

Chapter 1

INTRODUCTION

A Database Management System (DBMS) refers to the technology for creating and managing databases. Basically, DBMS is a software tool to organize (create, retrieve, update or manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient by data, we mean known facts that can be recorded and that have the embedded meaning. Database systems are meant to handle large collection of information.

1.1 Problem Statement

Pharmacy management has kept paper record in filing cabinets. Managing a very large pharmacy with records on papers will be tedious and difficult to keep track of inventories with regards to the drugs in the store, quantity of drugs available based on the categories and their functions. The pharmacist has to order drugs to replenish the already diminishing stock. In addition, ordering of drugs is being carried out manually. Significant amount of time is allocated for writing the order as the pharmacist needs to go through the stock balance and make rough estimate of the amount to order based on the prescription.

The proposed project allows the pharmacist to add prescription and the stocks to the database as it is required for the proper running of the pharmacy and it will also helps to update any, if it is required to make changes for the stored database.

1.2 Objectives

- a. Provide for mass storage of relevant data.
- b. To provide facility for updating the stocks of medicine in the pharmacy.
- c. Provide facility for storing medicine details and patient information.
- d. Provide facility for viewing the stocks in the pharmacy.

1.3 SQL

(Structured Query Language) is a computer-based structured, formatted database language designed for managing data in relational database managing systems SQL (RDBMS). SQL is a

standardized computer language which was initially developed by IBM for querying, altering and defining relational databases, using declarative statements.

- SQL is Structured Query Language, which was initially developed by IBM.
- SQL is pronounced as "sequel".
- SQL is a computer language for storing, manipulating, and retrieving data in a relational database.
- SQL is the standard language for Relation Database System.

1.4 PHP

PHP is an open-source server-side language which is used for creating dynamic web pages. It can be embedded into HTML. PHP is usually used in conjunction with a MySQL database on Linux/UNIX web servers. It is probably the most popular scripting language. And it is a widely-used general-purpose scripting language and interpreter that is freely available.

1.5 HTML5

HTML is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

1.6 CSS3

CSS is abbreviated as Cascading Style Sheets and describes how HTML elements need to be displayed when represented in a web page format or other media. It also helps save a lot of work because controlling the layout of multiple web pages can be done all at a time. It helps in representing how markup-based documents can be presented in conjunction with HTML. CSS is said to as the cornerstone design tool of the World Wide Web along with HTML and JavaScript. CSS is intended for enabling the separation of appearance with content, which includes layout, coloring and font styles.

1.7 XAMPP

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MySQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

Chapter 2

REQUIREMENT ANALYSIS AND SPECIFICATION

Requirement analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and Non-Functional Requirements.

2.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Functional Requirements of the pharmacy database management system include:

Login module:

- Pharmacist can log on to the website with their Pharmacist ID and password which is provided by the admin.
- The Pharmacist can only log in if the Pharmacist ID and password is correct.

Prescription Management:

- The pharmacist can store the record of the prescription & repeat the medicine as & when required.
- Data stored in the system reduces the chances of errors.

Stock Management:

- Stocks present in the pharmacy can be stored in the database.

Functional requirements for a software system may be expressed in several ways. The functional Requirements are broadly classified into 2 categories, they are:

Hardware Requirements

Software Requirements

Hardware requirements

- Processor : Intel i3/i5,1.8GHz machine or above
- Main memory : 4GB RAM or more.
- Hard disk drive : 1TB

Software requirements

- Operating System : Windows 7 and higher
- Front end : HTML, CSS3
- Back end : PHP
- Software : Brackets, XAMPP
- Framework : Bootstrap

2.2Non-Functional Requirements

Non-functional requirements are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non behavioral requirements". Qualities, that are non-functional requirements, can be divided into two main categories: Execution qualities, such as security and usability, which are observable at run time. Some important non-functional requirements to consider when evaluating a pharmacy management system include

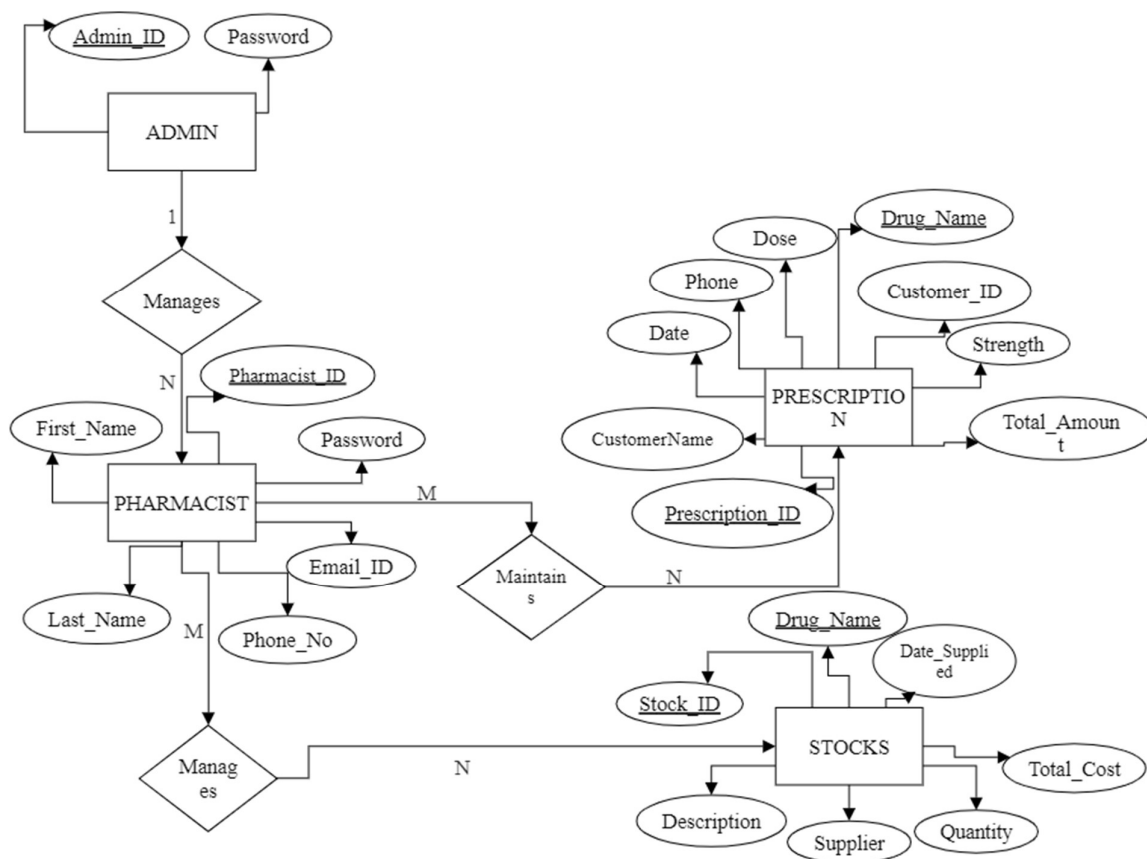
- System availability.
- Allowance for maintainability and enhancements.
- Response time.
- Throughput.
- Reliability.

Chapter 3

SYSTEM DESIGN

System Design process partitions the system into subsystems based on the requirements. It establishes overall system architecture and is concerned with identifying various components, specifying relationships among components, specifying software structure, maintaining a record of design decisions and providing a blue print for the implementation phase [6]. Design consists of architecture design and detailed design is concerned with the details of how to package processing modules and how to implement the processing algorithms, data structures and interconnections among modules and data structures.

3.1 ER Diagram



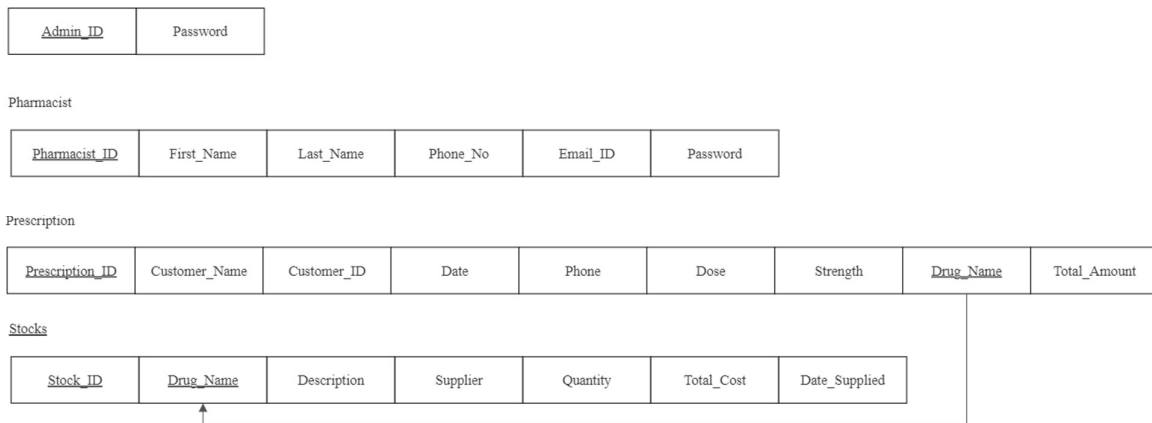
3.1 ER Diagram

The Entity-Relationship Data Model (ERD) perceives the real world as consisting of basic objects, called entity & relationship among these objects. It was developed to facilitate database

design by allowing specification of an enterprise schema, which represents overall logical structure of a database. The ERD model is very useful in mapping the meaning & interactions of the outside world enterprises onto a conceptual schema.

The ER diagram consists of four entities Admin, Pharmacist, Prescription, Stocks, each having its own attributes which defines its properties. The Primary keys under each entities are underlined. Admin manages the pharmacist having 1:N relationship whereas Pharmacist manages the prescription having M:N relationship.

3.2 Schema Diagram



3.2 Schema Diagram

Description of the Tables

Admin:

Admin_ID: Stores the Admin ID which is unique and not null.

Password: Password for admin login.

Pharmacist:

Pharmacist_ID: Stores the Pharmacist ID which Admin adds and it is a Primary Key.

First_Name: First Name of the Pharmacist.

Last_Name: Initial of Pharmacist.

Phone_No: Phone number of the Pharmacist.

Email_ID: Email ID of the Pharmacist.

Password: Password for Pharmacist login.

Prescription:

Prescription_ID: ID of the Prescription which is not supposed to be null and unique.

Customer_Name: Name of the Customer.

Customer_ID: ID generated by Pharmacist for customer.

Date: Date of Prescription generation.

Phone: Phone number of the customer.

Dose: Number of doses given.

Drug_Name: Name of the drug given to the customer.

Strength: Strength of each drug.

Total_Amount: Total amount payable.

Stocks:

Stock_ID: Stock ID which is Primary Key.

Drug_Name: Name of the drug

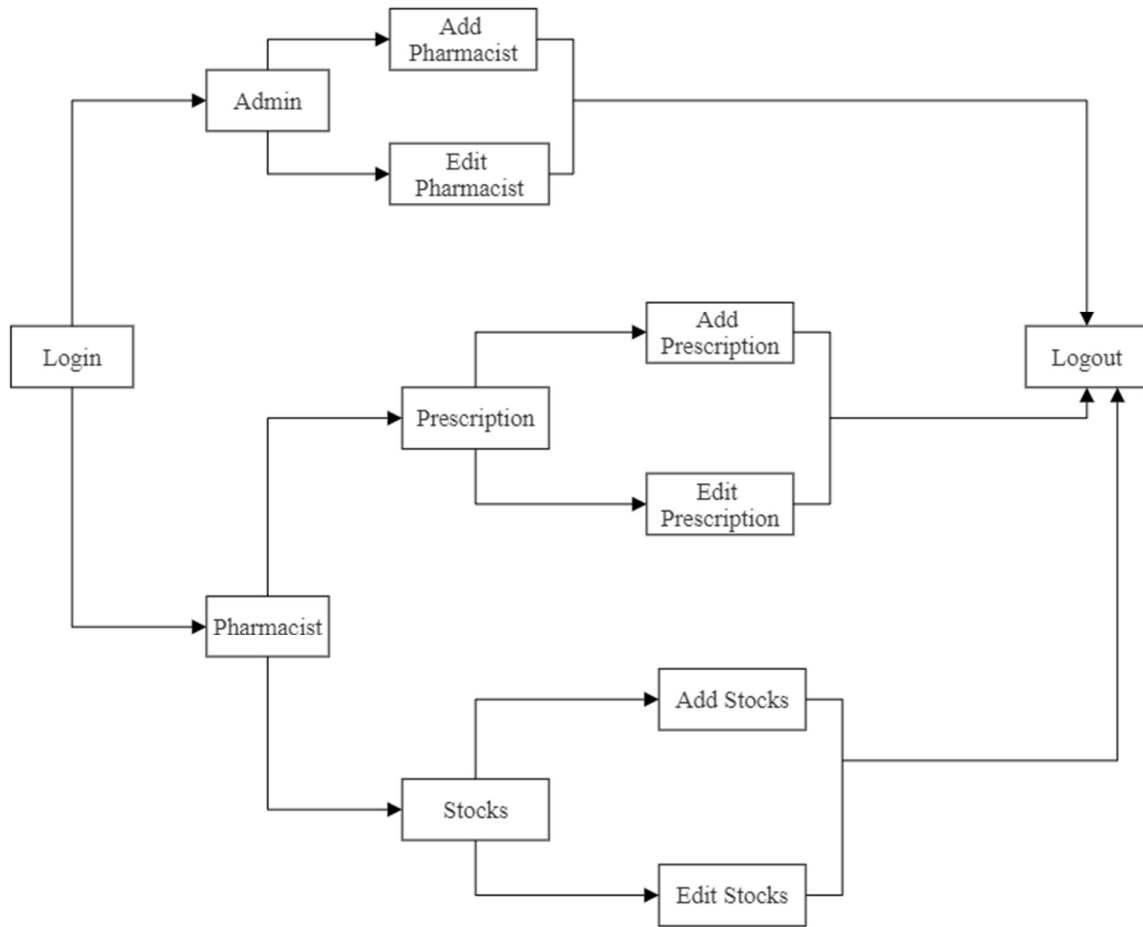
Description: Description of the drugs.

Supplier: Name of the pharmacy from where drugs have been taken.

Total_Cost: Total amount of the purchase.

Date_Supplied; Date when the stocks were bought.

3.3 Block Diagram



3.3 Block Diagram

Block diagram is used to represent the principal part or functions of a system. It is a high level representation. This makes easier to understand the data flow of a system. System session will start from login and will stop when user/admin logout from website. They are heavily used in hardware design, electronic design, software design and process flow diagrams.

Here in the above block diagram the session starts with the login page where the person has to choose between admin and pharmacist. Then the session continues until the person logout from the website.

3.4 Flowchart

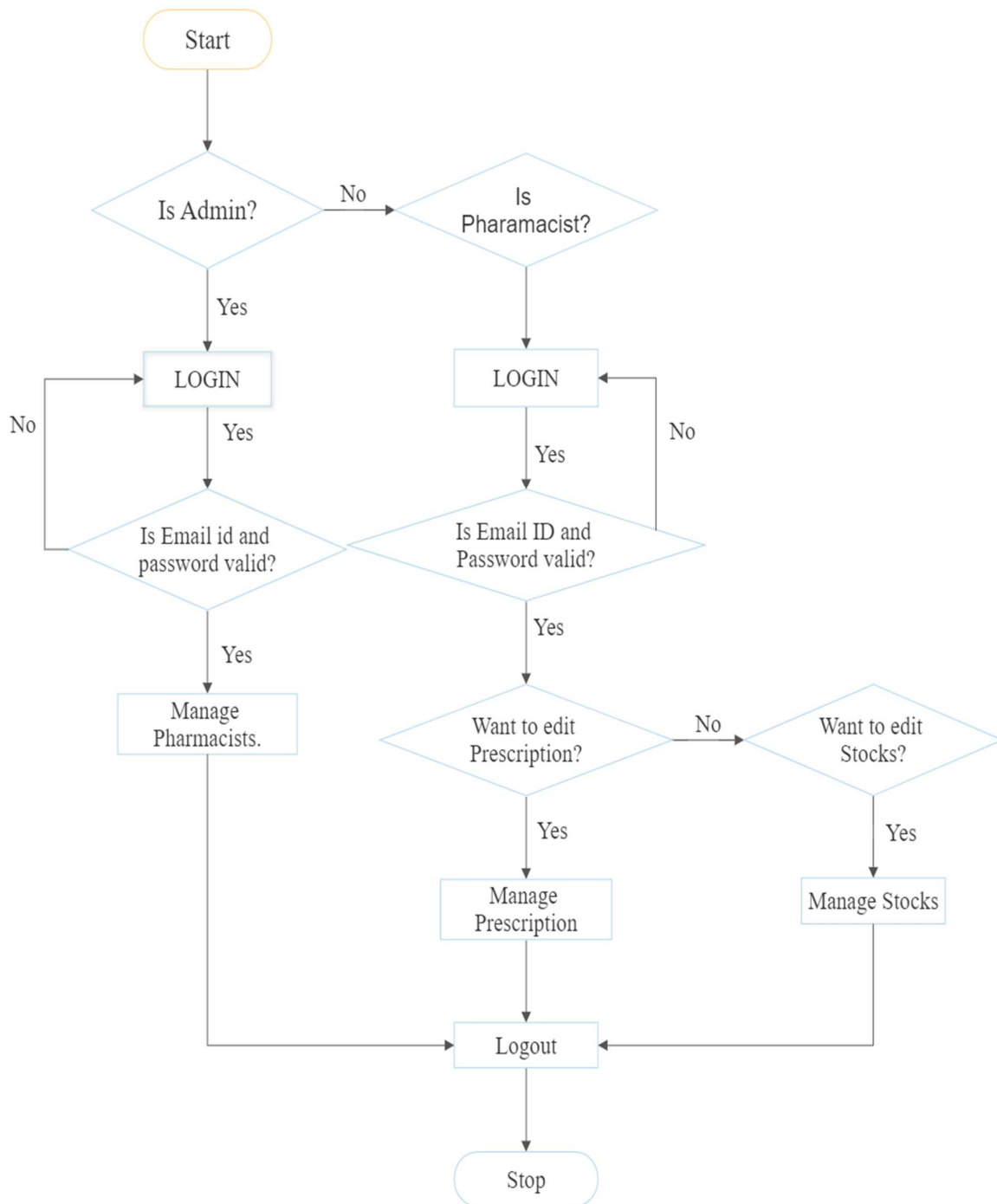


Figure 3.4 Flowchart

Flow chart represents the workflow or process of solving task. Based on conditions control will flow.

Chapter 4

IMPLEMENTATION

PHP: Hypertext Pre-processor (or simply PHP) is a server-side scripting language designed for web development, and also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

This project uses HTML as front-end tool. Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the world wide web. Web browser receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Browsers do not display the HTML tags, but use them to interrupt the content of the page.

```
<?php
$servername='localhost';
$username='root';
$password='';
$databasename = "pharmacy";
$conn=mysqli_connect($servername,$username,$password,$databasename);
if(!$conn)
{
    die("connection failed:".mysqli_connect_error());
}

?>
```

Figure 4.1 Code Snippet for connection module

The above code snippet in Figure 4.1 is used for connecting purpose. This is used to connect back end and front end. If the connection fails then a message occurs saying connection error.

```
<?php
$servername='localhost';
$username='root';
$password='';
$databasename = "pharmacy";

$conn=mysqli_connect($servername,$username,$password,$databasename);
if(!$conn)
{
    die("connection failed:".mysqli_connect_error());
} else {
    if(isset($_POST['login']))
    {
        $ADMIN_ID = $_POST['ADMIN_ID'];
        $PASSWORD = $_POST['PASSWORD'];
        $sql=mysqli_query($conn,"SELECT * FROM admin where ADMIN_ID='$ADMIN_ID' and
        PASSWORD='$PASSWORD'");

        $row = mysqli_fetch_array($sql);
        if(is_array($row))
        {
            $_SESSION["ADMIN_ID"]=$row['ADMIN_ID'];
            $_SESSION["PASSWORD"]=$row['PASSWORD'];
            echo "login successful";
            header("Location:admin.html");
        }
        else
        {
            echo "login unsuccessful";
            header("Location:loginadmin.php");
        }
        mysqli_close($conn);
    }
}
```

?>

Figure 4.2 Code Snippet for Admin login

The above code snippet in Figure 4.2 is for Admin Login purpose. If the Email Id and password entered is correct Admin can login or else the message Login unsuccessful is returned.

```
<?php
$servername='localhost';
$username='root';
$password="";
$databasename = "pharmacy";
$conn=mysqli_connect($servername,$username,$password,$databasename);
if(!$conn)
{
die("connection failed:".mysqli_connect_error());
}
if(isset($_POST['submit']))
{

$Prescription_ID = $_POST['Prescription_ID'];
$Customer_Name = $_POST['Customer_Name'];
$Customer_ID = $_POST['Customer_ID'];
$Date=$_POST['Date'];
$Phone = $_POST['Phone'];
$Dose = $_POST['Dose'];
$Drug_Name = $_POST['Drug_Name'];
$Strength = $_POST['Strength'];
$Total_Amount = $_POST['Total_Amount'];

$sql_query ="insert into prescription(Prescription_ID ,Customer_Name ,Customer_ID
,Date,Phone,Dose,Drug_Name,Strength,Total_Amount)
values('$Prescription_ID','$Customer_Name','$Customer_ID','$Date','$Phone','$Dose','$Drug_Na
me','$Strength','$Total_Amount')";
if(mysqli_query($conn, $sql_query))
{
echo "Prescription has been updated!";
}
else
{
echo "Error: " . mysqli_error($conn);
}

}

?>
```

Figure 4.3 Code Snippet for Prescription insertion

The above code snippet in figure 4.3 is for the purpose of the insertion of the prescription. Pharmacist needs to enter the required details about the Customer and the drugs he is going to take.

```

<?php
    $Prescription_ID=$_GET['Prescription_ID'];
    $showquery ="select * from prescription where Prescription_ID ={$Prescription_ID} ";
    $showdata=mysqli_query($conn,$showquery);
    $arrdata=mysqli_fetch_array($showdata);
    if(isset($_POST['submit']))
    {
        $Prescription_ID = $_POST['Prescription_ID'];
        $Customer_Name = $_POST['Customer_Name'];
        $Customer_ID=$_POST['Customer_ID'];
        $Date=$_POST['Date'];
        $Phone=$_POST['Phone'];
        $Dose=$_POST['Dose'];
        $Drug_Name=$_POST['Drug_Name'];
        $Strength=$_POST['Strength'];
        $Total_Amount=$_POST['Total_Amount'];
        $Quantity=$_POST['Quantity'];

        $query="update prescription set
        Prescription_ID='$Prescription_ID',Customer_Name='$Customer_Name',Customer_ID='$Customer_ID',Date='$Date',Phone='$Phone',Dose='$Dose',Drug_Name='$Drug_Name',Strength='$Strength',Total_Amount='$Total_Amount',Quantity='$Quantity'
        where Prescription_ID
        ='$Prescription_ID'";
        if(mysqli_query($conn, $query))
        {
            echo "values updated successfully";
        }
        else
        {
            echo "Error: " . $sql . ":" . mysqli_error($conn);
        }
        mysqli_close($conn);
    }
?>

```

Figure 4.4 Code snippet for Prescription updation

The above code snippet in Figure 4.4 is for the updation of the prescription. When there is the necessity of the updation of the prescription pharmacist can make changes to the prescription and it will return Values updated successfully.

```
<?php
$servername='localhost';
$username='root';
$password="";
$databasename = "pharmacy";
$conn=mysqli_connect($servername,$username,$password,$databasename);
if(!$conn)
{
    die("connection failed:".mysqli_connect_error());
}
$stockid=$_GET['Stock_ID'];
$query=" delete from stocks where Stock_ID=$stockid";
$result=mysqli_query($conn,$query);
if($result){
    ?>
    <script>
        alert("Deleted successfully");
    </script>
} else{
    ?>
    <script>
        alert(" Not Deleted");
    </script>
}
?>
header('location:stockview.php');
```

Figure 4.5 Code snippet for the Stocks deletion

The above code snippet in figure 4.5 is for the deletion of the stocks. If the stocks in the pharmacy needs to be deleted then above php code is used.

Chapter 5

TESTING

Software testing is the process of used to identify the correctness, security, completeness and quality of developed computer software. This includes the process of executing the program or applications with the intent of finding errors. An individual unit, functions or procedures of developed project is verified and validated and these units are fit for use.

5.1 Testing process

Best testing process is to test each subsystem separately, as we have done in project. Best done during implementation. Best done after small sub-steps of the implementation rather than large chunks. Once each lowest level unit has been tested, units are combined with related units and retested in combination. This proceeds hierarchically bottom-up until the entire system is tested as a whole. Typical levels of testing:

- Module- package, abstract data type, class
- Sub-system- collection of related modules, cluster of classes, method-message paths
- Acceptance testing- whole system with real data (involve customer, user, etc)

Alpha testing is acceptance testing with a single client (common for bespoke systems).

Beta testing involves distributing system to potential customers to use and provide feedback. In this project, beta testing has been followed. This exposes system to situations and errors that might not be anticipated by us.

5.1.1 Unit testing

Unit testing is the process of testing individual software components unit or modules. Since it needs the detailed knowledge of the internal program design and code this task is done by the programmer and not by testers.

5.1.2 Integration Testing

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole. After the completion

of each module it has been combined with the remaining module to ensure that the project is working properly as expected.

5.1.3 System Testing

System testing tests a completely integrated system to verify that it meets its requirements. After the completion of the entire module they are combined together to test whether the entire project is working properly.

5.2 Test Cases

A Test Case is a software testing document, which consists of events, action, input, output, expected result and actual result. Technically a test case includes test description, procedure, expected result and remarks. Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors.

Test cases no	Test Case	Expected results	Status
1	Logging into website	Email and password provided correct	Successful
2	Logging into website	Email incorrect	Unsuccessful
3	Logging into website	Password Incorrect	Unsuccessful
4	Logging into website	Any field left empty	Unsuccessful

Table 5.1 Test Case for Login

Table 5.1 represents the test case for login module. It shows both successful and unsuccessful results for the test cases.

Test cases no	Test Case	Expected results	Status
1	Inserting values	All details provided correctly	Successful
2	Inserting Values	Any one field is incorrect	Unsuccessful
3	Inserting Values	Any field left empty	Unsuccessful

Table 5.2 Test Case for Insertion

Table 5.2 represents the test case for insertion module. It shows both successful and unsuccessful results for the test cases.

Test cases no	Test Case	Expected results	Status
1	Updation	All details provided correctly	Successful
2	Updation	Any one field is incorrect	Unsuccessful
3	Updation	Any field left empty	Unsuccessful

Table 5.3 Test Case for Updation

Table 5.3 represents the test case for updation module. It shows both successful and unsuccessful results for the test cases.

Chapter 6

SCREENSHOT



Figure 6.1 Screenshot of Homepage

Figure 6.1 indicates the home page. This contains navigation bar, through which you can navigate to other pages. It also contains some details about the website at the home page.

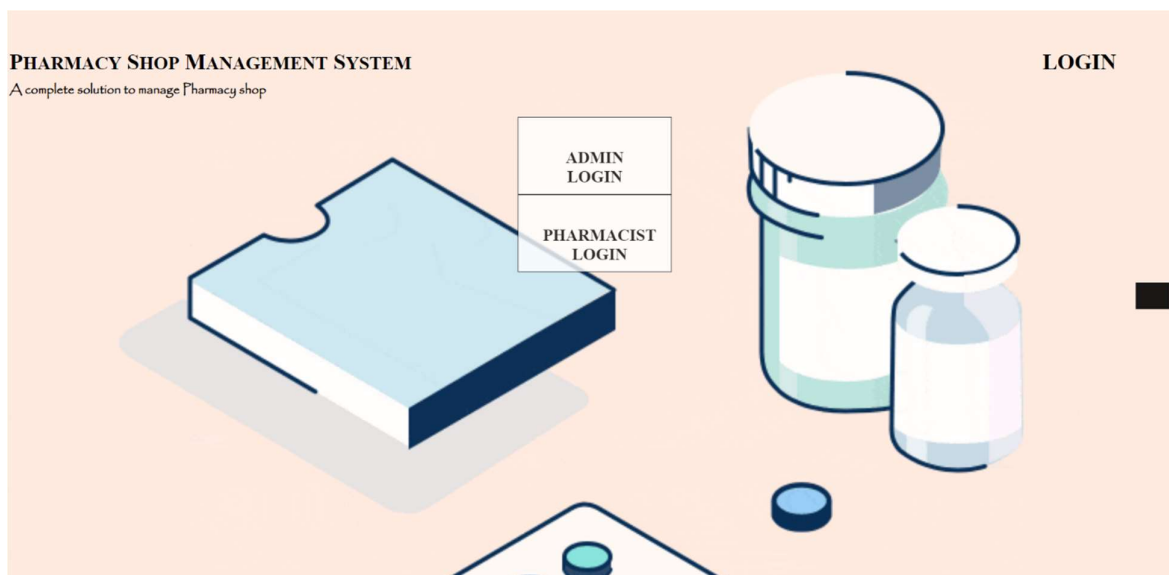


Figure 6.2 Screenshot of Login page

Figure 6.2 indicates the login page. This contains two options to select and the logout option to go back to home page.

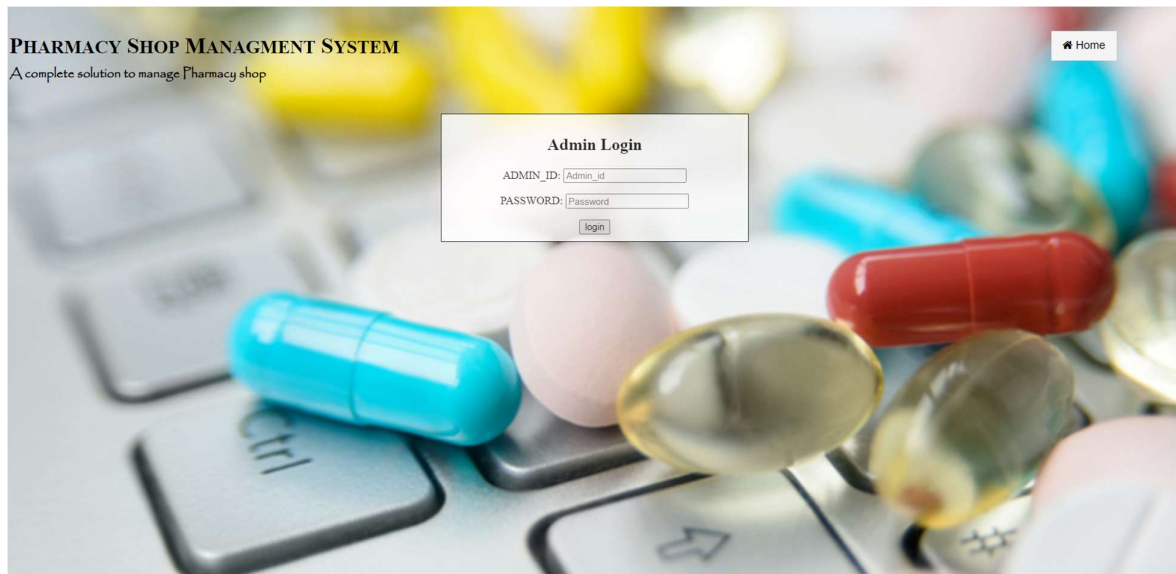



Figure 6.3 Screenshot of Admin Login

Figure 6.3 indicates the Admin Login page where the Admin needs to enter the Admin ID and the password. It also contains the option to logout from the admin page.



Figure 6.4 Screenshot of Admin Homepage

Figure 6.4 indicates the Admin Home page where the Admin needs to select between View and Add. View pharmacist is used for the edit purpose and the Add Pharmacist is used for Adding Pharmacist. It also contains the option to logout from the admin page.



ADD PHARMACIST

PHARMACIST_ID:

FIRST NAME:

LASTNAME:

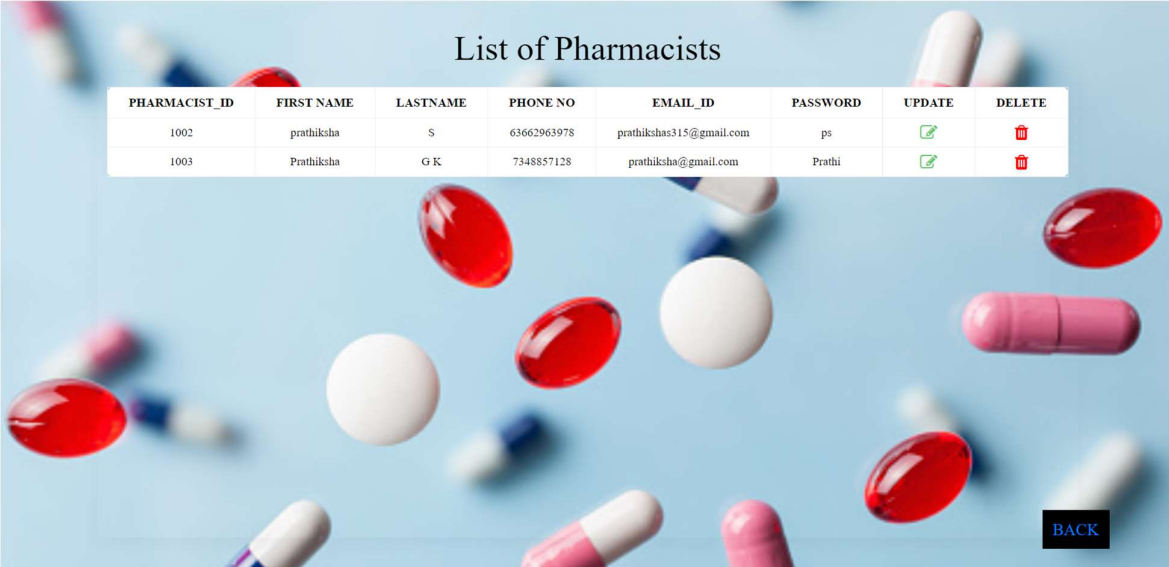
PHONE NO:

EMAIL_ID:

PASSWORD:

Figure 6.5 Screenshot of Add Pharmacist

Figure 6.5 indicates the Admin Add Pharmacist where the Admin needs to Add information regarding the Pharmacist such as Pharmacist ID ,Password etc. It also contains the option to go back to Admin home page.



List of Pharmacists





PHARMACIST_ID	FIRST NAME	LASTNAME	PHONE NO	EMAIL_ID	PASSWORD	UPDATE	DELETE
1002	prathiksha	S	63662963978	prathiksha315@gmail.com	ps		
1003	Prathiksha	G K	7348857128	prathiksha@gmail.com	Prathi		

Figure 6.6 Screenshot of Update Pharmacist

Figure 6.6 indicates the Update Pharmacist where the Admin is able to update the information of the pharmacist by clicking on update. It also contains the option to go back to Admin home page. Here the Admin have the option to delete the particular Pharmacist.

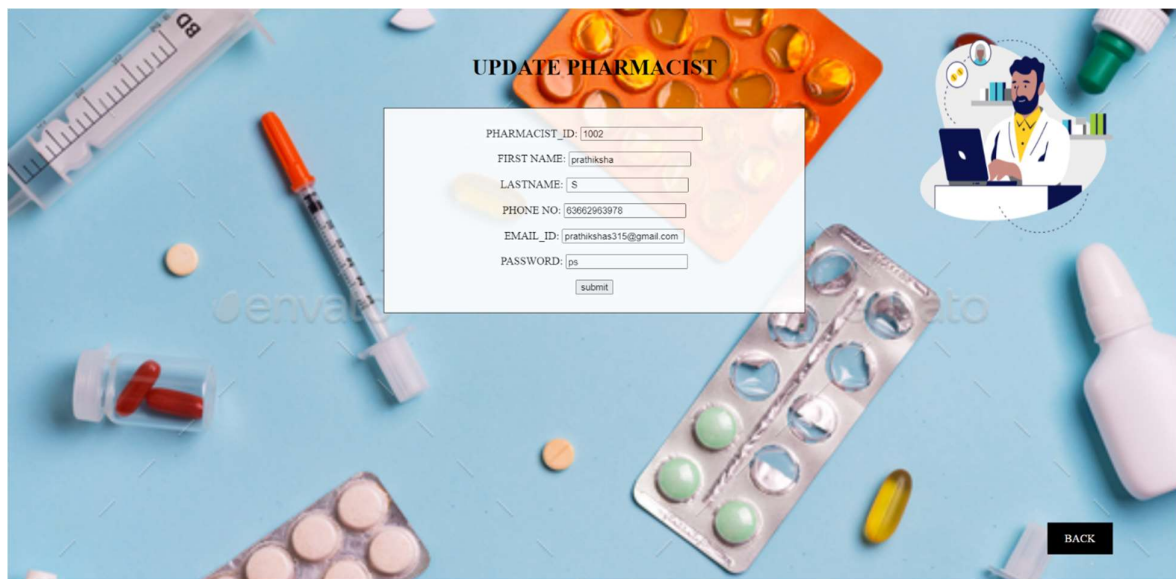


Figure 6.7 Screenshot of Update Pharmacist

Figure 6.7 indicates the Update Pharmacist where the Admin can update information regarding the Pharmacist such as Pharmacist ID , Password etc. It also contains the option to go back to Admin home page.



Figure 6.8 Screenshot of Pharmacist Login

Figure 6.8 indicates the Pharmacist Login page where the Pharmacist needs to enter the Pharmacist ID and the password. It also contains the option to logout from the Pharmacist page.



Figure 6.9 Screenshot of Pharmacist Homepage

Figure 6.9 indicates the Pharmacist Home page where the Pharmacist needs to select between Prescription and Stock.



Figure 6.10 Screenshot of Prescription Homepage

Figure 6.10 indicates the Prescription Home page where the Pharmacist needs to select between Edit and Add. Edit pharmacist is used for the edit purpose and the Add Pharmacist is used for Adding Prescription. It also contains the option to logout from the admin page.

Chapter 7

CONCLUSION

Effective implementation of this software will take care of the basic requirements of the pharmacy management system because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved. In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

REFERENCES

- [1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- [2] W3School <https://www.w3schools.com/>