COMPUTER INVENTORY MANAGEMENT SYSTEM SPECIFICATION DOCUMENT

CIMS

# 1. Introduction

**1.1 Purpose** The purpose of this document is to outline the specifications for the Computer Inventory Management System (CIMS). This system is designed to help organizations efficiently manage their computer inventory, including tracking hardware details, software installations, warranty information, and user assignments.

**1.2 Scope** The CIMS will support the following functionalities:

* Add, update, and delete computer inventory items.
* Track hardware specifications and software installations.
* Manage warranty information.
* Assign computers to users.
* Generate reports on inventory status.

**1.3 Audience** This document is intended for:

* System developers and programmers
* Project managers
* System administrators
* End-users and stakeholders

# 2. System Overview

**2.1 System Description** The Computer Inventory Management System will be a web-based application that provides a user-friendly interface for managing and tracking computer inventory. The system will store data in a relational database and offer various modules for different functionalities.

**2.2 Major Components**

* **Frontend:** User interface for interacting with the system.
* **Backend:** Server-side logic and database management.
* **Database:** Stores all inventory-related data.

# 3. Functional Requirements

**3.1 User Management**

* **FR1:** The system shall allow administrators to create, update, and delete user accounts.
* **FR2:** The system shall support different user roles (e.g., Admin).

**3.2 Inventory Management**

* **FR3:** The system shall allow users to add new computer items with details such as model, category, manufacturer, serial number, specifications, processor, software installed, operating system, warranty information, user assignment, and comments.
* **FR4:** The system shall allow users to update and delete existing computer items.
* **FR5:** The system shall display a list of all computer items in a tabular format with searchable columns.

**3.3 Alert and Notification System**

* **FR6:** The system shall display success or error messages upon performing operations such as adding, updating, or deleting items.

**3.4 Reporting**

* **FR7:** The system shall generate reports on inventory status, including items Working and Faulty.

# 4. Non-Functional Requirements

**4.1 Performance**

* **NFR1:** The system should handle up to 1,000 concurrent users without performance degradation.
* **NFR2:** The system should load the inventory list within 2 seconds.

**4.2 Security**

* **NFR3:** The system shall enforce user authentication and role-based access control.
* **NFR4:** The system shall encrypt sensitive data, such as user credentials.

**4.3 Usability**

* **NFR5:** The system shall have a user-friendly interface that is easy to navigate.
* **NFR6:** The system shall provide tooltips and help documentation for user guidance.

**4.4 Scalability**

* **NFR7:** The system shall be designed to scale with the growing number of inventory items and users.

# 5. System Architecture

**5.1 System Components**

* **Frontend:** Developed using HTML, CSS, Javascript, Bootstrap.
* **Backend:** Developed using a server-side framework PHP.
* **Database:** MySQL as the relational database system.

**5.2 Data Flow**

1. Admin interacts with the frontend interface.
2. Frontend sends requests to the backend.
3. Backend processes the requests and interacts with the database.
4. Database returns the required data to the backend.
5. Backend sends the data back to the frontend, which updates the UI accordingly.

# 6. User Interface Design

**6.1 Main Dashboard**

* A summary view displaying total inventory, Working and Faulty items

**6.2 Inventory List**

* A table listing all computer items with columns for each detail (model, category, manufacturer, etc.).
* Actions to edit or delete items.

**6.3 Add/Edit Inventory Item**

* Form to input or update computer item details.

**6.4 Reports**

* Options to view various reports on the inventory.

**7**. Implementation Plan

**7.1 Development Stages**

1. Requirements gathering and analysis.
2. Design system architecture and UI/UX.
3. Develop frontend and backend components.
4. Integrate frontend, backend, and database.
5. Perform testing (unit, integration, and system testing).
6. Deploy the system to a staging environment for user acceptance testing (UAT).
7. Deploy the system to a production environment.

**7.2 Testing**

* Develop test cases based on functional and non-functional requirements.
* Perform regular testing during development to ensure quality.

**7.3 Deployment**

* Deploy using a cloud platform (e.g., AWS, Azure) or on-premises infrastructure.
* Ensure backup and disaster recovery mechanisms are in place.

# 8. Maintenance and Support

**8.1 Maintenance**

* Regularly update the system to fix bugs and improve performance.
* Add new features based on user feedback and changing requirements.

**8.2 Support**

* Provide user support through a help desk or support ticket system.
* Offer training sessions and documentation to help users effectively use the system.

# 9. Conclusion

The Computer Inventory Management System is designed to streamline the management of computer assets within an organization. By implementing the specified features and adhering to the outlined requirements, the system will provide an efficient, secure, and user-friendly solution for inventory management.

**Appendix A: Glossary**

* **CIMS:** Computer Inventory Management System
* **UI:** User Interface
* **UAT:** User Acceptance Testing
* **API:** Application Programming Interface
* **CRUD:** Create, Read, Update, Delete