Anudip Foundation

MEDICAL STORE MANAGEMENT SYSTEM

Project supervisor

Miss. Priti L. Yadav

REPRESENTED BY

Mr. GAIKWAD ROHIT N.

Miss. KAMBLE SNEHA K.

OVERVIEW

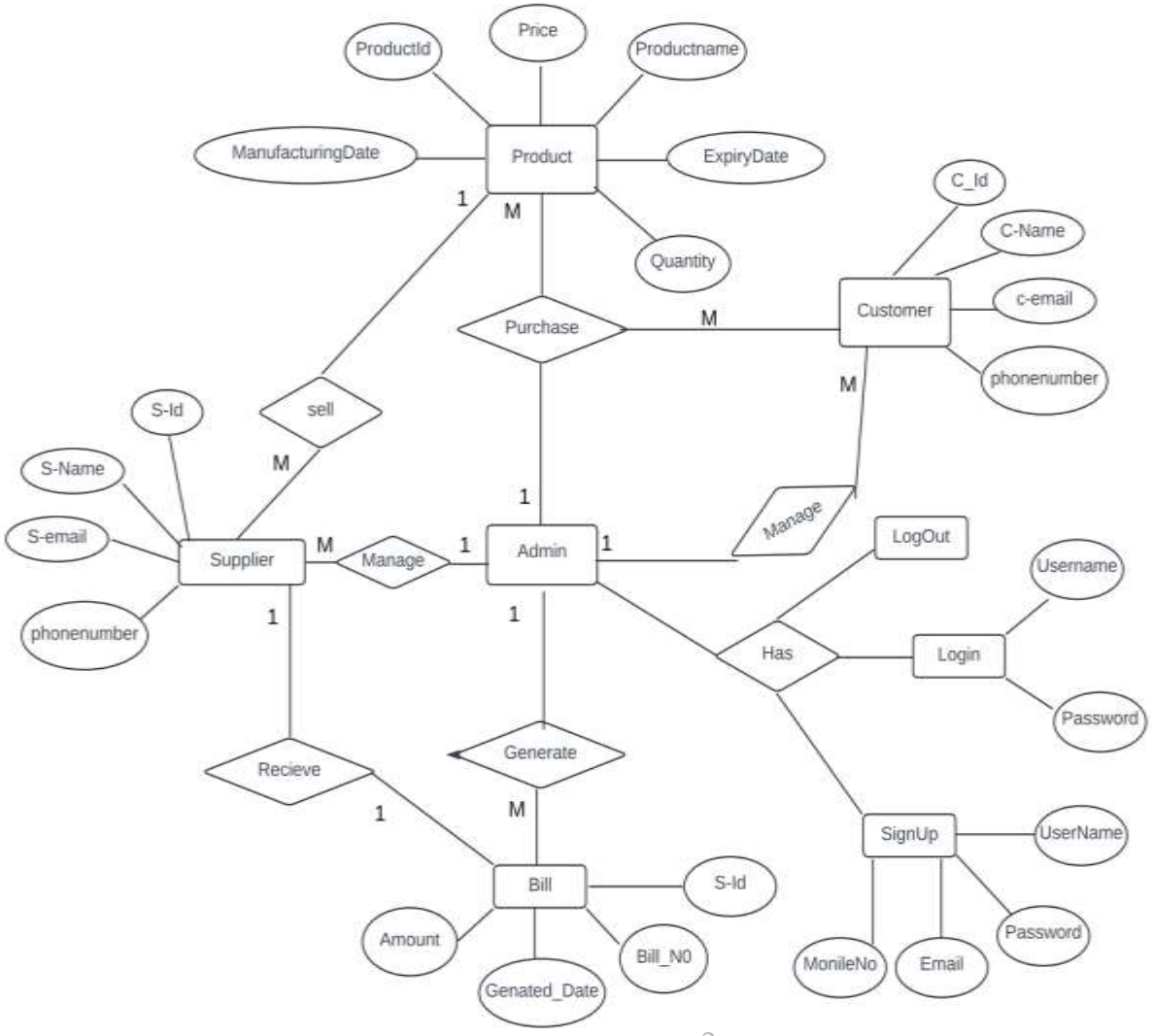
INTRODUCTION

- **In the modern world, efficient management of medical stores is crucial for providing quality healthcare services to the community.**
- **A Medical Store Management System implemented in Java offers a comprehensive solution for managing inventory, sales, and other operations of a medical store efficiently.**
- **This system aims to streamline the workflow of medical stores by automating various tasks, including inventory management, sales processing, customer management, and reporting.**
- **By leveraging Java's robust features and object-oriented programming principles, this system ensures reliability, scalability, and maintainability.**

KEY FEATURES

- **Inventory Management: The system allows easy management of inventory by providing functionalities to add new items, update quantities, search Item, track expiry dates, and remove expired items.**
- **Sales Processing: It facilitates smooth sales processing by enabling the creation of sales transactions, calculating total prices, updating inventory after sales, and generating invoices for customers.**
- **Customer Management: Optionally, the system can include features for managing customer information, such as adding new customers, updating contact details, and tracking purchase history.**
- **Supplier Management: Similarly, supplier management functionalities can be integrated to manage supplier information, track orders, and manage supplier relationships effectively.**
- **Reporting and Analysis: The system offers reporting and analysis capabilities to generate various reports, including inventory reports, sales reports, profit analysis, and trend analysis, aiding in informed decision-making.**
- **User-friendly Interface: A user-friendly graphical interface is provided to enhance user experience and facilitate easy navigation through different functionalities of the system.**
- **Security: Robust security measures are implemented to ensure data confidentiality, integrity, and availability.**
- **Database Integration: The system integrates with a relational database management system (e.g., MySQL, SQLite) to persistently store and retrieve data, ensuring data consistency and durability.**

E-R DIAGRAM


ProductId
Price
Productname
ManufacturingDate
Product
ExpiryDate
Quantity
1
M
C_Id
C-Name
c-email
Customer
phonenumber
Purchase
M
M
sell
S-Id
S-Name
S-email
Supplier
phonenumber
M
1
Manage
1
Admin
1
Manage
LogOut
Username
Has
Login
Password
1
1
Recieve
Generate
1
M
SignUp
UserName
Bill
S-Id
Amount
Bill_N0
Genated_Date
MonileNo
Email
Password

MODULES

- **Products: Represents an item available in the medical store.**
- **Attributes: ProductId, name, price, quantity, manufacturingDate, expiryDate.**
- **Supplier: Represents a supplier who provides products to the medical store.**
- **Attributes: supplierId, name, email, phone.**
- **Customer: Represents a customer who purchases items from the medical store.**
- **Attributes: customerId, name, email, phone.**
- **Sale: Represents a sale transaction made by a customer.**
- **Attributes: saleId, date, totalPrice.**
- **SaleItem : Represents an individual item sold in a sale transaction.**
- **Attributes: saleItemId, sale (reference to Sale), item (reference to Item), quantity.**

TECHNOLOGIES USED

- **Java: The core programming language for developing the application logic, business rules, and user interfaces.**
- **Hibernate: An ORM (Object-Relational Mapping) framework that simplifies data persistence by mapping Java objects to database tables and vice versa. Hibernate manages the database interactions and provides a higher level of abstraction over JDBC (Java Database Connectivity).**
- **Hibernate Annotations or XML Mapping: Hibernate can be configured using either annotations or XML mapping files to define the mapping between Java entities and database tables, specify relationships between entities, and configure caching, lazy loading, and other properties.**
- **Java Persistence API (JPA): JPA is a Java specification for managing relational data in applications using Java objects. Hibernate implements the JPA specification, allowing developers to use standard APIs for database interactions.**
- **Servlets and JSP (JavaServer Pages): For developing the web-based user interface, Servlets and JSP can be used along with HTML, CSS, and JavaScript. Servlets handle HTTP requests and responses, while JSP enables the creation of dynamic web pages by embedding Java code within HTML.**
- **Relational Database Management System (RDBMS): A database management system such as MySQL, PostgreSQL, Oracle, or SQL Server is used to store and manage the application data persistently. Hibernate abstracts the database interactions, allowing the system to work with different RDBMSs seamlessly.**