VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Report on

"Medical Store Management System"

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted by

SURABHI K C - 1RN20IS167

Under the Guidance of

Ms. VINUTHA G K

Assistant Professor

Dept. of ISE, RNSIT

ESTD . 2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

2023 -2024

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,

Channasandra, Bengaluru-560098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ESTD . 2001

An Institute with a Difference

CERTIFICATE

Certified that the Internship work entitled ***Medical Store Management System*** has been successfully completed by **Surabhi K C (1RN20IS167)** bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year 2023-2024. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Ms. Vinutha G K Internship Guide Assistant Professor Department of ISE	**Dr. Nirmalkumar S** **Benni/ Ms. Aruna U** Internship Coordinators Associate/Assistant Professors Department of ISE	**Dr. Suresh L** Professor and HOD Department of ISE RNSIT

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

I, **Surabhi K C [USN: 1RN20IS167]** students of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled **Medical Store Management System** has been carried out and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*during academic year 2023-2024.

Place: Bengaluru
Date:

SURABHI K C
1RN20IS167

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved director and Principal **Dr. M K Venkatesha, Dr. H S Ramesh Babu**, for providing us all the facilities.

I 'am extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all his wisdom.

I place my heartfelt thanks to **Ms. Vinutha G K** Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Kiran Gopalakrishna, Founder & CEO, TechieAid**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank my internship coordinator **Dr. Nirmalkumar Benni**, Associate Professor, and **Ms. Aruna U**, Assistant Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. We have made an honest effort in this assignment.

Surabhi K C

1RN20IS167

Date: 21 November 2023

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Ms. Surabhi K C** a student from RNSIT with USN: 1RN20IS167 has successfully completed Internship program in **Advanced Full Stack Development** from August 2023 till September 2023.

During this period of internship, Ms. Surabhi K C worked on various areas of softwaredevelopment including project work titled **"Medical Store Management System"**. He has successfully met the objectives of the internship program and him conduct was found to be satisfactory.

We wish her all the best for future endeavors

Kiran Gopalakrishna,
Founder & CEO,
TechieAid, Bangalore

ABSTRACT

A Medical Store Management System is a software application designed to streamline and automate the operations of a medical or pharmaceutical store. The primary goal of such a system is to efficiently manage the inventory, sales, purchases, and overall workflow of the store.

In the rapidly evolving healthcare industry, the efficient management of medical stores is critical to ensure the availability of essential medications and supplies. The Medical Store Management System (MSMS) is a comprehensive software solution designed to enhance the operational efficiency of medical and pharmaceutical stores.

The Medical Store Management System is equipped with a user-friendly interface, allowing store administrators to manage various aspects of their operations seamlessly. The system offers modules that cater to inventory management, sales tracking, purchase orders, and reporting, providing a holistic approach to streamline day-to-day activities. The Medical Store Management System aims to empower medical store owners and administrators with the tools needed to optimize their operations, reduce manual errors, and enhance customer service. By leveraging technology, this system contributes to the overall efficiency and sustainability of medical stores in the dynamic healthcare landscape.

The Medical Store Management System represents a paradigm shift in how medical stores operate, leveraging cutting-edge technologies to not only streamline day-to-day activities but also to position themselves as agile and responsive entities in the dynamic healthcare landscape. By embracing innovation, Medical Store Management System sets a new standard for efficiency, compliance, and customer satisfaction in the pharmaceutical retail sector.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

ERP - Enterprise resource planning

IT - Information Technology

MES - Manufacturing Execution Solutions

SRS - Software Requirement Specification

WMS: Warehouse Management System

WWW: World Wide Web

GUI: Graphical User Interface

MVC - Model-View-Controller

UI - User Interface

DFD - Data Flow Diagram

SQL - Structured Query Language

SDK - Software Development Kit

SSMS - SQL Server Management Studio

MS VS - Microsoft Visual Studio

MSMS – Medical Store Management System

Chapter 1

INTRODUCTION

"Medical Store" is a medicine management system that is designed to assist with the management, monitoring, and record-keeping of a medicines retail store. The healthcare industry plays a pivotal role in ensuring the well-being of individuals, and at the core of this ecosystem are medical stores responsible for the efficient dispensing of medications and healthcare products. As the demands on these establishments grow, the need for effective and streamlined management becomes increasingly apparent. The advent of technology has brought forth the Medical Store Management System, a comprehensive solution designed to revolutionize the way medical stores operate.

1.1 Background

The use of computer systems to manage and monitor retail stores is not a new concept. Formany years, businesses have been using computer systems to automate various aspects of their operations, including inventory management. These systems can help to improve the efficiency and effectiveness of retail stores by reducing the need for manual tasks, such as manually tracking inventory or generating sales reports.

In recent years, the use of computer systems for managing and monitoring retail stores has become even more prevalent. These technologies have enabled businesses to develop more sophisticated and feature-rich systems that can handle a wide range of tasks and provide real-time insights into the performance of the store.

The "Medical Store management System" project is likely part of this trend, as it is described as a computer-based system that is intended to improve the productivity of a retail store throughautomated functions. By automating various tasks and providing real-time data and insights,the "Medical Store" system is expected to help the store operate more efficiently and effectively.

1.2 Existing System

Before we analyze the design of the proposed system, we need to carefully highlight the problems of the existing system so as to avoid recurrence. This analysis serves as a pointer on how to embark on building the proposed system that will help the store retailer provide optimal inventory management by monitoring the medicines movement. The problems of the current system should be outlined.

Below are some of the problems associated with the existing system.

1. Early days Stores are managed manually. It required lot of time to record or to retrieve the details. The employees who have to record the details must perform their job very carefully. Even a small mistake would create a lot of problems.
2. Maintenance of store catalogue and arrangement of the books to the catalogue is very complex task.
3. Too much workload on employees.

1.3 Proposed System

The proposed system will manage record of all medicines, price details of medicines, the number of medicines sold, details of medicines and other important details. It deals with the recording and processing data so that the medicine store owners can easily manage the operations.

The advantages are:

- The sellers will register in the Store
- Individually each seller will have his account through which he can access theinformation he needs.
- Purchase dates are maintained carefully.
- medicine details like name, number of capsules totally maintained by store, present available number of medicines, reference medicines are made available.
- Data consistency
- Overcome the old procedures
- Easy information refreshing
- Backup information

Chapter 2

LITERATURE REVIEW

A literature review on Medical Store Management Systems (MSMS) explores existing research, studies, and developments in the field. It provides insights into the current state of MSMS, identifies gaps in knowledge, and highlights emerging trends. Here is a summary of key themes found in the literature:

1. Efficiency and Accuracy in Inventory Management:

Researchers emphasize the importance of efficient inventory management in medical stores. Automation, RFID technology, and real-time tracking systems are identified as key components for improving accuracy and reducing errors in stock management.

2. Technological Integration for Operational Streamlining:

Literature highlights the integration of technology as a driving force behind the modernization of medical stores. Advanced features such as barcode scanning, automated reorder systems, and IoT (Internet of Things) devices contribute to streamlining operations.

3. Regulatory Compliance and Drug Safety:

Ensuring compliance with pharmaceutical regulations and maintaining drug safety are critical considerations. Studies discuss the role of MSMS in automating compliance checks, generating regulatory reports, and implementing measures to prevent the sale of expired or recalled medications.

4. Patient-Centric Approaches and Customer Relationship Management (CRM):

A customer-centric approach is recognized as a key factor in the success of medical stores. CRM functionalities within MSMS are explored for enhancing customer interactions, implementing loyalty programs, and obtaining valuable feedback to improve service quality.

5. Data Security and Privacy Concerns:

As MSMS deals with sensitive patient and pharmaceutical data, literature emphasizes the importance of robust data security measures. Biometric authentication, encryption, and secure access controls are discussed as essential components to safeguard patient information.

6. Telemedicine Integration and Future Trends:

The literature points to the integration of MSMS with telemedicine platforms as a growing trend. This integration facilitates seamless communication between pharmacists and healthcare providers, leading to improved prescription processing and order fulfillment.

7. Analytics and Business Intelligence:

MSMS is recognized as a valuable tool for generating insights through data analytics. Researchers explore the use of analytics for business intelligence, including sales trends, inventory turnover analysis, and predictive modeling to optimize stock levels.

8. Challenges and Adoption Barriers:

Several studies acknowledge challenges in the adoption of MSMS, including resistance to change, initial implementation costs, and the need for training. Understanding and addressing these barriers are identified as crucial for successful system adoption.

In conclusion, the literature on Medical Store Management Systems underscores the transformative impact of technology on the efficiency, compliance, and customer service aspects of medical stores. As the healthcare landscape continues to evolve, MSMS is positioned as a key element in the modernization of pharmaceutical retail operations. Further research is encouraged to explore emerging technologies, evaluate system effectiveness, and address the evolving needs of medical stores in an ever-changing healthcare environment.

Chapter 3

ANALYSIS

3.1 Introduction

System is a collection of an interrelated components that works together to achieve a purpose. System analysis is referred to the systematic examination or detailed study of a system in order to identify problems of the system, and using the information gathered in the analysis stage to recommend improvements or solution to the system.

System analysis is the study of sets of interacting entities, including computer systems analysis. This field is closely related to requirements analysis or operations research. It is also "an explicit formal inquiry carried out to help someone identify a better course of action and make a better decision than he might otherwise have made.

System Analysis is a methodology that involves the application of systematic approaches to collects facts about an existing system with the aim of improving it or replacing it with more efficient system within the context of the available resources. In other words, System analysis can also be viewed as the process of investigating a system, identifying problems and using the information to recommend improvements to the system.

3.2 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz /1.80
- GHz.Installed RAM: 8.00 GB (7.89 GB usable).
- System type: 64-bit operating system, x64-based processor.

3.3 Software Requirements

The software requirements are description of features and functionalities of the system.

- Microsoft SQL Server Management Studio 2018.
- Microsoft Visual Studio 2019 or later with ASP.NET, MVC and other tools
- Web browser(Chrome/Edge/Brave).
- Bootstrap and XAMPP.

3.3.1 VS Code

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js.
Support for additional languages can be provided by freely available extensions on the VS Code Marketplace. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

3.4 General Description

A Medical Store Management System (MSMS) is a specialized software solution designed to streamline and automate the various operational aspects of a medical or pharmaceutical store. This system efficiently manages inventory, sales, purchases, and other critical processes to enhance the overall workflow of the medical store.

3.4.1 Product Perspective

From handling major and minor warehouse operations such as inspection, procurement, put-away, picking, packing, order assembly, and shipping, a Medical Store management system helps in directing or validating every step by recording, capturing inventory movement.

3.4.2 Product Functions

A Medical Store Management Store(MSMS) application enables users to access a centralized system in which various warehouse management duties are controlled through a uniform interface, ensuring warehouse operations are both simple to use and efficient. Following are the key functions of the Medical Store Management system.

- Tracking Inventory
- Labour Management
- Order Processing
- Analytics
- Paperless Documentation
- Reliable Customer Service
- Increases Productivity

3.4.3 User Characteristics

One of the most important steps to start using a warehouse management system is setting upthe user roles in the Medical Store. The common user roles in a Medical Store may be medicine Sales manager, supervisor, picker etc.

- Medicine Inventory
- Supervisor
- Order Picker
- Billing
- Receiver
- Customers

3.4.4 Assumptions and Dependencies

- We assume that the Office personnel do all the data entry based and the correct values obtained from forms and registers.
- We assume that the computers that will use the software will be part of the having proper platform to run it.
- Users with administrator (Admin) access should be careful in deleting or modifying any information knowingly or unknowingly which will lead to inconsistency of the database.
- The end users of this software are assumed to have basic level of computer knowledge i.e. point and click.

3.5 Requirements

A requirement in software engineering is a feature of new software that someone either wants, needs or commands.

It describes what the software does and any limitations it should have.

3.5.1 Functional Requirements

- Creation of New Record: This function creates a record for a new item.
- Deletion of Record: This function is used to delete the existing record of any items.
- Updating in Record: This function updates the information in a record of any items.
- Display of Data in Record: This function displays the record of the items.
- Searching a Record: This function searches the record of the items.

3.5.2 Non-Functional Requirements

- Security: Only authorized users can access the system with email and password.
- Performance: Easy tracking of records and updating can be done.
- User Friendly: The system is very
- interactive.Maintainability: Easy to maintain
  the records.

3.5.3 Software Requirements

- Microsoft SQL Server Management Studio 2018.
- Microsoft Visual Studio 2019 or later with ASP.NET , MVC and other required toolsinstalled.
- Web browser(Chrome/Edge/Brave).
- Bootstrap and XAMPP.

3.5.4 Hardware Requirements

- Processor: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz /1.80
- GHz.Installed RAM: 8.00 GB (7.89 GB usable).
- System type: 64-bit operating system, x64-based processor.

Chapter 4

SYSTEM DESIGN

4.1 Introduction

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

Elements of a System

- **Architecture -** This is the conceptual model that defines the structure, behavior and more views of a system. We can use flowcharts to represent and illustrate the architecture.
- **Modules -** This are components that handle one specific tasks in a system. A combination of the modules makes up the system.
- **Components -** This provides a particular function or group of related functions. They are made up of modules.
- **Interfaces -** This is the shared boundary across which the components of the system exchange information and relate.
- **Data -** This the management of the information and data flow.

4.1 System Architecture and diagram

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages

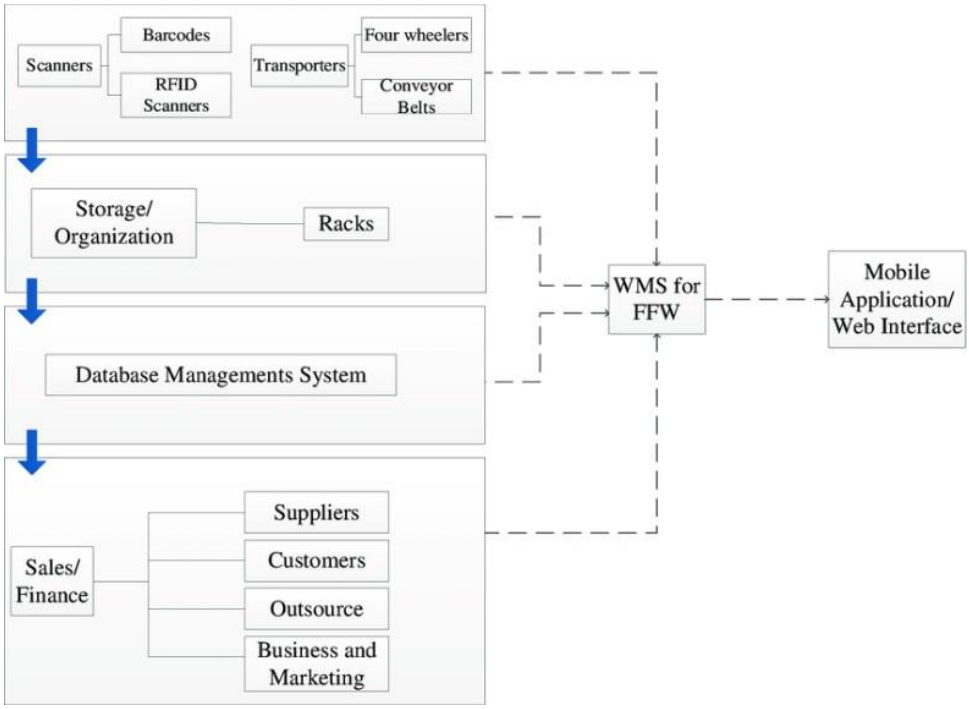

***Figure 4.1**:General Architecture of Management System*

The general architecture of a stock management system consists of a stock management software that manages the incoming and outgoing stock by communicating to a centralized private database.

The architecture of a Medical Store Management System (MSMS) is designed to provide a robust and scalable framework for the efficient management of various processes within a medical or pharmaceutical store.

4.1.1 Three-tier Architecture

Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.

The chief benefit of three-tier architecture is that because each tier runs on its own infrastructure, each tier can be developed simultaneously by a separate development team, and can be updated or scaled as needed without impacting the other tiers.

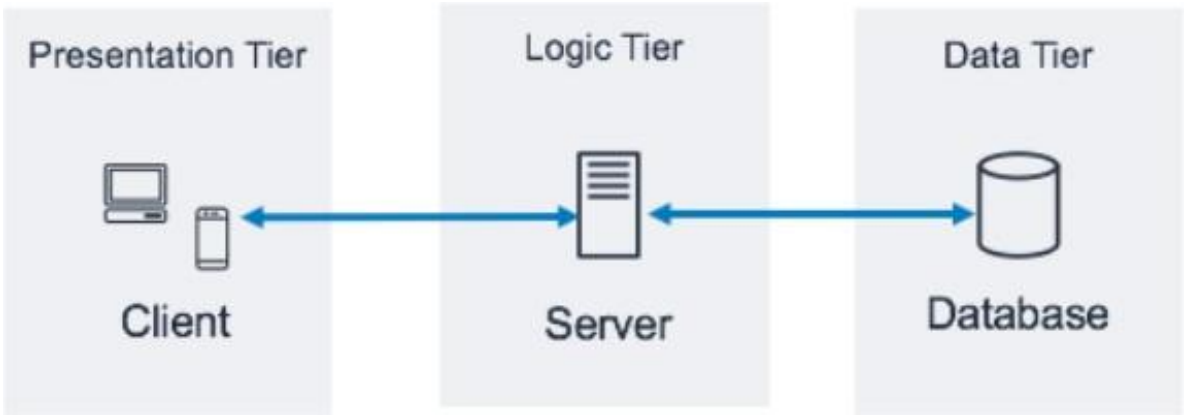

***Figure 4.2**:Three-tier architecture*

4.1.2 Data tier

The data tier, sometimes called database tier, data access tier or back-end, is where the information processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix or Microsoft SQL Server, or in a NoSQL Database server such as Cassandra, CouchDB or MongoDB.

4.2.3 Application Tier

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed - sometimes against other information in the data tier - using business logic, a specific set of business rules. The application tier can also add, delete or modify data in the data tier.

The application tier is typically developed using Python, Java, Perl, PHP or Ruby, and communicates with the data tier using API calls.

4.2.4 Presentation Tier

The presentation tier is the user interface and communication layer of the application, where the end user interacts with the application. Its main purpose is to display information to and collect information from the user. This top-level tier can run on a web browser, as desktop application, or a graphical user interface (GUI), for example. Web presentation tiers are usually developed using HTML, CSS and JavaScript. Desktop applications can be written in a variety of languages depending on the platform.

4.2.5 System Architecture

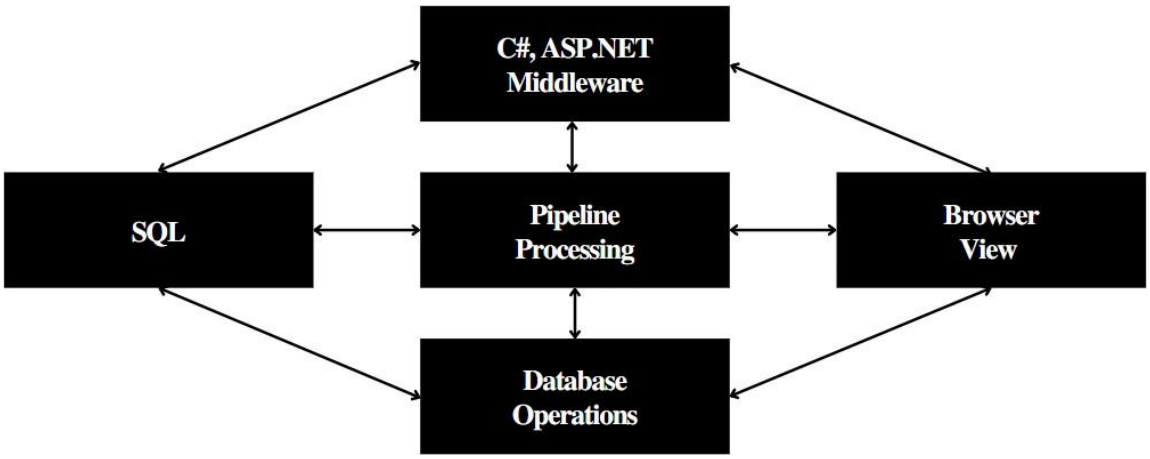

***Figure 4.3**:System Architecture*

4.2.5.1 Browser View

The browser or the client-side component or the front-end component is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.

4.2.5.2 Database Operations

Each database operation accesses one relational database. The data owner (typically a database administrator who has responsibility for that database) creates the database operation as an entry in the data catalog, giving it a name and a definition. The definition can be any SQL statement the database accepts (INSERT, SELECT, UPDATE, DELETE, CALL). When you define a database operation that contains a SQL statement and grant access to that database operation, you are granting rights to run that statement in the database. Database operations return SQL result sets. They can also accept and return parameters. For example, you could set up a statement that returned order details given a specific order number.

4.2.5.3 Middleware

Middleware is software that different applications use to communicate with each other. It provides functionality to connect applications intelligently and efficiently so that you can

innovate faster. Middleware acts as a bridge between diverse technologies, tools, and databases so that you can integrate them seamlessly into a single system. The single system then provides a unified service to its users. For example, a Windows frontend application sends and receives data from a Linux backend server, but the application users are unaware of the difference.

4.2.5.4 Pipeline Processing

Pipeline processing refers to overlapping operations by moving data or instructions into a conceptual pipe with all stages of the pipe performing simultaneously. For example, while one instruction is being executed, the computer is decoding the next.

Pipelining is a technique of breaking a sequential process into small fragments or sub-operations. The execution of each of these sub-procedure takes place in a certain dedicated segment that functions together with all other segments. The pipeline has a collection of processing segments which helps the flow of binary information.

The internal working in a pipeline is such that the outcome of one segment is conveyed to the next segment in the pipeline until the desired result is obtained. The outcome is acquired after the information is developed through all segments.

The term “pipeline” indicates that the flow of information takes place in parallel. Pipelining defines the temporal overlapping of processing. The overlapping of processing is done by relating a register with every segment in the pipeline. The registers help in providing isolation between each segment so that every segment can work on distinct data simultaneously.

Chapter 5

DETAILED DESIGN

Detailed design is the phase where the design is refined and plans, specifications and estimates are created. Detailed design will include outputs such as 2D and 3D models, P & ID's, cost build up estimates, procurement plans etc. This phase is where the full cost of theproject is identified.

The detail design phase defines the complete specification of the geometry, materials, and tolerances of all the parts through the provision of detail drawings, assembly drawings, and general assembly drawings.

5.1 High Level Design

High-level design explains the architecture that would be used to develop a system. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

5.1.1 Design Considerations

When designing a piece of software, there are many aspects to be considered. What is expected from the software will define the importance of each point of consideration.

- **Usability:** The end-user must be able to easily interact with the UI. Location of UI elements, colors and default value for parameters are factors to be considered.

- **Extensibility**: When adding new capabilities to the software, it should be done without significant changes to the architecture.

- **Compatibility:** How is the software going to interact with other products. For example backward-compatibility with an older version of itself

- **Modularity:** Various components of the software could be implemented and testedindependently.

- **Maintainability:** How easy is to apply bug fixes and modifications to the software.

5.2.1.1 Assumptions and Dependencies

All the requirements are assumed to be met before integrating the SMS solution. Any underlying assumptions will be verified and tested if they do not adhere to the requirements.

5.2.1.2 Mode of Operation of a System

There are two modes of operation in the operating system to make sure it works correctly. These are user mode and kernel mode.

5.2.1.3 User Operations

All the users of the system including the Admin, Customer and Retailer have specific operations that they are permitted to perform in the application. They are:

- **Create Record:** When you click on the create button, the form you configured when creating a new database will appear on your screen. Fill the fields with the info you need and click create record. After that, you will see a new line in your database with the information you just added.
- **Edit Record:** Edit will update an existing record. Editing a record is a full object replacement, which means the data you update the record with will replace the existing data.
- **View Record:** It contains a set of predefined SQL queries to fetch data from the database. It can contain database tables from single or multiple databases as well. In the following image, you can see the VIEW contains a query to join three relational tablesand fetch the data in a virtual table.
- **Delete Record:** The Delete command in SQL is a part of the Data Manipulation Language, a sub-language of SQL that allows modification of data in databases. This command is used to delete existing records from a table. Using this, you can either delete specific records based on a condition or all the records from a table.
- **Login/Signup:** It allows the user to login or singup to get access to the application and perform the above listed operations on the database.

5.1.2 Data flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

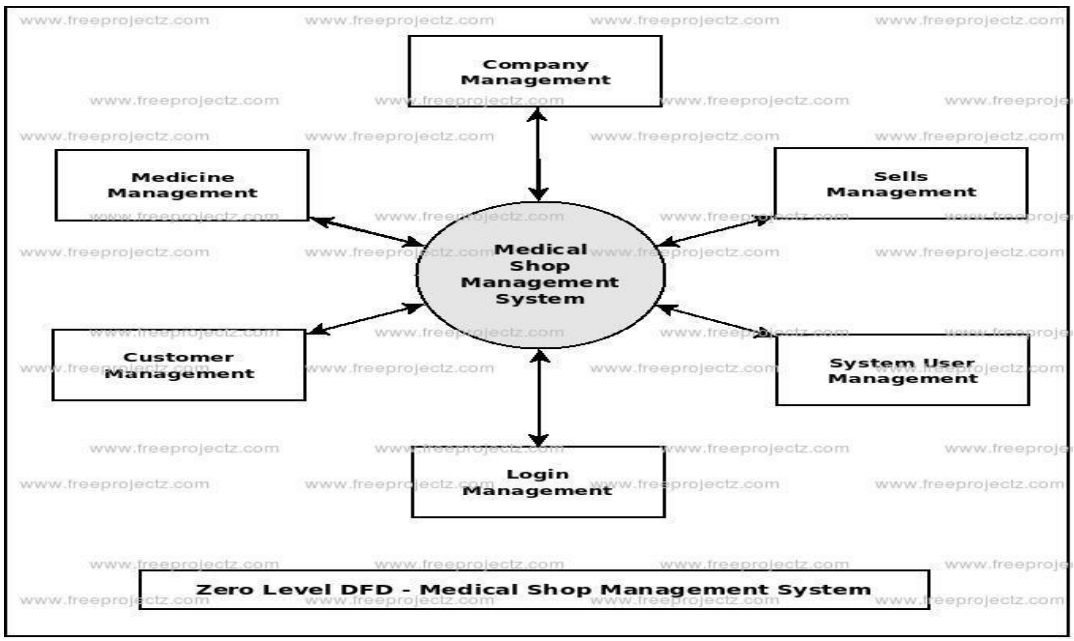

***Figure 5.1**:Data Flow Diagram*

5.2 Low Level Design

Low-level design is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms.

5.2.1 Use Case Diagrams

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models thetasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

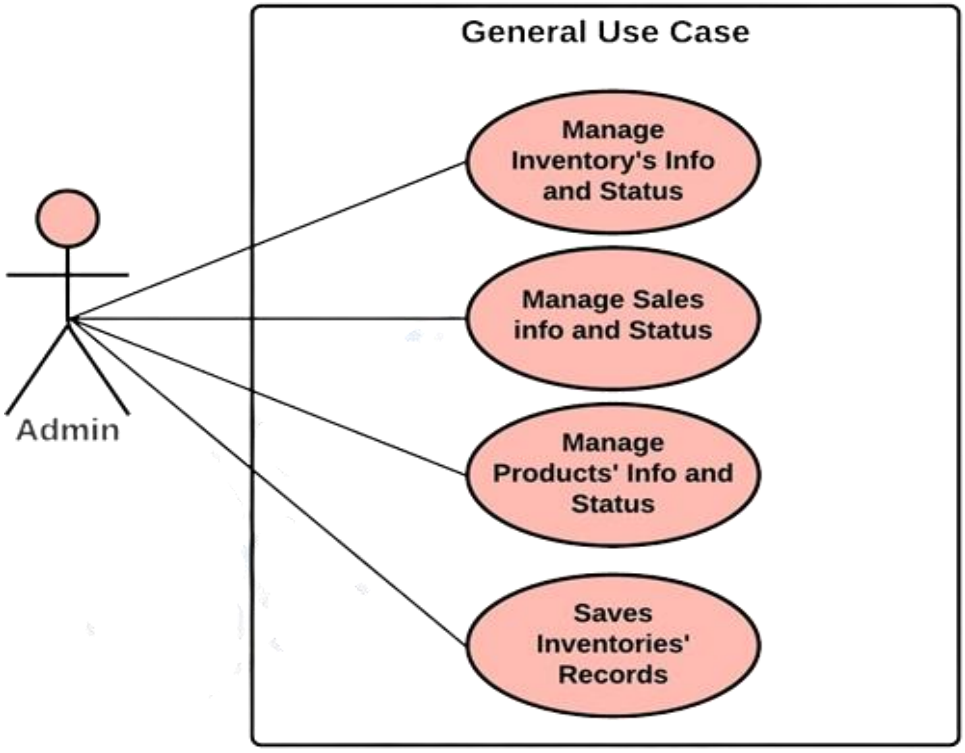

***Figure 5.2**:Use Case Diagrams*

5.2.2 Sequence diagram

A sequence diagram or system sequence diagram shows process interactions arranged in timesequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

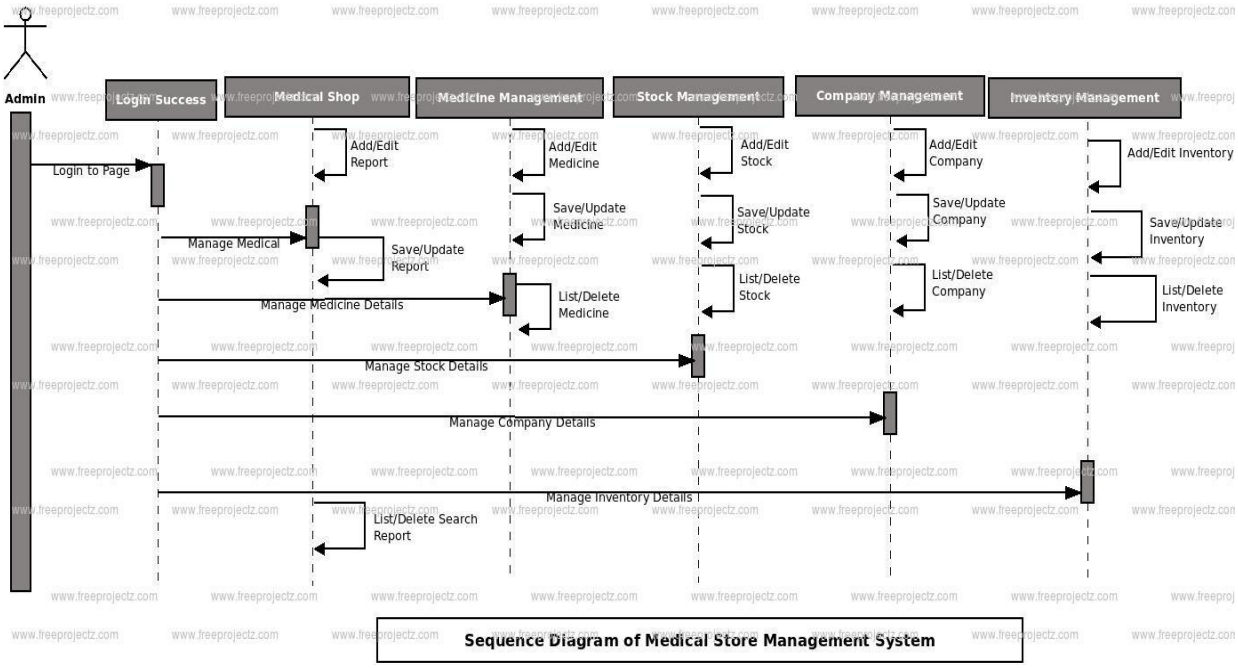

***Figure 5.3**:Sequence Diagram*

5.2.3 Class Diagrams

A sequence diagram or system sequence diagram shows process interactions arranged in timesequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

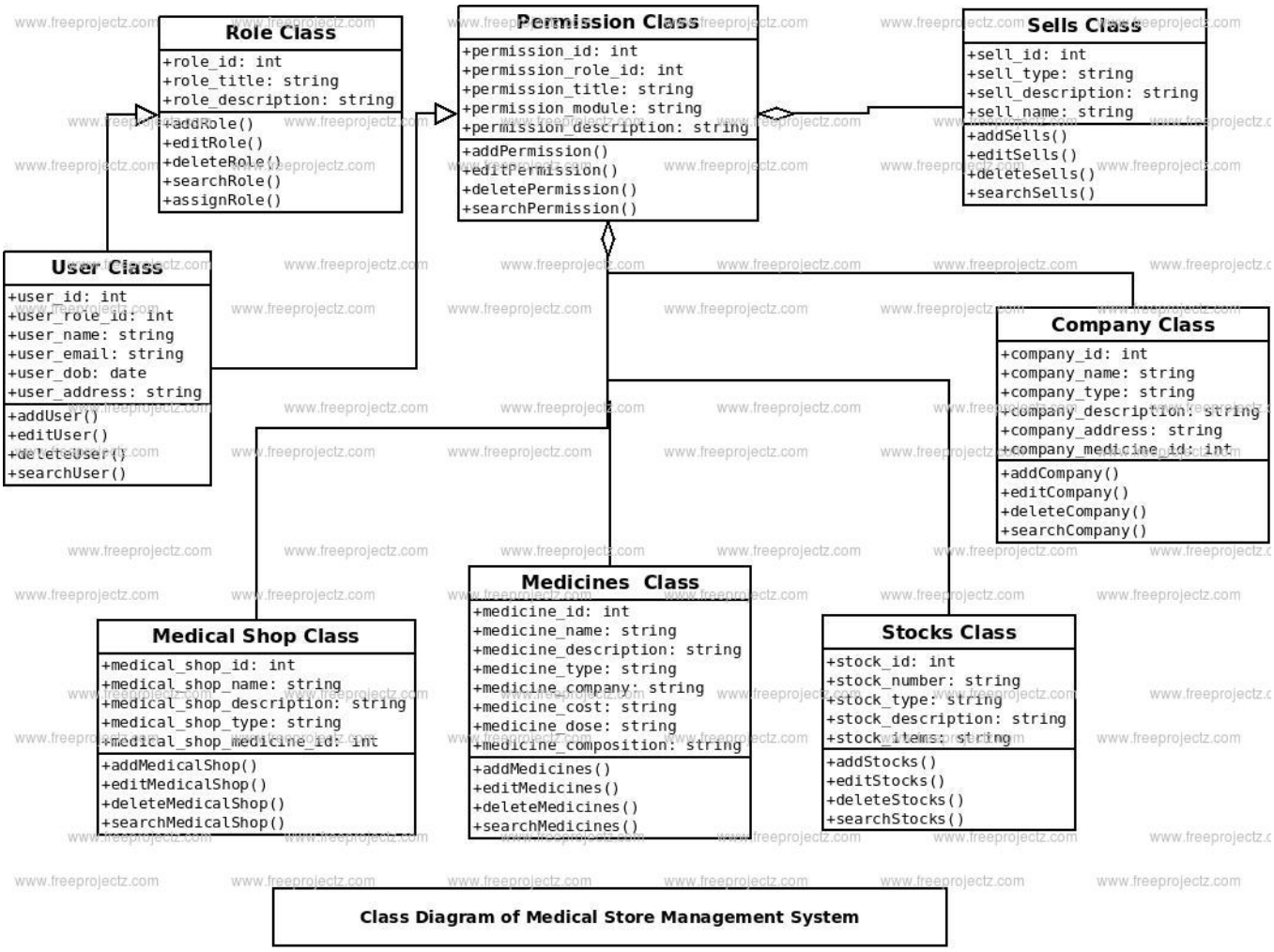

***Figure 5.4**:Class Diagram*

The implementation of SMS (StockMaze) has five modules as shown above. The fivemodules explain the database schema architecture and the fields. The five modules are:

- Product Module
- Customer Module
- Seller Module
- Invoices Module

Chapter 6

IMPLEMENTATION DETAILS

6.1 Introduction

Implementation is the execution or practice of a plan, a method or any design, idea, model, specification, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking for something to actually happen.

6.2 Overview of system implementation

The complexity of a WMS implementation varies with each business.

The physical dimensions and characteristics of each item to be stored in the warehouse should be collected and entered into the new system.

6.2.1 Usability Aspect

Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.
Usability is defined by 5 quality components:

- Learnability
  Efficiency
  Memorability
  Errors
  Satisfaction

6.2.2 Technical Aspect

A technical feasibility study or technical aspects of a project is an in-depth examination of tech factors related to the intended project. It is used to assess the overall performance, ease of learning, deployment, support, compatibility, scalability and licensing. It includes careful study on topics such as:

1. Hardware and Software Components.
2. Compatibility with other IT systems.
3. Capabilities of your engineering team.
4. Technical Risks and Constraints.

6.3 Implementation Support

This section covers the software requirements needed to deploy the project. It also covers the process of installation and deployment of the software services to put the project in action. The software services required and used in this project are mentioned below as follows:

6.3.1 Installation of Microsoft Visual Studio

1. Go to the Visual Studio website (https://visualstudio.microsoft.com/).
2. Click the "Download Visual Studio" button.
3. Select the edition of Visual Studio you want to install. There are several different editions available, including Community (free), Professional, and Enterprise.
4. Click the "Continue" button.
5. Sign in with a Microsoft account or create a new one if you don't have one.
6. Select the workloads you want to install. Workloads are groups of related tools and features that you can choose to install with Visual Studio. For example, the "ASP.NET and web development" workload includes tools for building web applications.
7. Click the "Install" button to begin the installation process.
8. Follow the prompts to complete the installation.

6.3.2 XAMPP

1. XAMPP, short for Cross-Platform Apache, MySQL, PHP, and Perl, is a popular opensource web server solution designed to simplify web development and testing environments.

2. It bundles together essential web development tools like the Apache web server, MySQL database, PHP scripting language, and Perl programming language into a single easy-to-install package.

3. XAMPP is available for various operating systems, including Windows, macOS, and Linux, making it a versatile choice for developers looking to set up a local web server environment for testing and developing web applications.

4. Its user-friendly interface and pre-configured components make it a valuable tool for both beginners and experienced developers working on web projects

6.3.3 Technologies, Languages and Framework used

- **Html:** HTML (HyperText Markup Language) is a markup language used to structure and format content on the web. It uses a system of tags to define the structure and layout of a document, and allows for the creation of links to other web pages and external sources.

- **CSS:** Cascading Style Sheets (CSS) is a stylesheet language used for describing the look and formatting of a document written in HTML. It is used to control the appearance of web pages, including colors, layout, and fonts. CSS is a key part of the web development process, as it allows developers to separate the content of a website (written in HTML)from its visual presentation.

- **JavaScript** is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly. Advantages are: Less server interaction, immediate feedback to the visitors, increased interactivity and richer interface

- **Bootstrap:** Bootstrap is a free and open-source front-end framework for developing responsive, mobile-first websites and web applications. It was developed by Twitter and released in 2011, and has since become one of the most popular front-end frameworks in use today. Bootstrap includes a set of pre-designed HTML and CSS templates that provide a basic structure and style for web pages, as well as a set of responsive designfeatures that ensure that a website looks good on any device. It also includes a set of custom JavaScript components, such as modal dialogs, carousels, and accordions, which can be easily added to a website using predefined classes and data attributes.

6.4 Pseudocode

Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creatingan outline or a rough draft of a program. Pseudocode summarizes a program's flow, but excludes underlying details.

6.4.1 Verifying Login Credentials

The following pseudocode shows the functionality of verifying login credentials when trying to access the application on the web.

```
FUNCTION verifyLoginCredentials(username, password):
    DECLARE Database
    Database = LoadUserCredentialsFromDatabase()  // Load user credentials from a secure database

    IF username NOT IN Database:
        RETURN "Invalid Username"  // Username not found in the database

    IF Database[username].password == hash(password):
        RETURN "Login Successful"  // Password matches, login successful
    ELSE:
        RETURN "Invalid Password"  // Password does not match

FUNCTION hash(input):
    // Implement a secure hashing algorithm (e.g., bcrypt, SHA-256) to hash the password
    // This step is crucial for storing and comparing passwords securely
    // Return the hashed value

FUNCTION LoadUserCredentialsFromDatabase():
    // Connect to the database and retrieve the user credentials
    // Store the credentials in a dictionary where keys are usernames, and values are password hashes
    // Example: {"user1": {"password": hash("pass123")}, "user2": {"password": hash("securePass")}}
    // Return the dictionary

// Example Usage:
usernameInput = GetUsernameInputFromUser()
passwordInput = GetPasswordInputFromUser()

loginResult = verifyLoginCredentials(usernameInput, passwordInput)

IF loginResult == "Login Successful":
    DisplayWelcomeMessage()
ELSE:
    DisplayErrorMessage(loginResult)
```

***Figure 6.1**:Function to Verify Login Credentials*

6.4.2 Login/Signup Functionality

The following pseudocode shows the login and signup feature that is implemented for creating an account which will be stored in the database.

```
FUNCTION main():
    WHILE True:
        DisplayMainMenu()
        choice = GetUserChoice()
        IF choice == 1:
            // Login
            usernameInput = GetUsernameInputFromUser()
            passwordInput = GetPasswordInputFromUser()
            loginResult = verifyLoginCredentials(usernameInput, passwordInput)
            IF loginResult == "Login Successful":
                DisplayWelcomeMessage(usernameInput)
                // Perform actions for a logged-in user
            ELSE:
                DisplayErrorMessage(loginResult)
        ELSE IF choice == 2:
            // Signup
            usernameInput = GetUsernameInputFromUser()
            passwordInput = GetPasswordInputFromUser()
            IF IsUsernameAvailable(usernameInput):
                // Username is available, proceed with signup
                AddUserToDatabase(usernameInput, hash(passwordInput))
                DisplaySuccessMessage("Account created successfully!")
            ELSE:
                DisplayErrorMessage("Username already taken. Please choose another.")
        ELSE IF choice == 3:
            // Exit the program
            DisplayGoodbyeMessage()
            BREAK
        ELSE:
            DisplayErrorMessage("Invalid choice. Please choose again.")
FUNCTION verifyLoginCredentials(username, password):
FUNCTION hash(input):
FUNCTION IsUsernameAvailable(username):
    DECLARE Database
    Database = LoadUsernamesFromDatabase()  // Load existing usernames from a secure database
    RETURN username NOT IN Database
FUNCTION AddUserToDatabase(username, hashedPassword):
    // Connect to the database and add the new user's information
    // Store the username and hashed password securely
    // Example: {"user1": {"password": hash("pass123")}, "user2": {"password": hash("securePass")}}
```

***Figure 6.2**: Login/Signup Functionality*

6.4.3 Class Declaration

The following pseudocode shows the declaration of one of the class models.

```
CLASS Medicine:
     DECLARE name
     DECLARE brand
     DECLARE dosage
     DECLARE quantityInStock
     DECLARE pricePerUnit
     // Constructor
     FUNCTION __init__(name, brand, dosage, quantityInStock, pricePerUnit):
          SET self.name = name
          SET self.brand = brand
          SET self.dosage = dosage
          SET self.quantityInStock = quantityInStock
          SET self.pricePerUnit = pricePerUnit
     // Methods
     FUNCTION updateStock(quantity):
          // Update the quantity of the medicine in stock
          SET self.quantityInStock += quantity
     FUNCTION calculateTotalPrice(quantity):
          // Calculate the total price for a given quantity of the medicine
          RETURN self.pricePerUnit * quantity
     FUNCTION displayMedicineInfo():
          // Display information about the medicine
          PRINT "Name:", self.name
          PRINT "Brand:", self.brand
          PRINT "Dosage:", self.dosage
          PRINT "Quantity in Stock:", self.quantityInStock
          PRINT "Price Per Unit:", self.pricePerUnit
// Example Usage:
medicine1 = Medicine("Paracetamol", "Generic", "500mg", 100, 0.5)
medicine1.displayMedicineInfo()
medicine2 = Medicine("Ibuprofen", "PainRelief", "200mg", 50, 1.0)
medicine2.displayMedicineInfo()
```

***Figure 6.3**:Declaration of Customer Class*

6.4.4 Html Structure of Home Page

The following pseudocode shows the Html structure of the homepage as an overview.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Medical Store Management System</title>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#home">Home</a></li>
            <li><a href="#inventory">Inventory</a></li>
            <li><a href="#sales">Sales</a></li>
            <li><a href="#reports">Reports</a></li>
            <li><a href="#account">Account</a></li>
        </ul>
    </nav>
    <!-- Main Content -->
    <div class="container">
        <section id="welcome-section">
            <h1>Welcome to the Medical Store Management System</h1>
            <p>Efficiently manage your medical store with our comprehensive system.</p>
        </section>
        <section id="store-info">
            <h2>Medical Store Information</h2>
            <p><strong>Store Name:</strong> Your Medical Store</p>
            <p><strong>Location:</strong> [Your Store Location]</p>
            <p><strong>Contact:</strong> [Your Contact Information]</p>
        </section>
        <section id="quick-links">
            <h2>Quick Links</h2>
            <ul>
                <li><a href="#view-inventory">View Inventory</a></li>
                <li><a href="#add-medicine">Add New Medicine</a></li>
                <li><a href="#generate-report">Generate Report</a></li>
            </ul>
        </section>
    </div>
    <!-- Footer -->
    <footer>
        <p>&copy; 2023 Medical Store Management System. All rights reserved.</p>
    </footer>
    <!-- Add your scripts and other body elements here -->
</body>
</html>
```

***Figure 6.4**:Html Structure of Home Page*

Chapter 7

TESTING

7.1 Introduction

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risksof software implementation.

7.2 Levels of Testing

There are generally Four levels of testing. Unit, Integration, System and Acceptance Testing.

7.2.1Unit Testing

This phase involves testing by isolating the smallest pieces of code possible. In this phase, we are going to test user input and login functionality as shown.

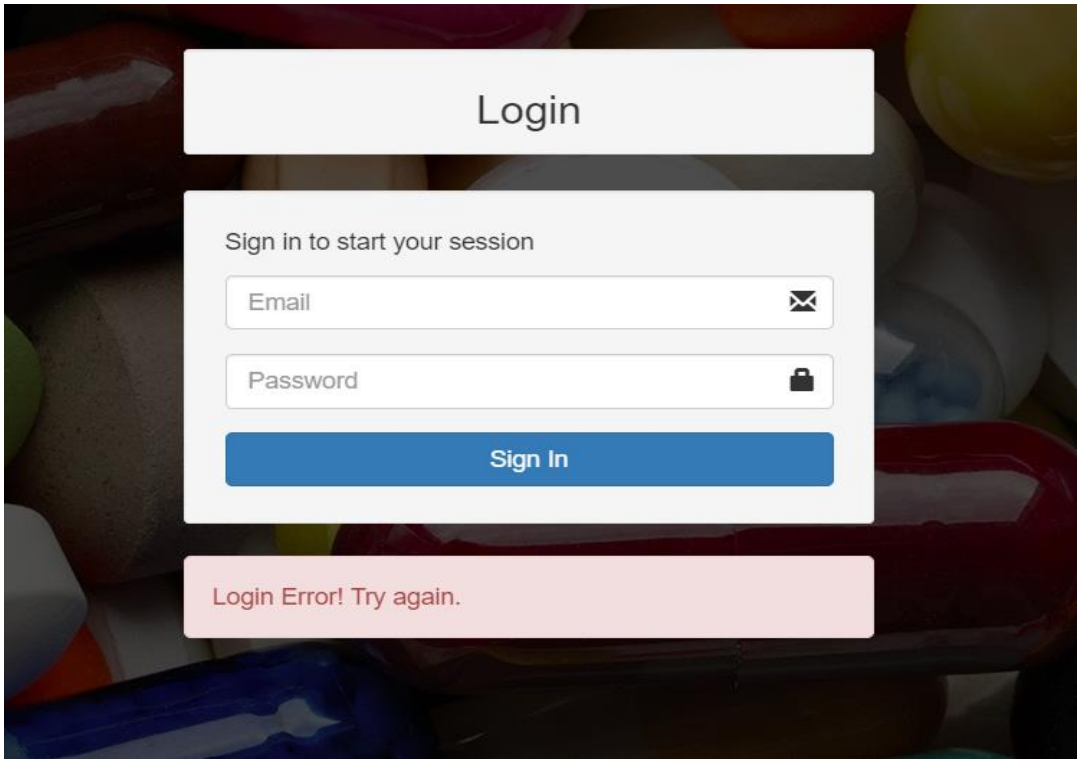

***Figure 7.1**:Unit Testing*

7.2.2 Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance ofa system or component with specified functional requirements.

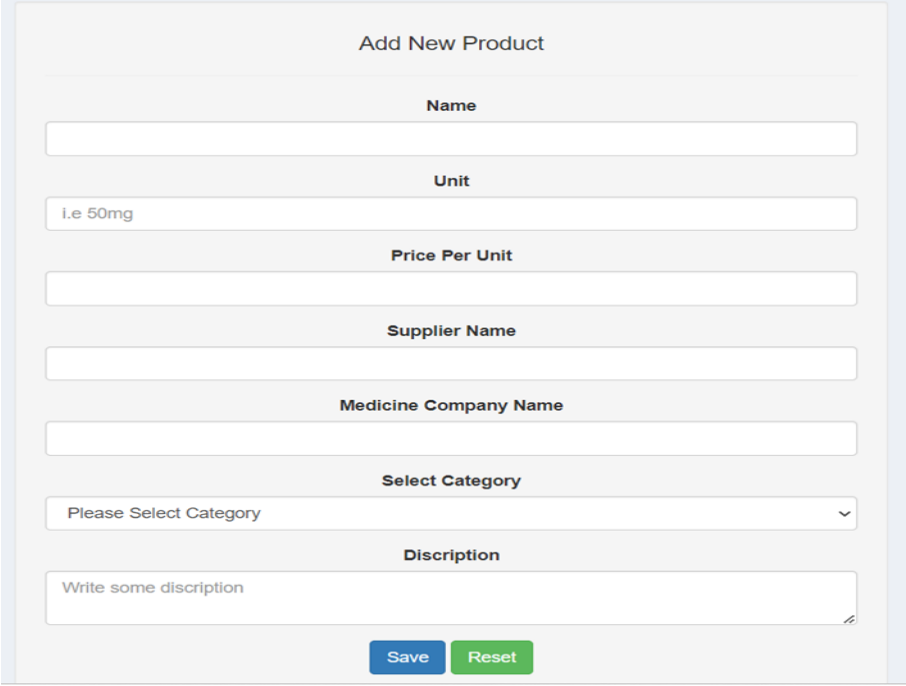

***Figure 7.2**:Integration Testing*

7.2.3 System Testing

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing.

7.2.4 Acceptance Testing

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.

Types of Acceptance Testing:

- User Acceptance Testing (UAT)
- Business Acceptance Testing (BAT)
- Contract Acceptance Testing (CAT)
- Regulations Acceptance Testing
- (RAT)Operational Acceptance Testing
- (OAT)Alpha Testing
- Beta Testing

Chapter 8

RESULTS

8.1 Home Page

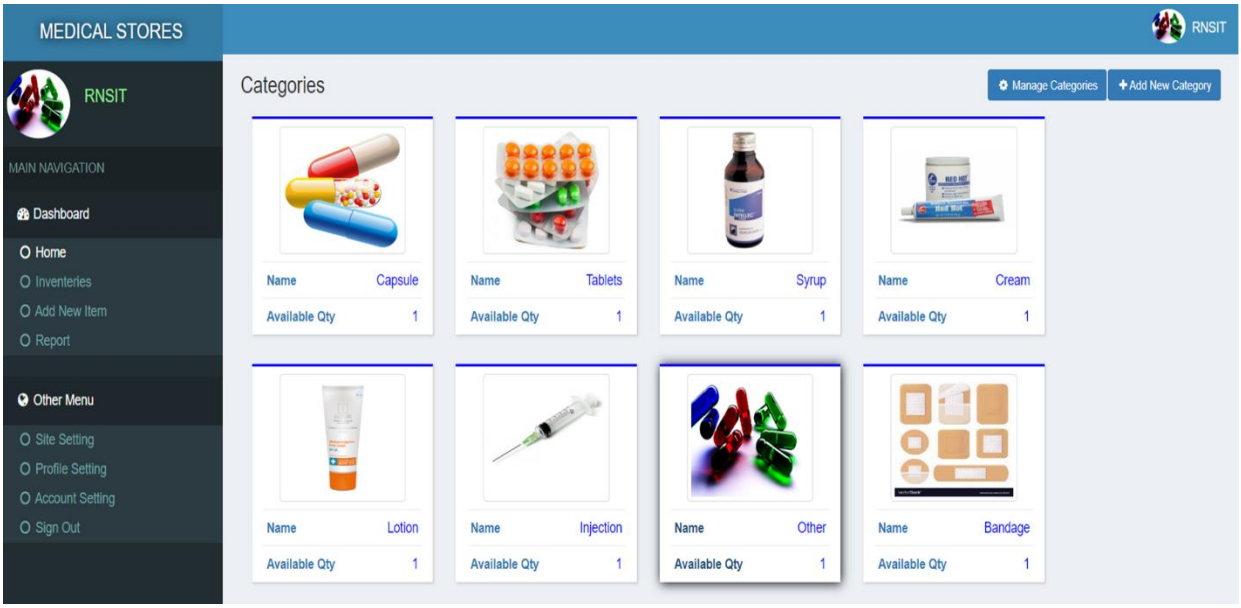

***Figure 8.1**: Home Page*

8.2 Login Page

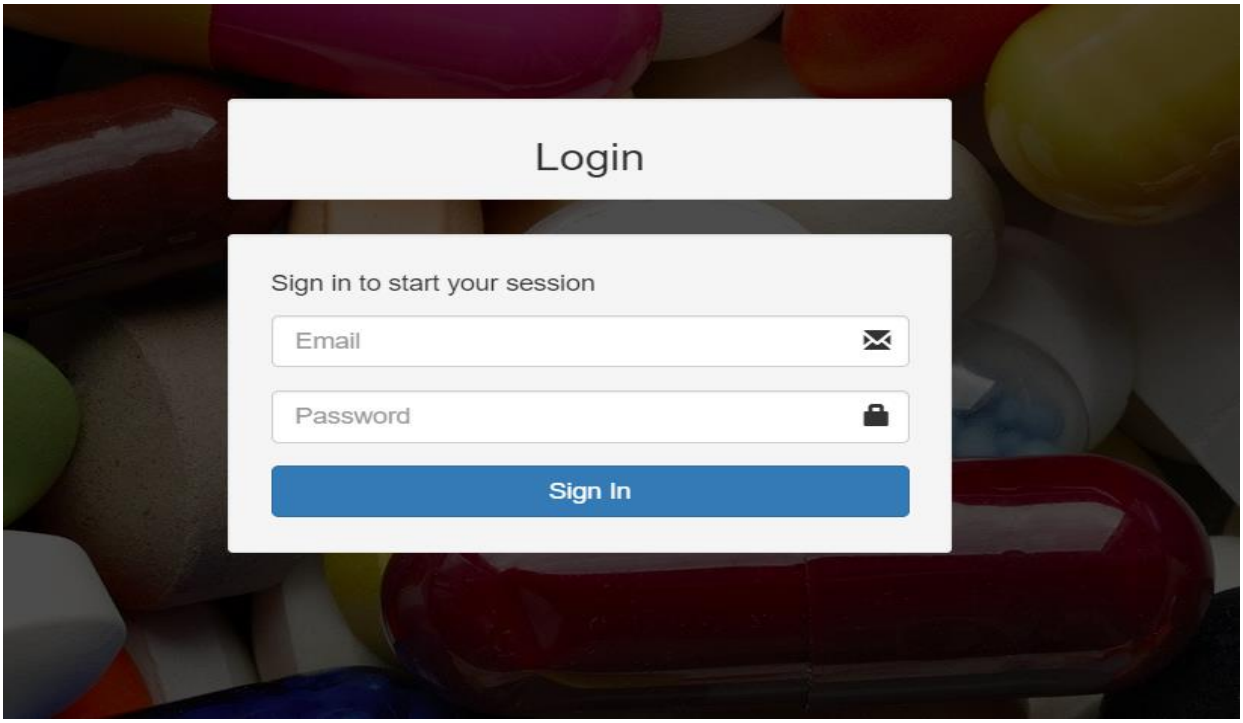

***Figure 8.2**:Login*

8.3 Sold Products Page

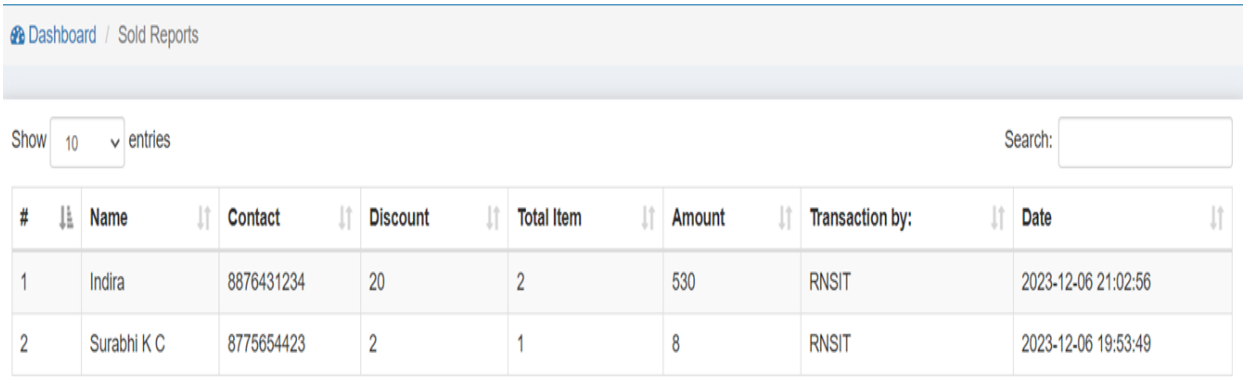

***Figure 8.3**:Sold Products Page*

8.4 Billing page

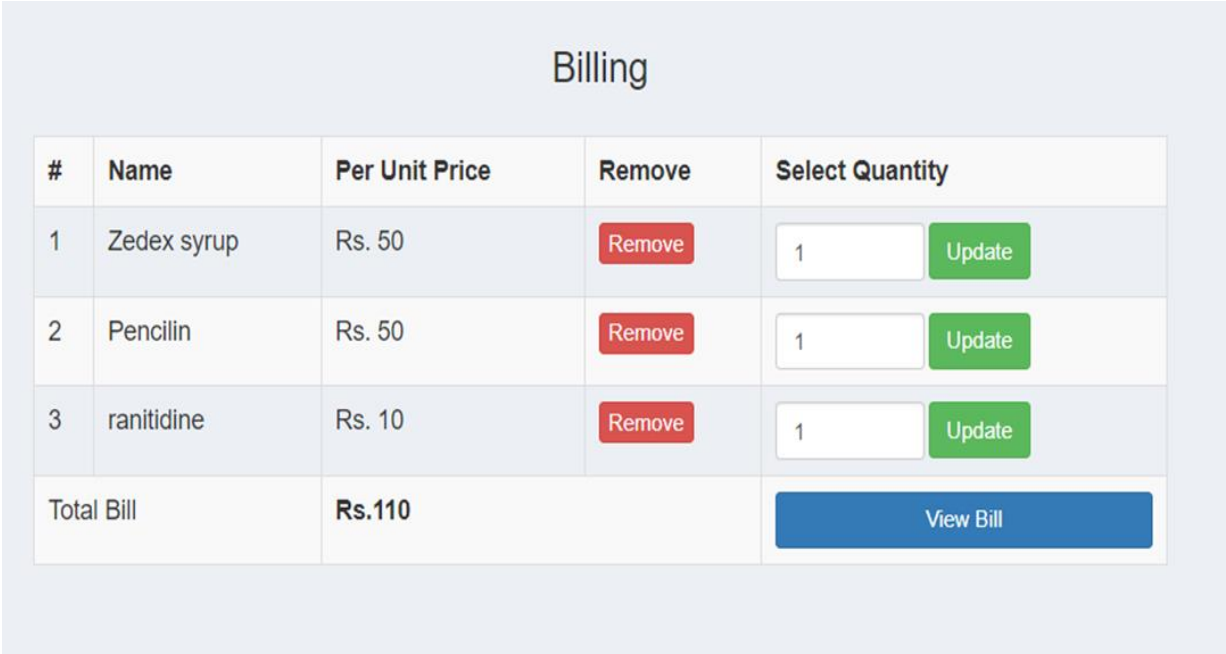

***Figure 8.4**:Billing Page*

8.4 Inventory page

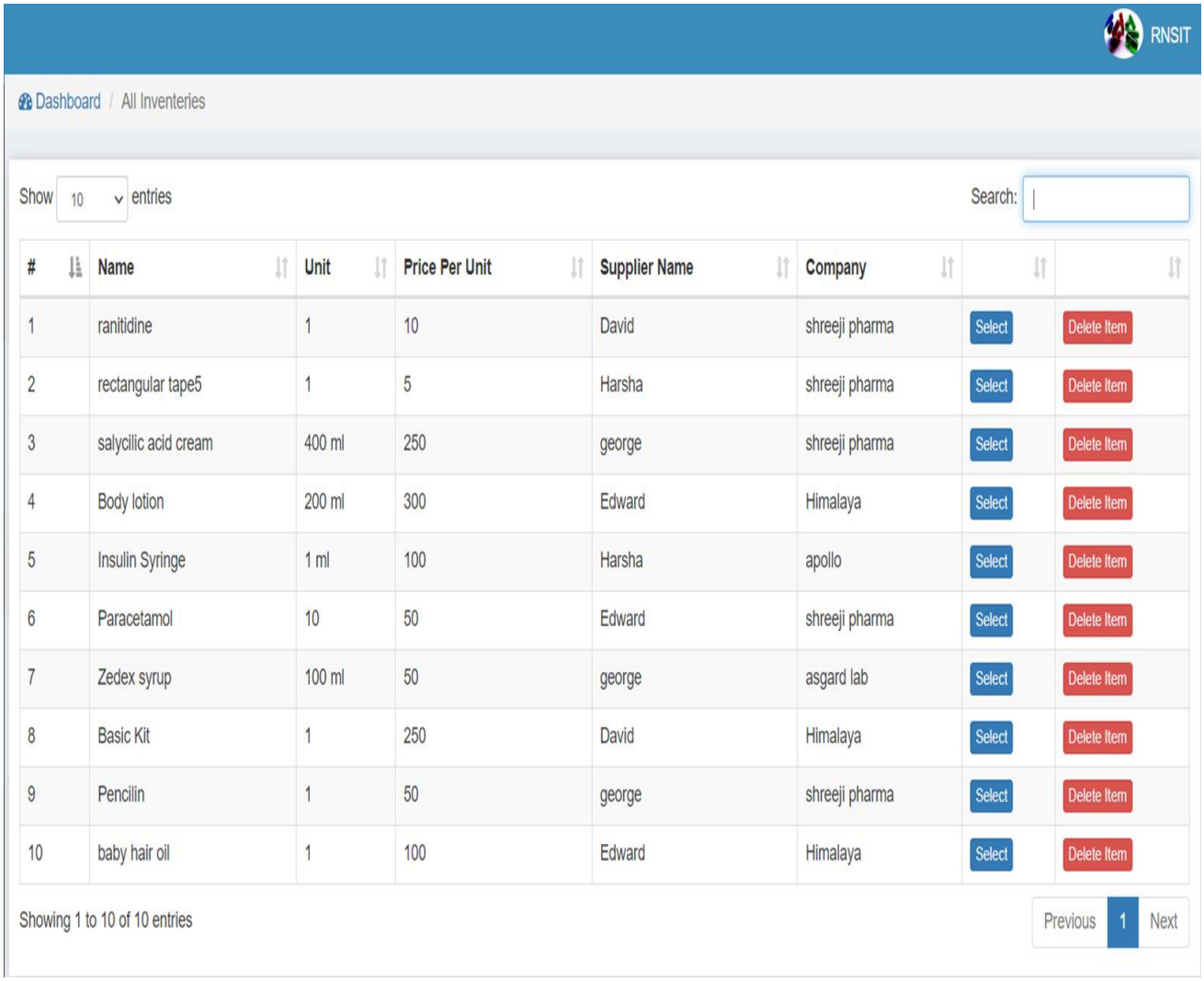

***Figure 8.5**:Inventory list*

Chapter 9
CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusion

In conclusion, the development and implementation of the Medical Store Management System have proven to be a pivotal solution for enhancing the efficiency and organization of our medical store operations. The system successfully addresses several critical aspects of medical store management, including inventory control, sales tracking, and report generation.

The introduction of a centralized system has significantly reduced manual errors in inventory management, ensuring accurate and up-to-date information on the availability of medicines. This not only streamlines the workflow but also minimizes the risk of stockouts and overstock situations. The automation of sales tracking has improved transaction processing speed and accuracy, providing better customer service and satisfaction.

Moreover, the system's reporting capabilities offer valuable insights into sales trends, stock turnover, and other essential metrics. This data-driven approach facilitates informed decision-making and enables us to optimize our inventory, identify popular medicines, and plan for future requirements effectively.

9.2 Future Enhancements

Some of the features that can be added for betterment and enhancement of the application are as below:

- **Barcode and QR Code Scanning:** Implement barcode and QR code scanning capabilities to expedite the process of adding new items to the inventory, updating stock levels, and enhancing the efficiency of sales transactions.
- **Automatic Reorder and Stock Prediction:** Incorporate machine learning algorithms to analyze historical sales data and automatically generate reorder suggestions, minimizing the risk of stockouts and overstock situations.
- **Online Ordering and E-commerce Integration**: Enable online ordering functionality for customers and integrate with e-commerce platforms to expand the reach of the medical store, providing convenience for customers and opening new avenues for revenue.

REFERENCES

[1] Berry, J.R. (1968), Elements of warehouse layout, International Journal of Production Research 7(2), 105-121.

[2] Ackerman, K.B., and LaLonde, B.K. (1980), Making warehousing more efficient, Harvard Business Review, March-April, 94-102.

[3] Amirhosseini, M.M., and Sharp, G.P. (1996), Simultaneous analysis of products and orders in storage assignment, Manufacturing Science and Engineering ASME 1996 MED-Vol. 4, 803-811.

[4] Ashayeri, J., and Gelders, L.F. (1985), Warehouse design optimization, European Journalof Operational Research 21(3), 285-294.

[5] Elliott, M. (2000), Warehouse management system software, IIE Solutions 32(9), 42-46.

[6] https://www.wikipedia.org/

[7] https://www.github.com/

[8] https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

[9] https://visualstudio.microsoft.com/downloads/