then that category will not be useful so we can delete it.

View sub Category:
One sub category has many products, so by view category we can see the products which are located in that category.

(3)Add Customer Details



Customer Details

Company Name: Enter the Company Name

Phone Number: Phone Number

Email Address: Email

Customer Name: Name

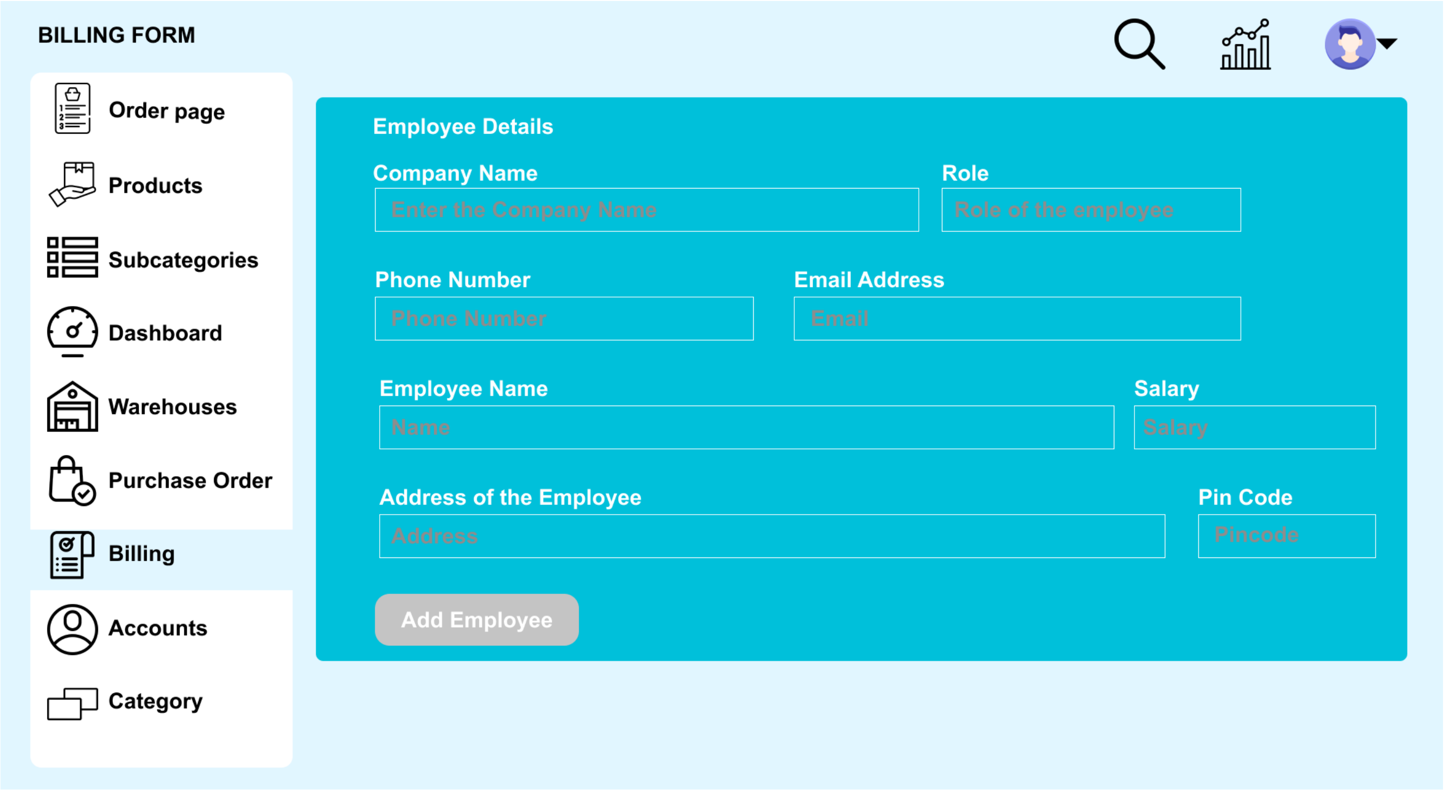
Address of the Customer: Address

Pin Code: Pincode

When a new customer arrives for billing then we have to register that customer to our system.

Add Customer:
To add the customer to our system, we need information like name,address,contact number,email address,pin code.

(4) Add Employee



Employee Details

Company Name: Enter the Company Name

Role: Role of the employee

Phone Number: Phone Number

Email Address: Email

Employee Name: Name

Salary: Salary

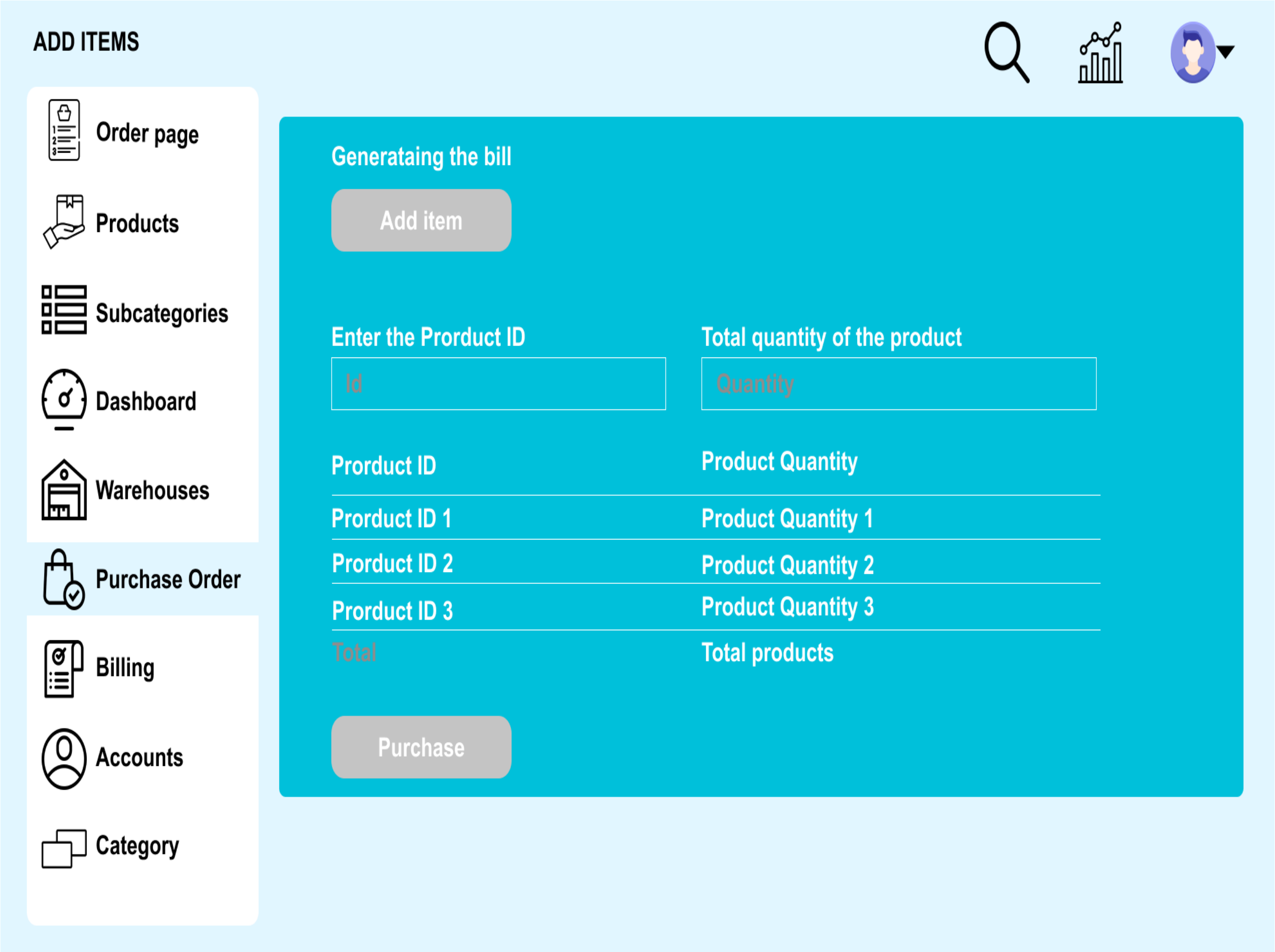
Address of the Employee: Address

Pin Code: Pincode

When a new employee is hired for the warehouse then we have to register that employee to our system.

Add Employee:
To add the employee to our system, we need information like name,address,contact number,email address,role,salary,pin code.

(5) Bill Screen

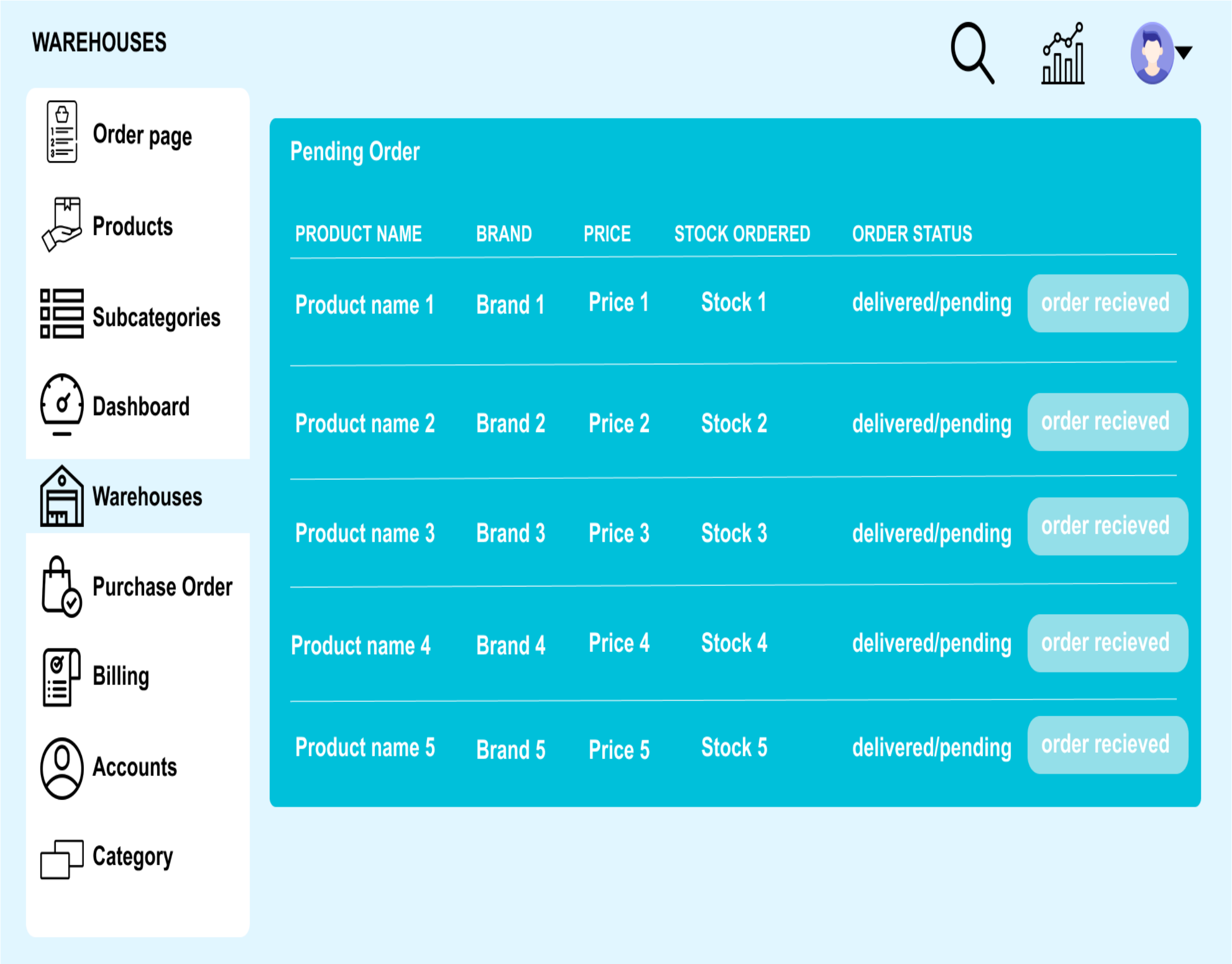


Prorduct ID	Product Quantity
Prorduct ID 1	Product Quantity 1
Prorduct ID 2	Product Quantity 2
Prorduct ID 3	Product Quantity 3
Total	Total products

When any customer comes to the billing counter at that time the employee at the billing counter adds the customer's product into the system and generates a bill for that customer.Once employee ads the all product customers bill will be generated with payable amount.

Add Bill : Add product by product id with its quantity and add all items which were bought by the customer.

(6) Stock order to receive.



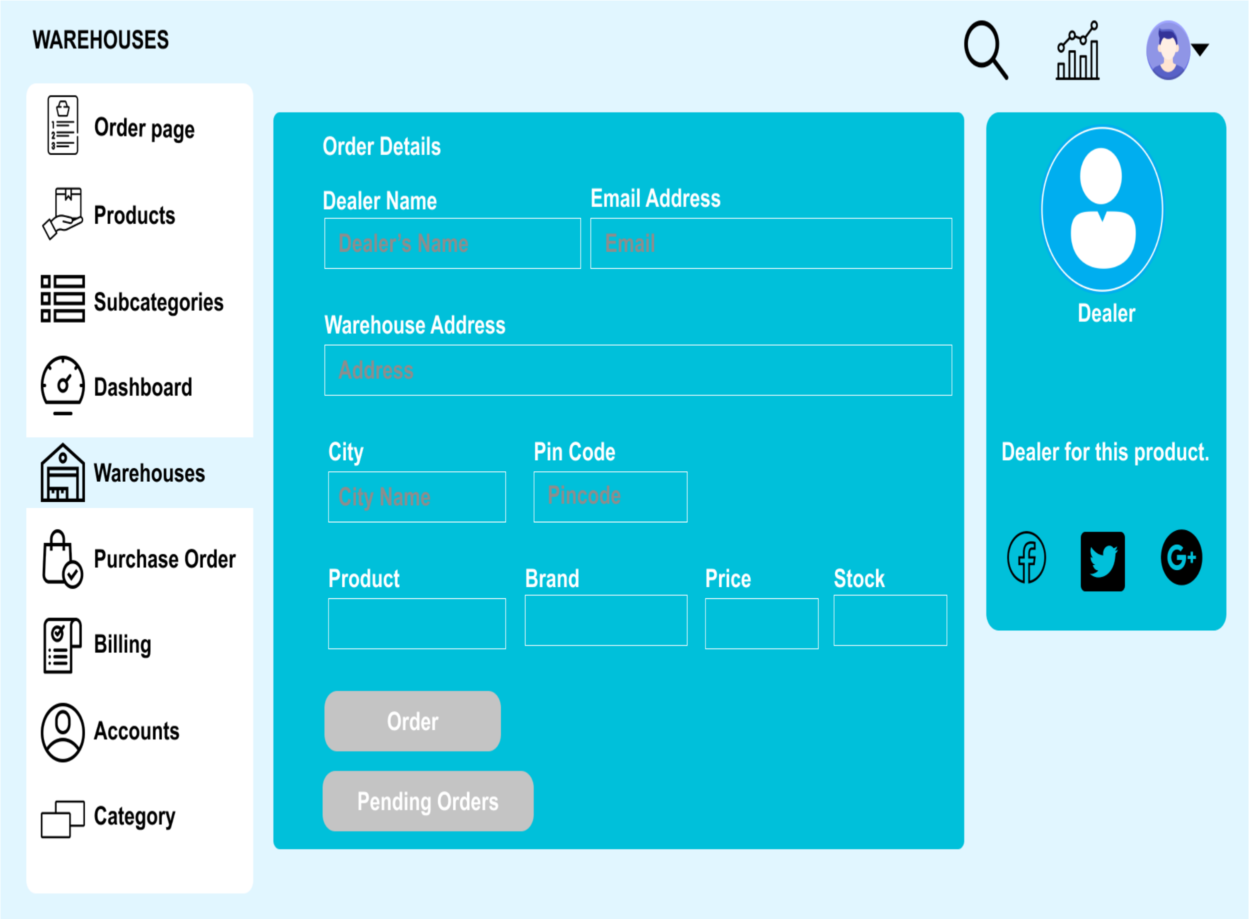
Pending Order

PRODUCT NAME	BRAND	PRICE	STOCK ORDERED	ORDER STATUS	
Product name 1	Brand 1	Price 1	Stock 1	delivered/pending	order recieved
Product name 2	Brand 2	Price 2	Stock 2	delivered/pending	order recieved
Product name 3	Brand 3	Price 3	Stock 3	delivered/pending	order recieved
Product name 4	Brand 4	Price 4	Stock 4	delivered/pending	order recieved
Product name 5	Brand 5	Price 5	Stock 5	delivered/pending	order recieved

When we(as a warehouse) need any items and when we order them, Here from the warehouse we can see the order status and order quantity.Once we receive it,the quantity of stock will increase and update quantity.

Order item : To order items we have to give details like product name,stock,brand,price and status.

(7)To give the order of a particular product.



Order Details

Dealer Name: Dealer's Name

Email Address: Email

Warehouse Address: Address

City: City Name

Pin Code: Pincode

Product

Brand

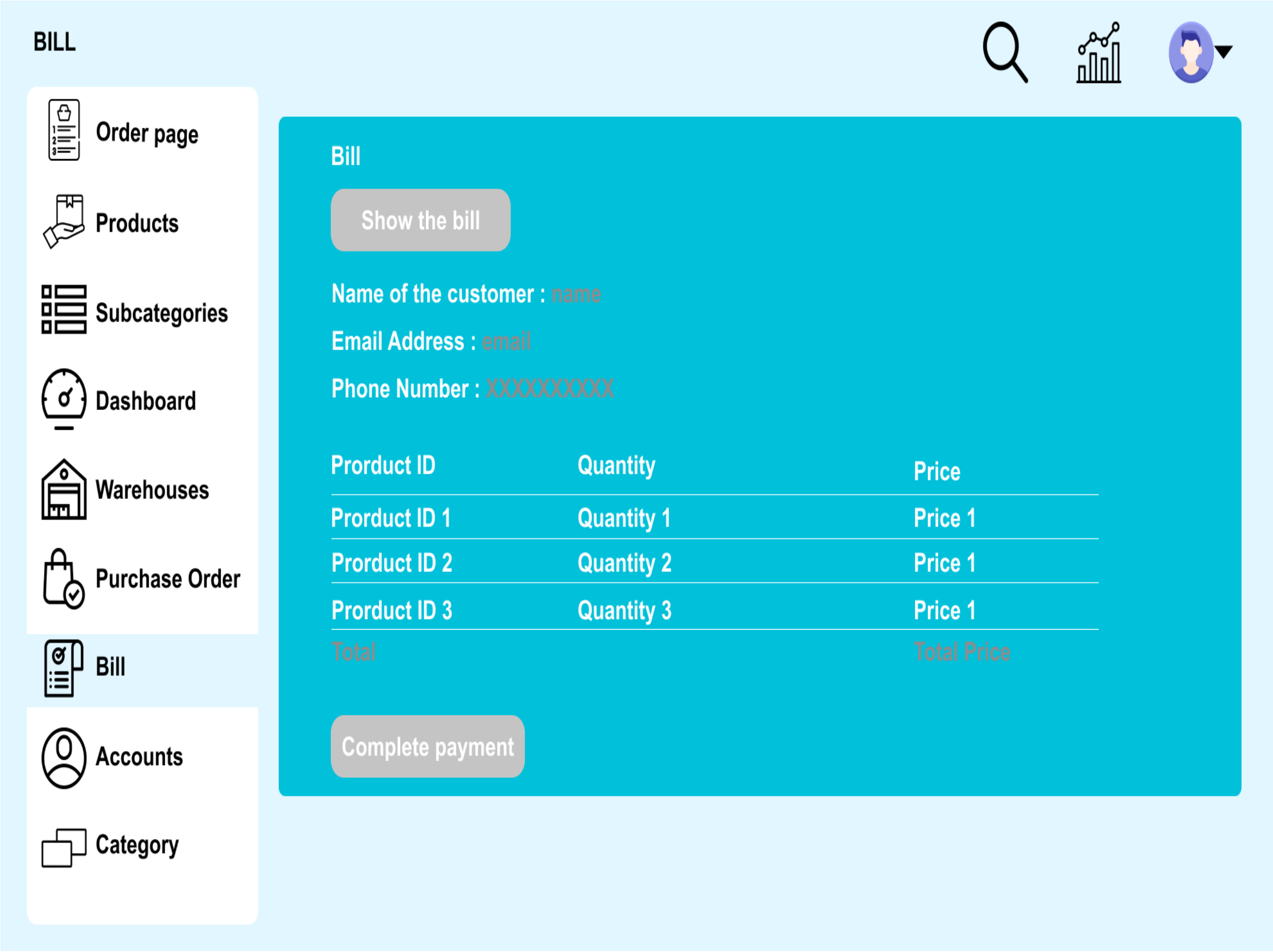
Price

Stock

As we mentioned above we ordered items from the dealer. For ordering new stock we need to add its details.

Add Order : To order items from the dealer we need to give information like dealer name,email id,warehouse details,city,price and stock.

(8)Bill Screen for product purchased.



Bill

Name of the customer : name

Email Address : email

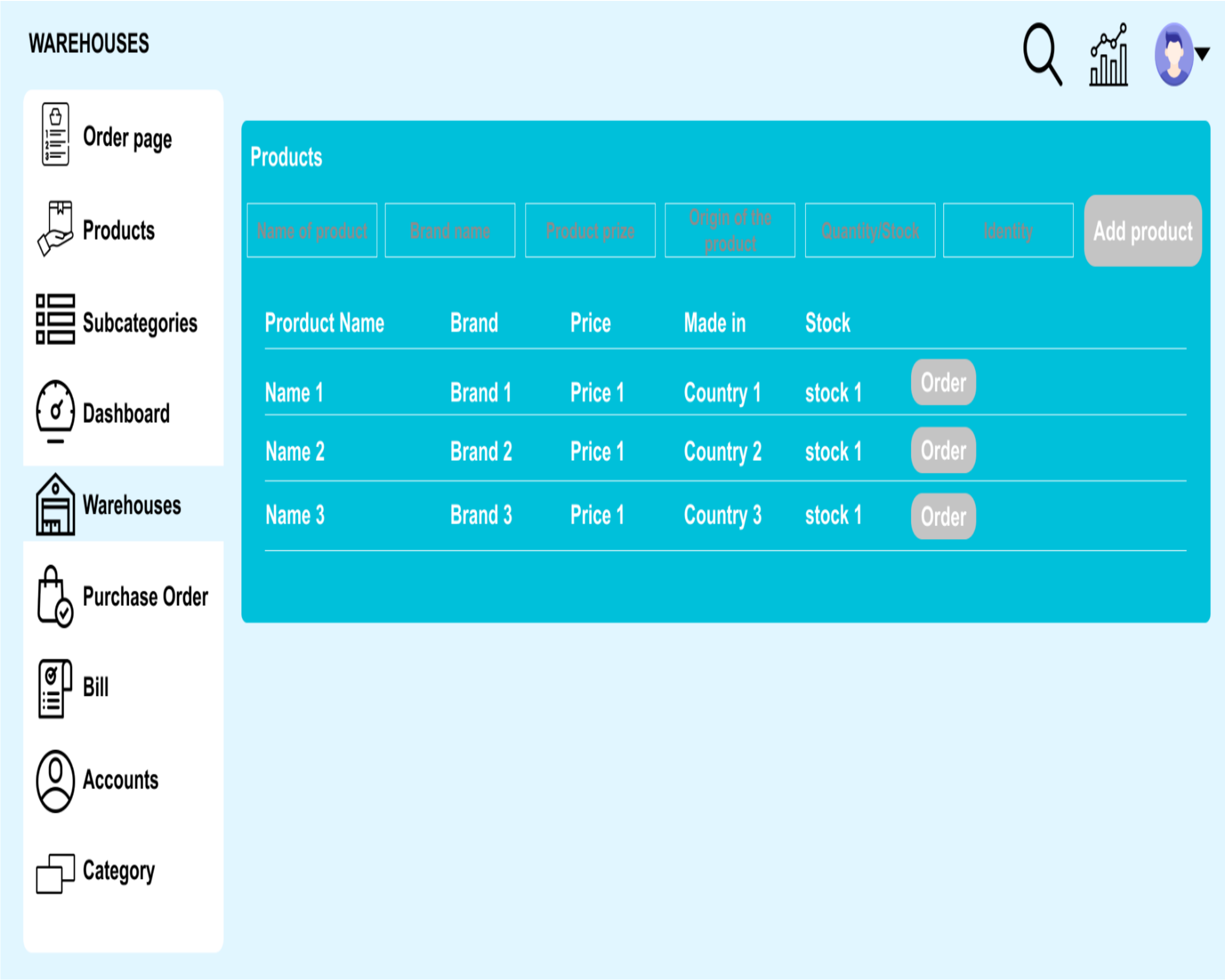
Phone Number : XXXXXXXXXX

Prorduct ID	Quantity	Price
Prorduct ID 1	Quantity 1	Price 1
Prorduct ID 2	Quantity 2	Price 1
Prorduct ID 3	Quantity 3	Price 1
Total		Total Price

When an employee makes the order then the bill will be automatically generated by the system.

Bill : Bill has information entered previously on the Generate order page and total amount which will be payable by the user.

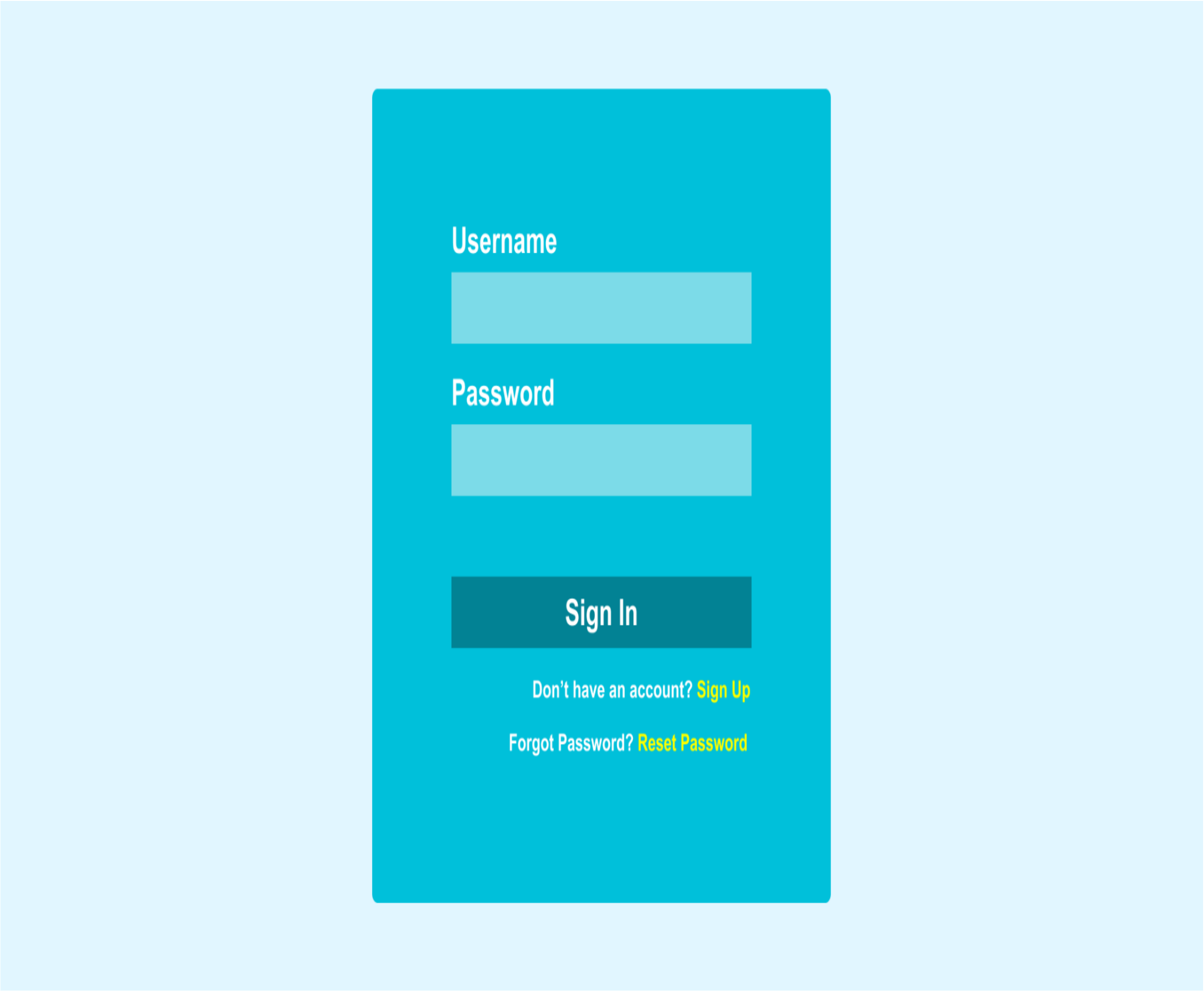
(9) Product list to order from vendor.



Prorduct Name	Brand	Price	Made in	Stock	
Name 1	Brand 1	Price 1	Country 1	stock 1	Order
Name 2	Brand 2	Price 1	Country 2	stock 1	Order
Name 3	Brand 3	Price 1	Country 3	stock 1	Order

Add product : add products based on different parameters namely, the product's name, brand name, price of the product, origin, quantity required and it's identity.
Order: place the order according to your required need of the above mentioned parameters.

(10)Login Screen


Username
Password
Sign In
Don't have an account? Sign Up
Forgot Password? Reset Password

Sign In : sign in using your designated email address and password.
Sign up : new customers can create their own new account.
Reset password : in case you’ve forgotten your password, you can reset your password using this function.

4.9 Construction details - which files do what

Admin.js	Rendering all the components and login authentication
Login.js	Login code
Categories.js	Show,add and remove the categories Ex.electronics, home care,sports
Subcategories.js	Show,add and remove the sub categories and section id (of warehouse) Ex.electronics have subcategories like phone,TV,etc
Products.js	Show and add products inside the sub categories
Order.js	Form to order the selected product from Vendor , confirm order and send email to the vendor
OrderHistory.js	Showing all the pending and delivered orders
OrderConfirm.js	Confirming the pending order and updating the database
AddCustomer.js	Add Customer details
AddItems.js	Add all the products while purchase
Bill.js	Display the Bill and consider payment here
Invoice.js	Mail the Billing details to the customer
UserProfile.js	Show and Update user details
Employee.js	Adding the employees

5 Testing

5.1 Introduction:

- What is Testing for project and its advantages
- Software testing is a process that evaluates the Software’s capabilities and compatibility which is already developed and to identify the issue of software to ensure that the product will be defect free. It also involves execution of software components using manual/automated tools to evaluate one or more properties of software.The motive of software testing is to discover errors, gaps or lacking necessities in assessment of real necessities.

5.2 Features to be tested:

Function	Description
ShowCategory(Categories.js)	To Display All the Categories.
AddCategory	To Add a New Category.
DeleteCategory	To Delete a Category
ShowSubCategory(SubCategories.js)	To Display Subcategory located at which subsection.
AddSubCategory	To Add a SubCategory.
DeleteSubCategory	To Delete a SubCategory.
AddProduct(Products.js)	To Add a New Product.
ViewProducts	To Display list of Products.
Order(Order.js)	Generate the Order for the required product.
SendEmail()	To Give order with a purchase list to the vendor.
AddPendingOrder	To Add the order which is pending to receive from the vendor.
PendingOrders (OrderHistory.js)	To list down pending orders from vendors.
OrderConfirm(OrderConfirm.js)	To change the Pending status to received performed by product collection operator.

UpdateOrders	Change in database when order is confirmed
clearInputs()	To Clear the input fields.
clearErrors()	To clear or ignore the error.
handleLogin()	To Login to the system with email id and password.
handleSignup()	To Add the New user to the system.
handleLogout()	To Logout from the system from your current email id.
handleForgetPassword()	To enable changing the password through email.
authListener()	Adds an observer for changes to the user's sign-in state.
AddCustomer	Adding Customer
AddProduct	Add product to the existing buying list of the customer
OnPurchase	This function updates the stock based upon the products bought by the customer
displaybill	Displaying the bill and calculating the total bill based upon the added list of products.
storeorder	Store the order to the database

5.3 Testing types and its advantages:

- Software testing is conducted is software development cycle to check whether developed software is meet the expected requirements
- Software testing is necessary to ensure that software is bug free ,sometimes bugs could be expensive.
- The important strategies in software engineering are: unit testing, integration testing, validation testing, and system testing.

- Unit testing : This software testing approach is followed by the programmer to check the unit of the program. It helps developers to know whether the individual unit of the code is operating properly or not.

- Integration testing :It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

- System testing :Software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

5.4 Approach:

- For overall testing the maximum testing is unit testing and specifically boundary testing. We attempt to discern various input values,various situations in which users will interact with the system.And to check whether the obtained output matches the expected output or not. It will help us to recognize the numerous errors that could arise within a system with the range of inputs.

- For each input field we've got attempted and taken consideration of diverse excessive inputs that the user can accidentally or intentionally feed into system.And the end result of those assessment will assist us to understand on which user input the system dalters and as a result precaution may be taken to prevent the any changes being submitted to the database for wrong inputs.

- The testing of the entire system is not possible at this time .So we selected to combine the components that are developed and test their functionality under the integration. We teste if the user is able to access the functions that are available and the user is able to go through the interface without trouble.

5.5 Testing Table:

- **Unit test:**

Test case Interface: login
1.1 To check that user give valid email address
1.2 To check whether the password is valid or not
1.3 To check the login of correct email address and password
1.4 To check Reset password functionality

Test ID	Test Item	Input	Output	Test status
1.1	Email text field	Not authorized email	Error:There is no user record corresponding to this identifier. The user may have been deleted	Pass
1.1	Email text field	blank	Error:The email address is badly formatted	Pass
1.1	Email text field	@ sign or dot sign is not present	Error:The email address is badly formatted	Pass
1.2	Password field	blank	Error:The password is invalid or the user does not have a password	Pass
1.2	Password field	wrong password	Error:The password is invalid or the user does not have a password	Pass
1.3	Email text field ,Password field and login button	valid email and password	Next page is rendered	Pass
1.4	Email text field	Not authorized email	Error:There is no user record corresponding to this identifier. The user may have been deleted	Pass
1.4	Email text field	blank	Error:The email address is badly formatted	Pass
1.4	Email text field	@ sign or dot sign is not present	Error:The email address is badly formatted	Pass
1.4	Email text field	Correct email	Mail is sent at the mail ID notification and mail is received by the user	Pass

Test Case Interface: SignUp

2.1 To check that user give valid email address

2.2 To check whether the password is valid or not

2.3 To check the sign up of correct email address and password

Test ID	Test Item	Input	Output	Test status
2.1	Email text field	blank	Error:The email address is badly formatted	Pass
2.1	Email text field	Not authorized email(email is not exist)	New user added in the database users list	Fail
2.1	Email text field	@ sign or dot sign is not present	Error:The email address is badly formatted	Pass
2.2	Password field	blank	The password must be 6 characters long or more	Pass
2.2	Password field	less than 6 letters	Password should be at least 6 characters	Pass
2.2	Password field	weak password	New user added in the database users list	Fail
2.2	Password field	6 letters and more	New user added in the database users list	Pass
2.3	Email text field andPassword field and sign up button	valid email and password	New user added in the database users list	Pass

Test Case Interface: Categories.js

3.1 Verify Add new category to the system database

3.2 To check view all the categories

3.3 To check the delete operation

3.4 To check the position of newly added category to the interface

3.5 Clear the textfield after adding the particular category

3.6 To verify that all the links of category grids are working correctly

3.7 Data Retrieval When Reload the screen

Test ID	Test Item	Input	Output	Test status
3.1	text field and add category button	valid category name	Added successfully	Pass
3.1	UI of the category page	-	Successfully added to the screen	Pass

3.2	UI of the category page	-	Successfully view all the categories from databse to the screen	Pass
3.3	Delete Button	delete particular category	Successfully delete the Category and product in it	Pass
3.3	UI of the category page	-	Successfully view all the categories from databse to the screen	Pass
3.4	UI of the category page	-	Get valid position on the interface	Pass
3.5	textfield for category name	-	Successfully clear the textfield	Pass
3.6	UI of the category page and category name button	click on the button	Successfully go to next page	Pass
3.7	UI of the category page	Reload	Successfully get the same data	Pass

Test Case Interface: SubCategories.js

4.1 Add Subcategory to the system

4.2 Check both input fields

4.3 Clear input fields after adding

4.4 Verify the UI of table view is not changed after adding

4.5 To check that the Subcategory and Subsections are Fetched correctly from database

4.6 Check the visit links provided for visiting the subcategory

4.7 Check the delete Icon is working or not

4.8 Data Retrieval When Reload the screen

Test ID	Test Item	Input	Output	Test status
4.1	text field and add subcategory button	valid subcategory name	Added successfully	Pass
4.2	Input Text field for subcategory name	name as string	Successfully write to the textfield	Pass
4.2	Input field for subsection Id	string		Pass
4.2	Input field for subsection Id	Integer value	Successfully write to the inputfield	Pass
4.3	UI of the subcategory page	-	Successfully clear the textfield	Pass
4.4	UI of the subcategory	-	Get valid position of table on the interface	Pass

	page			
4.5	content of the subcategory page	-	Successfully get data from the database	Pass
4.6	Visit button	click on the button	Successfully go to next page	Pass
4.7	Delete Button	click on the button	Successfully delete the subcategory	Pass
4.8	UI of the sub category page	Reload	Successfully get the same data	Pass

Interface:add products

5.1 To check that the products and their details are fetched from the database

5.2 Check all the input fields

5.3 Check if the add product button is working or not.

5.4 Clear input fields after adding

5.5 Verify the table view is not distorted

5.6 Check negative values in the stock input.

5.7 Check the order link

5.8 Data Retrieval When Reload the screen

Test ID	Test Item	Input	Output	Test status
5.1	UI of product page	-	Successfully get data from the database	Pass
5.2	Input Text fields	name as string	Successfully write to the textfield	Pass
5.2	Input field for Price and quantity	string	Not to accept characters in quantity field	Pass
5.2	Input field for Price and quantity	Integer value	Successfully write to the inputfield	Pass
5.3	Add product button	click on the button	Successfully add product to the database	Pass
5.4	UI of the product page	-	Successfully clear the textfield	Pass
5.5	UI of the product page	-	Get valid position of table on the interface	Pass
5.6	Input field for Price and quantity	negative value	Not to accept negative value	Pass
5.7	Order Button	click on the button	Successfully go to next page	Pass
5.8	UI of the sub category page	Reload	Successfully get the same data	Pass

Interface : Customer Addition
6.1 To check the company name text field.
6.2 To check the input field of the phone number.
6.3 To check the input field of the email.
6.4 To check the Address Text Field.
6.5 To check the postal code field.
6.6 Customer Database updates or not.
6.7 Existing Customer Check.

Test ID	Test Item	Input	Output	Test status
6.1	Company Name Text Field	blank	New Customer Added	Pass
6.1	Company Name Text Field	company exists or not	New Customer Added	Fail
6.2	Phone Number field	invalid Phone Number	New Customer Added	Fail
6.2	Phone Number field	Text written	Error it should be number	Pass
6.2	Phone Number field	Less than 10 numbers	New Customer Added	Fail
6.3	Email Field	invalid email		
6.3	Email Field	No @ symbol	Error Invalid Email	Pass
6.4	Address Text Field	Latitude,Longitude	New Customer Added	Pass
6.4	Address Text Field	Simple Text	New Customer Added	Pass
6.5	Postal code	Text written		
6.5	Postal code	Not equal to 6 digits.	Should not be less than 6 digits	Pass
6.6	Add Customer updates Database	Any Value Blank		
6.6	Add Customer updates Database	All Entries	Updates the Database	Pass
6.7	Has Existing customer	Object from Customer Database	new customer registration done of old customer	fail

Interface: Bill Generation
7.1 To check the Product ID field.
7.2 To Check the Quantity Field.

7.3 Remove Product Functionality.
7.4 Update Quantity Functionality.
7.5 Make Payment Functionalities.

Test ID	Test Item	Input	Output	Test status
7.1	Product id Field	Text	Error : It should be Number	Pass
7.1	Product id Field	id doesn't exist	It will add the product	Fail
7.2	Quantity	zero or Negative	Error : It should be greater than zero	Pass
7.2	Quantity	Quantity greater then available Stock	It will order	fail
7.3	remove the Product	Product	Product Purchased	Pass
7.4	Update Quantity	Quantity value	Product Purchased	Passl
7.5	Make Payment	Credit Card,Debit Card Online Banking	Product Purchased	Pass
7.5	Make Payment	Without Cash	Product not purchased	pass

Interface : Generate Invoice
8.1 To Check Printed Price.
8.2 Customer Receives invoice email.
8.3 Invoice Details update in database or not.

Test ID	Test Item	Input	Output	Test status
8.1	Price field	Negative price	Not Possible , auto generated	Pass
8.2	invoice email	email to customer	yes email received at registered email	pass
8.3	invoice order details to Database	Customer order	only Price is saved not products	Fail

Interface : Change Pending Order Status.
9.1 get order data from database.
9.2 Filter Order on basis of Order Status.
9.3 To check products ordered from which vendor.
9.4 To check the order received button.
9.5 Button mistakenly pressed.

Test ID	Test Item	Input	Output	Test status
9.1	Pending Order Fetch from Database	inbuilt query	list of orders	Pass
9.2	To Filter order on basis of Order Status	Select order status	list of filtered order	fail
9.3	Product ordered from which vendor	-	product from which vendor visible	pass
9.4	Order Received button	button press	changes status to deleiverd	pass
9.5	Wrong order Button Pressed	wrong button press	can't reach to it's normal stage	pass

Interface : To Give Stock Order
10.1 Dealer Name text field.
10.2 To check the stock field.
10.3 To check Email Field.
10.4 To check the Address Text Field.
10.5 To Check Postal Code.
10.6 To check City Field.
10.7 To check Product Field.
10.8 To check Brand Field.
10.9 To check Price Field.

Test ID	Test Item	Input	Output	Test status
10.1	Dealer Name Text Field	blank	New Customer Added	Pass
10.1	Dealer Name Text Field	Dealer exists or not	New Customer Added	Fail
10.1	Dealer Name Text Field	Unique Dealer	also added as new dealer	Fail

10.2	Stock Field	Not required	Properly Displays	Pass
10.3	Email Field	invalid email		
10.3	Email Field	No @ symbol		
10.3	Email Field	Unique email allowed	added as new Dealer not as existent Dealer	
10.4	Address Text Field	Latitude,Longitude	New Dealer Added	Pass
10.4	Address Text Field	Simple Text	New Dealer Added	Pass
10.5	Postal code	Text written		
10.5	Postal code	Not equal to 6 digits.		
10.6	City Field	invalid city name and numbers in city field	New Dealer Added	
10.7	Product Field	Not required	Properly Displays	Pass
10.8	Brand Field	Not required	Properly Displays	Pass
10.9	Price Field	Not required	Properly Displays	Pass

Integration test:

Test Id	TestCase	Output	status
101	verify that after login to the system it will redirect to dashboard	Successfully get the required page	Pass
102	sign up button will redirect to sign up page	Successfully get the required page	Pass
103	for Billing ,After add the customer it will switch to generate order	Successfully get the required page but not unique	Fail
104	After completion of payment,the customer should receive mail	Successfully Received the mail	Pass
105	Customer Order Database update	only order Price has been added not order product list	Fail
106	After Category page,it should redirect to subcategory page	Successfully get the required page	Pass
107	By Clicking on Category Button in Navbar all Category loads from Database	Successfully get the required page	Pass
108	succeedingly clicking on Visit Button of particular Subsection it will display Product under that Subsection	Successfully Displays Products only in that subsection	Pass

109	By clicking on visit category it will redirect to product page	Successfully get the required page	Pass
110	verify that after clicking on buy product button it will redirect to purchase order screen	Successfully get the product list	Pass
111	To Order New Product Stock.	You need to add the product first without ordering.	Fail
112	Vendor details and Product Details for purchase order	Product Details navigates,vendor details need to add	Fail
113	Warehouse Order Purchase added to pending order list.	successfully added to Pending order list	Pass
114	Verify that mail should be sent to vendor by confirm the purchase order	Successfully sent the mail	Pass
115	On Clicking update status button it will update order status	Successfully update the status	Pass
116	On Clicking update status button it will update order stock	Error in stock updation	Fail