**Wellness Program Management System**

**A PROJECT REPORT**

**for**

**Project (KCA451)**

**Session (2023-24)**

**Submitted by**

**Tulika**

**2200290140162**

**Vartika Khare**

**2200290140175**

**Unnati Rastogi**

**2200290140168**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Dr. Sangeeta Arora**

**Associate Professor**



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(May-2024)**

**DECLARATION**

I hereby declare that the work presented in report entitled “Wellness Program Management 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:** Tulika

**Roll No.:** 2200290140162

**Name:** Vartika Khare

**Roll No.:** 2200290140175

**Name:** Unnati Rastogi

**Roll No.:** 2200290140168

**CERTIFICATE**

Certified that **Tulika 220029014094632, Vartika Khare 220029014097627, Unnati Rastogi 2200290140168** have carried out the project work having “**Wellness Program Management System**” ( **Project-KCA451**) 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:**

**Tulika 2200290140162**

**Vartika Khare 2200290140175**

**Unnati Rastogi 2200290140168**

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

**Date:**

**Dr. Sangeeta Arora Dr. Arun Tripathi**

**Associate Professor Head**

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

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

**Wellness Program Management System**

**Tulika**

**Vartika Khare**

**Unnati Rastogi**

**ABSTRACT**

This project introduces a robust Wellness Program Management System (WPMS) 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 WPMS 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 thesis supervisor, Dr. Sangeeta Arora (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.

Tulika (2200290140162)

Vartika Khare (2200290140175)

Unnati Rastogi (22002901401)

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | Certificate | iii |
|  | Abstract | iv |
|  | Acknowledgement | v |
|  | Table of Content | vi - viii |
|  | List of Figures | ix |
|  | List of Tables | x |
|  | Abbreviation & Acronym | xi |
| **1** | **INTRODUCTION** | 1 - 6 |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 BACKGROUND | 2 |
|  | 1.3 PROBLEM STATEMENT | 3 |
|  | 1.4 OBJECTIVE | 3 |
|  | 1.4.1 General Objective | 3 |
|  | 1.4.2 Specific Objective | 3 |
|  | 1.5 SIGNIFICANCE | 3 |
|  | 1.5.1 Efficiency | 3 |
|  | 1.5.2 Patient Care | 4 |
|  | 1.5.3 Optimization | 4 |
|  | 1.5.4 Security and Data Accuracy | 4 |
|  | 1.5.5 Adaptability and Scalability | 4 |
|  | 1.5.6 Overall Impact | 5 |
|  | 1.6 SCOPE | 5 |
|  | 1.6.1 Functional Scope | 5 |
|  | 1.6.2 JavaScript Integration | 5 |
|  | 1.6.3 MySQL Database | 5 |
|  | 1.6.4 Technology Constraints | 6 |
| **2** | **LITERATURE REVIEW** | 7 - 8 |
|  | 2.1 OVERVIEW | 7 |
|  | 2.2 RECORDS & ELECTRONIC RECORDS | 7 |
|  | 2.3 DATABASES AS RECORD KEEPING SYSTEM | 8 |
|  | 2.4 IMPORTANCE OF ADMIN-EXCLUSSIVE ACCESS | 8 |
| **3** | **METHODOLOGY** | 9 - 11 |
|  | 3.1 INTRODUCTION | 9 |
|  | 3.2 SYSTEM DEVELOPMENT LIFECYCLE | 9 |
|  | 3.2.1 Requirement Analysis | 9 |
|  | 3.2.2 Planning | 9 |
|  | 3.3.3 System Design | 10 |
|  | 3.3.4 Technology Selection | 10 |
|  | 3.3.5 Development | 10 |
|  | 3.3.6 Testing & Quality Assurance | 10 |
| **4** | **SYSTEM DESCRIPTION** | 12 - 34 |
|  | 4.1 SYSTEM OVERVIEW | 12 |
|  | 4.1.1 Accessing the System | 12 |
|  | 4.1.1.1 Local Development Environment | 12 |
|  | 4.1.1.2 Authentication for Local Access | 13 |
|  | 4.1.1.3 3 Local Environment Accessibility | 13 |
|  | 4.1.2 User Privileges | 13 |
|  | 4.2 SYSTEM REQUIREMENTS | 14 |
|  | 4.2.1 Hardware Specifications | 14 |
|  | 4.2.2 Software Specifications | 14 |
|  | 4.3 SYSTEM ARCHITECTURE | 15 |
|  | 4.3.1 Backend Engine | 15 |
|  | 4.3.1.1 MySql Database | 15 |
|  | 4.3.1.2 PHP Programming Language | 15 |
|  | 4.3.1.3 Apache Web Server | 15 |
|  | 4.3.2 Frontend Interface Module | 15 |
|  | 4.3.3 Logical System Design | 16 |
|  | 4.3.4 Physical System Design | 17 |
|  | 4.4 ERD & DFD | 21 |
|  | 4.4.1 ER Diagram | 21 |
|  | 4.4.2 DFD Diagram | 22 |
|  | 4.5 SYSTEM FLOWCHART | 23 |
|  | 4.6 DATA INPUT | 24 |
|  | 4.6.1 Login Form | 24 |
|  | 4.6.2 Patient Registration Form | 24 |
|  | 4.6.3 Report Uploading Form | 25 |
|  | 4.6.4 Data Entry & Manipulation Form | 26 |
|  | 4.6.5 Adding user Form | 26 |
|  | 4.7 DATA OUTPUT | 27 |
|  | 4.7.1 Data Storage Interface | 27 |
|  | 4.7.2 Data Reports | 27 |
|  | 4.8 IMPLEMENTATION & TESTING | 28 |
|  | 4.8.1 Implementation | 28 |
|  | 4.8.2 Testing | 29 |
|  | 4.8.2.1 Unit Testing | 29 |
|  | 4.8.2.2 Integration Testing | 31 |
|  | 4.8.2.3 System Testing | 31 |
|  | 4.9 USE CASE DIAGRAM | 34 |
| **5** | **CHALLENGES AND SOLUTIONS** | 35 – 39 |
| **6** | **FUTURE WORK** | 40 - 45 |
|  | 6.1 POTENTIAL ENHANCEMENT AND ADDITIONAL FEATURES | 40 |
|  | 6.2 PLANS FOR SCALING THE SYSTEM | 41 |
|  | 6.3 IDEAS FOR FUTURE RESEARCH | 42 |
|  | 6.4 PLANS FOR SCALING THE SYSTEM | 43 |
|  | 6.5 IDEAS FOR FUTURE RESEARCH | 44 |
| **7** | **EVALUATION & CONCLUSION** | 46 - 48 |
|  | 7.1 EVALUATION | 46 |
|  | 7.2 LIMITATIONS OF SYSTEM | 46 |
|  | 7.3 PROBLEMS ENCOUNTERED | 47 |
|  | 7.4 RECOMMENDATION/FUTURE RESEARCH | 47 |
|  | **REFERENCES & BIBLIOGRAPHY** | 49 |

**LIST OF FIGURES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Figure No.** | | **Name of Figure** | **Page No.** | |
| 4.1 | | Logical Database Design | | 15 | |
| 4.2 | | ER Diagram | | 20 | |
| 4.3 | | DFD Level-0 | | 21 | |
| 4.4 | | DFD Level-1 | | 21 | |
| 4.5 | | DFD Level-2 | | 21 | |
| 4.6 | | System Flow Chart | | 22 | |
| 4.7 | | Login Form | | 23 | |
| 4.8 | | Patient Registration Form | | 24 | |
| 4.9 | | Uploading Report form | | 24 | |
| 4.10 | | Add Health Record Form | | 25 | |
| 4.11 | | Add User Form | | 26 | |
| 4.12 | | Data Storage Interface for Patient History | | 26 | |
| 4.13 | | Interface for viewing Uploaded Reports | | 27 | |
| 4.14 | | Interface to Generate Report | | 27 | |
| 4.15 | | Use Case Diagram | | 34 | |
|  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Table** | **Page No.** |
| 4.1 | Medicine | 16 |
| 4.2 | Position | 16 |
| 4.3 | Medicine detail | 17 |
| 4.4 | Patient | 17 |
| 4.5 | Patient visit | 18 |
| 4.6 | Reports | 18 |
| 4.7 | Users | 19 |

**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**

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.

Wellness Program Management System (WPMS) 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.

**1.2 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 Wellness Program Management Systems (WPMS).

**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 wellness program 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 Wellness Program Management Systems (WPMS) 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 Wellness Program 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 Wellness Program Management System (WPMS) 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.

**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 Wellness Program Management System (WPMS) 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 WPMS 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 WPMS 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 WPMS 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 WPMS.
* Session Handling: Secure session management within the local environment ensures authenticated access during the admin's interaction with the WPMS.

**4.1.1.3 Local Environment Accessibility**

* Device Compatibility: The WPMS 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 WPMS is limited to the local network, restricting access outside the local environment during the development phase.

**4.1.2 User Privileges**

The WPMS 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**

* **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 PHP Storm for code customization or review.

**Other Considerations**

* **User Account:**

Admin credentials to log in to the WPMS 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.

**4.3.2 Frontend Interface Modules**

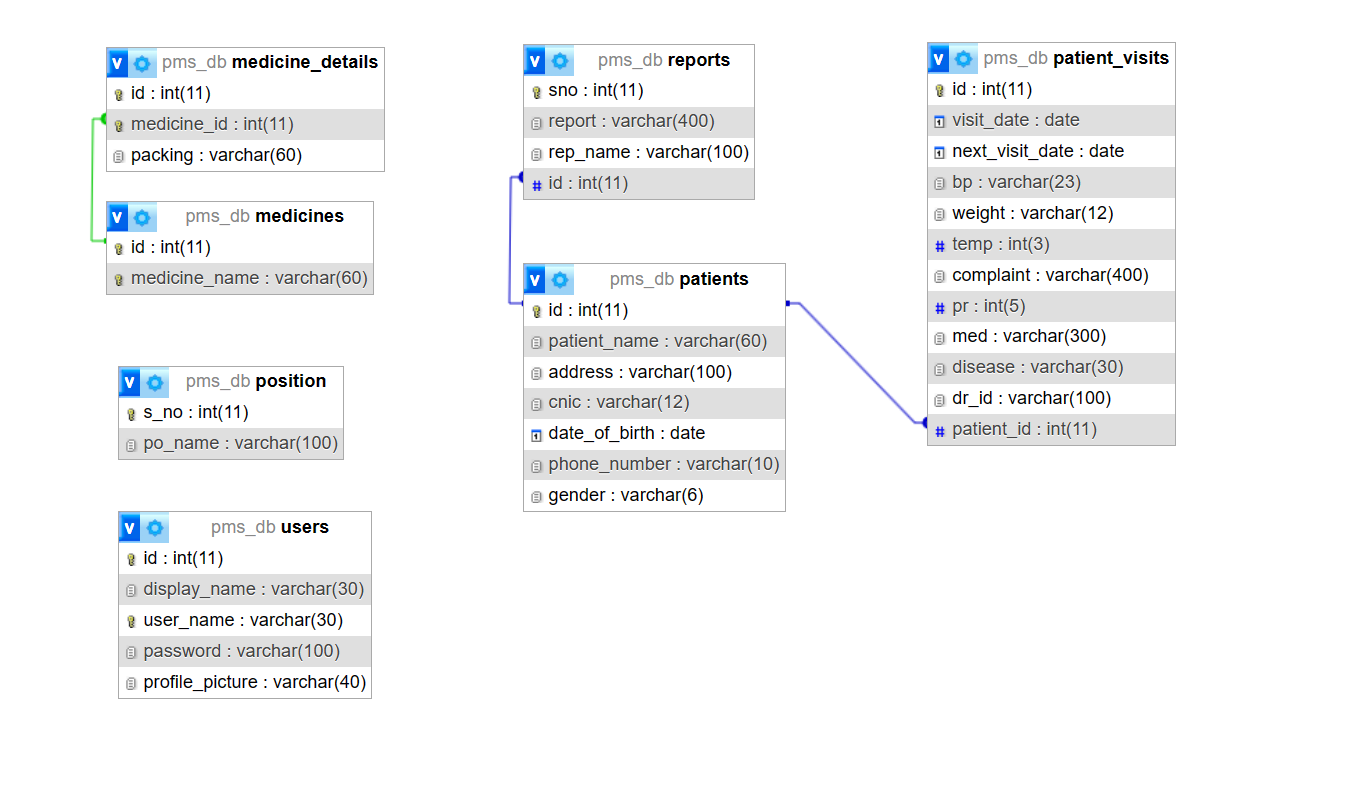
* **HTML (Hyper Text Markup Language)**

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.

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

**Table 4.1 Medicines**

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

#### **Table 4.2** **Position**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 4.3** **Medicine Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 4.4** **Patients**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 4.5** **Patient visits**

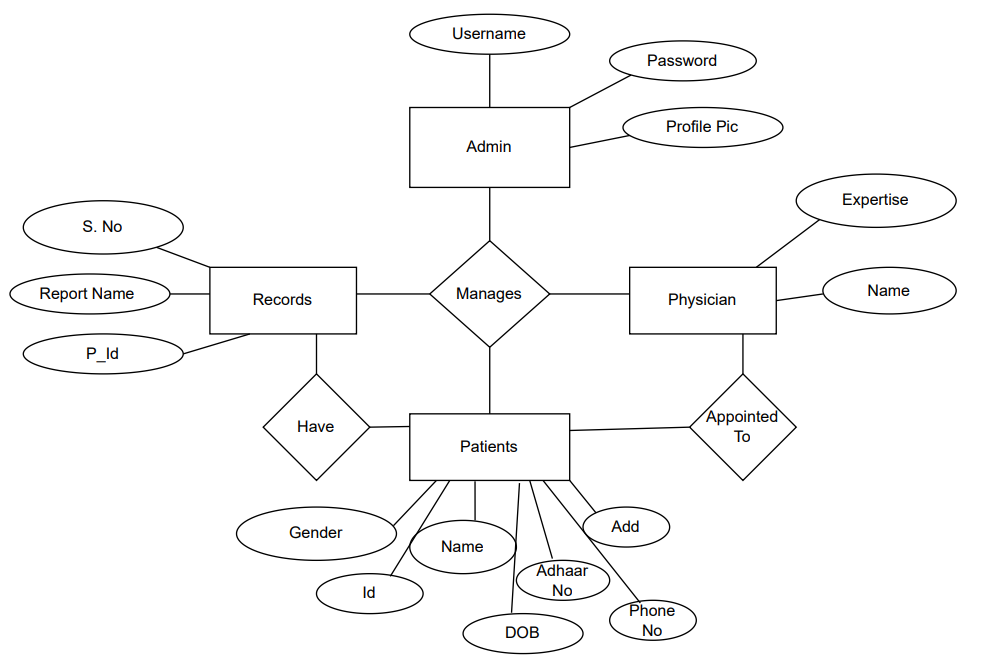
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Field** **type** | **Length/size** | **Description** |
| Sno | Int | 11 | Auto increment, primary key |
| Report | Varchar | 400 | Report location |
| Rep\_name | Varchar | 100 | Type of report |
| id | int | 11 | Foreign key, Patient id |

**Table 4.6** **Reports**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

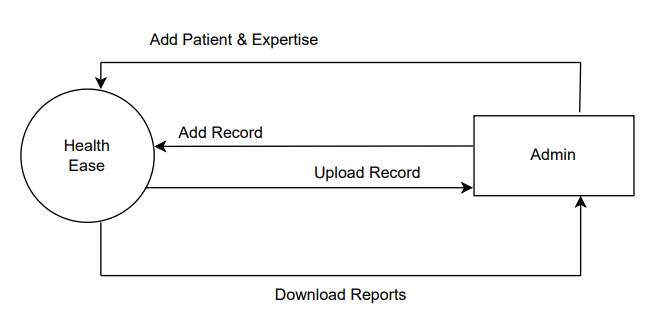
**Table 4.7** **Users**

## **4.4 ER DIAGRAMS & DFDs**

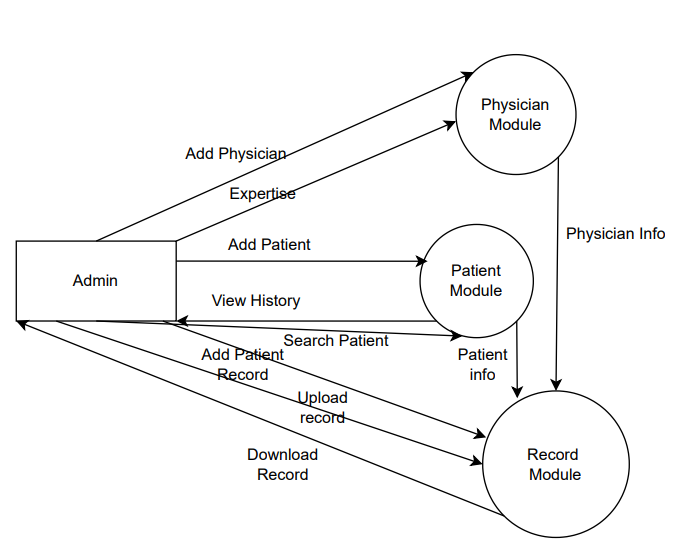
**4.4.1 ERD (Entity Relationship Diagram)**

**Fig 4.2 ER Diagram**

**4.4.2 DFD (Data Flow Diagram)**

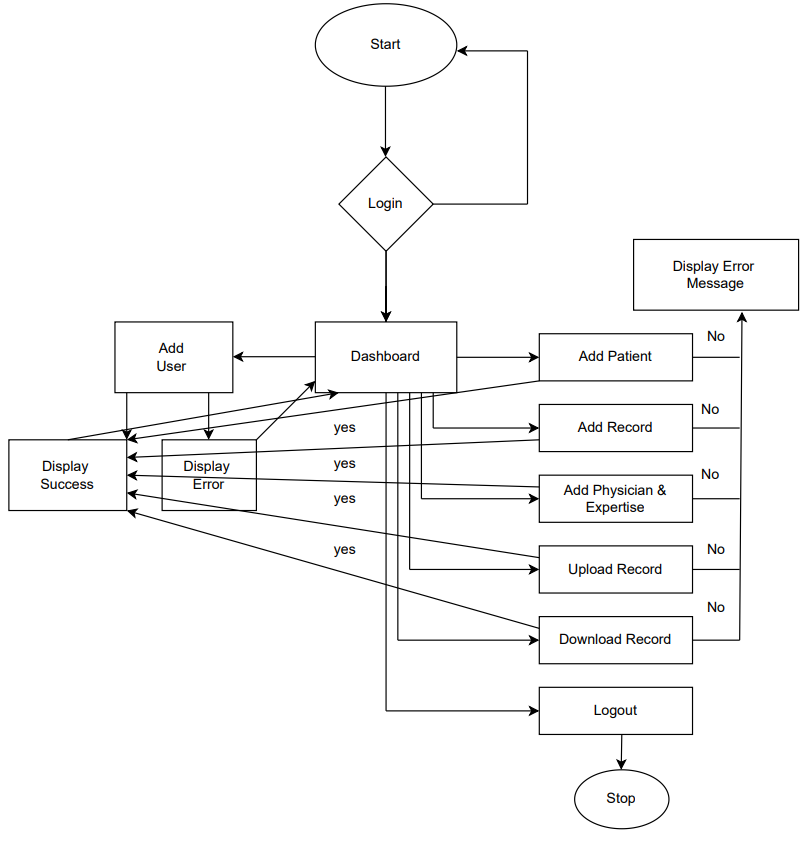
****

**Fig 4.3 DFD Level-0**

****

**Fig 4.4 DFD Level-1**

* 1. **System Flowchart**



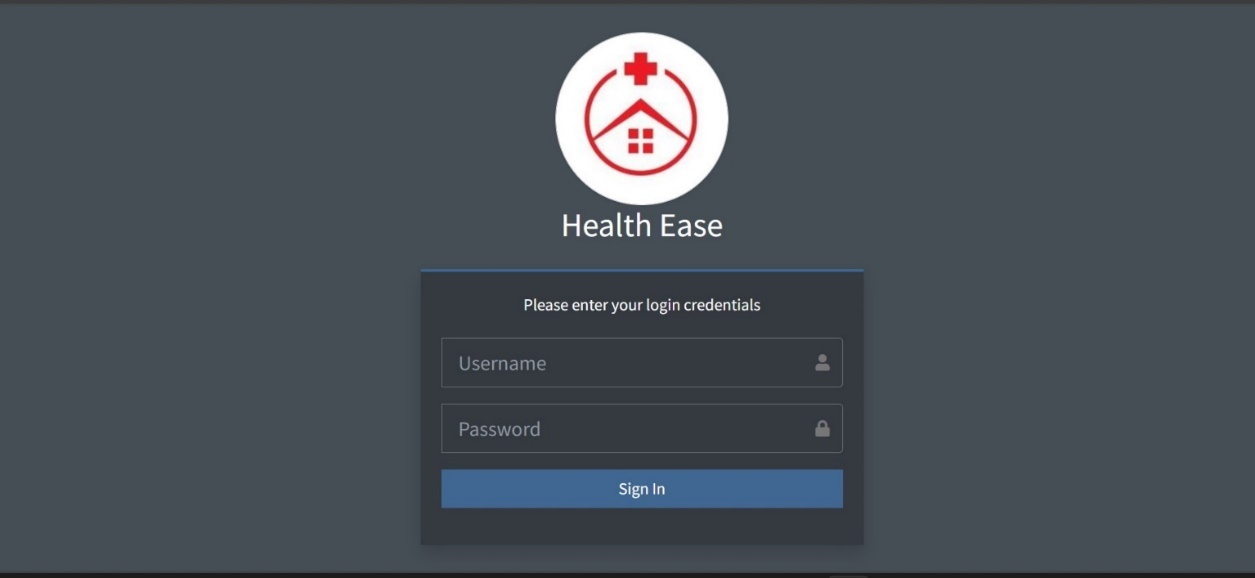
**Fig 4.6 System Flow Chart**

* 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**

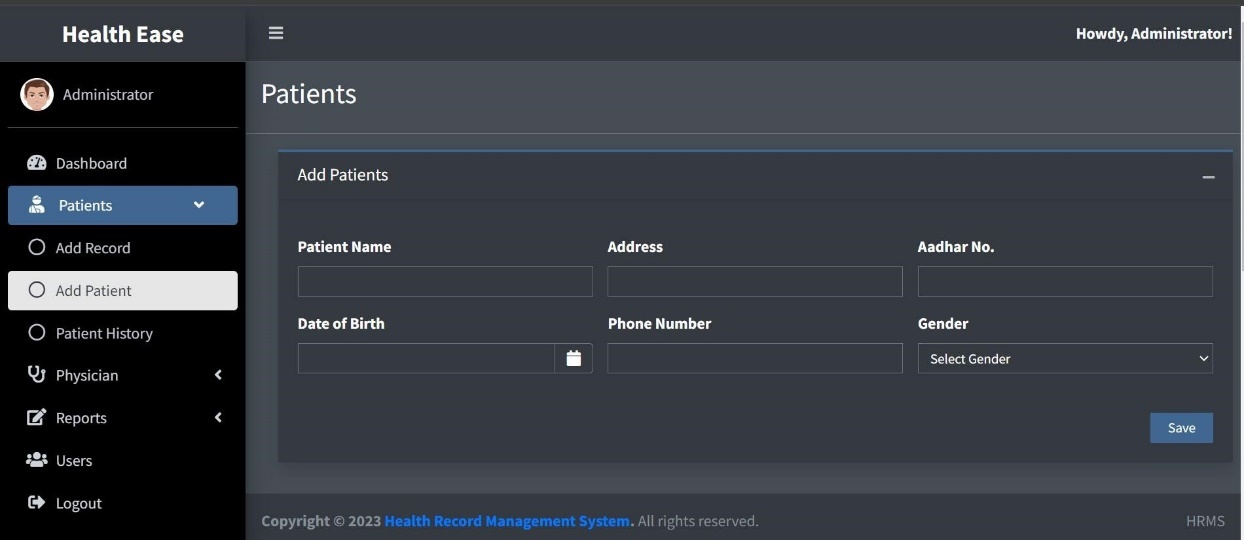
The login form above is the first page a person accessing the system sees. It is used to gain access to the system resources and determines, based on the user type, which users should access which resources.



**Fig 4.7 Login Form**

**4.6.2 Patient Registration Form**

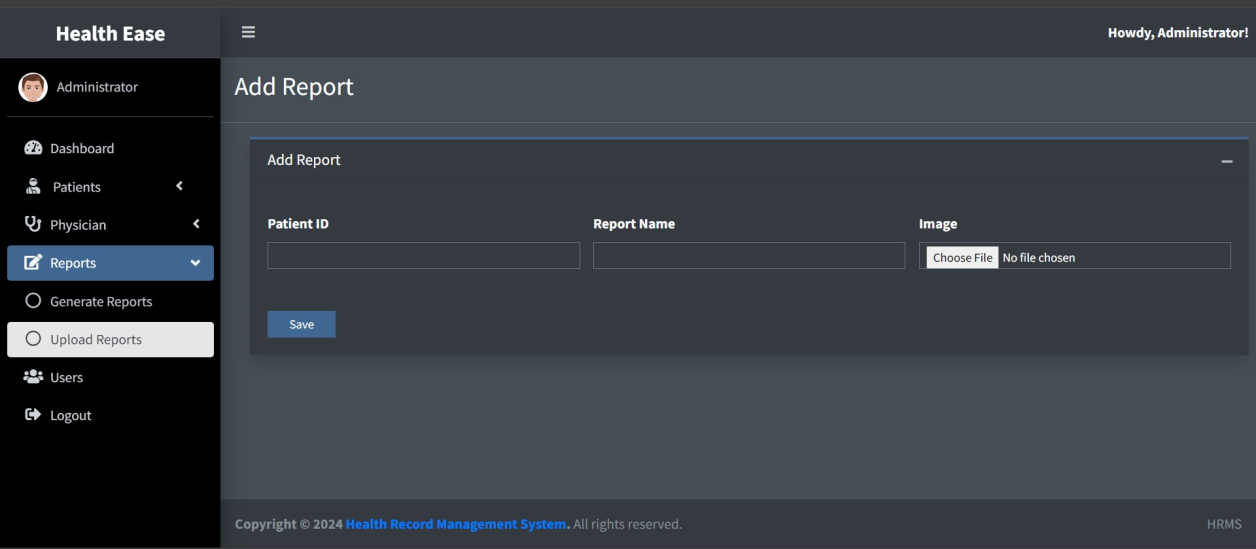
The patient addition interface, managed by the admin, serves as the portal for integrating new patients into the system. Admins utilize this interface to input comprehensive patient details, facilitating seamless incorporation of individuals into the healthcare management system. This feature empowers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

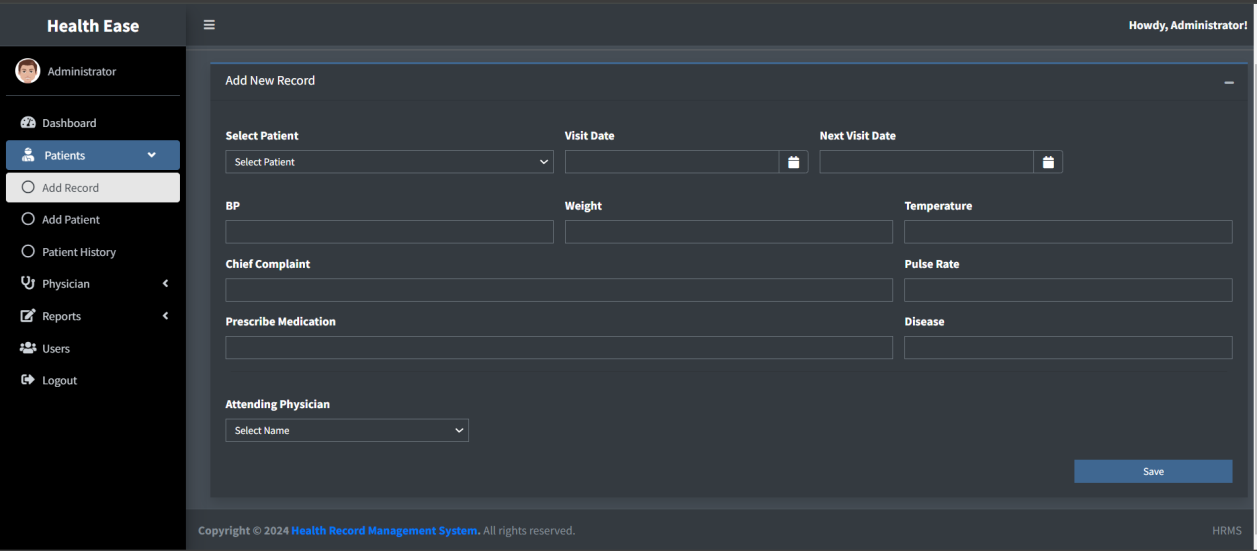
The report uploading feature, managed by the admin, serves as the gateway for integrating essential medical reports into the system. This functionality empowers administrators to securely upload and integrate diverse reports, enriching the system’s repository of patient information.



**Fig 4.9 Uploading Report form**

**4.6.4 Data Entry and Manipulation Forms**

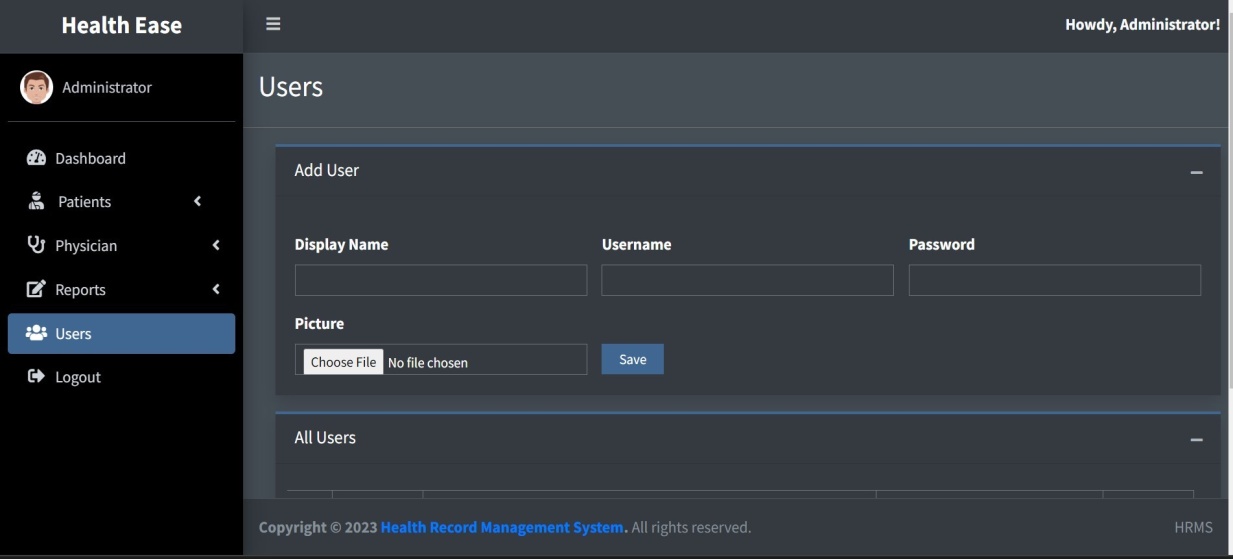
The patient data addition form offers administrators a structured interface to input comprehensive patient information into the system. This feature empowers admins to seamlessly add and organize patient data, ensuring a detailed and accurate database within the healthcare management system.



**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.

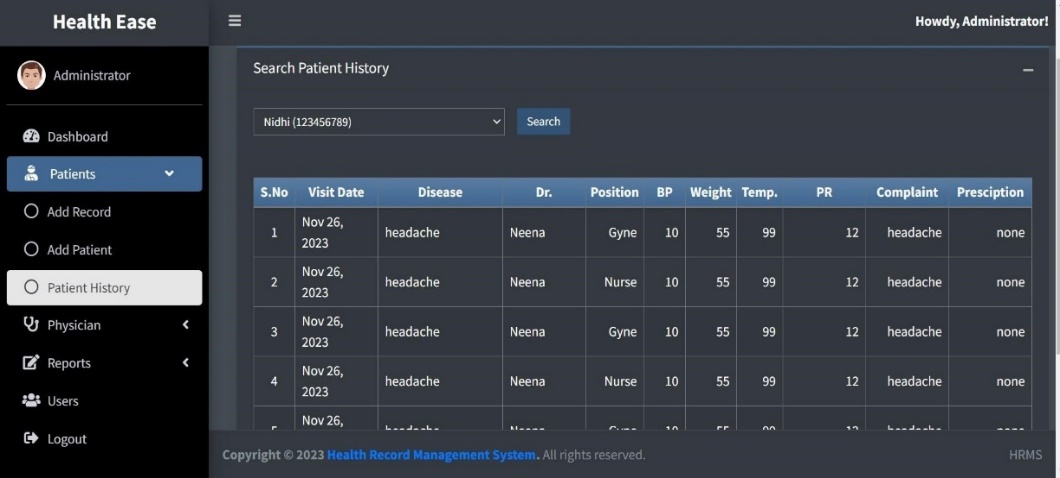


**Fig 4.11 Add User Form**

## **4.7 DATA OUTPUTS**

Outputs are selected from the database based on a certain criteria and displayed using form, below are some snap shots of the key forms.

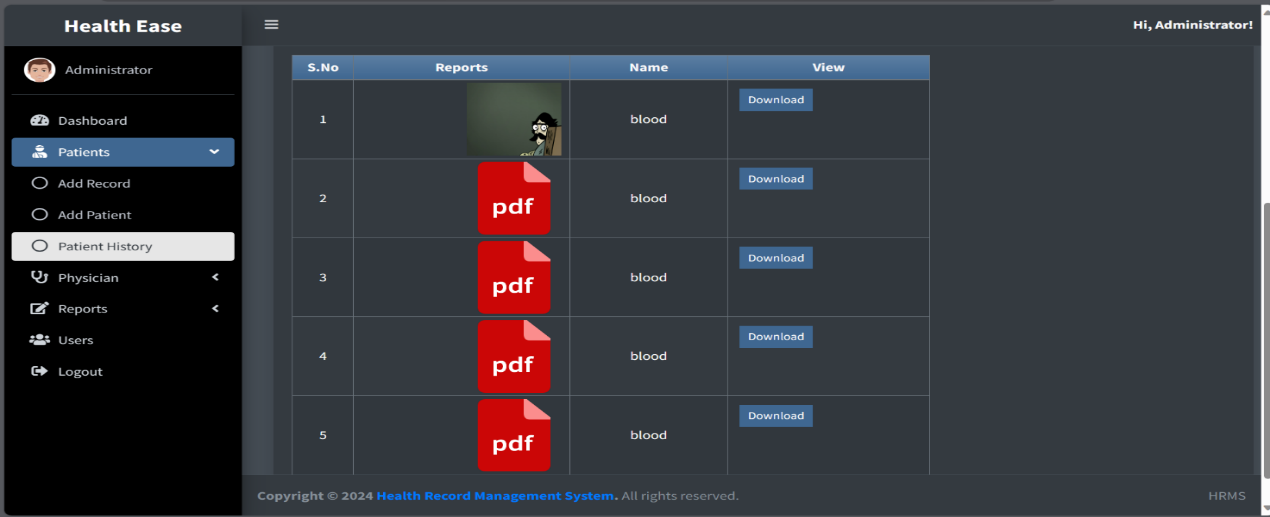
**4.7.1 Data Storage Interface**

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

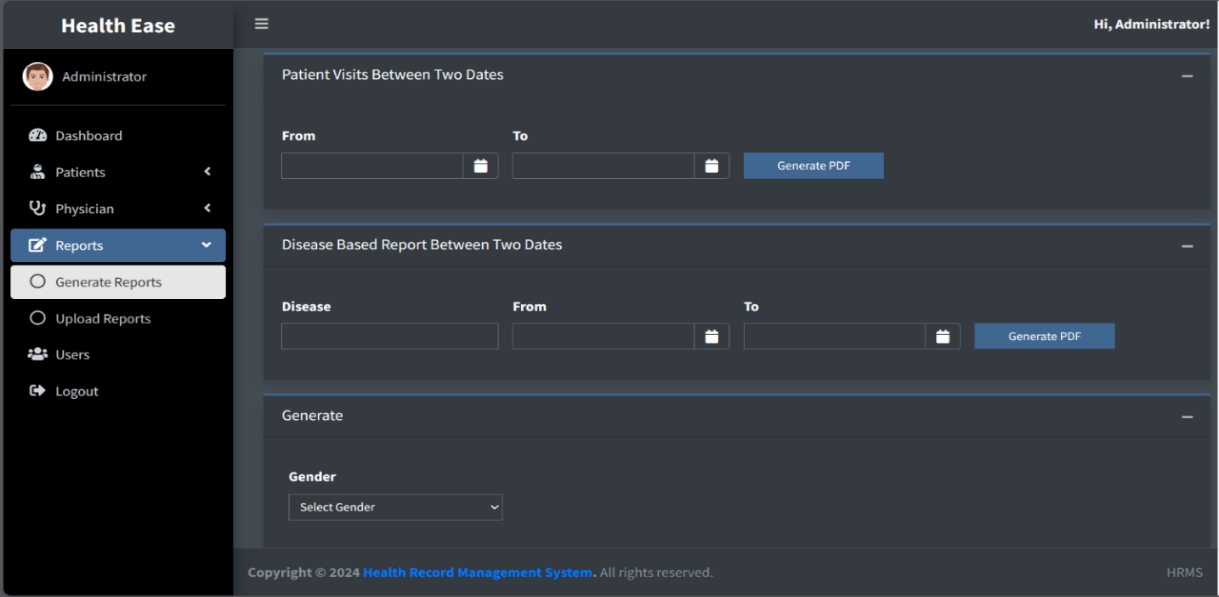
**Fig 4.12:** **Data** **Storage** **interface** **for** **Patient** **History**

**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**

* **Overview of Technologies Used**

The WPMS (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 WPMS 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 WPMS, 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 WPMS (Wellness Program 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 WPMS (Wellness Program 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 WPMS 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**

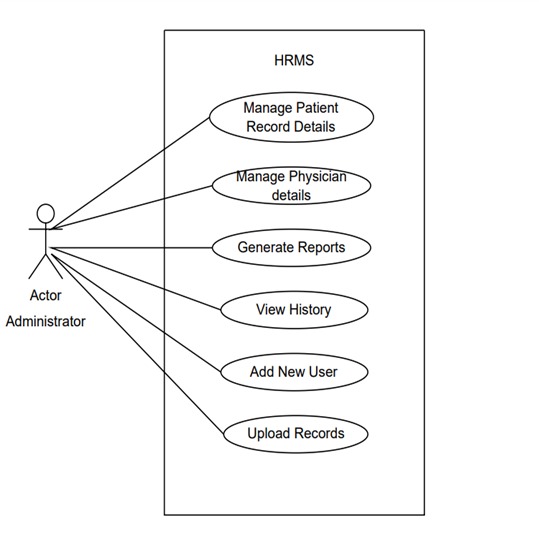
System testing was conducted in an environment that closely resembled the production environment, ensuring realistic testing scenarios and conditions.

**Results and Validation**

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

Validation**:** The WPMS 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**

**CHALLENGES AND SOLUTIONS**

In the development of the Wellness Program Management System (WPMS), several challenges were encountered. These challenges were addressed through strategic problem-solving and innovative solutions. Below is a detailed account of the major challenges and the corresponding solutions implemented:

### 1. Data Security and Privacy

**Challenge:** Ensuring the security and privacy of sensitive patient data was a paramount concern. The system needed to comply with healthcare regulations such as HIPAA, which mandates stringent data protection measures.

**Solution:** To address this, robust encryption techniques were employed for data storage and transmission. Secure Sockets Layer (SSL) was implemented to encrypt data exchanges between the client and server. Additionally, user authentication and authorization mechanisms were put in place to ensure that only authorized personnel could access sensitive information. Regular security audits and vulnerability assessments were conducted to identify and mitigate potential security risks.

### 2. Integration of Various Technologies

**Challenge:** The WPMS was developed using a combination of technologies including XAMPP, PHP, MySQL, JavaScript, HTML, and CSS. Integrating these technologies seamlessly posed a significant challenge.

**Solution:** A modular approach was adopted, where each technology handled specific functionalities. PHP and MySQL were used for server-side scripting and database management, respectively, while JavaScript, HTML, and CSS handled the client-side operations. Clear interfaces and APIs were defined to facilitate communication between the modules. Regular integration testing was conducted to ensure smooth interoperability between different components.

### 3. Performance Optimization

**Challenge:** Ensuring the system could handle a large number of concurrent users and substantial volumes of data without performance degradation was critical.

**Solution:** Performance optimization techniques such as database indexing, query optimization, and caching were employed to enhance system responsiveness. Load testing was performed to identify and rectify bottlenecks. Server-side code was optimized to reduce processing time, and efficient algorithms were implemented to handle data-intensive operations.

### 4. User-Friendly Interface Design

**Challenge:** Designing an intuitive and user-friendly interface that caters to healthcare professionals with varying levels of technical expertise was challenging.

**Solution:** Extensive user research and usability testing were conducted to understand the needs and preferences of potential users. The interface was designed with simplicity and ease of use in mind, incorporating clear navigation, accessible menus, and informative feedback mechanisms. Iterative design and testing cycles ensured that the final interface was both functional and user-friendly.

### 5. PDF Report Generation

**Challenge:** Generating detailed and accurately formatted PDF reports from dynamic data sources was technically demanding.

**Solution:** The TCPDF library was integrated into the system to facilitate the generation of PDF reports. Custom templates were designed to ensure that the reports were visually appealing and contained all necessary information. Dynamic data binding techniques were used to populate the templates with real-time data. Extensive testing was carried out to ensure the reports were generated correctly under various conditions.

### 6. Scalability

**Challenge:** Designing the system to be scalable to accommodate future growth in terms of users and data volume.

**Solution:** The system architecture was designed with scalability in mind. A modular and decoupled architecture was adopted, allowing individual components to be scaled independently. The database schema was normalized to reduce redundancy and improve efficiency. Load balancing techniques and distributed computing practices were planned for future implementation to handle increased loads.

By addressing these challenges with targeted solutions, the development team was able to create a robust, secure, and efficient Wellness Program Management System that meets the needs of healthcare providers and enhances patient data management.

### 1. Data Security and Privacy

**Challenge:** Ensuring the security and privacy of sensitive patient data was a paramount concern. The system needed to comply with healthcare regulations such as HIPAA, which mandates stringent data protection measures.

**Solution:** To address this, several layers of security were implemented:

* **Encryption:** Robust encryption techniques, including AES-256, were used for storing sensitive data in the database. For data transmission, SSL/TLS protocols were employed to secure the communication between the client and server.
* **Authentication and Authorization:** Multi-factor authentication (MFA) was introduced to enhance the security of user logins. Role-based access control (RBAC) ensured that users had access only to the data and functionalities necessary for their roles.
* **Data Masking:** Sensitive data fields such as social security numbers and medical history were masked in the user interface and logs to prevent unauthorized access.
* **Regular Audits:** Regular security audits and vulnerability assessments were conducted to identify and mitigate potential security risks. Penetration testing was performed periodically to evaluate the system's defense against cyber attacks.

### 2. Integration of Various Technologies

**Challenge:** The WPMS was developed using a combination of technologies including XAMPP, PHP, MySQL, JavaScript, HTML, and CSS. Integrating these technologies seamlessly posed a significant challenge.

**Solution:** A modular approach was adopted:

* **Clear Module Separation:** Each technology handled specific functionalities. PHP and MySQL managed server-side operations and database interactions, while JavaScript, HTML, and CSS managed client-side operations.
* **APIs and Services:** RESTful APIs were developed to facilitate communication between different modules, ensuring a clear separation of concerns and promoting maintainability.
* **Continuous Integration:** Tools like Jenkins and Git were used for continuous integration and version control, ensuring that changes in one module did not adversely affect others. Regular integration testing ensured compatibility and smooth interoperability between components.

### 3. Performance Optimization

**Challenge:** Ensuring the system could handle a large number of concurrent users and substantial volumes of data without performance degradation was critical.

**Solution:** Several performance optimization strategies were implemented:

* **Database Optimization:** Indexing frequently queried fields, optimizing SQL queries, and using stored procedures to reduce database load.
* **Caching:** Implementing server-side caching using technologies like Memcached or Redis to store frequently accessed data temporarily.
* **Asynchronous Processing:** Using asynchronous processing for non-critical tasks to improve response times for user interactions.
* **Load Balancing:** Planning for future implementation of load balancers to distribute incoming traffic across multiple servers, ensuring no single server becomes a bottleneck.
* **Resource Monitoring:** Continuous monitoring of system performance using tools like Nagios or New Relic to identify and address performance issues proactively.

### 4. User-Friendly Interface Design

**Challenge:** Designing an intuitive and user-friendly interface that caters to healthcare professionals with varying levels of technical expertise was challenging.

**Solution:** The following steps were taken to ensure a user-friendly interface:

* **User Research:** Conducting surveys, interviews, and usability studies with potential users to understand their needs, preferences, and pain points.
* **Wireframes and Prototypes:** Creating wireframes and interactive prototypes to visualize the user interface and gather feedback early in the design process.
* **Iterative Design:** Employing an iterative design process where feedback from usability testing was incorporated into successive versions of the interface.
* **Accessible Design:** Following accessibility guidelines (such as WCAG) to ensure the interface was usable by individuals with disabilities. This included features like keyboard navigation, screen reader support, and high-contrast modes.
* **Training and Documentation:** Providing comprehensive user manuals, tooltips, and video tutorials to help users understand and navigate the system effectively.

### 5. PDF Report Generation

**Challenge:** Generating detailed and accurately formatted PDF reports from dynamic data sources was technically demanding.

**Solution:** To address this challenge:

* **TCPDF Integration:** The TCPDF library was integrated into the system to facilitate PDF generation.
* **Custom Templates:** Custom report templates were designed to ensure consistency and clarity. These templates included headers, footers, and dynamic content sections.
* **Dynamic Data Binding:** Implementing dynamic data binding techniques to populate the templates with real-time data pulled from the database.
* **Formatting and Styling:** Ensuring proper formatting and styling of reports to make them professional and easy to read. This included tables, charts, and graphs where necessary.
* **Testing and Validation:** Extensive testing was carried out to ensure reports were generated correctly under various conditions. This included validating the accuracy of the data presented and the consistency of the formatting across different scenarios.

### 6. Scalability

**Challenge:** Designing the system to be scalable to accommodate future growth in terms of users and data volume.

**Solution:** Scalability was addressed through:

* **Modular Architecture:** Designing the system with a modular architecture allowed individual components to be scaled independently based on demand.
* **Database Scaling:** Using database sharding and replication techniques to handle large volumes of data and improve read/write performance.
* **Microservices:** Planning for a transition to a microservices architecture, where different services can be deployed and scaled independently.
* **Cloud Deployment:** Considering cloud infrastructure (such as AWS or Azure) for future deployment, providing on-demand scaling capabilities and high availability.
* **Load Testing:** Conducting extensive load testing to understand the system's limits and identify potential scaling bottlenecks. Tools like JMeter were used to simulate high user loads and measure system performance.

**CHAPTER 6**

**FUTURE WORK**

**6.1 Potential Enhancements and Additional Features**

1. **Advanced Analytics and Reporting**

* **Predictive Analytics:** Implementing machine learning algorithms to predict patient health trends and potential risks. This could help healthcare providers take proactive measures.
* **Customizable Reports:** Allowing users to create and customize their reports based on specific criteria and metrics. This feature could include drag-and-drop report builders and real-time data visualization tools.

1. **Mobile Application Development**

* **Mobile App:** Developing a mobile application for Android and iOS to provide healthcare providers with access to the WPMS on-the-go. This app could include functionalities like patient data access, appointment scheduling, and notifications.

1. **Telemedicine Integration**

* **Video Consultations:** Integrating video conferencing tools to enable telemedicine services. This would allow healthcare providers to conduct remote consultations and follow-ups.
* **Remote Monitoring:** Incorporating wearable device data to monitor patient vitals in real-time and provide alerts for any abnormalities.

1. **Enhanced User Interface**

* **Dark Mode:** Adding a dark mode option to reduce eye strain for users who work for extended periods.
* **Localization:** Supporting multiple languages to make the system accessible to non-English speaking users.

1. **Appointment and Prescription Management**

* **Scheduling System:** Implementing an advanced appointment scheduling system that manages patient appointments, sends reminders, and handles cancellations.
* **E-Prescriptions:** Allowing doctors to issue electronic prescriptions that can be sent directly to pharmacies.

1. **Interoperability with Other Systems**

* **EHR Integration:** Integrating with existing Electronic Health Record (EHR) systems to ensure seamless data exchange and interoperability.
* **Health Information Exchange (HIE):** Participating in HIE networks to facilitate the secure sharing of patient information across different healthcare organizations.

**6.2 Plans for Scaling the System**

1. **Cloud Infrastructure**

* **Migration to Cloud:** Moving the system to a cloud-based infrastructure (e.g., AWS, Azure) to leverage scalable resources and improve reliability and availability.
* **Auto-Scaling:** Implementing auto-scaling features to automatically adjust resources based on the system load, ensuring optimal performance during peak usage times.

1. **Database Scaling**

* **Horizontal Scaling:** Using database sharding to distribute data across multiple servers, improving performance and capacity.
* **Replication:** Setting up database replication to ensure high availability and disaster recovery.

1. **Microservices Architecture**

* **Service Decoupling:** Breaking down the monolithic architecture into microservices, where each service handles a specific functionality. This will allow independent scaling and easier maintenance.
* **Containerization:** Using containerization technologies like Docker and orchestration tools like Kubernetes to manage and scale microservices efficiently.

1. **Performance Optimization**

* **Load Balancing:** Implementing load balancers to distribute incoming traffic evenly across multiple servers, preventing any single server from becoming a bottleneck.
* **Caching:** Utilizing distributed caching solutions like Redis or Memcached to reduce database load and improve response times.

**6.3 Ideas for Further Research**

1. **Artificial Intelligence and Machine Learning**

* **Disease Prediction Models:** Researching and developing AI models that can predict the onset of diseases based on patient data and historical trends.
* **Natural Language Processing (NLP):** Using NLP to analyze unstructured data such as doctors’ notes and patient feedback to extract valuable insights.

1. **Blockchain for Data Security**

* **Immutable Records:** Investigating the use of blockchain technology to create immutable patient records that enhance security and trust.
* **Smart Contracts:** Exploring the application of smart contracts for automating and securing transactions between patients, healthcare providers, and insurers.

1. **Internet of Things (IoT) Integration**

* **Wearable Devices:** Studying the integration of IoT devices like fitness trackers and smartwatches to continuously monitor patient vitals and provide real-time data to the WPMS.
* **Remote Patient Monitoring:** Developing IoT solutions for remote patient monitoring, allowing healthcare providers to track the health status of patients outside of clinical settings.

1. **Data Privacy and Compliance**

* **Regulation Adherence:** Conducting research on global data privacy laws and regulations to ensure the system remains compliant as it scales to different regions.
* **Advanced Encryption:** Investigating advanced encryption techniques and privacy-preserving technologies such as homomorphic encryption to enhance data security.

1. **User Experience (UX) Improvements**

* **Human-Computer Interaction:** Researching how different user interaction models affect the efficiency and satisfaction of healthcare providers using the system.
* **Personalization:** Exploring methods to personalize the user experience based on individual user preferences and behaviors.

1. **Artificial Intelligence for Decision Support**

* **Clinical Decision Support Systems (CDSS):** Integrating AI-powered CDSS to provide healthcare providers with evidence-based clinical guidelines and recommendations. This can help in making more accurate diagnoses and treatment plans.
* **Automated Triage:** Implementing AI algorithms to assess patient symptoms and prioritize cases based on urgency. This feature can streamline emergency room operations and improve patient outcomes.

1. **Patient Engagement and Education**

* **Health Education Portal:** Creating an interactive portal where patients can access educational materials, such as articles, videos, and tutorials about various health conditions and wellness tips. This can empower patients to take an active role in their health management.
* **Gamification:** Introducing gamification elements such as health challenges, progress tracking, and rewards to motivate patients to adhere to treatment plans and healthy lifestyle choices.

1. **Integration with National Health Systems**

* **Interoperability with National Health Databases:** Developing features to integrate with national health databases and public health systems. This can facilitate the sharing of anonymized health data for public health monitoring and research.
* **Electronic Health Card Integration:** Supporting integration with electronic health cards used in various countries, allowing seamless access to patient health records across different healthcare providers.

1. **Enhanced Communication Tools**

* **Secure Messaging:** Implementing secure messaging features for direct communication between patients and healthcare providers. This can include chat functionality, secure email, and notifications.
* **Patient Feedback System:** Allowing patients to provide feedback on their care experience. This feedback can be used to improve service quality and patient satisfaction.
  1. **Plans for Scaling the System**

1. **Global Expansion**

* **Multilingual Support:** Expanding multilingual support to include more languages, making the system accessible to a broader international audience. This includes not only translation but also cultural adaptation of the user interface and content.
* **Regional Customization:** Customizing features to meet the specific needs and regulations of different regions, ensuring compliance with local healthcare standards and practices.

1. **Distributed Database Management**

* **Geographically Distributed Databases:** Implementing geographically distributed databases to improve data access speed and reliability for users in different locations. This can also enhance data redundancy and disaster recovery capabilities.
* **Data Partitioning:** Using data partitioning strategies to manage large datasets efficiently, ensuring quick access and processing of data based on specific criteria such as geographical location or department.
  1. **Ideas for Further Research**

1. **Behavioural Health Integration**

* **Mental Health Monitoring:** Researching and developing tools for monitoring and managing mental health conditions. This can include integration with mental health apps and the use of AI to detect early signs of mental health issues from patient interactions and data.
* **Teletherapy:** Exploring the integration of teletherapy services within the WPMS, providing remote counseling and therapy sessions for patients with mental health needs.

1. **Sustainability and Environmental Impact**

* **Green Computing Practices:** Investigating ways to reduce the environmental impact of the WPMS by adopting green computing practices. This can include optimizing energy consumption of servers, using sustainable data centers, and reducing electronic waste.
* **Carbon Footprint Monitoring:** Implementing features to monitor and report the carbon footprint of healthcare operations, helping organizations to become more environmentally responsible.

1. **Social Determinants of Health (SDOH)**

* **SDOH Data Integration:** Integrating data on social determinants of health, such as socioeconomic status, education, and living conditions, to provide a more holistic view of patient health. This can help in designing more effective and personalized care plans.
* **Community Health Initiatives:** Researching the impact of community health initiatives and programs on patient outcomes, and incorporating these insights into the WPMS to support community-based healthcare strategies.

1. **Robotic Process Automation (RPA)**

* **Automating Administrative Tasks:** Exploring the use of RPA to automate routine administrative tasks such as appointment scheduling, billing, and claims processing. This can reduce administrative burden and free up healthcare providers to focus more on patient care.
* **Document Processing:** Implementing RPA for automated processing and management of healthcare documents, including scanning, indexing, and data extraction from paper records.

1. **Patient-Centered Care Models**

* **Personalized Care Pathways:** Researching and developing personalized care pathways that adapt to individual patient needs and preferences. This can include dynamic care plans that adjust based on patient progress and feedback.
* **Patient Empowerment Tools:** Creating tools that empower patients to take control of their health, such as personalized health dashboards, goal setting, and progress tracking features.

**CHAPTER 7**

**EVALUATIONS & CONCLUSIONS**

**7.1 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.

## **7.2 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.

## **7.3 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.

## **7.4 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.

## **7.5 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**

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

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

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

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

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

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

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

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

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