A. Chosen Application Theme

Library Management System

B. Rationale for Team's Choice

For our project, we chose to develop a library management system because it provides a clear and easy way to demonstrate the core concepts of stacks and queues. A library system naturally aligns with these data structures, as stacks can effectively manage recently returned books, and queues can handle book reservations in a first-come, first-served manner. The library theme allows us to easily showcase these concepts, with stacks managing the most recent returns and queues organizing reservations.

We used Qt Creator for the project, which was essential in building this system, providing a user-friendly interface and facilitating seamless integration of the stack and queue functionalities. The stack feature uses three main functions: push, pop, and peek. When a book is returned, it is added to the stack using the push function, and the pop function removes the most recent return when needed. The peek function shows the latest returned book without removing it. We also included a feature to display a list of all returned books.

For book reservations, we implemented a queue, which also uses three functions: enqueue, dequeue, and peek. Reservations are added to the end of the queue, and when fulfilled, they are removed using dequeue. The peek function allows user to view the next reservation without removing it. An option to view the entire reservation list is also included.

In conclusion, the library management system was an excellent choice for our team to demonstrate the actual uses of stacks and queues. The theme clearly shows how various data structures may manage resources in real-world scenarios. Using Qt Creator, we were able to create a functioning and simple system.

C. Major Functions and Features:

The library management system design ensures quick and organized management of both returned and reserved books, with stacks handling "Last-In-First-Out" (LIFO) operations for recently returned books and queues providing "First-In-First-Out" (FIFO) handling for book reservations. The following are the features of the system:

1. **Returned Books (Stack)**
    - **Add Book (Push)**: Adds a returned book to the stack, storing both the book name and the person's name who returned it.
    - **Show Latest Book (Peek)**: Displays the most recently returned book without removing it from the stack.

- **View All Books**: Lists all returned books currently in the stack in the order they were returned. In this function, the pop button is located.
- **Remove Recently Returned Book (Pop)**: Removes the most recently returned book from the stack and displays the details of the removed book.

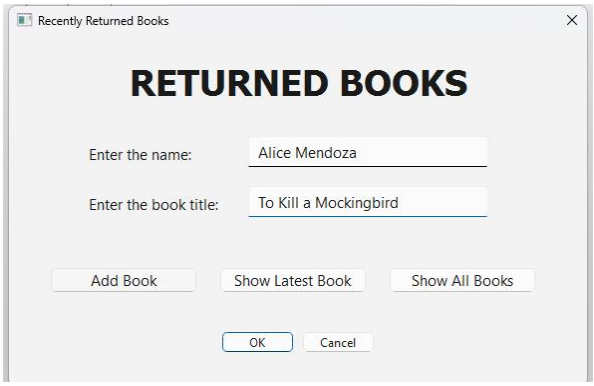
2. **Book Reservation (Queue)**
   - **Reserve Book (Enqueue)**: Adds a book reservation to the queue, keeping track of the order in which reservations were made.
   - **Show First in Queue (Peek)**: Shows the next reserved book to be issued, without removing it from the queue.
   - **Show Queue**: Lists all current reservations in the order they were made. In this function, the dequeue button is located.
   - **Dequeue Book (Dequeue)**: Removes the first reservation from the queue, effectively processing the next person in line.

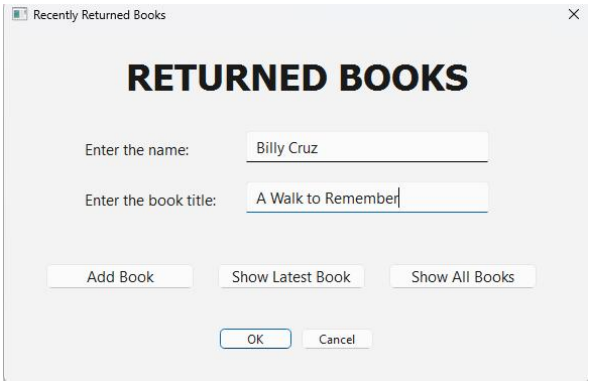
D. Three Test Cases for the chosen application

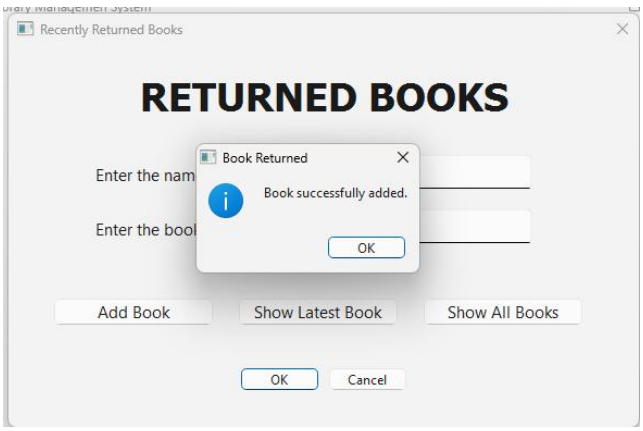
Test Case 1: Add and Peek Returned Book (Stack)

Action: Add the book "To Kill a Mockingbird", which was returned by Alice Mendoza and "A Walk to Remember", returned by Billy Cruz to stack; then peek.

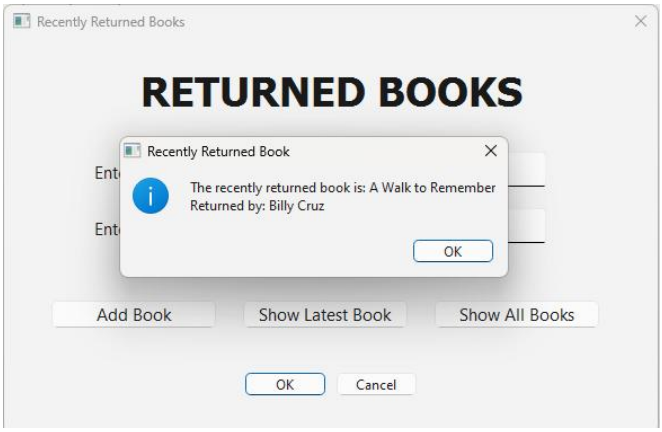
Action: Add Book (Push)







Action: Show Latest Book (Peek)

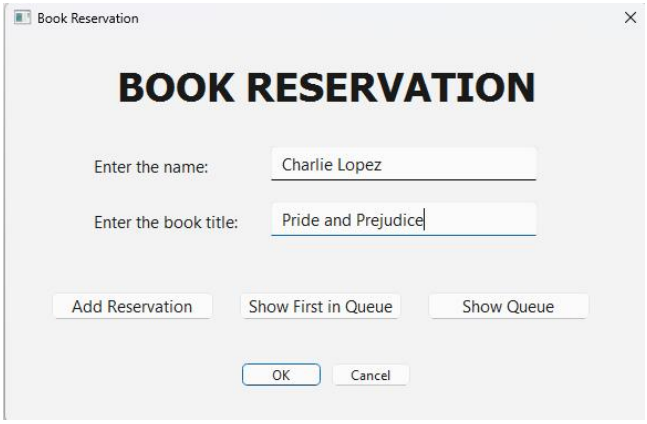


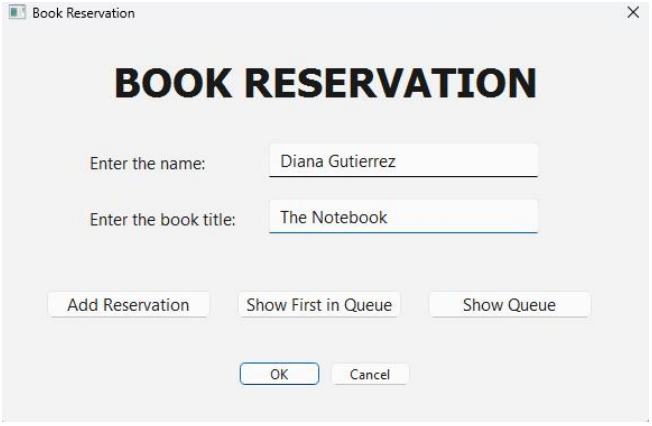
Expected Result: Peek displays the book named ("A Walk to Remember") as the top book, together with the person who returned the book (Billy Cruz).

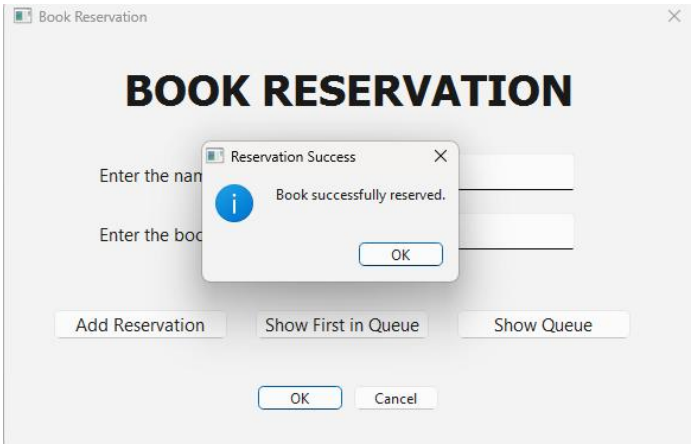
Test Case 2: Enqueue and Dequeue Book Reservation (Queue)

Action: Enqueue "Pride and Prejudice" returned by Charlie Lopez, and "The Notebook" returned by Diana Gutierrez; then dequeue.

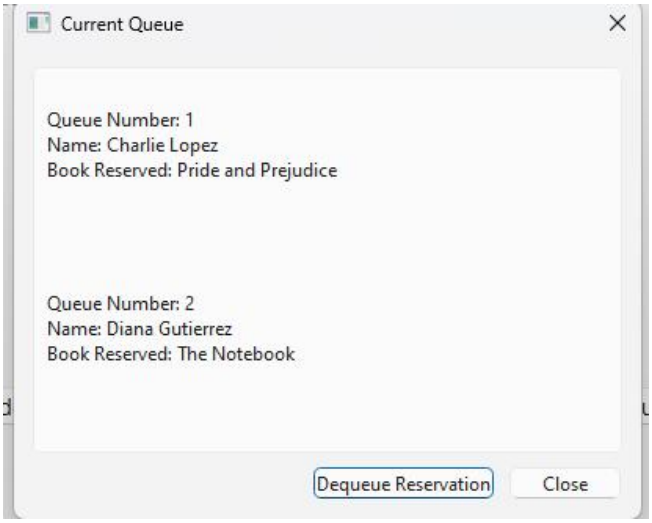
Action: Add Reservation (Enqueue)

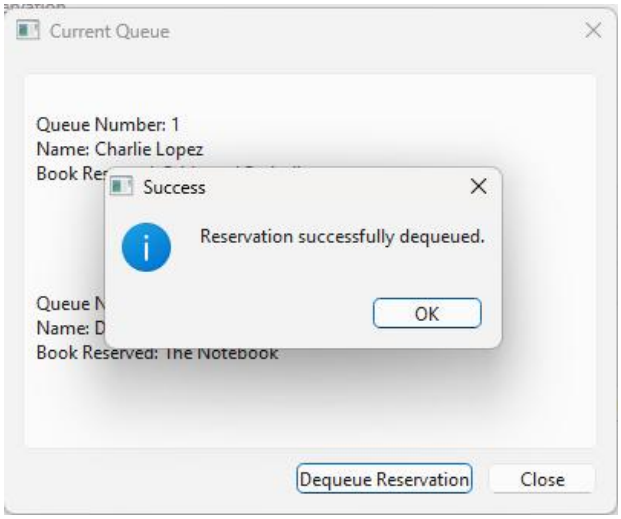




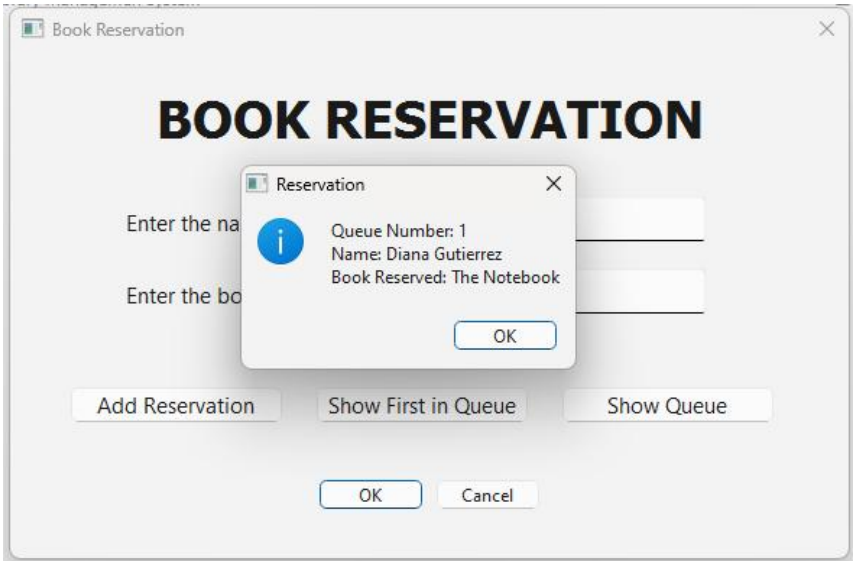


Action: Dequeue Reservation (Dequeue)





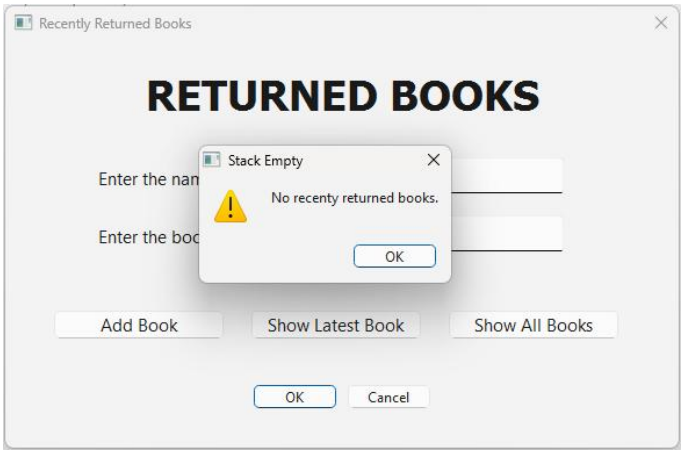
Action: Check the first in line (Peek)



Expected Result: Dequeue removes "Pride and Prejudice" returned by Charlie Lopez, making "The Notebook" returned by Diana Gutierrez next in line.

Test Case 3: Peek on Empty Stack

Action: Peek an empty stack.



Expected Result: Displays a message indicating the stack is empty or there's no recently returned books.

E. Challenges Faced During Development

During the development of our library management system, our team faced two main challenges: learning how to create a GUI using Qt Creator and dealing with time issues because of extracurricular activities.

1. Learning Qt Creator for GUI Development

Creating a graphical user interface (GUI) with Qt Creator was new to us. We knew how to code, but working with the platform's tools and layouts was difficult at first. It took time to figure out how to integrate the stack and queue features into the GUI and make it easy to use. However, after some practice, we became more comfortable with Qt Creator and were able to create a functional interface.

2. Time Issues Due to Extracurricular Activities

Two team members were busy with intramural sports, which affected the time we had to work on the project. Their practices and games often clashed with our project schedule. To manage this, we split the work based on each person's availability and communicated regularly. We also made sure to work extra hours when everyone was available to stay on track.

Despite these challenges, we completed the project successfully. The experience helped us improve our skills in GUI development and time management. We learned how to work together and adapt to problems as a team.

F. Roles of Each Member and their contributions

Each team member contributed to the success of our library management system project in their own way. By using each member's strengths, the team successfully designed a functioning and user-friendly library management system. The following are their contributions:

1. Jana Jane A. Mancenido – Programmer/Coder

- responsible for coding the core functionality of the system, including the stack and queue operations for managing book returns and reservations

- ensured that the system ran smoothly and efficiently

2. Jhoram Ezekiel C. Bala – Designer (UI Design)

- designed the user interface, focusing on creating a clean and intuitive layout

- ensured the design was both functional and visually appealing for the library staff.

3. Sherwin D. Roazol – Designer (Implemented UI in Qt Creator)

- took Jhoram Bala's design and implemented it in Qt Creator

- ensured that all UI elements were correctly placed and integrated the stack and queue features into the interface

4. Luis Truman N. Remolona – Tester

- tested the system, identifying bugs and usability issues

- ensured the system was reliable and functioned as intended