**Electricity Billing System**

**A PROJECT REPORT**

**for**

**Major Project (KCA353)**

**Session (2023-24)**

**Submitted by**

**Shivani Sharma**

**(2200290140147)**

**Ritik Tyagi**

**(2200290125)**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Dr. Shashank Bhardwaj**

**Associate Professor**



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(March-2024)**

**DECLARATION**

I hereby declare that the work presented in report entitled “Electricity billing System” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

**Name:** Ritik Tyagi

**Roll No.:** 2200290140125

**Name:** Shivani Sharma

**Roll No.:** 2200290140147

**CERTIFICATE**

Certified that **Ritik Tyagi , Shivani Sharma** have carried out the project work having “**Electricity Billing System**” (**Major Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the students themselves and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Ritik Tyagi 2200290140125**

**Shivani Sharma 2200290140147**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Date:**

**Dr. Shashank Bhardwaj Dr. Arun Tripathi**

**Associate Professor Head**

**Department of Computer Applications Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad KIET Group of Institutions, Ghaziabad**

**Electricity Billing System**

**Ritik Tyagi**

**Shivani Sharma**

**ABSTRACT**

This paper presents the design and implementation of an advanced electricity billing system aimed at enhancing efficiency, accuracy, and customer satisfaction in utility billing processes. Traditional electricity billing systems, often plagued by manual errors, delays, and inefficiencies, are being replaced by more sophisticated solutions that leverage modern technologies. Our proposed system integrates smart metering, automated data collection, and real-time monitoring to streamline the entire billing cycle.

The system employs smart meters to gather accurate consumption data, which is then transmitted to a central server using secure communication protocols. This data is processed to generate detailed and precise billing information, accessible to both utility providers and consumers through a user-friendly web portal. The portal also features tools for usage analysis, billing history, and online payment, thus enhancing transparency and user engagement.

Moreover, the system incorporates advanced analytics to predict consumption patterns and optimize energy usage, contributing to demand-side management and cost savings for both providers and consumers. By implementing robust security measures and ensuring compliance with regulatory standards, the system safeguards consumer data and maintains high levels of trust.

Overall, this electricity billing system represents a significant step towards modernizing utility services, offering a scalable and adaptable solution that can meet the evolving needs of the energy sector.

**ACKNOWLEDGEMENTS**

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, Dr. Shashank Bhardwaj (Associate Professor) for her guidance, help and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions have guided me a lot in completing this project successfully.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Ritik Tyagi (2200290140125)

Shivani Sharma (2200290140147)

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | Certificate |  |
|  | Abstract |  |
|  | Acknowledgement |  |
|  | Table of Content |  |
|  | List of Tables |  |
|  | Table of Figures |  |
|  | Abbreviation & Acronym |  |
| **1** | **INTRODUCTION** |  |
|  | 1.1 OVERVIEW |  |
|  | 1.2 BACKGROUND |  |
|  | 1.3 PROBLEM STATEMENT |  |
|  | 1.4 OBJECTIVE |  |
|  | 1.4.1 General Objective |  |
|  | 1.4.2 Specific Objective |  |
|  | 1.5 SIGNIFICANCE |  |
|  | 1.5.1 Efficiency |  |
|  |  |  |
|  | 1.5.2 Optimization |  |
|  | 1.5.3 Security and Data Accuracy |  |
|  | 1.5.4 Adaptability and Scalability |  |
|  | 1.5.5 Overall Impact |  |
|  | 1.6 SCOPE |  |
|  | 1.6.1 Functional Scope |  |
|  | 1.6.2 JavaScript Integration |  |
|  | 1.6.3 MySQL Database |  |
|  | 1.6.4 Technology Constraints |  |
| **2** | **LITERATURE REVIEW** |  |
|  | 2.1 OVERVIEW |  |
|  | 2.2 RECORDS & ELECTRONIC RECORDS |  |
|  | 2.3 DATABASES AS RECORD KEEPING SYSTEM |  |
|  | 2.4 IMPORTANCE OF ADMIN-EXCLUSSIVE ACCESS |  |
| **3** | **METHODOLOGY** |  |
|  | 3.1 INTRODUCTION |  |
|  | 3.2 SYSTEM DEVELOPMENT LIFECYCLE |  |
|  | 3.2.1 Requirement Analysis |  |
|  | 3.2.2 Planning |  |
|  | 3.3.3 System Design |  |
|  | 3.3.4 Technology Selection |  |
|  | 3.3.5 Development |  |
|  | 3.3.6 Testing & Quality Assurance |  |
| **4** | **SYSTEM DESCRIPTION** |  |
|  | 4.1 SYSTEM OVERVIEW |  |
|  | 4.1.1 Accessing the System |  |
|  | 4.1.1.1 Local Development Environment |  |
|  | 4.1.1.2 Authentication for Local Access |  |
|  | 4.1.1.3 3 Local Environment Accessibility |  |
|  | 4.1.2 User Privileges |  |
|  | 4.2 SYSTEM REQUIREMENTS |  |
|  | 4.2.1 Hardware Specifications |  |
|  | 4.2.2 Software Specifications |  |
|  | 4.3 SYSTEM ARCHITECTURE |  |
|  | 4.3.1 Backend Engine |  |
|  | 4.3.1.1 MySql Database |  |
|  | 4.3.1.2 java Programming Language |  |
|  | 4.3.1.3 Apache Web Server |  |
|  | 4.3.2 Frontend Interface Module |  |
|  | 4.3.3 Logical System Design |  |
|  | 4.3.4 Physical System Design |  |
|  | 4.4 ERD & DFD |  |
|  | 4.4.1 ER Diagram |  |
|  | 4.4.2 DFD Diagram |  |
|  | 4.5 SYSTEM FLOWCHART |  |
|  | 4.6 DATA INPUT |  |
|  | 4.6.1 Login Form |  |
|  | 4.6.2 Patient Registration Form |  |
|  | 4.6.3 Report Uploading Form |  |
|  | 4.6.4 Data Entry & Manipulation Form |  |
|  | 4.6.5 Adding user Form |  |
|  | 4.7 DATA OUTPUT |  |
|  | 4.7.1 Data Storage Interface |  |
|  | 4.7.2 Data Reports |  |
|  | 4.8 IMPLEMENTATION & TESTING |  |
|  | 4.8.1 Implementation |  |
|  | 4.8.2 Testing |  |
|  | 4.8.2.1 Unit Testing |  |
|  | 4.8.2.2 Integration Testing |  |
|  | 4.8.2.3 System Testing |  |
|  | 4.9 USE CASE DIAGRAM |  |
| **5** | **EVALUATION & CONCLUSION** |  |
|  | 5.1 EVALUATION |  |
|  | 5.2 LIMITATIONS OF SYSTEM |  |
|  | 5.3 PROBLEMS ENCOUNTERED |  |
|  | 5.4 RECOMMENDATION/FUTURE RESEARCH |  |
|  | 5.5 CONCLUSION |  |
|  | REFERENCES & BIBLIOGRAPHY |  |

**LIST OF FIGURES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Figure No.** | | **Name of Figure** | **Page No.** | |
| 4.1 | | Logical Database Design | |  | |
| 4.2 | | ER Diagram | |  | |
| 4.3 | | DFD Level-0 | |  | |
| 4.4 | | DFD Level-1 | |  | |
| 4.5 | | DFD Level-2 | |  | |
| 4.6 | | System Flow Chart | |  | |
| 4.7 | | Login Form | |  | |
| 4.8 | | Patient Registration Form | |  | |
| 4.9 | | Uploading Report form | |  | |
| 4.10 | | Add Health Record Form | |  | |
| 4.11 | | Add User Form | |  | |
| 4.12 | | Data Storage Interface for Patient History | |  | |
| 4.13 | | Interface for viewing Uploaded Reports | |  | |
| 4.14 | | Interface to Generate Report | |  | |
| 4.15 | | Use Case Diagram | |  | |
|  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Table** | **Page** |
| 4.1 | Medicine | 32 |
| 4.2 | Position | 32 |
| 4.3 | Medicine detail | 32 |
| 4.4 | Patient | 33 |
| 4.5 | Patient visit | 33 |
| 4.6 | Reports | 34 |
| 4.7 | Users | 34 |

**ABBREVIATIONS AND ACRONYMS**

|  |  |
| --- | --- |
| **ADMIN** | Administrator |
| **GUI** | Graphical User Interface |
| **HTML** | Hyper Text Markup Language |
| **IS** | Information System |
| **Lab** | Laboratory |
| **LAN** | Local Area Network |
| **PHP** | Hypertext Pre-Processor |
| **RAM** | Random Access Memory |
| **RM** | Records Management |
| **HRMS** | Health Record Management System |
| **SQL** | Structured Query Language |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Overview**

The electricity billing system is a critical component of modern utility infrastructure, responsible for ensuring that energy consumption is accurately measured, billed, and communicated to consumers Traditional billing methods, which often rely on manual meter readings and paper-based invoices, are increasingly inadequate in meeting the demands of today's fast-paced and technologically advanced society. These conventional systems are prone to human error, inefficiencies, delays, and lack the ability to provide real-time data, leading to customer dissatisfaction and operational challenges for utility providers.

To address these issues, the development of an advanced electricity billing system has become imperative. This system leverages modern technologies such as smart meters, automated data collection, and real-time monitoring to enhance the accuracy, efficiency, and transparency of the billing process. Smart meters, equipped with digital interfaces, enable precise measurement of electricity usage and facilitate automatic transmission of data to central servers. This eliminates the need for manual readings and significantly reduces the chances of errors.

The integration of a user-friendly web portal provides consumers with easy access to their consumption data, billing history, and payment options. This transparency empowers consumers to monitor their usage patterns, manage their energy consumption more effectively, and make informed decisions. Additionally, the system supports online payments, simplifying the payment process and enhancing customer convenience.

Furthermore, the electricity billing system incorporates advanced data analytics to predict consumption trends and optimize energy distribution. This capability not only aids in efficient energy management but also contributes to demand-side management, allowing utility providers to balance supply and demand more effectively. By implementing robust security measures, the system ensures the protection of consumer data and compliance with regulatory standards, fostering trust and reliability.

In conclusion, the electricity billing system represents a significant advancement in utility billing processes, offering a scalable and adaptable solution that addresses the limitations of traditional methods. By embracing technological innovations, this system aims to enhance operational efficiency, improve customer satisfaction, and support sustainable energy management practices.

**1.2 Background**

Electricity billing systems have historically relied on manual processes for meter readings and invoicing. This approach is labor-intensive and prone to human errors, resulting in inaccurate billing and consumer complaints. Moreover, the lack of real-time data collection and analysis limits the ability of utility providers to manage energy distribution effectively and forecast demand accurately. The inefficiencies of traditional systems are compounded by the growing demand for energy and the need for more sophisticated energy management solutions. As the adoption of digital technologies accelerates, there is a pressing need for modernized billing systems that can provide accurate, timely data, streamline operations, and enhance transparency for consumers.

**1.3 Problem Statement**

The current electricity billing systems are plagued by several significant issues:

Inaccurate Meter Readings: Manual meter readings are prone to human errors, leading to incorrect billing.

* Delays in Bill Generation and Delivery: The process of collecting data, generating bills, and delivering them to consumers is time-consuming and often delayed.
* Lack of Real-Time Data: Consumers do not have access to real-time consumption data, limiting their ability to manage energy usage effectively.
* Inefficient Energy Management: Without accurate and timely data, utility providers struggle to manage energy distribution and forecast demand accurately.
* Limited Customer Engagement: Traditional billing systems do not provide consumers with easy access to their usage data or billing history, reducing transparency and engagement.

These challenges highlight the need for an advanced, automated, and user-friendly electricity billing system that can address these shortcomings and improve the overall efficiency and effectiveness of the billing process.

**1.4 Objective**

**1.4.1 General Objective**

The general objective of this project is to design and implement an advanced electricity billing system that leverages modern technologies to improve the accuracy, efficiency, and transparency of the billing process. By doing so, the system aims to enhance customer satisfaction and operational effectiveness for utility providers.

**1.4.2 Specific Objectives**

* **Integration of Smart Meters:** Implement smart meters to ensure precise and automated data collection, reducing the need for manual readings.
* **Secure Data Transmission:** Develop a secure communication protocol to transmit consumption data from smart meters to central servers, ensuring data integrity .
* **User-Friendly Web Portal:** Create a web portal that allows consumers to access their billing information, monitor their energy usage, view billing history, and make online payments easily.
* **Real-Time Monitoring and Analytics:** Implement tools for real-time monitoring and advanced data analytics to improve energy management and demand.
* **Data Security and Compliance:** Incorporate robust security measures to protect consumer data and ensure compliance with relevant regulatory standards.
* **Consumption Analysis and Demand-Side Management:** Provide tools for analyzing consumption patterns and implementing demand-side management strategies to optimize energy usage.

**1.5 Significance**

**1.5.1 Efficiency:**

The proposed electricity billing system significantly enhances operational efficiency by automating data collection and reducing the need for manual meter readings. Automated data collection minimizes human errors, leading to more accurate billing and timely delivery of invoices. This automation also reduces labor costs and allows utility providers to allocate resources more effectively.

**1.5.2 Optimization:**

The integration of real-time monitoring and advanced data analytics allows utility providers to optimize energy distribution and manage demand more effectively. By analyzing consumption patterns and predicting demand, providers can implement strategies to reduce energy wastage, balance load during peak times, and improve overall energy efficiency. This optimization leads to cost savings and enhanced service reliability for both providers and consumers.

**1.5.3 Security and Data Accuracy:**

The proposed system includes robust security measures to protect consumer data, such as encryption, secure communication protocols, and regular security audits. These measures ensure that sensitive information is safeguarded against unauthorized access and breaches. Additionally, automated data collection and processing enhance data accuracy, reducing the risk of billing errors and fostering consumer trust..

**1.5.4 Adaptability and Scalability:**

The modular design of the system allows for easy adaptation and scalability. It can be customized to meet the specific needs of different utility providers and scaled to accommodate growing customer bases and evolving technological advancements.

**1.5.7 Overall Impact:**

The implementation of an advanced electricity billing system has a profound impact on both utility providers and consumers. For providers, it streamlines operations, enhances accuracy, and supports sustainable energy practices. For consumers, it offers greater transparency, convenience, and control over their energy usage and expenses, leading to increased satisfaction and engagement. Overall, this system represents a significant step

**1.6 Scope**

The scope of the electricity billing system encompasses the functionalities, technologies, and constraints that define the development and implementation of the system. This section outlines the functional scope, integration of JavaScript for user interface enhancements, the use of MySQL for database management, and the technological constraints that may impact the project.

**1.6.1 Functional Scope**

The functional scope of the electricity billing system developed using Java and JSP includes the following key components and features:

* **Automated Meter Reading (AMR):** Integration of smart meters to automatically collect electricity consumption data and transmit it to a central server.
* **Billing and Invoice Generation:** Automated generation of accurate bills based on consumption data, with options for electronic delivery to consumers.
* **User Portal:** Development of a web-based portal using JSP where consumers can access their usage data, view billing history, and make online payments.
* **Real-Time Data Monitoring:** Implementation of real-time data monitoring tools for both consumers and utility providers to track energy usage and detect anomalies.
* **Data Analytics:** Incorporation of advanced analytics to analyze consumption patterns, predict demand, and optimize energy distribution.
* **Customer Notifications:** Automated notifications to inform consumers about bill generation, due dates, and unusual consumption patterns.
* **Customer Support:** Provision of customer support features, including FAQs, contact forms, and live chat options to assist users with their queries.
* **Security Measures:** Implementation of robust security protocols to protect consumer data and ensure secure communication between smart meters and the central server

**1.6.2 Java Integration**

The use of JavaScript enhances user interaction, enabling real-time updates, form validations, and interactive features within the system’s frontend, improving user experience and system responsiveness.

**1.6.3 MySQL Database**

XAMPP an integrated database creation software tool was used as the software for creating the MYSQL database. MySQL serves as the central repository for storing patient data, medical records, schedules, and administrative information. Its role encompasses data management, retrieval, and ensuring the system’s scalability and reliability.

**1.6.4 Technology Constraints**

While these technologies offer robust capabilities, considerations such as compatibility, browser support, and scalability should be addressed to ensure the system’s seamless functioning across various platforms and devices.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Overview**

The electricity billing system is a complex and essential component of utility management, tasked with accurately measuring, recording, and billing electricity consumption for residential, commercial, and industrial consumers. As technological advancements continue to evolve, traditional methods of billing are being replaced by more sophisticated and automated systems. This literature review aims to provide a comprehensive overview of the current state of electricity billing systems, with a focus on the transition from manual to electronic records, the role of databases in maintaining these records, and the critical importance of admin-exclusive access for ensuring system integrity and security.

**2.2 Records & Electronic Records**

A record is recorded information produced or received in the initiation, conduct or completion of an institutional or individual activity and that comprises content, context and structure sufficient to provide evidence of the activity regardless of the form or medium.

The distinctive feature of electronic records is that the content is recorded on a medium and in symbols (binary digits) that need a computer or similar technology to read and understand.

The concepts of "record" and "electronic record" are linked to the concept of the "archival function" which was defined as that group of related activities contributing to, and necessary for accomplishing the goals of identifying, safeguarding and preserving archival records, and ensuring that such records are accessible and understandable.

**2.3 Databases & Recordkeeping System**

Recordkeeping systems in the electronic, as well as in the paper, world is designed for the use of operational staff in current office operations. Recordkeeping systems have concrete boundaries and definable properties, and they are critical to the preservation of the records’ origin and evidential value. In the paper world, recordkeeping systems range from a simple filing system to a central registry.

Databases are being used as the records management systems of preference because of their informational value. Such databases are created for their informational value -- as an information resource. Statistical databases are good examples of this kind of database.

**2.4 Importance of Admin-Exclusive Access**

Admin-exclusive access is a critical security feature in any system handling sensitive data, including electricity billing systems. This access control mechanism ensures that only authorized personnel can perform high-privilege tasks, such as modifying system configurations, accessing sensitive consumer data, and performing administrative functions.

Restricting access to administrative functions helps protect the system from unauthorized access and potential security breaches. It also maintains data integrity by preventing accidental or intentional modifications to critical system data. Admin-exclusive access is essential for regulatory compliance, as it ensures that the system adheres to data protection laws and industry standards. Additionally, it provides an audit trail of all administrative actions, enhancing accountability and enabling the detection and investigation of suspicious activities

**CHAPTER 3**

**METHODOLOGY**

**3.1 Introduction**

The methodology section outlines the structured approach employed to design, develop, and implement the electricity billing system. This systematic process ensures that all aspects of the project are carefully planned, executed, and evaluated to meet the project’s objectives and requirements. The methodology is based on the System Development Lifecycle (SDLC), which provides a framework for managing the project through various stages, from initial requirements gathering to final testing and deployment.

**3.2 System Development Life Cycle**

**3.2.1 Requirement Analysis**

Techniques such as interviews were conducted to elicit and document functional and non-functional requirements. Use case analyses, user stories, and personas were crafted to encapsulate diverse scenarios, ensuring a thorough understanding of system needs and user expectations.

**3.2.2 Planning**

A project plan was developed as well as other planning documents. It provided the basis for acquiring the resources needed to achieve a solution. This phase ensured that the problem solved was the one that needed to be solved and that the initial description was complete and consistent.

Under this phase:

The project team was formed and a project leader appointed

The system flowcharts were prepared

**3.3.3 System Design**

The System Design phase constituted an in-depth translation of gathered requirements into a detailed architectural blueprint. Design artifacts included high-level system architecture, wireframes, prototypes, and detailed data models. Unified Modelling Language (UML) diagrams, including use case diagrams, sequence diagrams, and class diagrams, were instrumental in visualizing system components, interactions, and hierarchies, offering stakeholders a tangible representation of the envisioned system.

**3.3.4 Technology Selection**

Careful consideration was given to selecting technologies aligning with project requirements and future scalability. Evaluations encompassed the backend, where PHP emerged as the scripting language due to its versatility, extensive libraries, and compatibility with MySQL, chosen as the backend database for its robustness and scalability. Frontend technologies centred on HTML for structuring content, CSS for styling, and JavaScript for dynamic and interactive user experiences, ensuring cross-browser compatibility and responsive design.

**3.3.5 Development**

The Development phase involved iterative coding and continuous integration. Agile methodologies were employed, enabling adaptive responses to evolving requirements. Backend development revolved around crafting PHP scripts to manage dynamic content, facilitate data processing, and orchestrate seamless interaction with the MySQL database. Frontend development encompassed HTML markup for content structure, CSS stylesheets for layout and visual aesthetics, and JavaScript for dynamic content updates and user interactivity, ensuring an intuitive user experience.

**3.3.6 Testing and Quality Assurance**

A meticulous testing strategy was executed, including unit tests, integration tests, and comprehensive system testing. Test-driven development principles guided the creation of test cases to validate individual components and system functionalities. Techniques such as black-box testing, white-box testing, and regression testing were employed to ensure reliability, robustness, and compliance with specified requirements.

**CHAPTER 4**

**SYSTEM DESCRIPTION**

**4.1 System Overview**

The System encompasses all the activities associated with the recording of patient details and progress all of which are integrated in the Wellness Program Management System. The main functionalities available in this system are

* Maintaining Patient details records
* Maintaining patients History records
* Maintaining Reports in Jpg/Pdf format

All these features include the ability to create, update (edit), retrieve through search results and truncate obsolete records. It also contains a report generation system that can be saved in a pdf file format. The system works in the following manner.

**4.1.1 Accessing the System**

Accessing the Electricity Billing System (EBS) on a local environment requires specific considerations to ensure secure and convenient access for multiple administrators.

**4.1.1.1 Local Development Environment**

Developers can access the system locally by setting up a development environment on their machines. This involves installing the necessary software components, including Java Development Kit (JDK), Apache Tomcat server, and MySQL database server. The development environment replicates the production environment, allowing developers to test and debug code efficiently.

**4.1.1.2 User Authentication for Local Access**

Local access to the system requires authentication to ensure that only authorized personnel can make changes and access sensitive data. Authentication mechanisms, such as username-password authentication or OAuth, can be implemented to verify the identity of users accessing the system locally.

**4.1.1.3 Local Environment Accessibility**

Access to the local environment is restricted to authorized personnel involved in system development and testing. Developers can access the system through their local machines using web browsers or integrated development environments (IDEs) such as Eclipse or IntelliJ IDEA.

**4.1.2 User Privileges**

The EBS incorporates an administrative delegation system, allowing admins to grant equivalent privileges to other users, thereby enabling them to access and perform similar functions as the granting admin.

* **Admin Creation and Delegation:** An admin with the necessary privileges can create new admins within the system, assigning equivalent access rights and functionalities.
* **Duplicate Admin Capabilities:** The newly appointed admin possesses similar access levels and capabilities as the granting admin, including data entry, modification, and access to system functionalities.
* **Delegation Control**: The granting admin retains the ability to manage and revoke the admin privileges granted to other users, ensuring control and oversight over delegated roles.
* **Equivalent Access Rights:** Delegated admins possess access rights and permissions identical to those of the granting admin, enabling them to perform similar tasks and access the same data and system features.

Data Security Measures: Security protocols remain consistent for both the granting admin and delegated admins, ensuring data confidentiality and integrity across the system.

**4.2 System Requirements**

The system requires a client-server architecture where a server is necessary to host the application and the database. The users will access the server to retrieve information from their desktops through their web-based interfaces. For this to work, the following will be required:

**4.2.1 Hardware Specifications**

* **Operating System:**

Windows 10/11, macOS, or Linux distributions supported by XAMPP

* **Processor:**

Intel Core i5 or AMD equivalent for optimal performance.

* **RAM:**

Minimum 4GB RAM for smooth operation; 8GB or higher recommended for better performance.

* **Storage:**

At least 20GB of available disk space for system files and data storage.

**4.2.2 Software Specifications**

The software specifications required for the system include:

* **Operating System:** Windows, Linux, or macOS
* **Java Development Kit (JDK):** Version 8 or higher
* **Apache Tomcat Server**: Version 8 or higher
* **MySQL Database Server:** Version 5.7 or higher
* **Integrated Development Environment (IDE):** Eclipse, IntelliJ IDEA, or NetBeans

These software components provide the necessary infrastructure for developing, deploying, and running the system.

**Other Considerations**

* **User Account:**

Admin credentials to log in to the EBS with appropriate access rights.

**4.3 System Architecture**

The Records Management System is architected to efficiently handle data storage, processing, and user interaction. The system's backbone is a robust backend engine, comprising a MYSQL database, PHP as the primary programming language, and Apache as the web server. Additionally, the frontend is structured using HTML and CSS, forming the user interface modules.

* + 1. **Backend Engine**
       1. **MySQL Database**

The MYSQL database serves as the foundational repository, storing patient records, medical histories, and associated data. It is structured to ensure data integrity, reliability, and efficient retrieval of information. The database architecture employs normalization techniques to minimize redundancy and optimize data storage.

* + - 1. **JAVA Programming Language**

Java is chosen as the programming language for the backend development of the system. It offers robustness, platform independence, and extensive libraries and frameworks for building scalable and secure web applications.

**4.3.1.3 Apache Web Server**

Apache serves as the robust web server, managing HTTP requests and responses. It hosts the PHP scripts and coordinates communication between the MYSQL database and the user interface modules. Apache's efficiency and stability contribute to the system's seamless operation.

**4.3.2 Frontend Interface Modules**

* **HTML (Hyper Text Markup Language)**

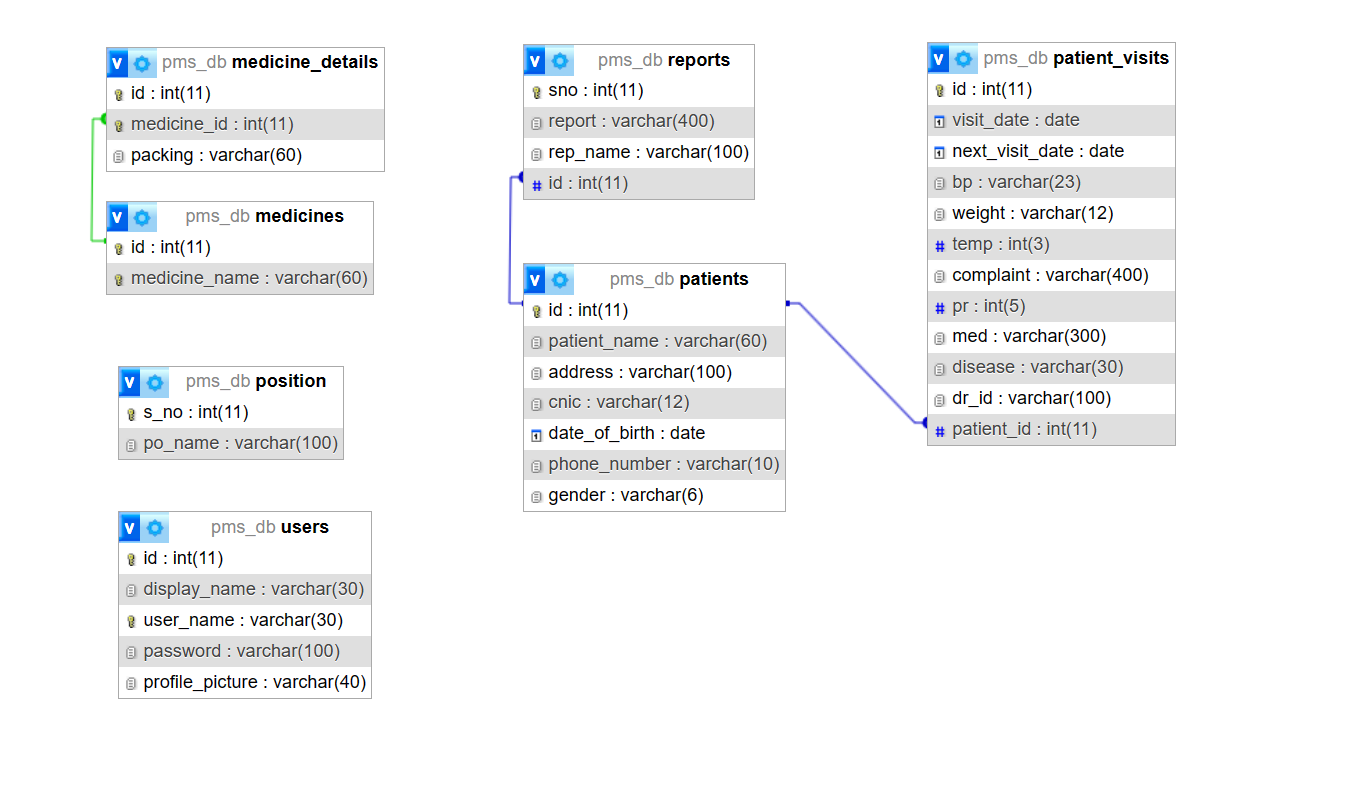
HTML forms the structural foundation of the user interface, defining the layout and structure of web pages. It provides the framework for presenting data and interacting with the system.

* **CSS (Cascading Style Sheets)**

CSS complements HTML by enhancing the presentation and visual appeal of the user interface. It controls the styling, layout, and design elements of the web pages, ensuring a cohesive and visually appealing interface for administrators.

**4.3.3** **Logical Database Design**

The logical database design is meant to describe the representation of the database in terms of its entities in form of tables and the existing relationships. Below is an illustration of the systems logical design as generated by the MYSQL workbench design tool.



**Fig. 4.1 Logical Database Design**

**4.3.4 Physical Database Design**

As one of the core elements of a Wellness Program management system, the database had to be designed in a meticulous systematic manner. This process started at the analysis phase of the project. From the analysis, the researcher was able to identify the necessary tables required for the database and the associated field names, format and length of each table. Below is a list of these tables.

#### **Table 4.1: Medicines**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Id | int | 11 | Primary Key, Auto Increment |
| Medicine\_name | varchar | 60 | To store the Physician Name |

#### **Table 4.2:** **position**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| S\_no | Int | 11 | Auto increment, primary key |
| Po\_name | varchar | 100 | Available Physician expertise |

#### **Table 4.3:** **Medicine Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| id | int | 11 | Auto Increment, Primary Key |
| Medicine\_id | int | 11 | Foreign Key |
| Packaging | varchar | 60 | Physician expertise corresponding their id |

#### **Table 4.4:** **Patients**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Id | Int | 11 | Auto Increment, Primary Key |
| Patient\_name | Varchar | 60 | patient name |
| Address | Varchar | 100 | address |
| Cnic | Varchar | 12 | Aadhaar number |
| dob | Date | - | DOB |
| Phone\_number | Varchar | 10 | contact number |
| gender | varchar | 6 | gender |

#### **Table 4.5:** **Patient visits**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Id | Int | 11 | Auto Increment, Primary Key |
| Visit\_date | Date | - | Store the current date |
| Next\_visit\_date | Date | - | NULL, |
| Bp | varchar | 23 | blood pressure |
| Weight | Varchar | 12 | Weight |
| Temp | Int | 3 | temperature |
| Complaint | Varchar | 400 | Patients’ complaint |
| Pr | Int | 5 | Pulse rate |
| Med | Varchar | 300 | Medication prescribed |
| Disease | Varchar | 30 | disease |
| Dr\_id | Varchar | 100 | Physician id |
| Patient\_id | int | 11 | Foreign key, patient id |

#### **Table 4.6:** **reports**

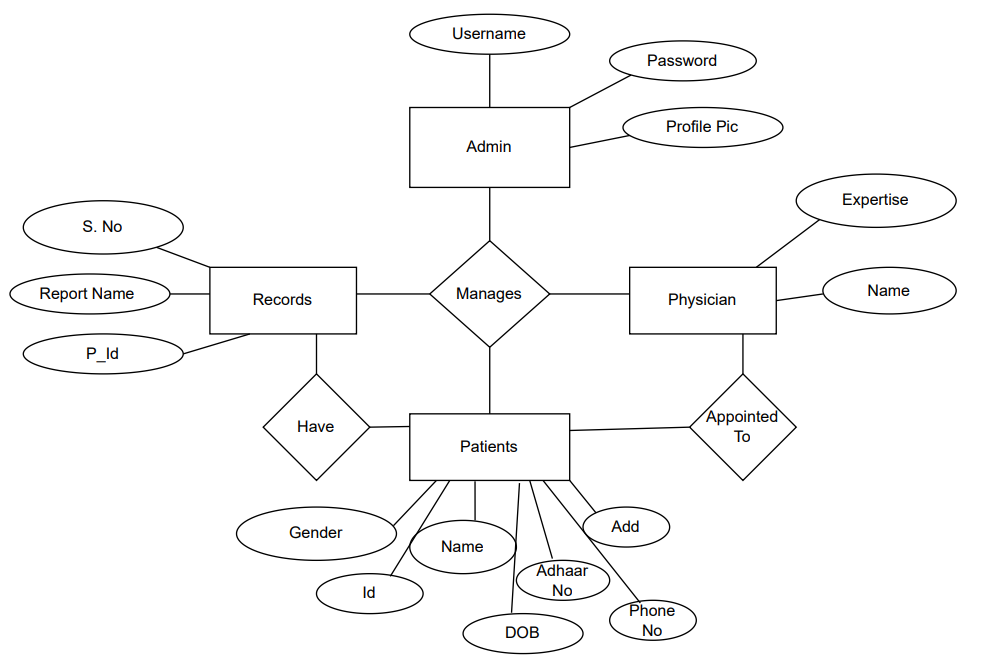
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Sno | Int | 11 | Auto increment, primary key |
| Report | Varchar | 400 | Report location |
| Rep\_name | Varchar | 100 | Type of report |
| id | int | 11 | Foreign key, Patient id |

#### **Table 4.7:** **users**

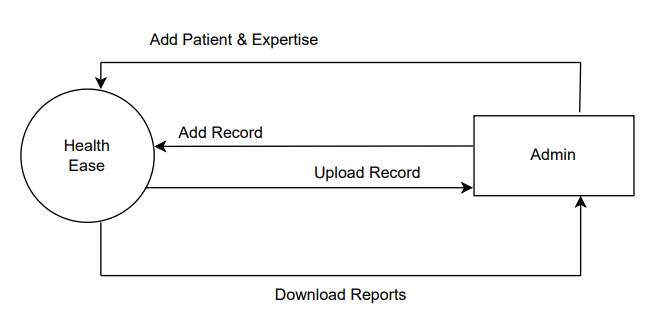
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Id | Int | 11 | Auto increment, primary key |
| Display\_name | Varchar | 30 | Name Displayed |
| User\_name | Varchar | 30 | User name |
| Password | Varchar | 100 | Encrypted password |
| Profile\_picture | varchar | 40 | Profile picture |

## **4.4 ER DIAGRAMS & DFDs**

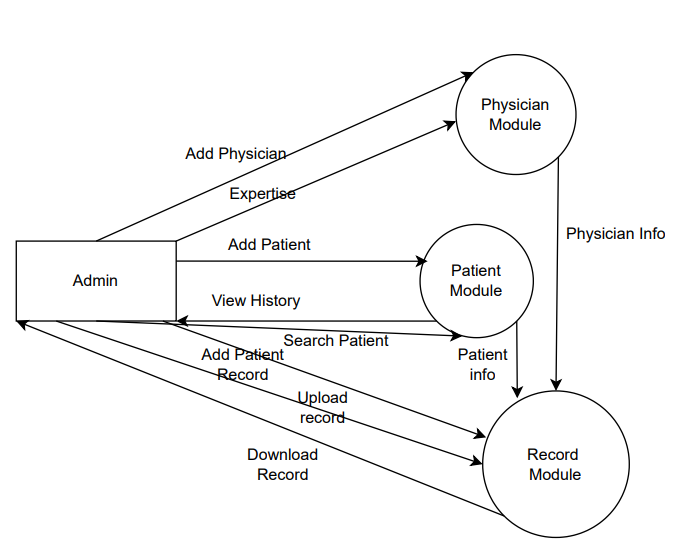
**4.4.1 ERD (Entity Relationship Diagram)**

**Fig 4.2 ER Diagram**

**4.4.2 DFD (Data Flow Diagram)**

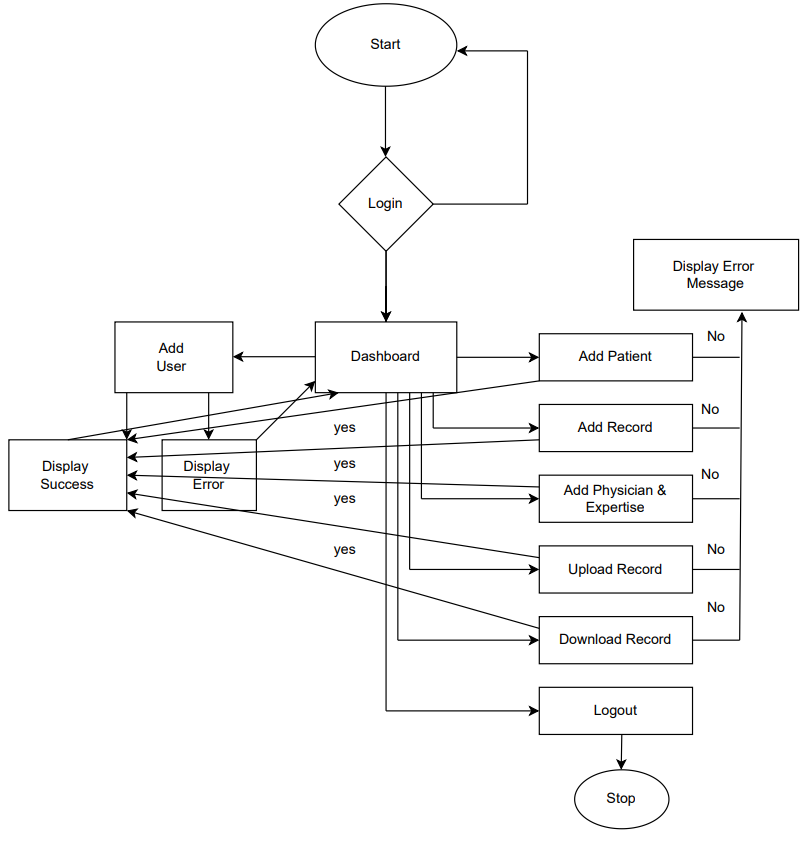
****

**Fig 4.3 DFD Level-0**

******

**Fig 4.4 DFD Level-1**

* 1. **System Flowchart**



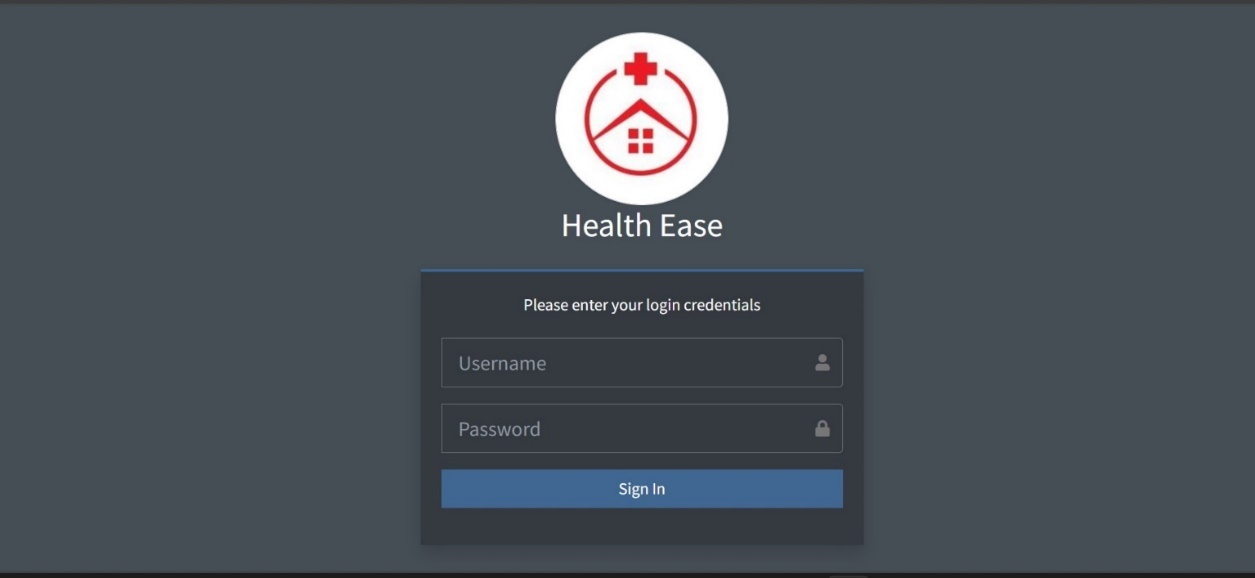
**Fig 4.6 System Flow Chart**

* 1. **Data Inputs**

Outputs are selected from the database based on a certain criterion and displayed using forms. The entire WPMS itself contains a number of forms, However, for the systems main components, below are some snap shots of the key forms.

**4.6.1 Login Form**

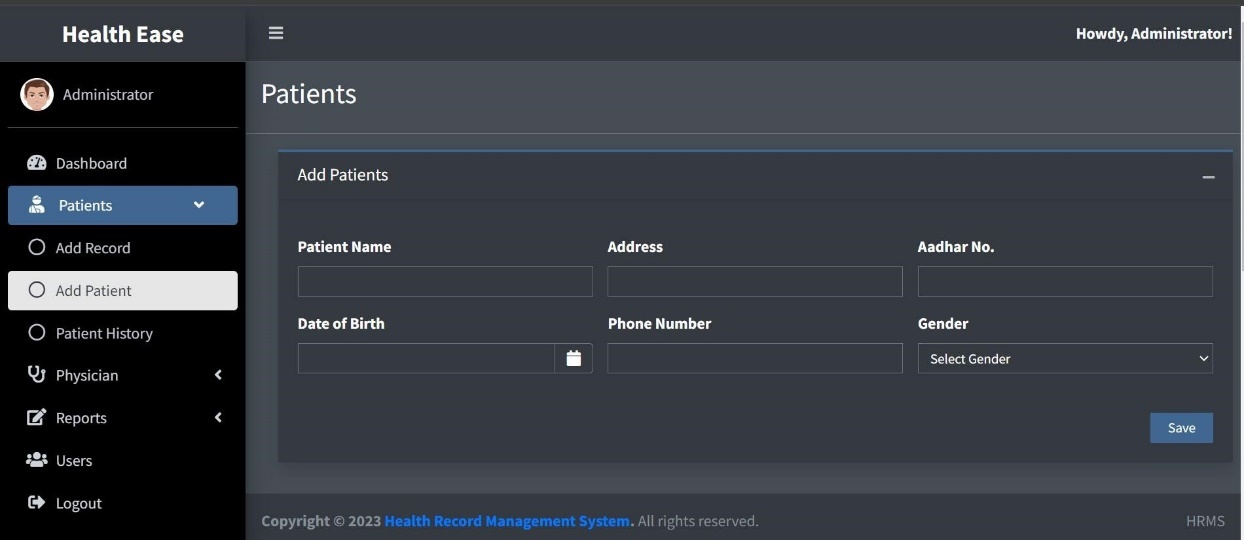
The login form above is the first page a person accessing the system sees. It is used to gain access to the system resources and determines, based on the user type, which users should access which resources.



**Fig 4.7 Login Form**

**4.6.2 Patient Registration Form**

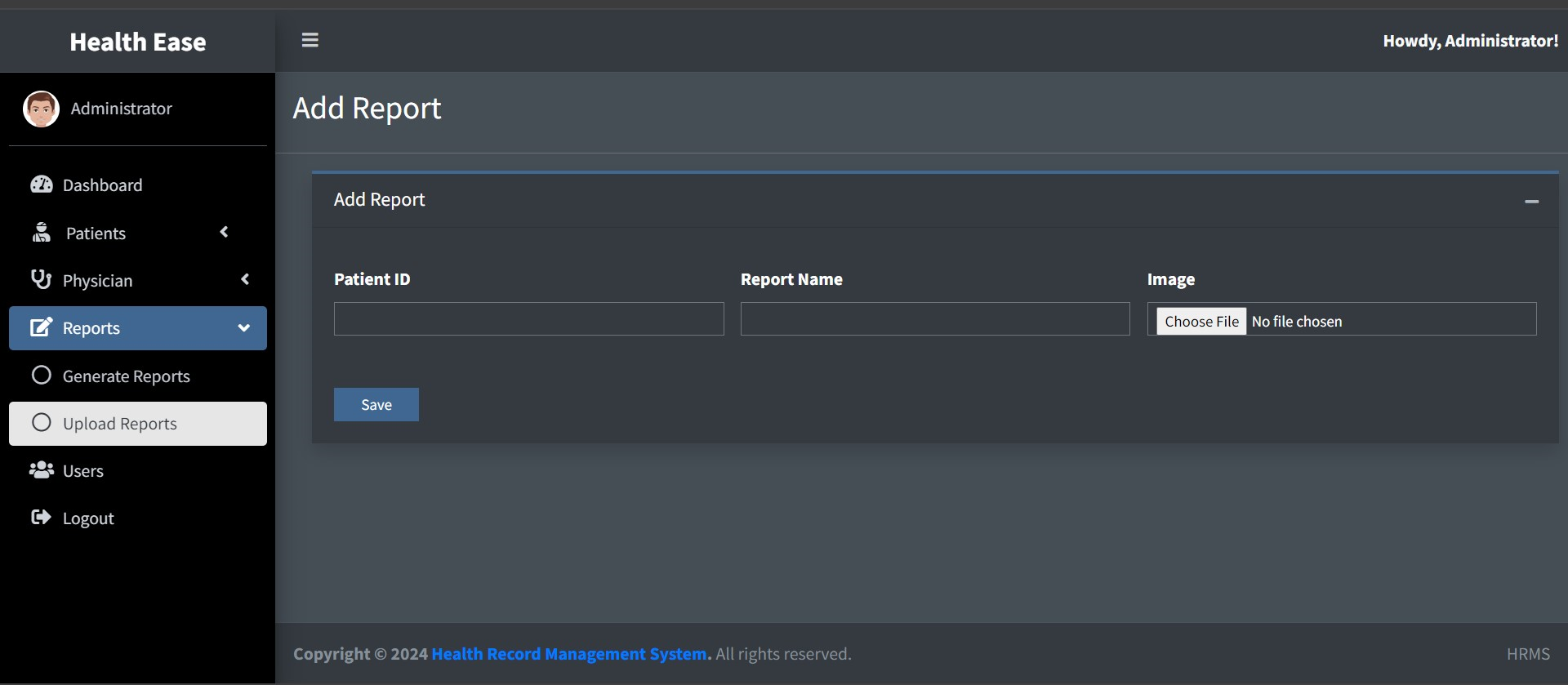
The patient addition interface, managed by the admin, serves as the portal for integrating new patients into the system. Admins utilize this interface to input comprehensive patient details, facilitating seamless incorporation of individuals into the healthcare management system. This feature empowers administrators to curate patient profiles while governing access to specific resources within the system, ensuring streamlined data input and appropriate categorization of patients within the healthcare network.



**Fig 4.8 Patient Registration Form**

**4.6.3 Report Uploading Form**

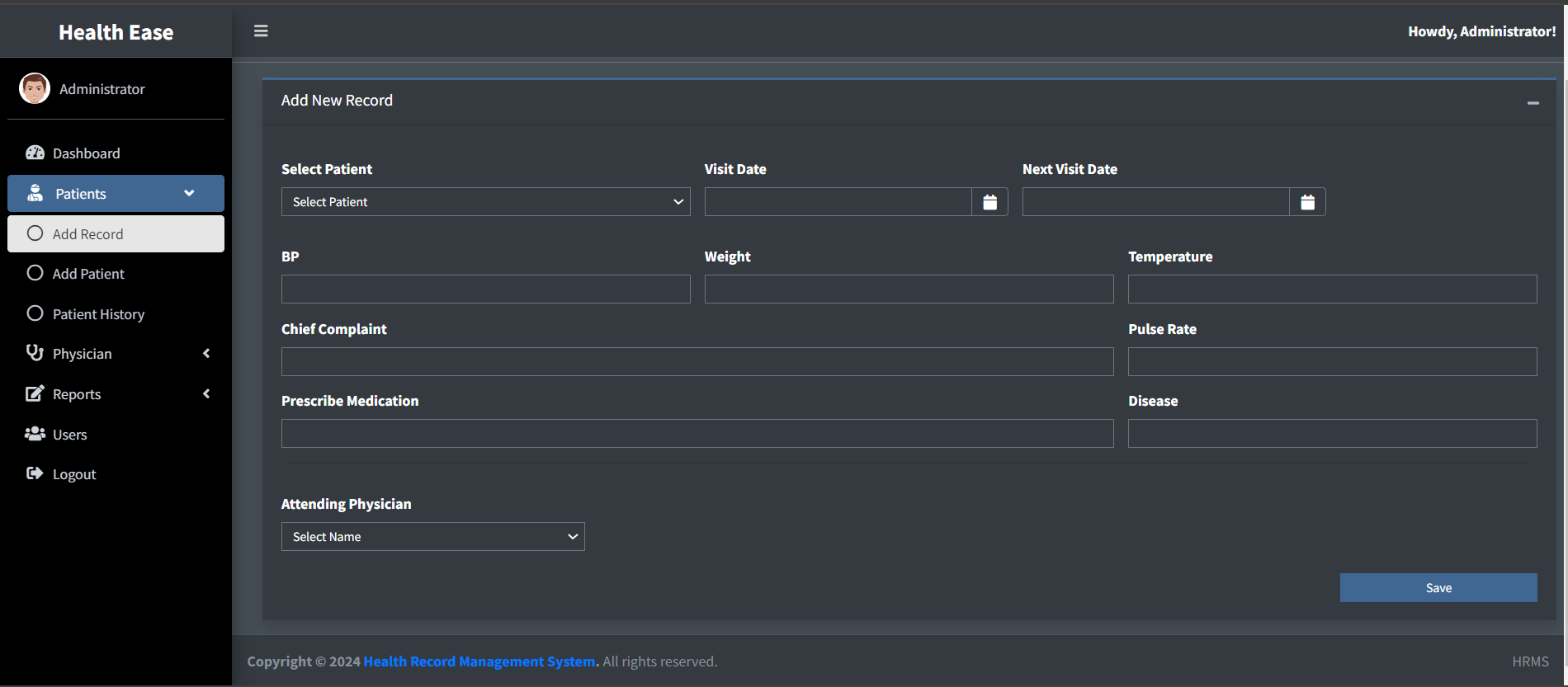
The report uploading feature, managed by the admin, serves as the gateway for integrating essential medical reports into the system. This functionality empowers administrators to securely upload and integrate diverse reports, enriching the system’s repository of patient information.



**Fig 4.9: Uploading Report form**

**4.6.4 Data Entry and Manipulation Forms**

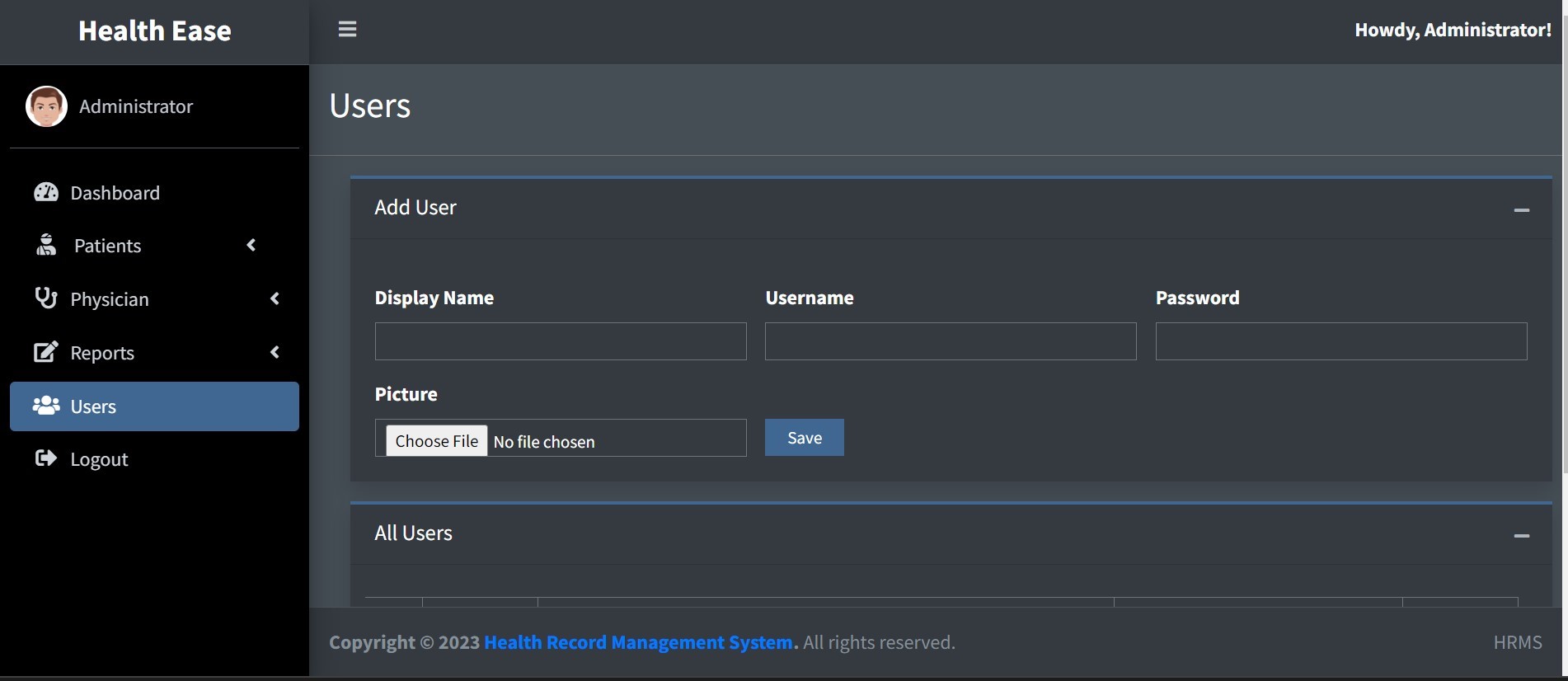
The patient data addition form offers administrators a structured interface to input comprehensive patient information into the system. This feature empowers admins to seamlessly add and organize patient data, ensuring a detailed and accurate database within the healthcare management system. By utilizing this form, administrators curate essential details, facilitating effective patient data management and enhancing the system's capacity to cater to individual healthcare needs.



**Fig 4.10: Add Health Record form**

**4.6.5 Adding User Form**

The user addition form serves as a gateway for authorized personnel to add new users into the system. This interface empowers designated administrators to securely integrate diverse users, ensuring controlled access and streamlined user management within the healthcare system.



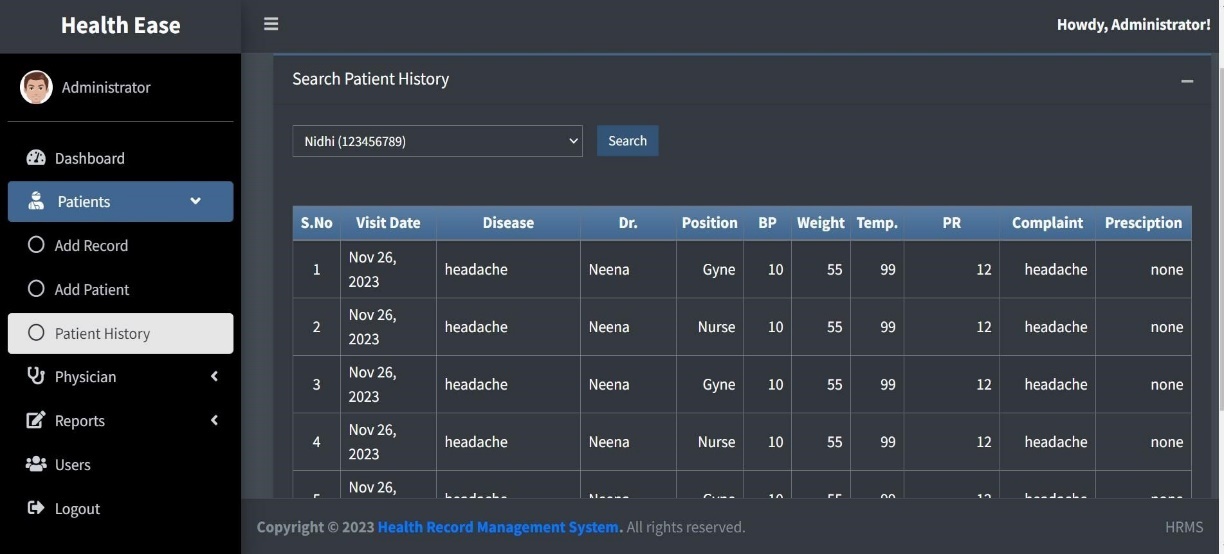
**Fig 4.11: Add User form**

## **4.7 DATA OUTPUTS**

Outputs are selected from the database based on a certain criteria and displayed using forms. The entire WPMS itself contains a number of forms, However, for the systems main components, below are some snap shots of the key forms.

**4.7.1 Data Storage Interface**

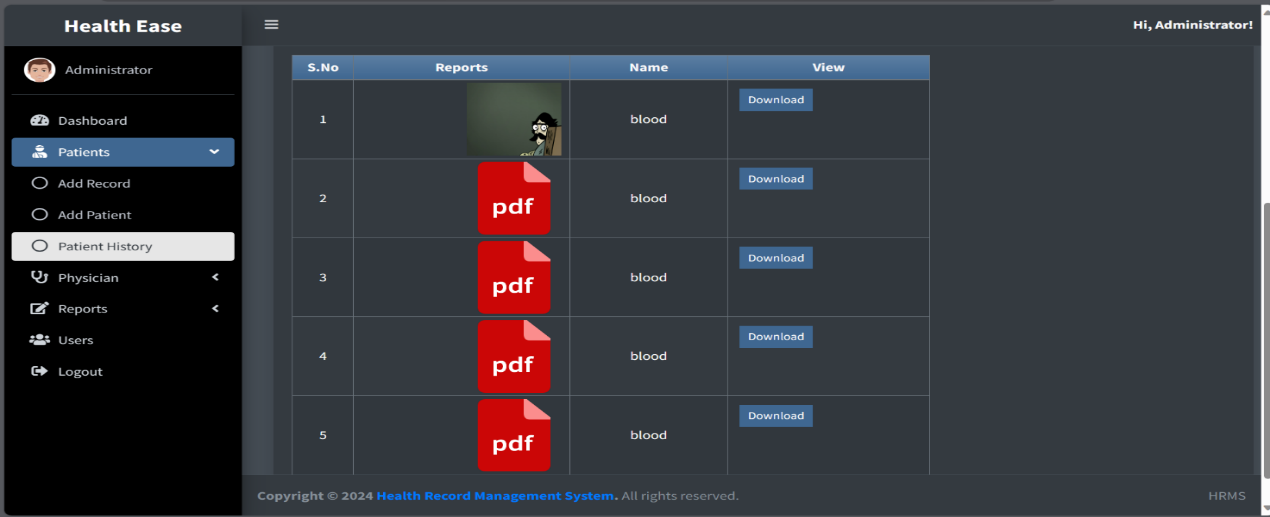
After the data in entered into the system, it is stored and can be retrieved at any time using the search functionality.



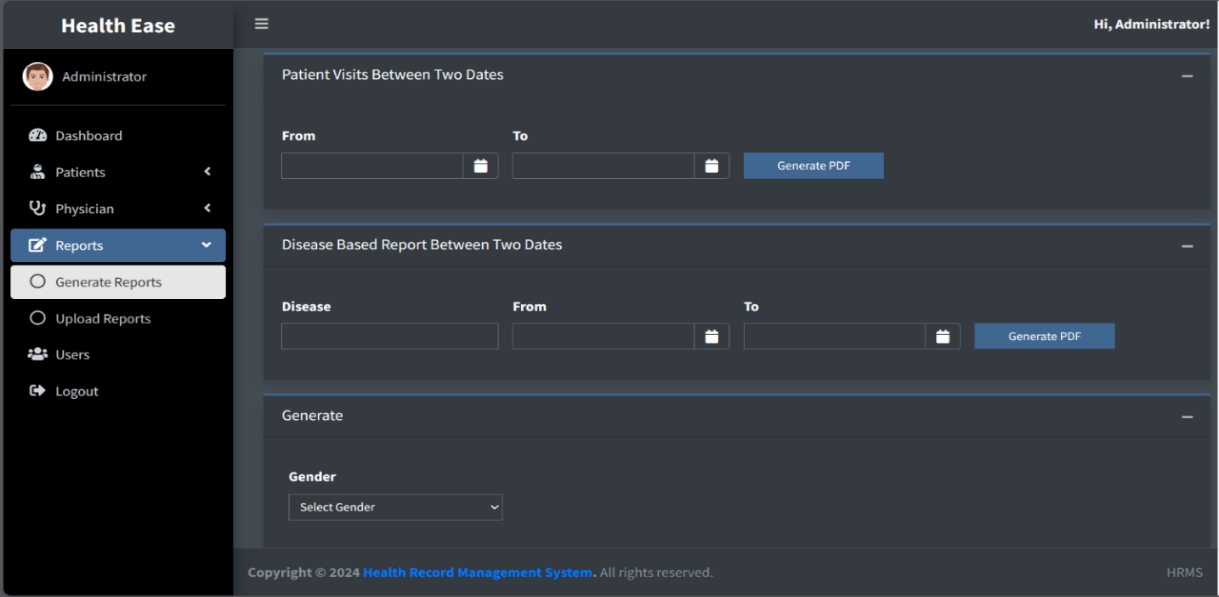
**Fig 4.12:** **Data** **Storage** **interface** **for** **Patient** **History**

**4.7.2 Data Reports**

The system was designed with a system of generating pdf reports for the records using the pdf package. This functionality was integrated in order to facilitate printing of the records in the system.



**Fig 4.13:** **Interface for viewing uploaded** **Report**

******

**Fig4.14:** **Interface** **to generate** **Report**

**4.8 Implementation & Testing**

**4.8.1 Implementation**

The implementation phase involves translating the design specifications into working code and deploying the system in the production environment. During this phase, developers write and integrate the backend and frontend components, configure the database, and ensure that the system functions according to the defined requirements.

Key activities during the implementation phase include:

* Writing code for backend functionalities using Java, JSP, and servlets.
* Developing frontend user interfaces using HTML, CSS, and JavaScript.
* Configuring and optimizing the MySQL database to store and manage data efficiently.
* Integrating various system components and ensuring seamless communication between them.

**4.8.2 Testing**

Testing is critical for a newly developed system as a prerequisite for it being put into an environment where the end users can use it. Exhaustive testing is conducted to ensure accuracy and reliability and to ensure that bugs are detected as early as possible. In the process of designing the EBS, three levels of testing were conducted, namely, unit testing, integration testing and system testing.

**4.8.2.1 Unit Test**

Unit test is where the system is tested partially and independently, component by component, to ensure that particular portion or module is workable within it. In the development of the records management system, each component was tested independently before finally integrating each of them into one system. This test was used by the researcher to verify that every input of data was assigned to the appropriate tables and fields.

Key activities during unit testing include:

* Writing test cases to cover different scenarios and edge cases.
* Executing test cases using testing frameworks such as JUnit.
* Analyzing test results and debugging code to fix defects.

**4.8.2.2 Integration Test**

Integration testing for the EBS (Wellness Program Management System) involved verifying the interaction and interoperability of different modules and functionalities to ensure they work together seamlessly.

Key activities during integration testing include:

* Identifying integration points between system components.
* Designing test cases to validate data exchange and interaction between modules.
* Executing integration tests and monitoring system behavior.
* Resolving integration issues and ensuring seamless communication between components.

**4.8.3.3 System Test**

System testing for the WPMS (Wellness Program Management System) involved comprehensive testing of the entire system as a whole to ensure its functionality, performance, and reliability.

Key activities during system testing include:

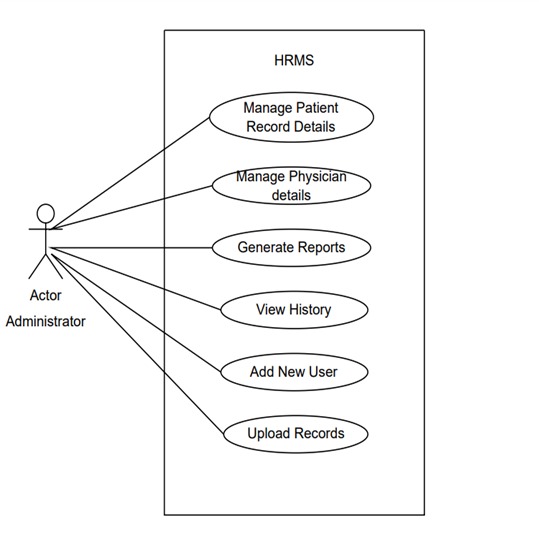
* Designing test scenarios to cover all system functionalities and use cases.
* Executing test cases to simulate real-world scenarios and user interactions.
* Performance testing to assess system responsiveness, scalability, and resource utilization.
* Usability testing to evaluate the user interface and overall user experience.
* Regression testing to ensure that new changes do not introduce unintended side effects.

**Results and Validation**

Outcome**:** The system testing phase successfully validated the functionalities, performance, usability, security measures, and robustness of the EBS.

Validation**:** The EBS demonstrated stability, meeting the defined criteria across all testing categories, ensuring a reliable and efficient system for managing HR-related tasks and generating reports.

## **4.9 USE CASE DIAGRAM**



**Fig 4.15: Use Case Diagram**

**CHAPTER 5**

**Evaluations & Conclusions**

**5.1 Evaluation**

In the attempt to evaluate the designed system, it is imperative that the researcher look back at the predefined functionalities, goals and objectives and analyse those in relation to the expectations met by the system. The Wellness Program Management System was evaluated based on the set of predefined objectives and expected functionalities it was able to fulfil. The Wellness Program Management System was designed to facilitate efficient records management in healthcare by providing an efficient, reliable computerized records management system and after a careful evaluation process; it met a considerable portion of those expectations.

The main objective was to design a system that enables faster and more efficient storage, retrieval and updating of hospital records. As far as this is concerned, the system met this expectation by giving direct benefit to the clinic such as fast records retrieval. Analysis was successfully completed. This evaluation is based on the fact that data requirements were collected that successfully enabled the design and development of the system.

The design objectives of creating an efficient records management system were further accomplished with the creation of add, delete, search and edit functionalities in the system that not only enable computerized but rather efficient, reliable and fast data entry. All these functionalities possess a relatively high level of accuracy. In evaluating this objective in relation to the system’s performance, it would therefore be accurate to state that it was achieved to a large extent.

## **5.2 Limitations of the System**

Throughout the development of the Wellness Program Management System, a few areas were overlooked. Some of these limitations can be presented as follows:

**Usability**

With regard to its use, the system only caters for English speakers. The GUI and associated documentation is in English. This may present a problem for non- English-speaking users

**Accessibility**

The system has only one user levels which only cater for the administrator. However, there is no facility for a guest, and data entrant. Such a facility would be useful if the patients themselves needed to access their electronic records via the system.

**Security**

The system also does not cater for the automatic back up of the data in the database. This may present a security problem in the event of data loss.

## **5.3 Problem Encountered**

**Wide project scope**

Defining the project scope was quite a challenge. This is because the system was meant to be designed for the entire hospital including all its departments, however with a view to the limited amount of time available for the project, the scope had to be narrowed down to one section of the hospital.

**Programming skills**

Learning PHP and MySQL requires considerable practice for one to gain the programming skills.

With limited knowledge and ability, the programming progress was rather slow and this limited the number of functionalities that the researcher could implement into the system.

## **5.4 Recommendations/Future Research**

As well as addressing the limitations presented in Section 4, there is scope for work to further the functionality and usefulness of this project. Therefore, the following recommendations for future enhancements to the system can be suggested

**Widening the scope**

Given the limited amount of time given to the developer, the project’s scope was rather limited to only one clinic in the hospital. The scope can further be widened to include all the other clinics to make a more integrated comprehensive system that covers the entire hospital’s records management

**Increased accessibility**

The system can also be further enhanced so that the patients themselves can be able to access their information online in a secure manner; this will lead to greater doctor-patient transparency

## **5.5 Conclusion**

In Conclusion, from a proper analysis and assessment of the designed system, it can be safely concluded that the system is an efficient, usable and reliable records management system. It is working properly and adequately meets the minimum expectations that were set for it initially.

## **References & Bibliography**

Craig, B. Central Children's Hospital Merger and Archives.

[4]. Iwhiwhu, B. A. The Management of Staff Records at Delta State University Library. Abraka, Nigeria.

[5]. Kalton, M. (1989). Survey Methods in Social Investigation (2nd Edition ed.). Hants, UK: Gower Publishing Company.

[6]. Kemoni, H. Managing Hospital Reords in Kenya- A Case Study of the Moi National Teaching and refferal Hospital, Eldoret. Eldoret, Kenya.

[7]. Patton, M. (1990). Qualitative Evaluation and Reserch Methods (2nd Edition ed.). Newbary Park, NewYork, USA: Sage Publications.

[8]. Roper, M. (2000). Managing Public Sector Records.

[9]. Taylor, J. (2004). Managing Information Technology Projects.

[10]. Wikipedia. (n.d.). Mbarara\_Hopital. Retrieved September 27th, 2010, from [http://www.wikipedia.org:](http://www.wikipedia.org/)<http://www.wikipedia.org/wiki/Mbarara_Hopital>

[11]. Yank, K. (February 2003). Build Your Own Database Driven Website Using PHP and MYSQL (Second Edition),. USA: SitePoint.

[12]. Erlandsson A. (April 1996) Electronic Records Management:- A Literature Review

[13]. Luciana Duranti and Heather MacNeil, “The Protection of the Integrity of Electronic Records: An Overview of the UBC-MAS Research Project”. December 1997)