Final Document on

"Cash & Carry"

Prepared for:

Dr. Taslim Taher — **Assistant Professor**
Mridha Md. Nafis Fuad — **Lecturer**

Course Number & Name:

CSE 3224 - Information & System Design Lab

Prepared by:

Lab Section: A1

Group No: 1

BIDYARTHI PAUL	**ID: 20200104023**
DIPTA BISWAS	**ID: 20200104013**
MD. ABDULLAH WALID RIAZ	**ID: 20200104014**
FARIHA TASNIM CHOWDHURY	**ID: 20200104001**
ZAKIA SULTANA	**ID: 20200104005**

Date: 01.08.2023

Ahsanullah University Of Science & Technology

Department of Computer Science and Engineering

Table Of Contents

1. Project Overview

The Super Shop Management System is a software solution designed to streamline and automate the operations of a retail supermarket or hypermarket. This system aims to enhance efficiency, accuracy and customer experience by integrating various functionalities into a cohesive platform. Key features include inventory management, product catalog management, customer information management, Invoice module and user customer relationship management.

1.1 Project implementation tools/ framework

- Netbeans 8.2
- Java Swing
- JDBC Driver**:** sqljdbc 4.0
- MSsql
- SQL Server 2014 Management Studio

1.2 User Story

As a store employee, I want to utilize the Super Shop Management System to efficiently add new products to specific inventory categories, update product details, input customer information, update customer profiles, and generate invoices with appropriate discounts based on the customer's membership status. This will streamline the process, enhance customer satisfaction, and ensure accurate record-keeping.

As a store employee,
I want to enter new products into the Super Shop Management System, categorizing them according to inventory categories, So that they can be easily managed and organized for efficient stock management.

- The system should provide an interface to input new products, update, delete and assign them to relevant inventory categories, such as groceries, electronics, household items, etc.
- I should be able to specify product details including name, description, price, and quantity.

As a store employee,
I want to input customer information into the Super Shop Management System, So that we can provide personalized service and maintain an up-to-date customer database.

- I should be able to input customer details, including name, contact information, address, and any relevant preferences or notes, as well as update and delete them.
- The system should store the customer information securely for future transactions and loyalty program enrollment.
- Upon successful update, the changes should be reflected in the customer's profile for future interactions and transactions.

As a store employee,
I want to generate invoices for shopping transactions using the Super Shop Management System, Applying appropriate discounts based on the customer's membership status, So that customers receive accurate bills reflecting their membership benefits.

- The system should calculate the total price of purchased items, including any applicable vat.
- It should apply appropriate discounts based on the customer's membership status, as specified in their profile.
- The system should provide options to print the invoice.

1.3 Module breakdown

1. **User Module:**
    - *User login:* Allows users to log in to the application using a username and a password by accessing the database.

2. **Product Information Management:**
    - *Add Product:* Allows store employees to input products with details such as name, description, date and price to the system according to their categories.

    - *Update Product:* Allows store employees to update the inventories of any products with their details.

- *Product Details:* Store employees can access the details of every product that's been in the inventory. Updation of any product will automatically update the details shown in the product details window.

- *Delete Product:* Store employees can delete any product details according to the store requirements.

3. Customer Information Management:

- *Add Customer:* Enables the input of product details such as name, contact number, email, gender and availability of membership.

- *Update Customer:* Allows store employees to update the Customer details.

- *Customer Details:* Store employees can access the details of every customer who has an entry in the database. Updation of any customer will automatically update the details shown in the customer details window.

- *Delete Customer:* Store employees can delete any customer details according to the availability.

4. Invoice generation management:

- Store employees can search any product or customer details using only one unique key for the Invoice.
- Calculates the total price of purchased items, including applicable vats based on the customer's selection.
- Applies appropriate discounts based on the customer's membership status, as specified in their profile.
- Generates itemized invoices that include the date, time, list of products, prices and discounts.
- Provides options to print invoices for customers.

5. **Daily Sales Management:**
    - Shows all the details of the products being sold with the purchased time of the customers.

6. **Log out & System Termination module**:
    - *Log Out:* Allows the user to log out of the application and shows the login interface.
    - *Close app:* Store employees can terminate or close the whole application.

1.4 Collaboration matrix

Tasks	Bidyarthi Paul	Fariha Tasnim Chowdhury	Zakia Sultana	Md Abdulla Walid Riaz	Dipta Biswas
Project Planning and Feature Selection	C	R	R	A	A
Create Roadmap	A	R	C	I	R
Project Proposal	C	R	I	A	C
Login Page Design & Implementation	A	R	C	C	A
Homepage UI Design	A	C	R	A	A
Database Connection & Design	C	A	A	A	C
Backend Implementation of Product Interface	R	A	A	A	C
Backend Implementation of Customer Interface	C	R	C	A	R
Backend Implementation of Invoice	C	R	A	A	R
Backend Implementation of Logout and Close App	R	A	R	C	R
Debugging	C	C	A	R	C
Cross Device Testing	I	I	R	A	C
Report Writing & Presentation	C	R	A	R	I
Final Launch	R	R	A	A	C

Fig-(1.1): Collaboration matrix

***R** = Responsible for the Task*
***A** = Accountable for the overall success or failure of the Task*
***C** = Consulted for the expertise*
***I** = Informed about the progress, decisions*

2. Process Model

Agile Scrum Methodology

As for our Supershop management system, the most suitable process model would be the Agile scrum methodology. The Agile Scrum model is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and delivering value to customers. In the Scrum framework, the project is divided into a series of fixed-length iterations called sprints. Each sprint consists of two- to four weeks of length where the goal of each sprint is to build the most important features first and come out with a potentially deliverable product. More features are built into the product in subsequent sprints and are adjusted based on customer feedback between sprints.

Reasons why the Agile Scrum methodology model is suitable for our project:

1. **Flexibility and Adaptability:**
   As our project progresses, we might discover new requirements or receive feedback from users. Agile Scrum allows the incorporation and adaptation of these changes and reprioritized features in subsequent sprints.

2. **Incremental Delivery:**
   With the ability to add, remove, and update products in inventories, customer details, and billing processes, incremental delivery allows the release of core functionalities early and incrementally enhances the system over time, ensuring quicker value delivery to users.

3. **Collaboration:**
   The scrum framework encourages collaboration throughout the project. Through Sprint Reviews, stakeholders can provide feedback on the implemented functionalities, ensuring the system meets their expectations.

4. **Iterative Development and Feedback:**
   Scrum follows an iterative approach, allowing for continuous improvement and feedback loops. This iterative feedback loop helps in refining the features related to inventory management, customer details, and billing processes, resulting in a more user-friendly and efficient system.

5. **Risk Management:**
   The Agile Scrum model supports effective risk management by promoting early detection. Regular Sprint Review provides opportunities to identify and address risk promptly, ensuring that any issues are resolved on time.

Due to its flexibility, incremental delivery, collaboration, iterative development, and risk management, the Agile Scrum model is best suitable for our project. This model allows for adapting to changing requirements, delivering value incrementally, involving users throughout the development process, incorporating iterative feedback, and managing risks effectively in our project.

3. Prototype

As a store employee,
I want to be able to log in to the SuperShop Management System so that I can access the necessary features and perform my duties effectively. I should be successfully logged into the system using the username name and password.

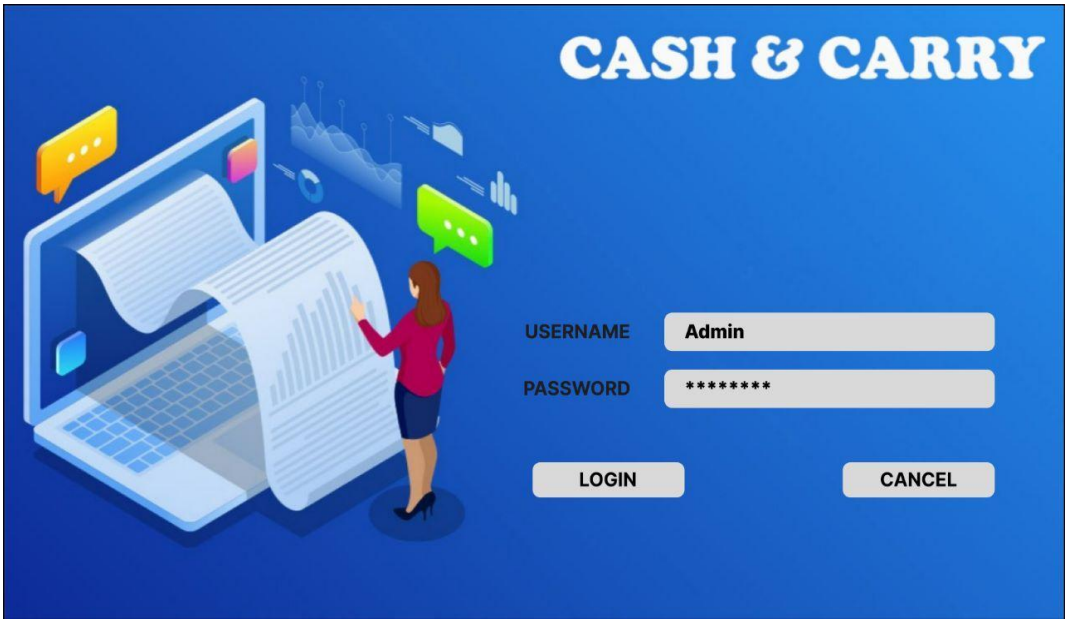


Fig-(3.1): Login UI

Fig-(3.2): Home Page UI

➢ Here, by implementing the ***Fig-(3.1): Login UI*** we can accomplish this user story. Action of the login button will open the ***Fig-(3.2) Home Page UI***, if the username and password are correct.

As a store employee,
I want to enter new products into the Super Shop Management System, categorizing them according to inventory categories, So that they can be easily managed and organized for efficient stock management.

- The system should provide an interface to input new products, update, delete, and assign them to relevant inventory categories, such as groceries, electronics, household items, etc.
- I should be able to specify product details including name, description, price, and quantity.

Fig-(3.3): Home Page UI

Fig-(3.4): New Product UI

➢ To add new products into the inventories, there will be an interaction between the ***Fig-(3.3): Home Page UI*** and ***Fig-(3.4): New Product UI.*** When the New product button is pressed from the ***Fig-(3.3): Home Page UI***, the ***Fig-(3.4): New Product UI*** will pop up, for the insertion of the details of new products and by clicking the save button from ***Fig-(3.4): New Product UI***, all the information will be saved in database.

Fig-(3.5): Home Page UI

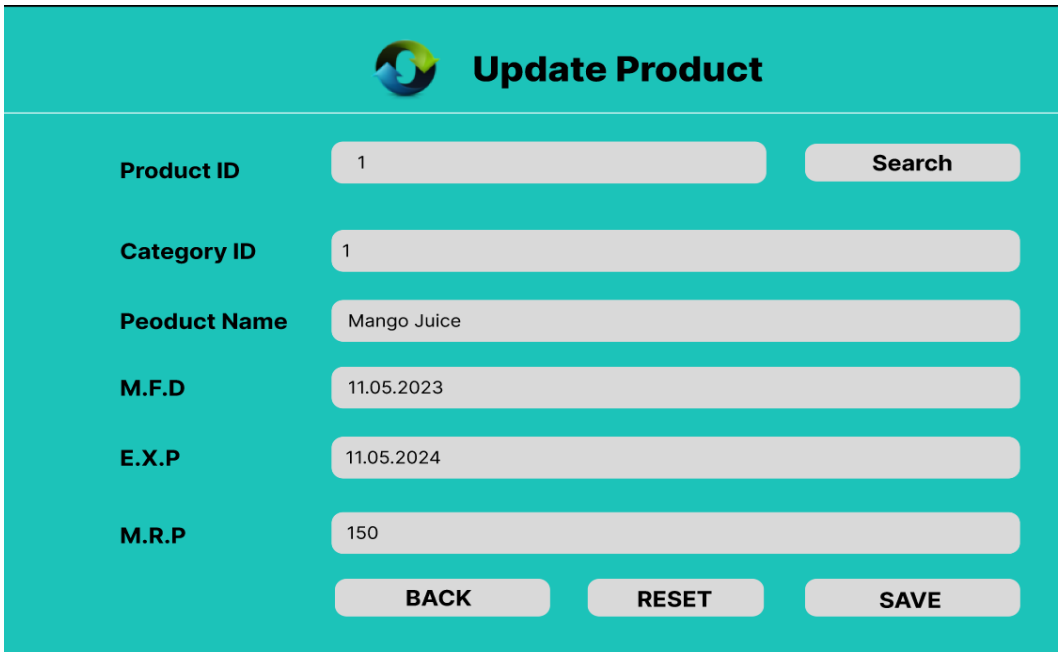


Fig-(3.6): Update Product UI

➢ To update products from the inventories, there needs to be an interaction between the ***Fig-(3.5): Home Page UI*** and ***Fig-(3.6): Update Product UI.*** When the Update product button is pressed from the ***Fig-(3.5): Home Page UI***, the ***Fig-(3.6): Update Product UI*** will pop up, for the insertion of the updation of the products and by clicking the save button from ***Fig-(3.6): Update Product UI***, all the information will be saved in the database and the reset button will clear all the fields.

Fig-(3.7): Home Page UI

Fig-(3.8): Product Details UI

➢ To see the details of the products from the inventories, there needs to be an interaction between the ***Fig-(3.7): Home Page UI*** and ***Fig-(3.8): Product Details UI.*** When the product details button is pressed from the ***Fig-(3.7): Home Page UI***, the ***Fig-(3.8): Product Details UI.*** will pop up and will give us all the detailed information of the products accessing from the database.

Fig-(3.9): Home Page UI

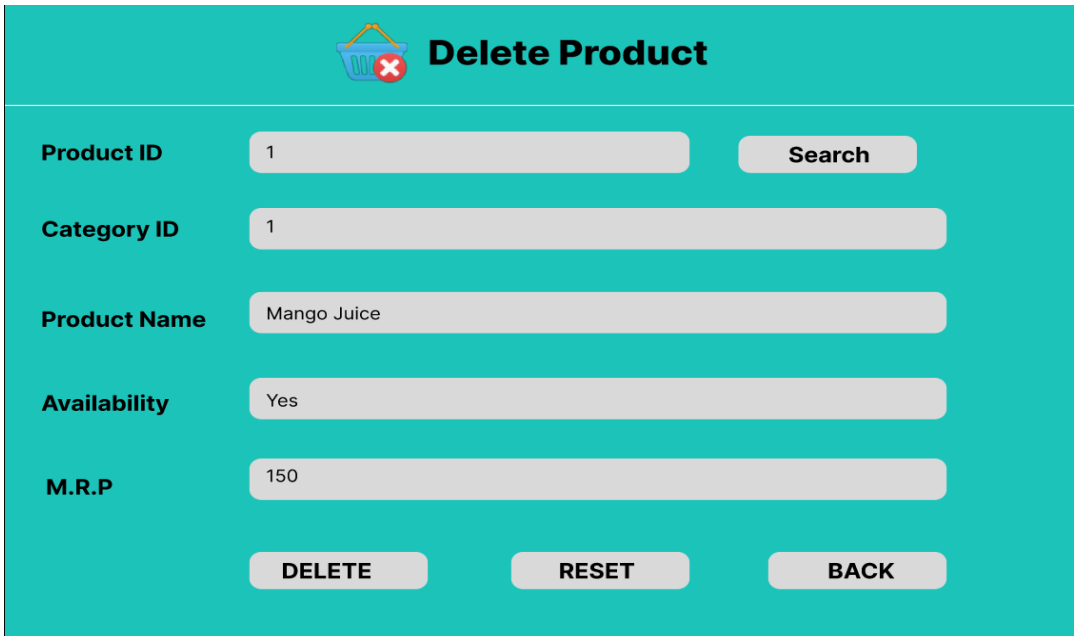


Fig-(3.10): Delete Product UI

➢ To delete products from the inventories, there needs to be an interaction between the ***Fig-(3.9): Home Page UI*** and ***Fig-(3.10): Delete Product UI.*** When the Delete product button is pressed from the ***Fig-(3.9): Home Page UI***, the ***Fig-(3.10): Delete Product UI*** will pop up, for the deletion of the products and by clicking the delete button from ***Fig-(3.10): Delete Product UI***, all the details of that particular product will be deleted from the database.

As a store employee,

I want to input customer information into the Super Shop Management System, So that We can provide personalized service and maintain an up-to-date customer database.

- I should be able to input customer details, including name, contact information, address, and any relevant preferences or notes, as well as update and delete them.
- The system should store the customer information securely for future transactions and loyalty program enrollment.
- Upon successful update, the changes should be reflected in the customer's profile for future interactions and transactions.

Fig-(3.11): Home Page UI

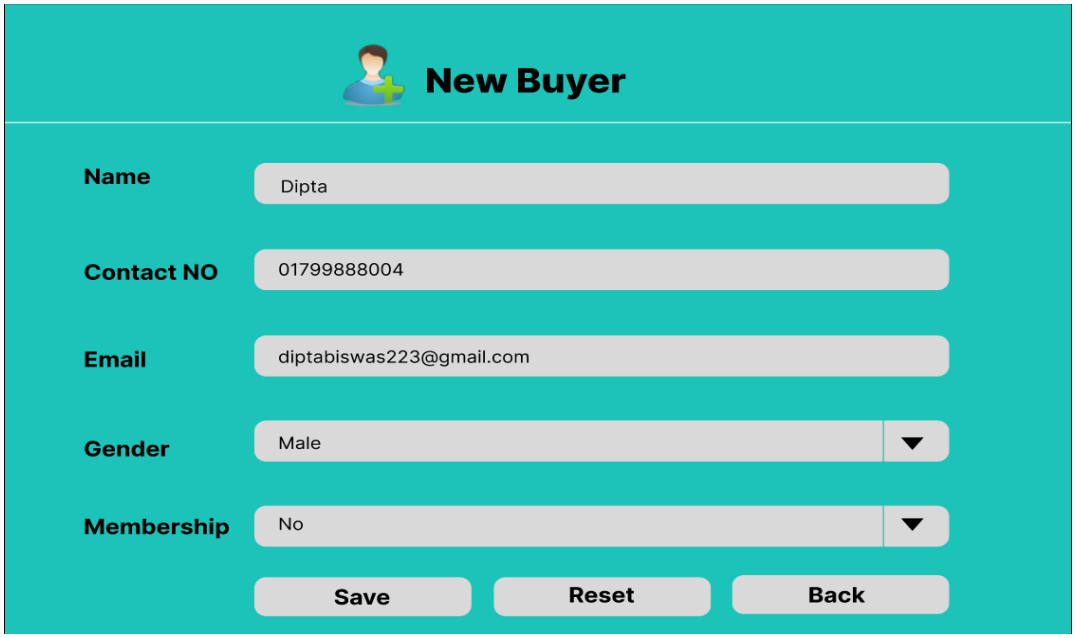


Fig-(3.12): New Buyer UI

➢ To add new buyers into the database, there will be an interaction between the ***Fig-(3.11): Home Page UI*** and ***Fig-(3.12): New Buyer UI.*** When the new buyer button is pressed from the ***Fig-(3.11): Home Page UI***, the ***Fig-(3.12): New Buyer UI*** will pop up, for the insertion of the details of a new buyer and by clicking the save button from ***Fig-(3.12): New Buyer UI***, all the information will be saved in database.

Fig-(3.13): Home Page UI

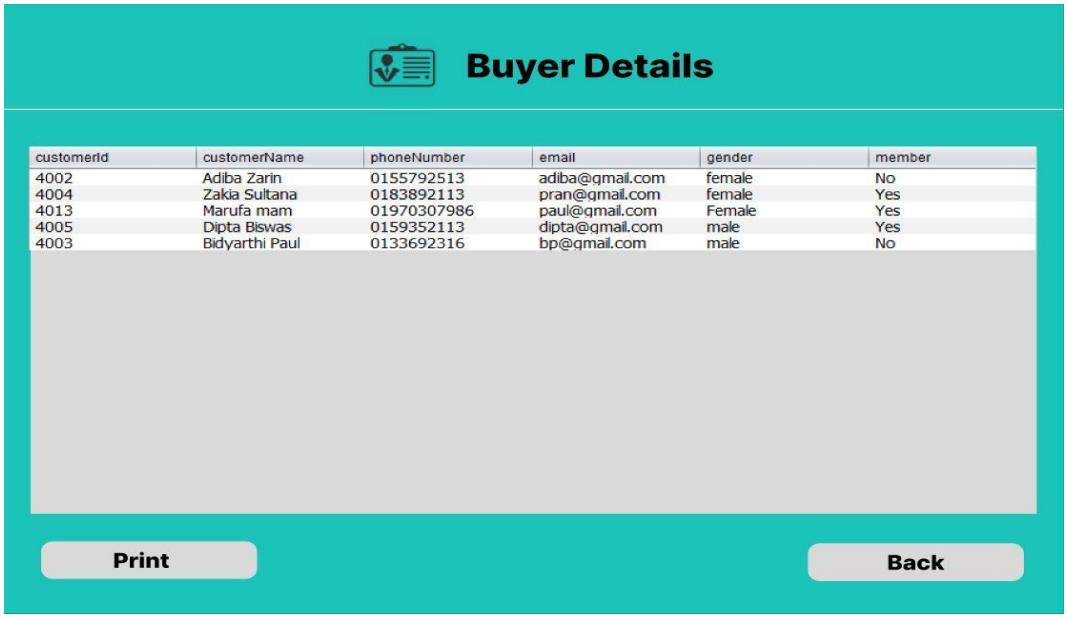


Fig-(3.14): Buyer Details UI

➢ To see the details of the buyers from the database, there needs to be an interaction between the ***Fig-(3.13): Home Page UI*** and ***Fig-(3.14): Buyer Details UI.*** When the buyer details button is pressed from the ***Fig-(3.13): Home Page UI***, the ***Fig-(3.14): Buyer Details UI.*** will pop up and will give us all the detailed information of the buyers accessing from the database.

Fig-(3.15): Home Page UI

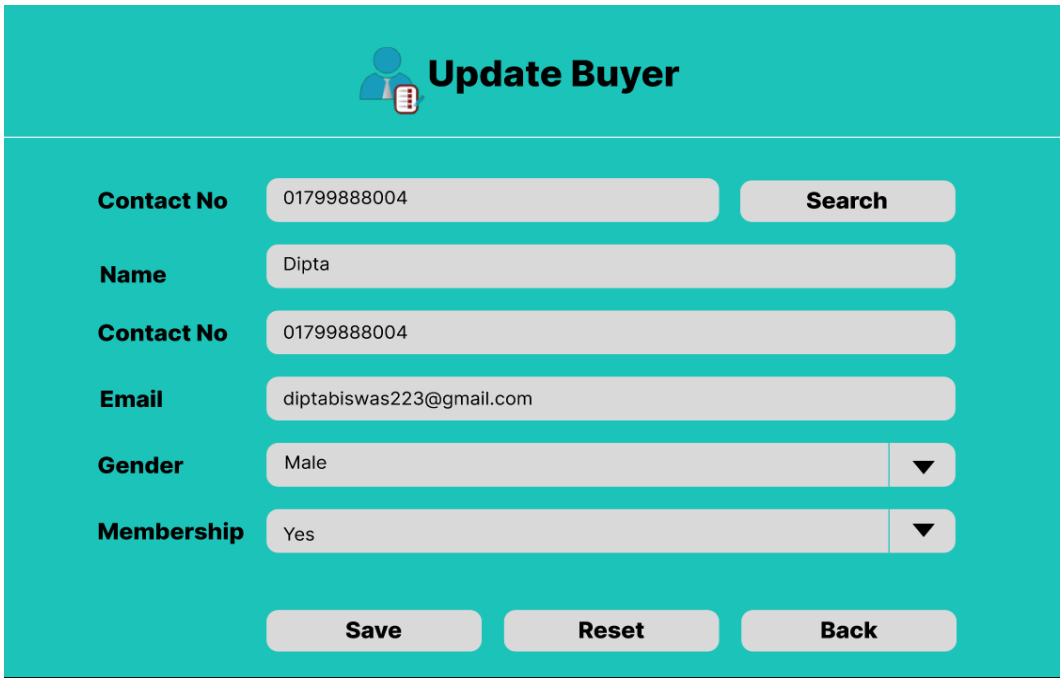


Fig-(3.16): Update Buyer UI

➢ To update buyer information from the database, there needs to be an interaction between the ***Fig-(3.15): Home Page UI*** and ***Fig-(3.16): Update Buyer UI.*** When the Update buyer button is pressed from the ***Fig-(3.15): Home Page UI***, the ***Fig-(3.16): Update Buyer UI*** will pop up, for the insertion of the updation of the products and by clicking the save button from ***Fig-(3.16): Update Buyer UI***, all the information will be saved in the database and the reset button will clear all the fields.

Fig-(3.17): Home Page UI

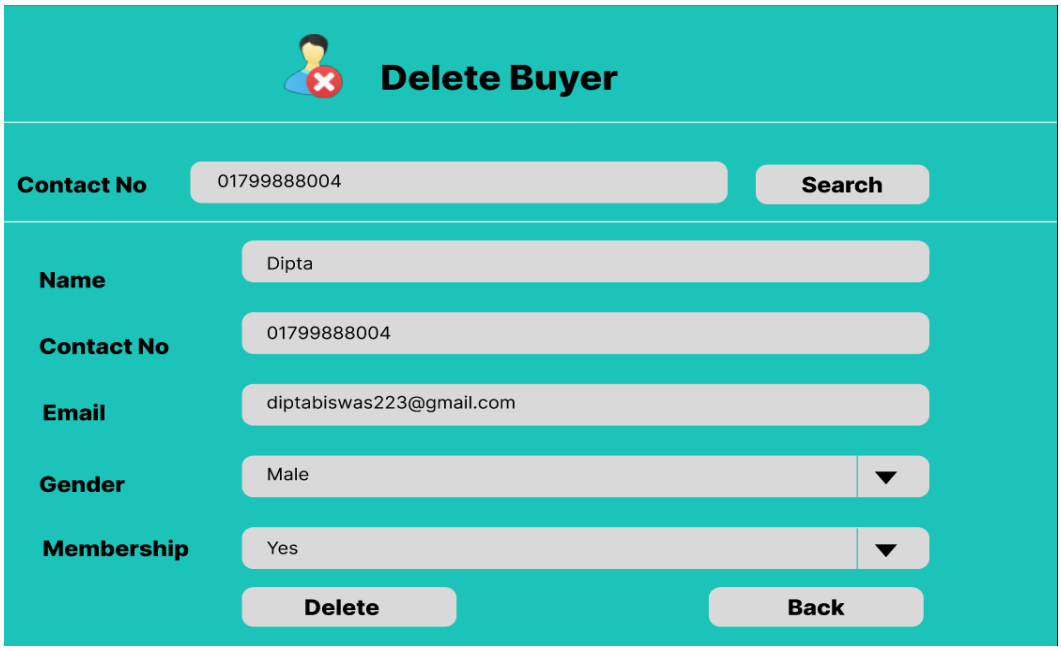


Fig-(3.18): Delete Buyer UI

➢ To delete any particular buyer from the database, there needs to be an interaction between the ***Fig-(3.17): Home Page UI*** and ***Fig-(3.18): Delete Buyer UI.*** When the Delete buyer button is pressed from the ***Fig-(3.17): Home Page UI***, the ***Fig-(3.18): Delete Buyer UI*** will pop up, for the deletion of the buyer's information and by clicking the delete button from***Fig-(3.18): Delete Buyer UI***, all the details of that particular customer will be deleted from the database.

As a store employee,

I want to generate invoices for shopping transactions using the Super Shop Management System, Applying appropriate discounts based on the customer's membership status, So that customers receive accurate bills reflecting their membership benefits. And also I should be able to keep track of the Daily Sales.

- The system should calculate the total price of purchased items, including any applicable vat.
- It should apply appropriate discounts based on the customer's membership status, as specified in their profile.
- The system should provide options to print the invoice.
- The system should provide an option to keep track of all the products that have been sold on that day.

Fig-(3.19): Home Page UI

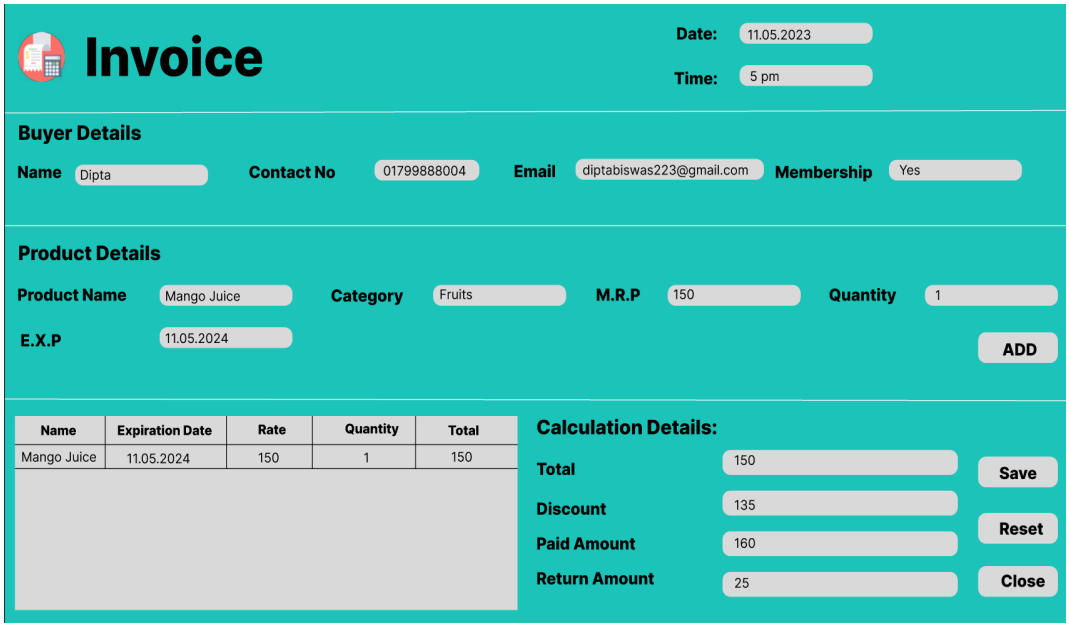


Fig-(3.20): Invoice UI (With Membership)

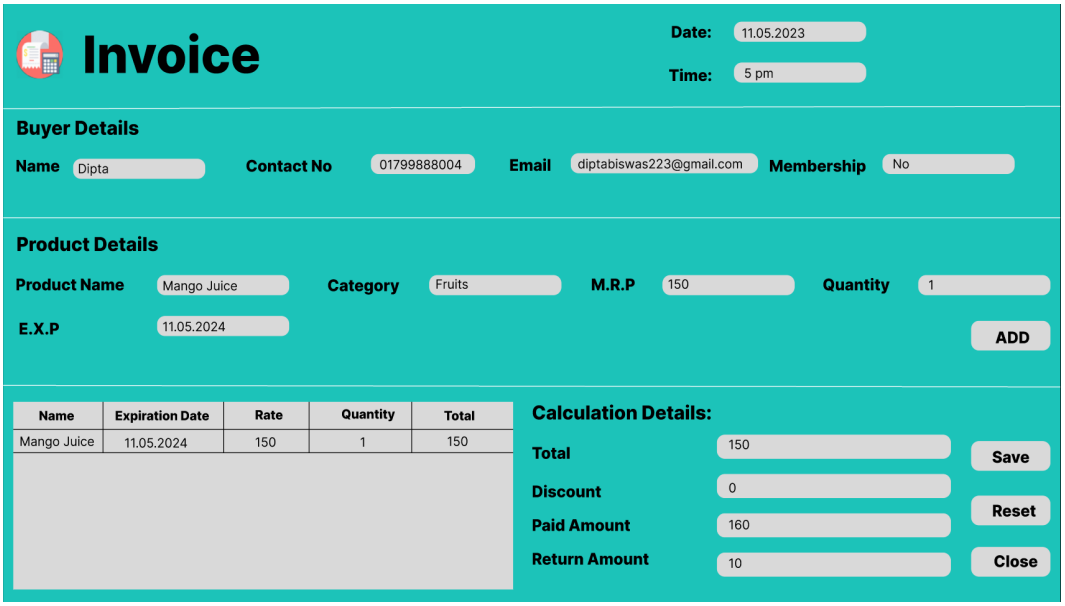
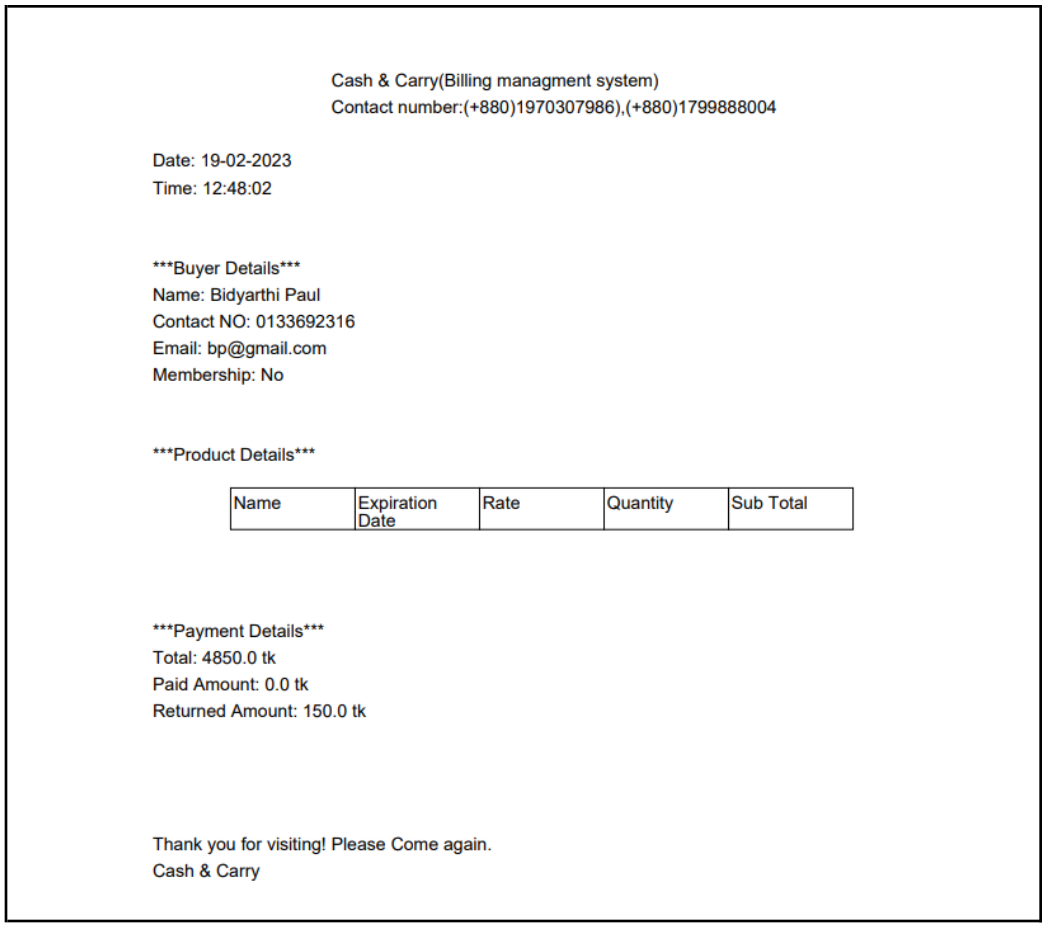


Fig-(3.21): Invoice UI (Without Membership)



Cash & Carry(Billing managment system)
Contact number:(+880)1970307986),(+880)1799888004

Date: 19-02-2023
Time: 12:48:02

Buyer Details
Name: Bidyarthi Paul
Contact NO: 0133692316
Email: bp@gmail.com
Membership: No

Product Details

Name	Expiration Date	Rate	Quantity	Sub Total

Payment Details
Total: 4850.0 tk
Paid Amount: 0.0 tk
Returned Amount: 150.0 tk

Thank you for visiting! Please Come again.
Cash & Carry

Fig-(3.22): Invoice Pdf

➢ To fulfill the above user story, there will be an interaction between ***Fig-(3.19): Home Page UI, Fig-(3.20): Invoice UI (With Membership)*** and ***Fig-(3.21): Invoice UI (Without Membership)*** **.** When the invoice button is clicked from ***Fig-(3.19): Home Page UI,*** an Invoice window will pop up. In this window, the user will be able to generate a bill by taking input of the quantity of the products that's been bought by the buyer. In the buyer details section, the user will be able to find the details of the customer using an unique key (Contact No). In the same way, the user will be able to find the details of the products using an unique key (Product name) and by taking the input of the quantity of the products user will be able to generate a bill of all the product that's been sold. In the lower table of the left hand side, details of all the sold products will be shown with their quantity and price. If the customer has a membership card, Invoice window will be like ***Fig-(3.20): Invoice UI (With Membership),*** where the buyer will get a discount of **5%** and the discount field will show the total price after the discount has been applied. If the buyer doesn't have any membership card, the Invoice window will be like ***Fig-(3.21): Invoice UI (Without Membership),*** where the buyer will not get any discount and the discount field will show zero value on it. When the save button is pressed from the invoice window we will be able to generate a bill/pdf like ***Fig-(3.22): Invoice PDF*** which includes all the information of products, customers and payment details.

Fig-(3.23): Home Page UI

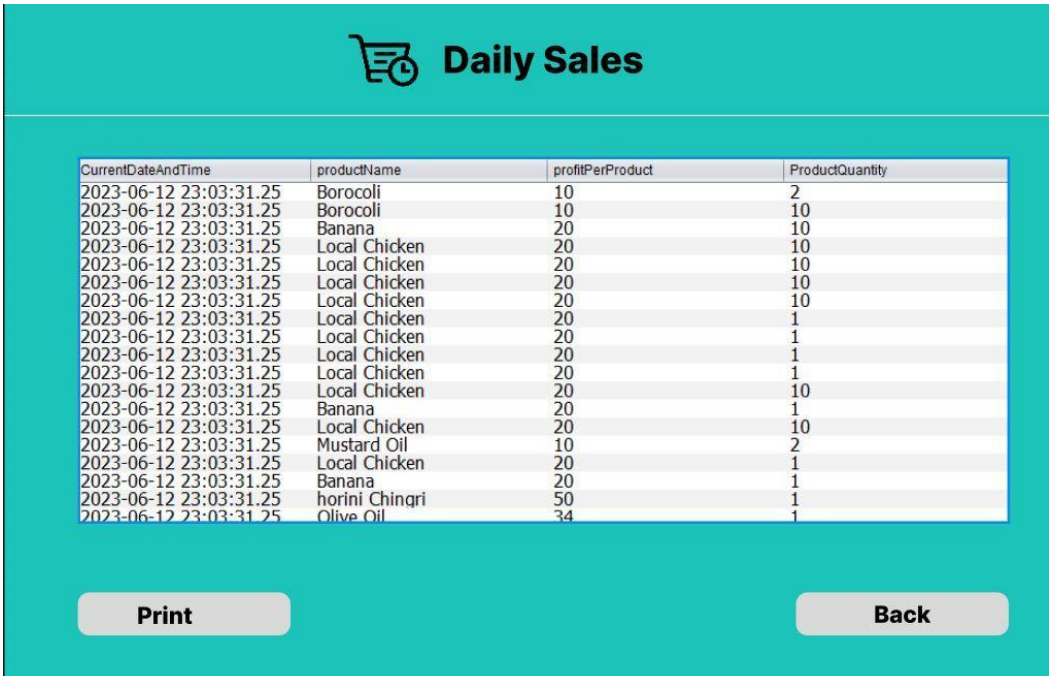


Fig-(3.24):Daily Sales UI

➢ To see the details of all the products that have been sold on a particular day, there will be an interaction between the ***Fig-(3.23): Home Page UI*** and ***Fig-(3.24): Daily Sales UI.*** When the Daily Sales button is clicked, the daily sales window will pop up and we will be able to see all the details of daily sales which will also include the selling date of the product.

Fig-(3.25): Home Page UI (Log Out)

➢ To log out of the system the close app button needs to be pressed from the ***Fig-(3.25): Home Page UI (Log Out),*** which will take the user back to the login page.

Fig-(3.26): Home Page UI (Close App)

➢ To terminate the system the close app button needs to be pressed from the ***Fig-(3.26): Home Page UI (Close App),*** which will close the whole system.

4. Function point Analysis

Table 1.1: Weights of 5-FP Attributes

Measurement Parameter	Low	Average	High
1. Number of external inputs (EI)	7	10	15
2. Number of external outputs (EO)	5	7	10
3. Number of external inquiries (EQ)	3	4	6
4. Number of internal files (ILF)	4	5	7
5. Number of external interfaces (EIF)	3	4	6

4.1 Calculate Unadjusted Function Point(UFP)

Estimate the Complexity Values for each Functional Point Components :

1. External Inputs (EI):

- **Admin login: (Low)** - It involves basic authentication, typically a simple username and password check.
- **Add new product details: (Low)** - A straightforward data entry operation with minimal validation and processing.
- **Update product details: (Average)** - The operation involves searching for a particular product from the database and updating the product information.
- **Show product details: (Low)** - The operation involves searching for a product by the product name and displaying its details.
- **Delete product details: (Average)** - The operation involves searching for a particular product from the database and removing the product from the system.
- **Add new customer details: (Low)** - A basic data entry task for capturing customer information.
- **Update customer details: (Average)** - The operation involves searching for a specific customer from the database and updating the customer information.
- **Show customer details: (Low)** - The operation involves searching for a customer by the customer ID and displaying the details.
- **Delete customer details: (Average)** - The operation involves searching for a specific customer from the database and removing the customer details from the system.

- **Search in daily sales: (Average)** - The operation involves searching of daily sales details according the requirements (Date, product, category)
- **Search customer details for billing calculation: (Average)** - This operation requires searching and processing customer information to calculate relevant billing details.
- **Search purchased product details for billing calculation: (Average)** - It requires searching and processing of purchased product details for billing purposes.
- **Add product quantity for billing calculation: (Low)** - A basic data entry task for storing quantity of the purchased products.
- I**nvoice generation: (Average)** - The generation of invoices involves combining product and customer details, applying discounts, calculating totals, and generating the final invoice, making it an operation of average complexity.

Total Weighing Factor for EI: 7 * 6 + 10 * 8 + 15 * 0 = 122

2. External Outputs (EO):

- **Add new product details: (Low)** - After adding new product details, the system will display a confirmation message or a success status indicating that the product has been successfully added.
- **Update product details: (Low)** - After updating product details, the system might display a confirmation message or a success status indicating that the product information has been updated.
- **Show product details: (Average)** - The system displays the product details retrieved from the database based on the search criteria.
- **Delete product details: (Low)** - After deleting product details, the system will display a confirmation message or a success status indicating that the product has been successfully removed.
- **Add new customer details: (Low)** - After adding new customer details, the system will display a confirmation message or a success status indicating that the customer information has been successfully added.
- **Update customer details: (Low)** - After updating customer details, the system will display a confirmation message or a success status indicating that the customer information has been updated.
- **Show customer details: (Average)** - The system displays the customer details retrieved from the database based on the search criteria.

- **Delete customer details: (Low)** - After deleting customer details, the system will display a confirmation message or a success status indicating that the customer has been successfully removed.
- **Search in daily sales: (Average)** - The system displays the daily sales details retrieved from the database based on the search criteria (e.g., date, product, category).
- **Search customer details for billing calculation: (Low)** - The system will display the relevant customer details required for billing calculation.
- **Search purchased product details for billing calculation: (Low)** - The system will display the relevant purchased product details required for billing calculation.
- **Invoice generation: (High)** - The system generates itemized invoices with the date, time, list of products, prices, and discounts. The invoice itself is the primary external output.

Total Weighing Factor for EO: 5 * 8 + 7 * 3 + 10 * 1 = 71

3. External Inquiries (EQ):

- **Show product details : (Low)** - The operation involves searching for a product by the product name or ID and displaying its details. It is a simple retrieval without any significant processing.
- **Show customer details: (Low)** - The operation involves searching for a customer by the customer ID or name and displaying their details. Similar to showing product details, it is a straightforward retrieval of data without complex processing.
- **Search in daily sales (Average)** - The operation involves searching for daily sales details based on specific criteria, such as date, product, or category. This requires moderate processing to filter and retrieve relevant data from the database.
- **Search customer details for billing calculation: (Average)** - The operation requires searching and processing customer information to calculate relevant billing details. This involves retrieving data from the database and performing calculations for billing purposes.
- **Search purchased product details for billing calculation: (Average)** - The operation involves searching and retrieving purchased product details required for billing calculation. It is a simple retrieval of data from the database with a little complex processing.

Total Weighing Factor for EQ: 3 * 2 + 4 * 3 + 6 * 0 = 18

4. Internal Logical Files (ILF):

- Admin: (Low)
- Product Information Management: (Low)
- Customer Information Management: (Low)
- Category Information Management: (Low)
- Daily Sales Management: (Average)

Total Weighing Factor for ILF: 4 * 4 + 5 * 1 + 7 * 0 = 21

5. External Interface Files (EIF):

- **Invoice PDF: (High) -** System allows the generated invoices to print them as PDF for customers upon request.

Total Weighing Factor for EIF: 3 * 0 + 4 * 0 + 6 * 1 = 6

Table 1.2: Computing weighing factor

Measurement Parameter	**Weighing Factor**
1. Number of external inputs (EI)	122
2. Number of external outputs (EO)	71
3. Number of external inquiries (EQ)	18
4. Number of internal files (ILF)	21
5. Number of external interfaces (EIF)	6
	Total Count : 238

So,**UFP** = 238

4.2 Calculate Complexity Adjustment Factor (CAF)

Table 2: 14-factors

General System Characteristics (GSC)	Degree Of Influence (DI) (0-5)
1.Data Communications	1
2.Distributed Data Processing	0
3.Performance	4
4.Heavily User Configuration	0
5.Transaction Rate	0
6.Online Data Entry	0
7.End-User Efficiency	5
8.Online Update	1
9.Complex Processing	3
10.Reusability	4
11.Installation Ease	5
12.Operational Ease	4
13.Multiple Sites	0
14.Facilitate Change	4
Total degree of influence	**TDI** : 31

Therefore, **CAF** = [0.65 + (0.01 * **TDI**)] = [0.65 + (0.01 * 31)] = 0.96

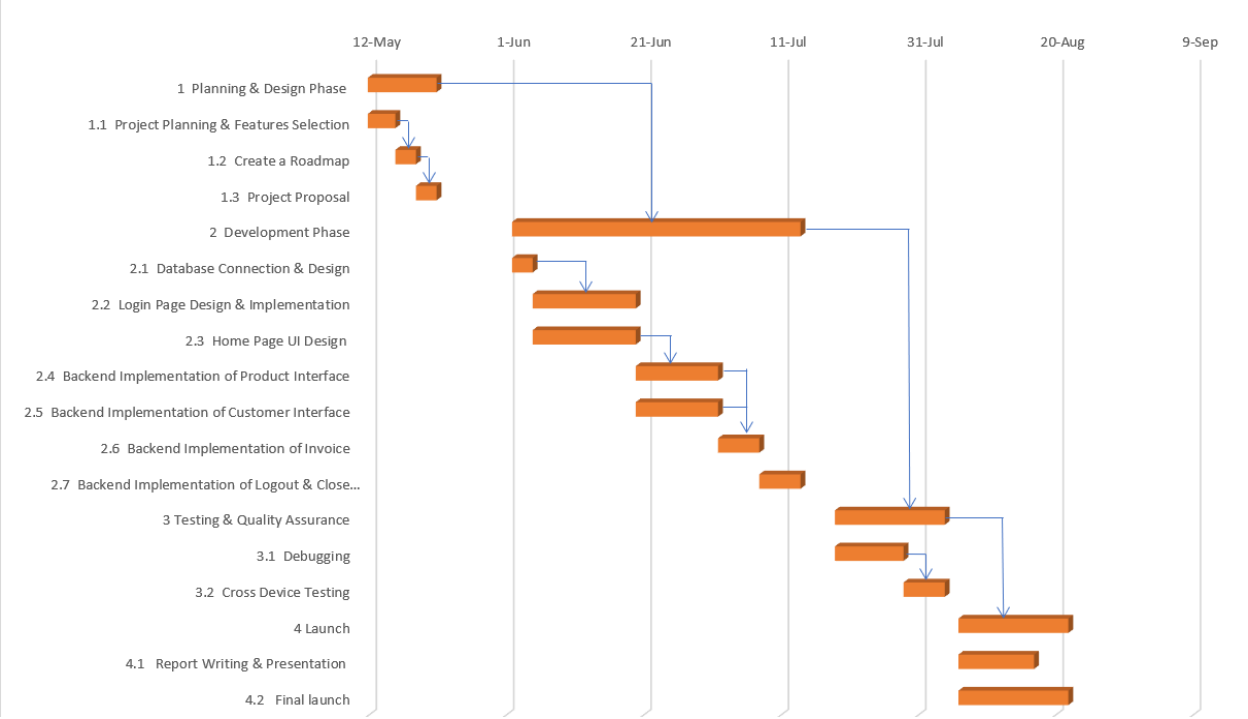
4.3 Calculate Functional Points(FP)

The required FP = **UFP** * **CAF** = 238 * 0.96 = 228.48

5. Gantt chart

The Gantt chart presented below provides a visual representation of the project schedule and a complete roadmap for the development of our Super shop management system.

Task	Start Date	Days to complete
1 Planning & Design Phase	12-May	10
1.1 Project Planning & Features Selection	12-May	4
1.2 Create a Roadmap	16-May	3
1.3 Project Proposal	19-May	3
2 Development Phase	2-Jun	42
2.1 Database Connection & Design	2-Jun	3
2.2 Login Page Design & Implementation	5-Jun	15
2.3 Home Page UI Design	5-Jun	15
2.4 Backend Implementation of Product Interface	20-Jun	12
2.5 Backend Implementation of Customer Interface	20-Jun	12
2.6 Backend Implementation of Invoice	2-Jul	6
2.7 Backend Implementation of Logout & Close App	8-Jul	6
3 Testing & Quality Assurance	19-Jul	16
3.1 Debugging	19-Jul	10
3.2 Cross Device Testing	29-Jul	6
4 Launch	6-Aug	16
4.1 Report Writing & Presentation	6-Aug	11
4.2 Final launch	6-Aug	16



6. Diagrams

6.1 Use case diagrams

LEVEL-0 USE CASE - Supershop Management System

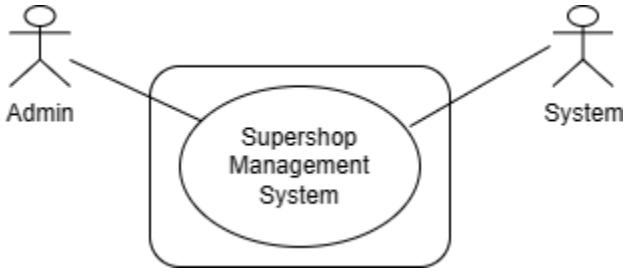


Fig-(6.1): Level-0 Use case-Supershop Management System

Description of Use case diagram level-0:
There are 2 actors who directly or indirectly interact with the system. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information. The actors are -
1. Administrator or Owner
2. Customer

LEVEL-1 USE CASE – Sub system

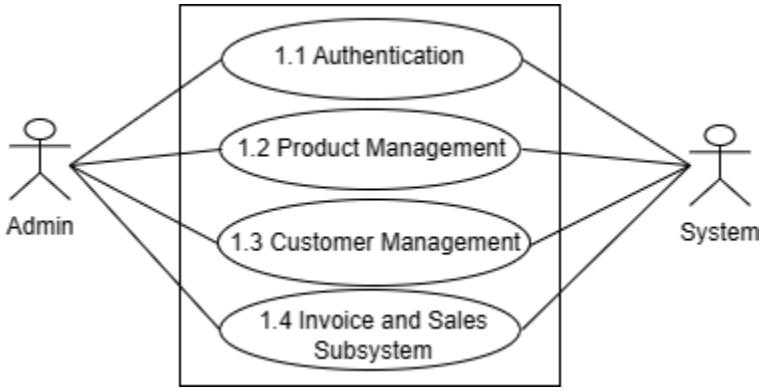


Fig-(6.2): Level-1 Use case-Sub system

Description of Use case diagram level-1:
There are four subsystems in the Super Shop Management System. They are :
1. Authentication
2. Product management
3. Customer management
4. Invoice and Sales Subsystem

The four subsystems are decomposed, in levels 1.1, 1.2, 1.3, and 1.4 respectively.

LEVEL-1.1 USE CASE - Authentication

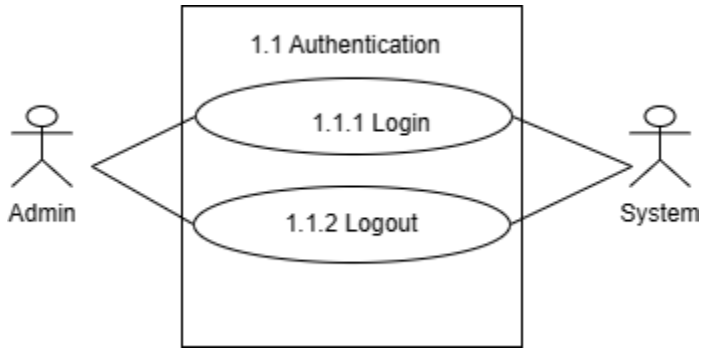


Fig-(6.3): Level-1.1 Use case-Authentication

Description of Use case diagram level-1.1:
When the admin wants to log in, he needs to enter the username and password. If the username and password match, then, the login is successful "Login successful" message is shown. The system shows an error message if the username or password, or both are wrong and the user can try again to log into the system. Admin can log out of the system and can log into the system again.

Action Reply:
User (Administrator):

A1: User provides username and password
R1: System checks validity. If the username and password is valid, the system will allow the user to log into the account.

A2: User provides invalid information`
R2: The system will show an error message and allow the user to try again.

A2: User selects the logout module`
R2: The system will be redirected to the login page and the admin will be logged out of the system.

LEVEL-1.2 USE CASE - Product management

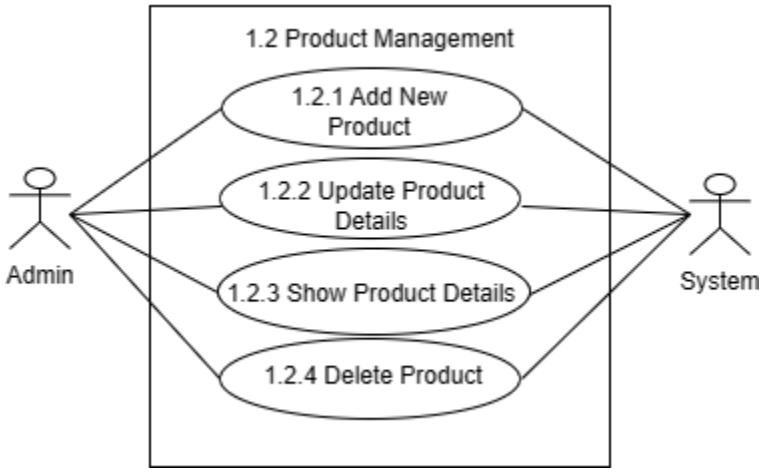


Fig-(6.4): Level-1.2 Use case-Product management

Description of Use case diagram level-1.2:
The Admin can add new products to the inventories, delete and update product details from the inventories as well as can also see the details of all the products present in the inventories.

Action Reply:
User (Administrator):

A1: User selects add new products, delete products, updates product details options
R1: The system acts accordingly and saves the changes.

A2: User selects the show details option`
R2: The system will show all the details of the available products in the inventories.

LEVEL-1.3 USE CASE - Customer management

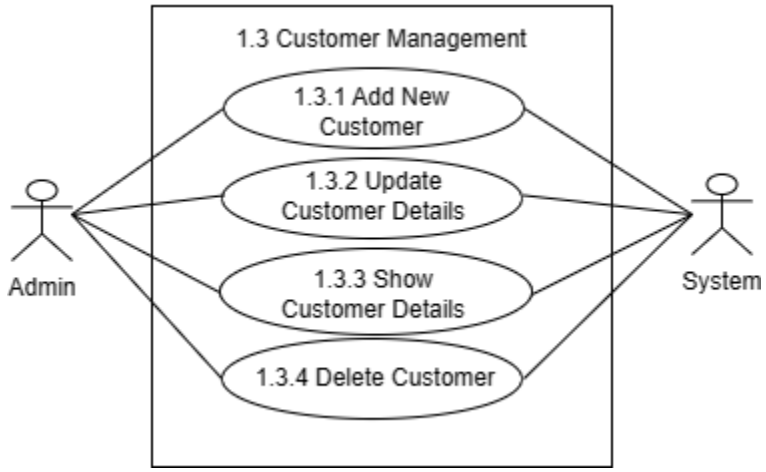


Fig-(6.5): Level-1.3 Use case-Customer management

Description of Use case diagram level-1.3:
The Admin can add new customers to the database, delete and update customer details from the database as well as can also see the details of all the customers from the database.

Action Reply:
User (Administrator):

A1: User selects add new customers, delete customers, updates customers details options
R1: The system acts accordingly and saves the changes.

A2: User selects the show details option`
R2: The system will show all the details of all the customers from the database.

LEVEL-1.4 USE CASE - Sales and Invoice management

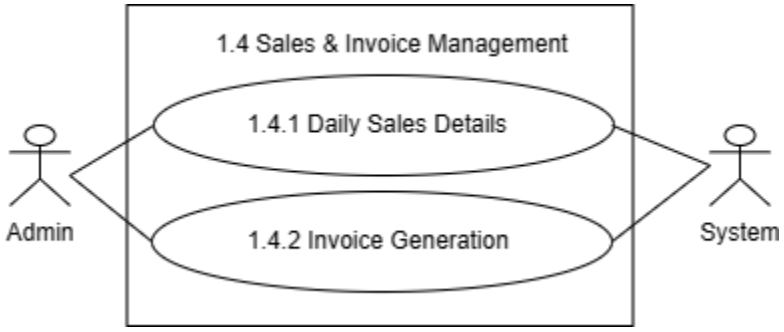


Fig-(6.6): Level-1.4 Use case-Sales and Invoice management

Description of Use case diagram level-1.4:
The Admin can see the details of the daily sales as well as will be able to generate invoices for the customers.

Action Reply:
User (Administrator):

A1: User selects show Daily sales option
R1: The system acts accordingly and shows the details.

A2: User selects the invoice generation
R2: The system will generate a PDF of the invoice for the customer.

LEVEL-1.4.2 USE CASE - Invoice generation

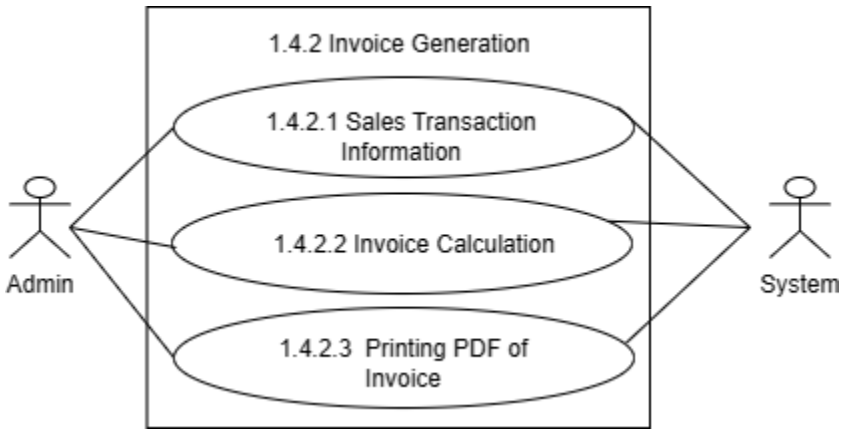


Fig-(6.7): Level-1.4.2 Use case-Invoice generation

Description of Use case diagram level-1.4.2:
The Admin can get the details of the customer and purchased products required for the generation of the invoice. The Calculation of the total price is generated depending on the membership status of the customer. A PDF of the invoice will be generated for the customers.

Action Reply:
User (Administrator):

A1: User gives input to the necessary attributes to get the sales transaction information (Customer and Purchased product)
R1: The system acts accordingly and shows the details.

A2: The user sees the membership of the customer and takes an input into the quantities of the purchased products
R2: The system will generate a total price according to the membership status.

A3: The user saves the invoice after generating all the information after each sale
R3: The system will automatically generate a PDF of the invoice for the customer.

LEVEL-1.4.2.1 USE CASE - Sales transaction information

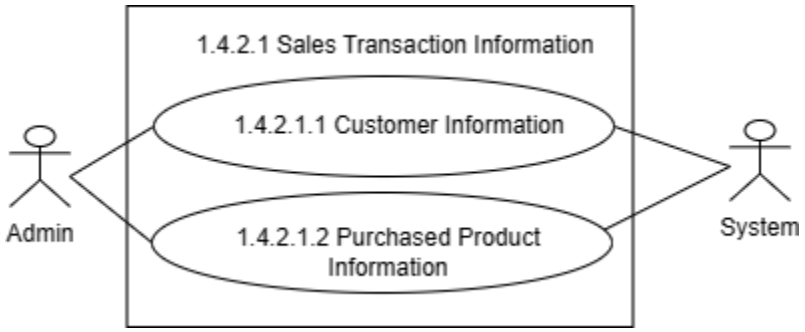


Fig-(6.8): Level-1.4.2.1 Use case-Sales transaction information

Description of Use case diagram level-1.4.2.1:
The Admin can automatically generate all the necessary information in the respective fields using a unique key(phone number). By giving the product name which is the unique key for the product fields, the user can automatically generate all the necessary information for the sales transaction.

Action Reply:
User (Administrator):

A1: User gives input in the Phone number field
R1: The system will automatically give the customer's information in the respective fields.

A2: User gives input in the Product name field
R2: The system will automatically give the customer's information in the respective fields.

LEVEL-1.4.2.2 USE CASE - Invoice Calculation

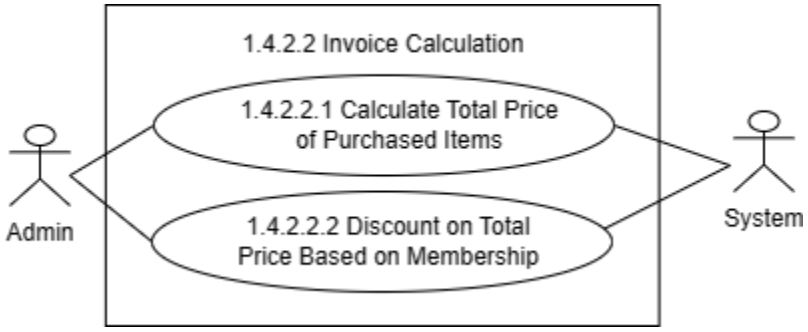


Fig-(6.9): Level-1.4.2.2 Price calculation and discount application

Description of Use case diagram level-1.4.2.2:
The Admin can automatically generate the total price in the invoice according to the membership status of the customer. If the customer has a membership, there will be a 5%
discount in the total price. On the other hand, if the customer doesn't have a membership, the total price will stay the same.

Action Reply:
User (Administrator):

A1: The user provides an input where the customer has membership status
R1: The system will automatically subtract 5% of the total price from the total price and include the total price in the invoice.

A2: The user provides an input where the customer doesn't have a membership status
R2: The system will not provide any discount and the total price will stay the same as before and will include the total price in the invoice.

6.2 Swimlane diagrams

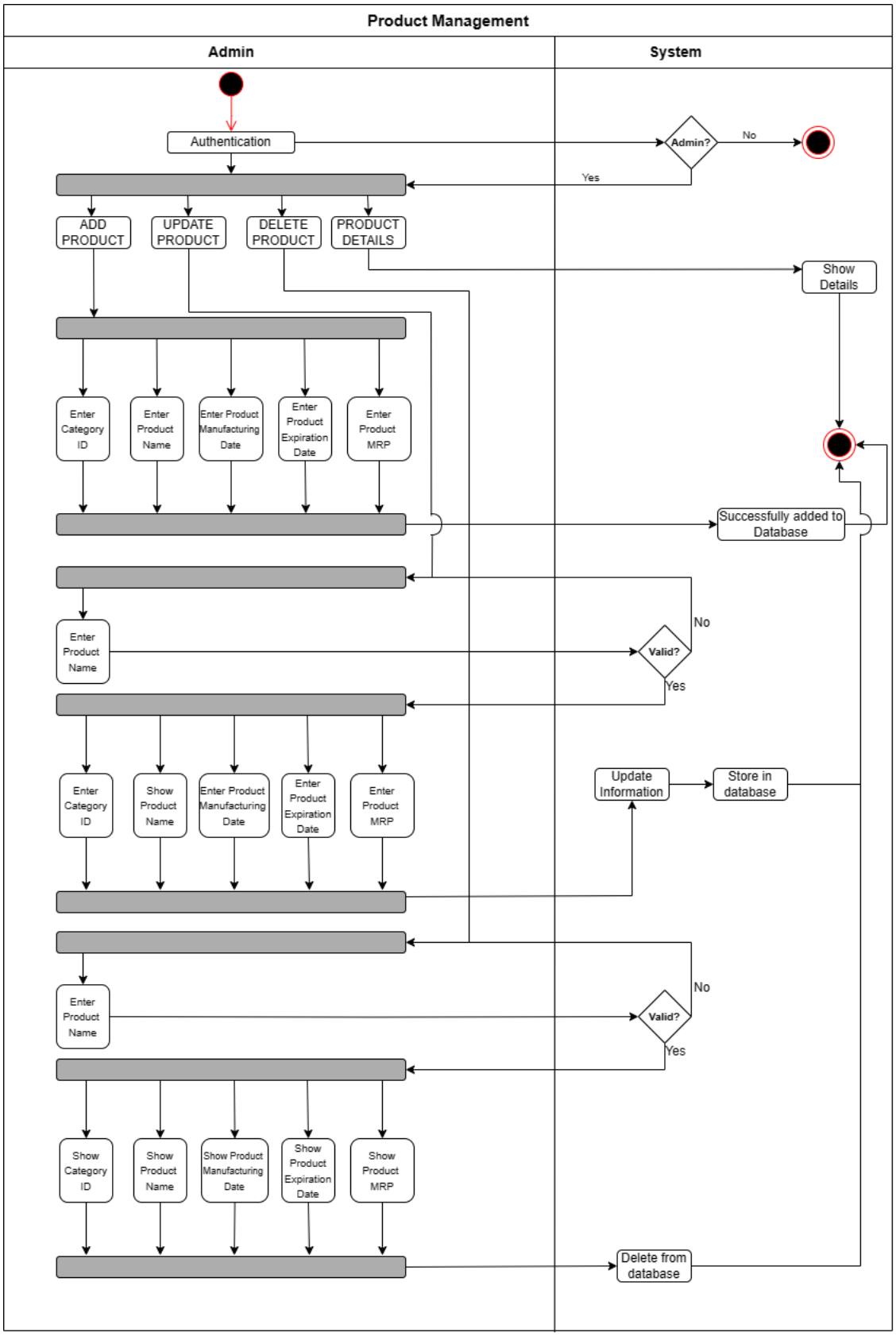


Fig-(6.10): Product Management

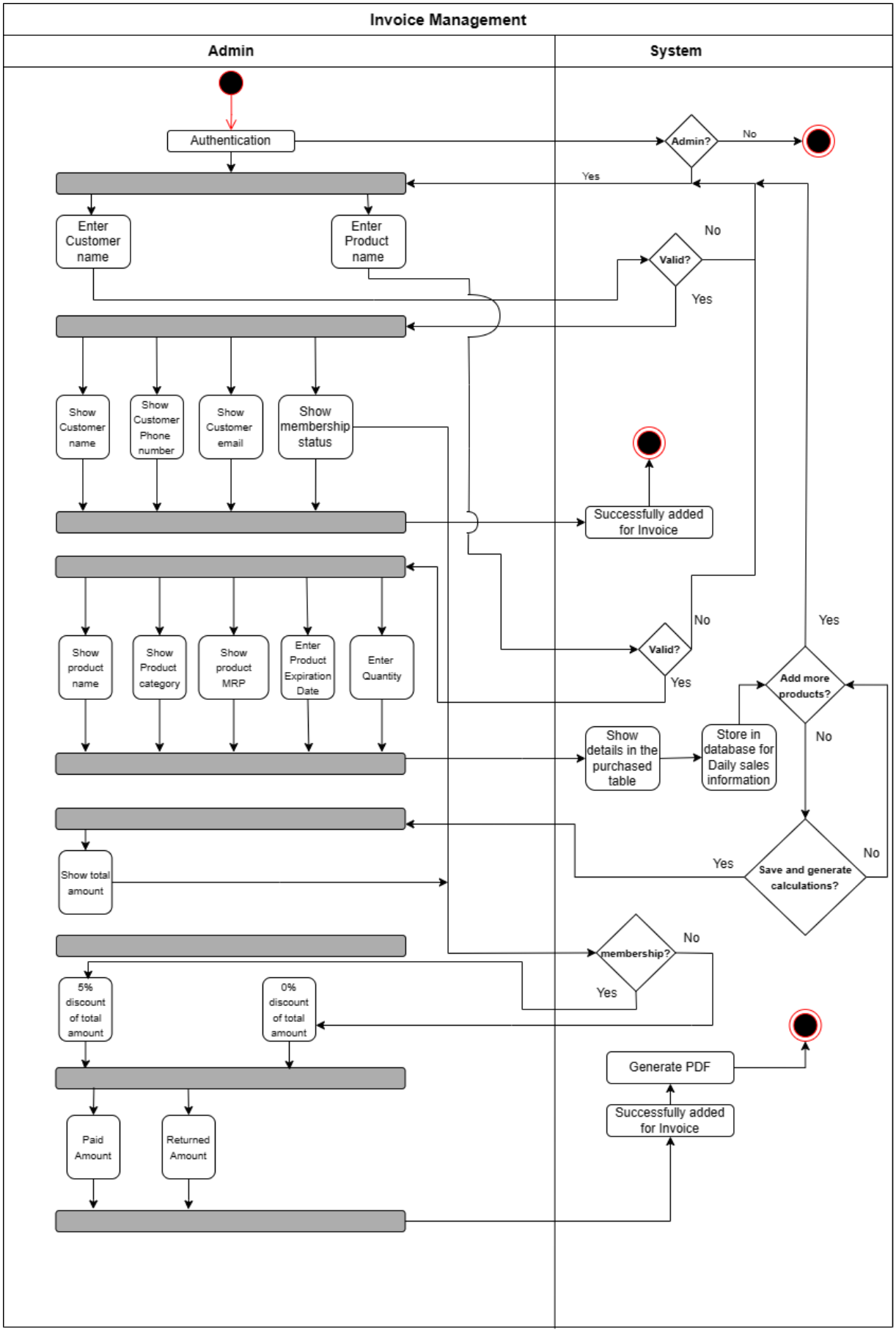


Fig-(6.11): Invoice Management

6.3 Class diagram

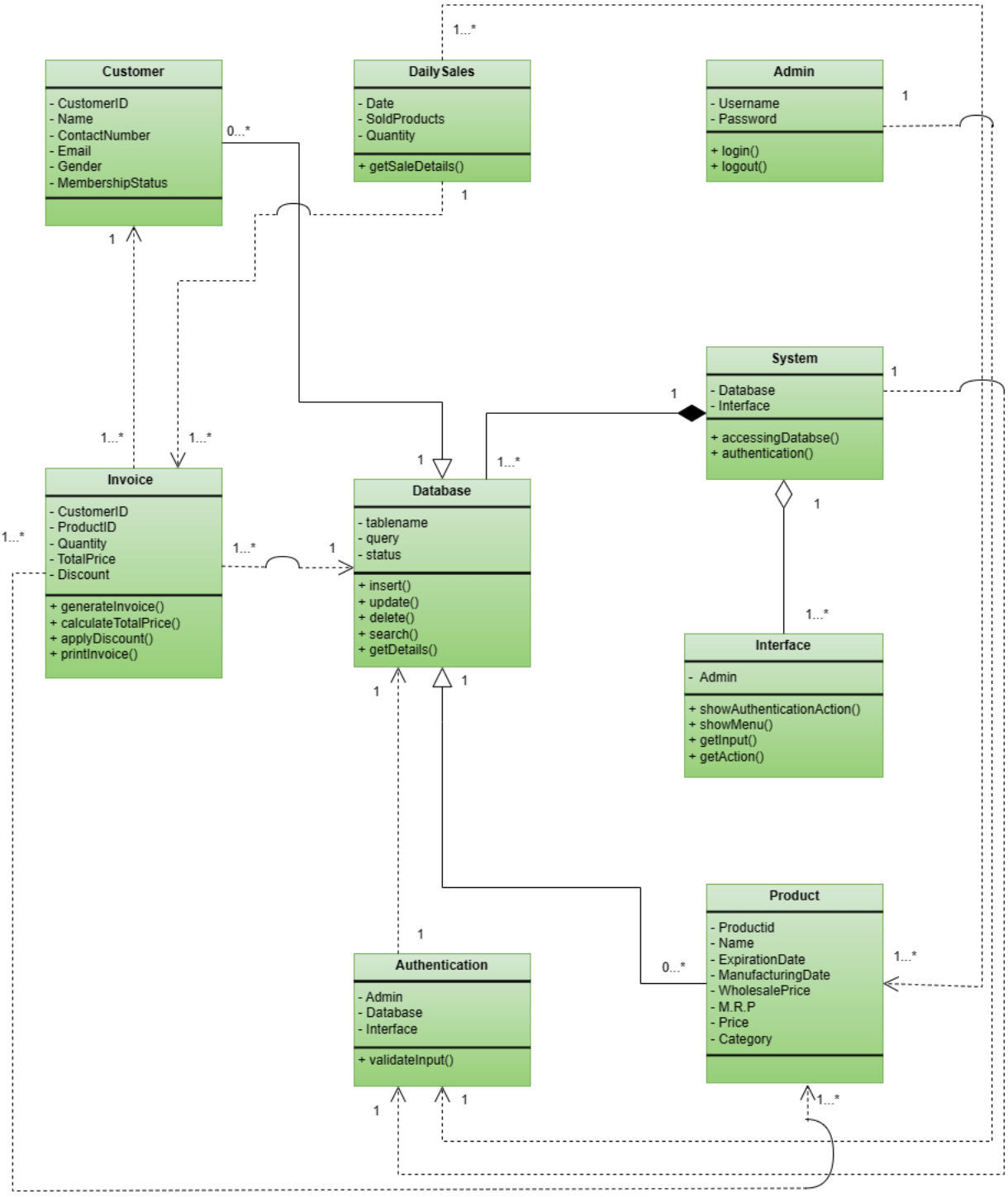
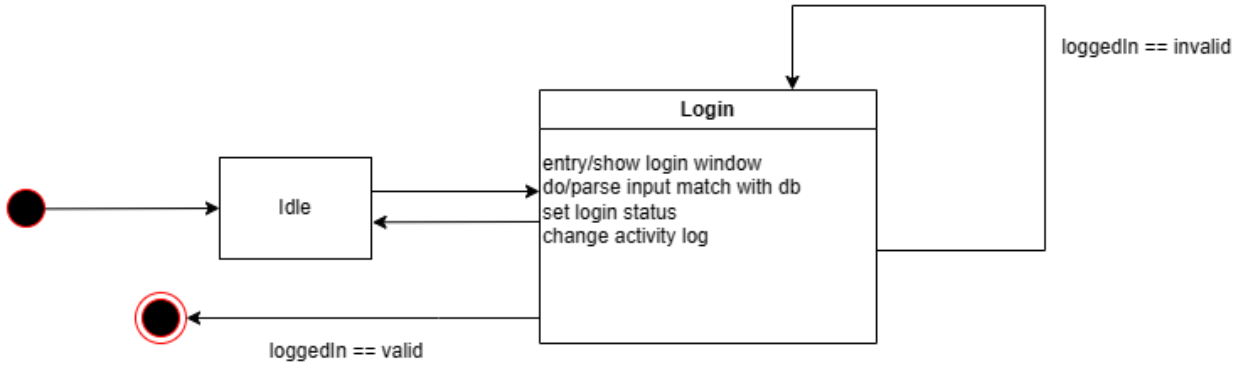
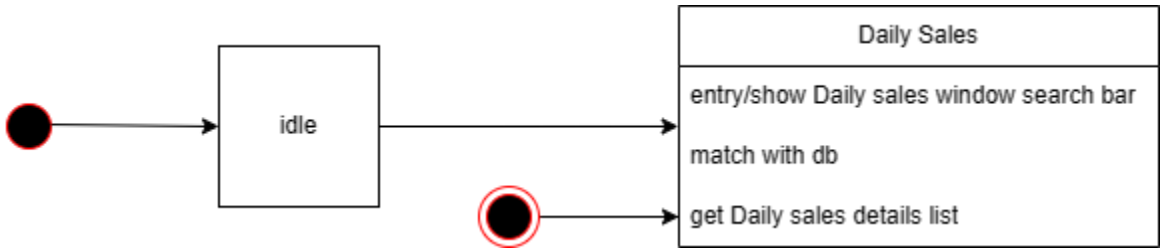


Fig-(6.12): Class Diagram

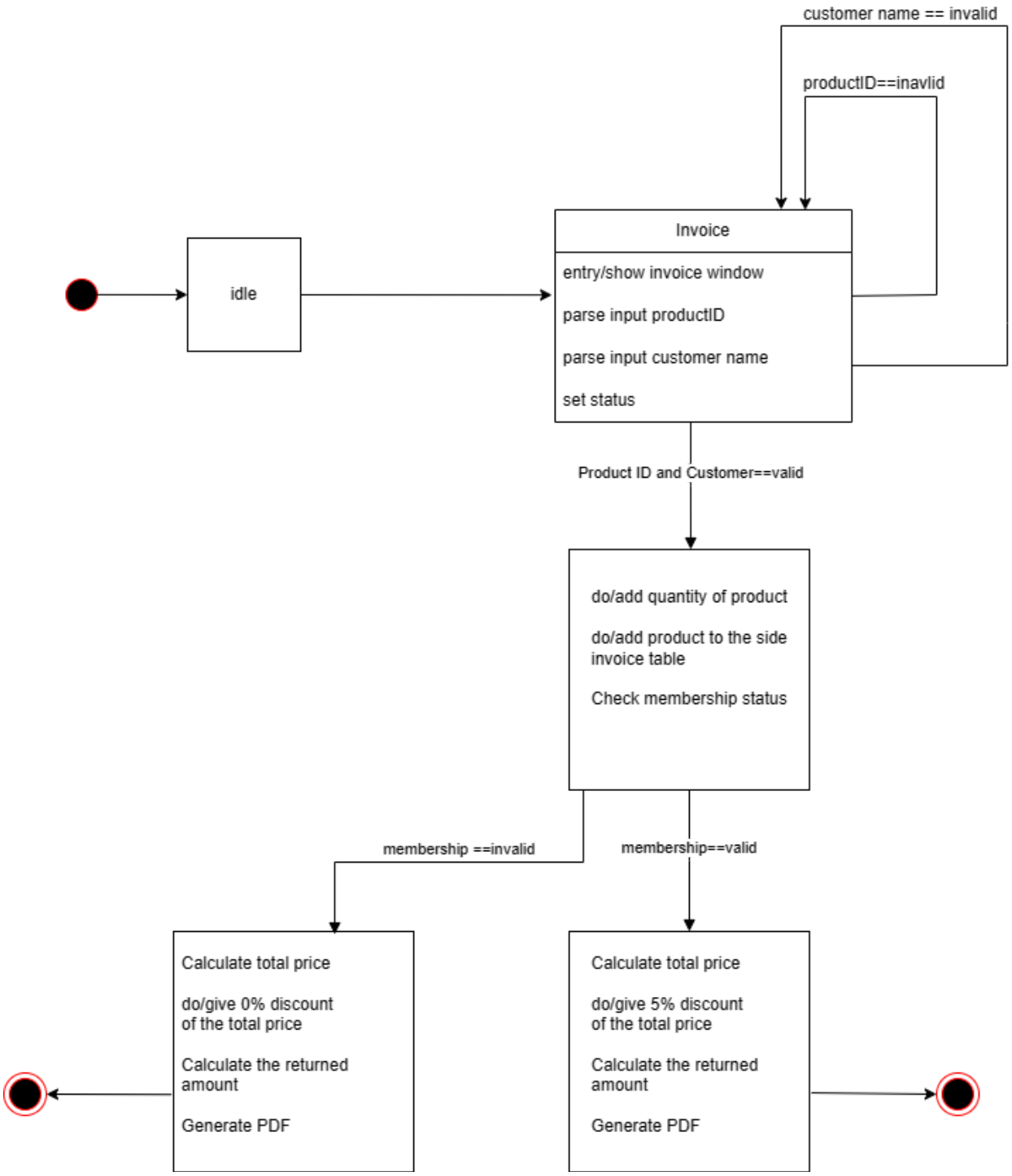
6.4 State diagrams



Fig(6.13): Login

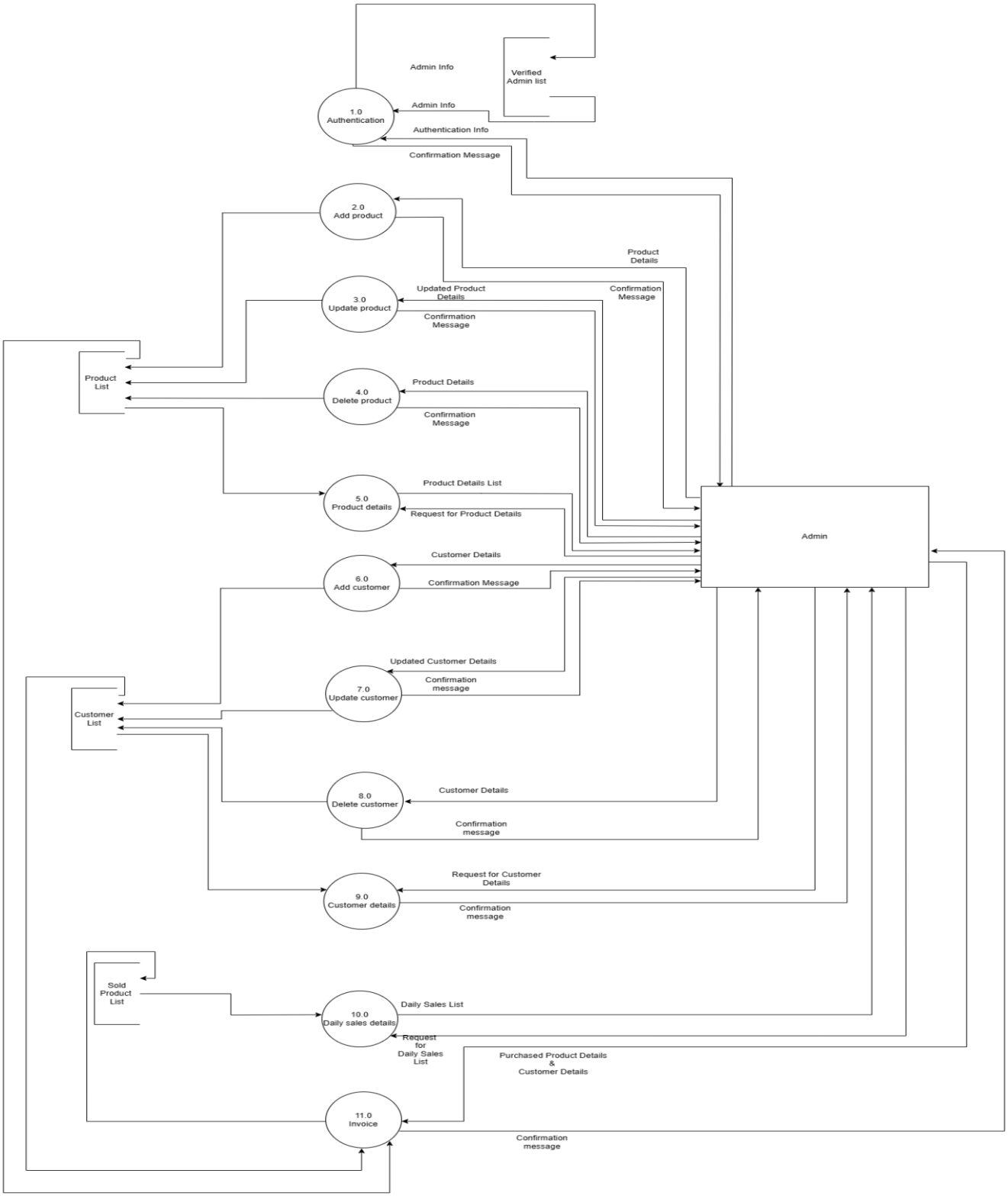


Fig(6.14): Daily Sales



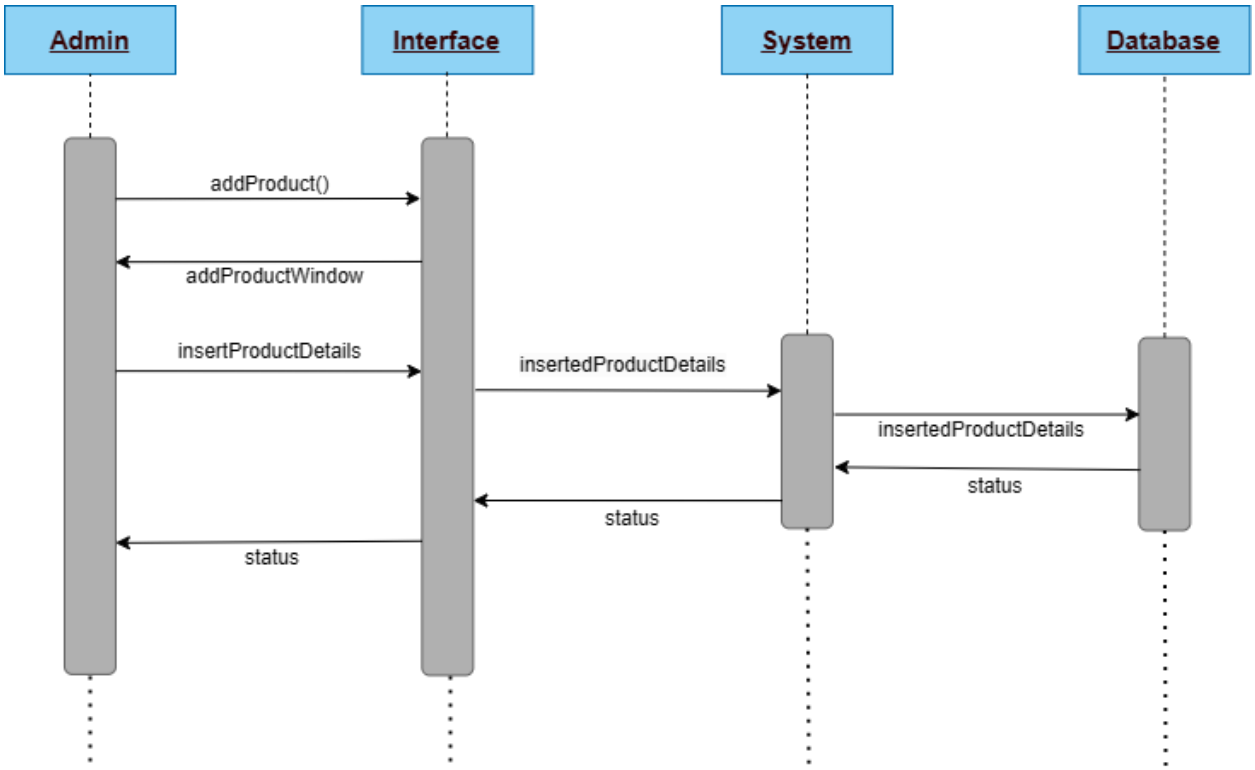
Fig(6.15): Invoice

6.5 Data flow diagram

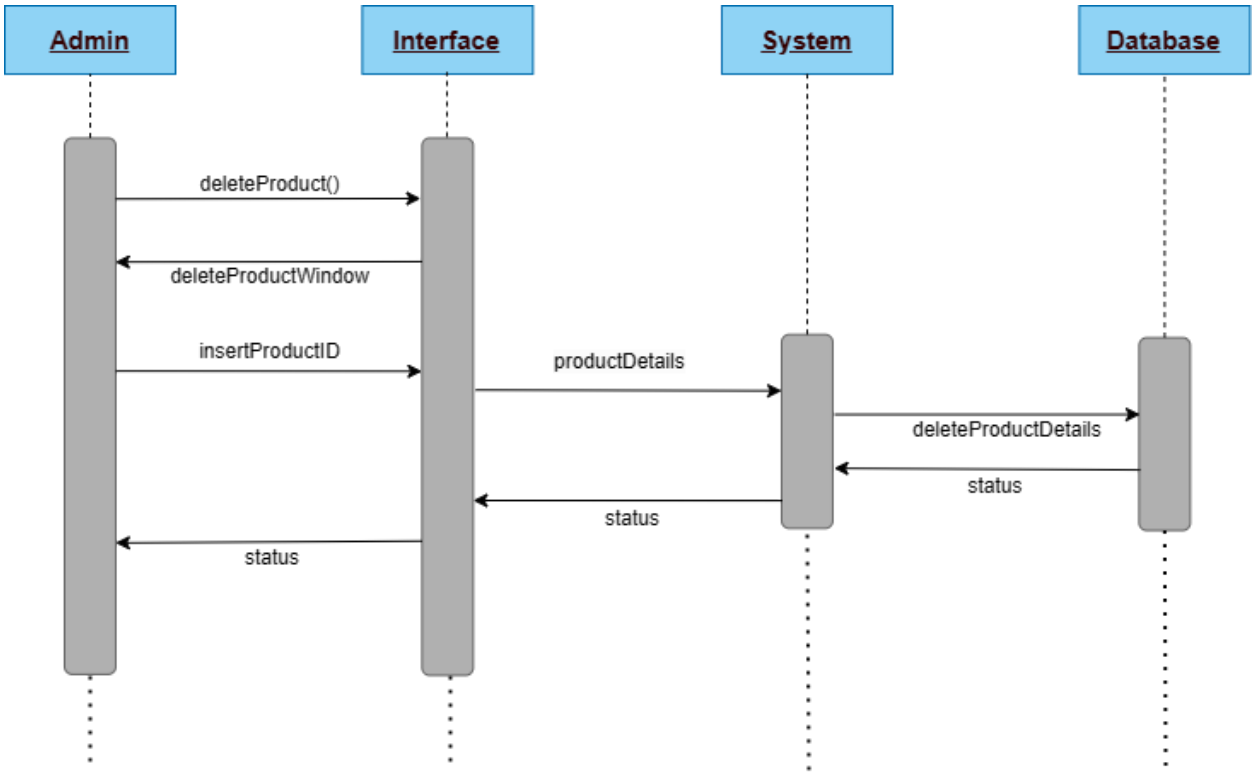


Fig(6.16): SuperShop Management System

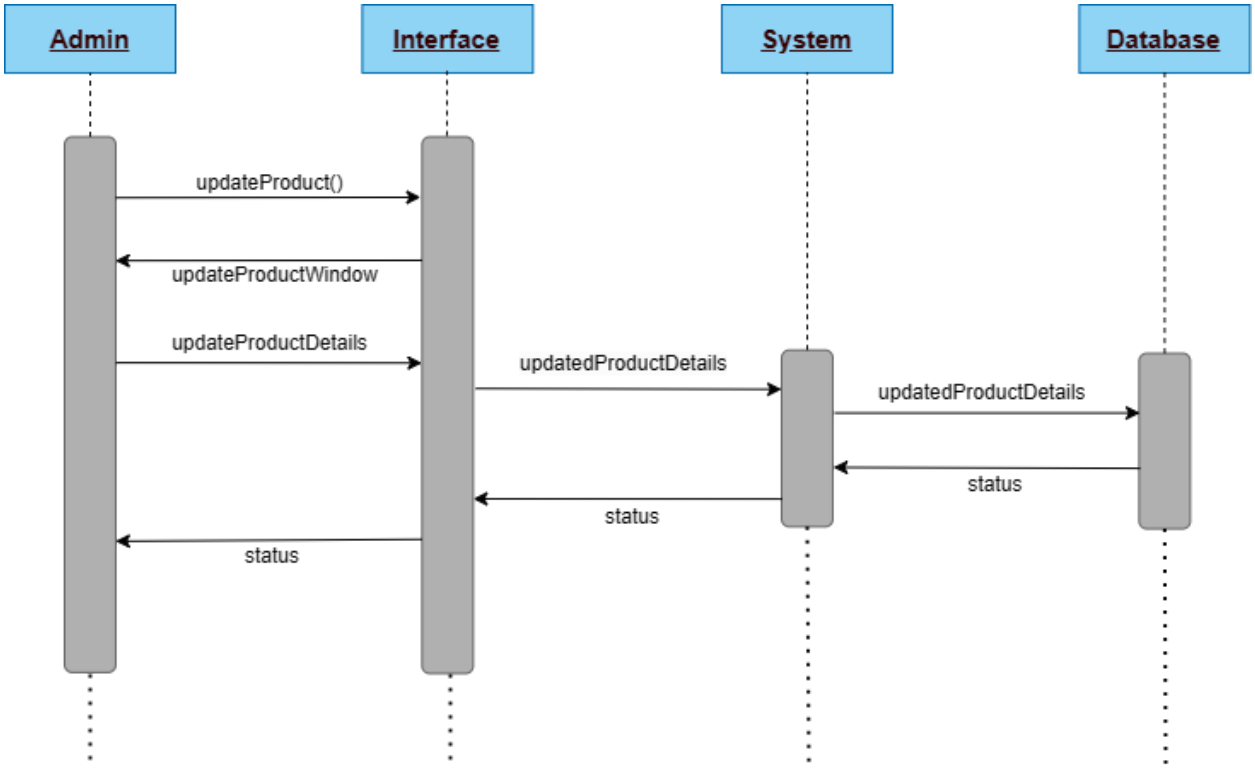
6.6 Sequence diagram



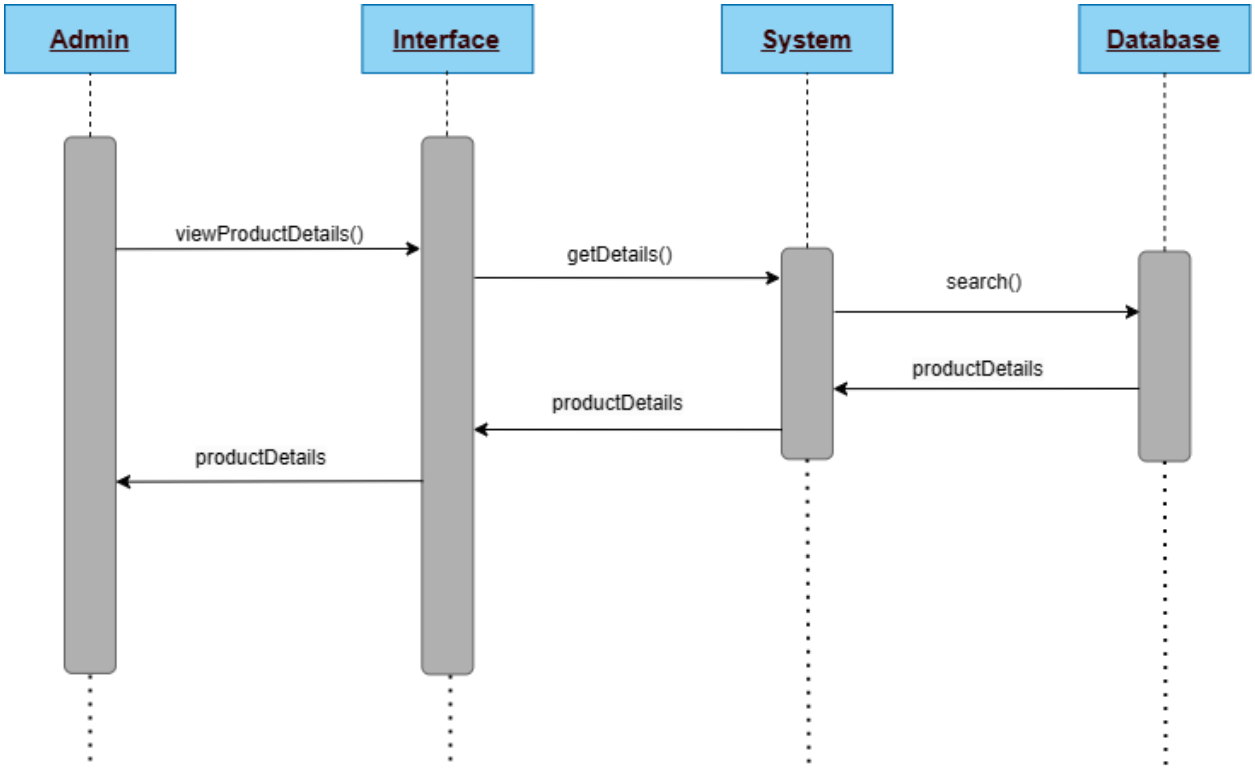
Fig(6.17): Add Product



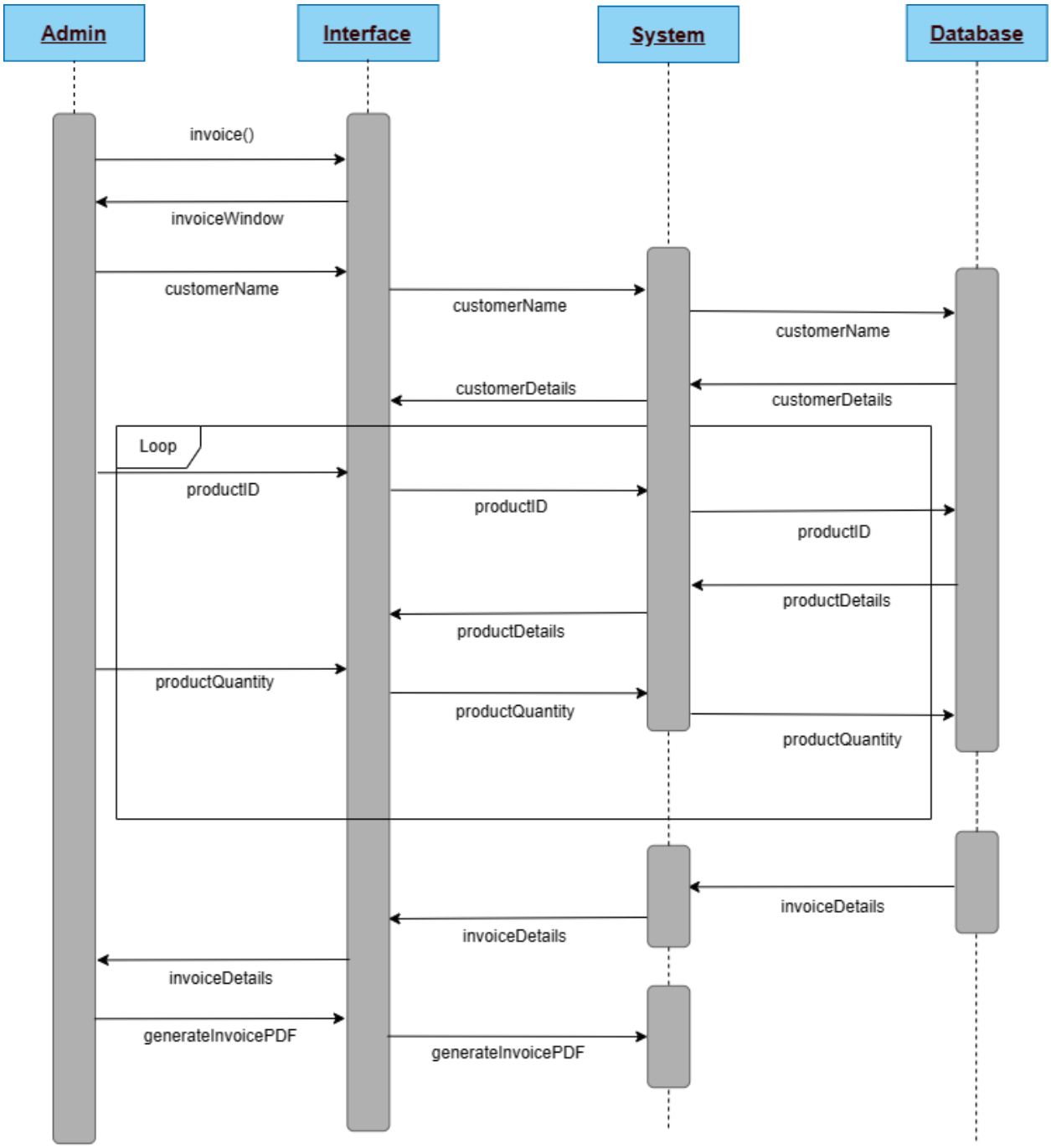
Fig(6.18): Delete Product



Fig(6.19): Update Product



Fig(6.20): Product Details



Fig(6.21): Invoice