IT108 – Project Guidelines

1. The project is an individual work or by pair of two.
2. The project must be a documentation of your information system in Computer Programming 1 (Java Programming).
3. You can use tools like Netbeans, Eclipse, Jcreator, Visual Studio.
4. The project is due on ***December 16, 2024***.
5. Schedule of Final Project Panel Defense will be on ***December 16, 2024, onwards.***
6. No late work will be accepted.
7. Marks will be awarded for Completeness of the project documentation.
8. Any images downloaded from the Internet should be given credence.
9. Project materials of each student cannot be shared with others. The results will be withheld for further investigation if any identical projects are found.
10. Proper backup has to be maintained. You should not copy the softcopy of projects into any of school computers to avoid any identical pairs of projects.
11. You can use alternative storage media like internet cloud services.
12. The school will not be responsible for any missing storage media due to the student's negligence.
13. Please refrain from using AI tools to generate any content (text, video, audio, images, code, etc.)
14. Passing off any AI generated content as your own (e.g., cutting and pasting content into project content, or paraphrasing AI content) constitutes a violation of BLUE PHENIX academic integrity policy. If you have any questions about using generative AI consult your trainer.

Tools Allowed

1. Students are to use tools that they have learned in the course.
2. Students may use alternative tools approved by the lecturer.

Project Submission

1. Soft copy of all raw and processed files must be submitted.
2. Student should **print the documentation** with a front page that follows the format.
3. The printed final project documentation must be in **soft bind / ring bind** (*see annex 2*).
4. Submit the printed documentation on the said due date.
5. Burn all your files in a CD/DVD.
6. Write your name on the CD/DVD and attached it at the back of your project documentation.

Project Phases and Deliverables

Phase 1: System Requirements and Design (Programming Focus)

Define the system architecture and develop the skeleton of the Java application.

1. **System Architecture Design:**
    - Develop a high-level design document that outlines the architecture of the system.
    - Include the MVC (Model-View-Controller) design pattern or similar patterns that will be used for system development.
2. **Class Diagrams:**
    - Provide UML class diagrams to represent the key classes and their relationships.
    - Clearly define attributes and methods for each class, and indicate relationships such as inheritance, association, or aggregation.

Deliverable for Phase 1 (Programming):

- System Architecture Design Document
- UML Class Diagrams

Phase 2: Basic System Implementation (Core Modules)

This phase involves implementing the core functionality of the system using Java, focusing on the main features as outlined in the proposal.

1. **Basic Functionality Implementation:**
    - Implement the core modules of the system, such as login, registration, and CRUD (Create, Read, Update, Delete) operations for primary entities (e.g., Users, Products, etc.).
    - Ensure proper input validation and exception handling for all operations.
2. **User Interface (UI):**
    - Develop a simple Java-based user interface
    - Ensure the UI is user-friendly and reflects the system's functionality.
3. **Database Connectivity:**
    - Write code to connect the Java application with the RDBMS.
    - Implement Data Access Objects (DAO) or similar design patterns to handle data interactions.

Deliverable for Phase 2 (Programming):

- Core Modules Implementation (Java Code)

- User Interface (UI) Implementation
- Database Connectivity Code

Phase 3: Advanced System Features

In this phase, the system will be extended to include advanced features and ensure that the system is scalable and secure.

1. **Advanced Features:**
    - Implement additional features such as reporting, search filters, data sorting, or user management (roles and permissions).
    - Optimize the performance of the application.
2. **Data Persistence and Transactions:**
    - Implement features such as data persistence and transaction management (using JDBC, Hibernate, or JPA).
    - Ensure that transactions are properly managed (commit/rollback) when handling database updates.
3. **Error Handling and Logging:**
    - Implement robust error handling and logging mechanisms to track errors or events in the system.
    - Use logging frameworks like java.util.logging or Log4j to capture system logs.

Deliverable for Phase 3 (Programming):

- Advanced Feature Implementation (Java Code)
- Transaction Management Code
- Error Handling and Logging Code

Phase 4: Testing and System Integration

This phase focuses on rigorous testing and integrating all components of the system.

1. **Unit Testing:**
    - Write unit tests for key system components to ensure that individual methods and classes work as expected.
    - Use JUnit or similar frameworks for testing Java code.
2. **Integration Testing:**
    - Conduct integration testing to ensure that the different modules (UI, database, and business logic) work together seamlessly.

3. **User Acceptance Testing (UAT):**
    - Perform UAT by demonstrating the system to a group of users and collecting their feedback.
    - Make necessary modifications based on the feedback.

Deliverable for Phase 4 (Programming):

- Unit Testing Scripts and Results (JUnit, etc.)
- Integration Testing Results
- User Feedback Report

Example System UI