**Abstract**

The Doctor Appointment System is a digital platform designed to streamline and improve the process of scheduling and managing appointments between patients and medical practitioners. This system offers a user-friendly interface for patients to view available time slots, select preferred doctors, and conveniently book appointments. Medical practitioners can effectively manage their schedules, track patient appointments, and optimize their time allocation. Utilizing PHP and a database-driven approach, the system aims to reduce wait times, enhance communication between patients and doctors, and elevate the overall quality of healthcare service delivery.

**Keywords:** PHP, web application, database

**Acknowledgement**

We extend our heartfelt gratitude to our supervisor, Mr. [Supervisor's Name], for providing us with the invaluable opportunity to work on this project on the topic of ONLINE DOCTOR APPOINTMENT SYSTEM. His guidance and encouragement enabled us to conduct extensive research and explore numerous new tools and technologies.

Special thanks are also due to Mr. [BCA Program Coordinator's Name], our BCA Program Coordinator, for his unwavering support and assistance in our personal development and the successful completion of this project.

We are deeply indebted to Morgan International College for their continuous guidance, supervision, and provision of essential project information. We also appreciate the cooperation and encouragement from the library staff and members of Morgan International College, which greatly facilitated the completion of this project.

Our heartfelt appreciation goes to our parents and friends whose support was invaluable in meeting the project deadline.

Lastly, we acknowledge Tribhuvan University for offering us the opportunity through the Computer Application course to understand project ethics at an early stage, which has significantly enriched and expanded our knowledge.

**TABLE OF CONTENTS**

**ABSTRACT ................................................................................................................... i**

**ACKNOWLEDGEMENT ........................................................................................... ii**

**LIST OF ABBREVIATIONS....................................................................................... v**

**LIST OF FIGURES..................................................................................................... vi**

**LIST OF TABLES....................................................................................................... vii**

**CHAPTER 1: INTRODUCTION**

1.1 Introduction ............................................................................................................ 1

1.2 Problem Statement ............................................................................................... 1

1.3 Objectives .............................................................................................................. 1

1.4 Scope and Limitation ......................................................................................... 2

1.4.1 Scope ............................................................................................................. 2

1.4.2 Limitations ................................................................................................... 2

1.5 Report Organization ........................................................................................... 3

**CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

2.1 Background Study ............................................................................................... 4

2.2 Literature Review ................................................................................................. 4

**CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

3.1 System Analysis .................................................................................................. 6

3.1.1 Requirement Analysis ................................................................................... 7

3.1.2 Feasibility Analysis ....................................................................................... 9

3.1.3 Data Modeling (ER-Diagram) .................................................................... 11

3.1.4 Process Modeling (DFD) ............................................................................ 12

3.2 System Design .................................................................................................... 13

**CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

3.1 System Analysis .................................................................................................. 6

3.1.1 Requirement Analysis ................................................................................... 7

3.1.2 Feasibility Analysis ....................................................................................... 9

3.1.3 Data Modeling (ER-Diagram) .................................................................... 11

3.1.4 Process Modeling (DFD) ............................................................................ 12

3.2 System Design .................................................................................................... 13

3.2.1 Architectural Design ................................................................................... 13

3.2.2 System Flowchart ....................................................................................... 13

3.2.3 Database Schema Design .......................................................................... 15

3.2.4 Interface Design ........................................................................................ 16

3.2.5 Physical DFD ............................................................................................ 18

**CHAPTER 4: IMPLEMENTATION AND TESTING**

4.1 Implementation ................................................................................................. 20

4.1.1 Tools Used (CASE tools, Programming language, Database platforms) ... 21

4.1.2 Implementation Details of Modules (Description of procedures/functions) ... 21

4.2 Testing .............................................................................................................. 21

4.2.1 Test Cases for Unit Testing ........................................................................ 21

4.2.2 Test Cases for System Testing .................................................................... 24

**CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION**

5.1 Lesson Learnt / Outcome .................................................................................. 26

5.2 Conclusion ........................................................................................................ 26

5.3 Future Recommendations ................................................................................... 27

**REFERENCES .......................................................................................................... 28**

**APPENDIX: SYSTEM SCREENSHOTS**

**LIST OF ABBREVIATIONS**

CRUD - Create, Read, Update and Delete  
CSS - Cascading Style Sheet  
DFD - Data Flow Diagram  
ERD - Entity Relationship Diagram  
HTML - Hyper Text Markup Language  
MySQL - Microsoft Server Structured Query Language  
PHP - Hypertext Preprocessor  
UI - User Interface

**LIST OF FIGURES**

Figure 3.1: Waterfall Methodology for Doctor Appointment system  
Figure 3.2: Use Case Diagram of Doctor Appointment system  
Figure 3.3: Gantt Chart of Doctor Appointment system  
Figure 3.4: Entity Relational Diagram for Doctor Appointment system  
Figure 3.6: Level-1 DFD for Doctor Appointment system  
Figure 3.7: Architectural Design of Doctor Appointment system  
Figure 3.8: Flow chart of Doctor Appointment system for patient  
Figure 3.9: Flowchart of Doctor Appointment system for Suppliers  
Figure 3.10: Database Schema Design of Doctor Appointment system  
Figure 3.11: UI Home page of Doctor Appointment system  
Figure 3.12: Registration page of Doctor Appointment system  
Figure 3.13: Login page of Doctor Appointment system  
Figure 3.14: Dashboard of Doctor Appointment system  
Figure 3.15: Physical DFD for patient

**LIST OF TABLES**

Table 3.1: Gantt chart Table for Doctor Appointment system  
Table 4.1: Test case for Farmer Registration of Doctor Appointment system  
Table 4.2: Test case for Supplier Registration of Doctor Appointment system  
Table 4.3: Test cases for Doctor Appointment system

**CHAPTER 3**  
**SYSTEM ANALYSIS AND DESIGN**

**3.1 System Analysis**

The architecture of the whole project is analyzed. System analysis is the process of defining the architecture, components, and data of a system to satisfy specified requirements. Design is a method of studying a system by examining its component parts and their interactions before implementation began. The system was analyzed and designed. In this section, use cases, requirement analysis, and other parts are described in detail.

[DIAGRAM HERE bash1] done

**Figure 3.1: Waterfall Methodology for Online Doctor Appointment System**

**3.1.1 Requirements Analysis**

The requirements are to be collected before starting project development lifecycle. To design and develop the system, functional as well as non-functional requirements of the system have been studied.

**i. Functional Requirements**

Different functional requirements of the system have been identified and are listed as below:

1. **User Registration and Authentication:**

- Users should be able to create accounts.

- Users should be able to log in securely.

1. **Appointment Scheduling:**

- Users should be able to view available time slots for appointments.

-Users should be able to select a preferred time slot.

- The system should prevent double booking of time slots.

1. **Doctor Profiles:**

- Display detailed profiles of doctors, including their specialties, qualifications, and availability.

1. **Cancellation and Rescheduling:**

- Users should be able to cancel or reschedule appointments within a reasonable time frame

[USE CASE DIAGRAM HERE bash2]

1.  
ii. Non Functional Requirement  
Different non-functional requirement have been studied and identified and are listed as below:

Security  
The system is secure from outside attacks as authorized user and admin are allowed to access the data. Admin representative on duty can log into the system and have access to the Doctor Appointment System but access to have various subsystems is protected by the user login screen that requires a username and password. This system uses at least 8-character passwords for security. Different validation process is used.

Performance  
The performance of the system is fast and accurate as in this system database is normalized so it provide fast operations.

3.1.2 Feasibility Analysis  
i. Technical feasibility study  
The system is technically feasible as the requirement for the development of the system is easily accessible. The necessary hardware and software required for the development and implementation of the system is available. The basic programming language which is suitable for project is available and the libraries required for project is capable of achieving the result that we are aiming for. All the existing resources can be used for the development and maintenance system.

ii. Operational feasibility study  
The system is easy to operate with the basic knowledge of computer and internet and well trained manpower is not necessary. User can also easily access the system as it is user friendly in many aspects with good User Interface (UI). This system include all the requirements used for Doctor Appointment System and this system is completely operational and can be successfully implemented and administration feel easy to use this system as it is userfriendly.

2.   
iii. Economic feasibility study  
The system is economically feasible and cost effective. As all the tools and resources required are either open sources or free. After the completion of the system organization didn’t need to deploy any new hardware and software as the required software and hardware. The existing resource of the system can be used.

iv. Schedule feasibility study  
The system is completed within scheduled time and do not exceed the scheduled time.

Table 3.1: Gantt chart Table for AGRO system

| **Task Name** | **Duration** |
| --- | --- |
| Getting Started | 2 weeks |
| System Design & Architecture | 2 weeks |
| Implementation | 7 weeks |
| Deployment | 4 weeks |
| Documentation | 12 weeks |

[DIAGRAM HERE bash3|4]

Figure 3.3: Gantt Chart of ONLINE DOCTOR APPOINTMENT SYSTEM

3.1.3 Data Modelling (ER Diagram)  
In the Doctor Appointment System's Entity-Relationship (ER) Diagram, essential entities include "Patient," "Doctor," and "Appointment." Patient and Doctor entities have attributes like ID, name, and contact information. The Appointment entity links patients and doctors, with attributes such as appointment ID, date, and time. Relationships depict associations, such as "Booked By" connecting Patient and Appointment, and "Manages" connecting Doctor and Appointment. Additional entities like "TimeSlot" and "Feedback" may be included. Cardinality notations show the relationships between entities. The ER Diagram serves as a visual representation of how data is structured and related in the Doctor Appointment System, facilitating effective database design.

[DIAGRAM HERE bash5 ]

Figure 3.4: Entity Relational Diagram for ONLINE DOCTOR APPOINTMENT SYSTEM

3.1.4 Process Modelling (DFD)  
Data Flow Diagram of Doctor Appointment System consists of two levels of DFD: context diagram and level one DFD. Both these levels are used for making data flow diagram of Doctor Appointment System.

[DIAGRAM HERE bash6]

Figure 3.6: Level-1 DFD for Doctor Appointment System.

3.2 System Design  
To realize the different functional requirement of the system in graphical form, different design diagram of the system has been prepared which are as follows:

3.2.1 Architectural Design  
For this system, three tier architecture is used which includes user interface, web server and database. In architectural design, basic structure of the system is shown.

[DIAGRAM HERE bash7]

Figure 3.7: Architectural Design of Doctor Appointment System.

3.2.2 System Flowchart  
The system flowchart for the Doctor Appointment System illustrates the sequential steps and interactions within the application. Beginning with the patient's initiation of the appointment request, the flowchart outlines the process of selecting a preferred date and time, checking the availability of the chosen healthcare provider, and confirming the appointment. Simultaneously, the system verifies and updates the appointment schedule. Once confirmed, the system generates automated reminders for both patients and healthcare providers. Additionally, the flowchart includes a feedback loop to capture patient experiences, contributing to continuous system improvement. This visual representation ensures a clear understanding of the end-to-end workflow, fostering efficient and transparent communication in the appointment management process.

[diagram here bash8]

Figure 3.8: Flow chart of ONLINE DOCTOR APPOINTMENT SYSTEM

3.2.3 Database Schema Design  
The database schema design for the Doctor Appointment System involves structuring the database to efficiently store and manage relevant information. The schema includes tables such as 'Patients,' 'Doctors,' and 'Appointments,' establishing relationships between them. The 'Patients' table stores patient details, 'Doctors' table holds information about healthcare providers, and 'Appointments' links patients and doctors with scheduled appointments. Additional tables may include 'TimeSlots' for available appointment times and 'Feedback' for patient reviews. Attributes like patient IDs, doctor IDs, and appointment details form the key connections. The schema design ensures data integrity, facilitates seamless queries, and supports the system's overall functionality for streamlined doctor appointment management.

[Diagram HERE bash9]

Figure 3.10: Database Schema Design of ONLINE DOCTOR APPOINTMENT SYSTEM

3.2.4 Interface Design (UI Interface)  
Interface design is used to design how the AGRO SYSTEM looks like and this design is shown to user that how the system will look. And after finalizing the system development starts. The UI design of home page, register page, login page and dashboard page of AGRO SYSTEM are shown below:

[DIAGRAM HERE bash10]

Figure 3.11: UI Home page of ONLINE DOCTOR APPOINTMENT SYSTEM

[DIAGRAM HERE bash11]

Figure 3.12: Registration page of ONLINE DOCTOR APPOINTMENT SYSTEM

[Diagram here bash12]

Figure 3.13: Login page of ONLINE DOCTOR APPOINTMENT SYSTEM

[DIagram here bash13]

Figure 3.14: Admin Dashboard ONLINE DOCTOR APPOINTMENT SYSTEM

3.2.4 Physical DFD  
Here, users such as farmers and suppliers register and login to the system. If registration and login are successful, users' information is stored in the database and then they can apply and post for suppliers. All the product information and details are stored in the database.

[DIAGRAM HERE Bash14]

Figure 3.15: Physical DFD

CHAPTER-4 IMPLEMENTATION AND TESTING  
4.1 Implementation  
4.1.1 Tools Used (CASE tools, Programming language, Database platforms)

Front End Tools  
HTML  
In Doctor Appointment System, HTML is used for creating different web pages and sites. It is used to create and structure sections, headings, links, paragraphs using various tags and elements. Headers, paragraphs, links, and images of Doctor Appointment System are defined using HTML.

CSS  
In Doctor Appointment System, CSS is used for designing different tags of HTML. It is also used to design different components using classes and IDs. Different types of CSS such as inline CSS, internal CSS, and external CSS are used to design this system. CSS is used for defining styles for web pages, controlling text color, font style, spacing between paragraphs, column sizing, layout designs, and more.

Back End Tools  
PHP  
In Doctor Appointment System, PHP is used for backend purposes and for creating dynamic web pages. It is used for server-side scripting to add connectivity to the database, encrypt data, validate user data, control user access to certain pages and login pages. PHP also includes functionalities to add, update, and delete data from the database.

Server APACHE SERVER  
In Doctor Appointment System, Apache server is used to run PHP files and create fast and dynamic web pages.

Database MYSQL  
MySQL is used for storing all the information required in the database for the Doctor Appointment System. It performs CRUD operations such as create, delete, and update data as requested by the user.

Documentation Tools  
MS Office  
This is used for writing and editing the documentation of Doctor Appointment System.

Draw.io  
This is used to generate diagrams for system analysis and design of Doctor Appointment System. Diagrams were created using this tool to save time with drag and drop functions.

4.1.2 Implementation Details of Modules (Description of procedures/functions)  
Different modules of this system are described as below:

Login Module  
In the login module, the Doctor Appointment System involves a secure and user-friendly authentication process to ensure access control. Both patients and healthcare providers typically have distinct login interfaces. For patients, the login page captures unique identifiers such as username or email and password. Additional security measures like two-factor authentication may enhance protection. Healthcare providers log in with professional credentials, including a unique identifier and password. The system verifies these credentials against stored records in the database. Upon successful login, users gain access to personalized dashboards, allowing patients to schedule appointments and view health information, while healthcare providers manage appointments and patient records. Security protocols and encryption mechanisms are integral to safeguard sensitive healthcare data.

Register Module  
In the register module, we have implemented two sub-modules: patient register and doctor register. Doctors and patients register into the system by entering all the required details such as firstname, lastname, username, and password. They can then log in to the system with their valid username and password.

4.2 Testing  
System testing is done by giving different training and testing datasets. This test is done to evaluate whether the system is providing accurate summary or not. During the phase of the development of the system, our system is tested time and again. The series of testing conducted are as follow:

4.2.1 Test Cases for Unit Testing  
In unit testing, we designed the entire system in modularized pattern and each module is tested. Until we get the accurate output from the individual module, we work on the same module. The input form is tested so that they do not accept invalid input.

 Patent Registration  
Table 4.1: Test case for Online appointment system

| **S.N.** | **Test Name** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- |
| 1. | Open Application | projectregistration.PNG | Online appointment system Register Page | Online appointment system Register Page | Pass |
| 2. | Enter Valid firstname, Lastname, username, Password And click register button | projectloginpage.PNG FirstName: Amrit LastName: Sunam Username: amrit Password: 123G@S | Registration Pass | Registration Pass | Pass |

 Patent Registration  
Table 4.2: Test cases for Online appointment system

| **S.N.** | **Test Name** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- |
| 3. | Enter username, password and click register button | FirstName: Puja LastName: Rai UserName: P Rai Password: vbxcfg | Select Password Registration failed | Pass |  |

 Doctor Registration  
Table 4.3: Test cases for login of Online appointment system

| **S.N.** | **Test Name** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| --- | --- | --- | --- | --- | --- |
| 1. | Open Application | projectindexpage.PNG | Home page | Home page | Pass |
| 2. | Enter Invalid Username, Password and click Login button | Username: amrit Password: shdgft | Select Password Login Failed | Pass |  |
| 3. | Enter valid Username, Password and click Login button | Username: Rsewa Password: xyz.36 | Home Page | Home Page | Pass |

CHAPTER-5  
CONCLUSION AND FUTURE RECOMMENDATION

5.1 Lesson Learnt / Outcome  
Every project provides opportunities to learn and gain knowledge in various aspects. In this project, we acquired valuable problem-solving skills, teamwork experience, proficiency in following guidelines, communication and writing skills, and team management abilities.

**Teamwork**  
Being a team project, it taught us effective collaboration with team members in developing the system. We learned task delegation, teamwork, and resolving issues and errors encountered during system development.

**Problem Solving Skills**  
The project enhanced our problem-solving skills significantly. We gained experience in identifying and resolving various errors that occurred in the system.

**Writing Skills**  
We learned how to prepare proposals and documentation related to the project. Additionally, we acquired proficiency in using different CASE tools for creating diagrams such as use case diagrams, schema diagrams, data flow diagrams, and ER diagrams.

**Time Management**  
A critical lesson learned was efficient time management, especially in prioritizing system components based on complexity.

5.2 Conclusion  
In conclusion, the Doctor Appointment System represents a transformative solution to challenges in traditional healthcare appointment management. Through systematic design, development, and implementation, critical issues such as appointment inefficiencies, limited accessibility for patients, and operational burdens on healthcare providers have been addressed. The system's use of technology not only streamlines scheduling for patients but also improves workflow for healthcare professionals, enabling them to focus on delivering optimal patient care. While acknowledging limitations such as technological barriers and privacy concerns, the system's overall impact on enhancing patient experiences and operational efficiency underscores its significance in modern healthcare administration. The Doctor Appointment System marks a significant step towards a patient-centric, accessible, and efficient healthcare ecosystem.

5.3 Future Recommendations  
The development project could have been more efficiently handled in terms of design and development processes. Enhancements in the documentation process could have ensured more comprehensive coverage of all aspects Programming the project prior to any documentation. The system can be updated based on the users’ requirements recommendation. The page load and server load speed might be improved.

Some future recommendations for this system are:

1. **Advanced Reports:** Consider adding more detailed reports to assist healthcare administrators in making informed decisions. These reports could include insights into appointment trends and healthcare provider performance
2. **Telemedicine:** Explore adding the capability for patients to have video appointments with healthcare providers. This enhances convenience, particularly for patients who cannot visit in person.
3. **Better Mobile App:** Continue improving the mobile app to simplify appointment booking and management from smartphones. Given the widespread use of mobile devices, a user-friendly mobile experience is crucial.
4. **Smart Scheduling:** Investigate using intelligent scheduling algorithms to optimize appointment scheduling. These algorithms can consider healthcare providers' availability and patients' preferences, improving the scheduling process.
5. **Reducing No-Shows:** Implement predictive models using computer programs to forecast when patients might miss appointments. This helps in proactive planning and minimizes disruptions to the schedule.
6. **Connecting with Health Records:** Consider integrating the system with electronic health records (EHRs) to facilitate quick and easy access to essential patient information for healthcare providers.

**REFERENCES**

[1] Wiki.optimy. [https://wiki.optimy.com/HMS-system/](https://wiki.optimy.com/HMS-system/" \t "_new)  
[2] [https://nevonprojects.com/farming-assistant-web-service-php/](https://nevonprojects.com/farming-assistant-web-service-php/" \t "_new)  
[3] E Balagurusomy, “Fundamentals of Computer”, Tata McGraw Hill Education PVT. Ltd.  
[4] Thomas A. Powell, “Web Design: The Complete Reference 2ED”, Tata McGraw Hill.  
[5] L. B. Jeffery Whitten, “Systems analysis and design methods”, McGraw Hill India.  
[6] R. Lafore, "Data Structures and Algorithms", Sams Publishing.  
[7] Ian Sommervilla, “Software Engineering”, 9th Edition, Addison – Wesley, 2010.  
[8] R. Robin Nixon, “Learning PHP MySQL JavaScript and CSS”, 2nd Edition, O'Reilly Media, 2012.  
[9] A. Abraham Silberschatz, “Database System Concepts”, 6th Edition, McGraw Hill.  
[10] D. S. McFarland, “JavaScript and jQuery: The Missing Manual”, 2nd Edition, Pogue Press, 2011.