SOFTWARE REQUIREMENT SPECIFICATION

Title: ***Laguna Provincial Health Office Medical Assistance Indigent Program Records System***

Members:
Alcantara, Alyssa Mae D.
Ledesma, Jhon Carlo F.
Orellano, Kyla V.
Coronado, Nixon E.
Sofer, Erl Teodemar D.
Bagalso, Riana Alexis C.

1. Introduction and Purpose

1.1 Introduction

Healthcare access in the Philippines remains challenging for marginalized groups, especially indigent Filipinos who cannot afford medical care. Despite government initiatives like the Universal Health Care (UHC) law and PhilHealth, many still face high out-of-pocket costs, which accounted for 53.9% of health expenditures in 2019, putting poor families at risk of financial ruin. To address these issues, the Laguna Provincial Health Office (PHO) launched the Medical Assistance Indigent Program (MAIP).

However, the program struggles with managing patient records due to the lack of a centralized data system, leading to inefficiencies and concerns about transparency and fairness in resource distribution. To resolve this, a Records Management System was developed to streamline access to medical records, improve appointment scheduling, and enhance the efficiency and accountability of healthcare services. The system focuses on key areas like access control, data security, backup, and report generation, aiming to improve healthcare delivery for indigent populations in Laguna.

1.2 Purpose

The primary purpose of the Records Management System (RMS) being developed for the Laguna PHO is to address significant challenges in managing medical records effectively. Currently, the PHO faces issues such as difficulties in storing and retrieving records, inconsistencies in patient information, and concerns regarding data security and privacy. These challenges hinder the program's ability to provide timely medical assistance to indigent individuals who rely on the MAIP for support. By implementing a robust RMS, the PHO aims to enhance the organization of medical records, allowing healthcare providers to access complete and accurate patient information quickly. This accessibility is crucial for informed decision-making, which can lead to improved patient outcomes and more efficient healthcare delivery.

Moreover, a comprehensive RMS will centralize patient data, reducing the fragmentation currently experienced across various sources. This centralization is essential for tracking which patients have received aid, how funds are utilized, and

identifying gaps in service provision. By addressing these implementation challenges, the RMS will ensure that medical assistance is distributed fairly and transparently, thereby maintaining public trust and ensuring resources are allocated to those in need.

2. Overall Description

2.1 Product Perspective

The Laguna Provincial Health Office Record Management System (PHO-RMS) is an enhanced web-based application developed to resolve long-standing issues in managing medical assistance records. The original system relied on fragmented, manually maintained data, which led to inefficiencies in storing, retrieving, and securing patient information. The upgraded PHO-RMS centralizes and digitizes medical record management to ensure data integrity, improve accessibility, and support timely healthcare interventions under the Medical Assistance Indigency Program (MAIP).

2.2 Product Functions

The system retains its core functionalities while introducing key improvements, including:

- Medical Records Management: Centralized storage and retrieval of patient records.
- Data Accuracy and Privacy: Consistent, secure handling of sensitive patient information.
- Aid Tracking: Monitoring patient assistance history and fund utilization.
- Analytics Dashboard: Dedicated page for advanced data visualization and reporting.
- UI Utilities Panel: Persistent access to dark mode and integrated chatbot features.
- Chatbot Support: FAQ-based chatbot to assist users with system queries.
- User Experience: Improved navigation and dark mode for visual comfort.

2.3 User Classes

- Overall Administrator: Full system control including user management and analytics oversight

- Town Coordinators: Responsible for managing records and overseeing system usage at the town level

3. Functional and Non-functional Requirements

3.1 Functional Requirements

- Document Storage: Store patient records, requests, appointments, and user roles.
- Search Functionality: Search and filter records by keyword or date.
- Access Control: Restrict features based on user roles.
- User Authentication: Log in users and detect invalid attempts.
- User Management: Add, update, notify, and manage user accounts.
- Request Management: Add, edit, delete, and submit medical requests with attachments.
- Status Update: Update request statuses (approve, disapprove, comment, etc.).
- Municipality Management: Add and manage municipalities.
- Report Generation: Generate reports on status and expenditures.
- Export Reports: Export lists as Excel files.
- Guarantee Letter Generation: Print Guarantee Letters for approved requests.
- Coordinator Account Management: View and create coordinator accounts.
- Editable Request Info: Edit submitted request details.
- Account Settings: Edit profile and change password.
- Sign Out: Log users out securely.
- Backup and Recovery: Support data backup and recovery.

3.2 Non-Functional Requirements

- Backup and Recovery: Quickly recover from failures.
- Performance: Fast and consistent data processing.
- Usability: Easy to use with clear design and minimal issues.
- Efficiency: Quick and accurate access to records.
- Accessibility: Works on major browsers and accessible to all users.
- Transparency and Accountability: Ensure fair data access and decision-making.
- Reliability: Operate consistently, even with partial failures.

4. System Features and Interface

4.1 User Interface Requirements

- Responsive web design compatible with current browsers.
- Unified styling using Bootstrap framework.
- User-friendly navigation with access based on roles.
- Documents optimized for print layout.

4.2 Hardware Interfaces

- Standard server hardware for hosting web applications.
- Database server with sufficient storage and performance.
- Network setup capable of supporting multiple users simultaneously.

4.3 Software Interfaces

- Database: MySQL for persistent data storage.
- Web Server: Compatible with Apache or Nginx.
- Backend: PHP with support for PHP 8.x features.
- Frontend: Utilizes HTML5, CSS3, JavaScript, and Bootstrap.
- Chatbot: Python-based rule-driven chatbot for basic user interactions.

4.4 Communication Interfaces

- Secure data exchange over HTTPS.
- Flask-CORS for chatbot-website communication.
- Email notifications for password changes.

5. Assumptions and Constraints

5.1 Assumptions

- User Roles: Assumes defined roles: Admin, Staff, and Patients.
- Hardware/Software: Assumes users have basic computers and web browsers.
- MAIP Workflow: Assumes existing process includes status updates and request handling.
- Guarantee Letter: Assumes auto-generation of Guarantee Letters is essential.

- Patient Data: Assumes required fields like name, contact, and medical info.
- Document Uploads: Assumes users will upload supporting documents.
- User Acceptance: Assumes user satisfaction is key to system success (via TAM).
- Process Integration: Assumes support for existing MAIP reporting workflows.
- Auto-Population: Municipality auto-fills based on coordinator's assigned area.
- Agile Flexibility: Assumes evolving requirements during development.

5.2 Constraints

- Project Timeline: Agile was used to meet strict project deadlines.
- Available Resources: Development was limited by team size, budget, and available hardware/software.
- Tester Availability: Only 8 of 10 planned experts were available for testing.
- Data Security: Sensitive medical data required strict access control and protection.
- Infrastructure Context: System needed to fit Laguna's healthcare infrastructure.
- Data Volume: Must handle large amounts of patient and financial data.
- Quality Standards: System had to meet usability, efficiency, and reliability benchmarks.
- Technical Challenges: Faced risks like compatibility issues and integration complexity.
- Data Consistency: Maintaining consistency across related databases was challenging.
- Backup and Recovery: Required reliable backup and recovery features.
- Scope Limitations: Focused only on system design, development, and testing within Laguna PHO MAIP.

6. Use Case Diagram or Descriptions

Use Case Description: Laguna PHO MAIP Records Management System

Actors:

1. Admin
2. Coordinator

5.1 Use Cases

1. **Login**

- Description: User securely logs into the system using credentials.

- Actors: Admin, Coordinator

2. **Logout**
   - Description: User securely logs out of the system to end the session.
   - Actors: Admin, Coordinator

3. **Manage Account**
   - Description: View and update user profile details and change password.
   - Actors: Admin, Coordinator

4. **Manage Requestor List**
   - Description: View, add, filter, and update patient or requestor information.
   - Actors: Admin, Coordinator

5. **Manage Request**
   - Description: Add, edit, delete, and process medical assistance requests with attached documents and remarks.
   - Actors: Admin, Coordinator

6. **Update Request**
   - Description: Update the status of patient requests (approve, disapprove, comment, assign code or proponent).
   - Actors: Admin, Coordinator

7. **Manage Municipality**
   - Description: Add or edit municipality data in the system.
   - Actors: Admin, Coordinator

8. **Manage Account Type**
   - Description: Define and update access roles and permissions within the system.
   - Actors: Admin

9. **Manage Coordinator**
   - Description: View, add, or update coordinator accounts.
   - Actors: Admin

10. **Access Chatbot**
    - Description: Interact with a built-in chatbot to ask help-related questions about how to use the system.
    - Actors: Admin, Coordinator

11. **Toggle Dark Mode**
    - Description: Enable or disable dark mode for improved user comfort.
    - Actors: Admin, Coordinator

7. Testing Tool Documentation

7.1 Testing Framework: ApacheBench & Web Vitals

Tool Selection Rationale: ApacheBench and Core Web Vitals tools were used to validate the performance and responsiveness of the Laguna PHO Record Management System based on the following criteria:

- Performance Testing: ApacheBench provides benchmarking for concurrent load handling
- Web Metrics Compliance: Tools provide Core Web Vitals metrics such as LCP, CLS, and INP
- Simplicity: Lightweight setup ideal for local testing environments
- Real-User Metrics: Captures and reflects actual user interaction experiences
- Compatibility: Works seamlessly with localhost server environments during development

7.2 Testing Implementation

Performance Benchmark:

Test Tool: ApacheBench

Total Requests: 100

Concurrency Level: 10

Failed Requests: 0

Requests per Second: 329.04 req/sec

Average Time per Request: 30.391 ms

Longest Request Duration: 164 ms

Transfer Rate: 1755.42 KB/sec

Core Web Vitals:

Largest Contentful Paint (LCP): 1.66 s – Good

Cumulative Layout Shift (CLS): 0.01 – Good

Interaction to Next Paint (INP): 16 ms – Good

Response Element Samples:

- LCP Element: p.welcome-text
- INP Trigger: Keyboard interaction

7.3 Additional Testing Tools

Supporting Tools:

- Browser DevTools: Collected Web Vitals metrics via local Lighthouse
- HAR File Analysis: Exported and evaluated using Chrome DevTools
- XAMPP: Local environment simulation using Apache 2.4.58

Quality Assurance Metrics:

- Response time: Maintained under 200 ms for 99% of requests
- System stability: 0 failed requests across load test
- Performance: Exceeded standard thresholds for local development benchmarks
- User experience: Optimized layout stability and fast interaction response