

Wu Yujin's Project Portfolio Page

Project: ConciergeBook

ConciergeBook is a **desktop app for hotel receptionists to efficiently manage guest bookings via a Command Line Interface (CLI)** while still having the benefits of a Graphical User Interface (GUI). If you can type fast, ConciergeBook can help you optimise how you manage your rooms, your guests and all new and existing bookings - faster than traditional GUI apps.

- **Code contributed:** [RepoSense link](#)

Given below are my contributions to the project.

- **New Feature:** Add the ability to find Booking with different parameters [#68](#)
 - What it does: allows the user to find booking(s) with specific room ID, person ID, start date, end date and isArchived state.
 - Justification: This feature improves the product significantly because a user need this command to find the booking when he/she only has partial information about the booking, e.g. to find out which room a customer is allocated to within the start date and end date period.
 - Highlights: This enhancement takes multiple parameters, and they can also be optional. The implementation also involves writing multiple predicates and compressing multiple predicates into one in the command level to feed into the filteredList in model.
- **New Feature:** Add the ability to delete booking [#89](#) [#109](#)
 - What it does: allows the user to delete the booking permanently with the given booking ID. This is different from archiving as it wipes out the entry from the database entirely.
 - Justification: This feature is essential for the users to manage bookings.
- **New Feature:** Implement the Booking class [#50](#)
 - What it does: The Booking class keeps information about a booking in the form of person ID, room ID, start date and end date and archive state.
 - Justification: It is essential to the app's main purpose: manage bookings.
 - Highlights: This enhancement include change to both model and storage, which require in-depth knowledge about the programme flow. It is also closely related to the Person and Room class.
- **Contribution to team-based task:** Implement BookingBook [#58](#)
 - Modify model and storage

- Add tests for JsonAdaptedBooking, JsonBookingBookStorage [#136](#)
- **Contribution to team-based task:** Refactor addressBook to bookingBook [#199](#)
 - Rename occurrence in relevant code and .json files
- **Enhancements to existing features:**
 - Modify delete person feature so that when a person is deleted, all the related booking(s) are deleted (cascading delete). [#04](#)
- **Documentation:**
 - Developer Guide:
 - Add the section on findBooking, deleteBooking and Booking class.
 - Add the UML sequence diagram for findBooking and deleteBooking.
 - User Guide:
 - Add the section on findBooking and deleteBooking command.