Supermarket Database Management Project

A Project Report Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Puregella Manas Bharadwaj, Vansh Jogani, Vedansh Makharia

220953372, 220953126, 220953050

Under the guidance of

Dr. Sumith N (Lab Faculty 1)
Associate Professor
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

Ms. Chetana Pujari (Lab faculty 2)
Assistant Professor - Senior Scale
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

April 2024

ABSTRACT

The Supermarket Database Management System (SDMS) is a comprehensive solution designed to efficiently manage the inventory, sales, and customer data of a supermarket. SDMS offers a centralized platform with various tables including Member, Bought, Manufacturer, Stock, Product, Discount, Bill, Billing, Guest Bill, Member Bill, Guest, Employee, and Stock.

Key features of SDMS include the ability to store and manage information about members, purchases, manufacturers, products, discounts, bills, and stock levels. The system enables seamless tracking of inventory, sales, and customer data, facilitating effective decision-making and resource allocation.

SDMS provides a user-friendly interface for supermarket staff to manage inventory, process sales, generate bills, and track customer transactions. The system enhances operational efficiency, improves customer service, and ensures accurate and timely reporting.

Overall, SDMS is an essential tool for supermarket management, offering a scalable and adaptable solution to meet the needs of supermarkets of all sizes.

ACM Taxonomy Terms: -

[Software]: Supermarket Management System; Grocery Management System; Billing Management System

[Computing Milieux]: Supermarket Software; Supermarket Technology

[Information systems applications]: Enterprise Information Systems; Goods Management Systems

Sustainable Development Goals (SDGs)

The SDMS aims to participate in attaining the following SDGs:

SDG Goal 9: Industry, Innovation, and Infrastructure
Foster innovation in supermarket management through the implementation of advanced database management systems.
Enhance infrastructure for supermarkets to improve efficiency and customer service.

SDG Goal 8: Decent Work and Economic Growth
Improve productivity and efficiency in supermarkets, leading to economic growth.
Create employment opportunities through the implementation and maintenance of the database management system.

Table of Contents

List of Tables

List of Figures

Abbreviations

1. SDMS – Supermarket Database Management System
2. SDG- Sustainable Development Goals
3. ACM – Association for Computing Machinery
4. UI – User Interface
5. NF – Normal form
6. 1NF - 1st Normal Form
7. 2NF – 2nd Normal Form
8. 3NF – 3rd Normal Form
9. BCNF – Boyce Codd Normal Form

Chapter 1
Introduction

A database serves as a meticulously structured repository of data, leveraging database systems like Microsoft SQL Server for efficient management and accessibility. Relational database management systems (RDBMS) such as SQL Server are frequently employed for their adept handling of intricate queries, maintenance of ACID (Atomicity, Consistency, Isolation, Durability) properties, and facilitation of data relationships via foreign keys.

Microsoft SQL Server offers vital functionalities including concurrency control, backup and recovery mechanisms, and streamlined query processing to ensure optimal data management. Its robust features and scalability make it an ideal choice for businesses of all sizes seeking reliable and high-performance database solutions.

1.1 Three tier Architecture

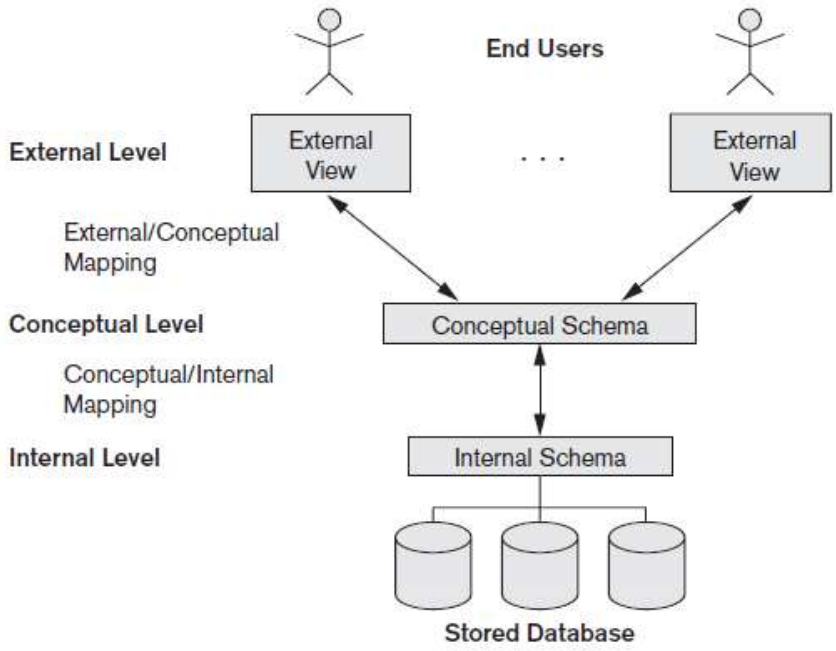


Figure 1.1: Three Tier Architecture

1. External Level (User view) :-

This is the highest level of data abstraction. This is where everyday users interact with the database, accessing the information they need without having to worry about how it's all set up underneath. It's like using an app on your phone or a website on your computer—you just see the interface and interact with it without knowing all the coding and technical details happening behind the scenes.

2. Conceptual Level (Logical view) :-

This level is like the map of the database. It shows how all the different pieces of data are organized and connected to each other. Instead of seeing individual files or records, you get a big-picture view of how everything fits together. It helps database managers understand the structure and relationships within the database without needing to know the specifics of how it's physically stored.

3. Internal Level (Physical view) :-

Here, we dive into the nuts and bolts of how the data is actually stored on the hardware, like hard drives or solid-state drives. It's all about the technical details of how files are stored, accessed, and managed on these devices. Think of it like peeking under the hood of your computer to see how the files are organized on the hard drive—indexes, data access paths, and other technical details that ensure the data is stored efficiently and can be retrieved quickly when needed.

Chapter 2
Background

2.1 Database Design:

Data Consistency and Integrity: Ensuring that data remains consistent and adheres to integrity constraints is crucial in database design. This involves maintaining accurate and reliable information.
Normalization: Apply normalization forms to identify anomalies and eliminate redundancy.

2.2 Function-Oriented Design:

Decompose the system into interacting units, each with a clearly defined function. Techniques like data modelling and normalization are used to visualize data and prevent redundancy.

2.3 Frontend Functionality:

A good web design includes organized text, color contrast, stylish pages, appropriate fonts, consistency, and graphics ensuring proper functionality and accessibility [4]. The web design should be user-friendly, responsive, and visually appealing to effectively communicate text, messages, and ideas therefore giving the optimum experience to the user.

Chapter 3

Problem Statement and Objectives

This chapter highlights the objectives that the SDMS aims to achieve.

This chapter highlights the objectives that the Supermarket Database Management System (SDMS) aims to achieve:

1. Efficient and User-friendly Interface:
- SDMS aims for a clean, intuitive, and easy-to-navigate user interface, ensuring clear labeling for enhanced usability.

2. Registration and Authentication of Users:
- The system accommodates various user classes including administrators, employees, and customers. Employees are authenticated through username, password.

3. Registration of Products, Manufacturers, and Members:
- SDMS facilitates the registration of products, manufacturers, and members.
- Products are registered with details such as product name, category, price, and manufacturer information.

4. Timely Updation of Inventory and Sales Data:
- SDMS ensures that inventory and sales data are updated in real-time to reflect changes accurately.
- Stock levels are updated automatically upon purchase or sale of products.

5. Enhancing Sales and Customer Satisfaction:
- The system aims to enhance sales by providing real-time data analytics and insights.
- SDMS facilitates discounts to improve customer satisfaction.

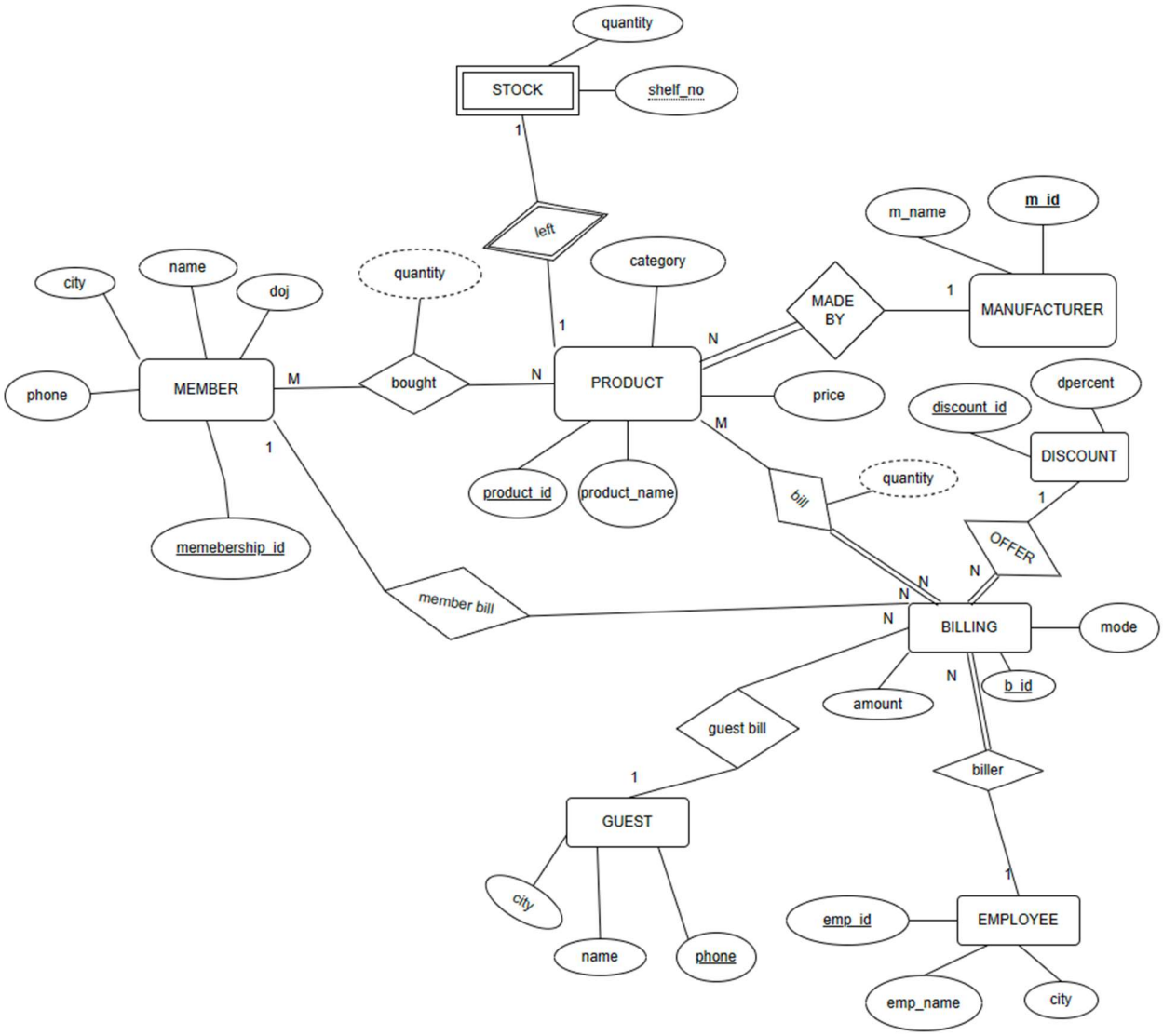
6. Tracking Financial Performance:
- SDMS generates financial reports to track income, expenditure, and profitability.
- Sales reports provide detailed information on revenue generated from various product categories, helping in strategic decision-making.

By addressing these objectives, SDMS aims to optimize inventory management, streamline sales processes, and improve overall efficiency and customer satisfaction in the supermarket.

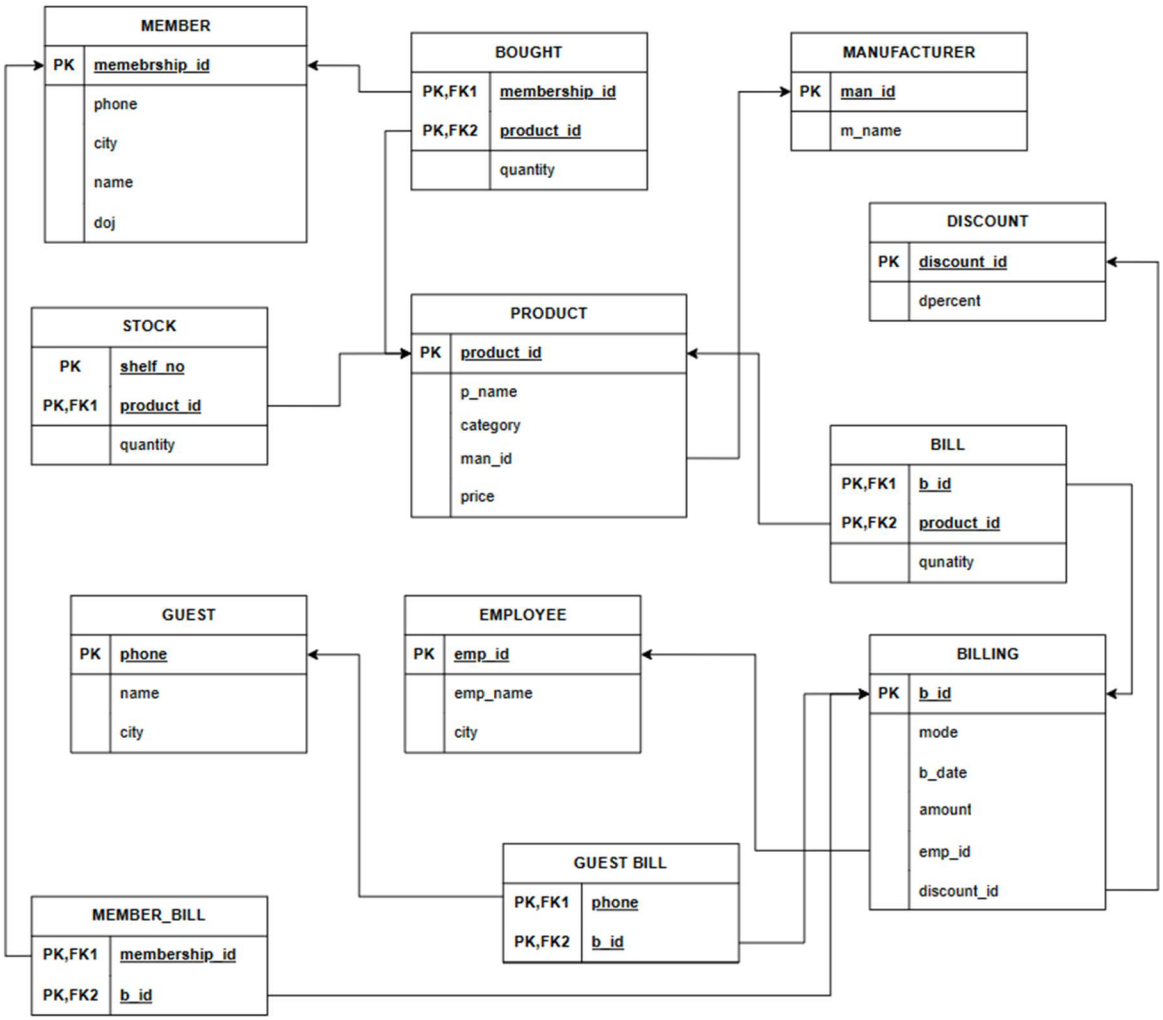
Chapter 4

Data Design

4.1 Entity Relationship Model



4.2 Schema Diagram (Before Normalisation)



4.3 Normalisation

FD = Functional Dependencies

CK = Candidate key

1. **Member** (membership_id, name, phone, city, doj)

   FD:
   FD1: membership_id → name, phone, city, doj

   Since there is only 1 FD, it is the canonical cover.

   1 NF is satisfied as no multivalued attributes
   CK: membership_id

   2NF is satisfied as no partial dependencies

   3NF is satisfied as no transitive dependency (only 1FD)

   BCNF is satisfied as CK is on alpha side

2. **Manufacturer** (man_id, m_name)

   FD:
   FD1: man_id → m_name
   Since there is only 1 FD, it is the canonical cover

   1NF is satisfied as no multivalued attributes
   CK: man_id

   2NF satisfied as no partial dependencies.

   3NF is satisfied since there is only 1 FD there are no transitive dependency.

   BCNF is satisfied as CK is on alpha side.

3. **Employee** (emp_id, emp_name, city)

   FD
   FD1: emp_id → emp_name, city
   Since there is only one FD it is the canonical cover.
   1NF is satisfied.

   CK: emp_id

   2NF is satisfied as no partial dependencies

   3NF is satisfied as no transitive dependencies

   BCNF as CK is on alpha side

4. **Guest** (phone, name, city)

   FD
   FD1: phone → name, city
   CK: phone

   1NF is satisfied as no multivalued attributes

   2NF is satisfied as no partial dependencies

   3NF is satisfied as no transitive dependencies

   BCNF (phone) CK is on alpha side.

5. **Discount** (discount_id, dpercent)

   FD
   FD1 : discount_id → dpercent
   CK: discount_id

   1NF is satisfied as no multivalued attributes.

   2NF is satisfied as no partial dependencies.

   3NF is satisfied as no transitive dependencies.

BCNF is satisfied as CK is on alpha side.

6. **Billing** (b_id, b_date, amount, emp_id, discount_id, mode)

   FD:
   FD1 : b_id → b_date
   FD2: b_id → amount
   FD3: b_id → emp_id, discount_id, mode

   b_id + → b_id, b_date, amount, emp_id, discount_id, mode

   CK: b_id

   Using union
   Fc: b_id → b_date, amount, emp_id, discount_id, mode
   No extraneous attributes

   1NF is satisfied as no multivalued attributes.

   2NF is satisfied as no partial dependencies.

   3NF is satisfied as no transitive dependencies.

   BCNF is satisfied as CK is on alpha side.

7. **Product** (product_id, p_name, price, category, man_id)

   FD:
   FD1: product_id → p_name, price, category
   FD2: product_id → man_id

   product_id+ → product_id, p_name, price, category, man_id
   CK: product_id

   Using union
   Fc : product_id → p_name, price, category, man_id
   No extraneous attributes

   1NF is satisfied as no multivalued attributes.

   2NF is satisfied as no partial dependencies.

3NF is satisfied as no transitive dependencies.

BCNF is satisfied as CK is on alpha side.

8. **Stock** (product_id, shelf_no, quantity)

   FD:
   FD1 : product_id, shelf_no → quantity
   Since only one FD it is the canonical cover.

   CK: product_id, shelf_no

   1NF is satisfied as no multivalued attributes.

   2NF is satisfied as no partial dependencies.

   3NF is satisfied as no transitive dependencies.

   BCNF is satisfied as CK is on alpha side.

9. **Bought** (membership_id, product_id, quantity)

   FD:
   FD1 : membership_id, product_id → quantity
   Only 1 FD, therefore it is the canonical cover.

   CK: membership_id

   1NF is satisfied as no multivalued attributes.

   2NF is satisfied as no partial dependencies.

   3NF is satisfied as no transitive dependencies.

   BCNF is satisfied as CK is on alpha side.

10. **Member_bill** (membership_id, b_id)

    FD: no FDs
    CK: membership_id, b_id

    1NF is satisfied.

    2NF is satisfied.

    3NF is satisfied.

    BCNF is satisfied.

    The candidate key will be in 3NF and BCNF as if is a trivial dependency.

11. **Guest_bill** (phone, b_id)

    FD: no FDs

    CK: phone, b_id

    1NF is satisfied.

    2NF is satisfied.

    3NF is satisfied.

    BCNF is satisfied.

    The CK will be in 3NF and BCNF.

12. **Bill** (b_id, product_id, quantity)
    FD:

    FD1 : b_id, product_id → quantity
    Only one FD therefore it is the canonical cover.

    CK: b_id, product_id

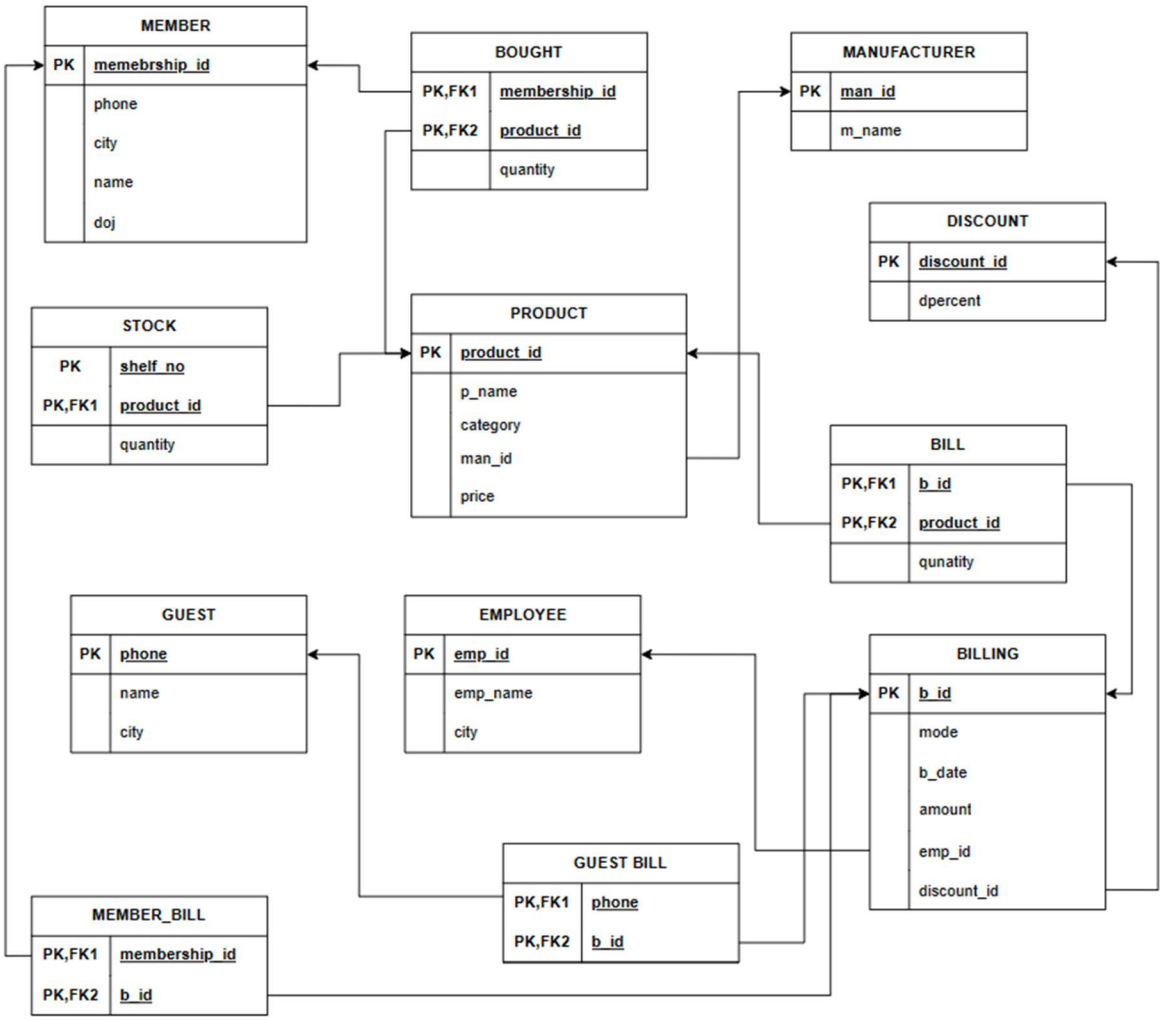
1NF is satisfied as no multivalued attributes.

2NF is satisfied as no partial dependencies.

3NF is satisfied as no transitive dependencies.

BCNF is satisfied as CK is on alpha side.

4.4 Schema Diagram (After Normalisation)



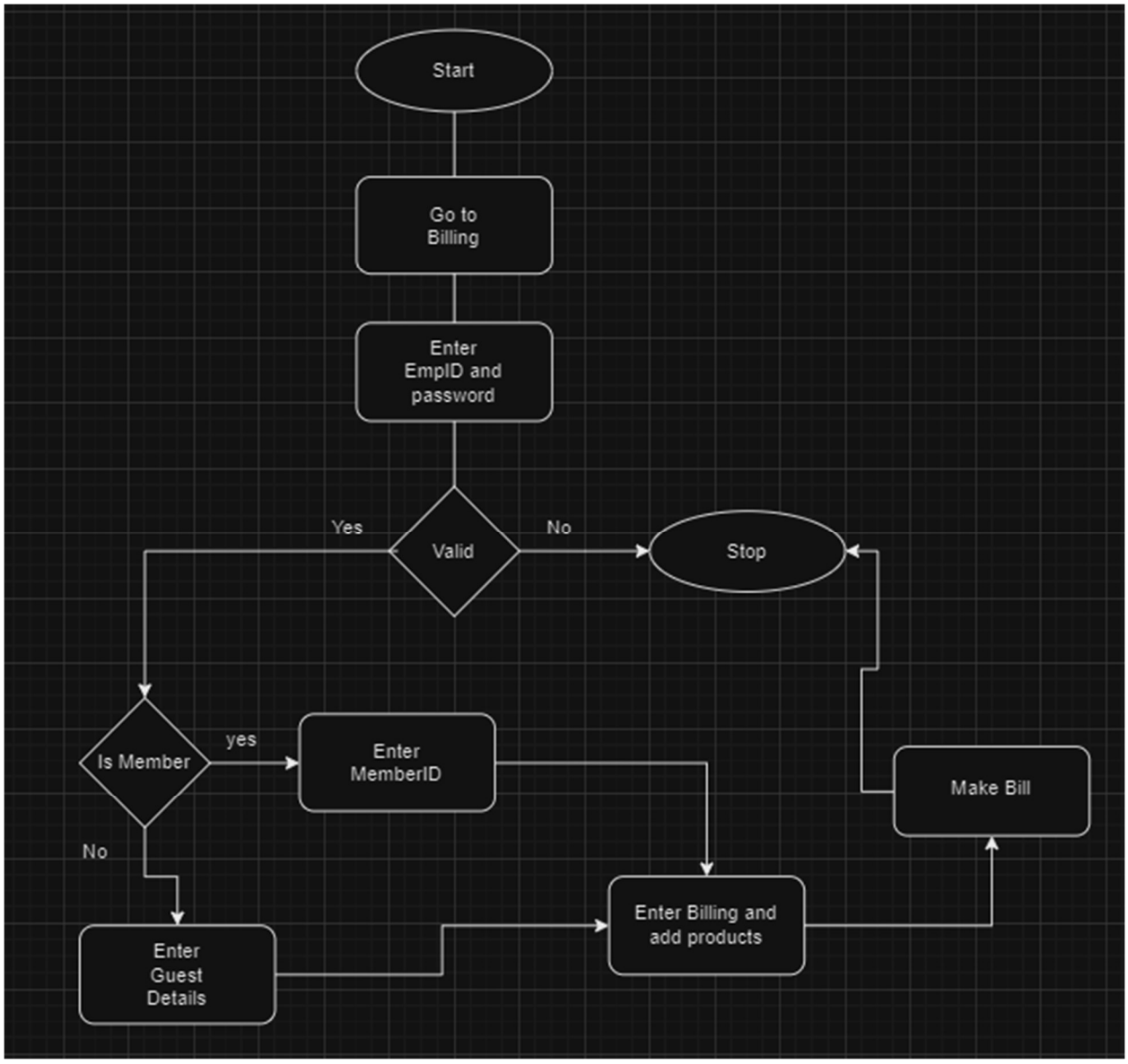
Chapter 5

Methodology

Feature Implementations:

1. Login Page

The Supermarket Database Management System (SDMS) login page provides access to authorized employees for billing purposes. Employees can log in using their valid employee ID and password, which are authenticated against the stored database.
The login page also includes registration facilities for employees.



2. Employee Dashboard

The employee dashboard provides member-information for the logged-in end user, and provides access to Member bill information retrieved via their respective memebrship-ID.

3. Inventory Management

Track inventory levels in real-time and Implements automatic stock updates upon purchase or sale of products.

4. Customer Management

Store customer information such as name, contact details, and purchase history.

5. Supplier Management

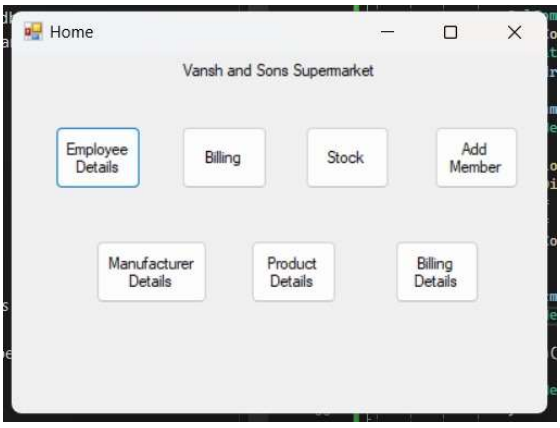
Store supplier information such as name, contact details, and product catalog.

Chapter 6

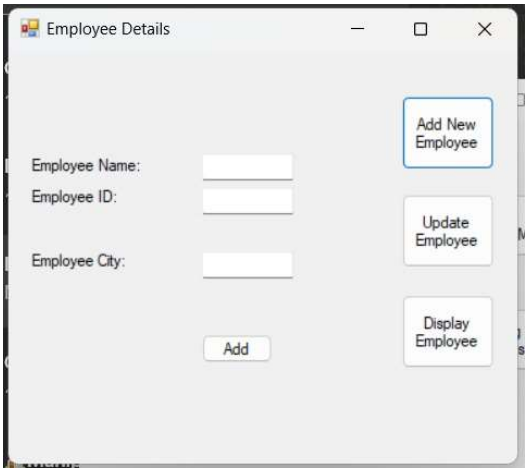
Results

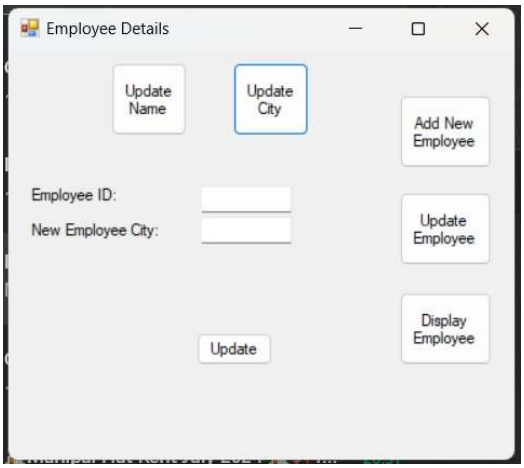
Given below are screenshots from the pages and functionalities implemented for Supermarket Database Management System:

Home:

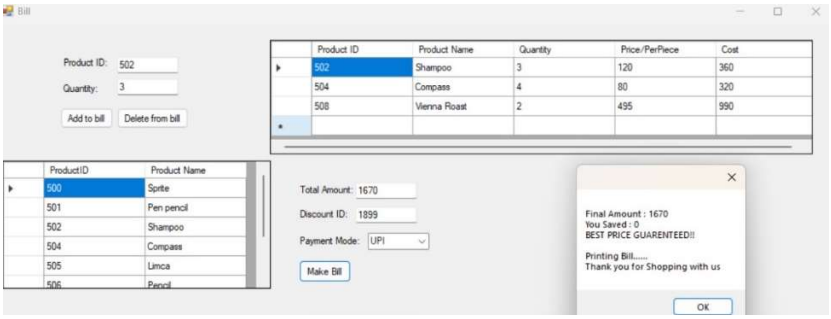


Employee Details:

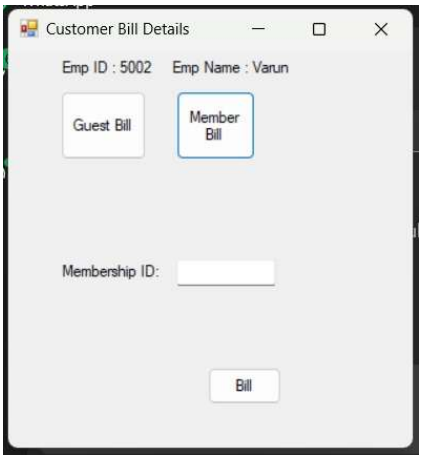




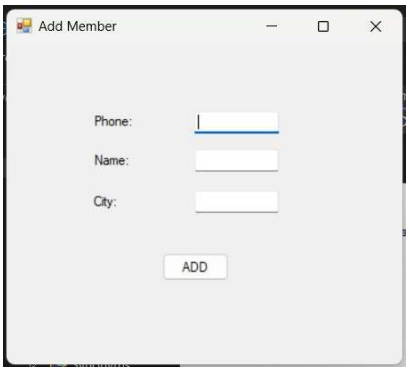
Bill:



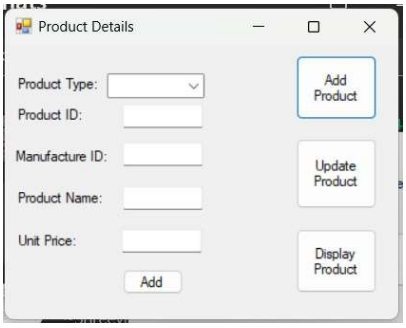
Billing details:



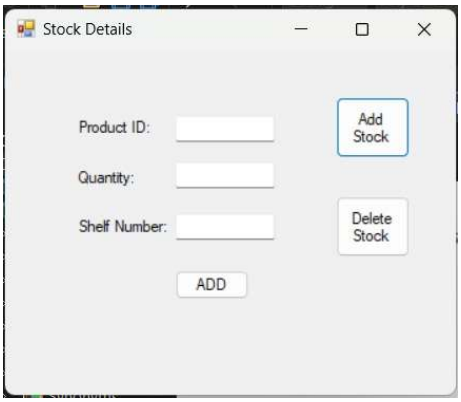
Add member details:



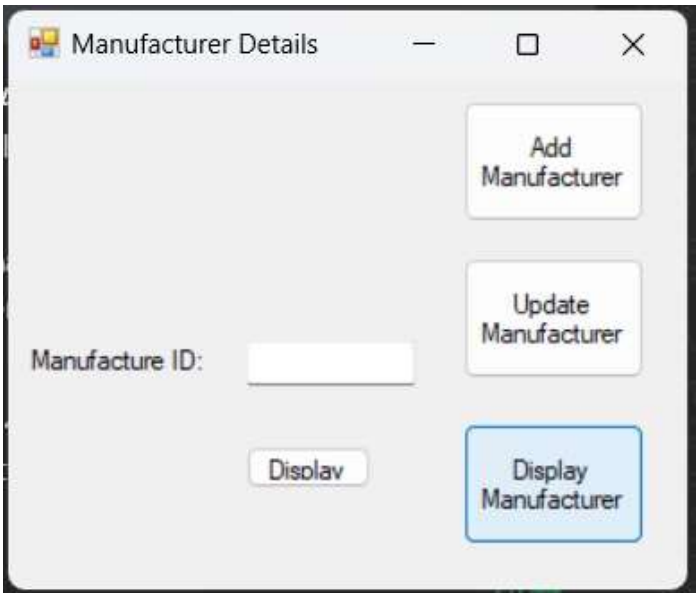
Product Details:



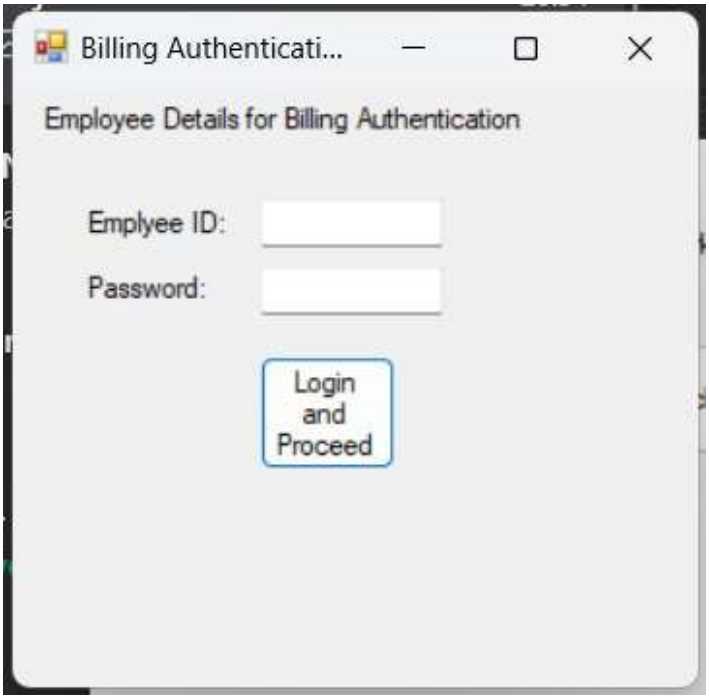
Stock details:



Manufacturer details:



Billing Authentication details:



Chapter 7

Conclusion and Future Work:

Conclusion:

In conclusion, the Supermarket Database Management System (SDMS) has significantly improved inventory management, streamlined sales processes, and enhanced data security.

Future Work:

Future enhancements include integrating online platforms, enhancing customer relationship management, implementing predictive. These developments will further improve efficiency, productivity, and customer satisfaction.

Chapter 8

References

Sure, here are the references in IEEE format:

Microsoft SQL Server Documentation:
Microsoft. (n.d.). SQL Server Documentation. [Online]. Available: https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15

Visual C# Documentation:
Microsoft. (n.d.). C# Documentation. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/