



College of Computer Studies  
Software Engineering 1

<b>Name</b>	John Marvin S. Miciano
<b>USN</b>	22011732410
<b>Course Code</b>	CS6209

**Activity Name: Practical Exam for Software Engineering 1/Web App 1**

# ***Computer Shop Management System Documentation***

*Miciano, John Marvin  
2024*



College of Computer Studies  
Software Engineering 1

## Table of Contents

<b>1. Introduction</b>	3
<b>2. System Overview</b>	4
<b>3. Requirements</b>	4
<b>4. System Design</b>	5
<b>5. Software Processes</b>	12
<b>6. System Architecture</b>	13
<b>7. Development and Testing</b>	14
<b>8. Cost-Benefit Analysis</b>	16
<b>9. Risk and Project Management</b>	16
<b>10. Conclusion</b>	17
<b>11. Appendix</b>	18



College of Computer Studies  
Software Engineering 1

## 1. Introduction

### 1.1 Overview of the System

The Computer Shop Management System is a web-based platform designed for management of a computer retail store's inventory. It enables store managers to efficiently handle computer product details, including adding, updating, and deleting records of available computers. The system is designed to improve operational efficiency and enhance data organization.

### 1.2 Objectives



## College of Computer Studies Software Engineering 1

The Objectives of this system is to provide an easy-to-understand interface for managing computer inventory data. By having a centralized management process, this system helps in reducing manual errors, maintain real time inventory data and enhances the productivity and efficiency of the shop's workers.

### 1.3 Scope of the Document

This document provides a detailed overview of what the system's designs are, its requirements, and its architecture and functionality. It also covers both the functional and the non-functional requirements, the schematics of the database, the user interface design and also the implementation details. This document focuses on giving a clear guide to the shop's staff for understanding, deploying and maintaining the system.

## 2. System Overview

### 2.1 Prototype Description

The CSMS prototype is a web-based application designed to manage a computer retail store's inventory. It serves as a centralized platform where the store staff can maintain up to date records of available computers. This includes essential details such as model, brand, price, and stock quantity. The system ensures secure access through user authentication and provides an organized interface for inventory management.

### 2.2 Modules/Features

The Computer Shop Management System features the following:

**User Authentication** – Secure login/logout functionality to ensure that only authorized personnel can access and manage the inventory.

**Computer Inventory Management:** – Allows users to add new computer records with details like model, brand, price, and stock.  
– Enables modification of existing computer records to keep information up to date.  
– Provides the option to remove outdated or unnecessary records from the inventory.  
– Displays a list of all available computers with details, allowing for quick reference and inventory checks.

**Session Management** – Manages session duration to ensure user security, with automatic logout after a period of inactivity.

## 3. Requirements

### Software Requirement Specifications (SRS)

#### 3.1 Functional Requirements

**User Authentication** – The system must allow users to log in securely and restrict access to authenticated users.

**Computer Inventory Management** – **Add Computer:** The system must provide a form to add new computer entries with fields for model, brand, price, and stock.  
– **Edit Computer:** The system should allow users to modify details of existing computer records.  
– **Delete Computer:** Users should be able to delete computer records that are no longer needed.  
– **View Inventory:** The system should display a list of all computers in the inventory with relevant details.

**Session Management** – Automated session expiration after a set period of inactivity of users, enhancing security.

#### 3.2 Non-Functional Requirements



## College of Computer Studies Software Engineering 1

**Performance** – The system should be able load inventory data efficiently, supporting fast access to the database.

**Usability** – The interface should be user-friendly and intuitive for a quick implementation to the store

**Security** – User passwords must be stored securely, and the system should prevent unauthorized access through session management.

**Reliability** – The system should maintain accurate data storage and retrieval, even under concurrent access.

### 3.3 Constraints

**Browser Compatibility** – The system should be accessible through modern web browsers including Google Chrome, Firefox, and Microsoft Edge

**Database Compatibility** – The system operates with a MySQL-compatible database to manage inventory data.

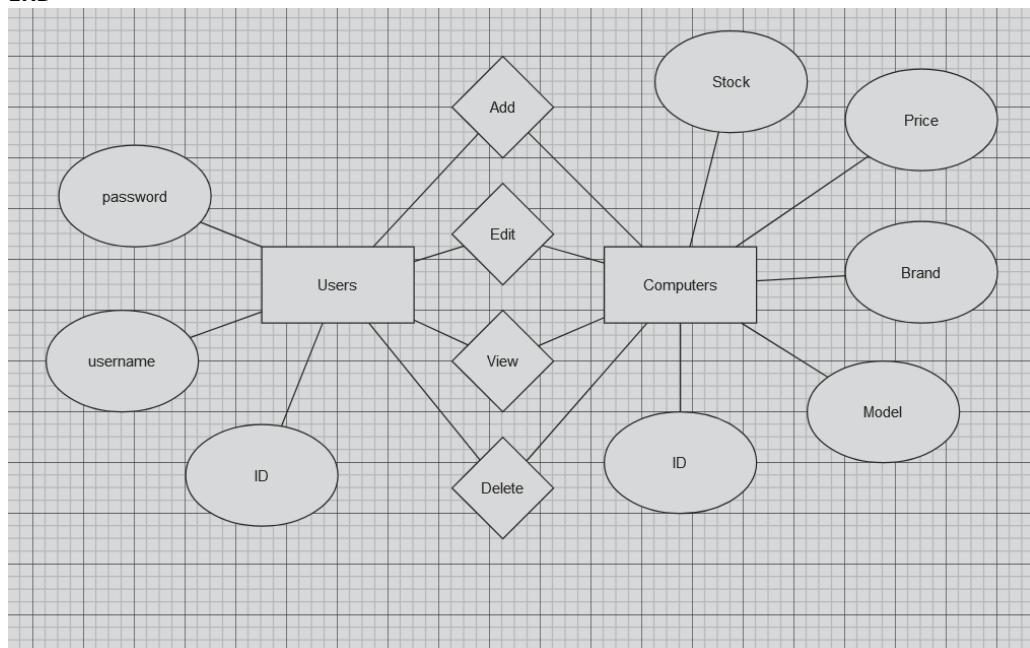
**Web Server** – The application is hosted on a PHP-enabled server, compatible with PHP 7.4 or higher.

**Limited Resources** – It should operate efficiently within limited hardware and network resources available to small retail stores with limited budget.

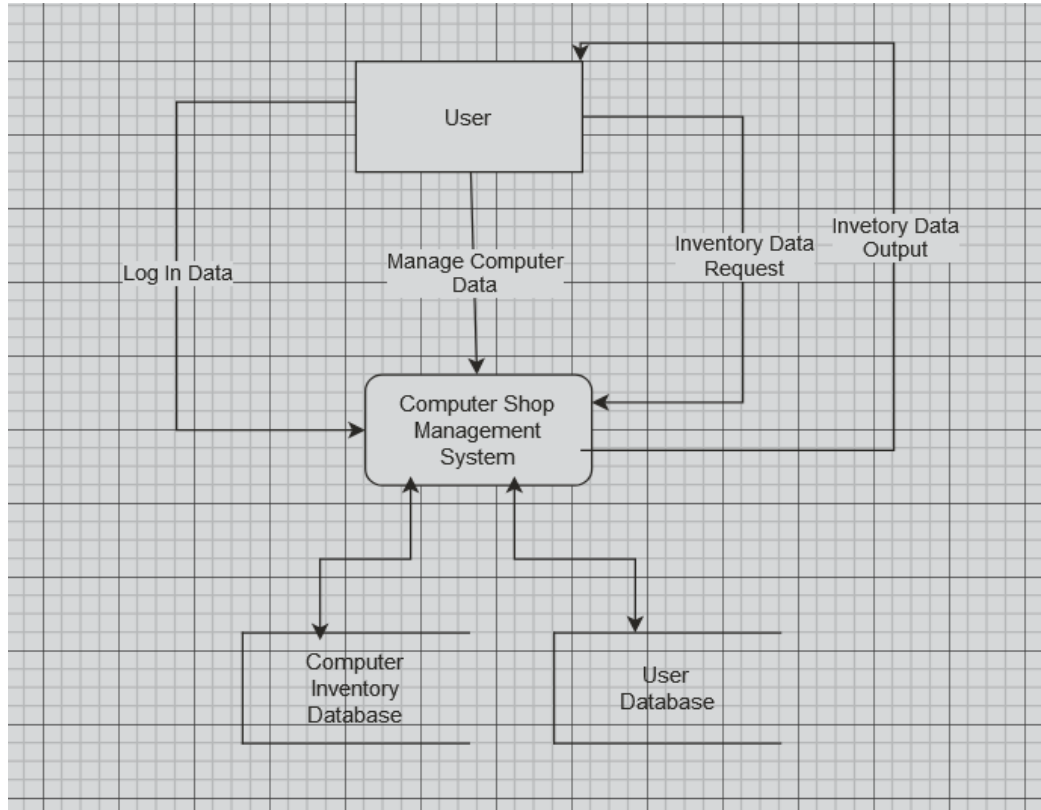
## 4. System Design

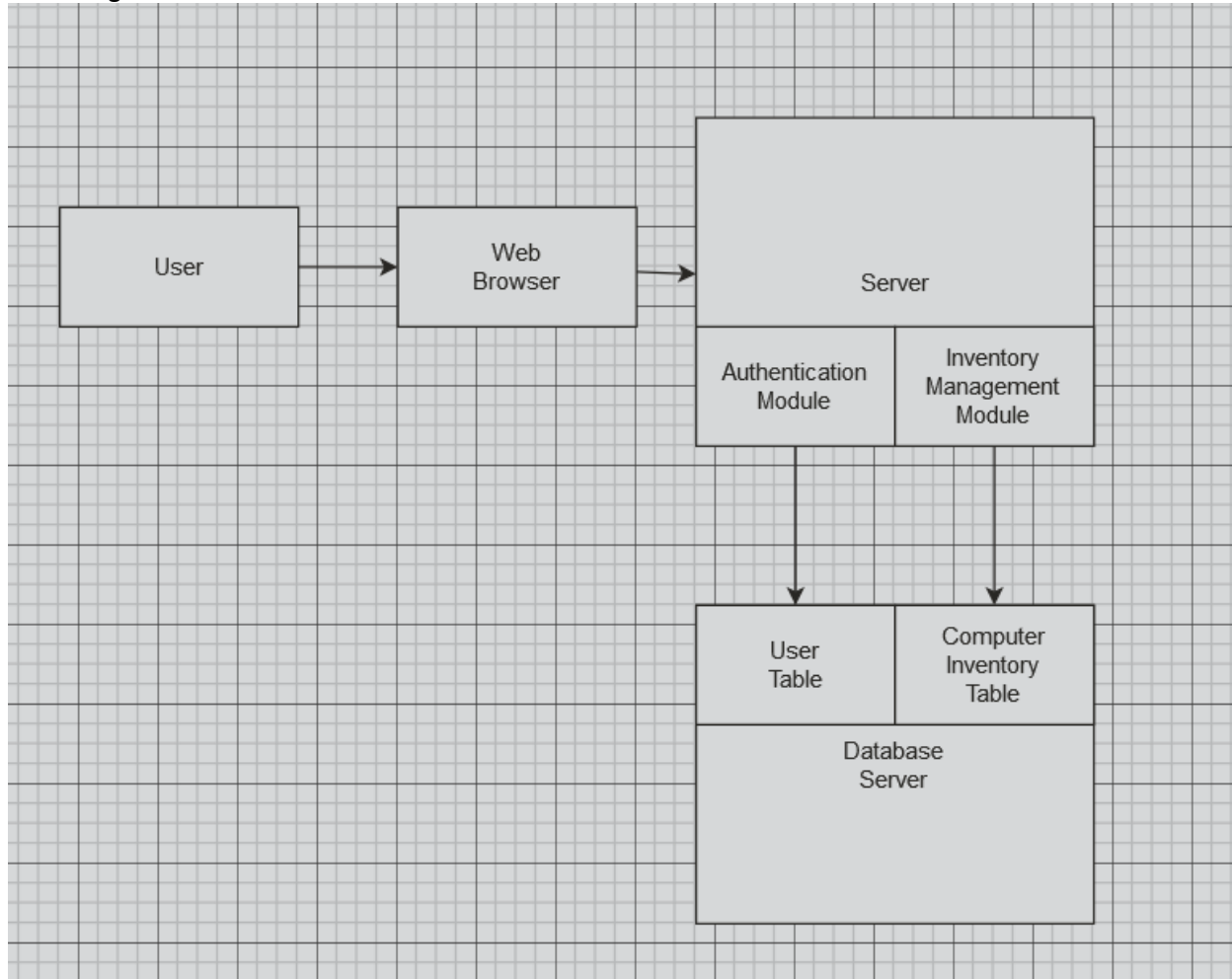
### 4.1 ERD, DFD and Block Diagrams

#### ERD



DFD Lvl 0



**Block Diagram**



College of Computer Studies  
Software Engineering 1

## 4.2 Wireframes and UI Design

### Log in Page

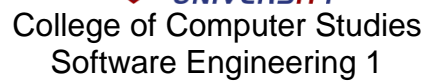
The wireframe shows a login page for the 'Computer Shop Management System'. At the top, there is a header area with the system title and two links: 'Log In' and 'Register'. Below the header is a central login form containing a 'Username:' label, a text input field, a 'Password:' label, another text input field, and a 'Log in' button. Above the form, an error message 'Invalid username or password' is displayed. The footer contains the copyright notice: '© 2024 Computer Shop Management System. All rights reserved'.

**Purpose:** Allows Users to Log in with their credentials, and if no account yet, then create through the Register button

**Layout:**

- 1.) Username Field – a text field for inputting the staff's username
- 2.) Password Field – a text field for inputting the staff's password
- 3.) Log in button – a button to log in once the credentials has been filled up
- 4.) Error Message – Display for error message if the login credentials are invalid
- 5.) Title – title of the website
- 6.) Log in link – move to the page for logging in staff members
- 7.) Register link – move to the page for registering staff members
- 8.) Footer



[illegible]

**Layout: 1.)** View Computers – Move to the view computer page

3.) Log out – Destroys the user's session and move them to the Log in Page

[illegible]

**Layout: 1.) View Computers** – Move to the view computer page



College of Computer Studies  
Software Engineering 1

- 2.) Main Menu – Move to the Menu page
- 3.) Log out – Destroys the user's session and move them to the Log in Page
- 4.) Add Computer – Move to the Add Computer page
- 5.) Edit button – Moves to the Edit Computer page
- 6.) Delete button – Deletes the Computer
- 7.) Back button – Moves back to the previous page

**Add Computer Page**

**Computer Shop Management System**

Main Menu | View Computers | Add Computer | Log out

**Add New Computer**

Model:

Brand:

Price(USD):

Stock:

Add Computer  
Back

© 2024 Computer Shop Management System. All rights reserved

**Purpose:** Add a new computer, that also allows to input their model, brand, price In USD and stock available

- Layout:**
- 1.) View Computers – Move to the view computer page
  - 2.) Main Menu – Move to the Menu page
  - 3.) Log out – Destroys the user's session and move them to the Log in Page
  - 4.) Add Computer – Move to the Add Computer page
  - 5.) Model Text box – Allows input of Model of the computer
  - 6.) Brand Text box – Allows input of Brand of the computer
  - 7.) Price(USD) Text box – Allows input of Price of the computer in USD
  - 8.) Stock – Allows input of number of that computer in stock
  - 9.) Add Computer – A button that submits all the credentials of the computer
  - 9.) Back button – Moves back to the previous page



College of Computer Studies  
Software Engineering 1

**Edit Computer Page**

The screenshot shows the 'Edit Computer' page of the 'Computer Shop Management System'. The page has a light gray header with the system name and a navigation bar with links: 'Main Menu', 'View Computers', 'Add Computer', and 'Log out'. The main content area is titled 'Edit Computer' and contains a form with four text input fields labeled 'Model:', 'Brand:', 'Price(USD):', and 'Stock:'. Below these fields are two buttons: 'Update Computer' and 'Back'. The footer of the page contains the copyright notice: '© 2024 Computer Shop Management System. All rights reserved'.

**Purpose:** Add a new computer, that also allows to input their model, brand, price In USD and stock available

**Layout:** 1.) View Computers – Move to the view computer page  
2.) Main Menu – Move to the Menu page  
3.) Log out – Destroys the user's session and move them to the Log in Page  
4.) Add Computer – Move to the Add Computer page  
5.) Model Text box – Allows input of Model of the computer  
6.) Brand Text box – Allows input of Brand of the computer  
7.) Price(USD) Text box – Allows input of Price of the computer in USD  
8.) Stock – Allows input of number of that computer in stock  
9.) Update Computer – A button that submits all the credentials of the computer  
9.) Back button – Moves back to the previous page



## 5. Software Processes

### 5.1 SDLC Overview

The chosen SDLC model for the Computer Shop Management System is the Waterfall Model due to its structured, straightforward, linear and systematic approach in developing the application.

Below are the SDLC phases of the project.

**Requirement Gathering and Analysis** – Understanding the requirements for the system which includes user roles, inventory management and database requirements

- Identify both the functional and non-functional requirements

**System Design** – Create the design of the system and its architecture, which includes the ERD, DFD and the block diagrams of the system

- Design the schematics of the database, the wireframes and user layout of the website

**Implementation** – Develop the code for the system using HTML, CSS, PHP and MySQL

- Develop and implement security measures like password hashing and automated session closing after a certain time

**Testing** – Integrates various modules and testing the system's functionality, performance and security

- Conducts unit and integration testing to ensure reliability

**Development** – Deploy the system on hosting provider, which is Hostinger, making it accessible at <https://mycomputershopmanagement.com>

- Configures the server environment through Hostinger for optimal performance and security

**Maintenance** – Provides ongoing support for bug-fixes, security patches and system updates

- Occasional database maintenance and optimization to cater the needs of the user

### 5.2 STLC and its phases

**Requirement Analysis** – Reviews the requirements as to create a testing strategy

- Ensures all requirements has corresponding test cases for verification

**Test Planning** – Creates a test plan that outlines the test objectives, scope, resources, schedule and deliverables

- Determine the type of tests required

**Test Case Development** – Writes detailed test cases for each feature

- Prepares test data to simulate the user's interactions and data flows

**Test Environment Setup** – Configure the test environment to mirror the user's environment

**Test Execution** – Execute the tests and record its results

- Document any issues, bugs or unexpected behavior from the system

**Test Cycle Closure** – Analyze and summarize test findings

- Fix any bugs and confirm that the system meets all its requirements and standards

#### Software Testing Model Used

The chosen Testing Model to use for CSMS is V-Model due to its alignment with the waterfall model, its emphasis on verification at early stages and its clear and structured testing phases.



## 6. System Architecture

### 6.1 Client-Server Model

The CSMS structure provides a scalable, secure and manageable approach for handling user interactions and data storage

#### Client (Front-end):

**Description** – The client is accessed through a web browser, enabling users (shop staff) to interact with the system's functionalities, such as adding, editing and viewing the stock

**Technologies used** – HTML, CSS and PHP were used to create a simple yet intuitive interface

**Responsibilities** – Sends user requests to the server  
– Displays information received from the server, such as inventory lists

#### Server (Back-end):

**Description** – The server processes client's requests, interacts with the database and send responses back to the client

**Technologies used** – PHP were used for server-side scripting, managing requests, session-handling and secure data handling

**Responsibilities** – Verifies user credentials and manages session and access  
– Process CRUD operations on handling computer records  
– Ensures inactive users to be logged out automatically as to secure the system

#### Database Server:

**Description** – The database stores and manages persistent data for the system, which includes user information and computer inventory

**Technologies used** – MYSQL is used for reliable, scalable and efficient data storage

**Responsibilities** – Stores user information  
– Stores each computer information  
– Ensures data integrity and security

### 6.2 Database Design

#### User Table:

**id (Primary key)** – Unique identifier for each user

**username** – user's login name

**password** – Hashed password of the user for security

#### Computer Table:

**id (Primary key)** – Unique identifier for each computer record

**model** – Model of the computer

**brand** – Brand of the computer

**price** – Price of the computer

**stock** – Quantity available in stock



## 7. Development and Testing

### 7.1 Technologies and Tools Used

- Frontend Technologies** – HTML and CSS for structuring and styling the user interface
  - PHP for adding interactivity
- Backend Technologies** – PHP for handling requests, processing business logic and managing sessions
- Database** – MySQL for managing persistent data storage, handling tables for user credentials and computer inventory
- Development Tools** – Notepad++ for developing and organizing project files
  - phpMyAdmin as database management tool for interacting with MySQL and perform queries
  - XAMPP for providing a local development environment for both PHP and MySQL
- Testing Tools** – Firefox Dev Tools for debugging, inspecting elements and testing responsive design
  - Manual testing for end to end functionality testing

### 7.2 Test Cases and Results

#### Login and Logout module:

[https://docs.google.com/spreadsheets/d/1oN7KKE8Nvo8\\_pmlACjQoIldw50r46bnm2oIQKHGWhSs/edit?gid=0#gid=0](https://docs.google.com/spreadsheets/d/1oN7KKE8Nvo8_pmlACjQoIldw50r46bnm2oIQKHGWhSs/edit?gid=0#gid=0)

#### Register module:

<https://docs.google.com/spreadsheets/d/1QcAKv3NuMEPVtVfmYz9cMQR6VEFd9xckKPAJZKd0yAU/edit?gid=0#gid=0>

#### Menu module:

<https://docs.google.com/spreadsheets/d/1EV6de63uYDEBwxDU1cqPRBx-sr6j6uWr9GVHp6r3kxM/edit?gid=0#gid=0>

#### Add Computer module:

<https://docs.google.com/spreadsheets/d/1t9aGfB8MxRn1ne1lAs1N6LV0kfrz41Y6qdi-EIYRdY/edit?gid=0#gid=0>



College of Computer Studies  
Software Engineering 1

**View Computers module:**

[https://docs.google.com/spreadsheets/d/1p6jqx7qY\\_3x7\\_-RSHsqS-VH2X0GbXwy4eokjGug9oNM/edit?gid=0#gid=0](https://docs.google.com/spreadsheets/d/1p6jqx7qY_3x7_-RSHsqS-VH2X0GbXwy4eokjGug9oNM/edit?gid=0#gid=0)

**Edit Computer module:**

[https://docs.google.com/spreadsheets/d/1GSe\\_5tNWgXzql0qPQJK7CTOQ24uwVijclHrxLvGYkHQ/edit?gid=0#gid=0](https://docs.google.com/spreadsheets/d/1GSe_5tNWgXzql0qPQJK7CTOQ24uwVijclHrxLvGYkHQ/edit?gid=0#gid=0)

### 7.3 Summary of Testing

Testing of the CSMS focused entirely on functionality and usability aspects to ensure a reliable user experience. Some security aspects were also tested.

**Functionality testing** – Verified that all the CRUD operations work as expected, allowing users to manage the inventory without errors

**Security testing** – Ensures session management works to protect user data  
– Verifies that only correct credentials can start the session

**Usability testing** – Ensures that the user interface is intuitive and responsive to provide a smooth experience for future users



## 8. Cost-Benefit Analysis

### 8.1 Cost Estimation

#### Development Costs:

**Developer Time** – Estimated at 8 hours at \$5/hr = \$40

**Design and Wireframing** – Estimated at 4 hours at \$2.5/hr = \$10

#### Hosting and Deployment Costs:

**Domain Registration** – \$14/year

**Premium Web Hosting** – \$4.39/month or \$52.73

**SSL Certificate** – free as it is included in the Domain Registration

**Total Annual Costing** = \$66.73

#### Maintenance and Support:

**Bug Fixes and Updates** – Estimated at \$5/hr

**Total estimated Cost (Initial Development + 1 year of hosting)** = \$116.73

### 8.2 Benefits and ROI

**Increased Efficiency** – By using the system, store staff can manage inventory efficiently without manual inventory-keeping. This reduces the time spent on data entry and searching for the information, translating into cost savings on labor

**Data Accuracy** – Automated inventory management reduces human error, ensuring accurate stock levels, which can improve ordering decisions and reduces losses due to stockouts or overstocking.

**Scalability** – The system provides a scalable solution that can grow with the business accommodating additional staff and inventory without extra costs  
– Upon request, the system can be improved to include record keeping of customers

**Return on Investment** – Assuming that the system will be used in the Philippines, an average cost of labor is \$10.94 per day, at least 10 hours of labor is saved per month, thus the store has a monthly saving of \$13.68 and an annual savings of \$164.10  
– With the total annual savings of \$164.10, the initial investment of \$116.73 could be recouped in approximately 8 and a half months.

## 9. Risk and Project Management

### 9.1 Risk Analysis and Mitigation Strategies





## College of Computer Studies Software Engineering 1

### **Risk: Security Vulnerabilities**

**Description** – The website may be susceptible to security threats like SQL injection, unauthorized access and data breaches

**Mitigation Strategy** – Implement input validation, use prepared statements, enforce strong password policies on the staff and regularly update the system for security patches

### **Risk: Data Loss**

**Description** – Risk of losing data due to server failure, accidental deletion and etc

**Mitigation Strategy** – Implement regular database backups, use transaction controls for critical operations and set up data recovery procedures

### **Risk: System Downtime**

**Description** – Server downtime, network issues and electrical disruption may lead to the system being inaccessible, affecting productivity of the staff

**Mitigation Strategy** – Use a reliable hosting provider, establish contingency plan for network issues and electrical disruption (use of mobile data and generator on standby)

### **Risk: Usability Issues**

**Description** – If the system is not intuitive, it may take some time for staff to learn using the system, increasing cost in training

**Mitigation Strategy** – Conduct user testing, gather feedback and ensure a simple and intuitive design.

## **9.2 Project Timeline**

<https://docs.google.com/spreadsheets/d/1hAONqy0ek2CFb50K8-EKdLRAWQ6WY3-CY-cwFcmGFVc/edit?gid=0#gid=0>

## **9.3 Resource Allocation**

[https://docs.google.com/spreadsheets/d/1Y6Pc59GVwMuQg8KhF8E9IQv-1YA5qIGb\\_Wjq5CRUG-o/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1Y6Pc59GVwMuQg8KhF8E9IQv-1YA5qIGb_Wjq5CRUG-o/edit?usp=sharing)

# 10. Conclusion

## **10.1 Summary**

The CSMS is a comprehensive solution designed to help manage the management of a compute store's inventory. Through its client-server interaction, it provides a secure and a intuitive platform for handling tasks such as adding, editing, viewing and deleting computer records. The system thus ensures data integrity of the shop and optimizes it's inventory management saving time and cost for the shop. It uses mainly PHP with HTML and CSS for is designing and MYSQL for database management. With these proven programs used, the system much more reliable and scalable than manual labor.



## College of Computer Studies Software Engineering 1

### 10.2 Future Enhancements and Improvements

**Reporting and Analytics** – Adding features for sales inventory reports would give the managers or the owner of the shop insights into trends, stock levels, and demand, helping in improving the shop

**Role-Based Access Control** – Implementing roles for different user levels, for example admin, owner, staff, as to restrict access to certain functionalities

**Inventory Alerts** – Implement a notification alert whenever the stock levels are low as to help staff manage inventory proactively

**Search and Filter** – Implement search and filter as to make it easier for users to find specific records in the inventory

## 11. Appendix

### 11.1 Acronyms and Abbreviations

Acronyms/Abbreviations	Full Form
CSMS	Computer Shop Management System
CRUD	Create, Read, Update, Delete
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
SDLC	Software Development Life Cycle
STLC	Software Testing Life Cycle
UI	User Interface
ROI	Return on Investment

### 11.2 References

PHP Documentation: <https://www.php.net/docs.php>

MySQL Documentation: <https://dev.mysql.com/doc/>

HTML and CSS Standards: <https://www.w3.org/standards/webdesign/htmlcss>

Software Engineering 1 PPTs:

<https://www.canva.com/design/DAGPzQsx0SM/2IxHcrNNs9L8Kkg2vtKJlAQ/edit?u>



College of Computer Studies  
Software Engineering 1

[tm\\_content=DAGPzQsx0SM&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton&authuser=0](https://www.canva.com/design/DAGPzQsx0SM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton&authuser=0)

<https://www.canva.com/design/DAGQyAkpeyw/GSJxyCt2RAsYCAIyJ1Pnhw/edit?authuser=0>

[https://www.canva.com/design/DAGUuN2kSw0/KQ9lBm4MiQjFMsdIDcSbEA/edit?utm\\_content=DAGUuN2kSw0&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton&authuser=0](https://www.canva.com/design/DAGUuN2kSw0/KQ9lBm4MiQjFMsdIDcSbEA/edit?utm_content=DAGUuN2kSw0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton&authuser=0)

<https://www.canva.com/design/DAGRb4xy3vU/sq4tz-AKuoU5-WTtnNnb4g/edit?authuser=0>

<https://www.canva.com/design/DAGTPtNZEMU/0XVDVGxSZEqYac3ZjPZokw/edit?authuser=0>

### 11.3 Glossary of Terms

**Client-Server Model** – A network architecture where client devices interact with server resources

**Data Flow Diagram (DFD)** – Diagram showing how data flow through a system

**Entity Relationship Diagram (ERD)** – Diagram used to visualize relationship between different entities in a database

**Inventory Management** – Process of ordering, storing and managing inventory of a shop

**Scalability** – Ability of a system to handle growth