**A PROPOSED OFFERING OF A CLINIC RECORDS MANAGEMENT SYSTEM**

**FOR**

**HI-PRECISION DIAGNOSTICS – MALABON BRANCH**

A Thesis Project Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Catubay, Mark Lawrence L.

**REQUIREMENT SPECIFICATION**

**CHAPTER I**

**INTRODUCTION**

**Background of the Study**

The modern healthcare landscape is increasingly reliant on the timely and accurate management of information. For diagnostic facilities like Hi-Precision Diagnostics, which operate at the critical intersection of patient care and medical data, the efficiency of their internal processes directly impacts service quality. The Malabon branch, like many others, processes a high volume of patients daily, generating a substantial amount of sensitive data that includes personal demographics, medical histories, consultation notes, diagnoses, and treatments. The method used to store, retrieve, and manage this information is fundamental to the clinic's operational success and its ability to provide exceptional patient care.

Traditionally, many clinics have relied on paper-based filing systems or rudimentary digital tools such as spreadsheets. While functional to a degree, these methods are fraught with inherent limitations. Paper records are susceptible to physical damage, loss, and unauthorized access, posing significant risks to data integrity and patient confidentiality as mandated by the Data Privacy Act of 2012 (RA 10173). Furthermore, manual processes are labor-intensive and prone to human error, leading to operational inefficiencies such as prolonged patient wait times for file retrieval, difficulty in tracking consultation histories, and inaccurate inventory management. The inability to quickly search, aggregate, or analyze patient data also hinders the clinic's capacity for informed decision-making and quality control.

To address these pressing challenges, there is a clear and urgent need for a specialized digital solution. This project proposes the development of a Clinic Records Management System (CRMS), a dedicated, standalone desktop application designed specifically for the operational context of Hi-Precision Diagnostics – Malabon Branch. The CRMS aims to replace outdated record-keeping methods with a centralized, secure, and intuitive digital platform. By automating core tasks such as patient registration, consultation logging, and inventory tracking, the system will empower clinic staff to perform their duties more effectively, reduce administrative overhead, and ultimately enhance the quality and speed of patient service.

**Purpose of this Document**

This Requirement Specification Document (RSD) serves as the formal foundation for the Clinic Records Management System project. Its primary purpose is to provide a detailed and unambiguous description of all the system's requirements, establishing a clear agreement between the project proponent and stakeholders on the expected functionalities and characteristics of the final product.

This document is the definitive source of truth for what the system *will do*. It will guide all subsequent phases of the project lifecycle, including system design, software development, quality assurance testing, and final user acceptance. By clearly defining the project's boundaries and deliverables, this RSD aims to prevent scope creep, minimize misunderstandings, and provide a concrete basis for evaluating the success of the completed system.

**Scope of the Project**

The scope of the CRMS is to deliver a comprehensive solution for managing the core operational data within the Hi-Precision Diagnostics – Malabon Branch. The system's capabilities are focused on enhancing the day-to-day efficiency of the clinic's staff.

**The following functions are considered IN-SCOPE for this project:**

* User Authentication: Secure login and logout capabilities for authorized clinic staff.
* Patient Information Management: The ability to create, read, update, and delete (CRUD) patient demographic records.
* Consultation History Management: A module for adding new consultation notes to a patient's record and viewing their complete chronological history of visits.
* Medicine Inventory Control: Basic functionality to manage a list of clinic medicines, including adding new items and updating stock quantities.
* Patient Search: A real-time search feature to quickly locate patient records.
* Simple Reporting: The ability to generate a printable summary of a single patient's profile and consultation history.

**The following functions are considered OUT-OF-SCOPE for this project:**

* Billing, invoicing, and financial management.
* Integration with external laboratory equipment or third-party information systems.
* Online appointment scheduling or a web-based patient portal.
* Advanced business analytics or multi-clinic data synchronization.
* Human Resources or staff payroll management.

**Intended Audience**

This document is intended for the following audiences:

* Project Stakeholders (Clinic Management/Staff) - To review and confirm that the specified requirements accurately reflect their operational needs.
* Project Proponent/Developer - To use as the primary guide for designing and building the software.
* Faculty and Project Advisors - To evaluate the project's completeness, feasibility, and adherence to academic standards.

**CHAPTER II**

**FUNCTIONAL REQUIREMENTS**

This chapter defines the specific functions and behaviors the system must perform; what the system does.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority | Dependencies | Acceptance Criteria |
| User & Session Management |  |  |  |  |
| FR-001 | User Authentication | High | - | The system shall present a login screen. An authorized user must be able to log in with a valid username and password. The system must deny access for invalid credentials. |
| FR-002 | User Logout | High | FR-001 | A logged-in user must be able to log out of the system, which terminates their session and returns them to the login screen. |
| Patient Management |  |  |  |  |
| FR-003 | Create New Patient | High | FR-001 | An authorized user must be able to add a new patient record by providing their full name, date of birth, and address. The system shall assign a unique ID upon successful creation. |
| FR-004 | View All Patients | High | FR-001 | An authorized user must be able to view a list of all patients in the system, displaying at least their name and date of birth in a tabular format. |
| FR-005 | Update Patient Information | Medium | FR-004 | An authorized user must be able to select a patient from the list and modify their name, date of birth, or address. |
| FR-006 | Delete Patient | High | FR-004 | An authorized user must be able to select and delete a patient record from the system. The system must prompt for confirmation before deletion. Deleting a patient must also delete all associated consultation records. |
| FR-007 | Search for Patient | High | FR-004 | An authorized user must be able to search for patients by name. The patient list must dynamically filter to show only matching results as the user types. |
| Consultation Management |  |  |  |  |
| FR-008 | Add Consultation Record | High | FR-004 | When viewing a specific patient, an authorized user must be able to add a new consultation record, which includes the consultation date, complaint, diagnosis, and treatment. |
| FR-009 | View Consultation History | High | FR-004 | When viewing a specific patient, an authorized user must be able to see a chronological list of all past consultations for that patient. |
| Medicine Inventory Management |  |  |  |  |
| FR-010 | Create New Medicine | High | FR-001 | An authorized user must be able to add a new medicine to the inventory by providing its name, quantity, and an optional description. |
| FR-011 | View All Medicines | High | FR-001 | An authorized user must be able to view a list of all medicines in the inventory, displaying their name, quantity, and description. |
| FR-012 | Update Medicine Information | Medium | FR-011 | An authorized user must be able to select a medicine and update its quantity and/or description. |
| FR-013 | Delete Medicine | Medium | FR-011 | An authorized user must be able to select and delete a medicine from the inventory. The system must prompt for confirmation before deletion. |
| FR-014 | Search for Medicine | High | FR-011 | An authorized user must be able to search for medicines by name. The inventory list must dynamically filter to show only matching results as the user types. |

**CHAPTER III**

**NON-FUNCTIONAL REQUIREMENTS**

This chapter defines the quality attributes and constraints of the system; how well the system performs its functions. These requirements are critical for ensuring the application is not only functional but also reliable, secure, and user-friendly in its intended desktop environment.

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Requirement ID | Requirement Description | Acceptance Criteria |
| Performance | **NFR-001** | **Application Startup Time** | The packaged application (.exe) shall launch and display the login screen, ready for input, within 5 seconds on a machine meeting the minimum hardware specifications. |
|  | **NFR-002** | **API Response Time** | All backend API responses for standard CRUD operations (create, update, delete) shall complete in under 500 milliseconds. |
|  | **NFR-003** | **UI Responsiveness** | The user interface shall remain responsive during data fetching operations. Asynchronous API calls must be used to prevent the UI from freezing. |
| Usability | **NFR-004** | **Learnability** | A new user with basic computer literacy must be able to perform all core tasks (add patient, add consultation) without formal training. |
|  | **NFR-005** | **Navigational Efficiency** | All primary modules (Patients, Inventory) shall be accessible with a single click from any main page via the sidebar navigation. |
| Reliability | **NFR-006** | **Data Persistence** | All data entered by the user (new patients, consultations, etc.) must be immediately and permanently saved to the SQLite database file located in the user's local application data directory. |
|  | **NFR-007** | **First-Run Stability** | The application must successfully launch, create its database, and seed the initial admin user on the very first run after a fresh installation on a new machine. |
| Security | **NFR-008** | **Password Storage** | All user passwords must be one-way hashed using the bcryptjs library before being stored in the users table. Plain-text passwords shall never be stored. |
|  | **NFR-009** | **SQL Injection Prevention** | All database queries that include user-provided data must use parameterized statements to prevent SQL injection vulnerabilities. |
| Deployment | **NFR-010** | **Standalone Executable** | The entire system (frontend, backend, database engine) must be packaged into a single installer (.exe) for Windows using Electron and electron-builder. |
|  | **NFR-011** | **Dependency Encapsulation** | The installed application must run without requiring the end-user to pre-install any external dependencies like Node.js or SQLite. |

**CHAPTER IV**

**USE CASES**

This chapter describes the interactions between users (actors) and the system to achieve specific goals.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | Use Case Name | Actors | Description | Preconditions | Postconditions |
| UC-001 | Log into the System | Clinic Staff | A clinic staff member with valid credentials authenticates themselves to gain access to the system's functionalities. | The application is running and displaying the login screen. | The user is successfully authenticated, their session is initiated, and they are redirected to the main application dashboard. |
| UC-002 | Manage Patient Records | Clinic Staff | A clinic staff member performs Create, Read, Update, and Delete (CRUD) operations on patient records. | The user is logged into the system and is on the "Patients" page. | The patient records in the database are accurately modified as per the user's actions. |
| UC-003 | Manage Patient Consultations | Clinic Staff | A clinic staff member views a patient's medical history and adds a new consultation record for a recent visit. | The user is logged into the system. A patient record exists. The user has navigated to the patient's detailed view (via UC-002). | The patient's consultation history is updated with the new entry and saved to the database. |
| UC-004 | Manage Medicine Inventory | Clinic Staff | A clinic staff member performs Create, Read, Update, and Delete (CRUD) operations on the medicine inventory. | The user is logged into the system and is on the "Inventory" page. | The medicine inventory in the database is accurately modified as per the user's actions. |

**CHAPTER V**

**DATA REQUIREMENTS**

This chapter details the structure, entities, and relationships of the data that the system will manage.

**Data Entities**

The system will manage four primary data entities: users, patients, consultations, and medicines.

**Attributes:**

* + **users:**
    - id (Primary Key, Integer)
    - username (Text, Unique)
    - password\_hash (Text)
  + **patients:**
    - id (Primary Key, Integer)
    - name (Text)
    - dob (Text)
    - address (Text)
  + **consultations:**
    - id (Primary Key, Integer)
    - patient\_id (Foreign Key, Integer)
    - complaint (Text)
    - diagnosis (Text)
    - treatment (Text)
    - consultation\_date (Text)
  + **medicines:**
    - id (Primary Key, Integer)
    - name (Text, Unique)
    - quantity (Integer)
    - description (Text)

**Relationships:**

A one-to-many relationship exists between patients and consultations. One patient can have many consultations.

**CHAPTER VI**

**ASSUMPTIONS AND CONSTRAINTS**

This chapter outlines the key assumptions that have influenced the project's requirements and design, as well as the specific constraints that limit the project's scope, resources, or technical approach. A clear understanding of these factors is essential for realistic planning, development, and evaluation of the Clinic Records Management System.

**Assumptions**

Assumptions are conditions or factors that are believed to be true for the purpose of this project, but have not been formally verified. The successful outcome of the project may depend on these assumptions being correct.

* User Proficiency: It is assumed that the end-users (clinic staff) are computer literate and proficient in using standard desktop applications, including interacting with forms, buttons, and menus. The system is designed for users with basic computer skills, and extensive computer literacy training is not within the project's scope.
* Hardware Availability: It is assumed that Hi-Precision Diagnostics – Malabon Branch will provide a suitable and dedicated desktop or laptop computer for the deployment of the CRMS. This machine is expected to meet the minimum hardware specifications required for the stable operation of the software.
* Data Availability and Entry: It is assumed that the clinic staff will be responsible for the initial and ongoing entry of all patient, consultation, and medicine data. The project does not include a one-time, bulk migration of existing paper records into the new digital system.
* Stable Operating Environment: It is assumed that the computer hosting the application will have a stable power supply and will be maintained in a standard office environment to prevent data loss or hardware failure.

**Constraints**

Constraints are real-world limitations or restrictions that the project must operate within. These factors can be technical, financial, or resource-based and directly impact the development process and final deliverables.

* Technical Constraints:
  + Offline Operation: The system is explicitly required to function without a persistent internet connection. This dictates the choice of a local database (SQLite) and a standalone application architecture (Electron.js), precluding the use of cloud-based services or web-only frameworks.
  + Platform Dependency: The initial development and deployment target is the Windows operating system. While the chosen technologies support cross-platform builds, official support and testing will be limited to Windows.
  + No Third-Party Integrations: As defined in the scope, the system will not integrate with any external APIs, laboratory machines, or existing Health Information Systems. All data will be managed exclusively within the CRMS.
* Resource Constraints:
  + Sole Developer: The project will be designed, developed, tested, and documented by a single proponent. This limits the complexity of features that can be implemented and necessitates a focused approach to the core requirements.
  + No Financial Budget: The project operates with a zero-dollar budget. This restricts the technology stack to open-source software and free development tools, prohibiting the use of paid libraries, proprietary software, or commercial assets.
* Schedule Constraints:
  + Fixed Academic Timeline: The project must be completed within the fixed timeframe set by the academic calendar of Datamex College of Saint Adeline, Inc. This includes the delivery of all documentation, the final application, and the thesis defense, which imposes a strict deadline on all development activities.

**CHAPTER VII**

**GLOSSARY**

This chapter provides definitions for key terms, acronyms, and abbreviations used throughout the document to ensure clear communication.

|  |  |
| --- | --- |
| Term | Definition |
| CRMS | Clinic Records Management System. The official name of the software project. |
| API | Application Programming Interface. The backend layer that the frontend communicates with to access and manipulate data. |
| CRUD | Create, Read, Update, Delete. The four basic operations for data management. |
| UI | User Interface. The visual part of the application that the user interacts with (the frontend). |
| SQLite | A self-contained, serverless database engine used for data storage in this project. |
| Electron.js | A framework for creating native desktop applications with web technologies (HTML, CSS, JS). |

**CHAPTER VIII**

**REVISION HISTORY**

This chapter tracks the changes and updates made to this document over time.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 1.0 | August 27, 2025 | Catubay, Mark Lawrence L. | Initial draft of the Requirement Specification Document. |
| 1.1 | September 3,  2025 |  | Made introduction verbose and assumptions and requirements |
| 1.2 | September 7, 2025 |  | Refined Non-Functional Requirements |
| 1.3 | September 8, 2025 |  | Refined Functional Requirements, and Use Cases |

**CHAPTER IX**

**APPENDIX**

This chapter contains supplementary materials, diagrams, and other information that supports the main body of the document.