## **LIBRARY MANAGEMENT SYSTEM FOR LIMKOKWING UNIVERSITY**

### 1. ****Project Title****

Library Management System (LMS) for Limkokwing University

### 2. ****Objectives****

The main objective of this project is to design, develop, and implement a Library Management System (LMS) that will automate and streamline the process of managing book records at Limkokwing University. The system will provide an efficient interface for users to interact with the library's collection, with full support for CRUD (Create, Read, Update, Delete) operations. Specific objectives include:

* **Implement CRUD Functionality:** Enable users to add, view, update, and delete book records.
* **User Interface Design:** Develop an intuitive graphical user interface (GUI) using PyQt.
* **Database Management:** Set up a secure SQLite database to store and retrieve book information.
* **Team Collaboration:** Utilize agile development practices and collaborative tools for effective teamwork.
* **Comprehensive Documentation:** Provide clear and detailed technical and user documentation.

### 3. ****Scope****

The Library Management System will include the following features:

* **Create:** Users will be able to add new books to the system with details such as title, author, ISBN, genre, and publication year.
* **Read:** Users will be able to view a list of books in the library. They will also be able to search and filter books by title, author, ISBN, and genre.
* **Update:** Users can modify the details of existing books.
* **Delete:** Books can be deleted from the system when necessary.
* **Search:** A powerful search feature will allow users to quickly find books by various attributes.
* **Clear Form:** Input fields can be cleared after adding or updating book records.

**Technology Stack:**

* **Frontend:** PyQt (for the graphical user interface)
* **Backend:** SQLite (for database management and CRUD operations)
* **Version Control:** Git and GitHub (for collaborative development)

### 4. ****Methodology****

This project will follow the **Agile Software Development Methodology**, which focuses on flexibility, iterative progress, and collaboration. The following steps will be employed:

* **Sprint Planning:** The project will be divided into manageable sprints, each focused on completing specific features.
* **Daily Stand-ups:** Short daily meetings will track progress, identify challenges, and plan the next steps.
* **Iteration Reviews:** After each sprint, a review meeting will evaluate the deliverables and provide feedback for continuous improvement.

### 5. ****Timeline****

The development timeline is as follows:

* **Days 1-5:** Project Proposal & Requirements Analysis – Defining the project's objectives, scope, and functional requirements.
* **Days 6-9:** System Design – Creating UML diagrams, the database schema, and the system architecture.
* **Days 10-14:** Frontend Development – Implementing the user interface with PyQt.
* **Days 15-19:** Backend Development – Implementing database integration and CRUD functionalities.
* **Days 20-24:** Testing & Debugging – Unit testing, system testing, and fixing bugs.
* **Days 25-29:** Documentation – Preparing technical documentation and user manuals.
* **Day 30:** Final Presentation and Submission – Delivering the final project, including all documentation and code.

### 6. ****Deliverables****

The project will deliver the following key outputs:

* **Project Proposal:** A detailed document outlining the project’s goals, scope, and methodology.
* **System Design:** UML diagrams (Class Diagram, Use Case Diagram) and a database schema representing the system architecture.
* **Implementation:** A fully functional Library Management System supporting CRUD operations and database integration.
* **Testing:** Test cases and documentation, including results and bug reports.
* **Documentation:**
  + **Technical Documentation:** A comprehensive guide for developers covering system architecture, code structure, and database design.
  + **User Manual:** Clear instructions for end-users on how to operate the system effectively.

### 7. ****Resources****

The following resources will be required to successfully execute the project:

* **Software Tools:**
  + **PyQt5:** For building the graphical user interface.
  + **SQLite:** For database management and storing book records.
  + **Git/GitHub:** For version control and team collaboration.
  + **IDE (PyCharm):** For writing and testing Python code.
* **Hardware:**
  + Personal computers for development, testing, and deployment.

### 8. ****Team Members and Roles****

The project team will consist of the following roles:

* **Frontend Developer:** Responsible for designing and implementing the GUI using PyQt.
* **Backend Developer:** Responsible for implementing CRUD operations and managing database interactions.
* **Testing & Debugging:** Responsible for ensuring that the system is functional and bug-free.
* **Documentation Specialist:** Responsible for preparing technical and user documentation.

### 9. ****Evaluation Criteria****

The project will be evaluated based on the following criteria:

* **Functionality:** The system should meet all requirements, including CRUD operations and a user-friendly interface.
* **Code Quality:** Code should be clean, well-organized, and adhere to object-oriented programming principles.
* **Usability:** The system should be intuitive and easy to navigate.
* **Documentation:** The user and technical documentation should be clear, comprehensive, and well-organized.
* **Team Collaboration:** The effectiveness of teamwork and individual contributions will be assessed.

### 10. ****System Design****

* **Use Case Diagram:**
  + **Actors:**
    - **Admin:** Manages book records (add, update, delete, and view books).
    - **Guest/User:** Can view and search books but cannot modify records.
* **Class Diagram:**
  + **Classes:**
    - **Book:** Stores information about each book (title, author, ISBN, genre, publication year).
    - **Library:** Manages the list of books and supports CRUD operations.
    - **DatabaseHandler:** Handles interactions with the SQLite database.
    - **GUI:** Manages the user interface and user interactions.

### 11. ****Implementation****

The implementation phase will include the following steps:

* **Development Environment Setup:** Install Python, PyQt5, SQLite3, and Git for version control.
* **Database Management:** Set up the SQLite database and create a schema to store book records.
* **Backend Development:** Implement CRUD operations and integrate them with the frontend.
* **Frontend Development:** Develop the GUI using PyQt to enable user interactions.
* **Testing and Debugging:** Perform thorough testing to ensure that all features work as expected.

### 12. ****User Interface Design****

The user interface will include the following components:

* **Search Bar:** Allows users to search for books by title, author, ISBN, or genre.
* **Books Table:** Displays book information in an organized format (ID, title, author, ISBN, genre, publication year).
* **Add/Update Form:** Allows users to add new books or update existing ones, with fields for title, author, ISBN, genre, and publication year.
* **CRUD Action Buttons:**
  + Add Book
  + Update Book
  + Delete Book
* **Clear Button:** Clears the input fields or search bar.

### 13. ****Testing and Debugging****

Testing will follow these stages:

* **Unit Testing:** Tests individual components like adding or updating books.
* **Integration Testing:** Ensures that the UI and database interact correctly.
* **Functional Testing:** Verifies that the system performs all required CRUD operations.
* **UI Testing:** Ensures the interface is responsive and intuitive.
* **Regression Testing:** Ensures updates do not break existing functionality.

### 14. ****Documentation****

* **Technical Documentation:** Detailed explanation of the system architecture, code structure, and database schema.
* **User Manual:** Step-by-step instructions for users, including installation and usage guidelines.

### 15. ****Conclusion****

The Library Management System for Limkokwing University is designed to offer a robust and efficient solution for managing the library’s book collection. With the integration of a user-friendly GUI, a powerful database backend, and thorough testing, the system is expected to meet the needs of both administrators and library users, ensuring seamless management of book records and providing a valuable tool for the university.