CHAPTER IV

# Design and Methodology

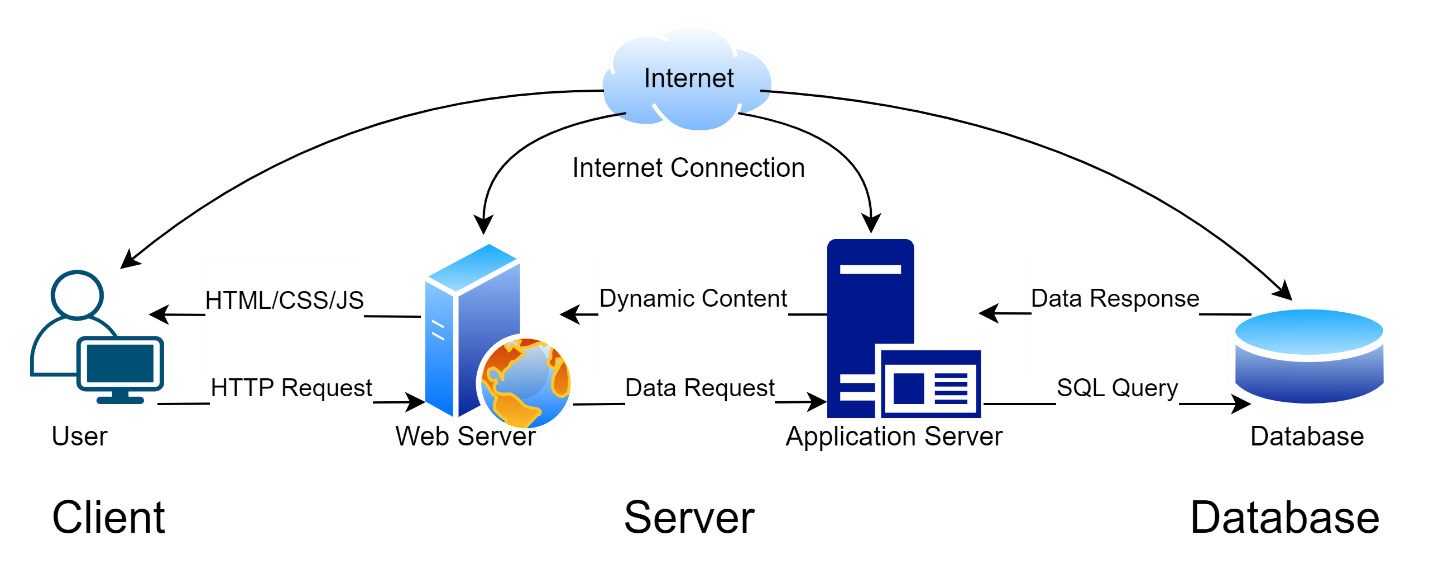
This chapter provides a detailed explanation of the overall approach, tools, techniques, and procedures used to address the system’s objectives. This also includes various diagrams, figures, and tables that can further help visually discuss the development model and approach used in this study.

4.1 Concept

The proposed system is designed specifically for Immaculate Medico-Surgical Clinic in Bulan, Sorsogon. The project aims to develop a web-based health records management system with medication alert notification system. The project follows the Iterative Incremental development model, and RAD methodology approach. Additionally, this chapter discusses the development tool used by the proponents in developing the system.

4.1.1 System Architecture and Design

The system architecture diagram presents a comprehensive overview of the system's structure and functionality. At its core, the internet connection serves as the foundation, enabling seamless communication between four essential components, each playing a central role in the system's operation.



# Figure 4.1 System Architecture Diagram

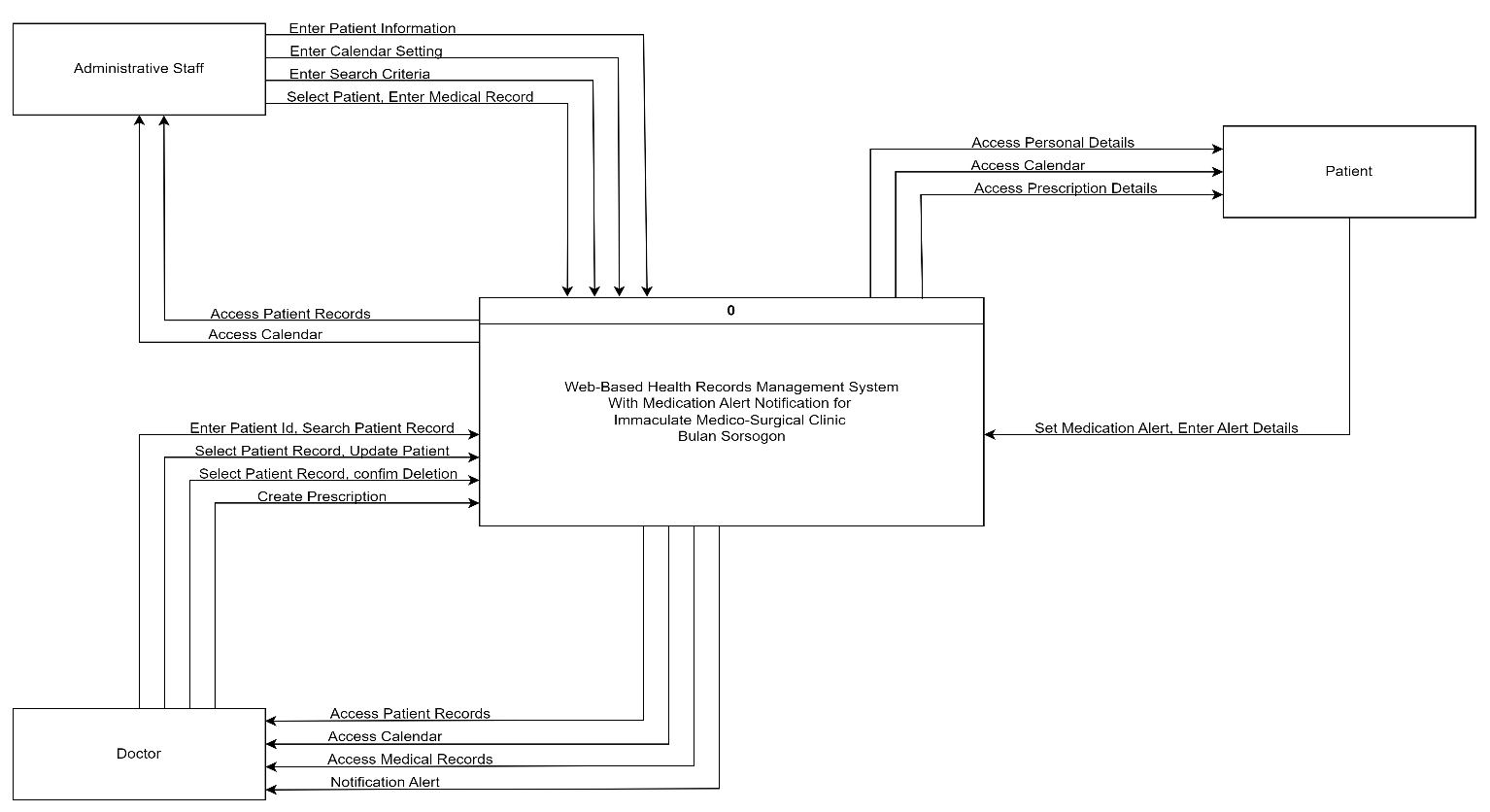
In this architecture, the internet connection serves as the foundation, facilitating communication between four main components, each serving a distinct role in the system. The first component is the user interface, where end-users interact with the system. It could be through a web browser. The second component, the web server, is responsible for handling HTTP/HTTPS requests from users. It manages the presentation logic of the application, generating HTML, CSS, and JavaScript to be rendered in the user's interface. Before connecting to the internet, the web server needs to have access to it to process incoming and outgoing requests. The third component, the application server, executes the core business logic of the application. It processes requests received from the web server, interacts with the database to retrieve, or modify data, and generates dynamic content to be sent back to the user. Like the web server, the application server also requires access to the internet to communicate with external services. The fourth component, the database server, which stores and manages the data used by the application, is integrated within the internet, allowing for connectivity and data exchange between the application and its users. Together, these components form a three-tier architecture that separates concerns and allows for better scalability, maintainability, and efficient data management in the system.

4.1.2 Data Flow Diagram

Data Flow Diagram (DFD)

The Data flow illustrate the flow of the data and process within the web-Based

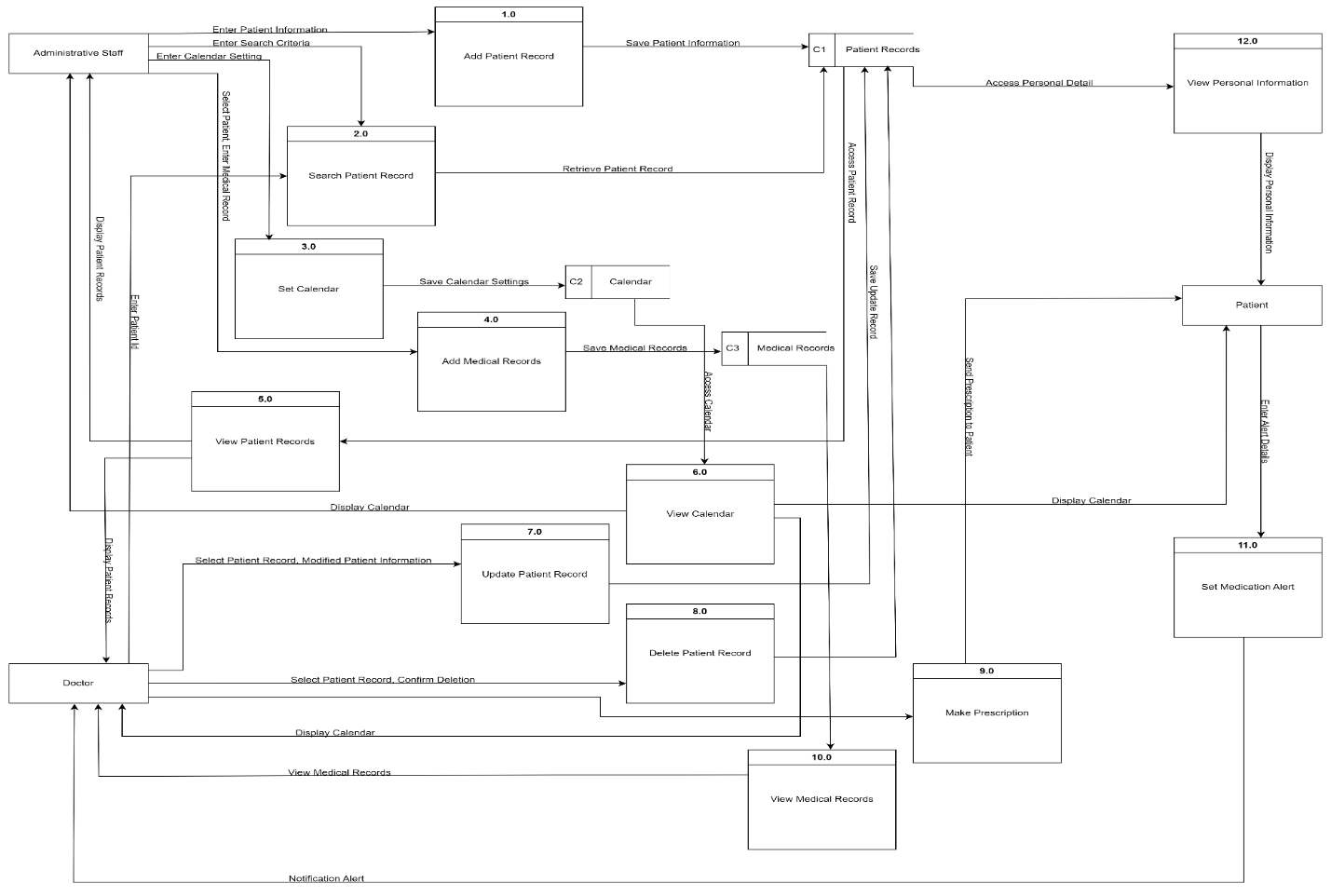
Health Records Management System with Medication Alert Notification for Immaculate Medico-Surgical Clinic in Bulan Sorsogon. Highlighting the interaction and task performed by each entity.



# Figure 4.2 Context Diagram Level 0

At level 0 Context Diagram, The System involves three entities:

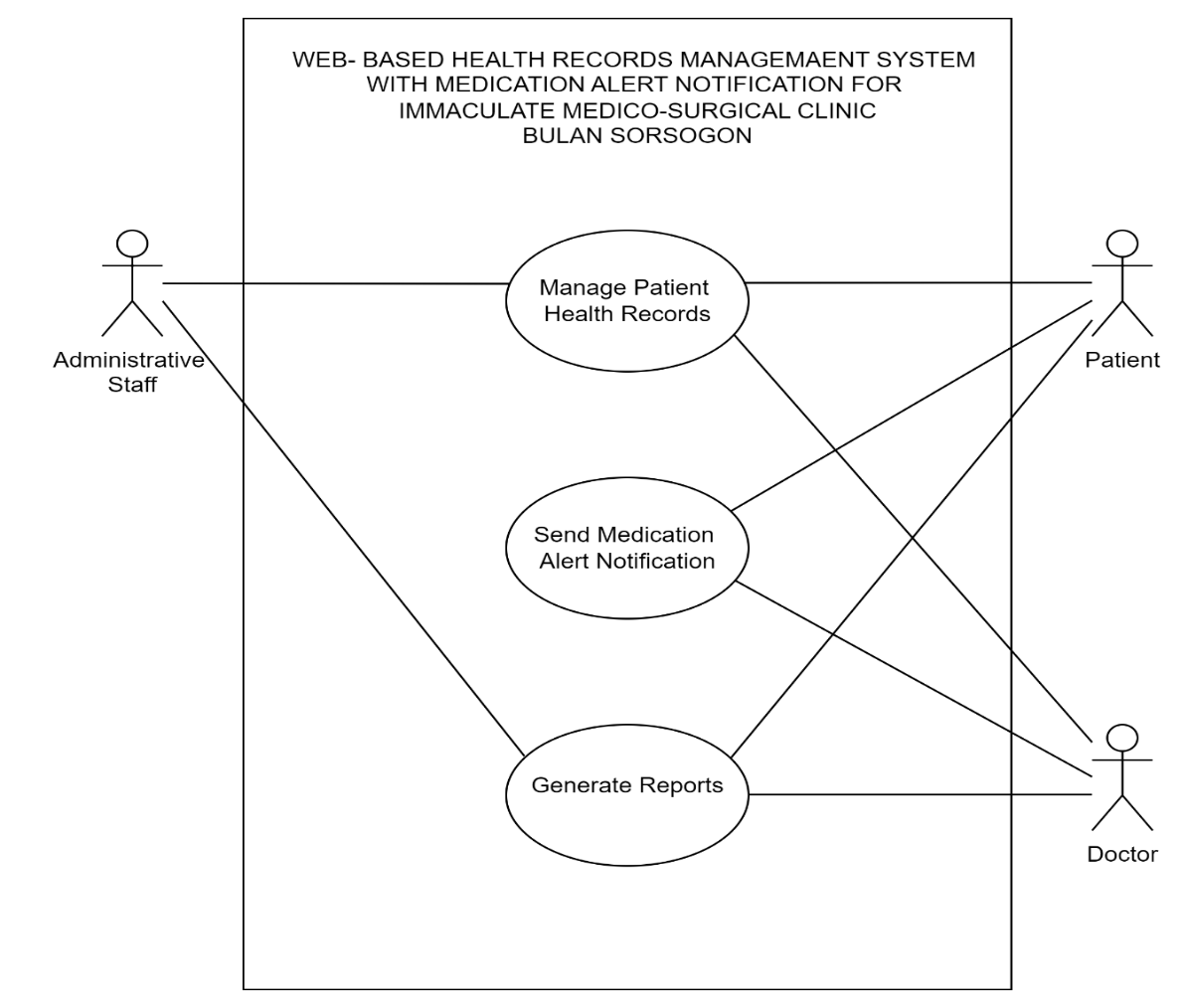
Administrative Staff, Doctor, and Patient, each with specific roles and processes. For the Administrative Staff, the task includes adding patient information, setting the calendar by inputting calendar settings, search patient records based on criteria, adding medical records by selecting a patient and entering medical information, and viewing patient records and calendar by accessing the respective data flow. The doctor entity is responsible for searching patient records using patient id, viewing patient records, updating patient information by modifying records, deleting patient records upon confirmation, making prescription to patient, and accessing the calendar and receive notification alerts. Patient entity on the other hand can view their personal information, access the calendar, view prescription details, and set medication alert.



# Figure 4.3 Data Flow Diagram Level 1

The level 1 of the Data Flow Diagram, the process becomes more detailed. For instance, Administrative Staff are now responsible for saving patient information after inputting it into the system, saving calendar settings, retrieving patient records based on specific criteria, saving medical records after input, displaying patient records, and calendar information. Doctors have the ability to retrieve patient records using the patient ID, save updated patient information, delete records, when necessary, send prescriptions to patients, and receive and view information alerts. Patients now have the option to access and view their personal information, calendar details, prescription information, and set medication alerts by entering alert details which are then processed as notification alerts. This level of the Data Flow Diagram highlights the various tasks and responsibilities of different users within the system, emphasizing the importance of accurate data management and communication between all parties involved in the healthcare process.

4.1.3 UML Use Case Diagram

A Unified Modeling Language (UML) use case diagram provides a high-level overview of the interactions between actors (users or external systems) and the system being modeled. It serves as a visual representation of the functional requirements of the system, capturing the various ways users interact with it to achieve specific goals or tasks. In essence, a use case diagram offers a simplified yet

comprehensive view of the system's functionality, helping stakeholders understand the system's scope, requiements, and intended behavior.

only capable

in sending and receiving the medication alert

notification;

lastly the

system can generate reports and all the actors are capable in viewing

these reports

.

Each objective will be discussing the following diagrams

.

A document with black text

Description automatically generated

# Figure 4.4 Main Use Case Diagram

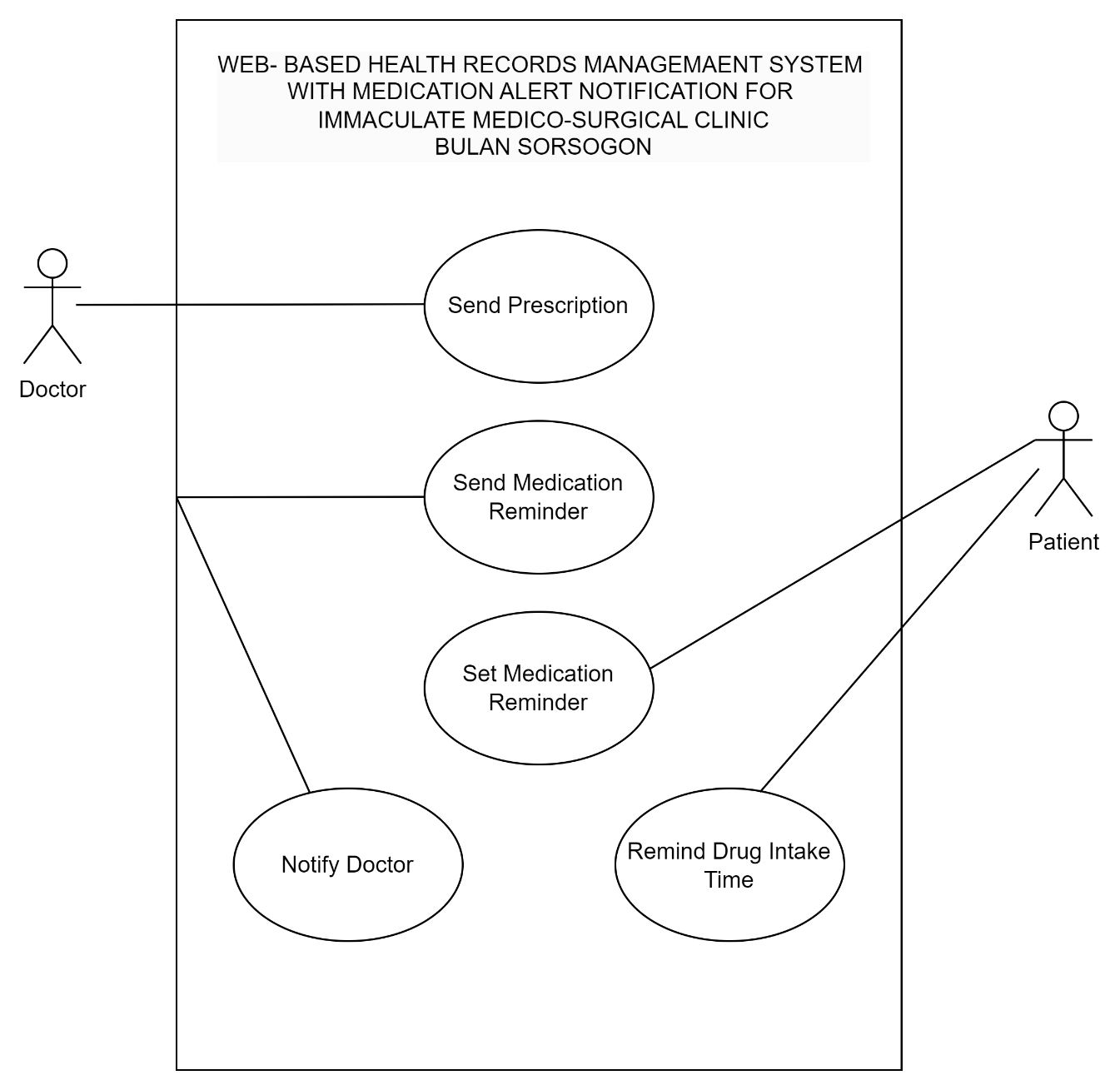
The Uml Diagram (Use Case) for the Web-Based Health Records Management System with Medication Alert Notification for Immaculate Medico-Surgical Clinic in Bulan Sorsogon shows the main objectives of the system. The actors include Patient and Doctor as the reactor, and Administrative Staff as the initiator.

The first main use case in the diagram is to manage the patient health records, all the actors are capable in accessing the patient’s health records, but each other has distinct role in they interact with this information. Next the patient and the doctor it Figure 4.5 Manage Patient Health Records Use Case Diagram

The Administrative Staff can view patient list, includes add patient records that is required in viewing the patient records. However, the use case Search patient records, update patient records and view more about patient are optional. Additionally, the Administrative Staff sets a calendar for viewing of availability of clinic days and hours, also the Administrative Staff can add medical records into the system.

The patient has limited access to the system that includes viewing of their personal records and calendar viewing.

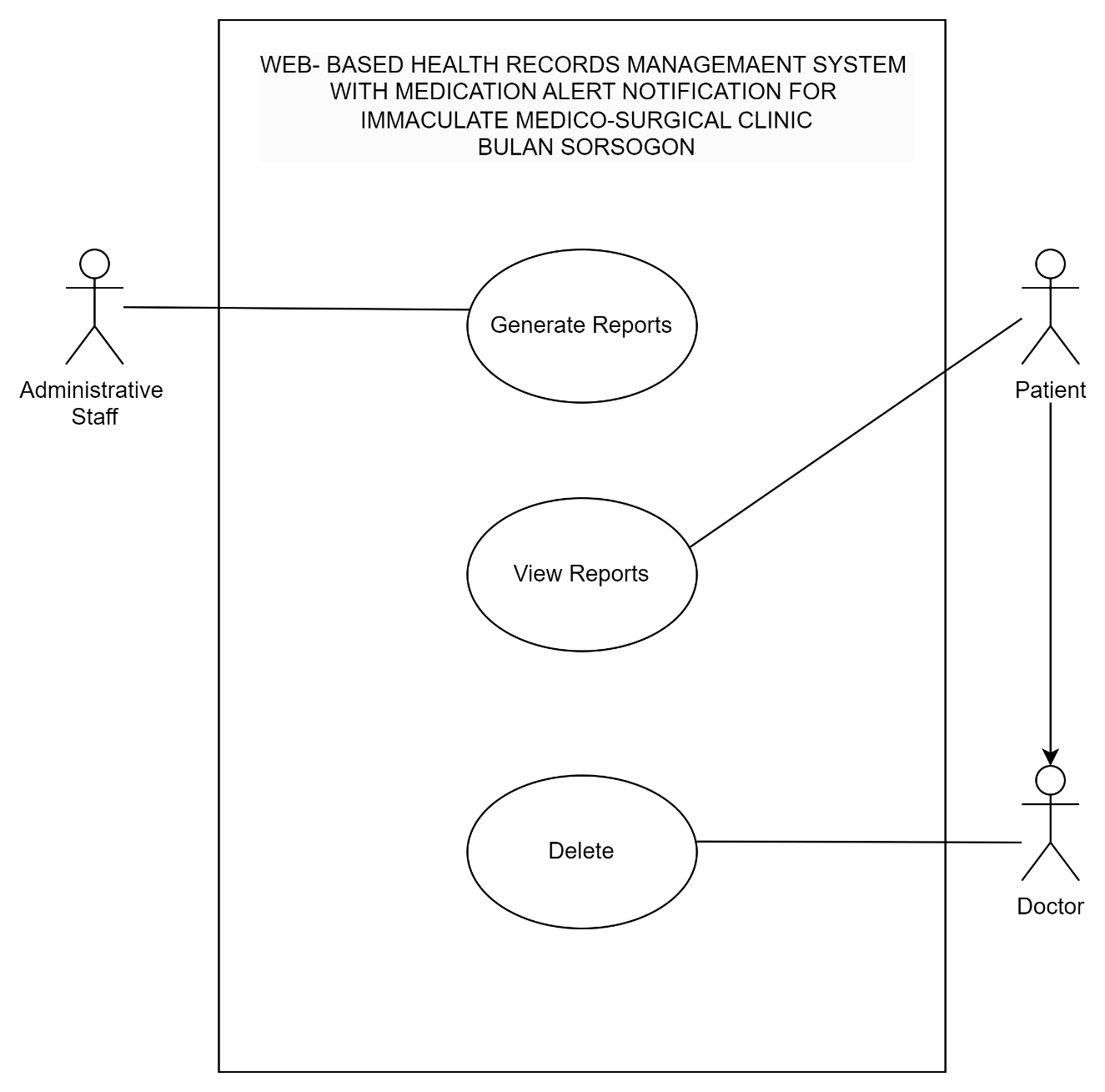
The doctor is capable to search patient record using patient id and patient name, it includes view patient records to view the list of patients, if the doctor wants to update the patient records but its optional to update the records and extend to delete the patient records.



# Figure 4.5 Send Medication Alert Notification Use Case Diagram

The Doctor will send the prescription to the patient after that the patient will set the medication reminder based on the prescription. The system will send medication reminders to the patient who will be reminded of the drug intake time.

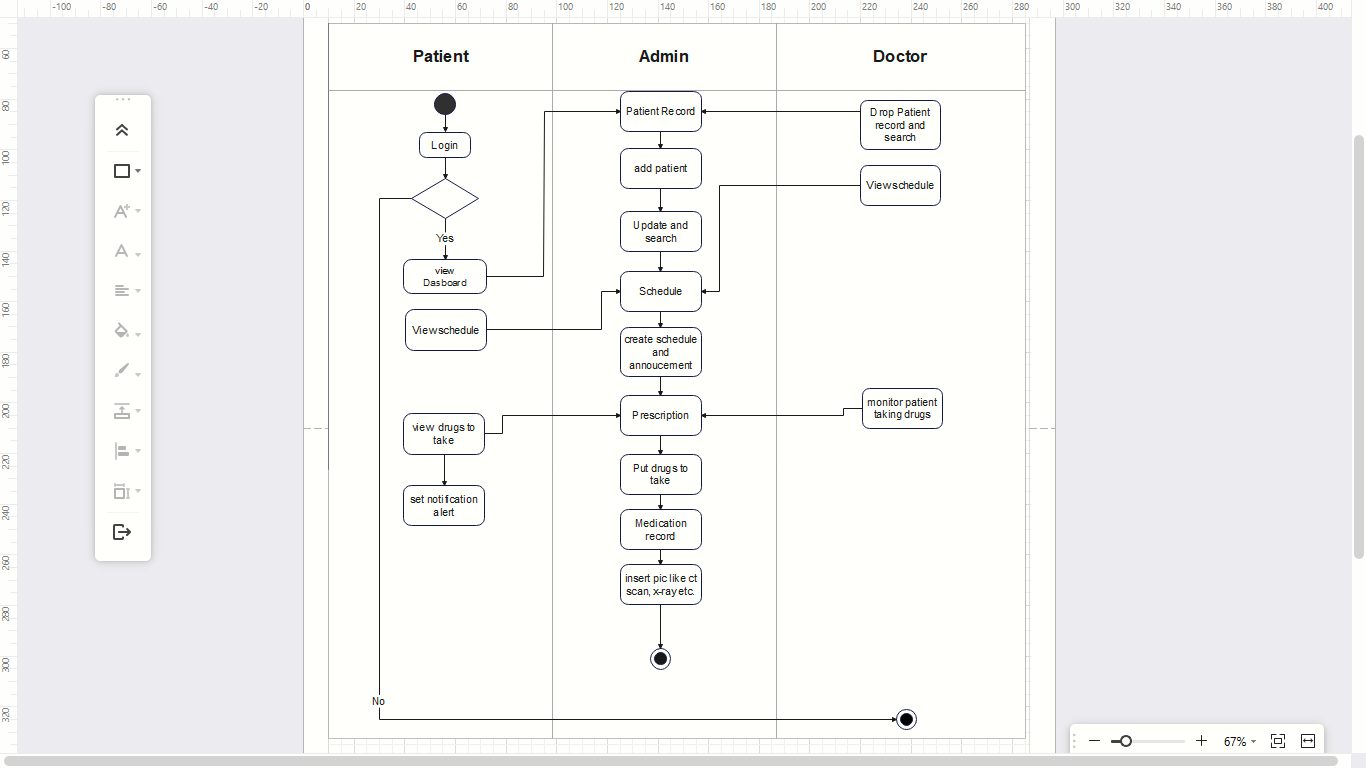
Additionally, the system will notify the doctor to track the medication adherence of the patients and ensure that the prescribed treatment is being followed.



# Figure 4.6 Generates Reports Use Case Diagram

The Administrative Staff of the clinic will insert all the patient medical information, medical history summaries, and medication adherence reports, collected during the cause of care. The Patient can view the generated reports as well the Doctor. But only the doctor can delete generated reports when it’s not needed anymore.

4.1.4 UML Activity Diagram

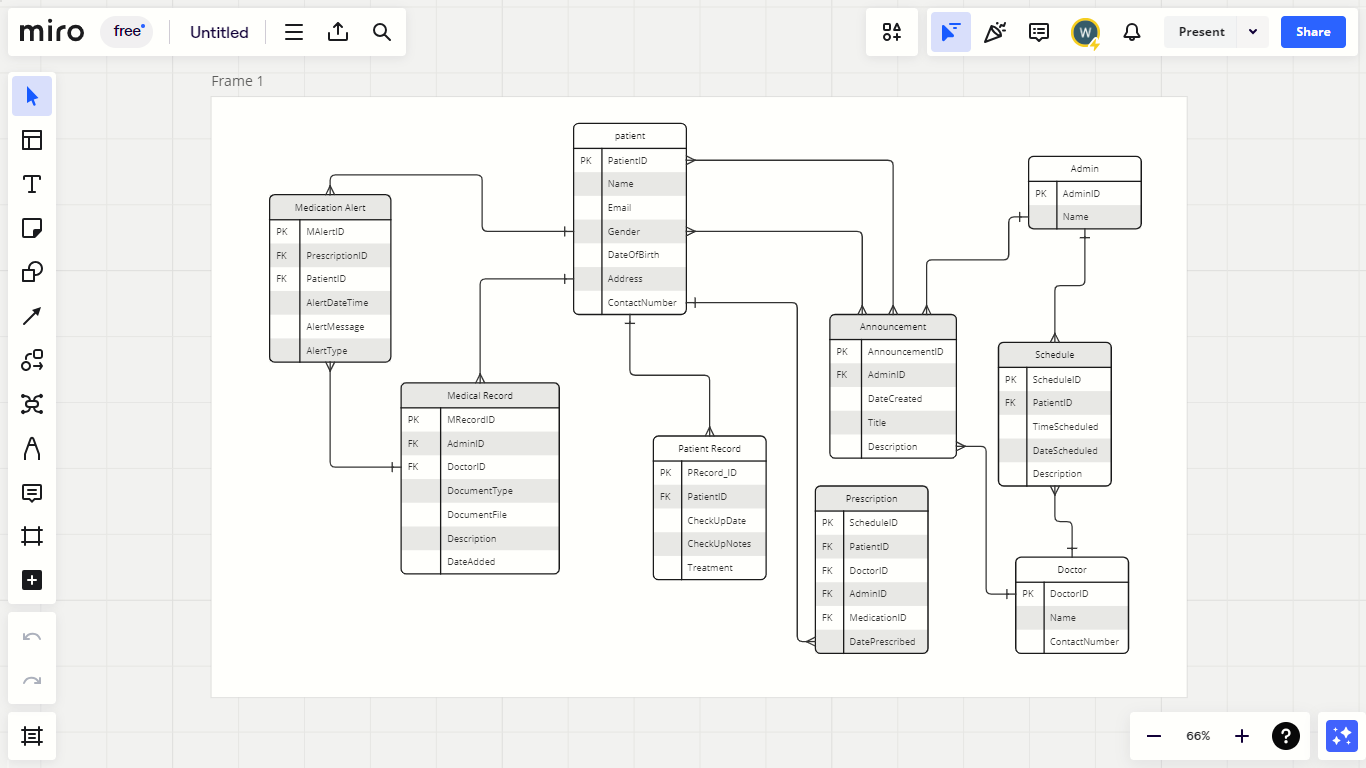


# Figure 4.7 UML Activity Diagram

The diagram below illustrates how users interact with our system. Among the patients, their activities are limited to viewing the dashboard, accessing their medical records, checking their schedules and announcements, reviewing their prescriptions, and setting notification alerts. In case of a patient's demise, a doctor is authorized to remove their record and search for it.

The dashboard, prescription, schedule, and announcement features are designed solely for viewing purposes. The administrative team is responsible for all related tasks such as adding, updating, and searching for patient records, scheduling appointments, and posting announcements. When it comes to prescriptions, the admin can enter the medicines that the patient is taking, the dosage, and the number of days per drug. Lastly, the medication record allows the admin to upload the patient's documents such as X-rays, CT scans, and other relevant files.

4.1.5 Entity-Relationship Diagram



# Figure 4.6 Entity-Relationship Diagram

The system architecture described in this document appears to be a comprehensive and well-planned approach for managing patient medical records. It includes various entities such as Patient, Admin, Doctor, Patient Record, Prescription, Schedule, Announcement, Medication Alert, and Medical Record that work together to create a complete ecosystem for managing patient health information. The relationships between these entities ensure that the information is organized and easily accessible to the relevant parties. For example, the one-to-many relationship between Patient and Patient Record allows for the tracking of their medical journey over time, while the many-to-one relationship between Medical Record and Patient ensures that medical documents are accessible to the right patient. The system also provides personalized medication management through the many-to-one relationship between Medication Alert and Patient, which is useful for ensuring medication adherence and timely reminders for patients to take their medication

4.2 Analysis and Design

This section outlines the process to ensure that it meets the specific needs of Immaculate Medico-Surgical Clinic that lays for the development of the system.

4.2.1 Requirement Analysis

Requirement analysis is an essential aspect in developing a system. Analyzing requirements is an essential aspect of developing a system. The following tables detail the functional and non-functional requirements that the system complies with. The task requirements and task reference directly link to the objectives of this study.

4.2.1.1 Functional Requirements

Functional Requirements presents the features that must be implemented in the system and to be used by the users. This section presents the functional requirements of the system specifically its task description and reference.

# Table 4.1. Functional Requirements

|  |  |
| --- | --- |
| Task Requirements | Task Reference |
| The system must have the capability to store and manage patient health records securely and efficiently. | Patient Health Records Management |
| The system must help to ensure  medication adherence and patient safety by providing timely reminders to patients for taking their prescribed medications. | Real-time Medication Reminders |
| The system should generate reports to  provide valuable insights and  information. | Generates Needed Reports |

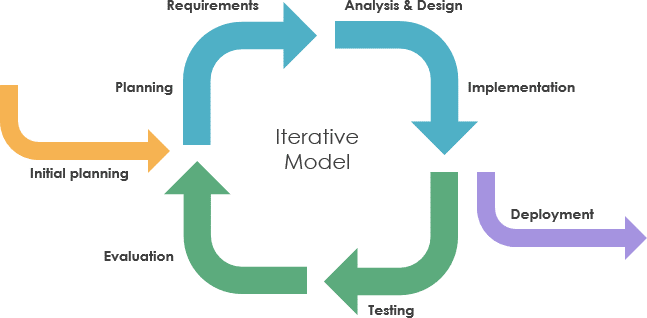
4.2.1.2 Nonfunctional Requirements

Nonfunctional requirements, also known as the system’s quality attributes, dictate how the system should behave, defining its quality characteristics. This section presents the nonfunctional requirements of the system, with the stated references adapted from the Software Quality Standards of the International Organization for Standardization, specifically in ISO 25010.

# Table 4.2 Nonfunctional Requirements

|  |  |
| --- | --- |
| Task Requirements | Task Reference |
| The system must be able to perform all the functional requirements and provide the appropriate results. | Functional Suitability |
| The system must be accessible to its different users and has an intuitive and user-friendly interface that is easy to navigate and understand. | Usability |
| The system must be compatible across different browser, operating system, and devices, particularly those commonly used by the clinic’s staff and patients. | Compatibility |

4.2 Development Model



The terminology Iterative Incremental Model is a software development methodology or software development process model which collaborates aspects of the iterative model and the incremental model. The Iterative Incremental Model is a smart way to design complex systems. Instead of trying to figure everything out at the beginning, this approach breaks the project into smaller parts. Each part is then developed, tested, and added to the system, bit by bit. This method is great for projects where things might change or if the requirements aren’t clear from the start. It allows designers to focus on making each part work well before adding more, leading to a better overall system.

1. Planning Phase

In this phase, the team identifies the goals and objectives of the project, along with the project scope, requirements, and constraints on them. The team then identifies different iterations that would be needed to complete the project successfully.

1. Requirements Analysis and Design Phase

In this phase, the requirements met are then analyzed and the according system is designed based on these requirements. The projected design should be modular, which would allow easy modification and testing in subsequent iterations.

1. Implementation Phase

In this phase, the system is implemented based on the design created in the previous phase. The implementation should be done in small, manageable pieces or increments, which can then be tested in the next phase of the cycle.

1. Testing Phase

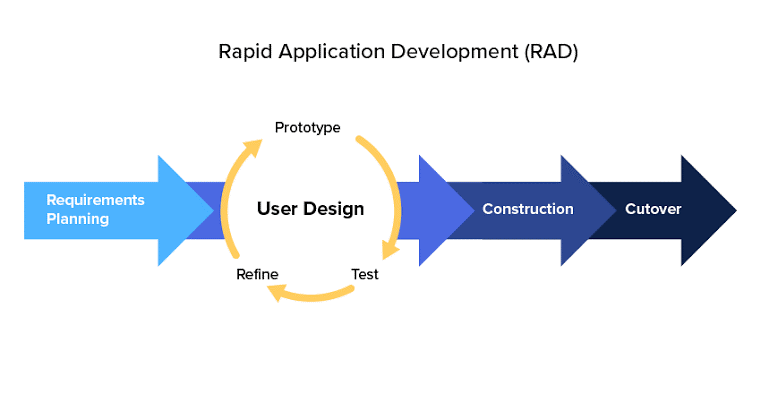
In this phase, the system is tested against the requirements identified in the planning phase. Testing is done for each iteration, and any defects or issues are identified and resolved, and this helps in each iteration. 5. Evaluation Phase

In this phase, the team evaluates the performance of the system based on the results of testing. Feedback is gathered from users and stakeholders, and changes are made to the system as needed which makes the system more scalable and flexible.

6. Incremental Release

In this phase, the completed iterations are released to users and stakeholders. Each release builds on the previous release, providing new functionality or improving existing functionality to a great extent.

4.4 Development Approach



The proponents selected the RAD (Rapid Application Development) approach due to its capacity to provide a quick and adaptable solution for the Web-based health records management system with medication alert notification at Immaculate Medico-Surgical Clinic in Bulan, Sorsogon. RAD's focus on rapid prototyping and iterative development matched the urgency of the project. This approach facilitated active involvement of clinic staff and patients, ensuring alignment with their specific requirements and workflows. Through iterative cycles, issues were identified and addressed early on, mitigating risks and ensuring the final product's effectiveness. RAD's flexibility enabled adaptation to change in healthcare regulations and technological advancements efficiently. Overall, the proponents selected the RAD approach for its speed, adaptability, and cost-effectiveness, making it an ideal fit for the clinic's needs and constraints.

By embracing RAD, the proponents aimed to streamline the development process, allowing for quick turnaround times and continuous improvement. This approach enabled them to swiftly develop prototypes and gather feedback from stakeholders, ensuring that the system met the evolving needs of Immaculate Medico-Surgical Clinic. The iterative nature of RAD allowed for flexibility in accommodating changes and enhancements, ensuring that the final product would be both functional and user-friendly. Additionally, RAD's emphasis on active user involvement fostered a sense of ownership among clinic staff and patients, leading to greater acceptance and adoption of the system. Overall, the proponents saw RAD as the ideal methodology to deliver a high-quality, tailored solution that would effectively address the clinic's requirements while staying within budget and timeline constraints.

4.5 Development Tools

The following are the software development tools and applications used in developing the system:

• Frontend Development Tools:

HTML (Hypertext Markup Language):



Conceptual Definition: HTML is a standard markup language used to create the structure and content of web pages, serving as the backbone of the web pages. It provides a way to organize and format content on the internet.

Operational Definition: The page structure was implemented using HTML [15]. HTML defines the elements of a webpage, such as headings, paragraphs, links, and images, ensuring proper content display and semantic structure.



CSS (Cascading Style Sheets):

Conceptual Definition: CSS is a styling language that controls the presentation and layout of HTML elements on a webpage. It allows for the separation of content from presentation, enabling developers to style web pages effectively.

Operational Definition: The presentation and styling were achieved using CSS [16]. CSS defines the visual style, design, and formatting of web content, including colors, fonts, spacing, and responsiveness. It enhances the user experience by providing a visually appealing and consistent interface across different devices and screen sizes.



Bootstrap:



Conceptual Definition: Bootstrap is a front-end framework that simplifies the design and development of responsive and mobile-first websites. It provides a collection of pre-built components and tools for faster and easier web development.

Operational Definition: Bootstrap offers ready-made CSS styles, responsive grid layouts, and JavaScript plugins for creating consistent and visually appealing web interfaces. Bootstrap was utilized to create a user-friendly and accessible interface [17]. It streamlines the development process by providing a flexible and customizable framework that adapts to various design requirements and devices.

JavaScript (JS):

Conceptual Definition: JavaScript is a programming language used for adding interactivity and dynamic behavior to web pages. It enables developers to create interactive elements and enhance user engagement on websites.

Operational Definition: JavaScript allows developers to manipulate the Document Object Model (DOM), handle user events, validate input forms, and communicate with servers asynchronously. And JavaScript was employed for GUI behaviors and alerts [18]. It plays a crucial role in creating dynamic and interactive web applications, making them more responsive and engaging for users.



Google Chrome Browser:

Conceptual Definition: Google Chrome is a cross-platform web browser developed by Google. It is known for its speed, simplicity, and security features, and is one of the most popular browsers worldwide.

Operational Definition: Google Chrome provides developer tools for inspecting and debugging web pages, monitoring network activity, and optimizing performance. It allows developers to test their web applications in a real-world environment and ensures compatibility across different browsers and devices. Additionally, Chrome supports various extensions and plugins for web development, enhancing the overall development workflow.

• Backend Development Tools:

PHP:



Conceptual Definition: PHP is a server-side scripting language designed for web development. It is used to create dynamic web pages and interact with databases to generate content on the fly.

Operational Definition: PHP was utilized as the backend programming language for the system [19]. PHP scripts are executed on the server to process data, handle form submissions, authenticate users, and perform various backend tasks. It facilitates server-side programming and enables the development of dynamic web applications with database integration.

MySQL:



Conceptual Definition: MySQL is an open-source relational database management system (RDBMS) that uses SQL (Structured Query Language) to manage and manipulate data.

Operational Definition: MySQL is used to store, retrieve, and manage data in backend applications. It offers features like data security, scalability, and ACID compliance, making it a popular choice for web applications requiring structured data storage. PHP was utilized as the backend programming language for the system [20]. MySQL interacts with backend programming languages like PHP to perform database operations.

XAMPP:



Conceptual Definition: XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of Apache HTTP Server, MariaDB database, and interpreters for scripts written in PHP and Perl.

Operational Definition: XAMPP provides a local development environment for PHP-based web applications. It includes Apache as the web server, MySQL as the database server, and PHP for server-side scripting. XAMPP simplifies the setup of a local server environment for testing and development purposes, allowing developers to work on web projects locally before deploying them to production servers. and XAMPP local server was utilized for data storage [21].

• Code Editor:

Visual Studio Code (vscode):

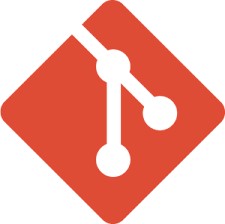


Conceptual Definition: Visual Studio Code (VS Code) is a lightweight, cross platform source=code editor developed by Microsoft, designed to facilitate efficient coding across multiple programming languages with features such as syntax highlighting, code completion, and debugging

tools.

Operational Definition: Visual Studio Code (VS Code) provides developers with a versatile tool for writing, editing, and debugging code across multiple programming languages.

Version Control System Git:



Conceptual Definition: Git is a distributed version control system designed to track changes in source code during software development projects.

Operational Definition: Git provides a system for tracking changes to source code files. It allows developers to create branches to work on different features or fixes independently, merge changes back into the main codebase, and resolve conflicts that may arise when multiple developers modify the same files.

4.6 Schedule and Time

This section introduces the estimated schedule and timeline. The developer uses the Work Breakdown Structure (WBS), Critical Path Method (CPM), and Gantt Chart to identify the tasks, their patterns, and the duration required to complete the project.

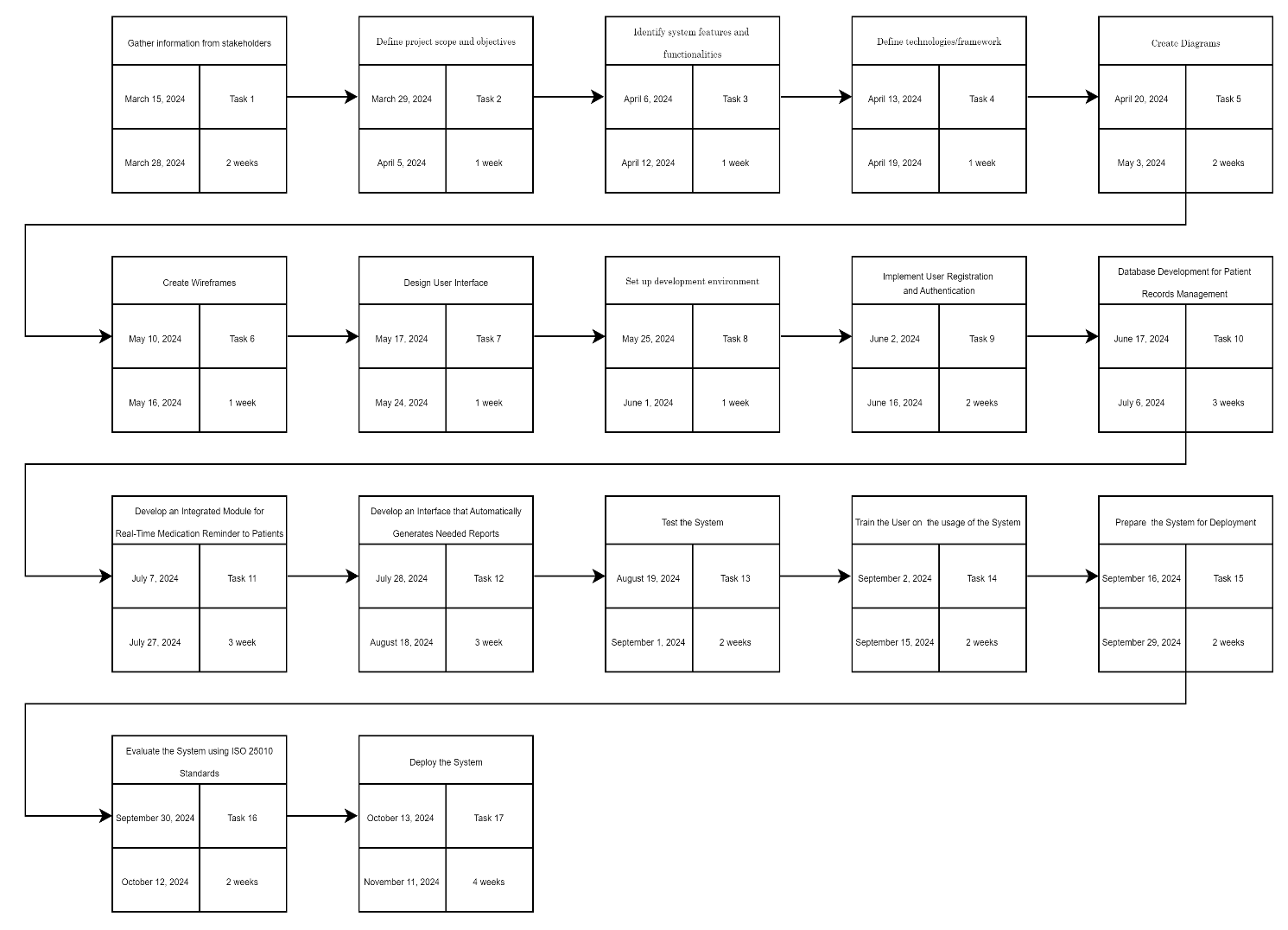
# Table 4.3 Work Breakdown Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task ID | |  | Task Name | Duration (week) | Predecessor |
| PHASE  Planning | | 1: | | |  |
| Task 1 | |  | Gather information  from stakeholders | 2 |  |
| Task 2 | |  | Define project scope and objectives | 1 | 1 |
| PHASE  Requirements | | 2: | | |  |
| Task 3 | |  | Identify system features and  functionalities | 1 | 2 |
| PHASE | 3: | |  |  |  |
| Analysis Design | and | |  |  |  |
| Task 4 |  | | Define  technologies/framework | 1 | 3 |
| Task 5 |  | | Create diagrams | 2 | 4 |
| Task 6 |  | | Create wireframes and UI mockups | 2 | 5 |
| Task 7 |  | | Design user interface | 2 | 6 |
| PHASE  Implementatio | 4: n | |  |  |  |
| Task 8 | | | Set up development environment | 1 | 7 |
| Task 9 | | | Implement user registration and  authentication | 2 | 8 |
| Task 10 | | | Database development for Patient Records Management | 3 | 9 |
| Task 11 | | | Develop an integrated module for real-time medication reminders to patients | 3 | 10 |
| Task 12 | | | Develop an interface that automatically generates needed reports. | 3 | 11 |
| PHASE 5: Testing | | |  |  |  |
| Task 13 | | | Test the System | 2 | 9,10,11,12,13 |
| Task 14 | | | Train the user on the usage of the system | 1 | 14 |
| Task 15 | | | Prepare the system for deployment | 2 | 15 |
| PHASE 6:  Evaluation | | |  |  |  |
| Task 16 | | | Evaluate the system using ISO 25010  standards | 3 | 16 |
| PHASE 7:  Deployment | | |  |  |  |
| Task 17 | | | Deploy the system | 3 | 17 |

Table 4.4 Gantt Chart

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task ID | | | Task Name | Start Date | End date | Duration (weeks) |
| PHASE 1:  Planning | | |  | |  |  |
| Task 1 | | | Gather information  from stakeholders | 03/15/2024 | 03/28/2024 | 2 |
| Task 2 | | | Define project scope and objectives | 03/29/2024 | 04/05/2024 | 1 |
| PHASE 2:  Requirements | | |  | |  |  |
| Task 3 |  | | Identify system features and  functionalities | 04/06/2024 | 04/12/2024 | 1 |
| PHASE | 3: | |  |  |  |  |
| Analysis Design | and | |  |  |  |  |
| Task 4 |  | | Define  technologies/framework | 04/13/2024 | 04/19/3024 | 1 |
| Task 5 |  | | Create diagrams | 04/20/2024 | 05/03/2024 | 2 |
| Task 6 |  | | Create wireframes | 05/10/2024 | 05/16/2024 | 1 |
| Task 7 |  | | Design user interface | 05/17/2024 | 05/24/2024 | 1 |
| PHASE  Implementa | 4:  tion | |  |  |  |  |
| Task 8 | | | Set up development environment | 05/25/2024 | 06/01/2024 | 1 |
| Task 9 | | | Implement user registration and  authentication | 06/21/2024 | 06/16/2024 | 2 |
| Task 10 | | | Database development for Patient Records Management | 06/17/2024 | 06/06/2024 | 3 |
| Task 11 | | | Develop an integrated module for real-time medication reminders to patients | 06/07/2024 | 06/27/2024 | 3 |
| Task 12 | | | Develop an interface that automatically generates needed reports. | 06/28/2024 | 07/18/2024 | 3 |
| PHASE 5:  Testing | | | | |  |  |
| Task 13 | |  | Test the System | 07/19/2024 | 08/01/2024 | 2 |
| Task 14 | |  | Train the user on the usage of the system | 08/02/2024 | 08/15/2024 | 2 |
| Task 15 | |  | Prepare the system for deployment | 08/16/2024 | 08/29/2024 | 2 |
| PHASE  Evaluation | | 6: | |  |  |  |
| Task 16 | |  | Evaluate the system using ISO 25010  standards | 08/30/2024 | 08/12/2024 | 2 |
| PHASE  Deployment | | 7: | |  |  |  |
| Task 17 | |  | Deploy the system | 08/15/2024 | 09/15/2024 | 4 |

Critical Path Analysis



4.7 Responsibilities

This section provides the lists of the names, roles, and responsibilities. However, although roles are specified per member, they were also in control of supervising the different modules in the system.

# Table 4.5 Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Position | Responsibilities |
| Jonel B. Prado | Capstone Adviser | The responsibilities of the capstone adviser is to track the progress of the students, serve as a guiding path in creating the capstone projects in terms of reviewing the papers and providing feedback and comments on the work |
| Jayric G. Espadero | Programmer | Responsible for designing, coding, and maintaining the software that runs the system. |
| Christian Jade Paras | System Analyst | Responsible for analyzing the requirements, design system architecture,  create system  specifications, and |
|  |  | collaborate with the developer to ensure the system meets the desired functionality. |
| Shajara Mae A. Ronao | Technical Writer | Responsible for preparing, reviewing, revising, and maintaining technical papers and documentation  of this project. |

4.8 Budget and Cost Management

This project needs a budget for acquiring materials and services needed to create the documentation and the whole system. The approximate budget and cost allocation for this project and its operational cost when deployed were indicated below:

4.8.1 Proponent’s Budget and Costs Management

This section shows the materials needed for printing the documentation of the study. Printed copies of the manuscript were needed for proposal and final defense and the safekeeping of the deliverables.

# Table 4.6. Materials and Supplies Budget

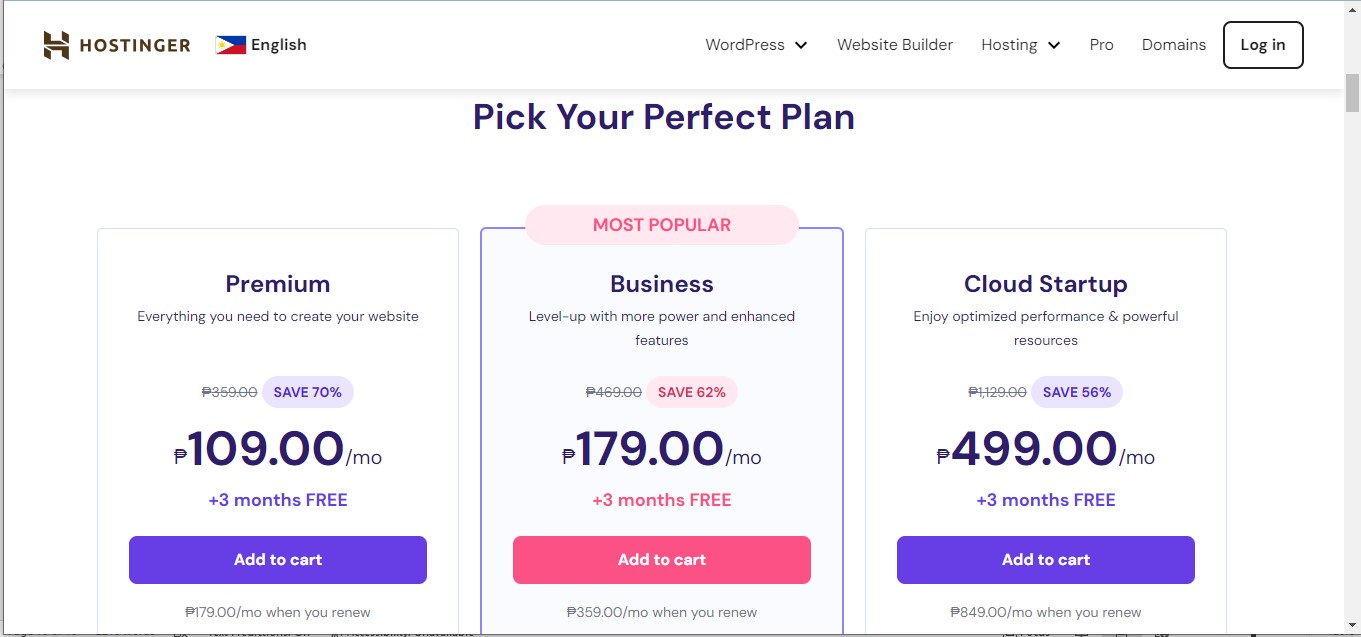
|  |  |  |  |
| --- | --- | --- | --- |
| Item Name | Quantity | Unit Price | Amount |
| Bond Paper (Hard  Copy, 70gsm, 500  sheets, letter) | 2 | Php.200.00 | Php.400.00 |
| Ink (Epson L3110  Ink 003) | 1 set (4 pieces) | Php.1,000.00 | Php.1,000.00 |
|  |  | Total Estimated  Cost: | Php.1, 400.00 |

# Table 4.7. Services and Hosting Budget

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Provider | Monthly Fee | Amount |
| Internet Service | PLDT | Php.1,699.00 | Php.11, 893.00 |
| Web Hosting | Hostinger | Php.179.00 | Php.1, 432.00 |
|  |  | Total Estimated Php.13,325.00  Cost | |

4.8.2 System’s Operation Cost in Development

The system’s operation cost in development covers expenses for creating and maintaining the proposed system. This section presents the minimum budget of the operational cost of the system when deployed using the following service: Hostinger for the system’s web hosting service.



Date taken: April 29, 2024

Hostinger Services’ costs for its cloud hosting start from Php.109.00 to Php.499.00 monthly plan as of writing of this study. The proponents are going to subscribe for the Business Plan, and it is Php.179.00 per month, and renewing at Php.359.00 per month after the first year of subscription. Additionally, Business plan builds upon the Premium plan by offering daily backups, Hostinger CDN to speed up the site, and 200GB of NVMe storage.

# Table 4.8 Service Operation Costs

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Provider | Monthly Fee | Amount |
| Web Hosting | Hostinger | Php.179.00 | Php.1,253.00 |
|  |  | Total Estimated Cost: Php.1,253.00 | |

4.9 Verification and Validation

4.9.1 Verification

Since the project’s development method involves iterative process, the system is verified and check by the client per module over time. This will include walkthroughs and any other forms of client engagement to ensure alignment with project goals and requirements.

During the review phase of the system, the proponents’ adviser and panelists verify the system if it meets the proposed requirements and is necessary for the

clients.

4.9.2 Validation

To check the system’s actual effectiveness and if it meets the requirements and expectations of the client, the system will undergo validation.

The 5-point Likert scale that was employed as a metric for the attributes in the system’s validation is shown in Table 4.12. It displays the rating together with its verbal interpretation and mean range.

|  |  |  |
| --- | --- | --- |
| Rating | Mean Range | Description |
| 1 | 1.00-1.80 | Strongly Disagree |
| 2 | 1.81-2.60 | Disagree |
| 3 | 2.61-3.30 | Neither Agree or  Disagree |
| 4 | 3.31-4.20 | Agree |
| 5 | 4.21-5.00 | Strongly Agree |

# Table 4.9 Likert Scale for System Validation

|  |  |
| --- | --- |
| Sub-characteristics | Description |
| Completeness | The set of functions covers all the specified tasks and user objectives. |
| Correctness | The system provides the correct results with the needed degree of precision |
| Appropriateness | The functions facilitate the accomplishment of specified tasks and objectives. |

Table 4.10 Functional Suitability Characteristic

|  |  |
| --- | --- |
| Sub-characteristics | Description |
| Co-existence | The system can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product. |
| Interoperability | Two or more systems, products or components can exchange information and use the information that has been exchanged |

# Table 4.11 Compatibility Characteristic

|  |  |
| --- | --- |
| Sub-characteristics | Description |
| Appropriateness | Recognizability Users can recognize whether a product or system is appropriate for their needs. |
| Learnability | The system can be used by specified users to achieve specified goals of learning to use the system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use. |
| Operability | The system has attributes that make it easy to operate and control. |
| User error protection | The system protects users against making errors. |
| User interface aesthetics | The user interface enables pleasing and satisfying interaction for the user |
| Accessibility | The system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use. |

# Table 4.12 Usability Characteristic

4.10 Testing

Testing was conducted to ensure the system operated smoothly without encountering any bugs or errors, particularly in its input acceptance and overall processes. The aim was to verify that the system adhered to the proposed functional requirements. The functional requirements were assessed through black-box testing, utilizing the cause-effect testing technique. These tests were manually conducted by the project team. During alpha testing, both black-box and white-box testing methods were employed by the project advisor and panelists to evaluate the system. Black-box testing focused on assessing the system's functionality and its ability to process inputs accurately, encompassing positive and negative test scenarios. Positive scenarios involved valid inputs, while negative ones involved invalid inputs. Test outcomes were based on the system's capability to handle both types of inputs appropriately. Cause-effect testing involved identifying desired outcomes and creating corresponding test cases to achieve those outcomes. White-box testing, on the other hand, scrutinized the system's code and internal structure to ensure the correctness of its logic and algorithms.

|  |  |
| --- | --- |
| System | Web-Based Health Records Management System with  Medication Alert Notification or Immaculate Medico-  Surgical Clinic Bulan Sorsogon |
| Test Module | Login > Sign Up > Add Patient > View Patient List |
| Access Requirements | URL: https://IMSClinic\_sytem.com  Login Credentials:   * Username * Password * N/A |
| Goal of the Test | The goal of the test is to validate the functionality of the web-based Health Records Management System across various user types, including Administrative Staff, Doctors, and Patients. Specifically, the test aims to ensure the following: |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | •  •  •  • | | Login Functionality | : Validate the ability of users efficiency.  : Validate the process of    Functionality: Validate the  functionality of adding new patients to the system  .  Functionality: Validate the  functionality of viewing the list of patients in the | |  |
| to log in to the system Signup Functionality registering new patients efficiently. Add Patient  by Administrative Staff View Patient List  system, particularly for doctors. |
| Out of Scope | The functionality of the patient is not included in the testing. | | | | | | |
| Additional  Requirements |  | | • The login process should be secure and efficient. | | | |  |
| • The signup process should collect necessary | | | |
| information securely and efficiently. | | | |
| • The process of adding a patient should capture | | | |
| essential patient information accurately. | | | |
| • | The patient list should be displayed accurately. | |  |
|  | |
| Testing Steps |  | 1. Open the web browser and navigate to the   Healthcare Management System login page.   1. Enter the provided test username and password into the respective fields. 2. Click on the "Login" button. | | | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1. Verify that the system successfully logs in the user. 2. Ensure that appropriate error messages are   displayed for incorrect login credentials.   1. Test the "Forgot Password" functionality (if available). 2. Verify that the user is redirected to the correct dashboard or landing page upon successful login. 3. Navigate to the signup page (if available). 4. Fill in the required information on the signup form. 5. Click on the "Sign Up" button. 6. Verify that the system successfully creates a new user account. 7. Ensure that appropriate error messages are displayed for invalid or incomplete signup information. 8. Navigate to the "Add Patient" section. 9. Fill in the required patient information on the form. 10. Click on the "Save" or "Add" button. 11. Verify that the new patient is successfully added to the system. | |  |
|  |  | 1. Ensure that appropriate error messages are displayed for incomplete or invalid patient information. 2. Once logged in as a doctor, navigate to the patient list section. 3. Verify that the list of patients is displayed correctly. 4. Ensure that appropriate filters and search functionalities are working as expected. | |  |
| Device |  | | |  | | --- | | Target Device |   • : Desktop or Laptop |  |
| • Software Required: Web browser (Chrome,  Firefox, Safari) |