**Clinic Inventory System and Dental Appointment System for Mindoro State University Calapan Campus**

An

Application Development Project

Presented to the Faculty of

**Mindoro State University Calapan City Campus**

Masipit, Calapan City

Oriental Mindoro

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

by:

**Mayordo, John Paul**

**Cuasay, Charlyn Iriejo G.**

**De Lara, Geralyn F.**

**Leynes, Rose Ericka**

October 7, 2024

Table of Contents

Title page ………………………………………………………………………………………………………………… 1

Table of Contents………………………………………………………………………………………………… 2

**CHAPTER**

1. INTRODUCTION

Project Context

Objectives

Scope and Limitations

Definitions of Terms

1. REQUIREMENTS SPECIFICATION

Hardware and Software Requirements

Functional Requirements

Non-Functional Requirements

Operational Requirement

Performance Requirement

Security Requirement

1. DESIGN AND DEVELOPMENT METHODOLOGIES

System Design

Database Design

Architectural Diagram/Block Diagram

DFD Level 0

UML Use-Case Diagram

Sample Mock Up

Methodology

SDLC Phases

Testing

1. DEVELOPMENT, TESTING AND EVALUATION RESULT

System Output

Testing Results

1. CONCLUSIONS AND RECOMMENDATIONS

Conclusion

Recommendation

REFERENCES

APPENDICES

**CHAPTER I**

**INTRODUCTION**

In this chapter, the researchers will discuss the overview of the project, including its project context, objectives and scope and limitations.

**Project Context**

Healthcare services are essential for maintaining the health and well-being of students, with university clinics often serving as their primary source of medical care(Al-khafajiy et al., 2019). Aside from that, patients’ satisfaction is also important to know if there is a need of change(Manzoor et al., 2019). At Mindoro State University, the clinic offers essential services such as medical consultations, first aid, and student health monitoring. However, the clinic faces challenges due to outdated management systems, including inefficient manual processes for inventory tracking, difficulties in scheduling appointments, limited communication between healthcare providers and students, and a lack of tools for anticipating health-related issues. Manual process of health service frustrate not just the patients but also the nurses and doctors due to time consuming(Tian et al., 2019). The manual processes in healthcare services frustrate not only patients but also nurses and doctors, as they are time-consuming (Tian et al., 2019). These challenges can lead to delays, mismanagement of resources, and dissatisfaction among students who rely on the clinic's services (Gonzales & Rivera, 2019).

To address these challenges, the development of a University Clinic Inventory System and Dental Appointment System is proposed. These systems will introduce several features to improve the clinic's operations. The dental appointment will allow students to schedule and reschedule appointments online, making the process more convenient and reducing waiting times. The system will also log dental visits and list available services, including tooth filling (pasta), extraction, and cleaning. To ensure better communication, dentists can send messages to their patients, and students will be able to reply through the system. Monthly reports will also be generated to summarize patient visits, treatments, and overall clinic activity.

The inventory management will allow admins to add, delete, and monitor supplies and medicines, ensuring efficient inventory tracking. A log history will maintain records of transactions for transparency and accountability. The system will also produce reports on stock levels, usage, and replenishment needs, helping the clinic maintain an adequate supply of resources. Admin will also be notified if stocks need to be restocked and as well as the notifications for every audit logs.

This system will integrate dental appointment scheduling and inventory to improve the operations of the Mindoro State University Clinic.

**Objectives**

The researchers aim to develop an inventory system and dental appointment system to address the inefficiencies in the current operations of the Mindoro State University clinic. By integrating smart technologies, the study seeks to improve healthcare service and enhance student satisfaction.

Specifically, the objective of this study aims to:

1. Implement secure user authentication and access control for both students and healthcare personnel to ensure data privacy and security.
2. Establish an efficient inventory system that allows real-time tracking of medical supplies, ensuring the clinic remains fully stocked and prepared for student healthcare needs.
3. Introduce an appointment system that enables students to schedule appointment online, reducing waiting times and improving convenience.
4. Integrate a visit log feature in the dental appointment system to simplify and automate the check-in process, ensuring accurate and efficient recording of clinic visits.
5. Enhance communication between healthcare providers and students by integrating a messaging system like Gmail that facilitates uninterrupted information exchange.
6. Provide the capability to generate reports and data analytics in Inventory and Dental Appointment system.

**Scope and Limitations**

The Clinic Inventory System and Dental Appointment System for Mindoro State University Calapan Campus aims to improve clinic operations and enhance healthcare access. The system is designed to ease the exhausting and time-consuming manual tasks for nurses and doctors. Key features of the Inventory System include inventory tracking, a stock expiration tracker using batch names, PDF report generation (both yearly and monthly, with print options), and the ability to import Excel files for adding supplies. The Dental Appointment System includes Google Calendar integration for scheduling, a Gmail-based messaging system, PDF report generation (yearly and monthly, with print options), and user-friendly interfaces for both students and dentists. The goal is to make an efficient appointment scheduling and manage resources.

The Dental Appointment System allows students to schedule appointments online for services such as tooth fillings, extractions, and cleaning, with real-time updates. The inventory management feature offers real-time tracking of medical supplies, low-stock alerts, and product expiration monitoring. Additionally, the inventory system is standalone, with only an admin interface, while the Dental Appointment System includes both a Patient Portal and an Admin Dashboard for efficient appointment scheduling, health data analysis, and reporting.

However, there are some limitations. The Dental Appointment System is deployed using Firebase, making it online based, while the Inventory System is deployed locally. As a result, the Dental Appointment System requires a stable internet connection for better performance, which means the clinic, must ensure reliable internet access. The admin can’t place an order for new supplies. The system focuses primarily on clinic-related tasks such as inventory management and dental appointment and does not cover broader health monitoring.

**Definition of Terms**

* **API** **(Application Programming Interface)** - This includes allowing admin to perform inventory and dental appointment actions meaning it is a request to backend.
* **UI** **(User Interface)** - Refers to the design and layout of the system’s web and mobile interfaces however mobile interfaces is for dental appointment only. We have two UI, UI for inventory which is standalone system and Dental Appointment which admin and students has separated interfaces. Student Interface includes their registration and appointment whereas admin interface includes all the needed features for the system.
* **Restock Level**: This is the minimum amount of stock you need to have before it's time to reorder. When your inventory drops to this level, it’s a signal to restock so you don’t run out.
* **Disbursement**: This is the act of giving out or distributing goods or resources. For example, it’s when you release products from storage to be used, sold, or delivered to customers.
* **Clinic Staff**: Personnel responsible for managing the operations and services of the university clinic, including inventory, patient care, and scheduling.
* **Student**: A registered individual at the university who can access the clinic’s healthcare services, including appointments and medical support.

**CHAPTER II**

**REQUIREMENTS SPECIFICATION**

In this chapter, the researchers will provide the hardware and software requirements, as well as functional and non-functional requirements necessary for the project development.

**Software Requirements**

Software Specifications refer to the representation of the software used by the system. Table 1 below presents the software specifications to be used by the project.

|  |  |
| --- | --- |
| Software Used | Description |
| GitHub | GitHub to enable collaboration and project management. |
| Google Calendar | Google Calendar to enable scheduling, event management, and reminders for personal and professional use. |
| Node.js | Enable server-side JavaScript execution and build scalable network applications. |
| Node Mailer | Enable to send emails easily from Node.js applications with support for various transport methods. |
| Vue.js | Vue.js v3.2.45 with Vue CLI for a simplified frontend development with intuitive API, fast rendering, and robust build setup. |
| Mysql | MySQL v8.0.32 for a high-performance relational database management system. |
| RESTful API | Enable communication between client and server using standard HTTP methods for resource manipulation. |
| JWT (JSON Web Tokens) | Enable secure transmission of information between parties as a JSON object, often used for authentication and information exchange. |
| Firebase | It is used for the development of Dental Appointment System |
| Sequelize | Sequelize is an Object-Relational Mapping (ORM) library for Node.js, simplifying interaction with relational databases like MySQL. It lets developers use JavaScript objects and methods to manage data, eliminating the need for raw SQL queries. |
| Tailwind | It enables developers to style elements quickly without writing custom CSS, promoting rapid development and consistency. |
| Google Auth | Used to integrate login functionality into web and mobile applications. |

**Table 1: Software Requirements**

**Functional Requirements**

This part enumerates the operations and activities that the system must perform. Table 2 represents the descriptions of data requirements, process requirements, and output requirements.

|  |  |
| --- | --- |
| **ID NO** | **Requirement Description** |
| **FR1** | **Data Requirements** |
| **General** | |
| 1.1 | The system must provide secure user authentication, real-time data updates, and efficient operation for both inventory management and dental appointment scheduling, with features such as notifications, data validation, and reporting. It should also be scalable, ensure strong security, and support integrations to maintain smooth performance and user satisfaction. |
| **Inventory Management** | |
| 1.2 | Items and Batch Item Addition   * The admin logs into the system. * The admin selects the "Add Items in Batch" option or upload a CSV or Excel file with details such as item name, quantity, and expiry date. * The system validates the uploaded data for consistency (e.g., ensuring no negative quantities and valid expiry dates). * The inventory list is updated, and the admin receives a confirmation message. * The admin can add new supplies/items or remove expired stock as necessary. * The admin can also disburse stock and specify the recipient. |
| 1.3 | Low Stock and Expiry Notifications   * The system monitors inventory levels and generates automatic notifications when stock falls below a predefined threshold or when an item's expiry date is approaching. * Notifications about low-stock items and those nearing expiry are displayed on the dashboard and in the header bell icon for easy access. |
| 1.4 | Transaction and Audit Logs   * All inventory transactions (e.g., adding new stock, updating quantities, removing expired items) are automatically logged. * The log captures the date/time of action and type of transaction. * Admin can review the log for transparency and accountability since all nurses are possible to use the system. * Log entries are protected from alteration or deletion to maintain data integrity. |
| 1.5 | Monthly and Yearly Report Generation   * At the end of each month and year, admin can generate reports summarizing inventory usage, including details such as total usage per item, stock-outs, expiry-related wastage, and restocking patterns. * These reports can be exported in PDF format, ready for printing. |
| 1.6 | Inventory Dashboard   * The Inventory Dashboard provides admin with key metrics.   + Current stock levels.   + Low-stock items and those near expiry.   + Recent inventory transactions. |
| **Dental Appointment Scheduling** | |
| 1.7 | Student Portal for Booking Appointments   * Students will log in to the Dental Appointment Portal and select a dental service and a time slot from the available options. * The system will prevent double bookings by validating the appointment time and notifying students of any schedule changes in real-time. * The student can respond to the messages from the dentist |
| 1.8 | Appointment Management   * Dentists can manage appointments by rescheduling or canceling them as necessary. * Dentists can message patients through the system using the Gmail integration. * The system can generate PDF reports for both monthly and yearly summaries. * Appointment changes will trigger automatic notifications to students, informing them of any modifications. |

**Table 2: Functional Requirements**

**Non-Functional Requirements**

These are requirements that pertain to behavior properties that a system must have. It defines how a system is supposed to be or its system properties. It contains the following:

**Operational Requirement**

Table 3 represents the requirement description that will specify the operating environment(s) in which the system must perform and how these might change over time.

|  |  |
| --- | --- |
| ID No. | **Requirement Description** |
| 1.1 | The system must maintain an uptime of 99.9%, ensuring availability during critical times, especially for inventory system and dental appointment system. |
| 1.2 | Data must be automatically backed up every 24 hours and stored securely for at least30 days to protect against data loss. |
| 1.3 | The Patient Portal and Admin Dashboard must have a simple, intuitive interface to ensure easy navigation for both clinic staff and students. |
| 1.4 | The system should be built with a modular architecture, allowing easy updates and integration of new features, such as additional health services. |
| 1.5 | The system will seamlessly integrate with third-party services, such as GoogleCalendar for appointment scheduling and reminders. |
| 1.6 | The system must handle API changes without disrupting the core functionality. |

**Table 3: Operational Requirement**

**Performance Requirement**

Table 4 represents the requirement description that will emphasize the response time, capacity, and reliability of the system.

|  |  |
| --- | --- |
| ID No. | **Requirement Description** |
| 1.7 | Data updates, such as appointment bookings and inventory changes, must sync across all connected devices within 30 seconds to ensure a smooth user experience. |
| 1.8 | The dental appointment system must handle up to 10,000 concurrent users without significant performance degradation. |
| 1.9 | The system must remain stable and responsive during peak periods which is high-demand appointment scheduling. |

**Table 4: Performance Requirement**

**Security Requirement**

Table 5 represents the requirement description that will address issues with security, such as who has access to the system's data and must have the ability to protect data from disruption or data loss.

|  |  |
| --- | --- |
| ID No. | **Requirement Description** |
| 1.10 | Password must be encrypted using bcrypt library function, hash during authentication and while stored in the database. |
| 2.0 | JWT is used for secure authentication and authorization by generating, validating, and storing tokens containing encoded user information, enabling stateless access to protected resources. |
| 2.1 | The inventory system includes a voucher code feature to enhance security. Access to the inventory system requires a unique voucher code, which is generated and sent exclusively by the system. This ensures that only authorized users can access the inventory. |

**Table 5: Security Requirements**

**Chapter III**

**Design and Development Methodologies**

**System Design**

The Clinic Inventory System and Dental Appointment System for Mindoro State University Calapan Campus is an integrated platform designed to enhance healthcare services and improve operational efficiency by managing key aspects like inventory tracking and dental appointment. This system provides an improved approach to healthcare management, ensuring that resources are efficiently allocated, and appointments are easily scheduled.

The frontend of the system is developed using Vue.js, a progressive JavaScript framework that allows for the creation of dynamic, user-friendly interfaces. Vue.js ensures that the system’s user interface is responsive and interactive, making it easy for users to manage inventory and schedule or manage dental appointments. With Vue.js, the frontend smoothly integrates with the backend.

The backend of the system is built using Node.js and Express.js, which provide a better foundation for handling server-side logic and data management. Node.js ensures that the system can handle numerous simultaneous requests efficiently, while Express.js Express.js handles the core RESTful API, which facilitates structured data communication between the client interface and the database. The API manages tasks such as appointment scheduling and inventory control.

The backend architecture also integrates middleware for authentication and authorization, using JWT (JSON Web Tokens) for secure session management. This ensures that only authorized users, such as clinic staff and students, can access sensitive medical data and system features.

The MySQL relational database serves as the central repository designed using a normalized schema to maintain data integrity and consistency.

**Database Design**

The database design for the Clinic Inventory System and Dental Appointment System at Mindoro State University Calapan Campus is developed to support efficient healthcare operations. Using MySQL as the backbone, it organizes data for inventory tracking, dental appointment management, user accounts, and system logs. Each table is tailored to a specific function, such as monitoring inventory levels, scheduling appointments, or managing user credentials. This structured and secure design ensures smooth and reliable performance while remaining flexible for future enhancements, forming a strong foundation for effective healthcare management.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Admin ID (PK) |
| email | varchar(255) | NOT NULL | Admin email for login |
| password | varchar(255) | DEFAULT NULL | Admin account password |
| voucher | varchar(255) | DEFAULT NULL | Optional code for admin purposes |
| createdAt | datetime | NOT NULL | Timestamp when the account was created |
| updatedAt | datetime | NOT NULL | Timestamp when the account was last updated |

**Table 6. Fields for Admins**

**Table 6**: The Admins table contains the fields for managing administrator accounts, including id, email, password, voucher, createdAt, and updatedAt. It has a primary key on id.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Batch ID (PK) |
| inventory\_item\_id | int | DEFAULT NULL | Links to the Inventory Items table |
| batch\_number | varchar(100) | NOT NULL | |  | | --- | |  |  |  | | --- | | Batch identifier | |
| quantity | int | DEFAULT ‘0’ | Quantity of items in the batch |
| expiry\_date | datetime | DEFAULT NULL | Expiration date of the batch |
| supplier | varchar(255) | DEFAULT NULL | Supplier of the batch |
| received\_date | datetime | DEFAULT NULL | Date the batch was received |
| is\_active | tinyint(1) | DEFAULT ‘1’ | Indicates if the batch is active (1) or not (0) |

**Table 7. Fields for Batches**

**Table 7**: The Batches table contains fields for tracking inventory batches, including id, inventory\_item\_id, batch\_number, quantity, expiry\_date, supplier, received\_date, and is\_active. It has a primary key on id and a foreign key on inventory\_item\_id, linking to the Inventory Items table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Category ID (PK) |
| name | varchar(255) | NOT NULL | Name of the category |
| description | text |  | Description of the category |

**Table 8. Fields for Categories**

**Table 8**: The Categories table contains fields for categorizing inventory items, including id, name, and description. It has a primary key on id.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Inventory Item ID (PK) |
| name | varchar(255) | NOT NULL | Name of the inventory item |
| description | text |  | Description of the inventory item |
| category\_id | int | NOT NULL | Links to the Categories table |
| quantity\_in\_stock | int | DEFAULT ‘0’ | Current stock of the item |
| min\_stock\_level | int | DEFAULT ‘0’ | Minimum stock level before restocking |
| unit\_price | decimal(10,2) | DEFAULT ‘0.00’ | |  | | --- | |  |  |  | | --- | | Price per unit of the item | |
| reorder\_level | int | DEFAULT ‘0’ | Stock level to trigger reorder |
| is\_active | tinyint(1) | DEFAULT ‘1’ | Indicates if the item is active (1) or not |

**Table 9. Fields for Inventory Items**

**Table 9**: The Inventory Items table contains fields for managing inventory details, including id, name, description, category\_id, quantity\_in\_stock, min\_stock\_level, unit\_price, reorder\_level, and is\_active. It has a primary key on id and a foreign key on category\_id, linking to the Categories table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Notification ID (PK) |
| notification\_type | enum ('LOW\_STOCK','EXPIRED','SOON\_EXPIRING','REORDER') | NOT NULL | Type of notification (e.g., LOW\_STOCK) |
| batch\_id | int | DEFAULT NULL | Links to the Batches table |
| inventory\_item\_id | int | DEFAULT NULL | Links to the Inventory Items table |
| quantity\_left | int | DEFAULT NULL | Remaining stock triggering the alert |
| expiry\_date | datetime | DEFAULT NULL | Expiry date of the item triggering alert |
| title | varchar(255) | NOT NULL | Short title of the notification |
| message | text | NOT NULL | Detailed notification message |
| seen | tinyint(1) | DEFAULT ‘0’ | Indicates if the notification was read |
| created\_at | datetime | DEFAULT NULL | Timestamp when the notification was created |

**Table 10. Fields for Notifications**

**Table 10**: The Notifications table contains fields for logging system alerts, including id, notification\_type, batch\_id, inventory\_item\_id, quantity\_left, expiry\_date, title, message, seen, and created\_at. It has a primary key on id and foreign keys on batch\_id and inventory\_item\_id, linking to the Batches and Inventory Items tables, respectively.

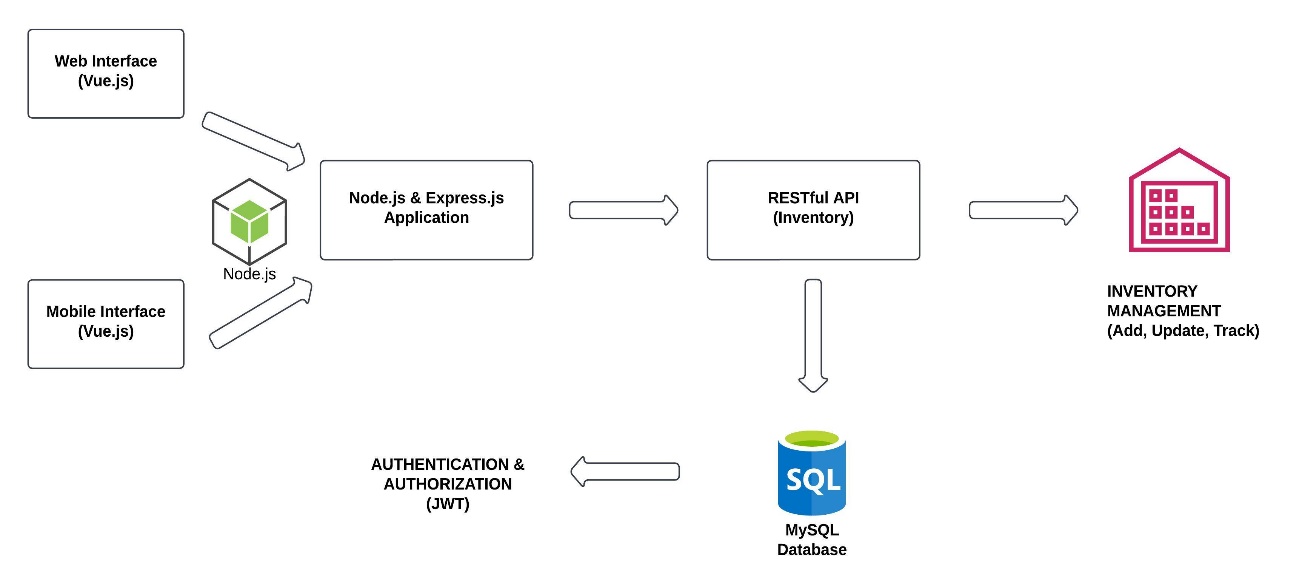
|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Default** | **Description** |
| id | int | NOT NULL | Transaction ID (PK) |
| inventory\_item\_id | int | DEFAULT NULL | Links to the Inventory Items table |
| batch\_id | int | DEFAULT NULL | Links to the Batches table |
| transaction\_type | enum ('ADD','REMOVE','UPDATE','DISPOSE') NOT NULL, | NOT NULL | Type of transaction (e.g., ADD) |
| quantity\_change | int | NOT NULL | Quantity added or removed |
| date | datetime | DEFAULT NULL | Date of the transaction |
| remarks | text |  | Notes or comments about the action |
| patient\_name | varchar(255) | DEFAULT NULL | Patient name associated with the transaction |

**Table 11. Fields for Transactions**

**Table 11**: The Transactions table contains fields for recording inventory activities, including id, inventory\_item\_id, batch\_id, transaction\_type, quantity\_change, date, remarks, and patient\_name. It has a primary key on id and foreign keys on inventory\_item\_id and batch\_id, linking to the Inventory Items and Batches tables, respectively.

**Architectural Diagram/ Block Diagram**

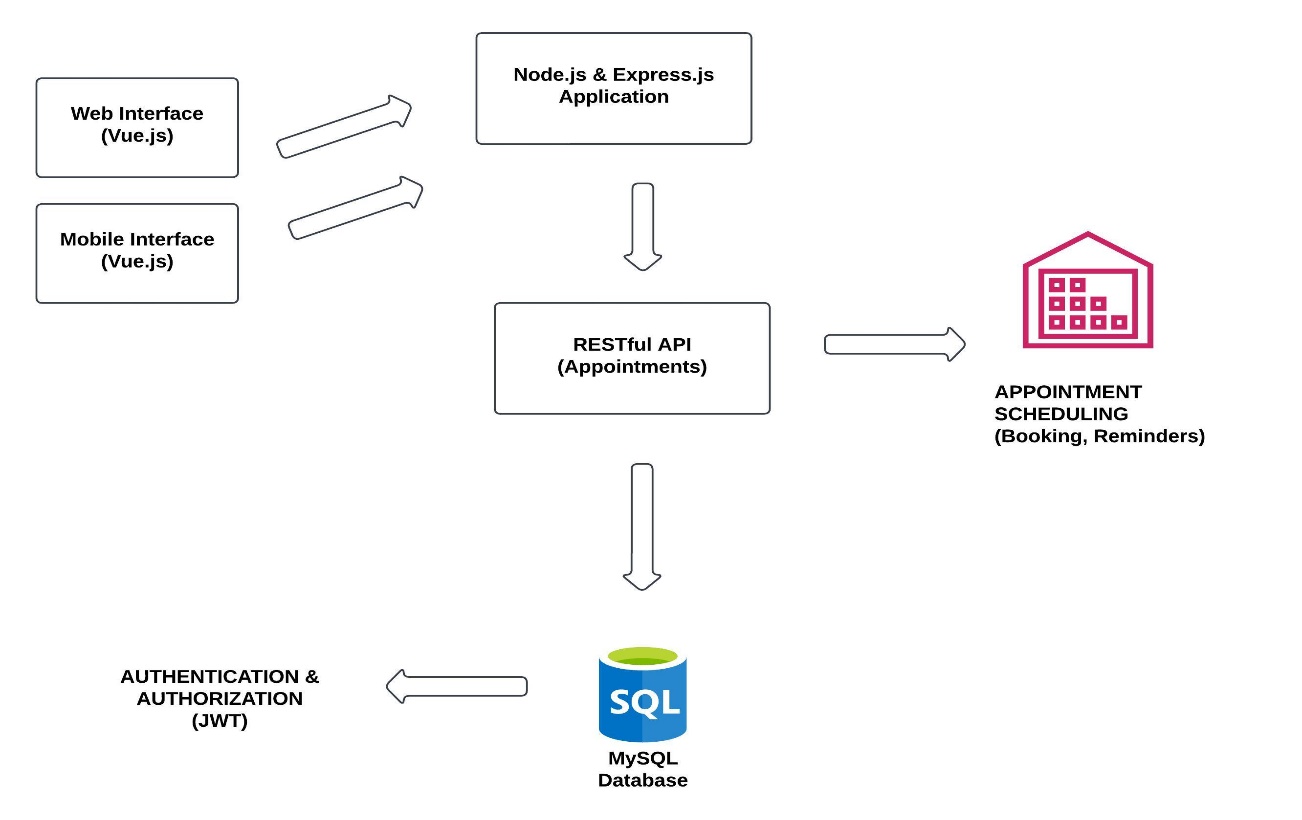
In this section, system architecture was designed to define the flow and behavior of the system’s functionalities to execute its high-quality performance. This covers the formal illustration and description of the project structure.



**Figure 1.0 System Architecture of**

**Clinic Inventory System**

This Figure allows users to manage inventory through both web and mobile interfaces, such as adding new items, adjusting stock levels, and tracking existing products. When a user sends a request through the interface, the backend, which is powered by Node.js and Express.js, processes the application logic. The system interfaces with the inventory management services using a RESTful API, ensuring that data flows smoothly between the user interface and the database. The inventory management component enables users to conveniently manage product data and stock levels. Security is provided by JSON Web Tokens (JWT) for user authentication, which ensures that only authorized personnel can access and edit the inventory. All inventory data, including stock details, is maintained in a MySQL database, which serves as a centralized repository for data consistency and integrity while also making it available when processing requests.

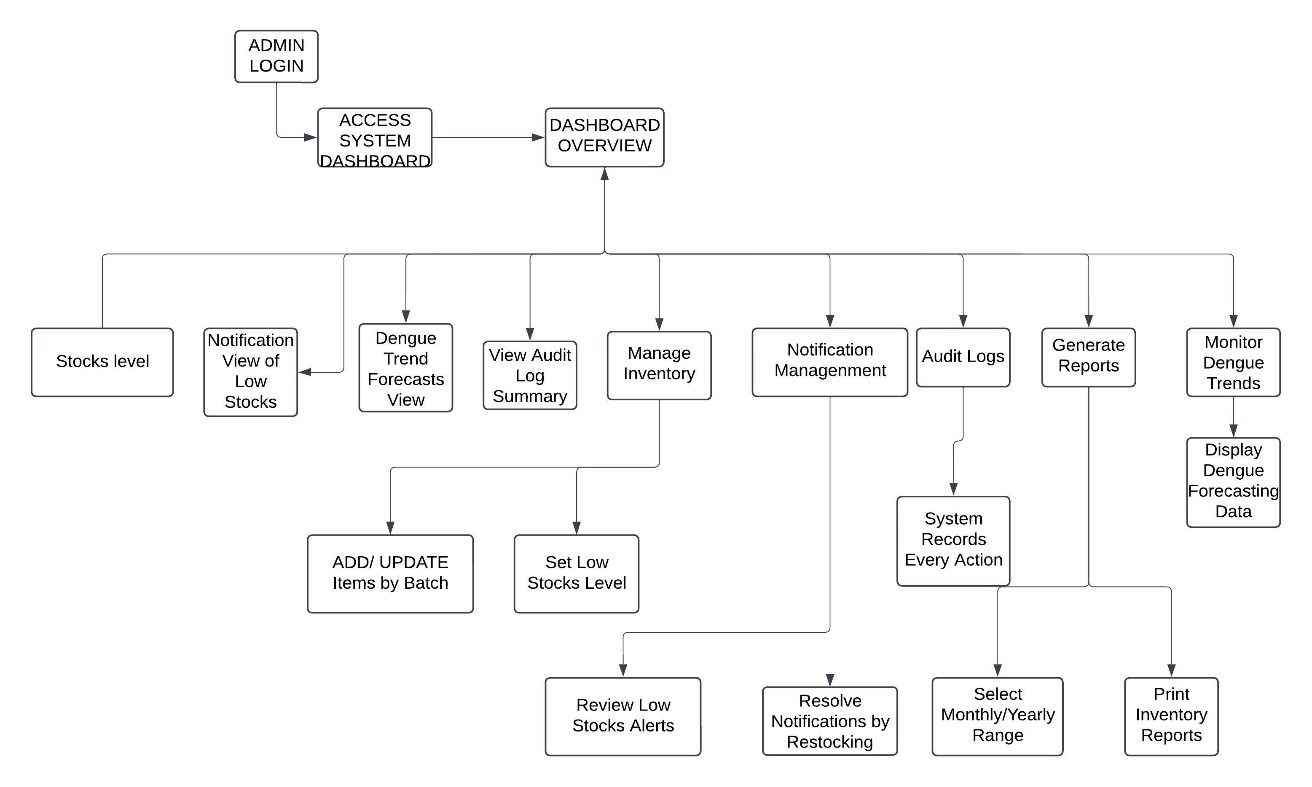


**Figure 2.1 System Architecture of Dental Appointment System**

This Figure assists with dental appointments where Users can access the web or mobile interface, which is developed with Vue.js to ensure a responsive and user-friendly experience. When a user makes an appointment, the request is routed to the Node.js and Express.js application, which processes the data and communicates with the RESTful API to handle duties such as booking, reminders, and updates. The system maintains all appointment information in a MySQL database, which ensures data security and accessibility. To ensure system security, JWT (JSON Web Tokens) is utilized for authentication and authorization, allowing only authorized users to access the system. The Notification Service (NodeMailer) sends email reminders to users about upcoming appointments to assist them remember. This system helps manage appointments in an easy and secure way.

**DFD Level 0**

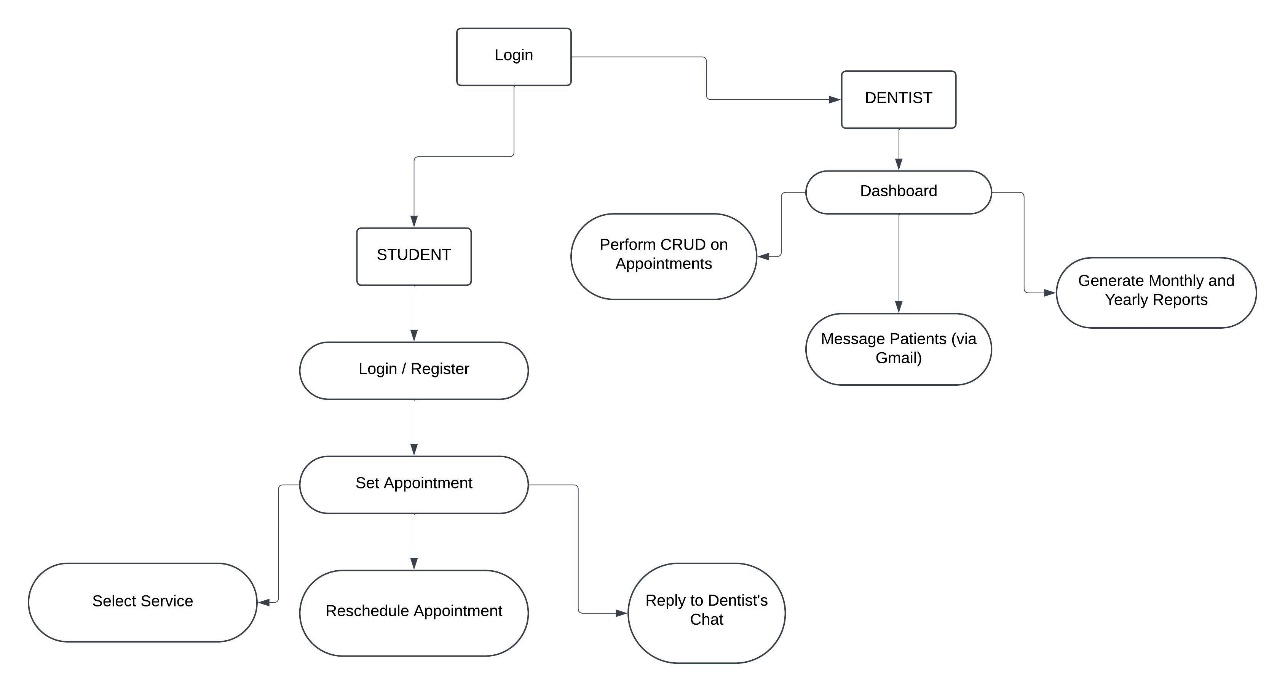
The Level 0 Data Flow Diagram is an enhanced representation of the context diagram that provides a thorough analysis of the project's operating processes. It serves as a roadmap, visually showing the complicated connections and data flows throughout basic activities, allowing for a thorough knowledge of the system's performance.



**Figure 2.0 Data Flow Diagram Level 0 of**

**Clinic Inventory System**

Figure 2 outlines the Clinic Inventory System workflow, with a focus on the administrator's responsibility in ensuring efficient stock management. The Admin Dashboard is the central location where administrators may examine important information such as current stock levels, low-stock notifications, and audit logs. This helps the administrator to keep track of inventory status and take appropriate actions. Administrators can update goods in batches, set low-stock criteria, and receive alerts when stock levels fall below a certain level. These warnings allow the administrator to take prompt action, such as restocking, to avoid shortages. The system automatically monitors every action in the Audit Logs to keep a clear record of all inventory changes, such as updates and notifications. Administrators can also create thorough reports to track inventory utilization over time, providing insights into trends and assisting with future planning. This system ensures that the administrator may effectively manage inventory by automating critical operations, providing user-friendly dashboards, and enabling quick responses to low-stock situations.



**Figure 2.1 Data Flow Diagram Level 0 of**

**Dental Appointment System**

This dental appointment system flowchart outlines the workflow of two key user roles: students (patients) and dentists. Students begin by logging into the system or creating an account. After logging in, clients can schedule an appointment by selecting the required dental service, such as a check-up or cleaning. If their plans change, students can easily reschedule their appointments. Furthermore, the technology lets students contact with their dentist via a chat tool, allowing them to ask questions or clarify specifics about their meetings. The system gives a full dashboard for dentists to manage their tasks. Dentists can use CRUD (Create, Read, Update, and Delete) activities on appointments to ensure effective scheduling and modifications. They can also use Gmail to communicate with patients, sending them reminders or vital appointment updates. Furthermore, the system allows dentists to generate extensive monthly and yearly reports, which aids in trend analysis, workload management, and productivity monitoring. The entire system is intended to allow easy interactions between students and dentists, ensuring effective appointment administration, clear communication, and an organized workflow for both sides.

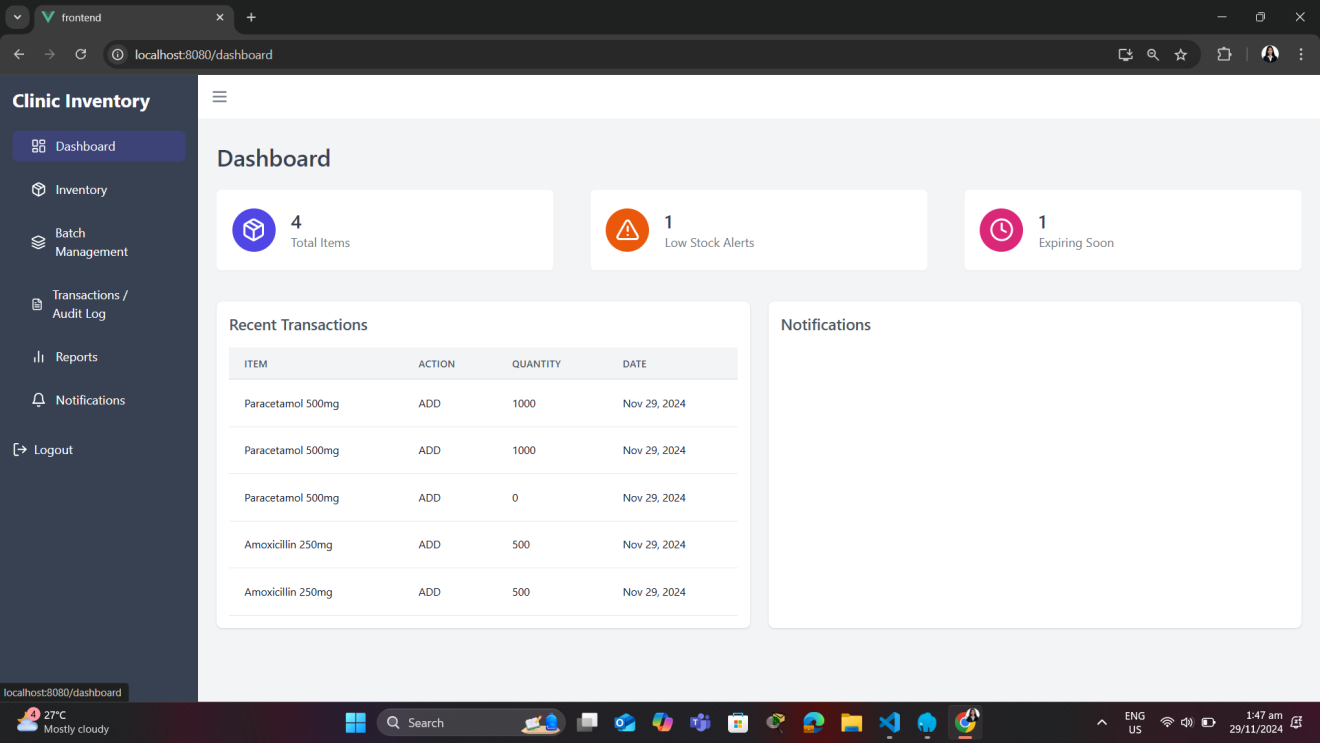
**UML Use-case Diagram**

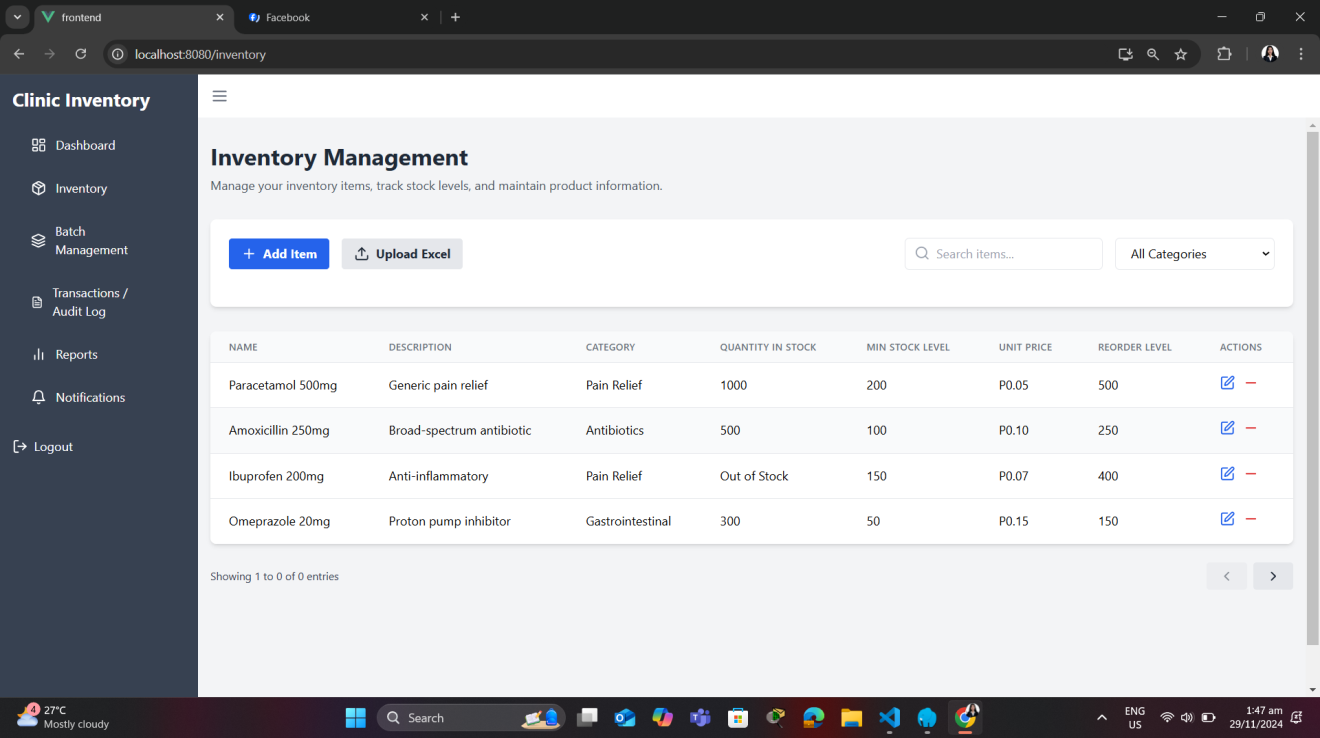
ADD HERE THE UML

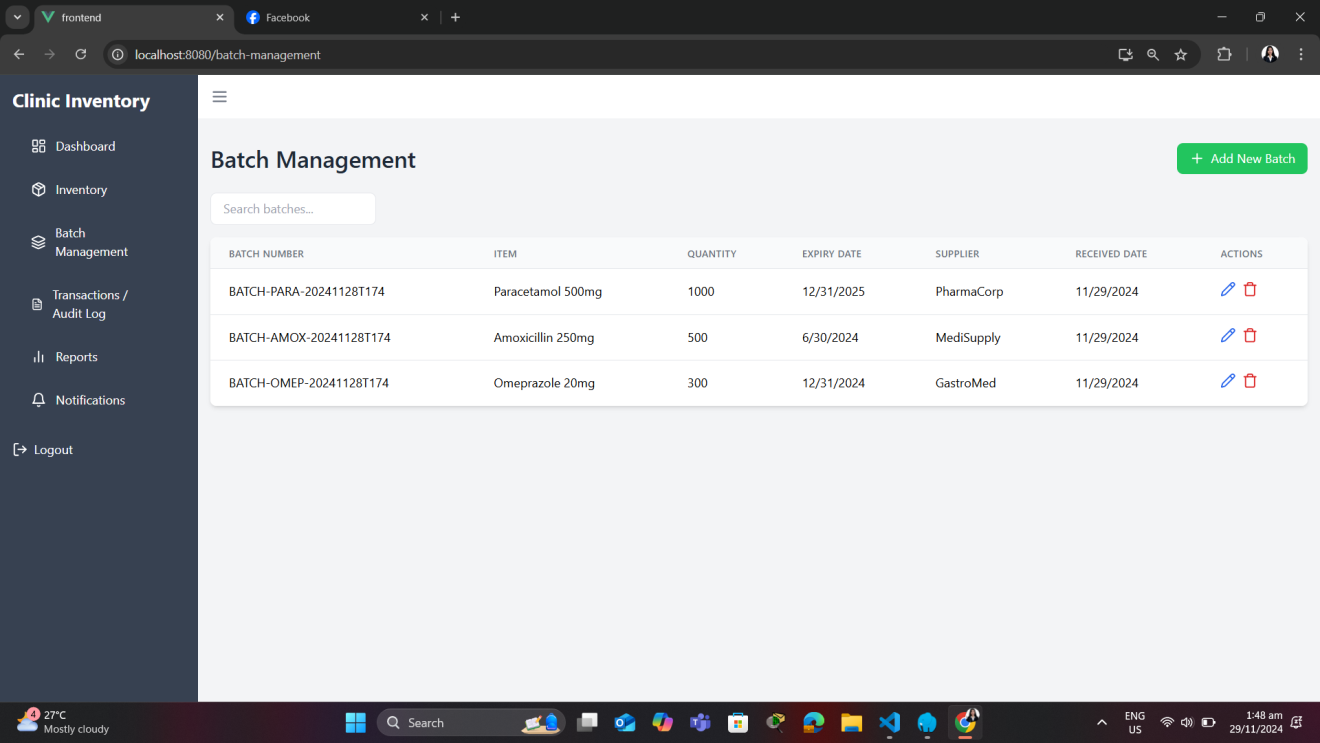
**Figure 3. UML Use-case Diagram**

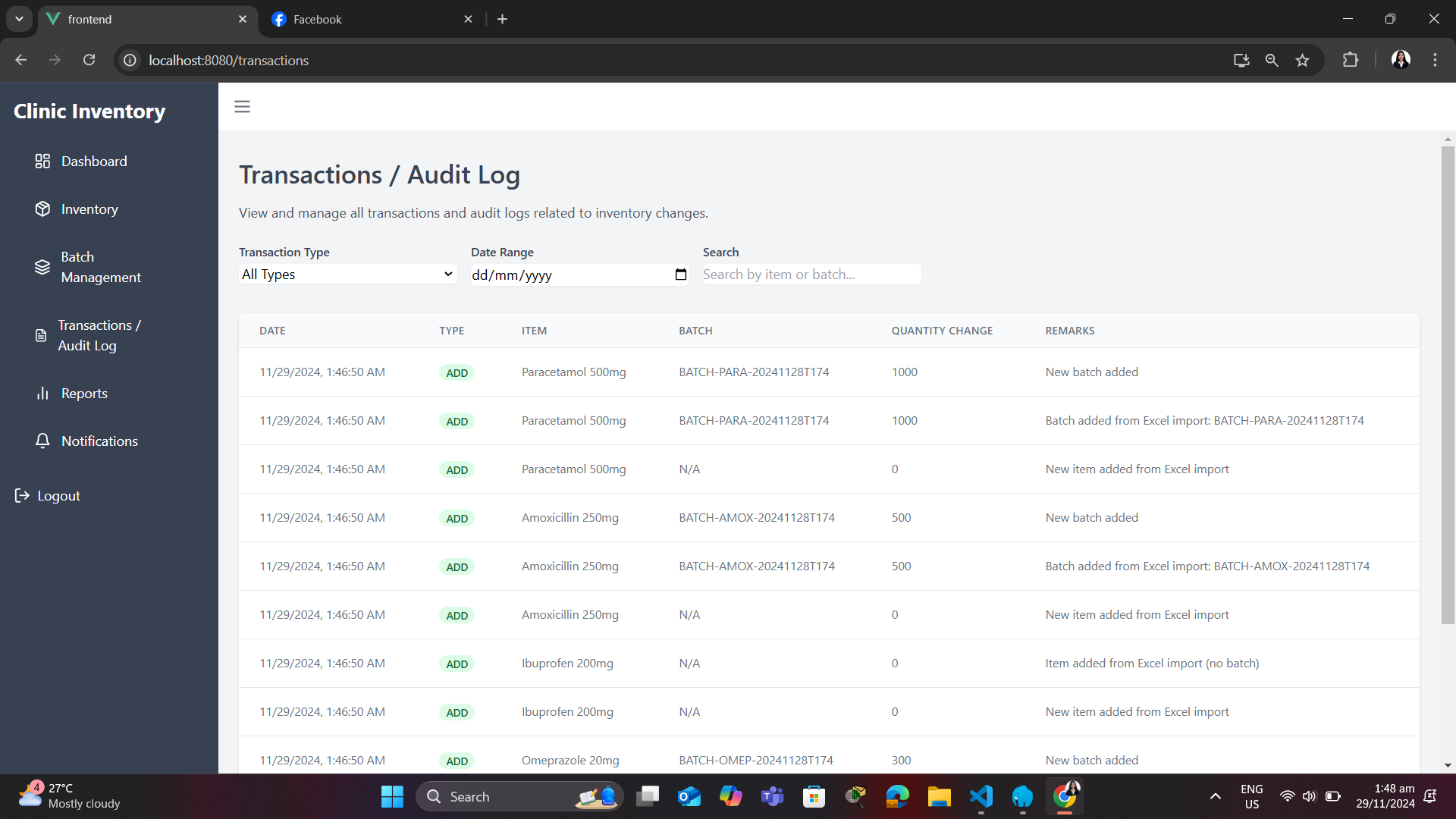
**Sample Mock-up**

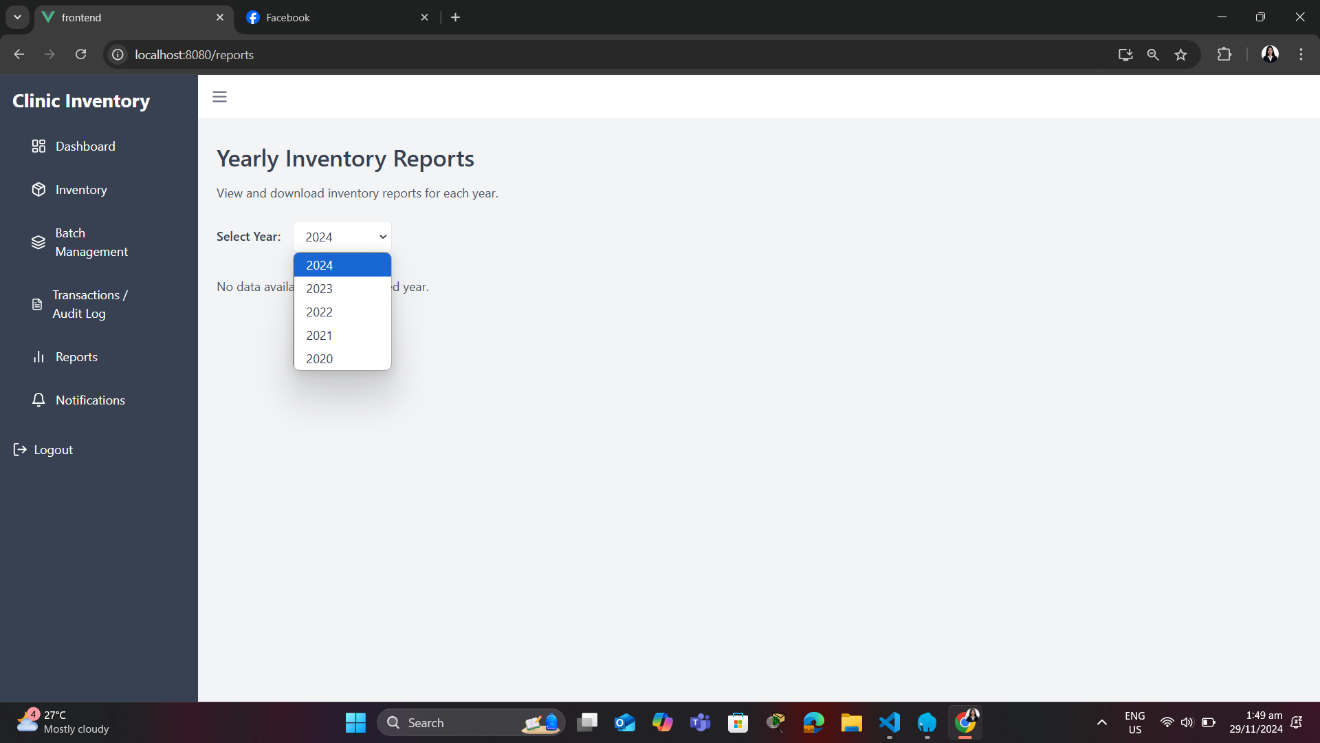
Sample mock-up is the visual representation of the website, that showing the functions and layout. It helps in considering the overall user experience, ensuring that the design meets user expectations and usability standards.

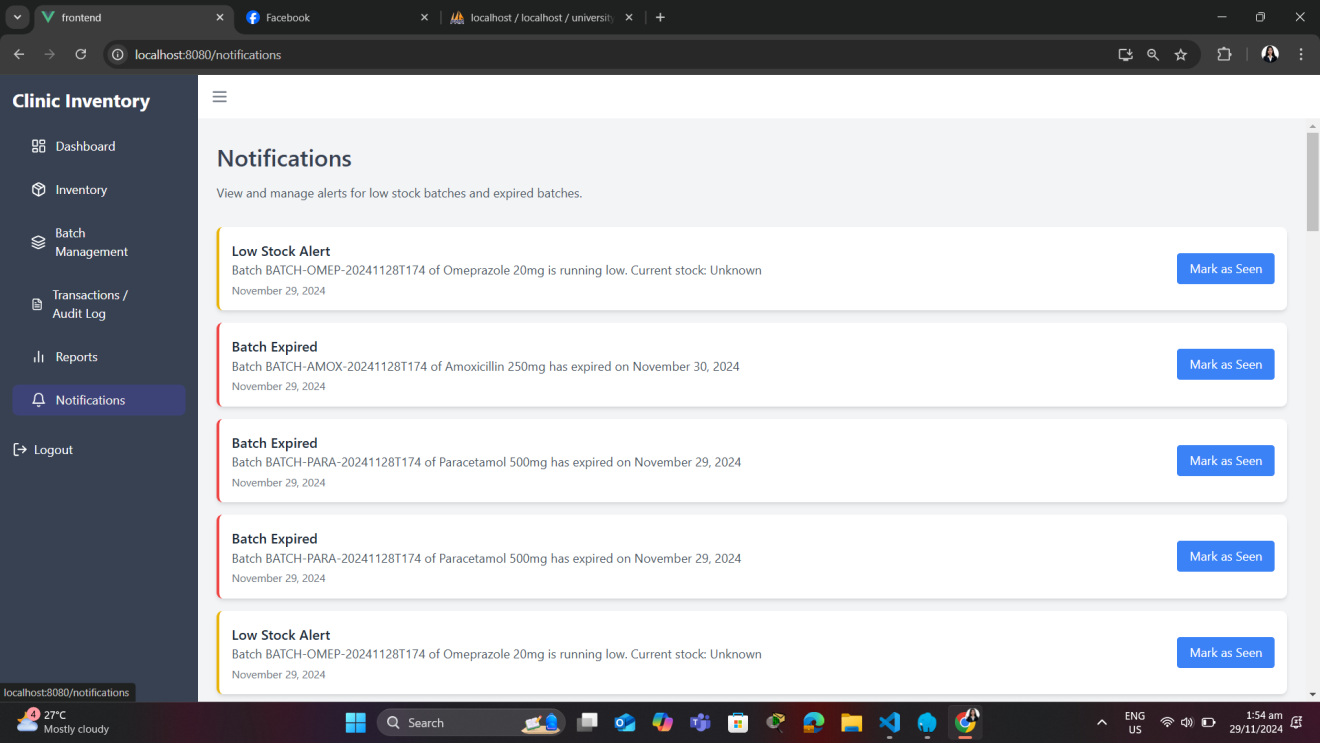
 **Figure 3. Admin Interface**









****

**Development Methodology**

In developing the Clinic Inventory System and Dental Appointment System for Mindoro State University Calapan Campus, a Rapid Application Development (RAD) methodology was employed to ensure a quick and adaptive approach to address the evolving needs of Mindoro State University’s healthcare services. RAD emphasizes user involvement and the iterative refinement of system prototypes, allowing the system to adapt to feedback from clinic staff and students during the development process. This section covers the phases of Planning, Requirements Gathering, Design, Development, Testing, and Feedback.

**Figure 6. SDLC Rapid Application Development Model**

The development process followed the following phases:

1. Planning.  
   In this phase, initial discussions were conducted with Mindoro State University’s clinic personnel to identify the goals and objectives for the project. The main focus was on understanding the key challenges faced by the clinic, such as manual inventory management, long waiting times, and inefficient appointment scheduling.
2. Requirements Gathering.  
   During this phase, researchers consulted with clinic staff to define the system's requirements. The main components identified were the digitalization of inventory management and the dental appointment system, aimed at addressing the clinic's challenges with time-consuming and labor-intensive manual processes.
3. Design.

In this phase, researchers developed system prototypes and created a blueprint for the database using MySQL, incorporating the necessary relationships between inventory management and dental appointment. Interface designs were created to ensure user-friendliness for both students and clinic personnel.

1. Development.  
   During the development phase, the system was built incrementally with the tentative data of clinic in inventory. The process is trial and error due to the sudden change of the system, but we address it as soon as possible. A combination of Vue.js and Tailwind for the frontend and Nodejs for the backend ensured a responsive and secure system. Modules for authentication, dental appointment scheduling, and inventory management were developed.
2. Testing.

This phase involves testing of the system, including unit tests for each individual module and integration tests to verify the smooth operation between different system components. User acceptance testing (UAT) was performed with clinic staff and our AppDev professor to ensure the system met functional and performance requirements. The system’s ability to handle multiple concurrent users and deliver real-time updates was tested.

1. Feedback.

Following the testing phase, feedback from clinic staff and our AppDev professor was gathered. Researchers conducted review sessions to assess whether the system met the clinic’s needs and where further improvements could be made. This feedback was used to make adjustments and fine-tune the system, ensuring that it was aligned with the clinic’s operational requirements and user expectations.

**Gantt Chart**

In this section, Gantt Chart is the presentation of project schedule. This shows when to start and when will be the ending date of the project. This helps the researchers to prioritize the task to meet the deadlines, and ensuring the project stays on schedule.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backend and Frontend** | **Task Date** | | | |  | | | | |  | | | | |
| Oct | | | | Nov | | | | | Dec | | | | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | | Week 2 | Week 3 | Week 4 |
| **1.Planning** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  | |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  | |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| **6.Implementation** |  |  |  |  |  |  |  |  |  | |  |  |  |
| **7.Maintenance** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.1 Project monitoring |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.2 Resolve system errors |  |  |  |  |  |  |  |  |  | |  |  |  |

**Table 11. Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Gatherer and Technical Writer** | **Task Date** | | | |  | | | | |  | | | | |
| Oct | | | | Nov | | | | | Dec | | | | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | | Week 2 | Week 3 | Week 4 |
| **1.Planning** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  | |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  | |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| **6.Implementation** |  |  |  |  |  |  |  |  |  | |  |  |  |
| **7.Maintenance** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.1 Project monitoring |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.2 Resolve system errors |  |  |  |  |  |  |  |  |  | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Technical Writer** | **Task Date** | | | |  | | | | |  | | | | |
| Oct | | | | Nov | | | | | Dec | | | | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | | Week 2 | Week 3 | Week 4 |
| **1.Planning** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  | |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  | |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| **6.Implementation** |  |  |  |  |  |  |  |  |  | |  |  |  |
| **7.Maintenance** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.1 Project monitoring |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.2 Resolve system errors |  |  |  |  |  |  |  |  |  | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Technical Writer** | **Task Date** | | | |  | | | | |  | | | | |
| Oct | | | | Nov | | | | | Dec | | | | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | | Week 2 | Week 3 | Week 4 |
| **1.Planning** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  | |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  | |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  | |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  | |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  | |  |  |  |
| **6.Implementation** |  |  |  |  |  |  |  |  |  | |  |  |  |
| **7.Maintenance** |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.1 Project monitoring |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7.2 Resolve system errors |  |  |  |  |  |  |  |  |  | |  |  |  |

**Legend:** - Mayordo

* Charlyn
* De Lara
* Leynes

Table 11 provides a comprehensive timeline of the system development process. Each team member is represented by a designated color, illustrating their step-by-step progress throughout the project. This ensures to understand the project progress and avoid missing stages.

**CHAPTER VI**

CONCLUSIONS AND RECOMMENDATIONS

**REFERENCES**

1. Gonzales, A., & Rivera, M. (2019). Challenges of traditional healthcare management in university settings: A case study of Mindoro State University. Journal of University Health Services, 34(1), 45-56. <https://doi.org/10.1234/juhs.2019.45>
2. Al-khafajiy, M., Kolivand, H., Baker, T., Tully, D., & Waraich, A. (2019). Smart hospital emergency system: Via mobile-based requesting services. Multimedia Tools and Applications, 78(14), 20087–20111. <https://doi.org/10.1007/s11042-019-7274-4>
3. Manzoor, F., Wei, L., Hussain, A., Asif, M., & Shah, S. (2019). Patient satisfaction with health care services; an application of physician’s behavior as a moderator. International Journal of Environmental Research and Public Health [revista en Internet] 2019 [acceso 20 de octubre de 2022]; 16(18): 1-16. Int J Environ Res Public Health, 1–16. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6765938/>
4. Tian, S., Yang, W., Grange, J. M. Le, Wang, P., Huang, W., & Ye, Z. (2019). Smart healthcare: making medical care more intelligent. Global Health Journal, 3(3), 62–65. <https://doi.org/10.1016/j.glohj.2019.07.001>

APPENDICES

**Curriculum Vitae**

**ROSE ERICKA M. LEYNES**

**Suqui, Calapan City**

**Mobile No: 09611977206**

**Email address:** **[roseerickaleynes8@hmail.com](mailto:roseerickaleynes8@hmail.com)**

**PERSONAL INFORMATION**

**Date of Birth June 14, 2004**

**Age 20 years old**

**Sex Female**

**Civil Status Single**

**Citizenship Filipino**

**Language Spoken Filipino and English**

**Father Ricardo Leynes**

**Mother Remedios Leynes**

**EDUCATION**

**Tertiary Bachelor of Science in Information Technology**

**Mindoro State University**

**Masipit, Calapan City Oriental Mindoro**

**Secondary HUMSS**

**Oriental Mindoro National High School**

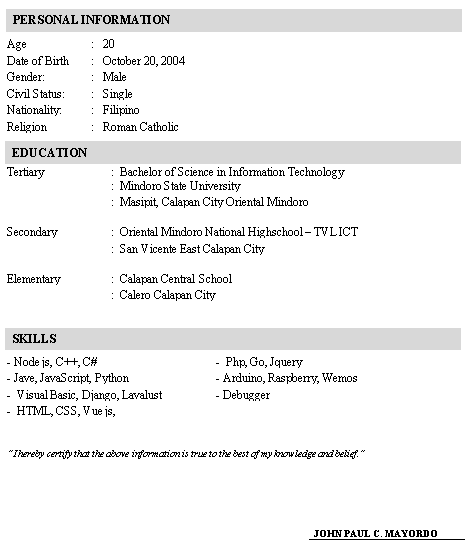
**San Vicente East, Calapan City**

**Elementary Boac South Central School**

**Laylay, Boac Marinduque**

**MAYORDO, JOHN PAUL C.**

Ibaba West, Calapan City  
Oriental Mindoro, Philippines  
Mobile No.: 09670449774  
Email Address: [danmchi@gmail.com](mailto:danmchi@gmail.com)

****

**CUASAY, CHARLYN IRIEJO G.**

Sto. Niño, Calapan City  
Oriental Mindoro, Philippines  
Mobile No.: 0916-746-5332  
Email Address: [charlyncuasay94@gmail.com](mailto:charlyncuasay94@gmail.com)

**A close-up of a paper

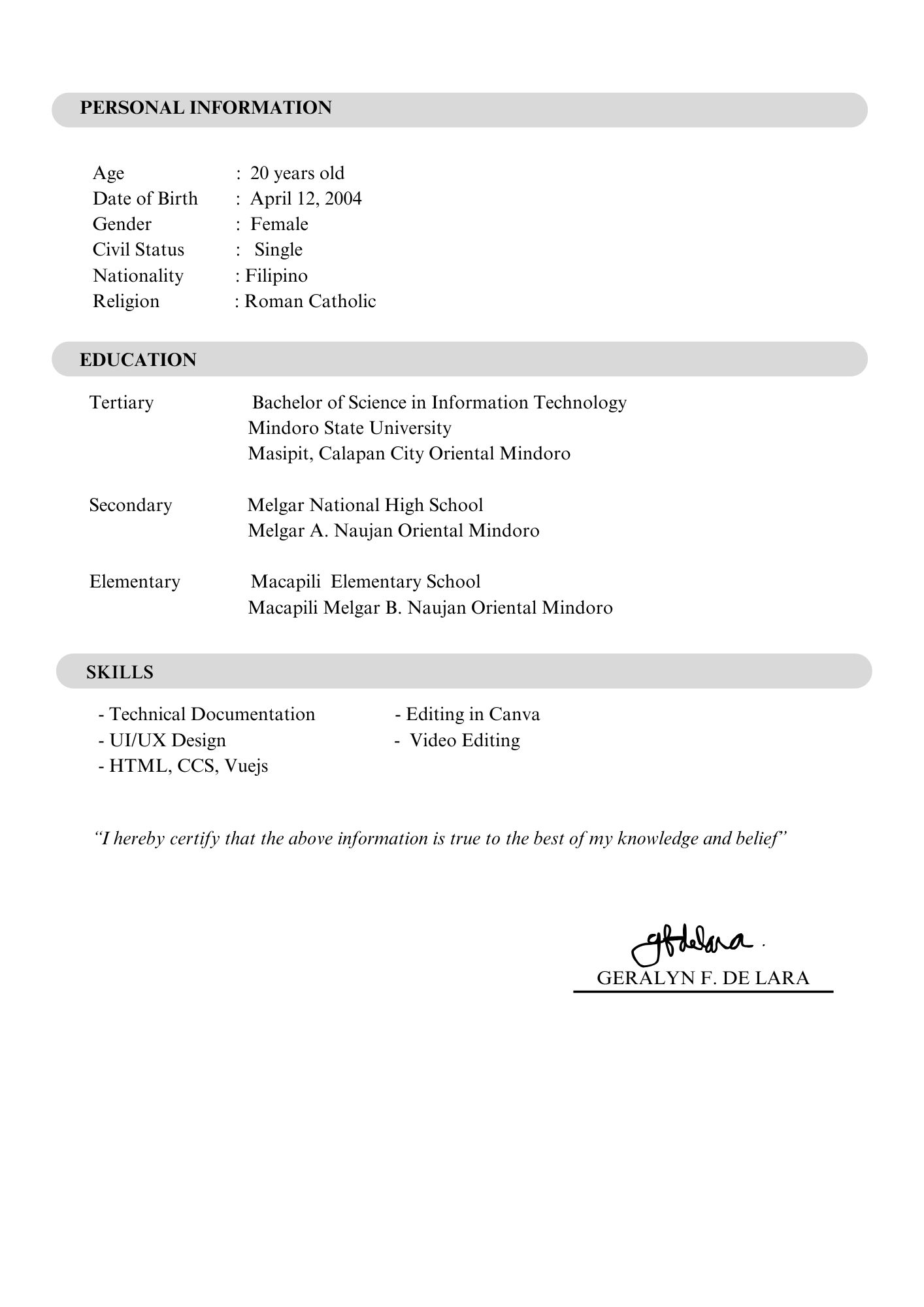
Description automatically generatedA person in a suit

Description automatically generated**

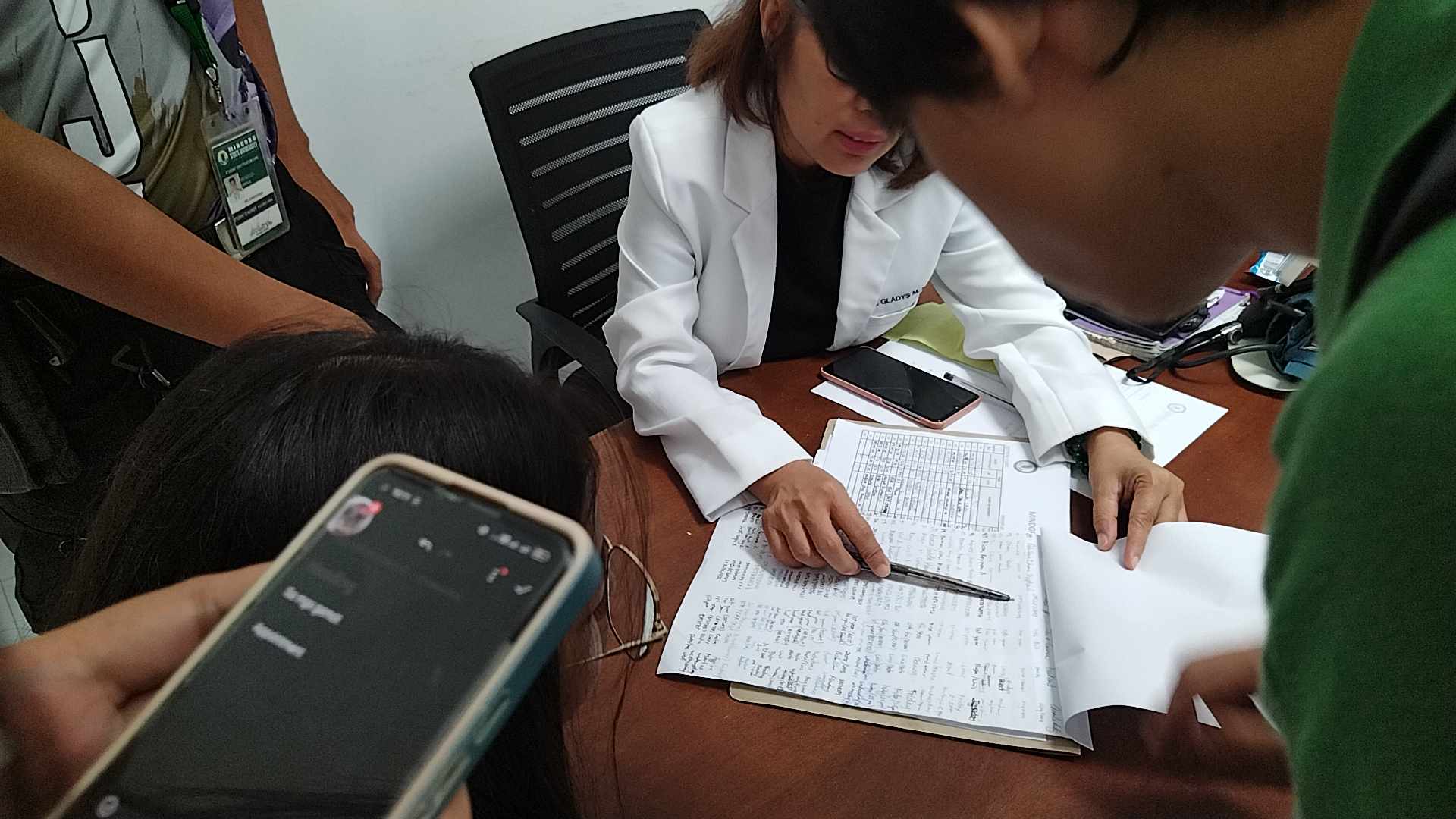
****

**DE LARA, GERALYN F.**

Melgar B, Naujan  
Oriental Mindoro, Philippines  
Mobile No.: 09935902883  
Email Address: [delarageralyn12@gmail.com](mailto:delarageralyn12@gmail.com)

****

**Picture During Development, Testing & Evaluation**



A group of people sitting at a desk

Description automatically generated

