Principles of Programming

Coursework Specification | Villa College

Module Details

Module Code: UFCFHS-30-1	UFCFHS-30-1
Module Title	Principles of Programming
Module Leaders	Hassan Maheedh Mohamed
Component/Element	B
Total Assessments	One comprehensive assignment
Weighting	This coursework represents 50% of your total module grade.
Collaboration	Individual submission

Submission Details

Deadline	2nd September 2024, 11:59 PM
Platform	Moodle
Submission Format	Single ZIP file **(FIRSTNAME_LASTNAME_ID.zip)** **The file should contain:** **.py file, .pdf file and all other supporting files.**

Inventory Management System

A client has approached you with a critical project: **developing an inventory management system for their large-scale electronics warehouse.** This warehouse handles a vast number of electronic items such as smartphones, laptops, tablets, cameras, and accessories. The client expects a robust system that can efficiently manage their high volume of inventory, ensure accurate tracking, and implement security features to prevent theft and loss.

Program Features and Requirements:

1. **CLI-Based Interface**:
    - Develop a user-friendly command-line interface for system interaction, including viewing, adding, updating, and deleting inventory records.
2. **CRUD Operations**:
    - Implement Create, Read, Update, and Delete functionalities for inventory records within the database.
    - Each inventory record should include attributes such as item name, Serial, quantity, location within the warehouse, supplier details, and warranty status etc.
3. **Inventory Tracking**:
    - Monitor and update stock levels in real-time.
    - Generate alerts for low stock or overstock situations to ensure optimal inventory levels.
    - Track high-value items with unique identifiers for security.
4. **Order Management**:
    - Track purchase orders and sales orders.
    - Manage incoming and outgoing shipments.

5. **Reporting**:
    - Generate reports on stock levels, sales, order history, and other relevant metrics.
    - This can be a report(pdf/txt) or report view on CLI.
    - *Provide insights into inventory trends and performance (Bonus).*
6. **Data Validation**:
    - Ensure all input data is properly validated (e.g., positive quantities,valid Serial)
7. **User Authentication**:
    - Implement role-based access control (e.g., admin, staff).
    - Admins have full access, while staff have limited access to specific functionalities.
    - Use encryption for storing sensitive data.
8. **Error Handling**:
    - Implement robust error handling mechanisms to ensure system stability.
9. **Database Implementation**:
    - Use SQLite or text files to implement the database, leveraging Python's sqlite3 module.
10. **Testing**:
    - Include comprehensive testing to cover various use cases and ensure data integrity and system reliability.
11. **Documentation**:
    - Provide detailed documentation, including a system flowchart, setup instructions, a user guide, and a testing strategy.

Instructions

1. **User Authentication**:
    - Implement a role-based authentication mechanism that supports both admin and staff roles.
    - **Admins**: Have full access to the system, including CRUD operations and managing user accounts.
    - **Staff**: Can view and update inventory records but cannot delete items or manage user accounts.
2. **Data Pre-population**:
    - The system should come pre-populated with a set of fictional inventory records for demonstration purposes.
    - Include at least 20-30 inventory records in the database upon initial setup.
    - Ensure that the pre-populated data showcases a variety of information to demonstrate the system's search and filter capabilities effectively.
3. **Instructions for Data Storage and Initial Setup**:
    - Provide a .sqlite file pre-populated with the fictional inventory data.
4. **Login Details**:
    - Specify default login credentials for admin and staff roles in your documentation.
    - Discuss the method of storing and verifying these credentials securely, considering the sensitivity of password data.

Submission Requirements:

- **Python Scripts (.py):** Submit all Python code files, including the database schema, system functionalities, and CLI interface.
- All other supporting files
- Documentation (.pdf)
- **DEADLINE: 2nd September / 11:59 PM**
- **Late Submission:**
    - **-2 Marks for each day, late up to a maximum of 7 days.**
    - **Submissions more than 7 days late will not be accepted.**

Assignment Marking Criteria (Total: 100 Marks)

Category	Criteria	Marks
1. Code Functionality (60 Marks)		
1.1 Data Input and Validation (10)	Successful data input for each inventory record category	5 Marks
	Proper validation of input data	5 Marks
1.2 CRUD Operations (15)	Functional Create, Read, Update, Delete operations	5 Marks
	Efficiency and error handling in CRUD operations	10 Marks
1.3 Inventory Tracking (10)	Real-time stock tracking	5 Marks
		5 Marks
1.4 User Authentication and Security (10)	Implementation of role-based access control and multi-factor authentication	5 Marks
	Secure data storage	5 Marks

1.5 Order Management and Reporting (15)	Efficient tracking of purchase and sales orders	5 Marks
	Generation of detailed inventory reports Insights into inventory trends and performance	10 Marks
2. Documentation (30 Marks)		
2.1 Introduction (3)	Software purpose and Introduction	2 Marks
	User instructions for setup and navigation	1 Marks
2.2 Code Explanation and System Flowchart (15)	In-code comments and documentation	5 Marks
	Detailed function/method explanations with a system flowchart	10 Marks
2.3 Use Cases & Testing (7)	Use case explanations with inputs and outputs	5 Marks
	Testing strategy and results demonstration	2 Marks
2.4 Conclusions & Recommendations (5)	Software performance and reliability reflection	3 Marks
	Future improvement suggestions	2 Marks
3. Online Demo Presentation (10 Marks)		
3.1 Understanding & Clarity (5)	Software functionality demonstration	5 Marks
3.2 Q&A (5)	Question and feedback response during presentation	5 Marks