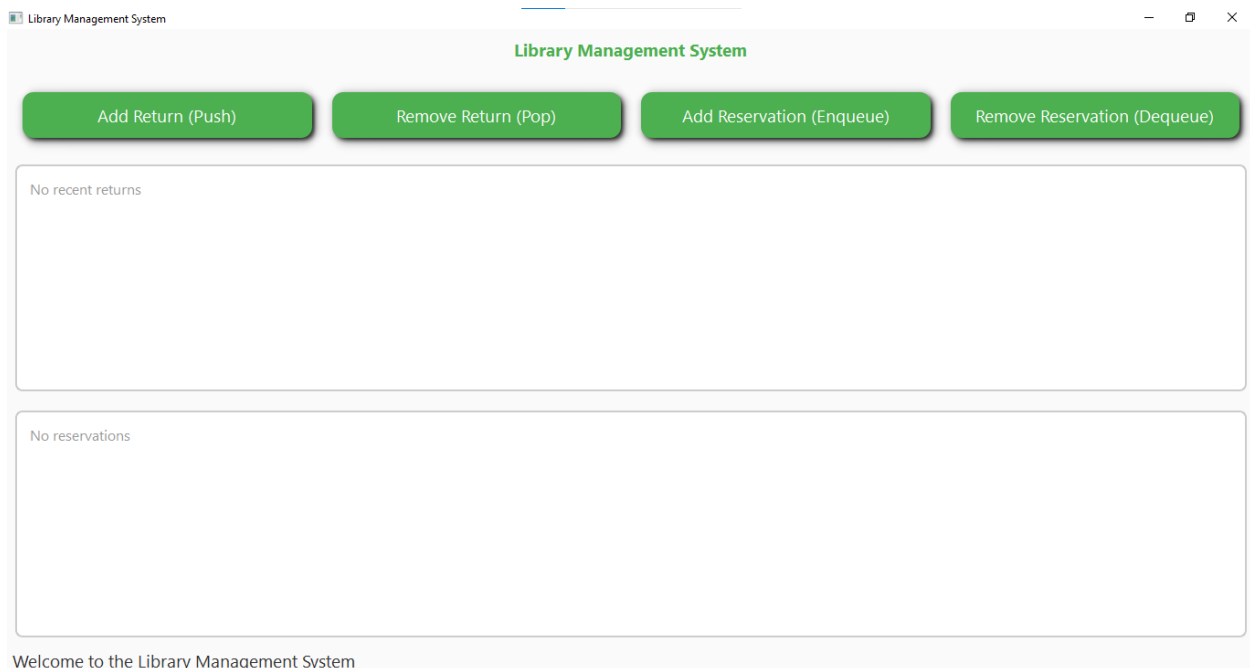


Gabriel Kyle Alcana

Aemiel Aliver Labasan

Loujean Hart De Leon

# Library Management System



## 1. Chosen Application Theme

- **Theme:** Library Management System
- **Objective:** To manage recently returned books and book reservations using stacks and queues. This system allows library staff to quickly view the most recent returns and process reservations in a first-come, first-served manner.

## 2. Rationale for Your Choice

- The Library Management System theme was chosen because it efficiently demonstrates the real-life applications of stacks and queues. Libraries often deal with a large number of returned books and book reservations, making it essential to organize and manage these efficiently. The stack helps track the latest returned books, while the queue ensures fair and orderly handling of reservations.

## 3. Explanation of the Implemented Application and its Features

- Core Features:
  - **Stack (Returned Books):** The stack is used to store the most recently returned books. Users can push new returns to the stack and pop the latest return off the stack.
  - **Queue (Reservations):** The queue is used to manage book reservations in a first-come, first-served manner. Users can enqueue new reservations and dequeue the oldest reservation.
- Graphical User Interface (GUI):
  - The GUI includes buttons for stack operations (Push and Pop) and queue operations (Enqueue and Dequeue).
  - The state of both the stack and the queue is displayed in separate text areas, allowing users to see the latest status of returned books and pending reservations.

## 4. Three Test Cases for the Chosen Application

- **Test Case 1: Adding a Return (Push Operation)**
  - **Action:** Click the "Add Return (Push)" button multiple times.
  - **Expected Result:** Each click should add a new "Book #N" to the stack display, where N increments with each addition. The stack area should show the books in reverse order of their addition.
  - **Outcome:** Confirms the stack's Last In, First Out (LIFO) functionality.
- **Test Case 2: Adding and Removing a Reservation (Enqueue and Dequeue Operations)**
  - **Action:** Click the "Add Reservation (Enqueue)" button multiple times, then click "Remove Reservation (Dequeue)" once.

- **Expected Result:** Each enqueue should add a "Reservation #N" to the queue display. The dequeue should remove the oldest reservation.
- **Outcome:** Confirms the queue's First In, First Out (FIFO) functionality.
- **Test Case 3: Removing a Return (Pop Operation)**
  - **Action:** Add several returns using the push operation, then click "Remove Return (Pop)".
  - **Expected Result:** The most recently added book should be removed from the stack display.
  - **Outcome:** Verifies that the stack correctly removes the latest return.

## 5. Challenges Faced During Development

- **Challenge 1:** Implementing the GUI with a user-friendly design in Qt required learning the layout system and widget customization.
- **Challenge 2:** Handling real-time updates for stack and queue displays, ensuring smooth operation feedback to users.
- **Challenge 3:** Designing intuitive and accurate representations of stack and queue data structures that accurately reflect the changes after each operation.

## 6. Roles of Each Member and Their Contributions

- **Gabriel Kyle Alcana (Developer):** Developed the main logic for stack and queue operations, implemented the basic functionality for push, pop, enqueue, and dequeue.
- **Aemiel Aliver Labasan (Designer):** Designed and implemented the GUI, focusing on layout and button actions to ensure usability.
- **Loujean Hart De Leon (Tester):** Conducted testing and debugging, created test cases, and refined the application based on test results.

