**HEALTH RECORD MANAGEMENT SYSTEM**

**A PROJECT REPORT**

**for**

**Mini Project (KCA353)**

**Session (2023-24)**

**Submitted by:**

**Pooja Kumari**

(**2200290140109**)

**MEGHA CHAUDHARY**

**(2200290140091)**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Dr. Amit Kumar**

**(Assistant Professor)**



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(March 2024)**

**CERTIFICATE**

Certified that **Pooja Kumari (2200290140109), Megha Chaudhary (2200290140091)** have carried out the project work having “**HEALTH RECORD MANAGEMENT SYSTEM**” (**Mini 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 student himself/herself 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:**

**Pooja Kumari**

**(2200290140109)**

**Megha Chaudhary**

**(2200290140091)**

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

**Dr. Arun Tripathi**

**Head of Department**

**Department of Computer Applications**

**Kiet Group of Institutions, Ghaziabad**

**Dr. Amit Kumar**

**Assistant Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Health Record Management System**

**Pooja Kumari**

**Megha Chaudhary**

**ABSTRACT**

This project introduces a robust Human Resource Management System (HRMS) designed to streamline patient data management in healthcare settings. Leveraging XAMPP, PHP, MySQL, JavaScript, HTML, and CSS, the system offers functionalities for patient registration, health record maintenance, and PDF report generation.

The primary goal was to create an intuitive and efficient platform for healthcare providers to register patients, manage health records, and generate detailed reports. Through meticulous development and rigorous testing, the system ensures accuracy, reliability, and security.

The HRMS underwent extensive testing, including unit testing, integration testing, system testing, and user acceptance testing, validating its functionality, performance, security, and usability.

This abstract encapsulates the project's core objectives, key functionalities, rigorous testing, and its significance in enhancing patient data management within healthcare facilities.

**ACKNOWLEDGEMENTS**

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, Dr. AMIT KUMAR for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

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 on 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 with 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.

**Pooja Kumari**

**Megha Chaudhary**

**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 Patient Care |  |
|  | 1.5.3 Optimization |  |
|  | 1.5.4 Security and Data Accuracy |  |
|  | 1.5.5 Decision Support |  |
|  | 1.5.6 Adaptability and Scalability |  |
|  | 1.5.7 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 PHP 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 TABLES**

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

## 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 Report |  |
| 4.14 | Interface to generate Report |  |
| 4.15 | Use Case Diagram |  |

## ABBREVIATIONS & 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**

## OVERVIEW

Hospitals deal with the life and health of their patients. Good medical care relies on well-trained doctors and nurses and on high quality facilities and equipment. Good medical care also relies on good record keeping. Without accurate, comprehensive and up to date and accessible patient notes, medical personnel may not offer the best treatment or may in fact misdiagnose the condition, which can have serious consequences. Associated records, such as x-rays, specimens and patient registers, must also be well cared for if the patient is to be protected. Good records care also ensures the hospitals administration runs smoothly; unneeded records are transferred or destroyed regularly, keeping storage areas clear and accessible; and key records can be found quickly, saving time and resources. Records also provide evidence of the hospital’s accountability for its actions and they form a key source of data for medical research, statistical reports and health information systems.

Managing Health Records addresses the specific issues involved in managing clinical and non- clinical hospital records. Records Management refers to an on-going process of managing the records in a media neutral basis in accordance with approved policies, procedures and schedules. Records Management as a discipline defines and applies business rules related to the creation, protection, retrieval and disposition of an organization as records over time. Retention schedules are the cornerstone of a successful Records Management process.

Records Management as a discipline involves records keeping. Record keeping is an important aspect of every organizations/ institution’s day to day operations. There cannot be a records management system without records and neither can there be efficient record keeping without a good records management system. Therefore, record keeping is the [Systemati](http://www.businessdictionary.com/definition/systematic.html)[c procedure](http://www.businessdictionary.com/definition/procedure.html) by which the [records](http://www.businessdictionary.com/definition/record.html) of an [organization](http://www.businessdictionary.com/definition/organization.html) are created, captured, maintained, and disposed of. This [system](http://www.businessdictionary.com/definition/system.html) also [ensures](http://www.investorwords.com/1709/ensure.html) their preservation for evidential purposes, [accurate](http://www.businessdictionary.com/definition/accurate.html) and efficient updating, timely [availability,](http://www.businessdictionary.com/definition/availability.html) and [control](http://www.businessdictionary.com/definition/control.html) of [access](http://www.businessdictionary.com/definition/access.html) to them only by authorized [personnel.](http://www.businessdictionary.com/definition/personnel.html) The record in question here refers to any item or collection of data.

Therefore, a good and efficient records management system should be able to incorporate specific aspects of the systems mentioned above in order to provide and efficient means of records storage and management.

Keeping track of all activities and reports on paper is very inefficient and time consuming and also error prone. Keeping records on paper is a traditional base system that sometimes do not make it robust, in any case of damage all files will be lost that will cost a lot to the organization Day in and day out many people visit the hospital and when using the traditional base system, it makes it unreliable in the sense that it will take longer time to enter or access data and also maintaining. It is not economically and technically feasible to maintain these records on paper.

Health Record Management System (HRMS) is a comprehensive web-based application designed to streamline healthcare facilities and enhance health record management. This project aims to provide a centralized platform that allows administrators to efficiently manage medical records.

## BACKGROUND

Healthcare, as an industry, has historically relied on paper-based systems for managing patient health records. In the past, medical professionals meticulously maintained physical files containing vital patient information, diagnoses, treatment histories, and test results. However, these traditional methods posed significant challenges that impeded efficient healthcare delivery.

Challenges of Traditional Record Management:

* Accessibility and Retrieval: Paper-based records often suffered from delays and inefficiencies in retrieval, hindering timely decision-making by healthcare providers.
* Error-Prone Processes: Manual data entry and transcribing medical records were susceptible to errors, potentially leading to incorrect diagnoses or treatments.
* Limited Storage and Space: The physical storage of voluminous records consumed space and incurred maintenance costs within healthcare facilities.
* Security and Confidentiality: Patient confidentiality was at risk due to the vulnerability of paper records to loss, theft, or unauthorized access.

Amidst these challenges, the healthcare industry embraced technological advancements, leading to the evolution of digital solutions in record management. This shift laid the groundwork for Electronic Health Records (EHR) and subsequently, the development of Health Record Management Systems (HRMS).

## 1.3 PROBLEM STATEMENT

The system design and development were undertaken in order to eliminate the problem of redundant, erroneous and incomplete data that was escalating the inefficiencies in data retrieval. These limitations were mainly caused by the fact that data, under the previous manual recording system was entered into books and paper files and was later stored in overcrowded storage rooms that made retrieval of archival records close to impossible.

## 1.4 OBJECTIVE

**1.4.1 General Objective**

To design and develop a records management system that would enable faster and more efficient storage, retrieval and updating of hospital records.

**1.4.2 Specific Objectives**

The project’s specific objectives were:

* To carry out a feasibility study for the possibility of developing system
* To design and develop a records management system
* To test and validate the records management system
* To implement the records management system

## 1.5 SIGNIFICANCE

**1.5.1 Efficiency:**

PHP and MySQL Integration: Utilizing PHP and MySQL ensures streamlined data processing, quick retrieval, and efficient database management, leading to faster operations in managing patient records and administrative tasks.

Automation of Processes: By automating tasks like appointment scheduling and report generation, the system reduces manual effort and saves time for hospital staff, enhancing overall operational efficiency.

**1.5.2 Patient Care:**

Enhanced Accessibility: HTML, CSS, and JavaScript contribute to creating an intuitive and user-friendly interface, allowing admin to access their records, schedule appointments online, and communicate with healthcare providers easily, ultimately improving patient experience and satisfaction.

**1.5.3 Optimization:**

Resource Management: The system, powered by PHP and MySQL, aids in optimizing resource allocation by efficiently managing staff schedules and ensuring optimal utilization of hospital resources.

Streamlined Processes: Automation and data-driven insights provided by the system contribute to smoother operations, reducing redundancies and optimizing workflow across various hospital departments.

**1.5.4 Security and Data Accuracy:**

MySQL Database Security: Utilizing MySQL ensures robust data security features, encryption methods, and access controls, maintaining the confidentiality and integrity of patient data, complying with privacy regulations and ensuring accurate records.

Validation through PHP: PHP’s server-side validation techniques help maintain data accuracy by ensuring proper input formats and preventing errors or inconsistencies in the database, further enhancing the reliability of stored information.

**1.5.5 Decision Support:**

Real-time Data Access: The system enables healthcare professionals to access real-time patient data, medical histories, and analytics through PHP and MySQL, empowering informed decision-making for diagnoses, treatments, and healthcare planning.

**1.5.6 Adaptability and Scalability:**

Technology Flexibility: Utilizing PHP, JavaScript, HTML, CSS, and MySQL allows for a flexible and scalable architecture, enabling the system to adapt to evolving healthcare needs, incorporate future enhancements, and handle increased data volumes or user loads.

**1.5.7 Overall Impact:**

Positive Patient Outcomes: The combined effect of improved efficiency, better patient care, optimized processes, data accuracy, and decision support results in overall enhanced healthcare services, leading to improved patient outcomes, satisfaction, and trust in the hospital’s services.

## 1.6 SCOPE

The scope provides for the boundary of the research in terms of depth of investigation, content, and methodology, geographical and theoretical coverage.

The health records management system was designed in such a way that makes it possible to access it through any web browser programme. This serves as the user interface. The web browser supported interface created is dynamic and as a result backed by a database system that enables users to have the ability to input, access, manipulate and delete data from the database

**1.6.1 Functional Scope**

Using PHP in the backend allows for seamless integration of user authentication, data processing, and server-side logic. HTML/CSS are employed to create an intuitive and responsive user interface for various functionalities like appointment scheduling, patient records management, and staff administration.

**1.6.2 JavaScript 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

In order to understand the concepts associated with records management and or computer-based records management systems, it is imperative to examine and analyse published material from experts regarding the field. The purpose of this review is to analyse and examine and obtain experience as regards the creation and archival processing of electronic records. The review is based on an exhaustive assessment of the literature on computerized electronic management and electronic records, and contains an overview of the main concepts associated with the creation of an electronic records management system from the perspective of published experts.

The evolution of Health Record Management Systems (HRMS) has seen a transformative shift from traditional paper-based records to the adoption of electronic health records (HER). This transition has been facilitated by technologies like PHP, MySQL, HTML, CSS, and JavaScript. PHP, in conjunction with MySQL, has played a pivotal role in revolutionizing HRMS, offering a robust framework for efficient storage, retrieval, and management of patient data. These technologies have significantly enhanced the accessibility and usability of health records, surpassing the limitations of paper-based systems. HTML, CSS, and JavaScript have contributed to creating intuitive user interfaces, enabling seamless interaction and presentation of health data, thereby fostering a more comprehensive and user-friendly Health Record Management System.

## 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

In the context of a Hospital Management System, administering exclusive access rights to authorized personnel holds immense significance. PHP serves as a robust platform for implementing stringent user authentication and access control mechanisms. These features ensure that only authorized administrators have privileged access to critical functionalities and sensitive patient data.

The MySQL database, integral to the system’s architecture, supports secure storage of user roles, permissions, and access credentials. This amalgamation of PHP and MySQL allows for the establishment of a secure framework, maintaining the integrity and confidentiality of patient records, thereby preventing unauthorized access and ensuring data security.

**CHAPTER 3**

# METHODOLOGY

### **3.1 INTRODUCTION**

Methodology is a term used to describe a process, technique or manner in which an action is performed. Under the development a system, a methodology refers to the process that was taken to ensure that a system is effectively and efficiently developed.

The methodology employed in developing the Hospital Management System (HMS) constitutes a structured framework guiding the project’s lifecycle. This section delineates the multifaceted approach used to conceptualize, design, develop, and deploy the system, ensuring a systematic progression towards achieving predefined objectives.

### **SYSTEM DEVELOPMENT LIFECYCLE**

**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.

* + 1. **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:

The characteristics of the proposed system were defined and identified

* + 1. **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 Modeling 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.

* + 1. **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.

* + 1. **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.

* + 1. **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 Hospital Records 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 Health Record Management System (HRMS) on a local environment requires specific considerations to ensure secure and convenient access for multiple administrators.

**4.1.1.1 Local Development Environment**

* Local Server Configuration: The HRMS operates within a local development environment, hosted on a local server like XAMPP, WAMP, or other development platforms.
* Access via Localhost: Administrators can access the HRMS by typing "localhost" followed by the designated port number in their web browser's address bar.
* Local File Structure: The system's files and databases are stored locally on the development machine, enabling admins to interact with the system within the local environment.

**4.1.1.2 User Authentication for Local Access**

* Admin Credentials: Unique login credentials, including usernames and passwords, are provided to authorized administrators to access the HRMS in the local environment.
* Local Authentication System: The system's authentication process is confined within the local environment, requiring valid credentials to grant access to the HRMS.
* Session Handling: Secure session management within the local environment ensures authenticated access during the admin's interaction with the HRMS.

**4.1.1.3 Local Environment Accessibility**

* Device Compatibility: The HRMS is designed to be accessible across various devices within the local environment, supporting desktops, laptops, and other compatible devices.
* Local Browser Support: Compatibility with common browsers within the local environment ensures administrators can access the system using preferred browsers.
* Local Network Considerations: Access to the HRMS is limited to the local network, restricting access outside the local environment during the development phase.

**4.1.2. User Privileges**

The HRMS 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:

* + 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.

* + 1. **Software Specifications**
* **XAMPP Installation:**

Download and install the latest version of XAMPP compatible with your operating system.

* **Web Browser:**

Latest versions of browsers like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge for accessing the HRMS web application.

* **Code Editor:**

Optional: Install a code editor like Visual Studio Code, Sublime Text, or PHPStorm for code customization or review.

**Other Considerations**

* **User Account:**

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

* **Internet Connectivity (Optional):**

Stable internet connection may be required for certain functionalities, such as remote access or cloud-based features (if implemented).

## 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. **PHP Programming Language**

PHP acts as the bridge between the MYSQL database and the user interface modules. It facilitates seamless interaction with the database, enabling data manipulation, validation, and logic implementation. PHP also handles server-side scripting, ensuring secure and dynamic content generation.

**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.

* + 1. **Frontend Interface Modules**
* **HTML (Hyper Text Markup Language)**

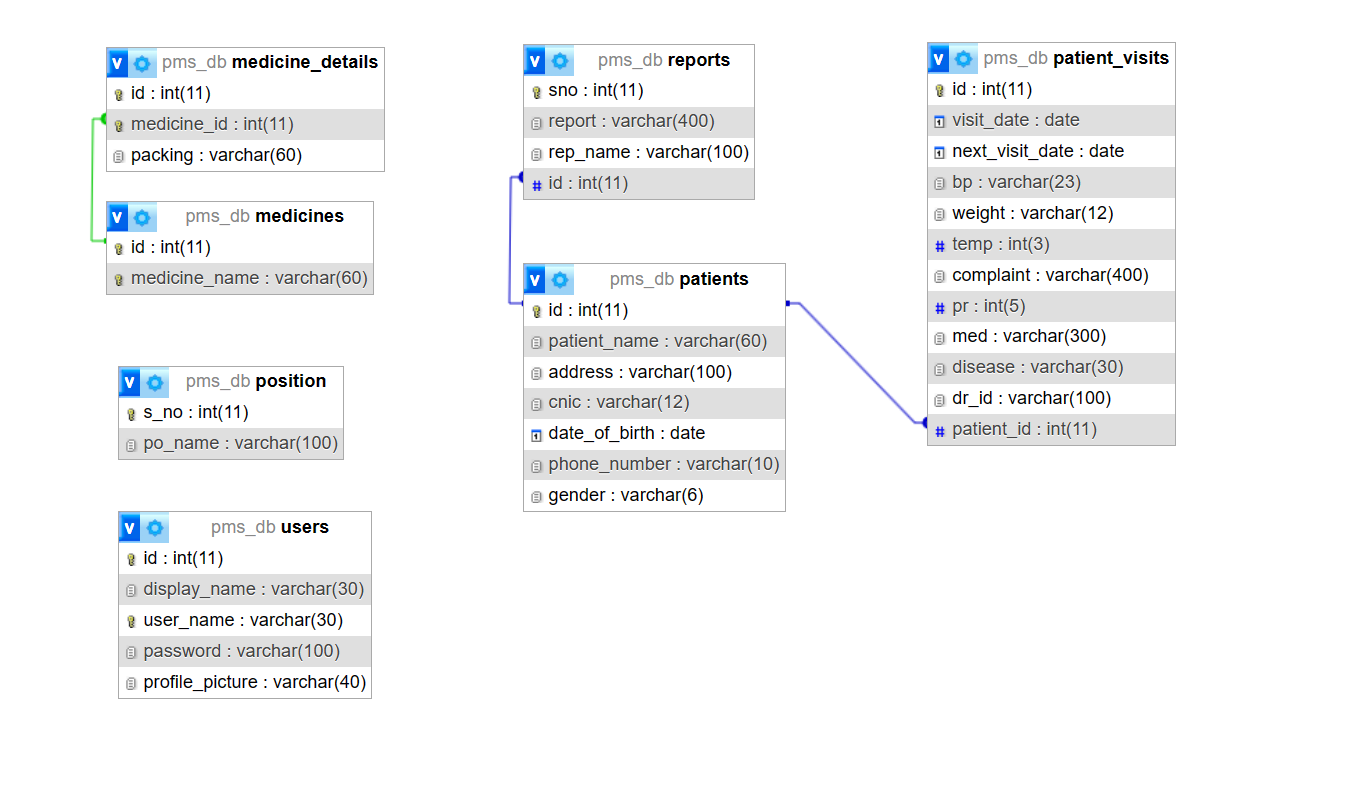
HTML forms the structural foundation of the user interface, defining the layout and structure of web pages. It provides the framework for presenting data and interacting with the system.

* **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 Health record 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**

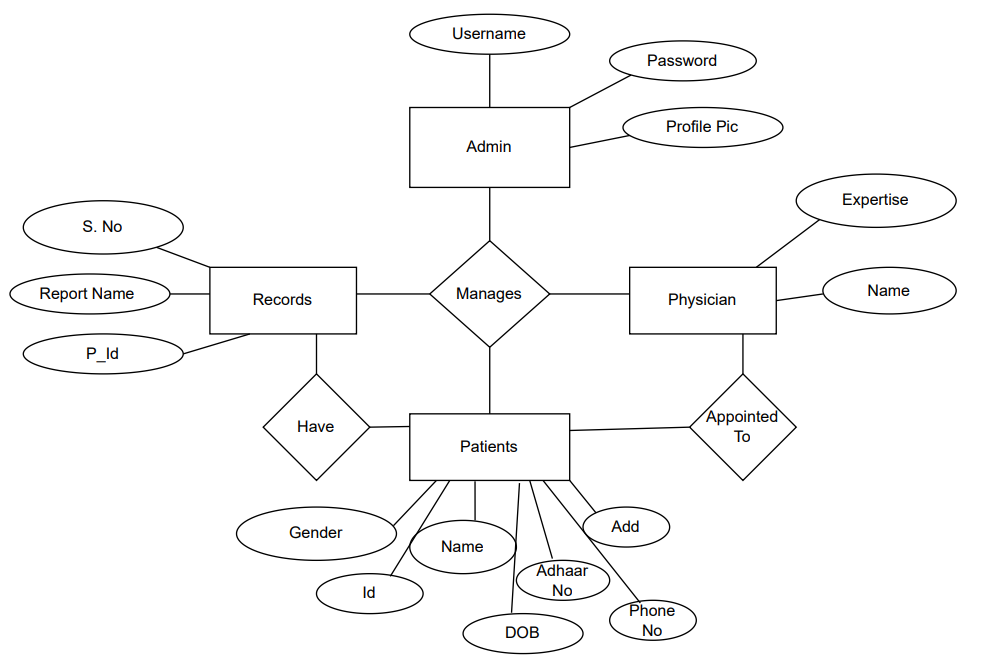
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Sno | Int | 11 | Auto increment, primary key |
| Report | Varchar | 400 | Report location |
| Rep\_name | Varchar | 100 | Type of report |
| id | int | 11 | Foreign key, Patient id |

#### **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 |

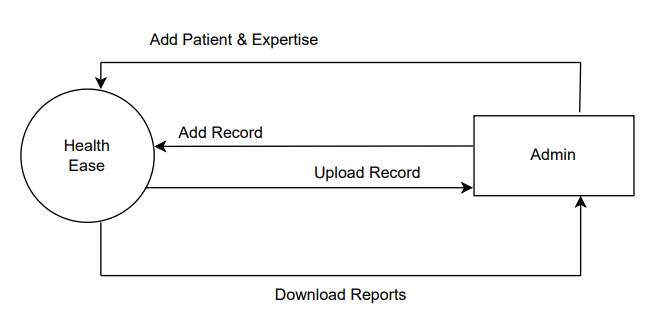
## 4.4 ER DIAGRAMS & DFDs

**4.4.1 ERD (Entity Relationship Diagram)**

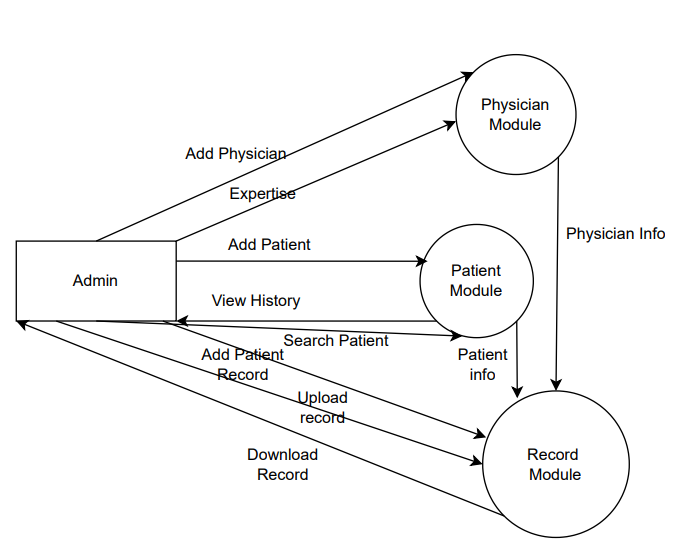
****

**Fig** **4.2:**  **ER Diagram**

**4.4.2 DFD (Data Flow Diagram)**

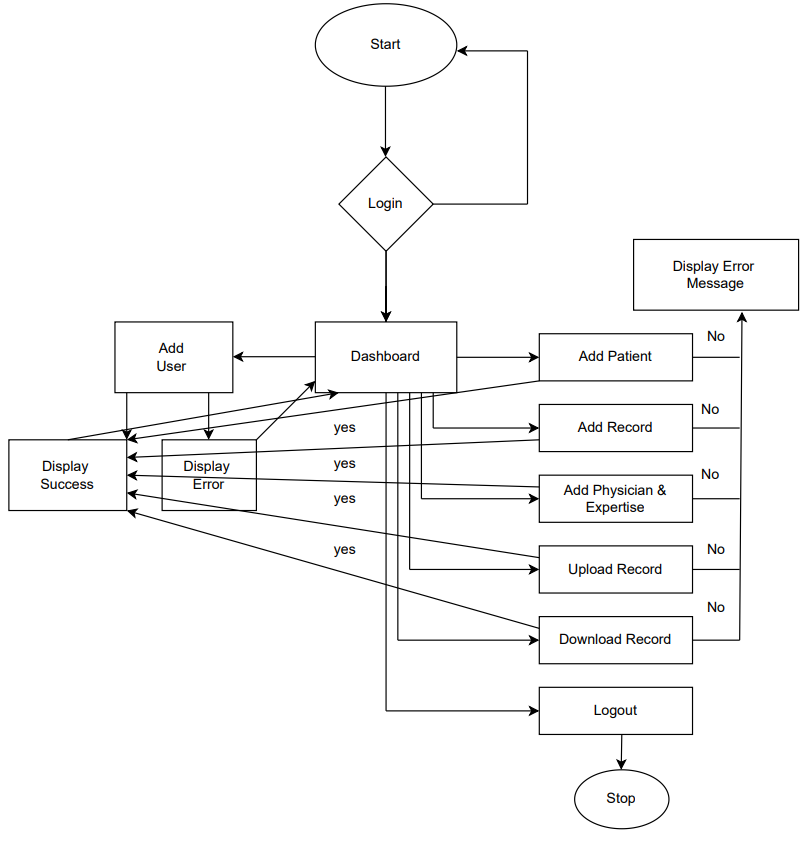
****

**Fig** **4.3:** **DFD Level-0**

******

**Fig** **4.4:**  **DFD Level-1**

## 4.5 SYSTEM FLOW CHART



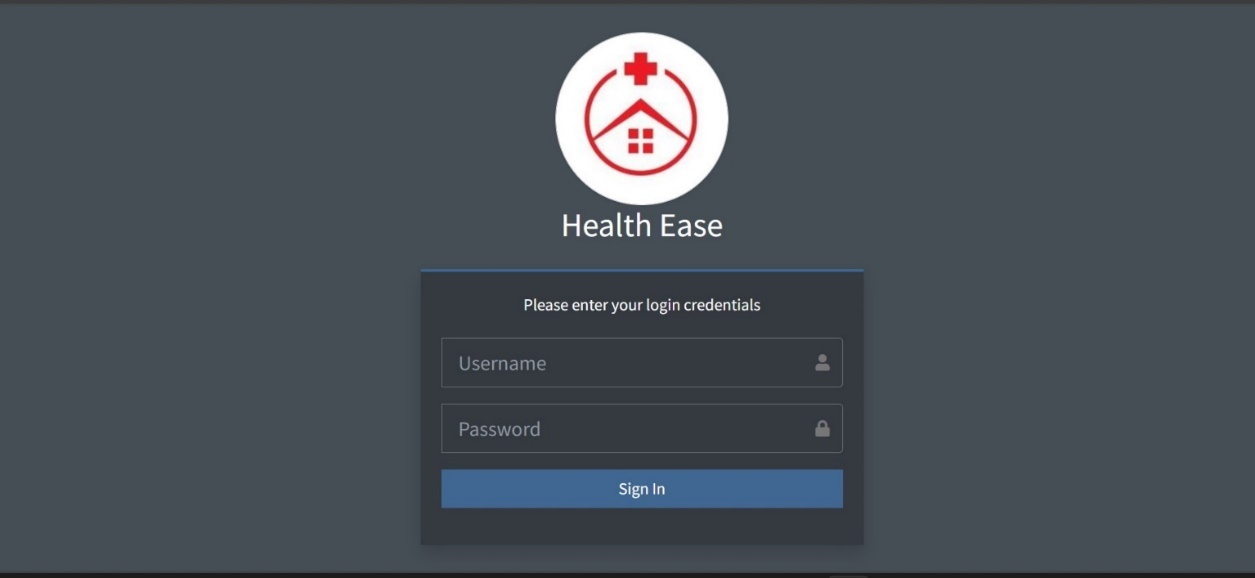
**Fig** **4.6:**  **System Flow Chart**

## 4.6 DATA INPUTS

Outputs are selected from the database based on a certain criterion and displayed using forms. The entire HRMS 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**

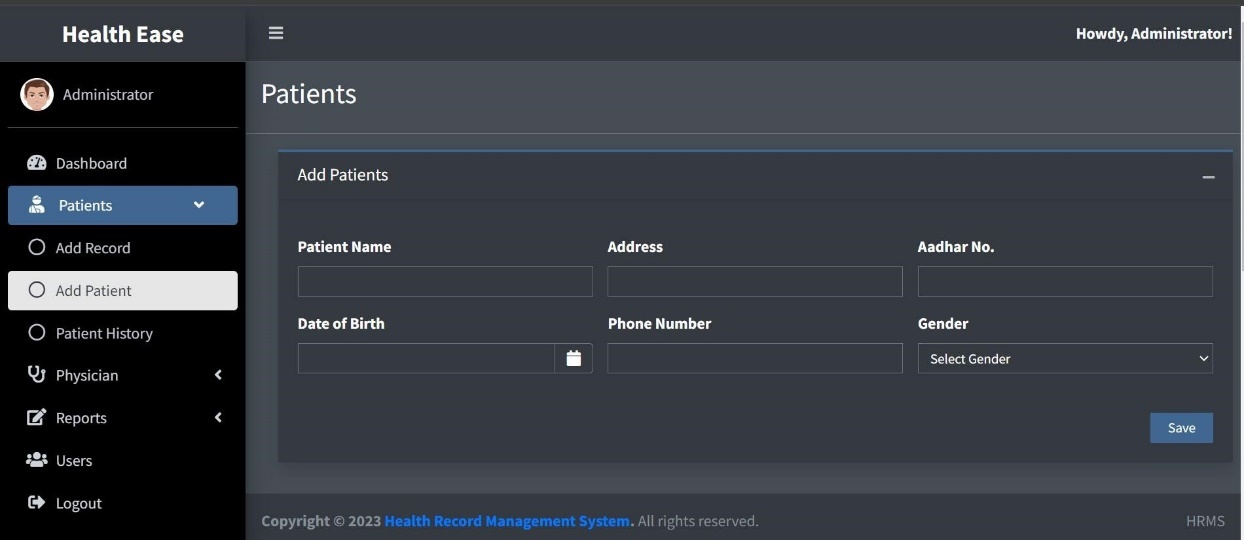
The login form above is the first page a person accessing the system sees. It is used to gain access to the system resources and determines, based on the user type, which users should acess which resources



**Fig 4.7: Login form**

**4.6.2 Patient Registration Form**

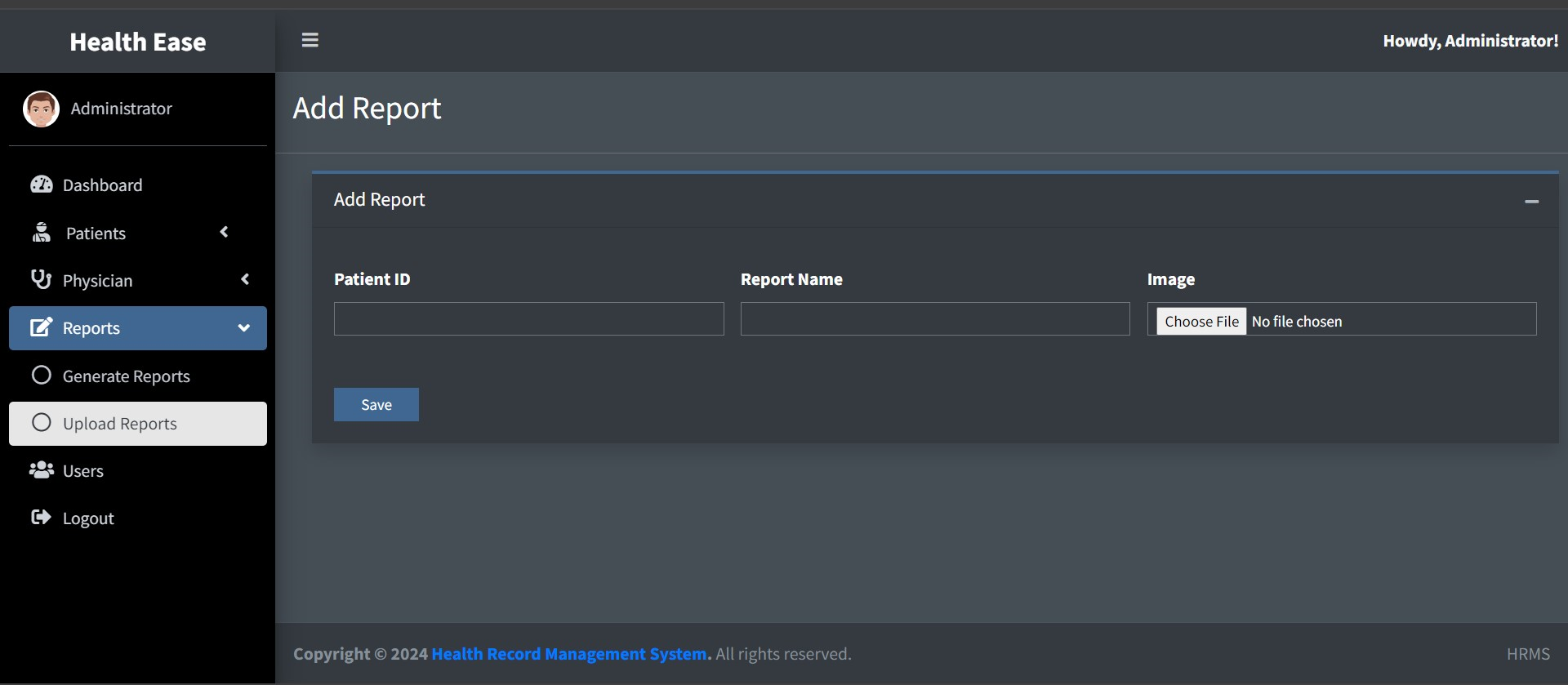
The patient addition interface, managed by the admin, serves as the portal for integrating new patients into the system. Admins utilize this interface to input comprehensive patient details, facilitating seamless incorporation of individuals into the healthcare management system. This feature empowers 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**

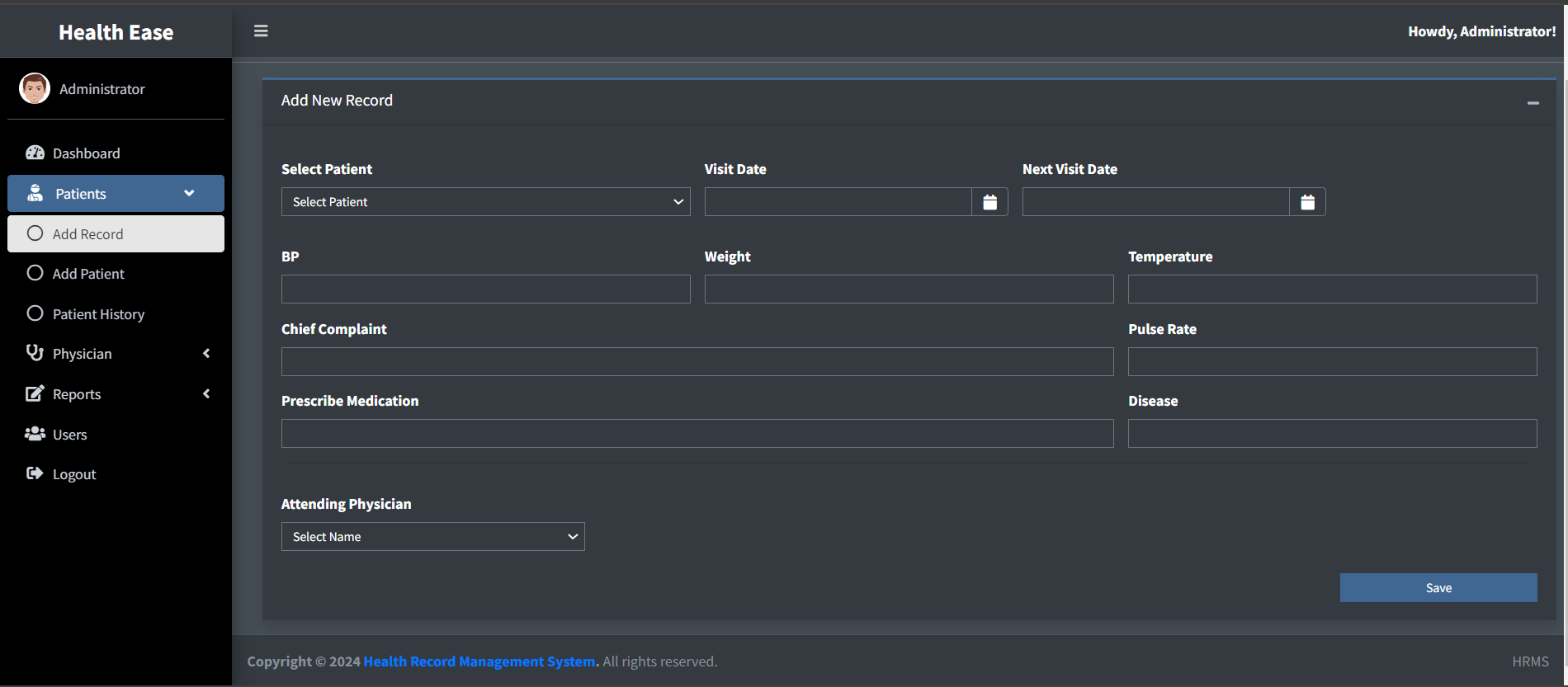
The report uploading feature, managed by the admin, serves as the gateway for integrating essential medical reports into the system. This functionality empowers administrators to securely upload and integrate diverse reports, enriching the system’s repository of patient information.



**Fig 4.9: Uploading Report form**

**4.6.4 Data Entry and Manipulation Forms**

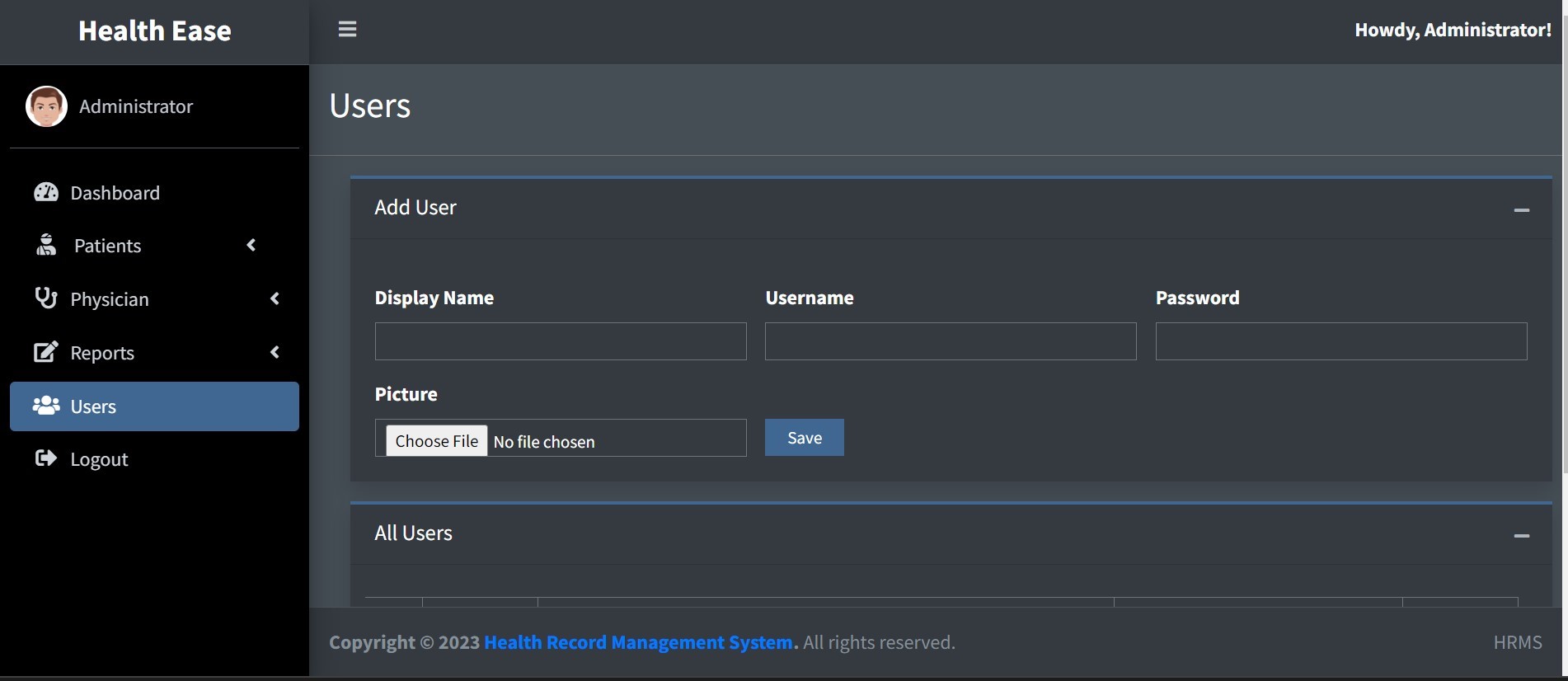
The patient data addition form offers administrators a structured interface to input comprehensive patient information into the system. This feature empowers admins to seamlessly add and organize patient data, ensuring a detailed and accurate database within the healthcare management system. By utilizing 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.

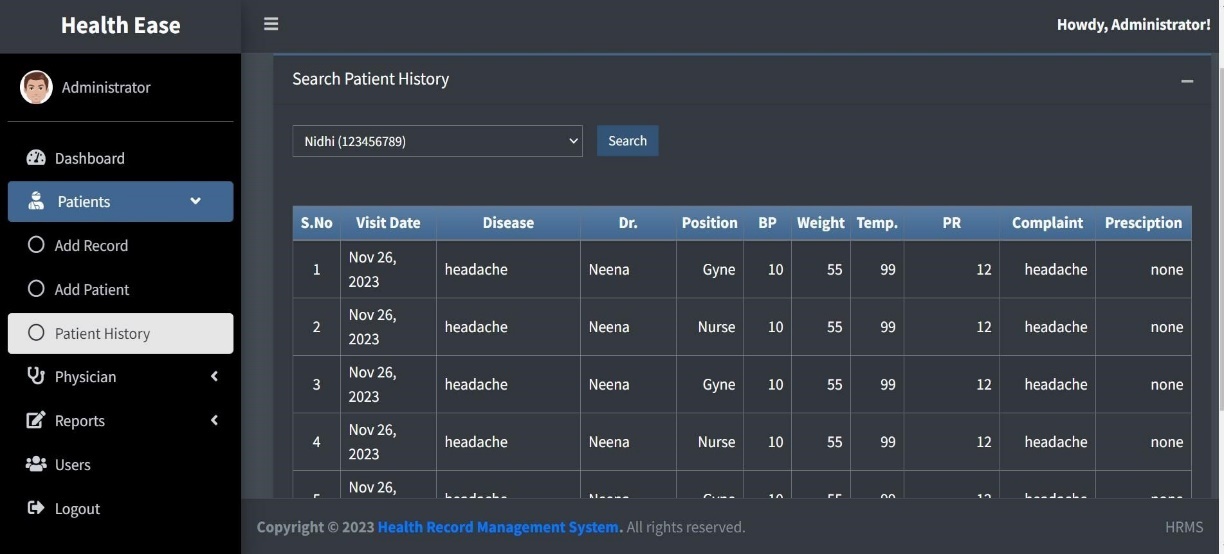


**Fig 4.11: Add User form**

## 4.7 DATA OUTPUTS

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

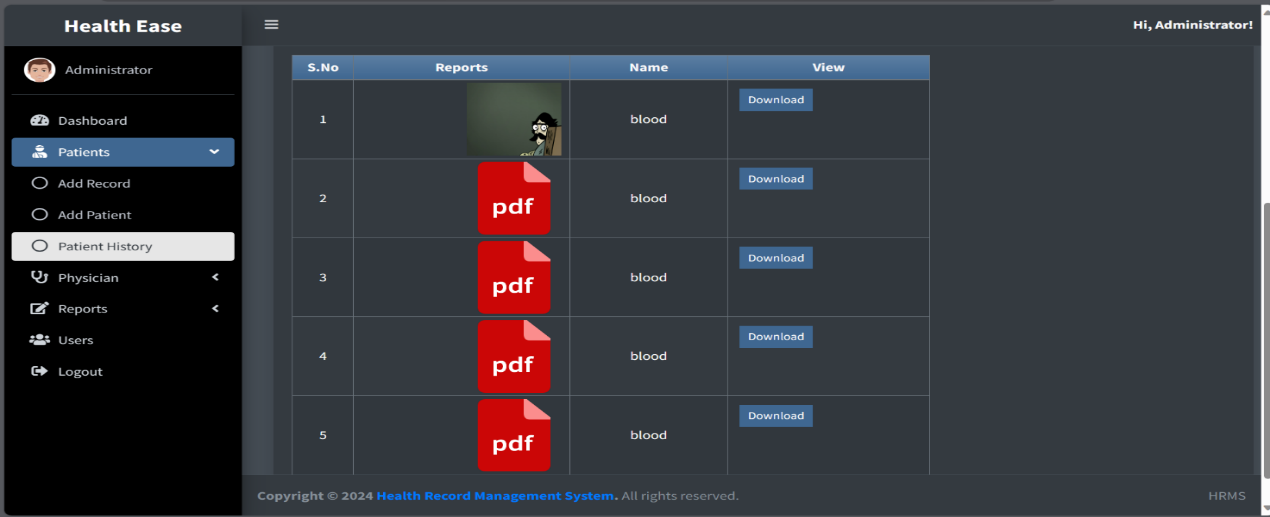
**4.7.1 Data Storage Interface**

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

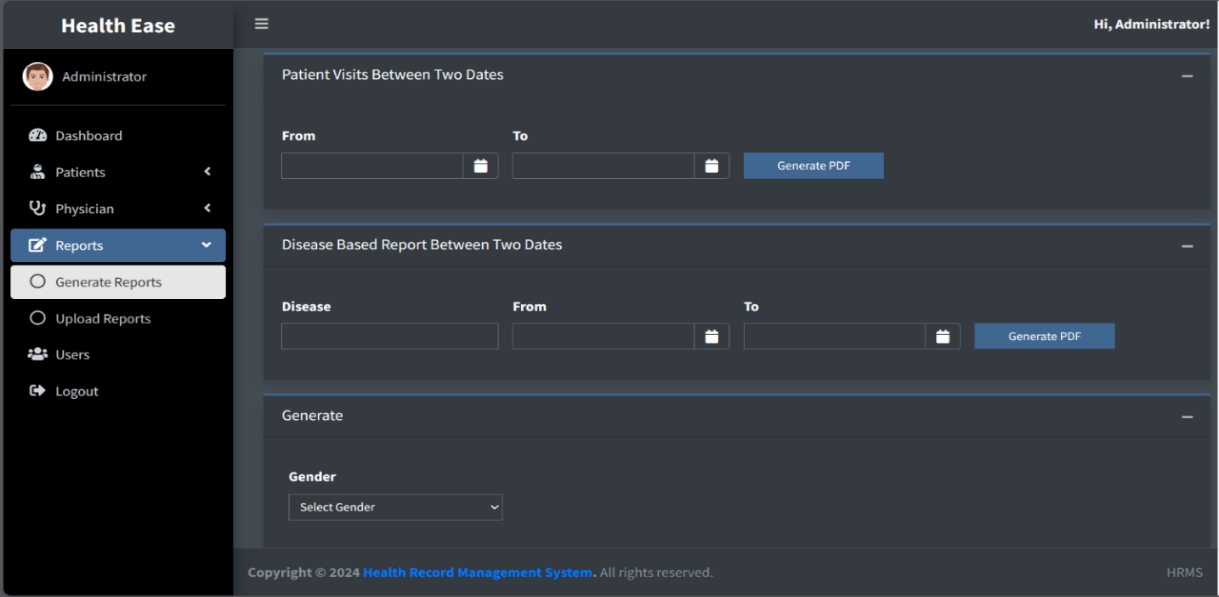
**Fig 4.12:** **Data** **Storage** **interface** **for** **Patient** **History**

**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.



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

******

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

## 4.8 IMPLEMENTATION & TESTING

**4.8.1 Implementation**

* **Overview of Technologies Used**

The HRMS (Human Resource Management System) was developed using a combination of XAMPP, PHP, MySQL, JavaScript, HTML, and CSS. Each technology played a crucial role in the development and functionality of the system.

* **Development Environment Setup**

The initial phase involved setting up a local development environment using XAMPP. This provided a server environment comprising Apache, MySQL, PHP, and Perl, facilitating seamless testing and development.

* **Database Design**

The backbone of the HRMS relied on a well-structured database. The design incorporated multiple tables enabling efficient data storage and retrieval.

* **Backend Development (PHP and MySQL)**

PHP was extensively utilized for server-side scripting, handling database interactions, and business logic implementation. MySQL was the chosen database management system, ensuring robust data management. PHP scripts were developed to manage various HRMS functionalities:

Patient Registration: Created PHP scripts to capture and store Patient details in the database.

Health Record: Implemented functionality to record and update health records based on timestamps.

* **Frontend Development (HTML, CSS, JavaScript)**

The user interface was designed using HTML for structure, CSS for styling, and JavaScript for interactivity. Key aspects of the frontend development included:

User-friendly Interface: Designed intuitive layouts and forms for easy navigation and data input.

Responsive Design: Ensured compatibility across different devices and screen sizes for enhanced accessibility.

Interactive Elements: Implemented JavaScript functionalities for dynamic features such as real-time data updates and form validations.

**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 HRMS, 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

**Patient Registration Module**

**Objective:** Ensure accurate registration and storage of patient data into the database.

**Test Cases:**

* Valid Registration**:** Test the registration process with valid patient details (name, contact, address) and verify if data is correctly stored in the database.
* Duplicate Prevention: Test if the system prevents duplicate registrations for patients already existing in the database based on unique identifiers (e.g., unique ID or contact number).
* Data Validation: Test the system's response to incomplete or invalid data entries during registration to ensure proper error handling and validation messages.

**Health Record Addition Module**

**Objective:** Validate the functionality to add health records associated with registered patients.

**Test Cases:**

* Successful Record Addition: Test adding health records for various patients and verify if the data is stored accurately, linking to the respective patient IDs.
* Data Integrity Check: Ensure that the health records correctly link to the registered patient and cross-verify the data consistency.
* Upload Validation: Test uploading documents in both JPG and PDF formats to confirm successful storage and retrieval.

**Viewing Patient Data Module**

**Objective:** Confirm the accurate display of patient details and health records.

**Test Cases:**

* Data Retrieval: Verify the retrieval of all patient data including personal details and associated health records.
* Search Functionality: Test the search feature using various criteria (name, ID) to ensure correct retrieval of specific patient information.
* Document Accessibility: Ensure that uploaded documents (in JPG or PDF) are accessible and can be viewed/downloaded.

**PDF Report Generation Module**

**Objective:** Validate the creation and accuracy of PDF reports.

**Test Cases:**

* Report Generation: Test the functionality to generate reports containing patient details and health records in PDF format.
* Report Content Verification: Verify the content of the generated PDF reports to ensure it accurately reflects the patient data and health records.
* PDF Accessibility: Confirm that the generated reports are easily accessible and can be viewed/downloaded without any issues.

**4.8.2.2 Integration Test**

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

**Patient Registration and Health Record Addition Integration**

**Objective:** Test the integration between the patient registration and health record addition modules.

**Test Cases:**

* Registration and Record Addition: Test the flow from registering a patient to adding health records for the same patient. Verify if the health records correctly link to the registered patient.
* Data Consistency: Ensure that the data entered during registration aligns accurately with the health records added subsequently.

**Viewing Patient Data and PDF Report Generation Integration**

**Objective:** Validate the integration between viewing patient data and generating PDF reports.

**Test Cases:**

* Data Retrieval for Reports: Test the functionality to generate PDF reports containing patient details and health records. Verify if the data extracted for the reports matches the information displayed in the patient data view.
* Report Generation from Different Data Sets: Generate PDF reports for various patient data subsets and verify the accuracy of the reports in reflecting the respective patient details and health records.

**4.8.3.3 System Test**

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

**Functional Testing**

**Objective:** Validate the system's functionalities in line with the specified requirements.

**Test Cases:**

* End-to-End Functionality: Test the entire HRMS workflow from user authentication to generating PDF reports.
* Use Case Scenarios: Test different use cases such as registering new patients, adding health records, viewing patient data, and generating reports.

**Usability and User Interface Testing**

**Objective:** Evaluate the system's usability and user interface for ease of navigation and interaction.

**Test Cases:**

* User Experience Testing: Assess the intuitiveness of the system in performing their tasks.

**Security Testing**

**Objective:** Verify the system's security measures to protect sensitive data.

**Test Cases:**

* Authentication and Authorization: Test user authentication and access control mechanisms to ensure data privacy and security.
* Data Encryption: Verify the encryption methods used for sensitive data storage and transmission.

**Robustness and Error Handling**

**Objective:** Test the system's resilience to errors and its error-handling capabilities.

**Test Cases:**

* Input Validation**:** Test for invalid inputs to ensure the system handles them gracefully without crashing.

**Test Environment**

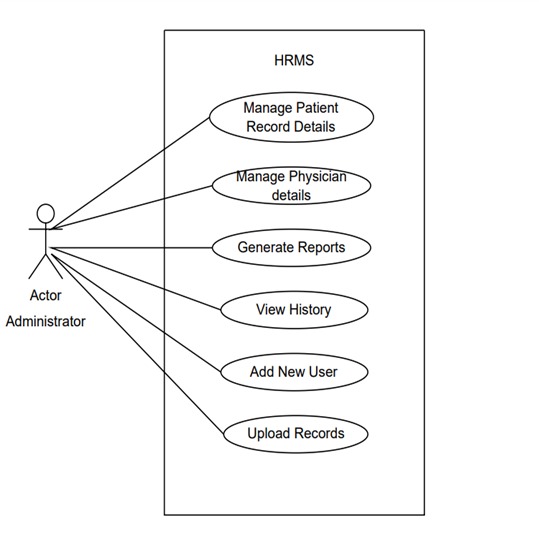
System testing was conducted in an environment that closely resembled the production environment, ensuring realistic testing scenarios and conditions.

**Results and Validation**

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

Validation**:** The HRMS 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 USER 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 Health Records Management System was evaluated based on the set of predefined objectives and expected functionalities it was able to fulfil. The Health Records 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 SYSTEM

Throughout the development of the Health Record 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).