***PHARAMACY MANAGEMENT SYSTEM***

Sai Thejas Chillagatu Ramesh - 2837401

Veera Venkata Jagannadham Ganta - 2833181

Rama Koteswara Rao Kolla - 2835029

Rajasekhar Gali - 2836153

**Table of Contents**

**1. Introduction**

1.1. Purpose and Scope

1.2. Product Overview

1.3. Structure of the Document

1.4. Terms, Acronyms, and Abbreviations

**2. Project Management Plan**

2.1. Project Organization

2.2. Lifecycle Model Used

2.3. Risk Analysis

2.4. Hardware and Software Resource Requirements

**3. Requirement Specifications**

3.1. Stakeholders for the system

3.2. Use cases

3.2.1. Graphic use case model

3.2.2. Textual Description for each use case

3.3. Rationale for your use case model

3.4. Non-functional requirements

**4. Architecture**

4.1. Architectural style(s) used

4.2. Architectural model (includes components and their interactions)

4.3. Technology, software, and hardware used

4.4. Rationale for your architectural style and model

**5. Design**

5.1. User Interface design

5.2. Database design

**6. Test Management**

6.1. A complete list of system test cases

6.2. Traceability of test cases to use cases

6.3. Techniques used for test case generation

6.4. Test results and assessments (how good are your test cases? How good is your software?)

6.5. Defects reports

**7. Conclusions**

7.1. Outcomes of the project (are all goals achieved?)

7.2. Lessons learned

7.3. Future development

**References**

**List of Figures**

Figure 2.3.2.1.1

Figure 4.2.1

**List of Tables**

Table 6.2.1

**1. Introduction**

**1.1. Purpose and Scope:**

Pharmacy Management System is a web application in PHP, and it is implemented by using windows appliance. The application provides an easy view of the details of the stocks present in the store. Pharmacy Management System is a windows application written for 32-bit and 64-bit operating systems which focuses on adding, editing, and deleting the medicines and suppliers. The main aim in the project is to create an application that is helpful while storing the stock details.

**1.2. Product Overview (including capabilities, scenarios for using the product, etc.):**

Pharmacy Management System is a web application in PHP, and it is implemented by using windows appliance. The application provides an easy and simple way to store the stock details. Admin or the store owner can view the details of the stocks present in the store. Pharmacy Management System is a windows application written for 32-bit and 64-bit operating systems which focuses on adding, editing, and deleting the medicines and suppliers. The main aim of the project is to create an application that is helpful while storing the stock details.

In the existing system it is difficult to store the medicine description and the supplier details. Admin as to face problem while the stocks have been emptied so that this application will help him/her to get knowledge over the left-over stock so he could pre-order the medicines through their respective suppliers. This application also stores the details of the suppliers who supply specific medicines. The valuable information is stored in a single domain. Our main vision is to develop a system where the store-holders can have an easy access over the stocks or medicines or suppliers who supply the medicines.

**1.3. Structure of the Document:**

The structure of the document used in the pharmacy management system project was intensely complex, so complex that it required a team of experts to develop it. However, despite all the hard work put in by the team, when they went to press 'publish' on the document, an unexpected error occurred, and all the information ended up being written in ancient Egyptian Hieroglyphs.

**1.4. Terms, Acronyms, and Abbreviations:**

Bill was the chief programmer assigned on a pharmacy management system project. He relied heavily upon terms, acronyms, and abbreviations to ensure all the analytics and tracking functioned smoothly. Having been so occupied with coding, he accidentally kept using these conventions in his everyday conversations, leaving his friends somewhat dumbfounded and confused by what he was saying.

**2. Project Management Plan**

**2.1. Project Organization:**

In this project four of us worked as a team in these we had taken different positions **Sai Thejas** worked as Coding and implementation, Rama Koteswara rao has worked as Requirements, Veera Venkata Jagannadham has worked as Documentation and Rajashekar has worked as Tester.

**2.2. Lifecycle Model Used:**

In this project pharmacy management system, the lifecycle model is used is storing. In this web application the complete data of the medicines, suppliers and the customer data will be stored.

**2.3. Risk Analysis:**

In this project pharmacy management system while we were giving connections from one page to another page, we faced some many difficulties, and we rectified that by using different types of connectors. While we were doing implementation like coding in that we got so many syntax errors we had rectified all the syntax errors and finally we had succussed in clearing all the errors.

**2.4. Hardware and Software Resource Requirements**

**2.4.1 Software Resource Requirements:**

* Database: phpMyAdmin
* User Interface: Google Chrome
* Server: XAMPP Server
* Front End Design: HTML
* Back End Design: php
* Code Developing Software: phpstorm

**2.4.2 Hardware Resource Requirements:**

* Processor: Intel Processor
* Clock Speed: 1.5Ghz
* Screen Resolution: 1920\*1080
* Ram: 8GB
* Keyboard: QWERTY

**3. Requirement Specifications**

**3.1. Stakeholders for the system:**

The stakeholders for the system used in a pharmacy management system project were the hospital staff, pharmacists, and insurance companies. Everyone worked together to ensure the system was adequate for all their needs and expectations. Unfortunately, due to an oversight, none of them had realized that small mammals such as hamsters are now stakeholders as well - after one of them entered the hospital through an open window and pressed a few buttons on the main console.

**3.2. Use cases**

**3.2.1. Graphic use case model:**

**Figure 3.2.1.1**

Diagram

Description automatically generated

|  |  |
| --- | --- |
| Actors | Admin, login |
| Description | A set of login credentials is used to verify a user. These typically include a username and password. A login, however, could also contain other data, like a passphrase, passcode, or PIN number is required for some logins. |
| Data | Customer Information stored to the database for the future references. |
| Stimulus | Login command issued by Customer/Admin login. |
| Response | Confirmation that user provided the details correctly. |
| Comments | The customer should have a login details and given names correctly. |

|  |  |
| --- | --- |
| Actors | Admin, logout |
| Description | By logging out, you can stop other users from entering the system without first confirming their identity. It contributes to security by helping to safeguard current user access or stop unauthorized activity during the current login session. |
| Data | Customer Information stored to the database for the future references. |
| Stimulus | Logout command issued by Customer/Admin login. |
| Response | Confirmation that user provided the details correctly. |
| Comments | The customer should have a logout details and given names correctly. |

|  |  |
| --- | --- |
| Actors | Admin, View Medicine |
| Description | When he/she access the view medicine page it shows all medicine information of the store and availability |
| Data | The reservation date and time should be saved in the Reservation registered list that way manager can estimate the flow and also can add, delete or modify the information. |
| Stimulus | When the customer clicks on the Reservation tab that should open a time and date slot to select. |
| Response | Selected date and time should save in the Reservations registered list data base. |
| Comments | The data should be listed in the Reservation registered list should only displayed to the manager not to the customers. |

|  |  |
| --- | --- |
| Actors | Admin, Add Medicine |
| Description | After completing the view medicine, then add the medicine for the sales |
| Data | The admin login page will be prefixed in the source code with the login code as Admin$123 |
| Stimulus | When the Manager try to access the application he/she should insert the admin pin that way it will give access to the edit menu and edit reservation data. |
| Response | Admin should have a login detail and given names correctly. |
| Comments | Admin should have a login detail and given names correctly. |

|  |  |
| --- | --- |
| Actors | Admin, Add sales |
| Description | Depending on the added medicine , customer is used to add the sales |
| Data | Edited list should save and updated according to the manager requirement and also manager can change the reservation full end. |
| Stimulus | When the manager clicks on this it will start open to edit the data that was already save or to create an entire new data. |
| Response | Edited data should by saved in the database. |
| Comments | Admin should have a login detail and given names correctly. |

|  |  |
| --- | --- |
| Actors | Admin, add new customer |
| Description | With using the referrals ,add a new customer |
| Data | Add a new customer for tracking future references. |
| Stimulus | If the customer is new and then need to add the new customer. |
| Response | Edited data should by saved in the database. |
| Comments | Required to give customer details for adding. |

|  |  |
| --- | --- |
| Actors | Admi , add new Supplier |
| Description | With using the referrals ,add a new Supplier. |
| Data | Add a new customer for tracking details of the medicine and supplier details |
| Stimulus | Whenever supplier give the new medicines will save the whole details with date and cost of the medicines |
| Response | Edited data should by saved in the database. |
| Comments | Admin should be enter correct supplier details |

|  |  |
| --- | --- |
| Actors | Admin, add stocks |
| Description | Add stocks for who supplied and what are the medicines. |
| Data | The main purpose to track the stocks available in the store and the prices of the cost depends on the supplier. |
| Stimulus | In this function we can add the stocks with select the supplier and medicines and cost of each pill. |
| Response | Edited data should by saved in the database. |
| Comments | Admin must select all the fields like supplier and medicines, count and price of the pill. |

|  |  |
| --- | --- |
| Actors | Admin, view medicines and its details. |
| Description | View the medicines report. |
| Data | View the medicines name, supplier, description. |
| Stimulus | In this function need to show the report of the medicines and full details. |
| Response | Edited data should by saved in the database. |
| Comments | Admin can be view the full details of the medicines and their details. |

|  |  |
| --- | --- |
| Actors | admin, view supplier and details |
| Description | View the supplier details. |
| Data | To show the supplier details of the who all are supplying the medicines. |
| Stimulus | When the admin select the view supplier report he can watch the supplier details and who all are |
| Response | Edited data should by saved in the database. |
| Comments | Admin can see the full detail of the supplier report |

|  |  |
| --- | --- |
| Actors | Admin, View Customer Details |
| Description | In this the admin can check all the information of the customer |
| Data | The complete details of the customer like name , address , phone number and the history of the customer like the previous billing etc. |
| Stimulus | When the admin clicks on the customer’s name then the complete information of the customer will be displayed. |
| Response | When the admin selects the customer then he can see all the details of the customer. |
| Comments | The data should be listed in the customer page will be displayed to the admin. |

|  |  |
| --- | --- |
| Actors | Admin, View Billing Details |
| Description | In this the admin can see all the billing of the customers |
| Data | To show the billing report for the whole transactions. |
| Stimulus | To show the billing whole report of the specific customer if the customer is requested for whole billing like the previous billings |
| Response | When the admin selects the billing of the customer in billing page then we will get complete report of the customers |
| Comments | The data will be stored in billing page |

**3.2.2. Textual Description for each use case:**

The pharmacy manager was thrilled to use textual description for each use case in their pharmacy management system project. This helped organize the information and made it simpler for all the staff to understand. However, the system failed when an employee mistook a verbal order for a prescription as a literal use case title - leading to much chaos and confusion in the store.

**3.3. Rationale for your use case model:**

The pharmacy management system project team was initially uncertain about the rationale for a use case model. After careful consideration, it was decided that a use case model would provide the utmost clarity and accuracy during implementation of the system. To their delight, they soon discovered that their model also doubled as an excellent tool for playing tic-tac-toe during lunch breaks.

**3.4. Non-functional requirements:**

* Intel Processor required.
* Minimum of 8GB ram required.
* Google Chrome Web browser required.
* Windows 10 Operating System required.
* HTML, CSS, PHP, JavaScript scripting languages are used.
* MySQL 8.0.12 version database is used.
* Apache web server is used.
* PHP Strom IDE is used.

**4. Architecture**

**4.1. Architectural style(s) used:**

Architectural Style can be defined as the design of a software system, or the structure of components that make up the system. For example, when building a pharmacy management system, developers must consider what type of architectural style to use. There are several different types of architecture available and each one serves a different purpose.

The most popular architectural styles used in pharmacy management systems include layered architecture, service-oriented architecture (SOA), microservices architecture, and model-view-controller (MVC) architecture. Layered architecture is based on the idea that layers within an application should be connected but remain independent from one another. SOA focuses on connecting services together for better performance and scalability. Microservices architecture provides more flexibility by allowing developers to break down applications into small independent parts which can then be scaled independently according to user needs.

Client Server Architecture is one of the most popular architectural styles used in pharmacy management systems. It allows for distributed computing and shared processing, where multiple computers communicate with one another to perform tasks. This type of system provides users with a centralized data storage location and can be accessed by all network users.

The server component typically handles authentication and authorization of user access as well as maintaining the entire database. On the other hand, client computer systems receive requests from users, process it locally and then send back a response to the server. This architecture offers many advantages such as scalability, improved performance, better security control, easier maintenance, higher reliability, and improved availability among others. Additionally, this architecture allows for rapid development of applications since information is centrally stored on the server side and separate programs are developed for each client computer system.

**4.2. Architectural model (includes components and their interactions)**

**Figure 4.2.1**

**Diagram

Description automatically generated**

**4.3. Technology, software, and hardware used:**

* Xamppler
* MySQL
* Apache
* Computer

**4.4. Rationale for your architectural style and model:**

The rationale for the architectural style was rooted in the team's determination to create an efficient, robust system which could help pharmacies better manage their inventory. The model chosen was based on agile methodology because of its ability to optimize costs and deliver results quickly. However, unexpected geese began showing up on the work site due to its proximity to a nearby park. This spurred a lot of discussion about how best to protect the project from avian intervention but, in the end, it couldn't be helped: the team had to learn to work around them

**5. Design**

**5.1. User Interface design:**

The user interface in the industrial design field of human-computer interaction in the space where interaction between humans and machines occurs. The goal of these interactions is to allow effective operations and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operator’s decision-making process. In this project, we use HTML as a user interface tool. The main goal is this HTML user interface design is to produce a user interface which makes it easy, efficient, and enjoyable to a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and that the machine minimizes undesired outputs to the human.

**5.2. Database design:**

The success of database systems in traditional applications encouraged developers of other types of applications to attempt to use them. Such applications traditionally used their own specialized file and data structures. Scientific applications that store large amounts of data resulting from scientific experiments in areas such as high energy physics or the mapping of the human genome. Storage and retrieval of images, from scanned news or personal photographs to satellite photograph images and from medical procedures such as x rays and MRI.

Traditionally, database technology applies to structured and formatted data that arises in routing applications in government, businesses, and industry. There has been a concurrent development of field called information retrieval (IR) that deals with books and various forms of based articles. Data is indexed, catalogued, and annotated using keywords.

**6. Test Management**

**6.1. A complete list of system test cases:**

Please follow the attached here embedded excel sheet for the complete list of system test cases for pharmacy management system



**6.2. Traceability of test cases to use cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Test Scenario ID** | **Test Scenario Description** | **Tes Case ID'S** |
| 1 | PMS\_TS\_001 | Validate the working of Login functionality | PMS\_TC\_01\_001 |
| 2 | PMS\_TS\_002 | Validate the working of Homepage functionality | PMS\_TC\_02\_002 PMS\_TC\_02\_003 PMS\_TC\_02\_004 PMS\_TC\_02\_006 PMS\_TC\_02\_007 |
| 3 | PMS\_TS\_003 | Validate the working of Add Stock functionality | PMS\_TC\_03\_001 PMS\_TC\_03\_002 PMS\_TC\_03\_006 |
| 4 | PMS\_TS\_004 | Validate the working of Medicine Page functionality | PMS\_TC\_04\_001 |
| 5 | PMS\_TS\_005 | Validate the working of Supplier Page functionality | PMS\_TC\_05\_001 |
| 6 | PMS\_TS\_006 | Validate the working of Customer Page functionality | PMS\_TC\_06\_001 |
| 7 | PMS\_TS\_007 | Validate the working of Billing Page functionality | PMS\_TC\_07\_001 |
| 8 | PMS\_TS\_008 | Validate the working of Logout Button functionality | PMS\_TC\_08\_001 PMS\_TC\_08\_003 |

**Table 6.2.1**

**6.3. Techniques used for test case generation**

‘PMS Team’ has communicated with ‘each other and has understood that we need to perform Functional Testing of all the functionalities mentioned in the above Scope section. As part of Functional Testing, we will follow the below approach for Testing:

**Step#1** – Creation of Test Scenarios and Test Cases for the different features in scope.

• We will apply several Test Designing techniques while creating Test Cases

o Equivalence Class Partition

o Boundary Value Analysis

o Decision Table Testing

o State Transition Testing

o Use Case Testing

• We also use our expertise in creating Test Cases by applying the below:

o Error Guessing

o Exploratory Testing

• We priorities the Test Cases

**Step#2** – Our Testing process when we get an Application for Testing:

• Firstly, we will perform Smoke Testing to check whether the different and important functionalities of the application are working.

• We reject the build if the Smoke Testing fails and will wait for the stable build before performing in-depth testing of the application functionalities.

• Once we receive a stable build, which passes Smoke Testing, we perform in depth testing using the Test Cases created.

• Multiple Test Resources will be testing the same Application on Multiple Supported Environments simultaneously.

• We then report the bugs in bug tracking tool and send dev. management the defect found on that day in a status end-of-the-day email.

• As part of the Testing, we will perform the below types of Testing:

o Smoke Testing and Sanity Testing

o Regression Testing and Retesting

o Usability Testing, Functionality & UI Testing

• We repeat Test Cycles until we get a quality product.

**Step#3** – We will follow the below best practices to make our Testing better:

• Context-Driven Testing – We will be performing Testing as per the context of the given application.

• Shift Left Testing – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build.

• Exploratory Testing – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.

• End-to-End Flow Testing – We will test the end-to-end scenario which involves multiple functionalities to simulate the end user flows.

**6.4. Test results and assessments (how good are your test cases? How good is your software?):**

Please follow the attached here embedded excel sheet for the complete list of system test cases for pharmacy management system



**6.5. Defects reports:**

Please follow the attached here embedded excel sheet for the complete list of system test cases for pharmacy management system

****

**7. Conclusions**

**7.1. Outcomes of the project (are all goals achieved?):**

All the goals are achieved in this project and every outcome has been executed successfully.

# 

* Admin can login through the webpage.
* Admin can view the stock details.
* Inventory of medicines in the pharmacy can be inserted into the database.
* Admin can add new medicines and supplies.
* Admin can edit/delete stock details.
* Admin can enter customers and sales details.

**7.2. Lessons learned:**

After weeks of hard work, the pharmacy management system project was finally completed. The project leader had learned many valuable lessons from the process - such as to better prepare for challenges, think outside of the box and be open to feedback. However, most importantly, he realized that it is important to remember not to leave the tablets lying around when people are hungry - otherwise you may end up with a lot more orders than you were ever expecting.

**7.3. Future development:**

Online medicine sales. Automatic bill generation. Online medical consultancy services. Improve accuracy, enhance safety and efficiency in the pharmaceutical store. Develop to ensure the security of information and reliability of Pharmacy records when accessing and providing services to the customers. Pharmacists are employed in regulatory control and drug management, community pharmacy, hospital pharmacy, the pharmaceutical industry, academic activities, training of other health workers, and research. In all these fields their aim is to ensure optimum drug therapy, both by contributing to the preparation, supply and control of medicines and associated products and by providing information and advice to those who prescribe or use pharmaceutical products.

**References:**

<https://www.javatpoint.com/xampp>

<https://en.wikipedia.org/wiki/PHP#References>

<https://en.wikipedia.org/wiki/MySQL_AB>

<https://en.wikipedia.org/wiki/HTML>

<https://en.wikipedia.org/wiki/CSS>