Project 2: Proposal

Group 18

Kevin Wang, Victor Wu, Zhen Yu Trevor Tong, Xinrui Lu

University of Toronto – Mississauga

CCT211H5F: Fundamentals of User Interface Programming

Professor Michael Nixon

Date: October 27, 2025

Project 2: Persistent Form

Github Link: https://github.com/Veco117/Group-18-Hotel-Room-Booking-Page.git

Project idea: We are developing a simple app interface using Python and Tkinter framework so that users can book a hotel room. At the opening, the welcome screen says, "Welcome to TVXK hotel" and shows a button "Book your room now". If users click on it, there are plenty of steps that users could follow along with. Selection of date and number of guests, room options, result view, filling information of the guests, review of cost and payments, confirmation and codes and user could view/edit/cancel the reservation with their confirmation code and last name by clicking on "Manage my booking" button at the beginning. Also, the data is stored to local files, so it's persistent.

Tentative approach: For the **selection of dates**, we plan to implement an elegant pop-up calendar interface, enabling users to drag to select their check-in dates and length of stay. For the **selection of guests**, we are thinking of creating a spinbox letting users select the number of guests. For **room preferences and filters**, we will use stylish radio buttons for users to select room types, such as single, twin, or king etc. A check button controls whether breakfast is included or not, and finally, two entry boxes for users to select price ranges. After the filters are set by users, clicking on the **search button** will smoothly transition to the result page. Here will show a listbox of available rooms that match the filter. Behind the scenes, the systems will cross-check existing bookings in the database to ensure real-time availability. Once the room is chosen, users are **required to fill in their essential info** such as name, email address and phone number by a nice-looking entry widget. And we will create an **error check**, for example, if users enter numbers in their names. The system will display an error message. Once the user finished filling their info and clicked the "confirm booking", it would turn to the nice-looking **summary page**, which would include the room you have selected, price, date and guests, etc…. Once users check everything, users now have 2 options: either go back or continue to pay. If the user clicks on "to pay", it will go to the **payment pages**, asking the user to enter their 16-digit card number, CVV, and the date of the card. We will also create an **error check** here if the user types the alphabet instead of numbers. Once payment is complete, systems will generate a **unique confirmation code** using uuid. The app stores all booking data securely in a JSON file and reassures the user with a friendly message: "Your stay at TVXK hotel is confirmed! Please save your confirmation code in case you want to make an adjustment :)". Now users will have two options: either stay on this page or go back to the home menu. Back on the home pages, there was a button **"Manage my booking"**, which would allow users to view/edit/cancel their reservation by entering their confirmation code and last name. It will show in a neatly formatted summary view. If the user clicks on the **edit button**, it will reopen the familiar form for easy updates, such as rechecking date availability, room type. So basically, going over the process. If the user clicks on the **cancel button**, it will show a message: "We look forward to seeing you

soon!". These options complete the full CRUD cycle and a well-organized business system, bringing wonderful experiences to customers.

Possible technologies and libraries of interest: Matplotlib, Pandas, Mintab, uuid

Date selection: Check in - Check out, no of pax.: Adults (> 12) & Children (<= 12)

Room requests: King or double beds, ascending/descending order of price range, cancellation refundable/chargeable for a fee, pets allowed/not allowed special needs (e.g. accessible rooms, rollaway beds, cribs etc.)

Filter results: check remaining room types based on needs and/or preferences (e.g. smoking/non-smoking)

Filling up necessary information: Value-added services, such as including/excluding breakfast, high/low floor, and proximity to the elevator.

Price summary: Include the room itself as well as add-ons like breakfast, rollaway beds, and tax

Manage stay: Cancel order, amend dates, room, or services (e.g. breakfast)