

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



An Internship Project Report

on

“PHARMACY 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

by

Harsha D

1RN19IS061

Hemanth Kumar M

1RN19IS063

Under the Guidance of

Mr. Ravi Kumar S G

Assistant Professor

Dept. of ISE, RNSIT



ESTD. 2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

**Dr. Vishnuvardhan Road, Rajarajeshwari Nagar
post, Channasandra, Bengaluru-560098**

2022 -2023

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan 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 **Pharmacy Management System** has been successfully completed by **Harsha D (1RN19IS061)** and **Hemanth Kumar M (1RN19IS063)**, bonafide students 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 2022-2023. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mr. Ravi Kumar S G

Internship Guide
Assistant Professor
Department of ISE

Dr. R Rajkumar

Internship Co-ordinator
Associate Professor
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. _____

Date: September 16th 2022

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Mr. Harsha D** a student from RNSIT with USN: 1RN19IS061 has successfully completed Internship program in **Full Stack Development** from August 2022 till September 2022.

During this period of internship, Mr. Harsha D worked on various areas of software development including project work titled "**Pharmacy Management System**". He has successfully met the objectives of the internship program and his conduct was found to be satisfactory.

We wish him all the best for future endeavors.



Kiran Gopalakrishna,
Founder & CEO,
TechieAid, Bangalore

Date: September 16th 2022

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Mr. Hemanth Kumar Ma** student from RNSIT with USN: 1RN19IS063 has successfully completed Internship program in **Full Stack Development** from August 2022 till September 2022.

During this period of internship, Mr. Hemanth Kumar M worked on various areas of software development including project work titled "**Pharmacy Management System**". He has successfully met the objectives of the internship program and his conduct was found to be satisfactory.

We wish him all the best for future endeavors.



Kiran Gopalakrishna,
Founder & CEO,
TechieAid, Bangalore

DECLARATION

We, **Harsha D [USN: 1RN19IS061]** and **Hemanth Kumar M [USN: 1RN19IS063]**, students of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled **Pharmacy Management System** has been carried out by us 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 2022-2023.

Place: Bengaluru

Date:

**Harsha D
(1RN19IS061)**

**Hemanth Kumar M
(1RN19IS063)**

ABSTRACT

Automation is the motto of this fast-moving world, and being able to do anything from anywhere is the luxury this generation has provided. Pharmacy Management system which evolved in Computer management system is used for managing data entries and sales of a particular drug. Pharmacy Management systems incorporate stock purchase, inventory management, employee details, sales records. This project aims at demonstrating the working of these systems with the above-mentioned services highly comparable to their real-world implementations.

The purpose of this Pharmacy Management System project is to improve the maintenance and manipulation of the drugs in the medicals. The pharmacy management system will be used to minimize the time and resource by maintaining the details of the drug systematically so that the data can be used in possible quickest time. While the resource which is minimized are workforce, money, papers, etc. The system is user-friendly and will help the pharmacist. This Pharmacy Management System will reduce the burden on pharmacist and will make the system efficient by providing the more accurate details about drugs in the medical.

ACKNOWLEDGMENT

At the very onset we 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, we 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, we express sincere gratitude to our beloved Principal **Dr. M K Venkatesha**, for providing us all the facilities.

We are 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.

We place our heartfelt thanks to **Mr. Ravi Kumar S G** 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.

We 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.

We also thank our internship coordinators **Dr. R Rajkumar**, Associate Professor, and **Mr. Pramoda R**, Assistant Professor, Department of Information Science and Engineering. We would thank my friends for having supported me with all their strength and might. Last but not the least, We thank my parents for supporting and encouraging me throughout. We have made an honest effort in this assignment.

Harsha D

Hemanth Kumar M

TABLE OF CONTENTS

Abstract	i
Acknowledgment	ii
Contents	iii
List of figures	vii
List of Abbreviations	viii

CONTENTS

1. Introduction	01
1.1 Background	01
1.2 Existing System	01
1.3 Proposed System	02
2. Literature Review	03
3. Analysis	05
3.1 Introduction	05
3.2 Hardware requirements	05
3.3 Software requirement specification	05
3.3.1 VS Code	06
3.3.2 MS Sql	06
3.3.3 Hypertext Markup Language	06
3.3.4 Cascading Style Sheets	07
3.3.5 JavaScript	07
3.3.6 C#	07

4. System Design	08
4.1 Introduction	08
4.2 System Architecture and diagram	08
4.2.1 Three-Tier Architecture	09
4.2.2 Data Tier	10
4.2.3 Application Tier	10
4.2.4 Presentation Tier	10
5. Implementation Details	11
5.1 Introduction	11
5.2 Overview of system implementation	11
5.2.1 Usability aspect	11
5.2.2 Technical aspect	11
5.3 Implementation Support	12
5.3.1 Installation of Microsoft Visual Studio (2019 or later)	12
5.3.2 Installation of SQL Server Management Studio (2018)	12
5.3.3 Installation of ASP.NET Core	13
5.4 Pseudocode	13
5.4.1 Verifying Login Credentials	13
5.4.2 Login/Signup Functionality	14
5.4.3 Class Declaration	14
5.4.4 Html Structure of Home Page	15
6. Testing	16
6.1 Introduction	16
6.2 Types of Testing	16

7. Results	18
7.1 Login Page	18
7.2 Dashboard Page	18
7.3 Inventory Page	19
7.4 Transactions Page	19
7.5 Creation or Adding new record	20
7.6 Update or Edit existing record	20
8. Conclusion and future work	21
8.1 Conclusion	21
8.2 Future Enhancements	21
References	22

List of Figures

Figure 4.1:	General Architecture of pharmacy management system	09
Figure 4.2:	Three-tier architecture	10
Figure 5.1:	Verify Login Credentials	13
Figure 5.2:	Login/Signup Functionality	14
Figure 5.3:	Declaration of Customer Class	14
Figure 5.4:	Html structure of Home Page	15
Figure 7.1:	Admin Login Page	18
Figure 7.2:	Dashboard Page	18
Figure 7.3:	Inventory Page	19
Figure 7.4:	Transactions Page	19
Figure 7.5:	Creation of New Record	20
Figure 7.6:	Update or Edit Existing Record	20

List of Abbreviations

API-Application Programming Interface

ARS-Pharmacy Management System

CGI-Common Gateway Interface

CRS-Computer Reservation System

CSS-Cascading style sheets

DBMS-Database Management System

ER-Entity Relationship

GPL-General Public License

HTML-Hypertext Markup Language

HTTP-Hypertext Transfer Protocol

TB-Terabyte

ID-Identification

JS-JavaScript

MVD-Multi Valued Dependency

PHP-PHP Hypertext Preprocessor

SQL-Structured Query Language

Chapter 1

INTRODUCTION

Pharmacy Management System is a computer-based website developed for managing, monitoring and recording various activities of medical stores. It also helps to overcome the manual pharmacy management issues. It plays a key role in minimizing human strain and errors when it comes to inventory management.

Pharmacy Management System will manage record of all medicines, price details of medicines, the number of medicines sold, brand details of medicines, staff at medical store and other important details. It is additionally used to produce reports of overall productivity of pharmacy. It plays a key role in improving the productivity of medical stores through automated functions.

1.1 Background

The Pharmacy Management System is a boon to the store-keepers at any Pharmacy store. It emerged out in early 2000s in various parts of the world replacing the age old tedious method of Book-Keeping/Ledger system.

The pharmacy management system, also known as the pharmacy information system, is a system that stores data and enables functionality that organizes and maintains the medication use process within pharmacies. These systems may be an independent technology for the pharmacy's use only, or in a hospital setting, pharmacies may be integrated within an inpatient hospital computer physician order entry (CPOE) system. Necessary actions for a basic, functioning pharmacy management system include a user interface, data entry and retention, and security limits to protect patient health information.

Pharmacy computer software is usually purchased ready-made or provided by a drug wholesaler as part of their service. Various pharmacy software operating systems are used throughout the many practice settings of pharmacy across the world.

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 pharmacist provide optimal drug inventory management by monitoring the drug movement and state in the pharmacy. The problems of the current system should be outlined.

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

1. Significant amount of time is allocated for writing the order and the Pharmacist needs to go through the stock balance and make rough estimate for the amount to order based on Figures.
2. The state of drugs in stock is manually checked.
3. Mistake of selling expired drugs to customers.
4. Too much workload on employees.
5. Filling cabinet in the pharmacy with paper record.

1.3 Proposed System

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

The advantages are:

- Easy management of pharmacy
- Managing staff Information
- Managing financial details
- Managing medicines details
- Data consistency
- Overcome the old procedures
- Easy information refreshing
- Backup information

Chapter 2

LITERATURE REVIEW

Managing a medical store is particularly complicated with regard to inventory and record keeping. Medical stores need to make sure they have enough stock of support and cater to the needs of clients. Proper handling of records is essential to ensure the success of all store operations. Medical stores are in the process of searching for an effective and reliable daily help and relief plan. There is a need for electronic intervention in the management of medical stores.

A pharmacist is a medical professional who dispenses medicine to patients according to a prescription ordered by a physician or other clinician. Pharmacists have an in depth knowledge of the chemistry of various medicine and how they react in humans, and also how drugs interact with each other Charles E. Rosenberg, 1980. Pharmacists must accurately measure and package medicine, ensuring its dosage and safety to be administered properly to a patient. While the pharmacist does not typically select or prescribe the medication, the pharmacist educates the patient on how to take the medication and what reactions or problems to be avoided.

As medication experts, pharmacists are concerned with safeguarding the public's health in matters relating to medication distribution and use and disease state management. Pharmacists play a vital role improving patient care through the medicine and information they provide.

Pharmacy Management Software:

In [1], The pharmacy management system kept paper and pen away mostly cause of the way it's Managing a very huge pharmacy with records stored online and on papers which surely seems difficult to keep track of inventories with dignity but this system makes it look easier. A pharmacist has to order drugs to replenish the already diminishing stock. In addition, the ordering of drugs is being carried out manually. A major amount of time is taken for writing the order as the pharmacist needs to check through the stock balance and make an estimate of the amount to order based on Figures. As we know drugs are not supposed to be used after they have expired. This project work will notify the pharmacist about drugs that are near to expire, preventing those drugs from being sold and also providing a solution to the earliest problems.

In [2], The system allows the user to enter the production date and expiration of a particular product or drug at the time of opening the stock and sale function. The system will also provide a report showing a list of expired products after the date specified before the expiration of the product. It also involves direct entry when new drug sets arrive and when drugs leave the pharmacy for a period of time, e.g. on a monthly basis, the pharmacist may want to produce a report on the release and release of drugs from the pharmacy, receiving information about the medication e.g. expiration date, purchase date, residual drug number, drug location at pharmacy. Currently, the hands-on system is used in pharmacies. It requires the pharmacist to personally monitor each drug available at the pharmacy. This often leads to mistakes as the pharmacist's job grows.

In [3], In pharmacy operations, inventory is referred to as the stock of pharmaceutical products retained to meet future demand. Inventory represents the largest asset in pharmacy practice, and its value continues to rise because of the growth in variety and cost of pharmaceutical products. From both financial and operational perspectives, efficient inventory management plays a great role in pharmacy practice. Inventory management aims at reducing procurement and carrying costs, while maintaining an effective stock of products to satisfy customer and prescriber demands. The author reviews methods of inventory management in pharmacy practice, and highlights approaches by which the process of inventory management is evaluated. Factors affecting inventory management, including the role of information technology, are summarized. The author outlines, with recommendations, the impact of inventory mismanagement on patient safety.

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 required tools installed.
- Web browser(Chrome/Edge/Brave).
- Bootstrap and Bootstrap Studio.

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.

3.3.2 MS SQL

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

3.3.3 Hyper Text Markup Language

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. HTML is the standard markup language for creating Web pages. It stands for Hyper Text Markup Language. It describes the structure of a Web page. It consists of a series of elements. Its elements tell the browser how to display the content. Its elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

3.3.4 Cascading Style Sheets

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications. Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. CSS solved that issue. CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file. CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

3.3.5 Java Script

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. Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. 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 interfaces.

3.3.6 C#

C# (pronounced "C-sharp") is a modern, object-oriented programming language developed by Microsoft in the early 2000s. It is a statically-typed language, which means that the type of a value must be known at compile-time, rather than at run-time. C# is used to build a wide range of applications, including Windows desktop applications, mobile apps, and web applications.

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 make 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 a the system exchange information and relate.
- **Data** - This the management of the information and data flow.

4.2 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

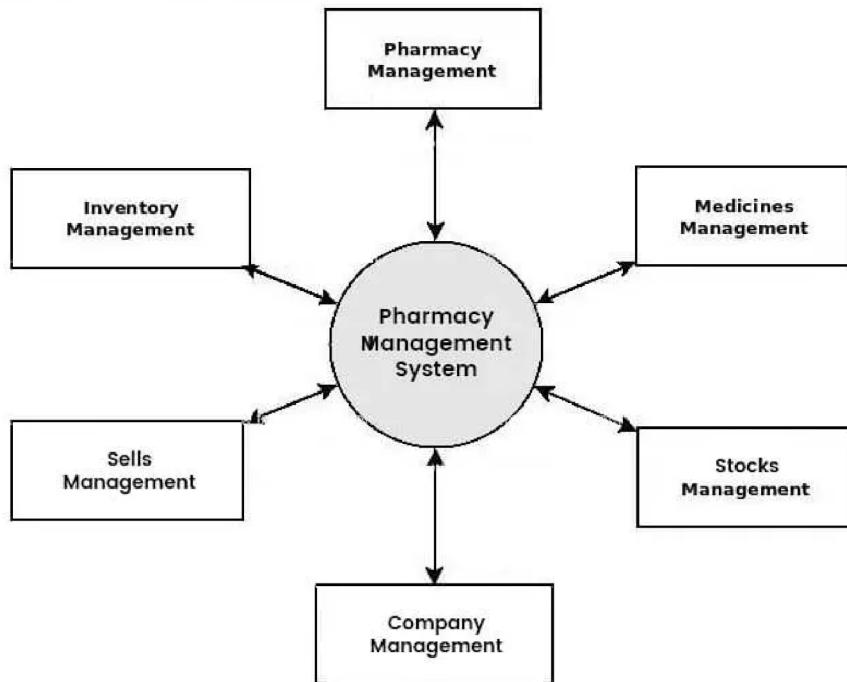


Figure 4.1:General Architecture of Pharmacy Management System

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

The exchange and transaction of inventory stocks is monitored and maintained by the PMS as shown above. The changes and modification incurred in the database are reflected on the user interface in the final step where users can also interact and perform operations with respect to inventory management.

4.2.1 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.

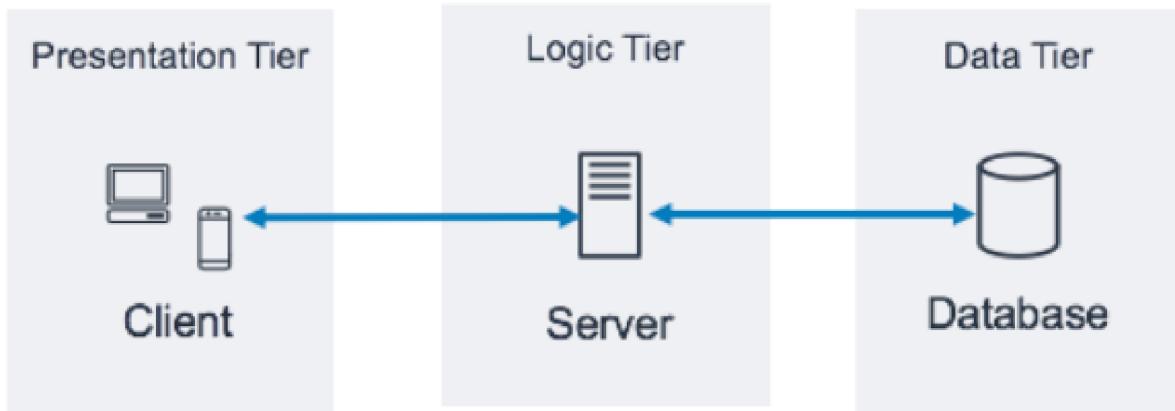


Figure 4.2:Three-tier architecture

4.2.2 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.

Chapter 5

IMPLEMENTATION DETAILS

5.1 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.

5.2 Overview of system implementation

The complexity of a inventory management system implementation varies with each business. The physical dimensions and characteristics of each item to be stored in the inventory should be collected and entered into the new system.

5.2.1 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

5.2.2 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 asses the overall performance, ease of learning, deployment, support, compatibility, scalability and licensing. It includes careful study on topics such as:

- Hardware and Software Components.
- Technical Risks and Constraints.
- Compatibility with other IT systems.
- Capabilities of your engineering team.

5.3 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:

5.3.1 Installation of Microsoft Visual Studio (2019 or later)

- **To install Microsoft Visual Studio, follow these steps:**
 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.

Note that the installation process may take some time, depending on the edition of Visual Studio you are installing and the workloads you have selected.

5.3.2 Installation of SQL Server Management Studio (2018)

- **To install SQL Server Management Studio, follow these steps:**
 1. Download the SQL Server Management Studio installation file from the Microsoft website.
 2. Double-click the downloaded installation file to launch the installation wizard.
 3. Follow the prompts in the installation wizard to complete the installation process.
 4. Once the installation is complete, launch SQL Server Management Studio from the Start menu or by double-clicking the icon on the desktop.
 5. Use your SQL Server login credentials to access the management studio.
 6. If you do not have login credentials, you can create a new login by following the prompts in the SQL Server Management Studio.

5.3.3 Installation of ASP.NET Core

To install ASP.NET Core, you'll need to have the .NET Core SDK installed on your system. The .NET Core SDK includes the runtime, libraries, and tools required to build and run .NET Core applications.

Download and install the .NET Core SDK from the official .NET website:
<https://dotnet.microsoft.com/download>

5.4 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 creating an outline or a rough draft of a program. Pseudocode summarizes a program's flow, but excludes underlying details.

5.4.1 Verifying Login Credentials

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

```
0 references
public ConfirmEmailModel(UserManager<IdentityUser> userManager)
{
    _userManager = userManager;
}

[TempData]
1 reference
public string StatusMessage { get; set; }

0 references
public async Task<IActionResult> OnGetAsync(string userId, string code)
{
    if (userId == null || code == null)
    {
        return RedirectToPage("/Index");
    }

    var user = await _userManager.FindByIdAsync(userId);
    if (user == null)
    {
        return NotFound($"Unable to load user with ID '{userId}'.");
    }

    code = Encoding.UTF8.GetString(WebEncoders.Base64UrlDecode(code));
    var result = await _userManager.ConfirmEmailAsync(user, code);
    StatusMessage = result.Succeeded ? "Thank you for confirming your email." : "Error confirming your email.";
    return Page();
}
```

Figure 5.1: Verify Login Credentials

Above figure shows code snippet for verifying Login credentials entered by the user in the pharmacy management system.

5.4.2 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.

```
async Task<IActionResult> OnPostConfirmationAsync(string returnUrl = null)
{
    returnUrl ??= Url.Content("~/");
    // Get the information about the user from the external login provider
    var info = await _signInManager.GetExternalLoginInfoAsync();
    if (info == null)
    {
        ErrorMessage = "Error loading external login information during confirmation.";
        return RedirectToPage("./Login", new { ReturnUrl = returnUrl });
    }

    if (ModelState.IsValid)
    {
        var user = new IdentityUser { UserName = Input.Email, Email = Input.Email };

        var result = await _userManager.CreateAsync(user);
        if (result.Succeeded)
        {
            result = await _userManager.AddLoginAsync(user, info);
            if (result.Succeeded)
            {
                _logger.LogInformation("User created an account using {Name} provider.", info.LoginProvider);

                var userId = await _userManager.GetUserIdAsync(user);
                var code = await _userManager.GenerateEmailConfirmationTokenAsync(user);
                code = WebEncoders.Base64UrlEncode(Encoding.UTF8.GetBytes(code));
                var callbackUrl = Url.Page(
                    "/Account/ConfirmEmail",
                    pageHandler: null,
                    values: new { area = "Identity", userId = userId, code = code },
                    protocol: Request.Scheme);
            }
        }
    }
}
```

Figure 5.2:Login/Signup Functionality

5.4.3 Class Declaration

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

```
naApp.Models

class Users
{
    int user_id { get; set; }
    String user_name { get; set; }
    String password { get; set; }

    class Medicines
    {
        int med_id { get; set; }
        String med_name { get; set; }
        String category { get; set; }
        double price { get; set; }
        int quantity { get; set; }
        int pack { get; set; }
    }
}
```

Figure 5.3:Declaration of Customer Class

5.4.4 Html Structure of Home Page

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

```
>


<h1>Pharmacy Management System</h1>
    <p style="margin-top: -20px; line-height: 1; font-size: 30px;">Database Management Systems</p>


</div>
<title>
    Pharmacy Management System
</title>
<d>

>

<br><br><br><br>


<form method="post" action="">
        <div id="div_login">
            <h1>Admin Login</h1>
            <center>
                <div>
                    <input type="text" class="textbox" id="uname" name="uname" placeholder="Username" />
                </div>
                <div>
                    <input type="password" class="textbox" id="pwd" name="pwd" placeholder="Password" />
                </div>
                <div>
                    <input type="submit" value="Submit" name="submit" id="submit" />
                    <input type="submit" value="Click here for Pharmacist Login" name="psubmit" id="submit" />
                </div>
            </center>
        </div>
    </form>


```

Figure 5.4:Html Structure of Home Page

Chapter 6

TESTING

6.1 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 risks of software implementation.

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried out in two phases. The first phase is during the software engineering that is during module creation. The second phase is after the completion of the software. this is system testing which verifies that the whole set of programs hanged together.

6.2 Types of Testing

There are numerous types of software testing techniques that can be used to ensure changes to the code work as expected. Not all testing is equal though, and also explore how some testing practices differ.

White Box Testing

In this technique, the close examination of the logical parts through the software is tested by cases that exercise specific sets of conditions or loops. All logical parts of the software were checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and errors resulting from using wrong controls and loops. When the box testing tests all the independent part within a module, the logical decisions on their true and the false side are exercised, all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Black Box Testing

This method enables the software engineer to devise sets of input techniques that fully exercise all functional requirements for a program. Black box testing tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

Alpha Testing

Acceptance testing is also sometimes called alpha testing. Bespoke systems are developed for a single Credit Card. The alpha testing proceeds until the system developer and the Credit Card agrees that the provided system is an acceptable implementation of the system requirements.

Beta Testing

On the other hand, when a system is to be marketed as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered to a number of potential users who agree to use it. The Credit Cards then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Unit Testing

Each module is considered independently. It focuses on each unit of software as implemented in the source code. It is white-box testing.

Integration Testing

Integration testing aims at constructing the program structure while at the same time constructing tests to uncover errors associated with interfacing the modules. Modules are integrated by using the top-down approach.

Validation Testing

Validation testing was performed to ensure that all the functional and performance requirements are met.

System Testing

It is executing programs to check logical changes made in it with intention of finding errors. A system is tested for the online response, the volume of interactions, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

Chapter 7

RESULTS

7.1 Login Page



Figure 7.1 Login Page

The figures 7.1 shows the login page through which the user can login by entering his/her username and password.

7.2 Dashboard Page

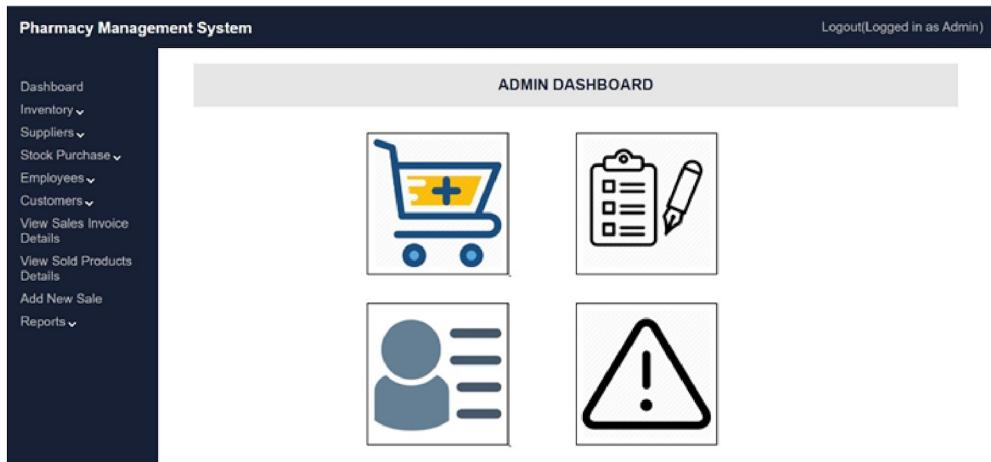


Figure 7.2: Dashboard Page

The figures 7.2 shows the dashboard page through which the user can navigate to frequently used pages.

7.3 Inventory page

MEDICINE INVENTORY						
Medicine ID	Medicine Name	Quantity Available	Category	Price	Location in Store	Action
123001	Dolo 650 MG	650	Tablet	1.00	rack 7	<button>Edit</button> <button>Delete</button>
123002	Panadol Cold & Flu	88	Tablet	2.50	rack 6	<button>Edit</button> <button>Delete</button>
123003	Livogen	25	Capsule	5.00	rack 3	<button>Edit</button> <button>Delete</button>
123004	Gelusil	440	Tablet	1.25	rack 4	<button>Edit</button> <button>Delete</button>
123005	Cyclopam	145	Tablet	6.00	rack 2	<button>Edit</button> <button>Delete</button>
123006	Benadryl 200 ML	35	Syrup	50.00	rack 10	<button>Edit</button> <button>Delete</button>
123007	Lopamide	15	Capsule	5.00	rack 7	<button>Edit</button> <button>Delete</button>
123008	Vitamin C	450	Tablet	5.00	rack 8	<button>Edit</button> <button>Delete</button>

Figure 7.3:Inventory Page

The figure 7.3 displays the current status of the inventory in which each record can be edited or deleted.

7.4 Transactions page

SALES INVOICE DETAILS				
Sale ID	Customer ID	Date and Time	Sale Amount	Employee ID
1	987101	2020-04-15 13:23:03	180.00	4567009
2	987106	2020-04-21 20:19:31	585.00	1
3	987103	2020-04-15 11:23:53	120.00	4567010
4	987104	2020-04-14 18:20:00	955.00	4567006
5	987103	2020-04-21 15:24:43	45.00	1
6	987102	2020-03-11 10:24:43	140.00	4567001
7	987105	2020-04-24 00:25:54	350.00	1
8	987104	2020-04-24 00:47:47	35.00	4567001
12	987103	2020-04-24 19:33:16	60.00	1
13	987104	2020-04-24 21:15:56	62.50	4567001
15	987107	2020-12-04 18:39:46	420.00	1
16	987106	2020-12-04 18:52:21	30.00	1

Figure 7.4:Transactions Page

The figure 7.4 shows the details of the transactions taken place.

7.5 Creation or Adding new record

The screenshot shows the 'Pharmacy Management System' interface. On the left, a dark sidebar lists various menu items: Dashboard, Inventory (selected), Add New Medicine, Manage Inventory, Suppliers, Stock Purchase, Employees, Customers, View Sales Invoice Details, View Sold Products Details, Add New Sale, and Reports. At the top right, it says 'Logout(Logged in as Admin)'. The main content area has a title 'ADD MEDICINE DETAILS'. It contains fields for Medicine ID (empty), Price (empty), Medicine Name (empty), Location (empty), Quantity (empty), and Category (set to 'Tablet'). A 'Add Medicine' button is located to the right of the quantity field.

Figure 7.5:Creation of New Record

The figure 7.5 displays the page that allows user to add the details of a new medicine.

7.6 Update or Edit existing record

The screenshot shows the 'Pharmacy Management System' interface. The sidebar and top navigation are identical to Figure 7.5. The main content area has a title 'UPDATE MEDICINE DETAILS'. It contains fields for Medicine ID (123001), Price (1.00), Medicine Name (Dolo 650 MG), Location (Rack 7), Quantity (650), and Category (Tablet). An 'Update' button is located to the right of the quantity field.

Figure 7.6:Update or Edit Existing Record

The figure 7.6 shows the page that allows user to edit the details of an existing medicine.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Pharmacy management system can significantly improve operational management and thus streamline the process. This allows to automate the process of Pharmacists collecting and retrieving information, improving the response time of the sales. This project was an attempt to make the structure and working of a pharmacy management system simpler, user-friendly and similar to the real world implementation. This is very easy to work and saves a lot of time because this system is indispensable for managing details such as regular customer records, medicines stock, etc. On this project all the necessary functions are implemented in a constructive manner.

8.2 Future Enhancements

- The project supports pharmacy management of drugs in a website, in the future it can be linked to a web application to allow easy access to data for the owner.
- Email and message verifications could be sent to customers on successful billing of drugs.
- Hosting it on an online web server.
- It can be made available to customer to book medicines from the same website i.e., similar to admin and pharmacist login, customer login can also be included.
- If a particular drug doesn't exist in the given branch it can be made to show the availability of the drug in other branches.

REFERENCES

- [1] https://www.irjmets.com/uploadedfiles/paper/volume_3/issue_12_december_2021/18007/final/fin_irjmets1641021488.pdf
- [2] <https://ijesc.org/upload/4c640fa99bfccdcf6f90430972e263e6.PharmacyManagementSystemAReview.pdf>
- [3] <https://archivepp.com/article/inventory-management-in-pharmacy-practice-a-review-of-literature>
- [4] <https://github.com/abhisheks008/Medical-Store-Management-System.git>
- [5] www.stackoverflow.com
- [6] www.mysql.com/doc
- [7] <https://docs.microsoft.com>
- [8] <https://www.geeksforgeeks.org>
- [9] <https://www.w3schools.com/default.asp>
- [10] <https://www.programiz.com/>