**Challenges Faced**

1. **UI Glitches**: Initial issues with layout adjustments caused misalignment of booking interface components. These were resolved by refining JavaFX settings.
2. **Error Handling**: Early versions lacked comprehensive error messages for invalid bookings and incorrect user inputs. Improved validation logic and exception handling addressed this.
3. **Data Synchronization**: Delays in updating booking information caused inconsistencies between the UI and backend data. This was resolved by refining the logic for booking updates.

**Design and Implementation Choices**

* Adopted a modular architecture separating UI, logic, and data management.
* Used JavaFX for building an intuitive and interactive graphical user interface.
* Chose lightweight and efficient data structures like HashMap and ArrayList to manage application data effectively.
* Integrated a custom doubly linked list to demonstrate understanding of custom data structures and improve navigation.

**Learning Outcomes**

1. Gained experience in choosing appropriate data structures based on application requirements.
2. Developed problem-solving skills for debugging and resolving UI and logic issues.
3. Enhanced understanding of the importance of modular design for maintainability and scalability.
4. Improved ability to write clean, efficient, and well-documented code.