Prototype Design on

"Cash & Carry"

Prepared for:

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

Course Number & Name:

CSE 3224 - Information & System Design Lab

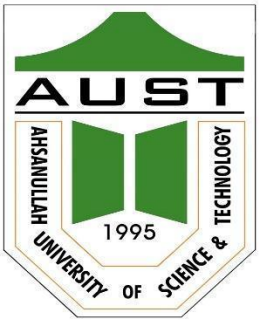
Prepared by:

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: 14.06.2023



Ahsanullah University Of Science & Technology

Department of Computer Science and Engineering

Project Overview:

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.

User Stories and Relevant UI:

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.

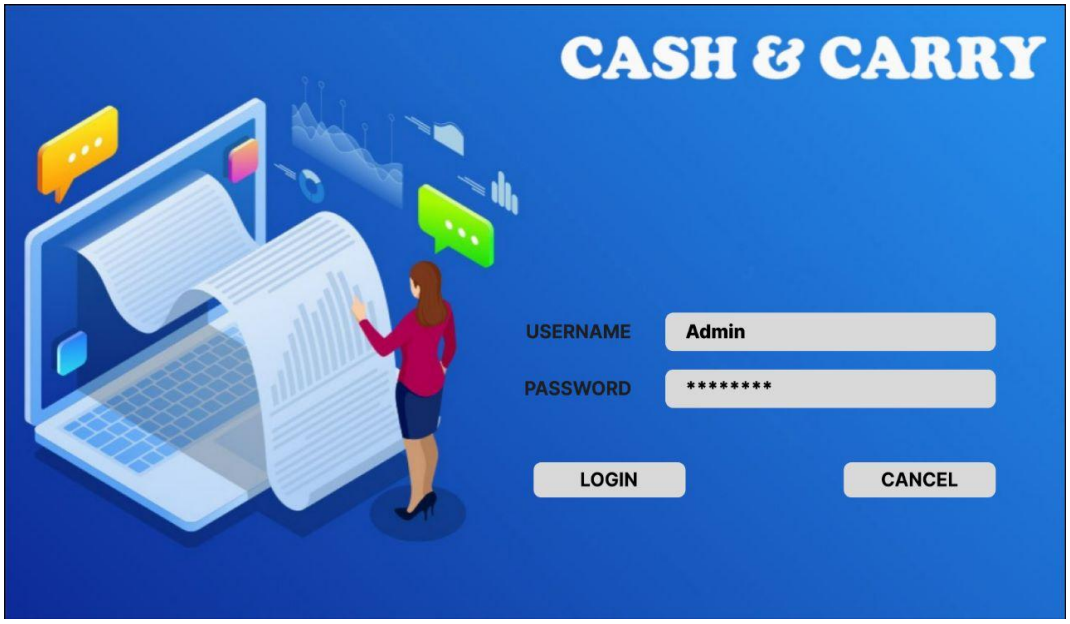


Fig-(1): Login UI

Fig-(2): Home Page UI

➢ Here, by implementing the ***Fig-(1): Login UI*** we can accomplish this user story. Action of the login button will open the ***Fig-(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): Home Page UI

Fig-(4): New Product UI

➢ To add new products into the inventories, there will be an interaction between the ***Fig-(3): Home Page UI*** and ***Fig-(4): New Product UI.*** When the New product button is pressed from the ***Fig-(3): Home Page UI***, the ***Fig-(4): New Product UI*** will pop up, for the insertion of the details of new products and by clicking the save button from ***Fig-(4): New Product UI,*** all the information will be saved in database.

Fig-(5): Home Page UI

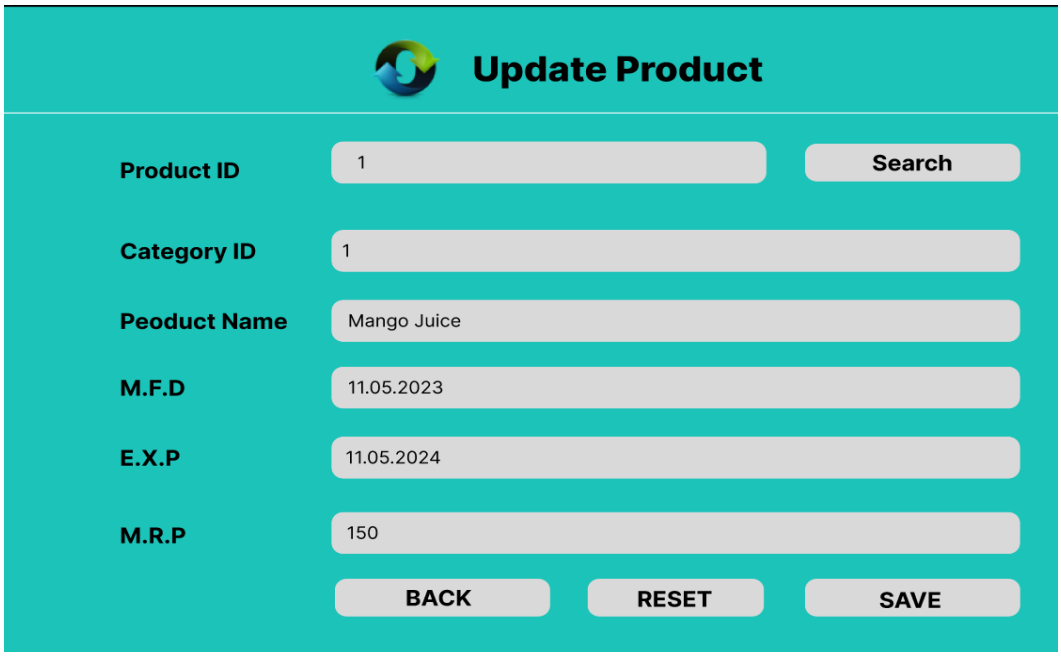


Fig-(6): Update Product UI

➢ To update products from the inventories, there needs to be an interaction between the ***Fig-(5): Home Page UI*** and ***Fig-(6): Update Product UI.*** When the Update product button is pressed from the ***Fig-(5): Home Page UI***, the ***Fig-(6): Update Product UI*** will pop up, for the insertion of the updation of the products and by clicking the save button from ***Fig-(6): Update Product UI***, all the information will be saved in the database and the reset button will clear all the fields.

Fig-(7): Home Page UI

Fig-(8): Product Details UI

➢ To see the details of the products from the inventories, there needs to be an interaction between the ***Fig-(7): Home Page UI*** and ***Fig-(8): Product Details UI.*** When the product details button is pressed from the ***Fig-(7): Home Page UI***, the ***Fig-(8): Product Details UI.*** will pop up and will give us all the detailed information of the products accessing from the database.

Fig-(9): Home Page UI

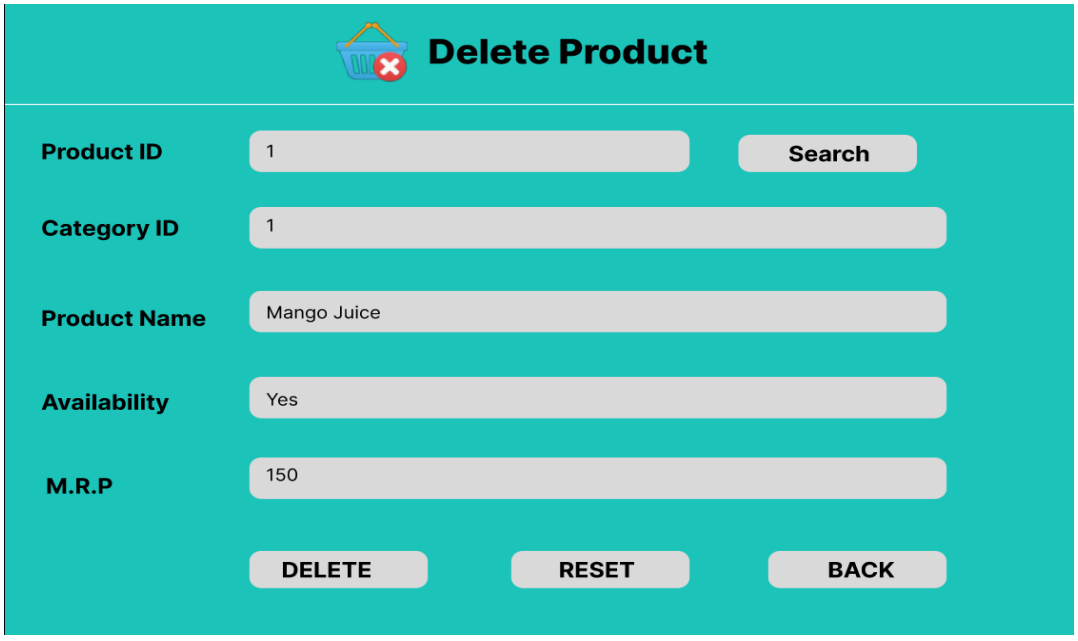


Fig-(10): Delete Product UI

➢ To delete products from the inventories, there needs to be an interaction between the ***Fig-(9): Home Page UI*** and ***Fig-(10): Delete Product UI.*** When the Delete product button is pressed from the ***Fig-(9): Home Page UI***, the ***Fig-(10): Delete Product UI*** will pop up, for the deletion of the products and by clicking the delete button from ***Fig-(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-(10): Home Page UI

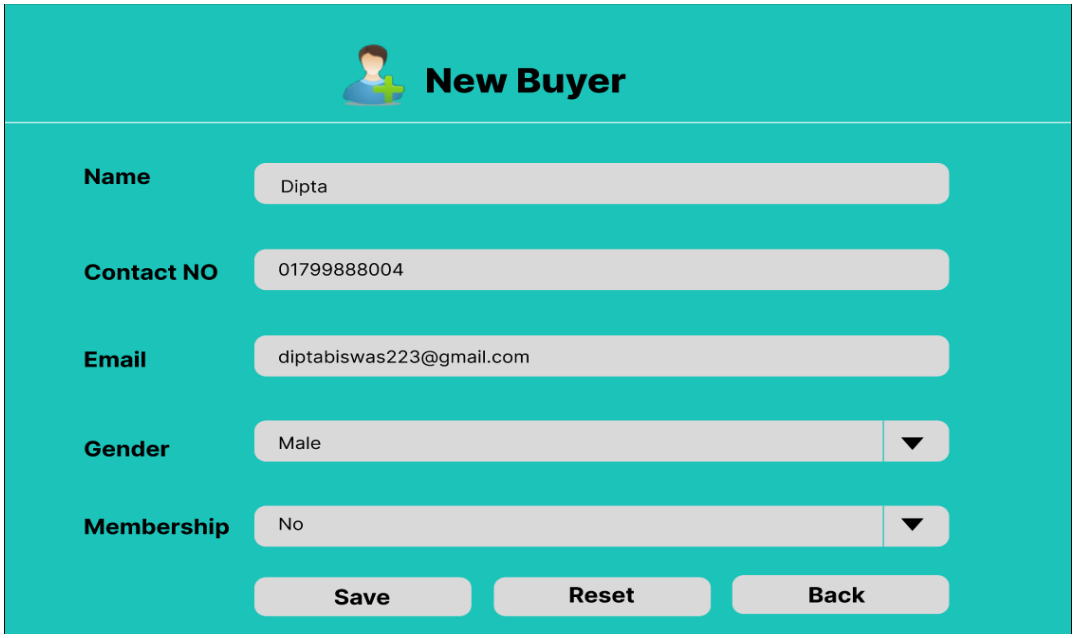


Fig-(11): New Buyer UI

➢ To add new buyers into the database, there will be an interaction between the ***Fig-(10): Home Page UI*** and ***Fig-(11): New Buyer UI.*** When the new buyer button is pressed from the ***Fig-(10): Home Page UI***, the ***Fig-(11): New Buyer UI*** will pop up, for the insertion of the details of a new buyer and by clicking the save button from ***Fig-(11): New Buyer UI***, all the information will be saved in database.

Fig-(12): Home Page UI

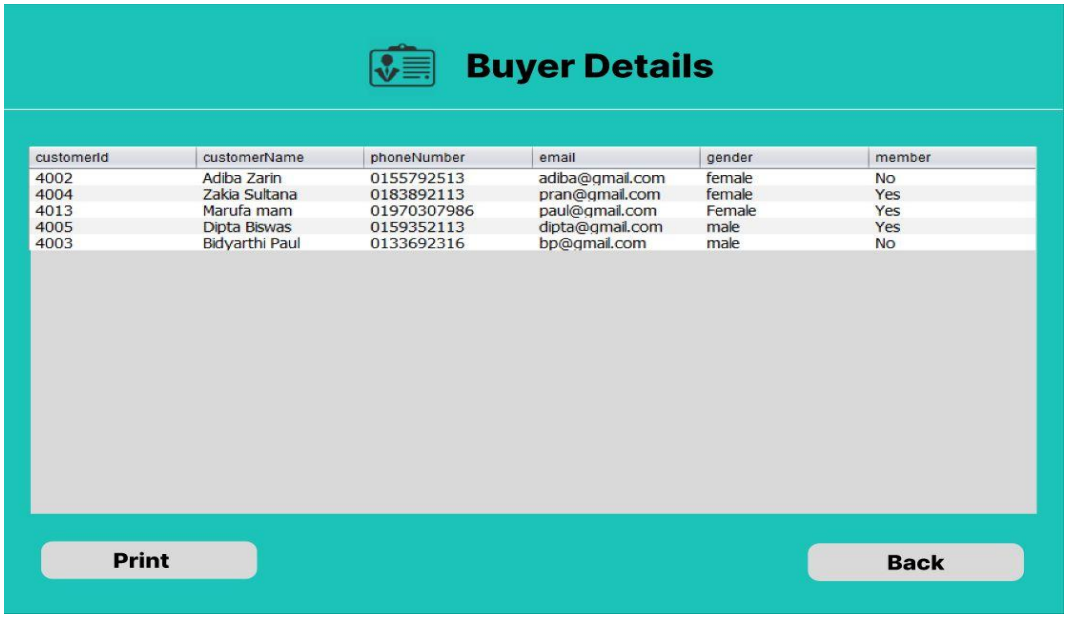


Fig-(13): Buyer Details UI

➢ To see the details of the buyers from the database, there needs to be an interaction between the ***Fig-(12): Home Page UI*** and ***Fig-(13): Buyer Details UI.*** When the buyer details button is pressed from the ***Fig-(12): Home Page UI***, the ***Fig-(13): Buyer Details UI.*** will pop up and will give us all the detailed information of the buyers accessing from the database.

Fig-(14): Home Page UI

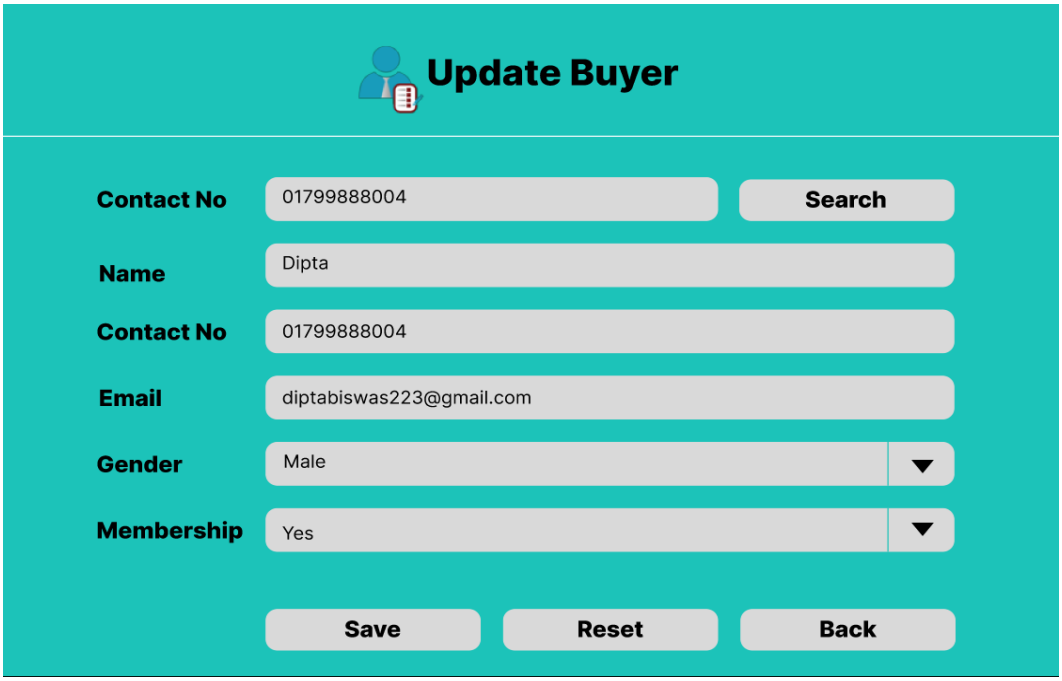


Fig-(15): Update Buyer UI

➢ To update buyer information from the database, there needs to be an interaction between the ***Fig-(14): Home Page UI*** and ***Fig-(15): Update Buyer UI.*** When the Update buyer button is pressed from the ***Fig-(14): Home Page UI***, the ***Fig-(15): Update Buyer UI*** will pop up, for the insertion of the updation of the products and by clicking the save button from ***Fig-(15): Update Buyer UI***, all the information will be saved in the database and the reset button will clear all the fields.

Fig-(16): Home Page UI

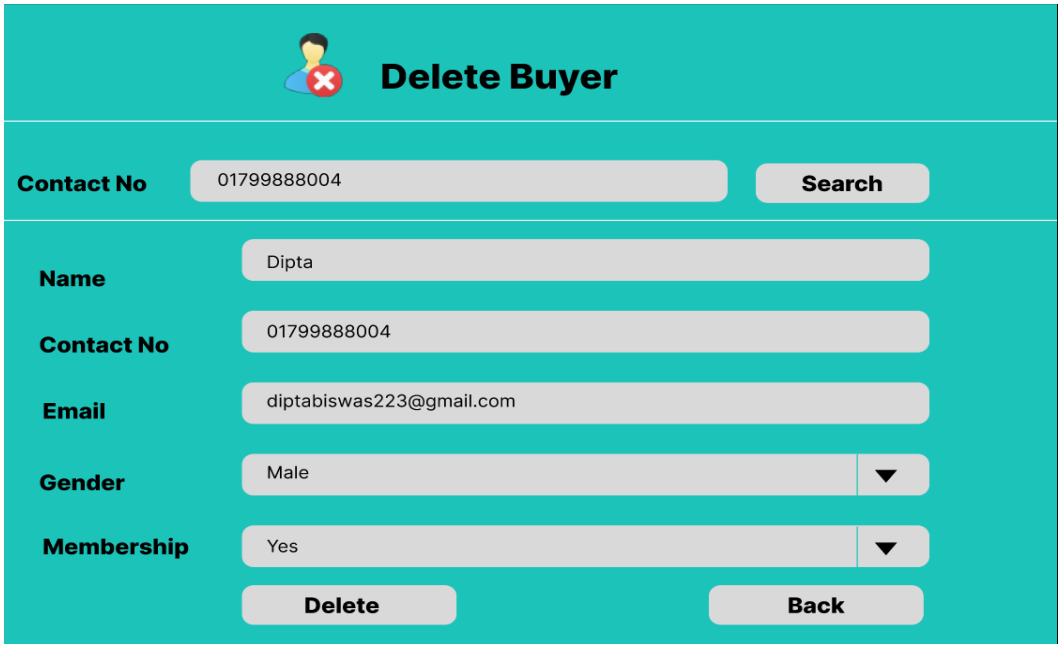


Fig-(17): Delete Buyer UI

➢ To delete any particular buyer from the database, there needs to be an interaction between the ***Fig-(16): Home Page UI*** and ***Fig-(17): Delete Buyer UI.*** When the Delete buyer button is pressed from the ***Fig-(16): Home Page UI***, the ***Fig-(17): Delete Buyer UI*** will pop up, for the deletion of the buyer's information and by clicking the delete button from***Fig-(17): 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-(18): Home Page UI

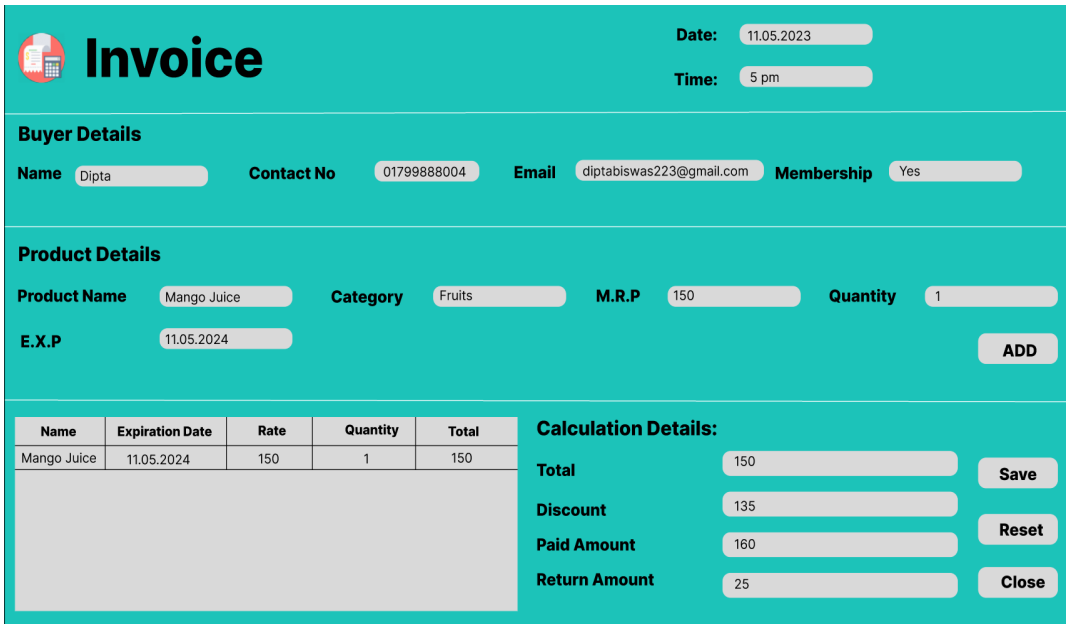


Fig-(19): Invoice UI (With Membership)

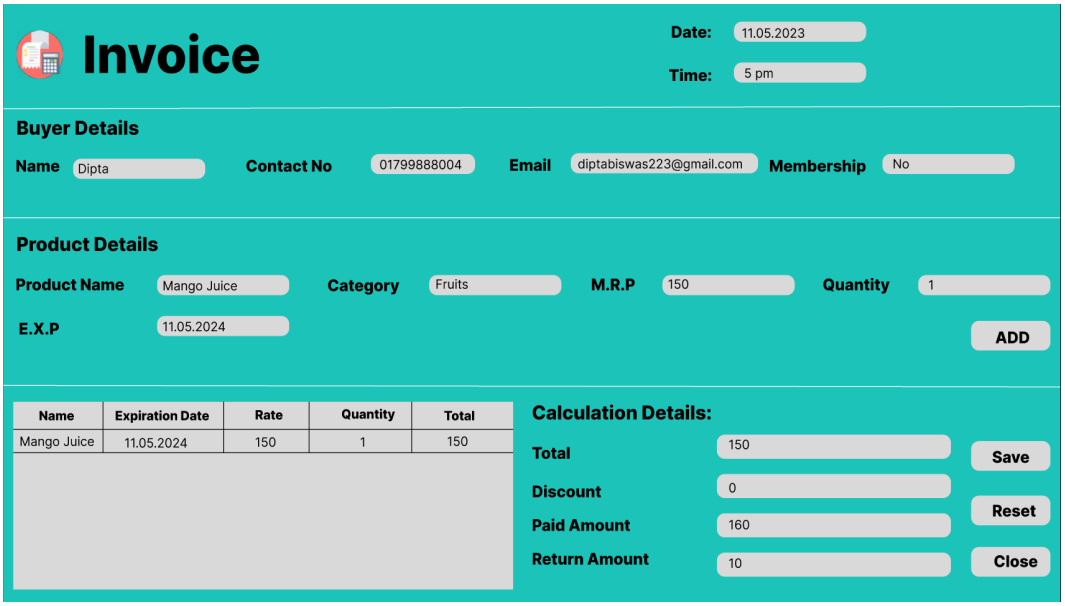
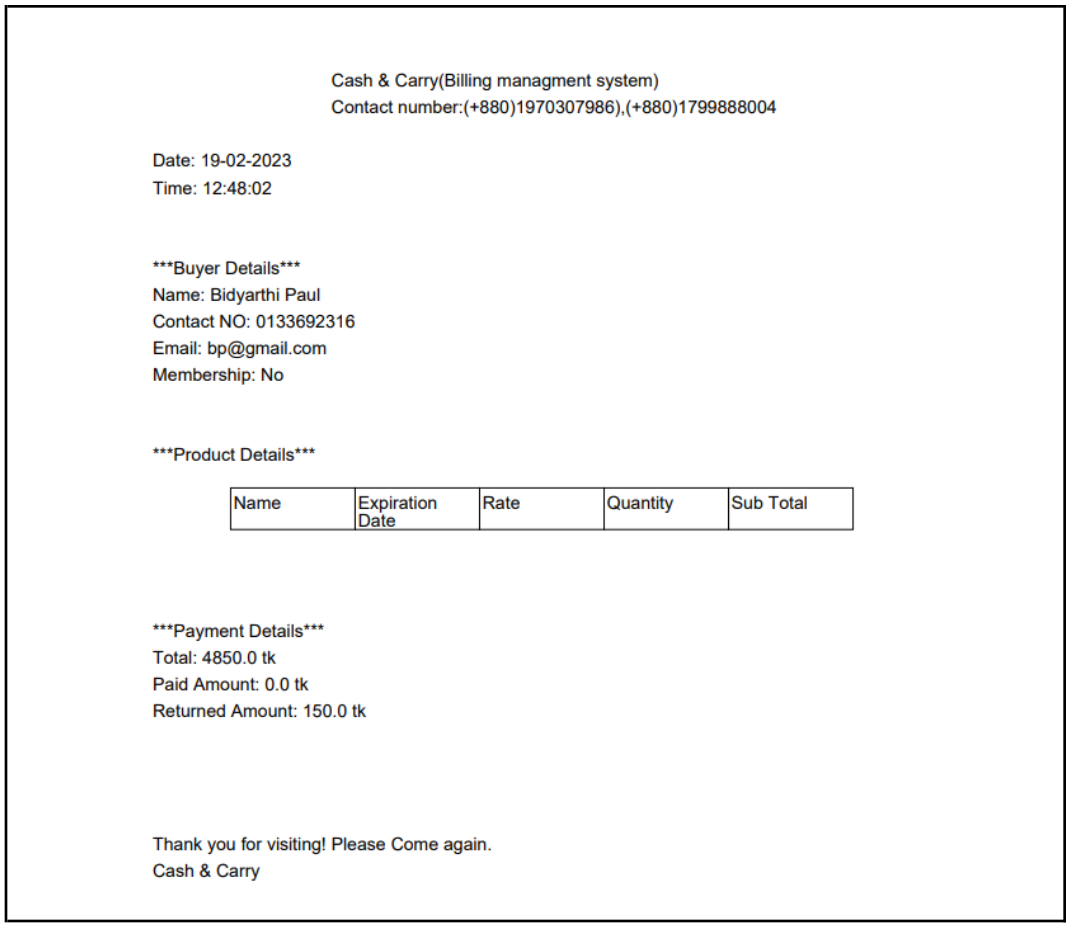


Fig-(20): Invoice UI (Without Membership)



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-(21): Invoice Pdf

➢ To fulfill the above user story, there will be an interaction between ***Fig-(18): Home Page UI, Fig-(19): Invoice UI (With Membership)*** and ***Fig-(20): Invoice UI (Without Membership)*** **.** When the invoice button is clicked from ***Fig-(18): 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-(19): 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-(20): 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-(21): Invoice PDF*** which includes all the information of products, customers and payment details.

Fig-(22): Home Page UI

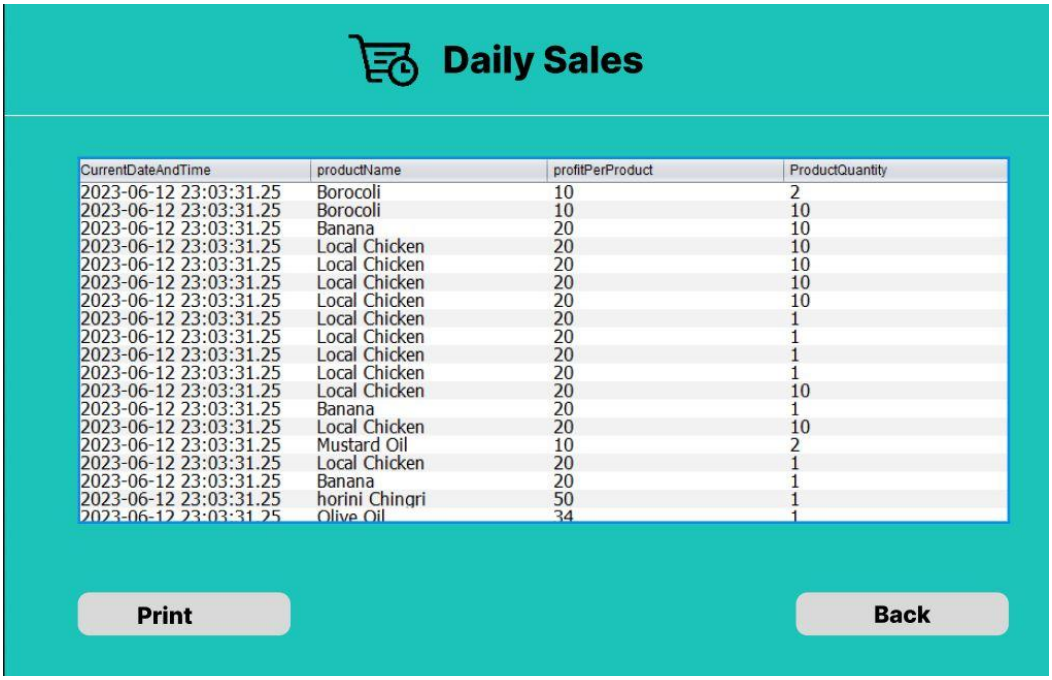


Fig-(23):Daily Sales UI

➢ To see the details of all the products that have been sold on a particular day, there will be an interaction between the ***Fig-(22): Home Page UI*** and ***Fig-(23): 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-(24): Home Page UI (Log Out)

➢ To log out of the system the close app button needs to be pressed from the ***Fig-(24): Home Page UI (Log Out),*** which will take the user back to the login page.

Fig-(25): Home Page UI (Close App)

➢ To terminate the system the close app button needs to be pressed from the ***Fig-(25): Home Page UI (Close App),*** which will close the whole system.