

Medical Store Management System

Submitted by

**Govind
Kulkarni
2317081
MCA**

**Aloysius Institute of Management and Information Technology (Aimit) St Aloysius
College (Autonomous)
Mangalore**

Under the guidance of

Mr. Rakesh Kumar

Assistant Professor

St. Aloysius institute of Management and IT
Beerli, Mangalore

Abdul Nasir

Owner of BabyShop

BabyShop
Bolwar, Puttur

Submitted to



**ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY (AIMIT)
ST ALOYSIUS COLLEGE (AUTONOMOUS)**

MANGALURU, KARNATAKA

2024



**ALOYSIUS INSTITUTE OF MANAGEMNT AND INFORMATION
TECHNOLOGY
ST ALOYSIUS COLLEGE (AUTONOMOUS)
MANGALORE, KARNATAKA**

CERTIFICATE

This is to certify that the project titled
Medical Store Management System

Submitted by

**Govind
Kulkarni
2317081**

MCA

**Aloysius Institute of Management and Information Technology (Aimit) St Aloysius
College (Autonomous)
Mangalore**

As a part of MCA 3rd/5TH SEM, is a bonafide record of the work carried out at

**BABY SHOP
Puttur**

**During the year
2024**

Dr Hemalatha N, Dean
PG Depts of Information Technology,
AIMIT, St. Aloysius College,
Mangalore-575 022.

Examiners

- 1.
- 2.

CERTIFICATE OF AUTHENTICATED WORK

This is to certify that the project report entitled Medical Store Management System submitted to Aloysius Institute of Management and Information Technology (AIMIT), St Aloysius College, Mangalore affiliated to Mangalore University in partial fulfilment of the requirement of MCA 3rd Sem is an original work carried out by Mr./Ms. Govind Kulkarni Register no. 2317081 under my guidance. The matter embodied in this project is authentic and is genuine work done by the student and has not been submitted whether to this University, or to any other University / Institute for the fulfilment of the requirement of any course of study.

Signature of the Student:

Date:

Name and Address of the Student:

Signature of the Guide:

Date:

Name, Designation and Address of the Guide:

INDEX

SR. NO	TABLE OF CONTENTS	PAGE NO
1.	INTRODUCTION 1.1 Background 1.2 Objectives 1.3 Purpose, Scope, and Applicability 1.3.1 Purpose 1.3.2 Scope 1.3.3 Applicability 1.4 Achievements 1.5 Organization of Report	
2.	SURVEY OF TECHNOLOGIES	
3.	REQUIREMENTS AND ANALYSIS 3.1 Problem Definition 3.2 Requirements Specification 3.3 Planning and Scheduling 3.4 Software and Hardware Requirements	
4.	SYSTEM DESIGN 4.1 Basic Modules 4.2 Data Design Project Structure 4.2.1 Schema Design 4.2.2 Data Integrity and Constraints 4.3 Procedural Design 4.3.1 Logic Diagrams 4.3.2 Data Structures 4.3.3 Algorithms Design 4.4 User interface design	
5.	IMPLEMENTATION AND TESTING 5.1 Introduction 5.2 Testing Objective 5.3 Unit Testing 5.4 Integration Testing	
6.	CONCLUSIONS 6.1 Conclusion 6.2 Limitations of the System 6.3 Future Scope of the Project	
	RERENCES	
	APPENDIX A	

CHAPTER 1: INTRODUCTION

1.1 Background

The Medical Store Management System aims to streamline the operations of a medical store by automating and integrating processes such as inventory management, billing, medicine procurement, and customer management. It addresses common challenges in medical stores, such as inventory tracking, stock levels, and efficient billing. By implementing this system, the medical store can provide faster, error-free services, ensuring that customers get the correct medicines and products promptly.

1.2 Objectives

Objectives provided by Medical Store Management System are as follows:

- To automate the process of managing medicine inventory, including adding, updating, and removing medicines from stock.
- To create a user-friendly interface for customers and store administrators.
- To ensure accurate billing and payment processing.
- To track sales, stock levels, and customer information efficiently.
- To provide detailed reports of daily transactions, medicine stock levels, and sales trends

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

This document specifies the requirements and design process necessary to produce a detailed description of the functionalities for the **Medical Store Management System**. This Software Requirements Specification (SRS) ensures a complete understanding of what is expected from the newly implemented system.

A clear understanding of the system and its functionalities will enable the development of software that meets the needs of both the administrators and end users. This SRS serves as a foundational document, outlining the necessary requirements for the system. It will guide the development of subsequent stages, including design, construction, and testing.

By adhering to this SRS, the **Medical Store Management System** can be efficiently developed to automate and optimize processes such as inventory management, user registration, billing, and reporting. This will result in a reliable, scalable, and user-friendly solution for medical store operations.

1.3.2 Scope

The project simplifies the management of purchasing, sales, and inventory of medicines and medical products. It helps to save time and costs by automating various processes. The system is accessible 24/7, allowing users to perform transactions and manage operations at any time.

The proposed system will have the following users:

Admin:

The admin is the owner or manager of the medical store who oversees and manages the entire system. The admin has authorization to manage the medicine catalog, monitor stock levels, handle user accounts, generate reports, and view sales transactions.

Customers:

The customers are the regular users who can browse, search, and purchase medicines and medical products. They can view their order history, manage their profiles, and receive notifications regarding their purchases.

The goal of this project is to develop a comprehensive system to streamline the sale of medicines and medical products. This system will be designed with the following objectives:

- Effective and efficient order management, ensuring a smooth purchasing process for customers.
- Simplified inventory management to monitor and update stock levels.
- Detailed sales tracking and reporting for better decision-making.

Additionally, by storing transaction history and user data in a database, the system enables easy tracking of customer behavior and purchase patterns. It efficiently manages all operations of a medical store on a single platform, making it scalable, user-friendly, and reliable.

1.3.3 Applicability

Medical Store Management System is a software application that enables users to purchase medicines, medical supplies, personal care products, and health-related items via the internet. The system provides a convenient platform for both **customers** and **admins** to manage their operations and transactions.

This system is applicable to:

- **Online Pharmacies:** Enabling customers to browse and purchase a variety of medicines and health-related products easily.
- **Pharmacy Chains:** Providing an efficient way for multiple locations to manage inventory, orders, and sales through a centralized platform.
- **Healthcare Providers and Clinics:** Allowing healthcare providers to prescribe and deliver medications directly to patients through the system.

The **Medical Store Management System** aims to streamline the purchasing process for customers and manage the backend operations of the store, such as inventory control, billing, and order tracking, all from a single platform. The system is designed to be accessible 24/7, ensuring customers can order products at any time.

1.4 Achievements

Undertaking this project allowed me to learn new technologies, including **React.js**, **JavaScript**, **Python**, and **Django**. I gained hands-on experience in frontend and backend development, user authentication, and inventory management. Additionally, I enhanced my skills in software testing, ensuring the system is functional, error-free, and user-friendly.

1.5 Organization of Report

Chapter 1: Introduction

This chapter introduces the system being developed, providing a description of the background and context of the project. It includes a concise statement of the project's aims, objectives, purpose, scope, and applicability.

Chapter 2: Survey of Technologies

This chapter details the technologies used to develop the system. It demonstrates an understanding of the technologies available for the project, such as **React.js**, **JavaScript**, **Python**, and **Django**.

Chapter 3: Requirements and Analysis

This chapter discusses the requirements leading to the development of the system. It includes problem definition, requirement specification, and planning and scheduling. It also specifies the hardware and software requirements, including the operating system and other necessary software to run the project.

Chapter 4: System Design

System Design describes the features and operations of the system in detail. This chapter covers data design, functional and interface design, process diagrams, function descriptions, and test case design. It includes the Basic Modules section, which explains all the modules and their functionalities, and the Data Design section, covering schema design and data integrity and constraints.

Chapter 5: Implementation and Testing

This chapter outline the plan for implementation, the standards used, code efficiency, and the testing approach. It includes integrated testing, modifications, and improvements to the project during its implementation phase.

Chapter 6: Results and Decisions

This chapter presents the test results based on the defined test cases, showing that the software performs correctly under different conditions. It also provides user documentation, explaining the system's functionality with screen shots to help users understand the software's operation.

Chapter 7: Conclusions

This chapter summarizes the entire work, reviewing the key points from previous chapters. It discusses the limitations of the system, the challenges faced during development, and the future scope of the project, proposing possible enhancements and new areas for investigation.

CHAPTER 2: SURVETY OF TECHNOLOGY

2.1 HTML & CSS

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receives HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2.2 JavaScript

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

2.3 Python

Python is a high-level, general-purpose, and interpreted programming language used in various sectors including machine learning, artificial intelligence, data analysis, web development, and many more. Python is known for its ease of use, powerful standard library, and dynamic semantics. It also has a large community of developers who keep on contributing towards its growth. The major focus behind creating it is making it easier for developers to read and understand, also reducing the lines of code.

2.4 Django

Django is a Python-based web framework that allows us to quickly create efficient web applications. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLite3, etc. When building a website, it always needs a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for website, forms, a way to upload files, etc. Django gives ready-made components to use and that too for rapid development.

2.5 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a popular choice as an embedded database for local/client storage in application software such as web browsers. It is also used in many other applications that need a lightweight, embedded database.

SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it's less than 500kb in size, which is significantly less than other database management systems.

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

3.2 Requirements Specification

1. Introduction

The problem that the Medical Store Management System seeks to solve is the inefficiency and inconvenience associated with traditional medical stores. Before the availability of an online system, customers had to visit medical stores physically to purchase medicines, which could be a time-consuming and inconvenient process. This approach also increased the likelihood of human errors in inventory and sales management.

An online Medical Store Management System addresses these issues by streamlining the process of buying and selling medicines through an online platform. Customers can conveniently browse and purchase medicines at any time and from anywhere, making the process more efficient and accessible. For store administrators, the system helps automate various tasks like inventory management, sales tracking, and payment processing, reducing errors and enhancing operational efficiency. The system is designed to meet the specific needs of the medical store to ensure smooth and effective operations.

1.1 Purpose

This document specifies the requirements for designing and implementing the Medical Store Management System. It outlines a comprehensive description of the system's functionality, providing a clear understanding of the features to be developed for end-users. The Software Requirements Specification (SRS) serves as the foundation for the project, enabling the design, construction, and testing of the system.

1.2 Document Conventions

The document is prepared using Microsoft Word 2019 and has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document. Standard IEEE template is the template used to organize the appearance of the document and its flow.

1.3 Intended Audience and Reading Suggestions

The intended audience includes:

- **Admins/Managers** of the medical store, who will use the system to manage products, sales, and customers.
- **Customers**, who will use the system to purchase medicines.
- **Development Team**, to understand and implement the requirements effectively.

Outline of the Document:

1. Overall Description
2. System Features
3. External Interface Requirements
4. Non-Functional Requirements

1.4 Project Scope

The Medical Store Management System streamlines the process of purchasing and managing medical products. It aims to save time and reduce costs for both customers and store administrators. Operating 24/7, the system allows users to purchase medicines and related products at any time and from anywhere.

The proposed system serves two primary user roles:

- **Admin:** The admin, typically the owner or manager of the medical store, oversees the online platform. They are responsible for managing products, tracking payments, and monitoring overall system operations.
- **Customers:** Customers are regular users who can browse, search, and purchase various categories of medical products through the platform.

The primary objective of the system is to create an efficient, user-friendly, and scalable online platform for selling medical products. By automating critical tasks, the system ensures effective order management, minimizes manual errors, and enhances customer convenience.

Additionally, the system provides detailed insights into customer purchasing behavior and patterns by storing order history in a centralized database. This feature enables informed decision-making and personalized service. Overall, the Medical Store Management System efficiently integrates all operations of an online store into a single, cohesive platform.

1.5 References

Textbooks referred:

- a. Singh Yogesh (2008). Software Engineering Textbook
- b. Pankaj Jalote (1991). An Integrated Approach to Software Engineering

Websites referred:

<https://ieeexplore.ieee.org/document/720574>

2. Overall Description

2.1 Product Perspective

The Medical Store Management System integrates both customer and admin perspectives. The system is database-driven and provides a web interface where customers can browse and purchase medicines, while admins can manage inventory and track sales. All data is stored in a centralized database, ensuring seamless interaction between subsystems

2.2 Product Features

- Allows customers to purchase medicines online without visiting the store.
- Provides an online payment facility for easy transactions.
- Saves time and effort for users by enabling anytime, anywhere access.
- Offers scalability for adding features in the future.
- Reduces manual labor through automation.
- Facilitates fast and accurate information retrieval.
- Ensures a user-friendly experience.

2.3 User Classes and Characteristics

The system has two user levels.

Admin: The admin can log in to this website and can perform the following activities:

- Login
- Manage Profile
- Manage Products
- View Payment

Customer: The Customer can register to this website and can perform the following activities:

- Registration
- Login
- View Product details
- Buying Products
- Make Payment

2.4 Operating Environment

- Operating system: Windows 10 and above
- Browsers: Chrome, Firefox or any other browsing application
- Processor: Intel dual core or above
- Processor Speed: Minimum 2 GHz
- RAM: 4 GB or above
- Hard Disk: Minimum 40 GB

2.5 Design and Implementation Constraints

The application design should ensure compatibility with all Windows devices while maintaining a user-friendly interface. The following are the design constraints from a software standard perspective:

- **Secure Login System:** The system must allow users to log in only after completing the registration process.
- **Admin Control:** The admin should have the capability to update the details of medical products available in the system, ensuring that the information remains accurate and up to date.
- **Product Display and Purchase:** The system should display detailed information about the medical products available. Customers should be able to view product details and purchase items as per their requirements.

2.6 User Documentation

A user manual will be provided to the client to offer a clear understanding of how to interact with the system. The manual will be written in simple and easily understandable language, effectively hiding the system's internal complexities. A hard copy of the user manual will be included with the delivery system, ensuring the client has a comprehensive guide for its operation.

2.7 Assumptions and Dependencies

The project is web-based and designed for computers with the following requirements:

- **User Authentication:** Each user needs a unique username and password.
- **Administrator:** Only one admin manages the system.
- **Server and Browser:** The server must run on Windows, and a compatible browser is required.
- **Database Connectivity:** Reliable database connections are essential.
- **Additional Needs:** A stable internet connection and a printer for invoice generation.

3.2. System Feature 2: Medicines

3.2.1. Description and Priority

The customer should be able to browse and order medicines.

Priority Type: High

3.2.2. Stimulus/Response Sequences

- **Stimulus:** User searches for a medicine.
Response: System displays a list of matching medicines.
- **Stimulus:** User clicks on a medicine for more information.
Response: System displays detailed medicine information.
- **Stimulus:** Customer requests to add a medicine to the cart.
Response: System displays cart details with the added medicine.
- **Stimulus:** Customer requests to place an order for medicines.
Response: System provides an order form for confirmation.

3.2.3. Functional Requirements

- **REQ1:** The customer should be able to browse, search, and order medicines.
- **REQ2:** The customer should be able to add medicines to the cart for checkout.

3.3. System Feature 3: Admin Management

3.3.1. Description and Priority

The admin can manage medicine details (Add, View, Update, Delete) and oversee orders, payments, and inventory.

Priority Type: High

3.3.2. Stimulus/Response Sequences

- **Stimulus:** Admin requests the login page.
Response: System displays the login page.
- **Stimulus:** Admin enters their username and password and clicks login.
Response: Admin dashboard is displayed upon successful login.

- **Stimulus:** Admin requests to add medicine details.
Response: System provides a form to add medicine details.
- **Stimulus:** Admin requests to view medicine details.
Response: System displays the medicine details.
- **Stimulus:** Admin requests to update medicine details.
Response: System provides a form to update medicine details.
- **Stimulus:** Admin requests to delete medicine details.
Response: System provides a confirmation option to delete the selected medicine.
- **Stimulus:** Admin requests to view order details.
Response: System displays order details.
- **Stimulus:** Admin requests to view payment details.
Response: System displays payment details.
- **Stimulus:** Admin requests to logout.
Response: System logs the admin out and redirects to the login page.

3.3.3. Functional Requirements

- **REQ1:** Admin should be able to add, update, delete, and view medicine details.
- **REQ2:** Admin should be able to manage orders, payments, and inventory efficiently.

3. External Interface Requirements

3.1 User Interfaces

User UI:

- Home Page: Displays products for browsing and searching.
- Product Pages: Shows detailed information for selected medicines, including price and description.
- Shopping Cart: Allows users to add medicines, view their cart, and proceed to check out.
- Checkout: Collects user shipping and payment details to confirm orders

Admin UI:

- Medicine Management: Adding, updating, and deleting product.
- Order Management: Tracking and handling customer orders.

3.2 Hardware Interfaces

This system requires basic computers that consist of CPU, monitor, keyboard and mouse or laptop for input and output. A printer is needed for print functions.

- Hard Disk: Minimum 40 GB Hard Disk
- RAM: Minimum 4 GB RAM
- Processor: Intel Pentium or Above

3.3 Software Interfaces

- Backend language: Python
- Backend Framework: Django
- Front-end languages: React, JavaScript, HTML, CSS
- Database: SQLite
- Code editor: VS code
- Operating system: Windows/Linux

3.4 Communications Interfaces

Like any web-based system HTTP protocol is used to transfer data the website is secured with HTTPS so that data is transferred securely.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

- The application works with any windows OS.
- Software should respond within the estimated time to display results to the user.
- Should have good memory space.

4.2 Safety Requirements

- The system should be capable of restoring itself to its previous state in the event of failure (e.g., a system crash or power loss).
- The system should be able to always display a menu to facilitate manual order taking should the need arise.
- Authorization: Checking for the entity and provide features for them.

4.3 Security Requirements

- Access to the system is restricted to registered users only.
- Only respective users can modify their personal details.
- Authentication is mandatory during login.
- During user registration, the system verifies the email ID and mobile number for uniqueness.

4.4 Software Quality Attributes

- **Availability:** System operational 24/7.
- **Correctness:** Meets user specifications and objectives.
- **Reliability:** Ensures consistent performance with minimal downtime; defines mean time between failures and recovery.
- **Robustness:** Handles functions accurately, maintaining database integrity without unexpected failures.
- **Testability:** Ensures the system can be tested efficiently to confirm intended functionality.
- **Usability:** Prioritizes user-friendliness for ease of use and efficiency.

5. Other Requirements

- Organizational permission is required to use proprietary logos and assets.
- The database must scale to manage large datasets in the long term.

3.3 Planning and Scheduling

- **Define Project Scope:** Clearly establish goals, objectives, and deliverables for the online baby shop.
- **Feature Identification:** List key functionalities and requirements.
- **Create Project Plan:** Develop a detailed plan outlining tasks, timelines, and required resources.
- **Task Dependencies:** Identify critical paths and interdependencies.
- **Assign Responsibilities:** Allocate tasks to team members, ensuring clarity of roles.
- **Develop Schedule:** Use tools like Microsoft Project to draft timelines and assign responsibilities.
- **Risk Management:** Identify potential risks and create mitigation strategies.
- **Monitor Progress:** Regularly track milestones and task completion using project management software.
- **Testing and Deployment:** Test the system comprehensively, then deploy it in a live environment.
- **User Training:** Train end users on how to use the system effectively.
- **Project Evaluation:** Review the project to assess successes and areas for improvement for future initiatives.

3.3 Software and Hardware Requirements

Hardware requirements

- Processor: Intel dual core or above
- Processor speed: Minimum 2 GHz
- Hard disk: Minimum 40GB
- RAM: 4GB or above

Software requirements

- Operating system: Windows 7 or above
- Text editor: VS Code
- Server: Apache
- Web Browsers: Google Chrome/ Mozilla Firefox/ Internet explorer

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

User Modules:

Registration/Login

Home

Products

Cart

Order

Admin Modules:

Admin Login

Add Medicine

Manage Order

4.2 Data Design Project Structure

Database design is done before building it to meet the needs of end-users within a given information system that the database is intended to support.

Primary key: The primary key of a table uniquely identifies each record in the table, so that an individual record can be located without confusion.

Foreign Key: It is a key used to link two tables together. The Primary key is the target that a foreign key can reference. The Foreign key identifies a column or a set of columns in one table that refers to set of columns in another table.

NOT NULL: It allows you to specify that a column may not contain NULL values.

Database Table: customer

Attribute	Data Type	Constraints
user	int(20)	Primary key
name	varchar(200)	Not Null
locality	varchar(200)	Not Null
city	varchar(50)	Not Null
zipcode	int	Not Null

state	varchar(50)	Not Null
-------	-------------	----------

product

Attribute	Data Type	Constraints
-----------	-----------	-------------

pid	int(20)	Primary Key
title	varchar(100)	Not Null
selling_price	float	Not Null
discount_price	float	Not Null
description	varchar(250)	Not Null
category	varchar(2)	Not Null
product_image	image	Not Null

cart

Attribute	Data Type	Constraints
cart_id	int(20)	Primary Key
user	int(20)	Foreign Key
pid	int(20)	Foreign Key
quantity	int	Not Null
razor_pay_order_id	varchar(100)	Null
razor_payment_id	varchar(100)	Null
razor_pay_payment_signature	varchar(100)	Null

orderplaced

Attribute	Data Type	Constraints
oid	int	Primary Key
user	int	Foreign Key
pid	int	Foreign Key
quantity	int	Not Null
ordered_date	datetime	Not Null
status	varchar(50)	Not Null

payment

Attribute	Data Type	Constraints
razorpay_payment_id	varchar(100)	Primary Key
user	int	Foreign Key
razorpay_order_id	varchar(100)	Null
Razorpay_payment_status	varchar(100)	Null
paid	boolean	Not Null

4.2.1 Schema Design

4.2.2 Data Integrity and Constraints

Primary key: The primary key of a table uniquely identifies each record in the table, so that an individual record can be located without confusion. The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns

Foreign Key: It is a key used to link two tables together. Primary key is the target which a foreign key can reference. Foreign key identifies a column or a set of columns in one table that refers to set of columns in another table. A FOREIGN KEY is a field in one table, that refers to the PRIMARY KEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

NOT NULL constraint: It allows you to specify that a column may not contain NULL values. The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

4.3 Procedural Design

4.3.1 Logic Diagrams

4.3.3.1 Use Case Diagrams

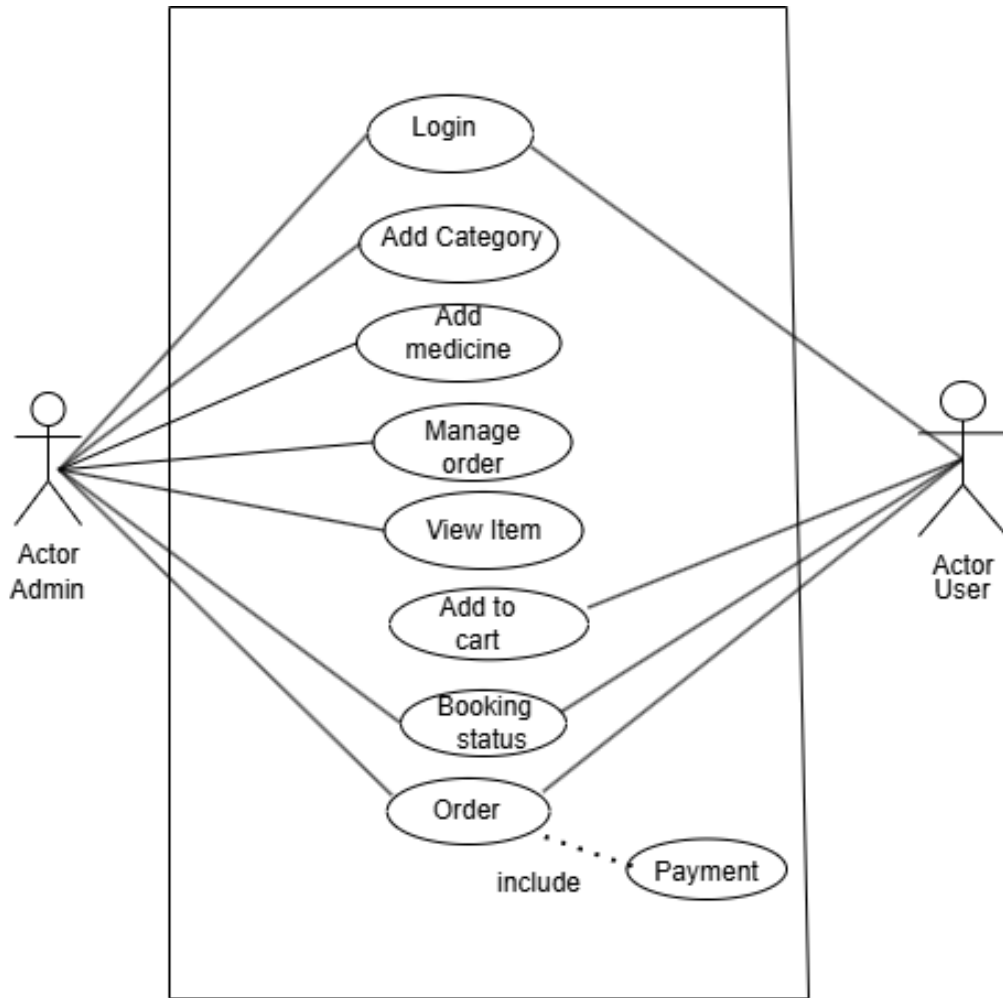


Figure 1: Use case Diagram

4.3.3.2 Class Diagram

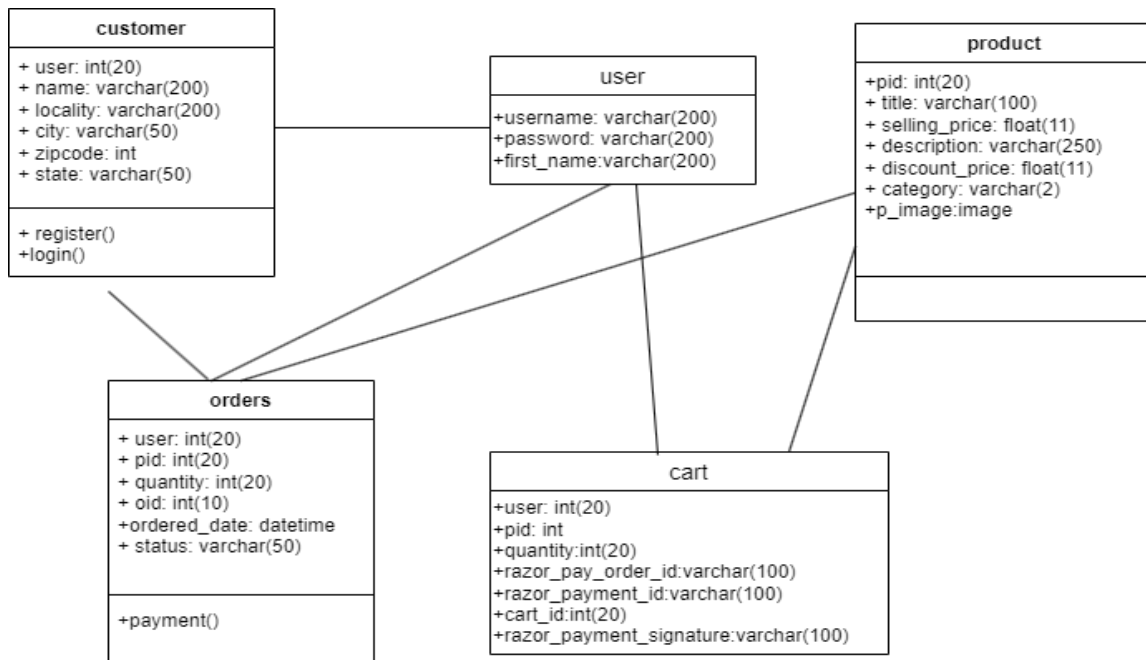


Figure 2:Class Diagram

4.3.3.3 Sequence State Chart

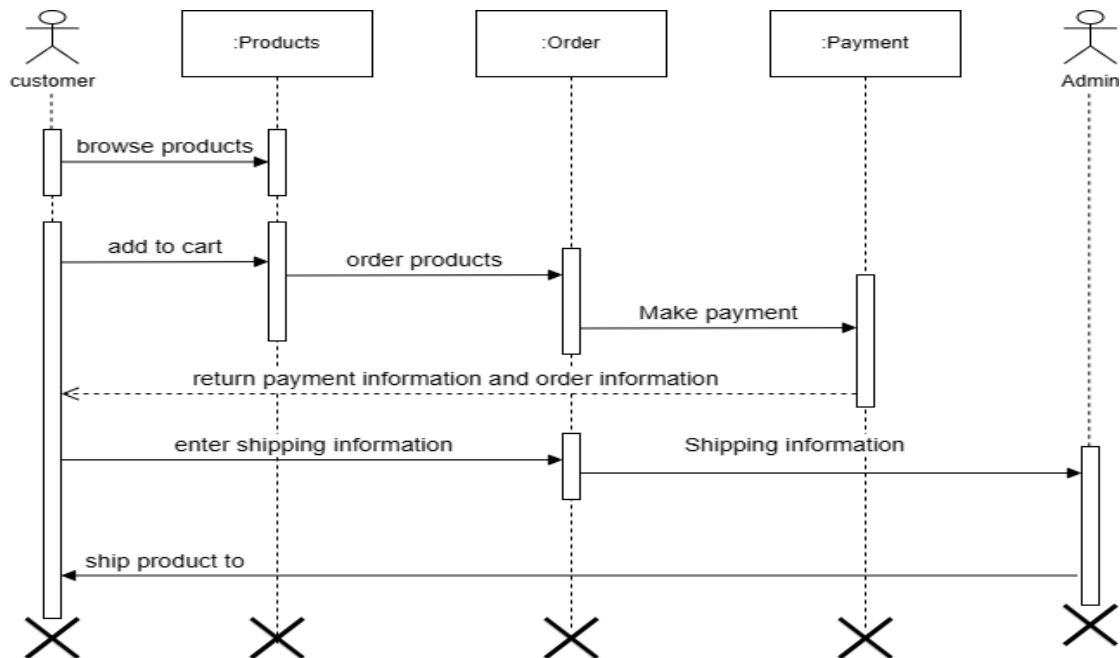


Figure 3: Sequence State Chart

4.3.3.4 State Chart Diagram

Figure 4: State Chart Diagram

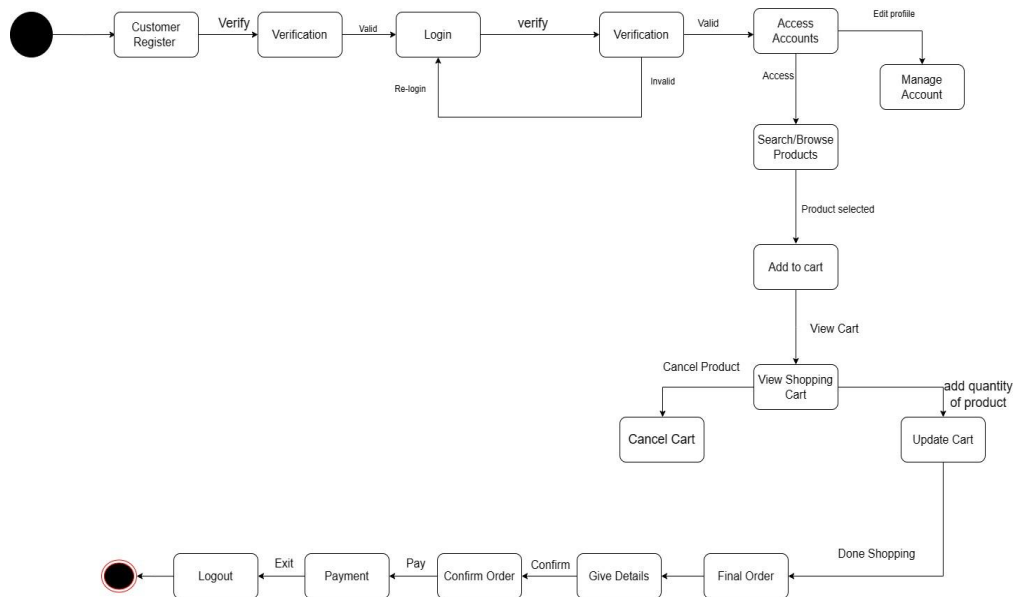


Figure 4: State Chart Diagram

4.3.3.5 Activity Diagram

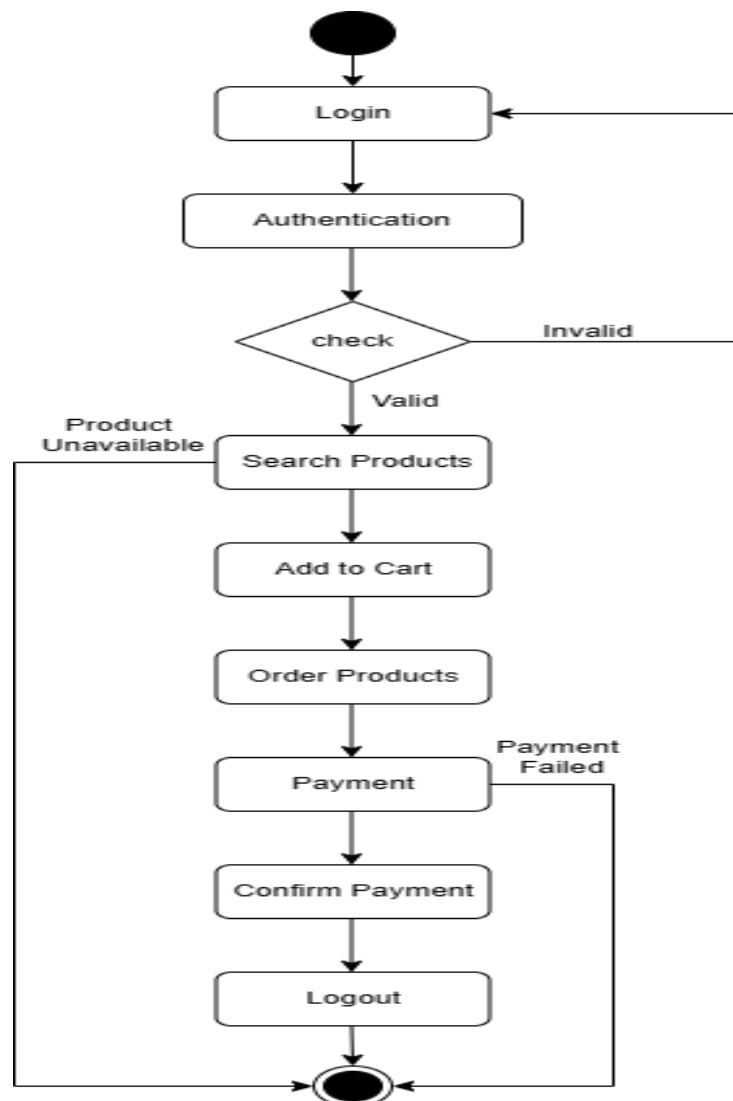


Figure 5: Activity Diagram

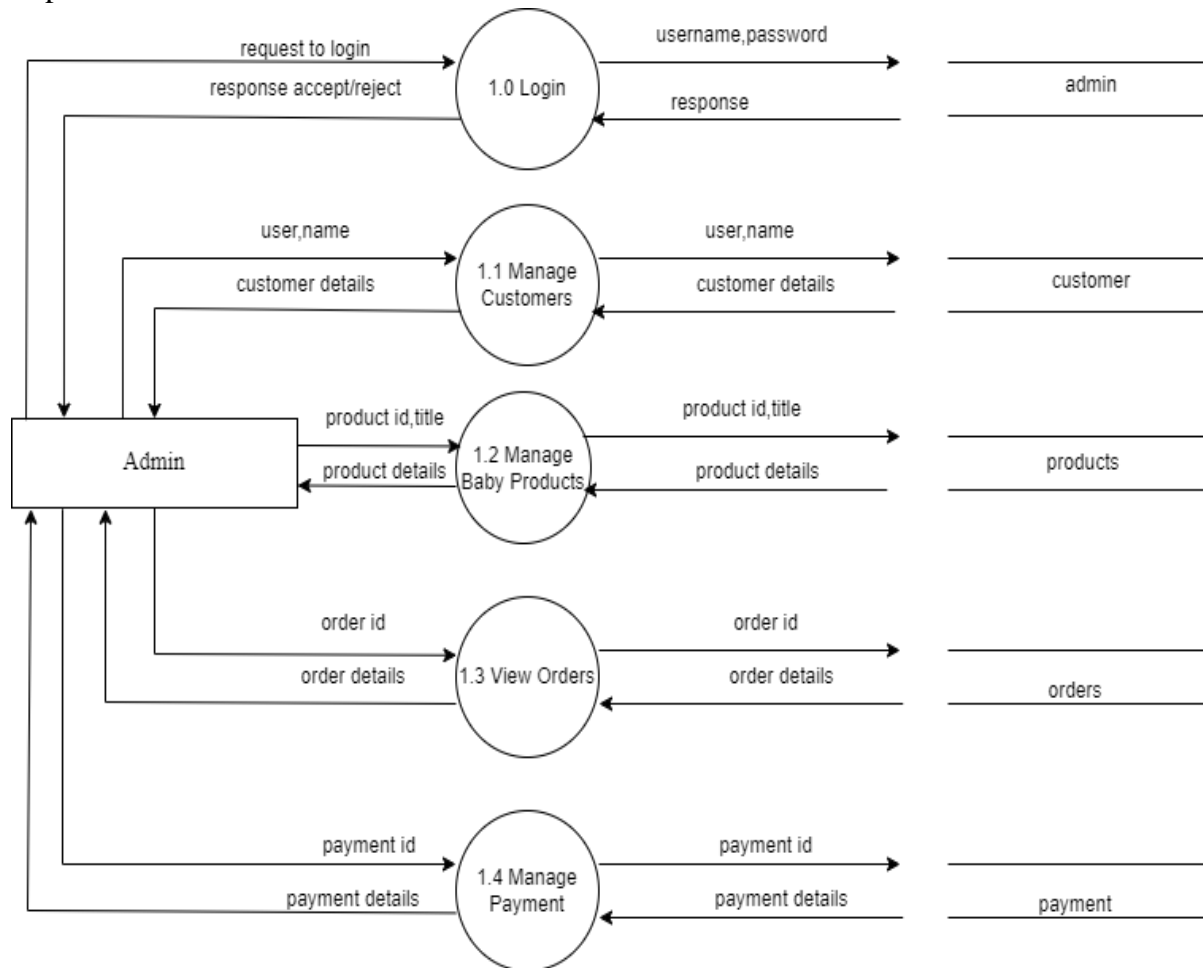
DFD (Level 0)

Context flow diagram is a top level (also known as level 0) data flow diagram. It contains only one process node that generalizes the functions of the entire system in relationship to external entities.

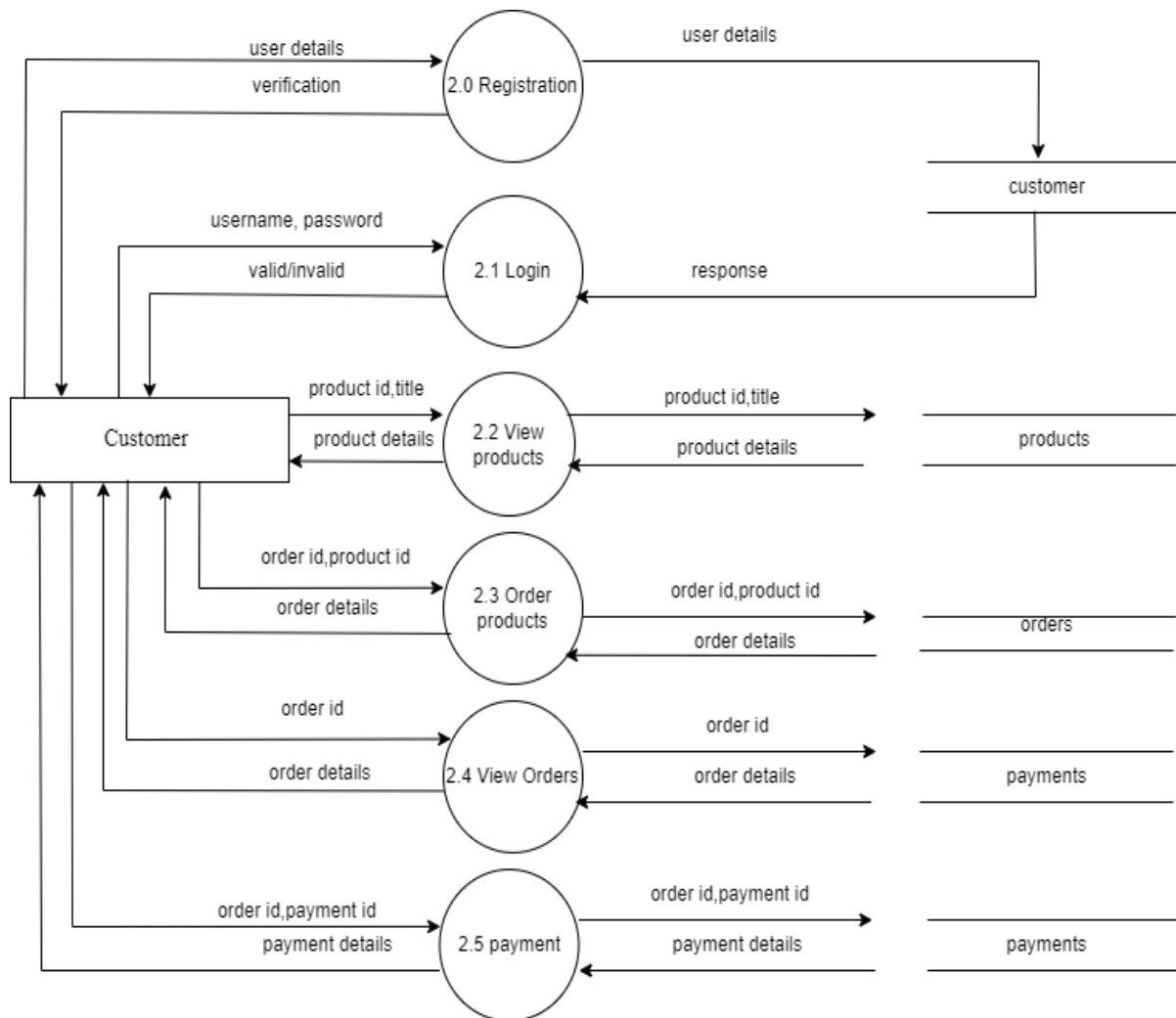


DFD (Level 1)

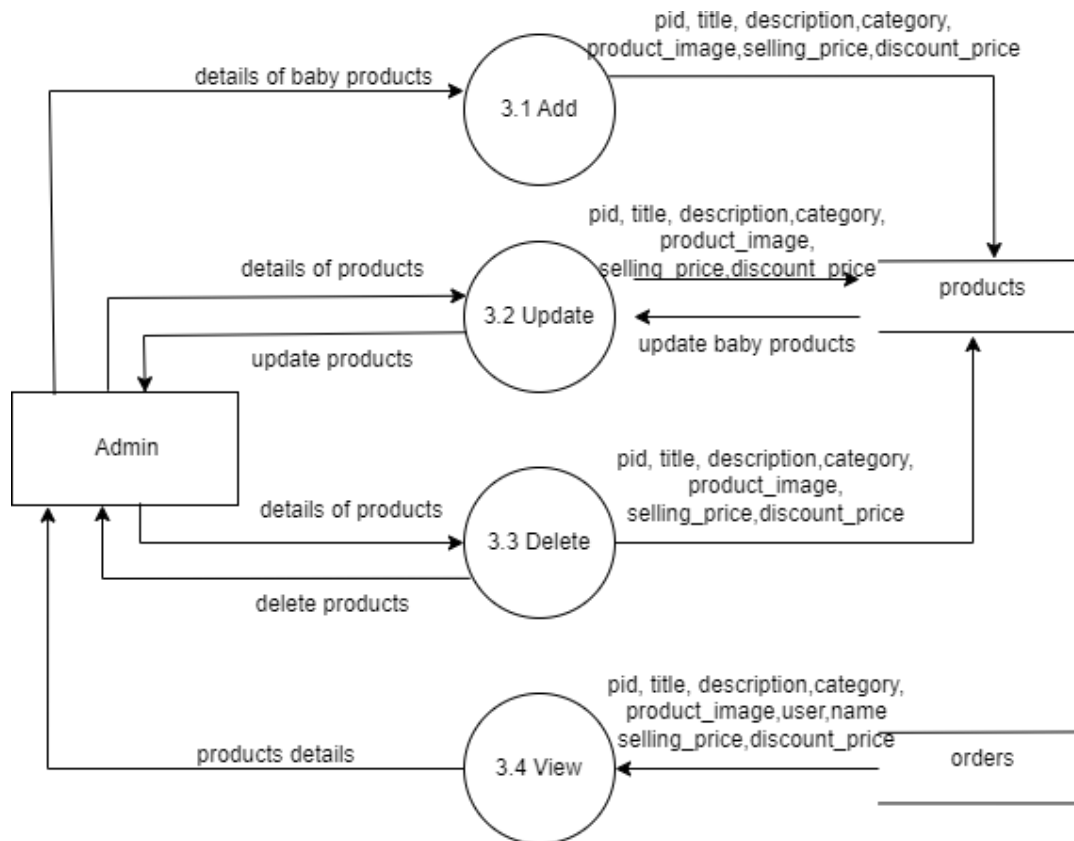
A level 1 data flow diagram (DFD) is more detailed than a level 0 but not as detailed as a level 2 DFD. It breaks down the main processes into sub processes that can then be analyzed and improved on a more intimate level.



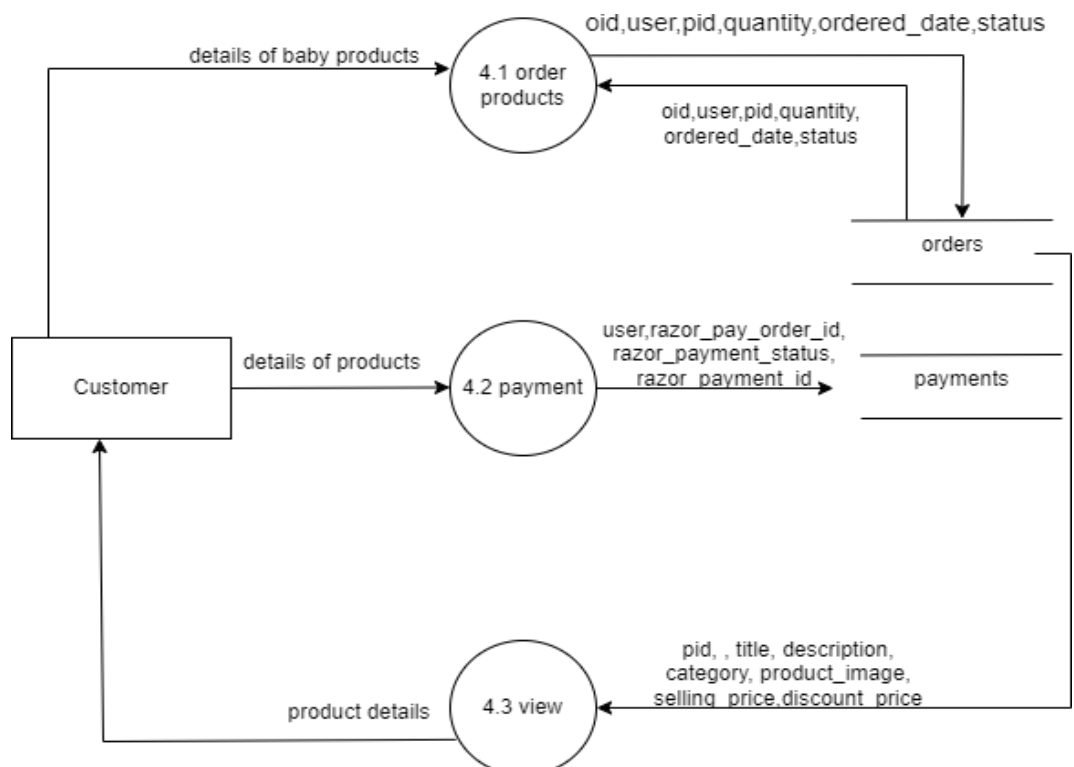
DFD Admin Level 1



DFD Customer Level 1

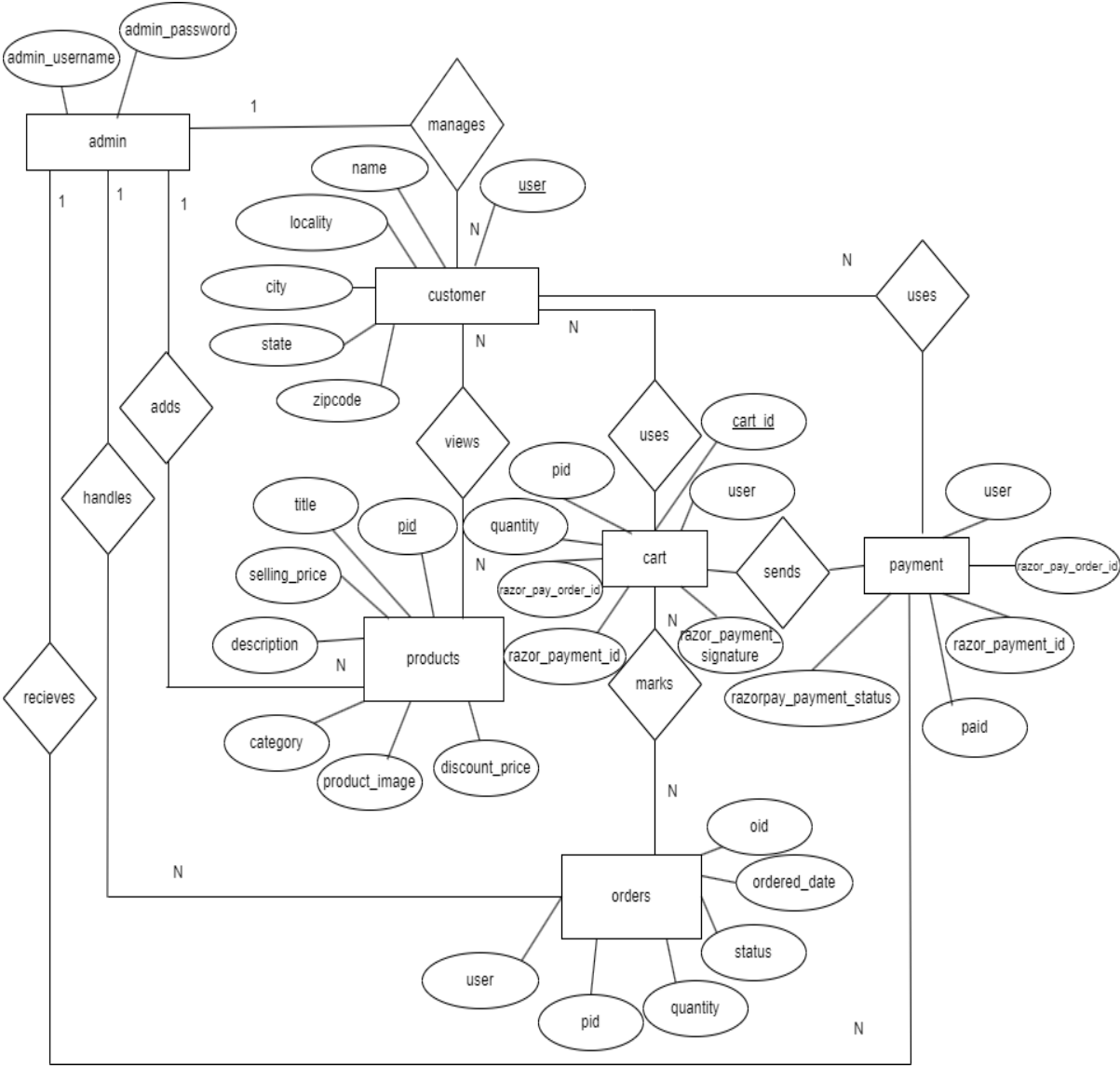


DFD Admin Level 2



DFD Customer Level 2

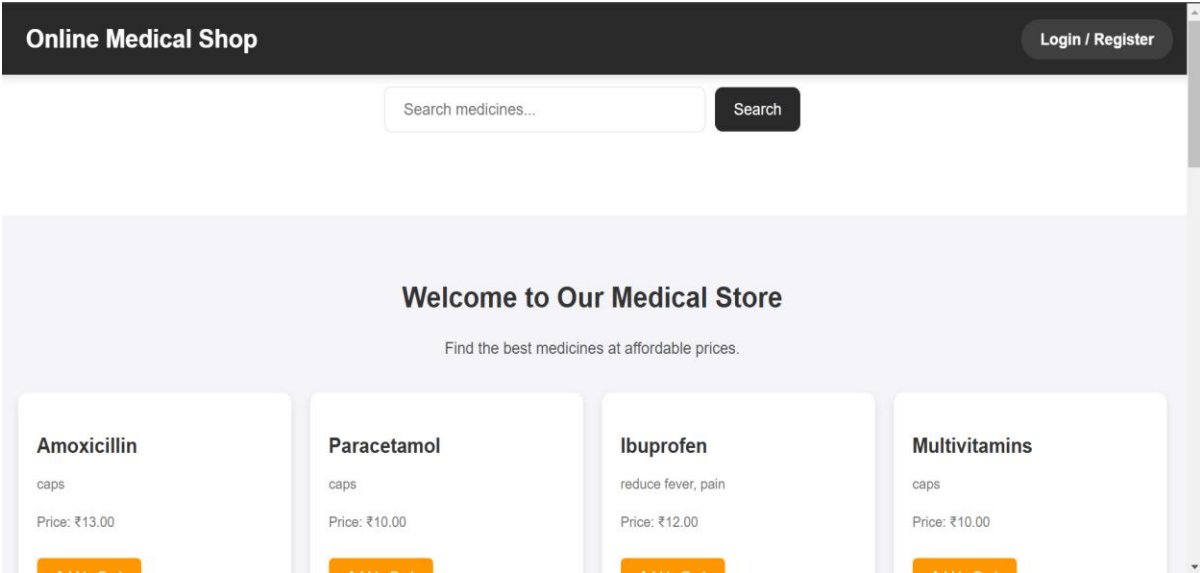
ER Diagram



Entity Relationship Diagram

4.4 User Interface Design

Home Page



Product Page:

Online Medical Shop Admin					
<div>View BookingsChange PasswordMedicines ▼Admin, qwerty</div>					
Medicine Name	Category	Price	Description	Stock Quantity	Action
Amoxicillin	CAPSULES	13.00	caps	50	Delete
Paracetamol	CAPSULES	10.00	caps	50	Delete
Ibuprofen	CAPSULES	12.00	reduce fever, pain	50	Delete
Multivitamins	CAPSULES	10.00	caps	50	Delete
Omeprazole	CAPSULES	15.00	caps	50	Delete
Dextromethorphan	SYRUP	100.00	Cough Syrup	30	Delete
Paracetamol	SYRUP	70.00	Syrup	30	Delete
Antacid	SYRUP	50.00	Syrup	20	Delete

Add to Cart:

Online Medical Shop

[View Bookings](#)[Change Password](#)[Welcome, husen](#)

Continue Shopping

Your Cart

Medicine	Quantity	Actions
Amoxicillin	1	Remove
Paracetamol	1	Remove
Ibuprofen	1	Remove
Multivitamins	1	Remove

Total: ₹45

Proceed to Booking

Registration Page:

Online Medical Shop

[Login / Register](#)

First Name:

first_name

Last Name:

last_name

Username:

username

Email:

email

Password:

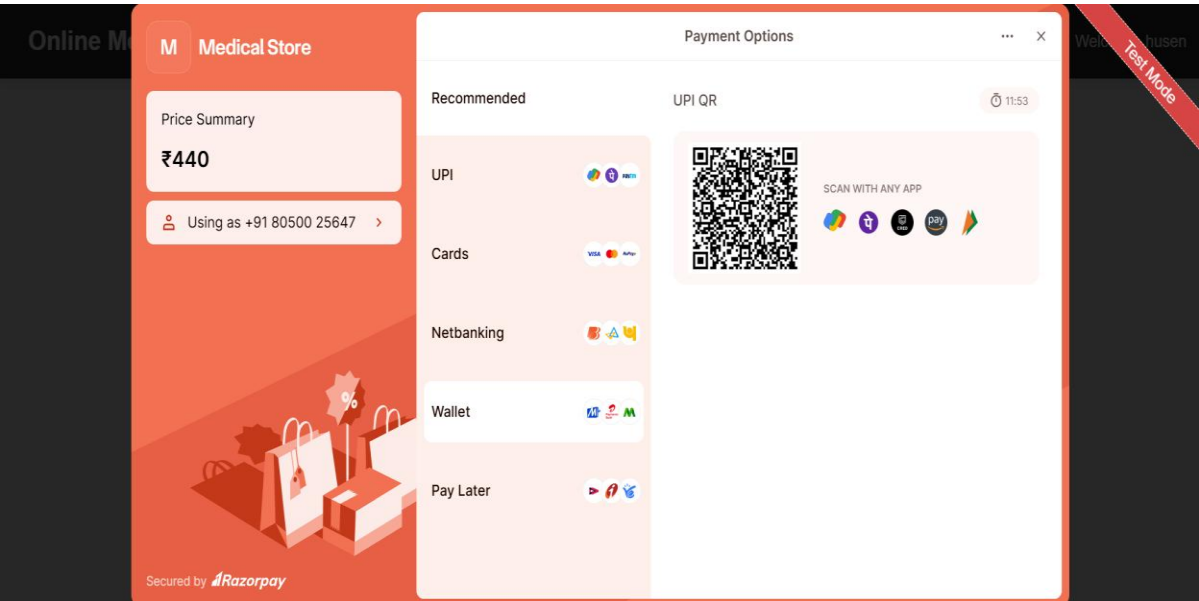
password

Confirm Password:

cpassword

Register

Payment Page:



CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Introduction

Implementation is the stage in the project where theoretical design is turned into a working system and is giving confidence on the new system for the users that will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning, the major task of preparing the implementation is education and training of users.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end delivery is the product itself. During the implementation phase, the system is built according to the specifications from the previous phases. This includes writing code, performing code reviews, performing tests, selecting components for integration, configuration, and integration.

5.2 Testing Objective

The objectives of testing are:

- Testing is a process of executing a program with the intent of finding errors.
- A successful test case is one that discovers an as of yet and discovered error. System testing is a stage of implementation which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences as stated before, testing is vital to the success of a system testing makes a logical assumption that if all parts of system are correct the goal will successfully be achieved. A series of tests are performed before the system is ready for the user acceptance test.

5.3 Unit Testing:

Unit testing is a software testing technique where individual components or modules of a system are tested in isolation to ensure they function correctly. In the context of a medical store management system, unit testing focuses on verifying the functionality of specific modules, such as inventory management, user registration, and order processing.

For example, in the user registration module, unit tests would verify that the system correctly handles user inputs, validates the entered data, and ensures that the user is successfully registered. In the inventory management module, tests could check that stock levels are updated properly when products are added or removed, and that alerts are generated when stock runs low.

By writing test cases for each module, the developer ensures that all parts of the system perform as expected. Unit testing helps catch errors early in the development process, making it easier and cheaper to fix issues before they grow into larger problems. This approach ultimately improves the overall quality and stability of the medical store management system.

5.4 Integration Testing

Integration testing is a type of software testing that checks the interaction between different modules or components of a system to ensure that they work together as intended. In the context of an online baby shop system, integration testing would involve testing the integration of the various modules or components to ensure that they function correctly when used together.

In an medical store management system, there might be modules or components for user registration, payment processing, and email notifications. Integration testing would involve testing how these components work together to complete a process, from user registration to payment processing to email confirmation. During integration testing, test cases would be created to verify the functionality of the integrated system.

Testing:

User Registration:

Test Case ID	Test Condition	Expected Output	Result
1	If numbers or symbols are entered in name field	"Name can only contain letters"	SUCCESS
2	If any field is left empty	"All fields are required"	SUCCESS
3	If email format is entered incorrectly	"Invalid email format"	SUCCESS
4	Confirm password	"Password and Confirm password do not match"	SUCCESS
5	Strong password	"Password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one number, and one special character."	SUCCESS
7	If registered email exists already	"Existing user found with the same email address"	SUCCESS
8	On successfully registering	Redirect to Home page of registered user	SUCCESS

User Login

Test Case ID	Test Condition	Expected Output	Result
1	If any field is empty	"All fields are required"	SUCCESS
2	If the username is not registered	"Username not registered"	SUCCESS
3	If the entered password is incorrect	"Wrong Password"	SUCCESS
4	If successfully logged in	"Redirect to home page of registered users"	SUCCESS

Cart page:

Test Case ID	Test Condition	Expected Output	Result
1	If cart is empty	It should show empty cart	SUCCESS
2	If '-' button is clicked	Should decrease the quantity, price and total price.	SUCCESS
3	If '+' button is clicked	Should increase the quantity, price and total price.	SUCCESS
4	If remove button is clicked	Should decrease the quantity, price and total price.	SUCCESS

Checkout page:

Test Case ID	Test Condition	Expected Output	Result
1	If payment successful	Should display modal payment successful in website	SUCCESS
2	If payment failed	Should redirect to payment failed page	SUCCESS

CHAPTER 6: CONCLUSIONS

4.3 Conclusion

The **Medical Store Management System** provides an efficient platform for both customers and admins to manage and access medical products. Customers can browse products, view their details, and place orders seamlessly. The system allows customers to access the latest product information, while the admin has control over the management of the store's inventory, including adding, updating, and deleting products.

Working on this project has provided valuable insights into the software development process, particularly in terms of system design and planning. The system is designed to simplify daily operations and improve the overall efficiency of a medical store. It benefits store owners by streamlining inventory management, while customers benefit from an easy-to-navigate platform for purchasing medications and other related products.

4.4 Limitations

- **Internet Dependency:** A stable internet connection is required for users to access the system, which may limit usage in areas with unreliable connectivity.
- **Web Application Only:** Currently, the system is available only as a web application, with no mobile application support for Android or iOS.
- **No Online Payment Integration:** The system does not currently support online payment processing for customer orders.
- **Limited Search Functionality:** Advanced product search features are not yet integrated, limiting product discovery for users.

4.5 Future Scope

The **Medical Store Management System** has great potential for future enhancements, including:

- **Mobile App Support:** Expanding to Android and iOS for easier access and purchases.
- **Online Payment Integration:** Enabling secure online payments for a smoother shopping experience.
- **Advanced Search:** Adding better filters and categories for quicker product discovery.
- **Inventory Management:** Introducing stock alerts, reporting, and analytics for efficient store management.

References

- PANKAJ JALOTE's SOFTWARE ENGINEERING: A PRECISE APPROACH by PankajJalote (Author).
- <https://www.lucidchart.com/pages/uml-use-case-diagram>
- <https://www.geeksforgeeks.org/design-documentation-in-software-engineering/>
- <https://www.ibm.com/docs/en/db2-for-zos/12?topic=modeling-entity-attributes>
- <https://ecomputernotes.com/software-engineering/software-design-documentation>

Appendix A: Glossary

SRS: Software Requirement Specification

OS: Operating System

GUI: Graphical User Interface

HTML: Hypertext Markup Language

SQL: Structured Query Language

CSS: Cascading Style Sheets

HTTPS: Secure Hypertext Transfer Protocol

HTTP: Hypertext Transfer Protocol