**A PROPOSED OFFERING OF WEB BASED PET MEDICAL RECORDS SYSTEMFOR PETLINK VETERINARY CLINIC, 10TH AVENUE, CALOOCAN CITY**

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:

Mamay, John Carlo P.

Remoto, Rachel

Solidum, Jasmin E.

**Technical document**

**Purpose of the Document**

The purpose of this document is to provide comprehensive technical documentation for the Pet Medical Record system. It is intended to guide developers, testers, system administrators, and other stakeholders in understanding the system’s architecture, functionalities, and technical specifications. This document will also assist in the maintenance and future development of the software.

**Overview of the Software System**

Pet Medical Record is a software application designed to manage and store medical information for pets. It allows veterinarians, pet owners, and clinic staff to easily access and update a pet’s health history, vaccination records, prescriptions, and other relevant data. The system aims to streamline the management of pet healthcare by providing a centralized, user-friendly, and secure platform for medical record keeping.

**Scope of the Technical Documentation**

This documentation covers the technical aspects of the Pet Medical Record system, including system architecture, database design, functional and non-functional requirements, user roles and permissions, API documentation (if applicable), and deployment instructions. It is focused on the backend and frontend development details, intended for use by individuals involved in system development and support.

**System Overview**

• System architecture. A client-server web-based application accessible on desktop

• High-level components. User interface, authentication module, pet medical records management, and database.

• Deployment architecture. The system can be deployed on a local server environment with secure access.

**Installation Guide**

• System requirements:

• Hardware: Minimum 8GB RAM, and 500MB storage.

• Software: Web browser (Microsoft edge & Chrome), database server (SQLite),

• Dependencies: Frameworks and libraries required for system operation.

• Installation steps: Step-by-step instructions for installing the application and database.

• Configuration settings: Adjustments for database connections, user roles, and server settings

**Configuration Guide**

• Configuration details: Instructions for setting up user roles (Staff, veterinarian and pet owner).

• File formats and parameters: Description of supported data formats for importing/exporting records.

• Customization practices: Recommendations for customizing the system based on clinic requirements.

**API Documentation**

• List of APIs: Endpoints for creating, reading, updating, and deleting pet records.

• Request/response formats: JSON-based API requests and responses.

• Authentication requirements: API key or token-based authentication for secure data access.

**Database Documentation**

• Entity-Relationship Diagram (ERD): Representation of database schema, including pets, owners, medical history.

• Tables and fields: Detailed description of database tables and their relationships.

• Data migration/backup: Procedures for importing legacy records and performing routine backups.

**User Manual**

• Software usage: Instructions for logging in, managing pet profiles, and accessing records.

• User interface: Navigation guide with labeled screenshots (print)

• Workflows: Common tasks such as adding a new pet record, updating medical history.

**Troubleshooting Guide**

• Common issues: Login errors, missing data, and connectivity problems.

• Resolutions: Step-by-step troubleshooting solutions.

• Support information: Contact details for technical support or system administrator.

**Code Documentation**

• Code structure: Description of modules and folder organization.

• Inline comments: Explanations for key functions and logic.

• Coding standards: Naming conventions and best practices followed in the project.

**Testing Documentation**

• Test plan: Objectives, testing levels, and strategies.

• Test cases: Functional tests (e.g., record creation) and non-functional tests (e.g., performance, security).

• Test results: Reports of passed/failed test cases and defect logs.

**Maintenance Guide**

• Procedures: Regular updates and maintenance checks.

• Version control: Use of Git for tracking changes and release management.

• Bug fixes/enhancements: Guidelines for reporting and resolving issues.

**Revision History**

**Approval**

**Appendix**