

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA**



Mini Project Report on

“Billing System”

Submitted in the partial fulfillment for the requirements of the course

“DBMS LABORATORY WITH MINI PROJECT (18CSL58)”

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

by

Mr. PRANAV R DESHKULKARNI

USN: 1BY18IS087

Mr. SHOHEBAHMED NAJEERAHMEED GADAWALE

USN: 1BY18IS112

Under the guidance of

Mrs. DRAKSHAVENI G

Assistant Professor,
Dept. of MCA, BMSIT&M



BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project based learning entitled “**Billing System**” is a bonafide work carried out by **Mr. PRANAV R DESHKULKARNI USN: 1BY18IS087**, **Mr. SHOHEBAHMED NAJEERAHMED GADAWALE USN: 1BY18IS112** during the academic year 2020-21. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library.

Signature of the Faculty

Mrs. Drakshaveni G,
Assistant Professor,
Dept. of MCA, BMSIT&M

Signature of the HOD

Dr. Pushpa S K,
Professor and HOD,
Dept. of ISE, BMSIT&M

ACKNOWLEDGEMENT

We are happy to present this Mini Project after completing it successfully. This Mini Project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us make this Mini Project a success.

We are grateful to **Dr. Mohan Babu G N**, Principal, BMS Institute of Technology & Management for his constant encouragement and support.

We heartily thank **Dr. Pushpa S K**, Head of the Department, Information Science and Engineering, for her constant encouragement and inspiration in taking up this project.

We would like to express our sincere thanks to, guide **Prof. DRAKSHAVENI G** for her guidance, right from the conception and visualization to the very presentation of the project.

We are greatly indebted to our faculties, both teaching and supporting staff, Department of Information Science and Engineering, who took great interest in our project work. They motivated and guided us throughout the accomplishment of this goal. We express our profound thanks for their meticulous guidance.

Finally, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends for their help and wishes for the successful completion of this project work

PRANAV R DESHKULKARNI
1BY18IS087

SHOHEBAHMED NAJEERAHMEED GADAWALE
1BY18IS112

DECLARATION

We hereby declare that we own the full responsibility for the information, results etc. provided in this PROJECT titled “**Billing System**” submitted to BMSIT&M for the award of B.E (ISE). We have taken care in all respect to honor the intellectual property rights and have acknowledged the contribution of others for using them in academic purpose and further declare that in case of any violation of intellectual property right or copyright we, as a candidate, will be fully responsible for the same. My supervisor should not be held responsible for full or partial violation of copyright or intellectual property right.



PRANAV R DESHKULKARNI



SHOHEBAHMED NAJEERAHMED GADAWALE

ABSTRACT

Billing System is a DBMS based standalone application which can be implemented in small scale stores and supermarkets. It can be used to store and process the data such as Customer details, billing details, employee details, etc. It helps to easily manage large number of transactions which occur each day. It can save a lot of time and can essentially store huge amounts of data which can be accessed with ease in future. The authority of accessing and modifying details in the application is given only to administrator. Any modifications to be done in the product name or the price can be done only by admin.

Our project aims to produce database system which manages the sales activity done in a store, maintaining the product details, maintaining the records of the sales done for a particular month/year. This application is built using VS Code with C# for Frontend, SqlLite Database for Backend. The application has a user friendly interface and any required modification can be done easily.

TABLE OF CONTENTS

| CHAPTER | TOPIC | Pg.No. |
|---------|--|--------|
| 1 | Introduction 1.1 Overview 1.2 Salient Features | 01 |
| 2 | Required Specifications 2.1 Hardware Requirements 2.2 Software Requirements | 03 |
| 3 | Background 3.1 Visual Studio 3.2 SQLite 3.3 Schema Diagram 3.4 Entity - Relationship Diagram 3.5 Table Description 3.6 Queries | 04 |
| 4 | Case Study 4.1 Grocery Store | 18 |
| 5 | Results | 19 |
| 6 | Conclusion References | 23 |

CHAPTER - 1

Introduction

1.1 Overview

Billing System is the standalone application which can be implemented in a small scale provision store or a supermarket to manage the transactions that occur everyday. All the aspects related to the proper management of Billing is done in the application. These aspects involve managing information about various Products, Orders, Employee, Customers, Payment. This system provides an efficient way of managing the Billing of Supermarket or any provisional store. The authority of accessing and modifying details in the application is given only to administrator. Any modifications to be done in the product name or the price can be done only by admin. The application has a user friendly interface and any required modification can be done easily.

1.2 Salient Features

- The proposed application is basically used in a typical provision store or supermarket to manage the store which includes managing the stock, managing customer details as well as to process the orders of various customers.
- This system provides list of various products currently available in the store and it also stores details of each product such as price, expiry date, etc
- This system also stores customer details as well as their payment details so that any transaction problems or payment issues can be monitored
- It also stores the details of employee working in the store currently and also manages their personal information such as Phone Number, Salary, etc which can be edited in future if required.

CHAPTER - 2

Required Specifications

2.1 Hardware Requirements

- Processor 2.4 GHZ processor speed
- Disk space 80 GB (including 20 GB for database Management system)
- SVGA colour monitor or higher quality.
- RAM 4GB.
- Backup storage hard disk of about 80MB.
- Flash drive for file transfer.
- An enhanced keyboard.
- A power stabilizer.

2.2 Software Requirements

- Visual Studio Code (C# for frontend).
- Operating system; Windows7/8/10.
- SQLite Database (for Backend).

CHAPTER - 3

Background

The Application is developed using

- Visual studio (language used c#, Frontend).
- SQLite (Database, Backend).

3.1 Visual studio

The Visual Studio Integrated Development Environment (IDE) is a creative launching pad that can be used to edit, debug, and build code, and also publish an app. An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process. The Windows Presentation Forms App (Wpf App) in Visual Studio lets us create seamless standalone applications and provides a user-friendly interface to work with. The same has been used to develop this application.

3.2 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

We have used SQLite as our database and as the backend of our application. Our database basically consists of 6 tables. The first one is the **Authentication** table which is used for managing the username and password of the users of the application. The following are the main tables used in our database:

- 1. Orders Table:** Stores data related to the purchases made by the customer.
- 2. Product Table :** Stores the products available in the store along with their details such as Product Name, Price, etc.
- 3. Customer Table:** Stores customer related information such as Customer name, Phone No, address, etc.
- 4. Payment Table:** Stores data related to the transactions made by the customers.
- 5. Employee Table:** Stores personal information of the employees working in the store currently.

We have used the stored procedures for processing each query in the database. Stored procedures are useful as each one can store a specific query and can be called by calling the name of the procedure. It also reduces many runtime errors and provide smooth functioning of the database. Stored procedure have the feature of taking user input and replacing it in the query for further processing. Hence we have used this feature of SQLite.

3.3 Schema Diagram

Orders

| <u>Order_ID</u> | Cust_ID | Pdt_ID | Quantity | P_Date | Emp_ID |
|-----------------|---------|--------|----------|--------|--------|
|-----------------|---------|--------|----------|--------|--------|

Product

| <u>Pdt_ID</u> | Pdt_Name | Price | Exp_Date |
|---------------|----------|-------|----------|
|---------------|----------|-------|----------|

Customer

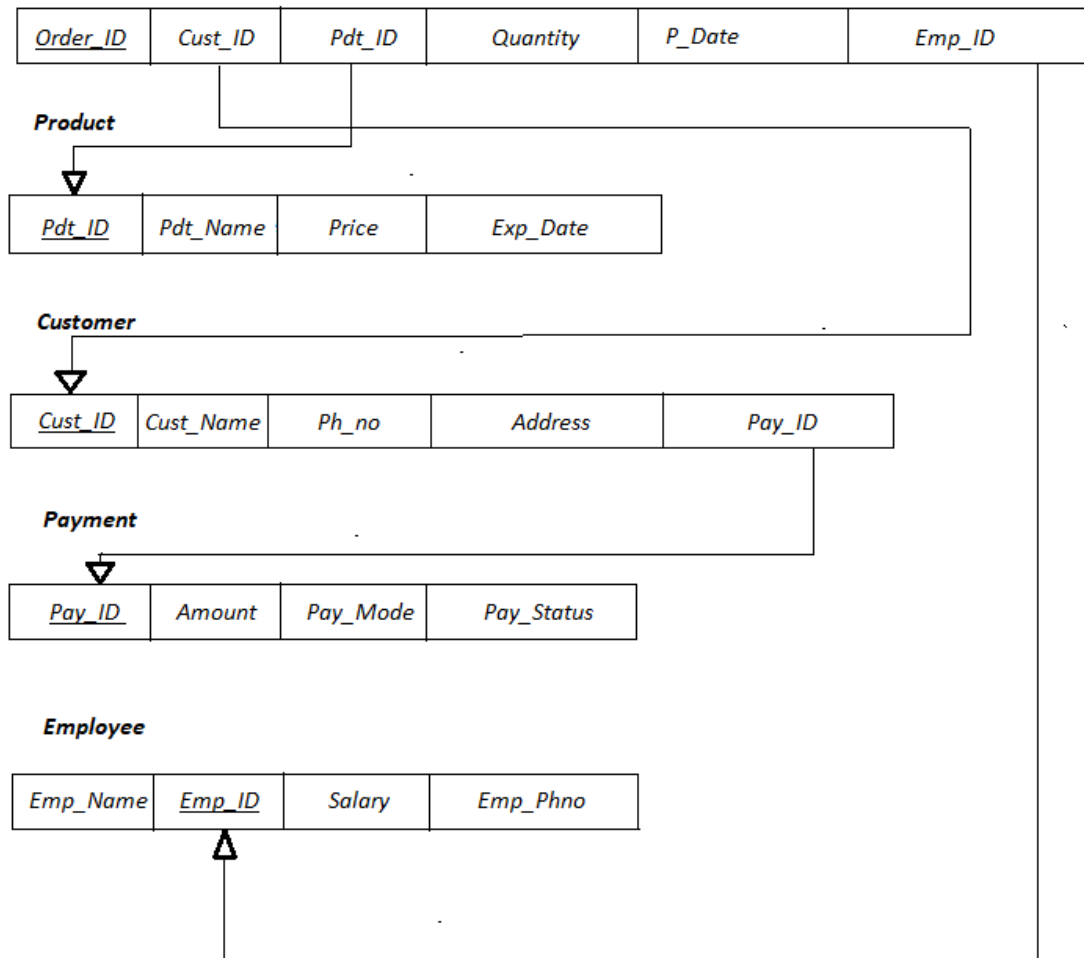
| <u>Cust_ID</u> | Cust_Name | Ph_no | Address | Pay_ID |
|----------------|-----------|-------|---------|--------|
|----------------|-----------|-------|---------|--------|

Payment

| <u>Pay_ID</u> | Amount | Pay_Mode | Pay_Status |
|---------------|--------|----------|------------|
|---------------|--------|----------|------------|

Employee

| Emp_Name | <u>Emp_ID</u> | Salary | Emp_Phno |
|----------|---------------|--------|----------|
|----------|---------------|--------|----------|

**Fig 3.3 (a)**

3.4 Entity-Relation Diagram

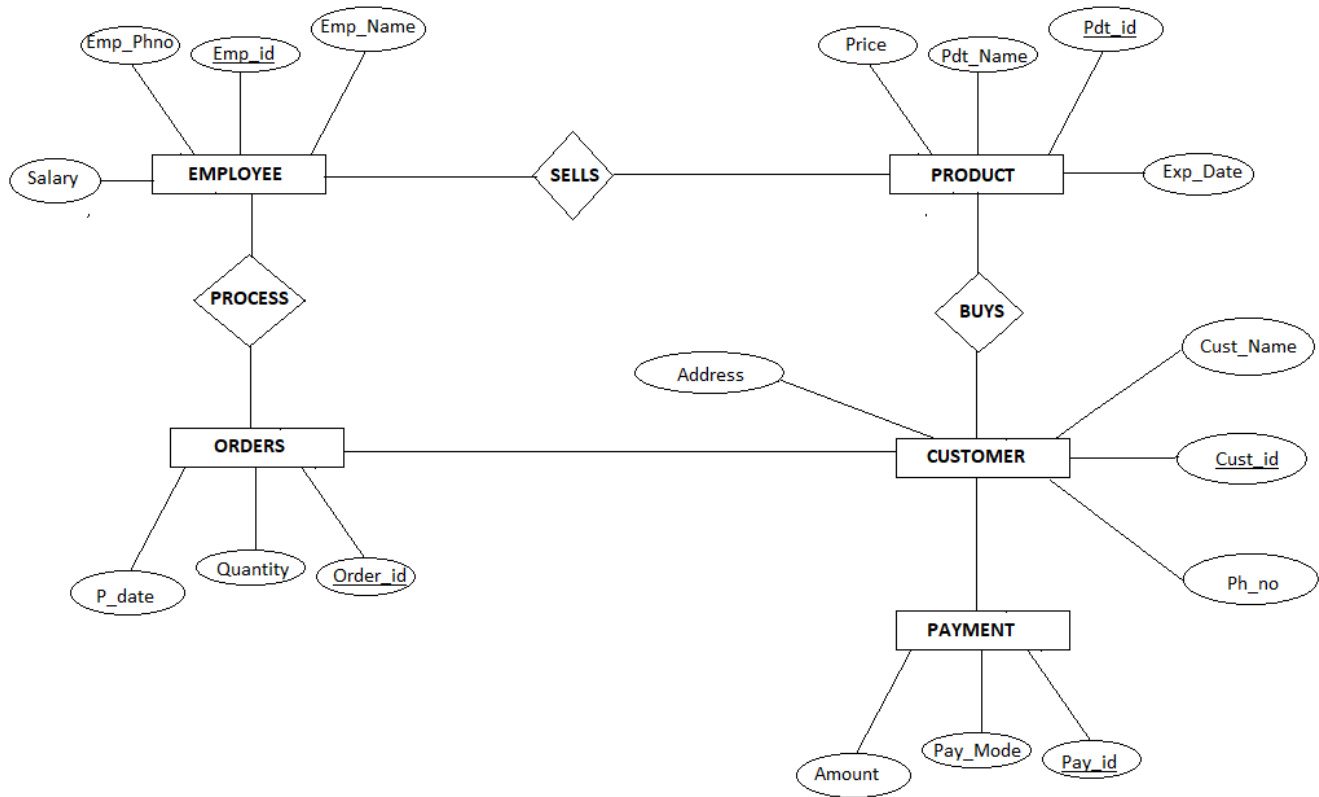


Fig 3.3 (b)

3.5 Table Description

3.5.1 Orders Table

Orders table is used to store database and fetch the information of sold products. Order table Consist of 6 tuples, namely Order_ID, Cust_ID, Product_ID, Quantity, Purchase_Date and Emp_ID.

The description of Orders Table is as follows:

Order_ID: Order Id(Integer) is a primary key which helps to uniquely identify each order.

Cust_ID: Customer ID (Integer) is a foreign key which helps to get Customer related information from Customer table.

Product_ID: Product ID (Integer) is also a foreign key which helps to get product information like Name, Price and Expiry date.

Quantity: Quantity (Integer) is used to store the amount product a customer has purchased.

Purchase_Date: Purchase_Date (date) is used to store the date of purchase of a particular product.

Employee_ID: Employee ID (Integer) is a foreign key and is used get/store information related Employee who has sold the product to the customer.

| Name | Data Type | Key (if exists) |
|---------------|-------------|------------------------|
| Order_ID | int | Primary Key |
| Cust_ID | int | Foreign Key (Customer) |
| Product_ID | int | Foreign Key (Product) |
| Quantity | int | - |
| Purchase_Date | date | - |
| Emp_ID | varchar(20) | Foreign Key (Employee) |

3.5.2 Product Table

Product table is important for a Supermarket or a provisional store to store the information related to the products like Product ID, Name, Price of the product and expiry date of the product. Hence, Product table has 4 tuples namely Product_ID (Primary key, integer), Product_Name (Char), Price and Expiry_Date. The description of Product Table is as follows:

Product_ID: Product_ID (int) is the primary key for the Product table and is used to uniquely identify each product.

Product_Name: Product Name (varchar(10)) is used to store the name of the product.

Price: Price (int) is used to store the price of the product.

Expiry_Date: Expiry_Date (date) is used to store the expiry date of the product.

| Name | Data Type | Key (if exists) |
|--------------|-------------|-----------------|
| Product_ID | int | Primary Key |
| Product_Name | varchar(10) | - |
| Price | int | - |
| Expiry_Date | date | - |

3.5.3 Customer Table

Customer table stores the information related to customer which include Customer Name, Customer ID, Phone Number, Address and Payment ID. Customer table consist of 5 tuples namely Cust_ID(Primary key,Integer), Customer_Name (Char), Phone_no (Integer), Address and Payment_Id (Foreign key, Integer). Customer Id is primary key is used to get customer information like name, phone number, address, and payment id.

The description of Customer Table is as follows:

Cust_ID: Customer ID (int) is the primary key for the Customer table and is used to uniquely identify each customer visiting the store.

Customer_Name: Customer Name (varchar(20)) is used to store the name of the customer.

Phone_No: Phone No (numeric(10,0)) is used to store the phone number of the customer.

Address: Address (varchar(50)) is used to store the address of the customer.

Payment_ID: Payment ID (int) is a foreign key and is used to refer to the transaction/payment made by the customer.

| Name | Data Type | Key (if exists) |
|---------------|--------------------|------------------------|
| Cust_ID | int | Primary Key |
| Customer_Name | <u>varchar(20)</u> | - |
| Phone_No | numeric(10,0) | - |
| Address | <u>varchar(50)</u> | - |
| Payment_ID | int | Foreign Key (Payments) |

3.5.4 Payment Table

Storing payment related data is very important for shops as well customer. Here we store payment related data in Payment table. The table consist of 4 tuples namely Payment_ID, Total_Amount, Payment_Mode and Payment_Status.

The description of Payment Table is as follows:

Payment_ID: Payment_ID (integer) is assigned to each transaction and it serves as primary key for the payment table which helps to get the payment related information.

Total_Amount: Total_amount (int) is an integer in which the total bill amount of a particular customer is stored.

Payment_Mode: Payment_Mode (varchar(10)) is use to store or get information related to the mode of transaction which can include Debit/Credit card, UPI or Cash.

Payment_Status: Payment_Status (varchar(10)) is used to store/get the Status (Successful/Failure) of payment.

| Name | Data Type | Key (if exists) |
|----------------|-------------|-----------------|
| Payment_ID | int | Primary Key |
| Total_Amount | int | - |
| Payment_Mode | varchar(10) | - |
| Payment_Status | varchar(10) | - |

3.5.5 Employee Table

Employee Table is helpful for owner of shop/Supermarket important. Employee table consists of 4 tuples namely Emp_ID(Primary key), Emp_Name, Salary and Emp_Phno.

The description of Employee Table is as follows:

Emp_ID: Emp_Id (varchar(20)) serves as the primary key for the Employee table and is used to uniquely identify each employee working at the store.

Emp_Name: Emp_Name (varchar(20)) is used to store the employees' names.

Salary: Salary (int) is use to store the salary of a particular customer.

Emp_Phno: Emp_Phno (numeric(10,0)) is used to store the contact details of the employee.

| Name | Data Type | Key (if exists) |
|----------|---------------|-----------------|
| Emp_ID | varchar(20) | Primary Key |
| Emp_Name | varchar(20) | - |
| Salary | int | - |
| Emp_Phno | Numeric(10,0) | - |

3.5.6 Authentication Table

Authentication Table is used to store the user name and the password that can be used to access the application. This Authentication Table consists of 2 tuples namely Property and Value. Property is used for Username and Value is for password.

| Name | Data Type | Key (if exists) |
|----------|-------------|-----------------|
| Property | varchar(15) | Primary Key |
| Value | varchar(15) | - |

3.6 Queries:

3.6.1 Order Table Queries

1. Query to view the table

```
SELECT o.Order_ID, o.Cust_ID, c.Customer_Name, o.Product_ID, p.Product_Name,  
o.Quantity, o.Purchase_Date, o.Emp_ID from Orders o, Customer c, Product p  
WHERE o.Cust_ID = c.Cust_ID and o.Product_ID = p.Product_ID
```

2. Query to add new row in the table

```
INSERT into Orders(Order_ID, Cust_ID, Product_ID, Quantity, Purchase_Date, Emp_ID)  
values(@Order_ID, @Cust_ID, @Product_ID, @Quantity, @Purchase_Date, @Emp_ID)
```

3. Query to delete an existing row in the table

```
DELETE from Orders where Order_ID = @Order_ID
```

4. Query to search for a row based on primary key in the table

```
SELECT o.Order_ID, o.Cust_ID, c.Customer_Name, o.Product_ID, p.Product_Name,  
o.Quantity, o.Purchase_Date, o.Emp_ID from Orders o, Customer c, Product p  
WHERE o.Cust_ID = c.Cust_ID and o.Product_ID = p.Product_ID and Order_ID = @Order_ID
```

3.6.2 Customer Table Queries

1. Query to view the table

```
SELECT * from Customer
```

2. Query to add new row in the table

```
INSERT into Customer(Cust_ID, Customer_Name, Phone_No, Address, Payment_ID)  
values(@Cust_ID, @Customer_Name, @Phone_No, @Address, @Payment_ID)
```

3. Query to delete an existing row in the table

```
DELETE from Customer where Cust_ID = @Cust_ID
```

4. Query to search for a row based on primary key in the table

```
SELECT * from Customer where Cust_ID = @Cust_ID
```

5. Query to generate bill

```
SELECT o.Order_ID, o.Product_ID, p.Product_Name, o.Quantity, p.Price from Orders  
o,Product p  
WHERE o.Cust_ID = @Cust_ID and o.Purchase_Date = @Purchase_Date and o.Product_ID =  
p.Product_ID
```

3.6.3 Product Table Queries

1. Query to view the table

```
SELECT * from Product
```

2. Query to add new row in the table

```
INSERT into Product(Product_ID, Product_Name, Price, Expiry_Date) values(@Product_ID,  
@Product_Name, @Price, @Expiry_Date)
```

3. Query to delete an existing row in the table

```
DELETE from Product where Product_ID = @Product_ID
```

4. Query to search for a row based on primary key in the table

```
SELECT * from Product where Product_ID = @Product_ID
```

3.6.4 Payment Table Queries

1. Query to view the table

```
SELECT * from Payments
```

2. Query to add new row in the table

```
INSERT into Payments(Payment_ID, Total_Amount, Payment_Mode, Payment_Status)  
values(@Payment_ID, @Total_Amount, @Payment_Mode, @Payment_Status)
```

3. Query to delete an existing row in the table

```
DELETE from Payments where Payment_ID = @Payment_ID
```

4. Query to search for a row based on primary key in the table

```
SELECT * from Payments where Payment_ID = @Payment_ID
```

3.6.5 Employee Table Queries

1. Query to view the table

```
SELECT * from Employee
```

2. Query to add new row in the table

```
INSERT into Employee(Emp_ID, Emp_Name, Salary, Emp_Phno) values(@Emp_ID,  
@Emp_Name, @Salary, @Emp_Phno)
```

3. Query to delete an existing row in the table

```
DELETE from Employee where Emp_ID = @Emp_ID
```

4. Query to search for a row based on primary key in the table

```
SELECT * from Employee where Emp_ID = @Emp_ID
```

CHAPTER - 4

Case Study

4.1 Grocery Store

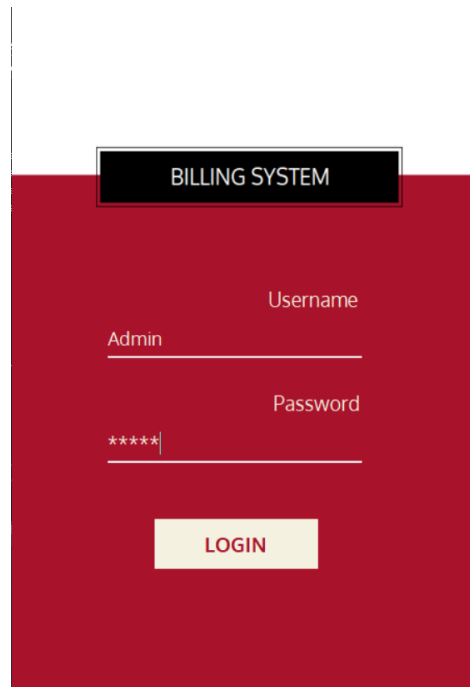
We referred to Grocery Store management from the internet to help us out in designing schema for our application.

A grocery store is a retail store that primarily sells food. A grocer is a bulk seller of food. Grocery stores often offer non-perishable food, with some also having fresh produce, butchers, delis, and bakeries. Large grocery stores that stock significant amounts of non-food products, such as clothing and household items, are called supermarkets. Some large supermarkets also include a pharmacy and an electronics section, the latter selling DVDs, headphones, digital alarm clocks, and similar items. Grocery stores operate in many different styles ranging from rural family-owned operations, such as IGAs, boutique chains, such as Whole Foods Market and Trader Joes to larger supermarket chain stores. In some places, food cooperatives or "co-op" markets, owned by their own shoppers, have been popular. However, there has recently been a trend towards larger stores serving larger geographic areas.

This system gave us a brief idea as of how to develop the tables for our project and also helped us to identify the foreign key constraints for our project. We designed the schema diagram as well as the Entity-Relationship diagram with the help of the Grocery store Database system.

CHAPTER - 5

Results



The authentication window features a dark red background. At the top, a black rectangular box contains the text "BILLING SYSTEM" in white. Below this, the word "Username" is displayed in a light gray font, followed by a text input field containing the text "Admin". Underneath, the word "Password" is displayed in a light gray font, followed by a text input field containing six asterisks "*****". At the bottom center, there is a yellow rectangular button with the word "LOGIN" in red capital letters.

Fig.5(a) Authentication Window

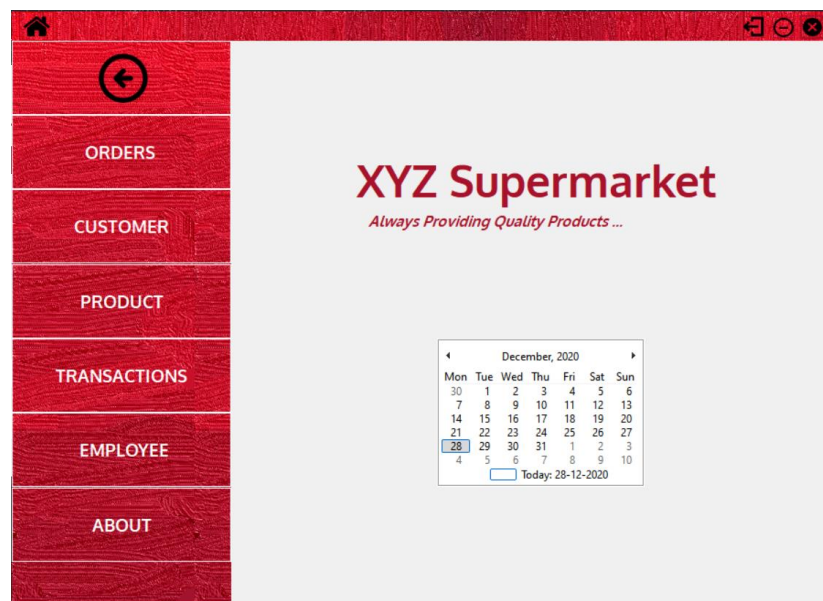


Fig.5(b) Home Page

Order ID : _____

Q Search

| | Order_ID | Cust_ID | Customer_N | Product_ID | Product_Ne | Quantity | Purchase_C | Emp_ID |
|---|----------|---------|------------|------------|--------------|----------|------------|--------|
| ▶ | 100001 | 1001 | Ramesh | 1 | Coconut ... | 3 | 12-10-2020 | 101 |
| | 100002 | 1001 | Ramesh | 4 | Sunpure ... | 3 | 12-10-2020 | 101 |
| | 100003 | 1001 | Ramesh | 2 | Milk (1 Ltr) | 1 | 12-10-2020 | 101 |
| | 100004 | 1002 | Arun | 3 | Rice (25 ... | 1 | 15-10-2020 | 102 |
| | 100005 | 1003 | Abdul | 6 | Hair Oil | 1 | 21-10-2020 | 103 |

+ Add New Delete

Clear

Order ID : _____ Employee ID : _____

Customer ID : _____ Product ID : _____

Quantity : _____ Purchase Date : _____

Fig.5(c) Orders

Customer ID : _____

Q Search

| | Cust_ID | Customer_Name | Phone_No | Address | Payment_ID |
|---|---------|---------------|------------|--------------|------------|
| ▶ | 1001 | Ramesh | 8754236585 | Yeshwantpur | 9810 |
| | 1002 | Arun | 9568547465 | Hebbal | 9820 |
| | 1003 | Abdul | 8451745263 | Jayanagar | 9830 |
| | 1004 | Nayana | 8743521486 | Banashankari | 9840 |
| | 1005 | Joseph | 5478659775 | Basavangudi | 9850 |

+ Add New Delete

Clear

Customer ID : _____

Customer Name : _____ Phone : _____

Address : _____ Payment ID : _____

Generate Bill

Fig.5(d) Customer Table

Product ID : _____

Q Search

| Product_ID | Product_Name | Price | Expiry_Date |
|------------|--------------------|-------|-------------|
| 1 | Coconut Cookies | 20 | 20-01-2021 |
| 2 | Milk (1 Ltr) | 40 | 31-12-2020 |
| 3 | Rice (25 KG) | 1000 | |
| 4 | Sunpure Oil (1Ltr) | 100 | 10-01-2021 |
| 5 | Toor Dal | 90 | |

+ Add New Delete

Clear

Product ID : _____ Product Name : _____

Price : _____ Expiry Date : _____

Fig.5(e) Product Table

Payment ID : _____

Q Search

| Payment_ID | Total_Amount | Payment_Mode | Payment_Status |
|------------|--------------|--------------|----------------|
| 9810 | 400 | Cash | Paid |
| 9820 | 1000 | Debit Card | Pending |
| 9830 | 1040 | Cash | Paid |
| 9840 | 230 | Cash | Paid |
| 9850 | 120 | Cash | Paid |

+ Add New Delete

Clear

Payment ID : _____ Total Amount : _____

Payment Mode : _____ Payment Status : _____

Fig.5(f) Payment Table

Employee ID : _____

Search

| | Emp_ID | Emp_Name | Salary | Emp_Phno |
|---|--------|----------|--------|------------|
| ▶ | 101 | Sanjay | 35000 | 7485632514 |
| | 102 | Pradeep | 35000 | 8456315472 |
| | 103 | Akshay | 25000 | 5487412684 |
| * | | | | |

+ Add New Delete

Clear

Employee ID : _____ Employee Name : _____

Salary : _____ Phone : _____

Fig.5(g) Employee Table

Bill

Customer ID : 1001

Purchase Date : 2020-10-12

Generate Bill

| | Order_ID | Product_ID | Product_Na | Quantity | Price |
|---|----------|------------|--------------|----------|-------|
| ▶ | 100001 | 1 | Coconut ... | 3 | 20 |
| | 100002 | 4 | Sunpure ... | 3 | 100 |
| | 100003 | 2 | Milk (1 Ltr) | 1 | 40 |
| * | | | | | |

Save

Fig.5(h) Bill Generator

CHAPTER - 6

Conclusion

To sum up, Billing system is a standalone application which can be used to store and process the data such as Customer details, billing details, employee details, etc and properly organize the data so that it can be retrieved and managed when required.

The project gave us a good idea on developing a full-fledged application which can satisfy the end-user requirements. The developed application is very flexible and versatile. This software has a user-friendly interface that enables the user to use the application without any inconvenience.

The project has been successfully implemented. The constraints are met and overcome successfully. Validation checks have been induced regularly to reduce errors. Provisions have been made to upgrade the software in future.

References

1. Christian Trahan, Brandon Jones Grocery Store Project, CMPS 342
2. Tapan Desai, Grocery Stop, Database Management Tool, IST 659
3. <https://visualstudio.microsoft.com/vs/support/>
4. <https://sqlite.org/forum/>
5. <https://stackoverflow.com/>

VISION: Emerge as center of learning in the field of information science & engineering with technical competency to serve the society.

MISSION: To provide excellent learning environment through balanced curriculum, best teaching methods, innovation, mentoring and industry institute interaction.

PEO1: Successful professional career in Information Science & Technology

PEO2: Pursue higher studies & research for advancement of knowledge in IT industry

PEO3: Exhibit professionalism and team work with social concern