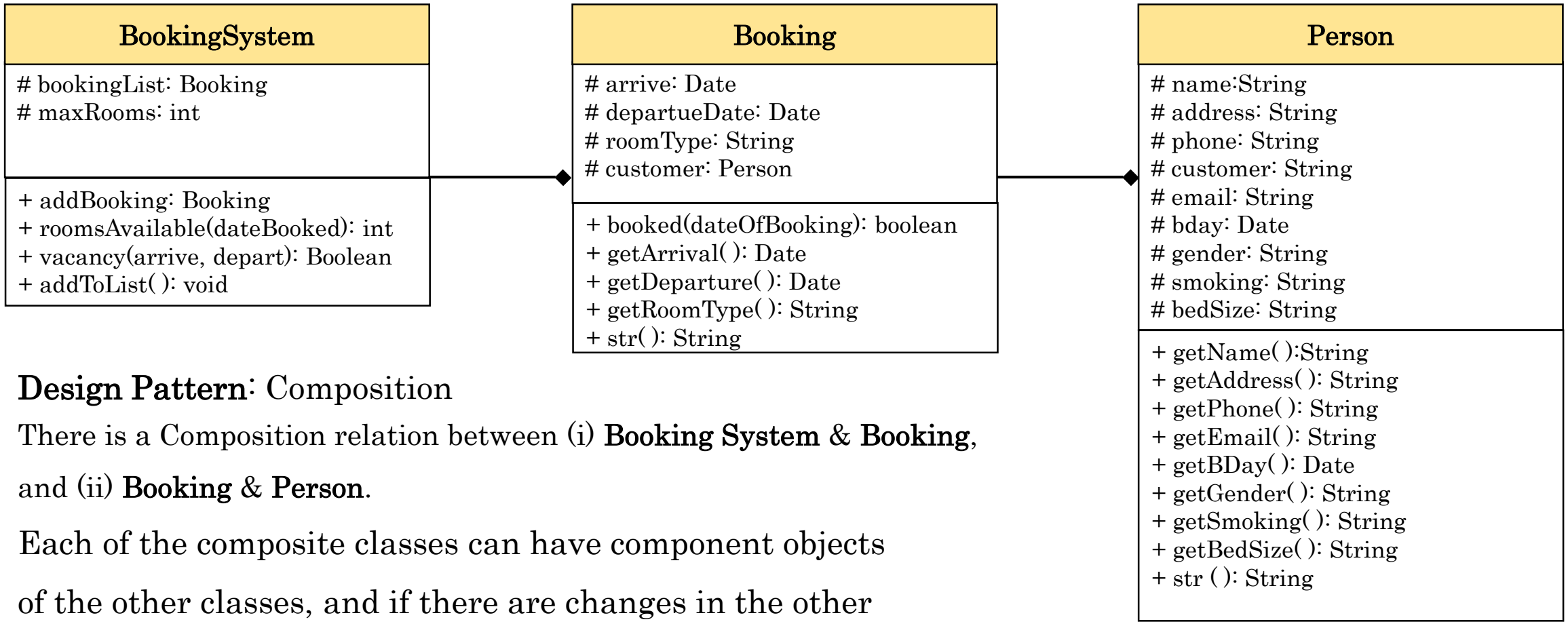
Booking App: Slides

Booking App's: Unified Modeling Language (UML)



Design Pattern: Composition

There is a Composition relation between (i) **Booking System** & **Booking**, and (ii) **Booking** & **Person**.

Each of the composite classes can have component objects of the other classes, and if there are changes in the other it won't matter as these classes are loosely coupled (non dependent on each other)

Booking App's: Class Definition/Design

Booking System class (keeps Booking instances), the member functions are intended to do the following:

addBooking( ) takes as parameters (i) arrival and departure dates, (ii) **Person instance** as well as (iii) **roomType** variable. It first checks to verify if there is a vacancy for the requested period by calling **vacancy( )**, if so, constructs a **Booking instance** & appends it to the list (only if that booking is successfully saved to the Booking database's bookingList table) .

vacancy( ) on the other hand takes two dates as parameters (arrival & departure dates), tests if the dates are valid first, arrival is before departure, and if so, checks whether there is at least one room available on each of the dates from the (i) arrival to the (ii) departure date (by call roomsAvailable)

roomsAvailable( ) takes a specific date as a parameter and returns the number of unoccupied rooms on that specified date. It accomplishes this by going through the booking list, checking how many of them include the given date (and decrements this from maxRoom variable/number of rooms)

Booking App's: Class Definition/Design

Booking class (keeps Person instances as Components), its member functions are intended to do the following:

- **booked( )** method is used in conjunction with the Booking System mechanisms in checking whether Booking instances have booked a room on particular day with the objective of determining the maximum number of rooms on a specific dates (per specific day)
- **str( )** simply returns a string comprising the values of its data members (i.e. arrival and departure dates in a certain format DD-MM-YYYY, room type - e.g Suite or Super Luxury or any other room type chosen by customer assigned to that Booking instance)

Booking App's: Application Logic

- The User has to sign-in, to use the booking App, and the login credentials are store in a file (authentication details in textfile - included with the source code files in a separate folder)
- The **Booking System class** not only keeps a **list** of **bookings**, but also manages the addition or non-addition of bookings depending if rooms are full (maxRooms static variable is used to limit bookings for a particular day – if rooms reach the limit a notification is sent to the user of the application notifying them that rooms are fully booked)
- Ordinary rooms cost R1000, Super Deluxe = R1200, Super Luxury = R1300, Suite = R1500
- totalBill/Cost: is determined by number of Days Staying **x** roomType
- User verification mechanisms are in place to ensure the end user enters valid inputs and only when the inputs are correct can the addition of a booking be made
- The application connects to DB Browser SQLite to store the successfully created bookings permanently (see SubWindow 2: makeReservation slides) and these records can be retrieved and viewed in the viewReservations subWindow (subWindow 3)

Booking App's: Application Logic II

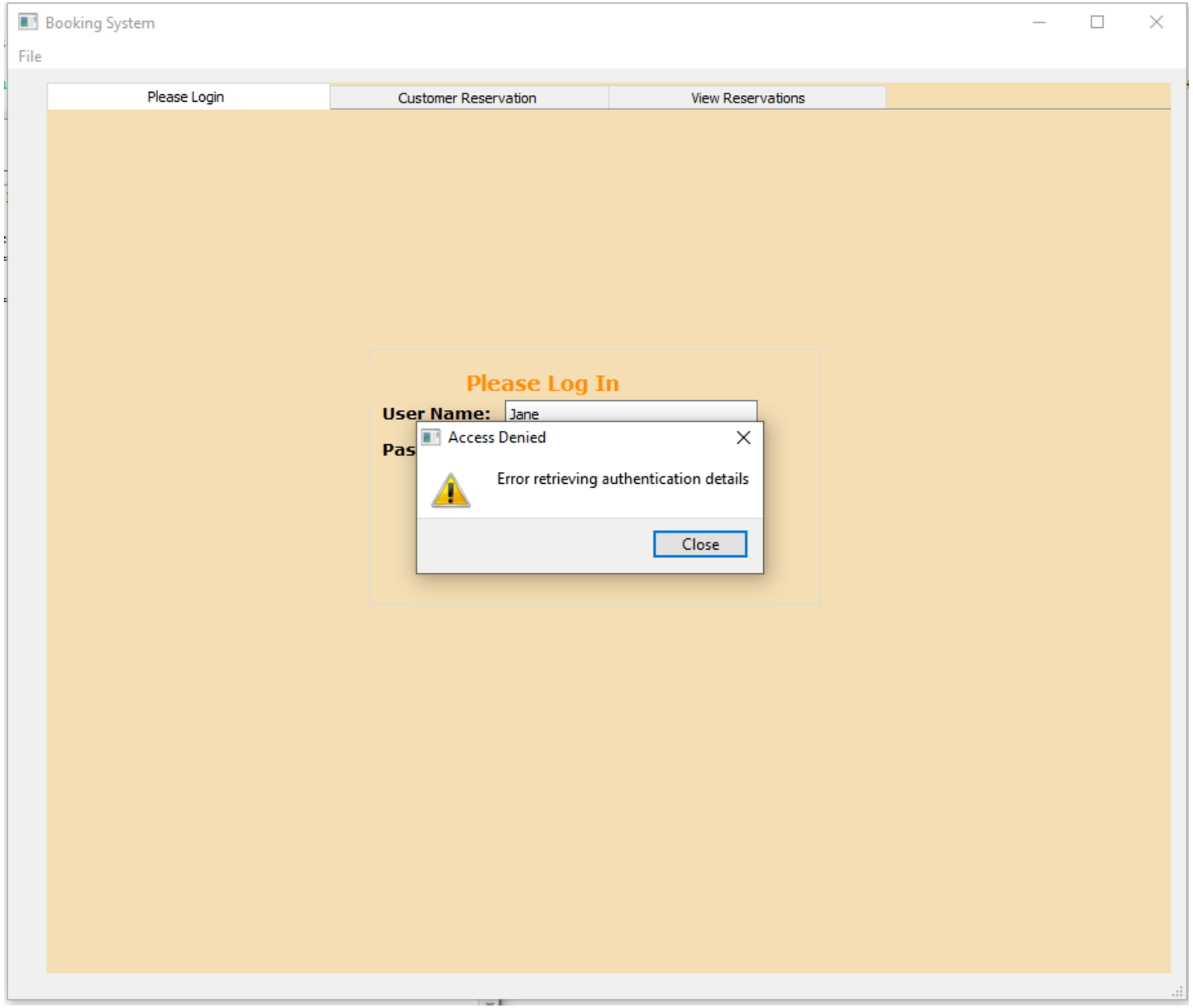
- The application temporarily creates a Booking instance in the User Interface Section (client program: main), appending the Booking instance to the Booking System's booking list only when the end user permanently stores (commits) the record to the Bookings.db (i.e. bookingList table).
- As records (bookings) are continuously committed, we only append the successful ones to the Booking System's static list (bookingList),only when they have actually been added to the database table
- A static variable maxRooms (maximum rooms available for a single day) is used as a safe guard to ensure bookings are made dependent on number of rooms available.

Booking App's: Business Rules

- The Guest House has a fixed number of rooms, all with single beds only. It is up to the customer which type of bed size they prefer (i.e. Queen, King, Twin etcetera)
- The rate for the night/days staying is determined by the room Type (e.g. how the room is arranged - luxury linen/bedding, foot massage machines could be found in a Super Luxury room, while an Ordinary room might not have luxury linen/machines, but have basic linen/no massage machine instead)
- There is an age restriction (e.g. booking must be made by a person older than 18 years, and person born after 2004 are not allowed to make bookings – see screen shots in the **Exceptions Handling** screen shot slides

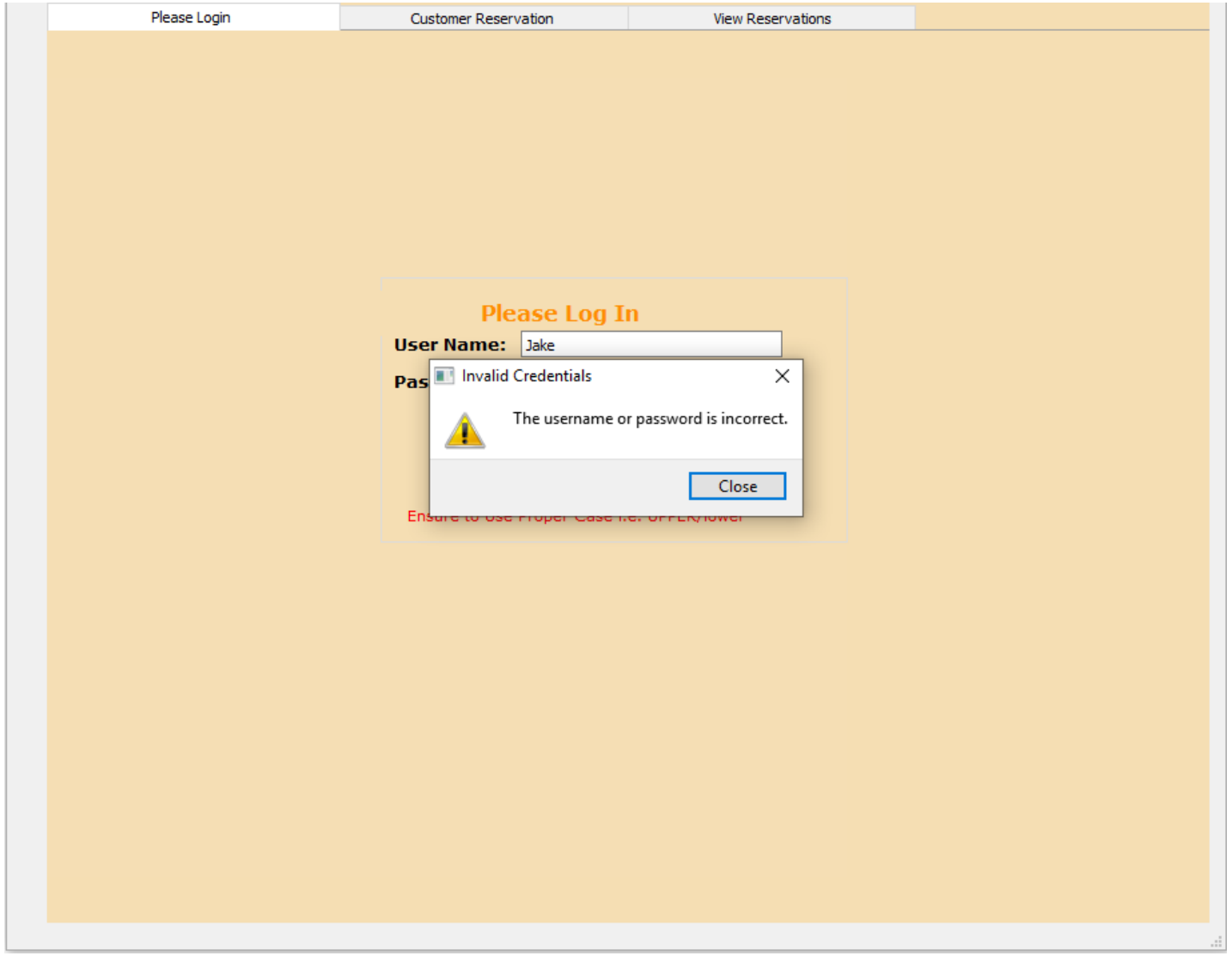
User Authentication: Login Screen

User Authentication: SubWindow



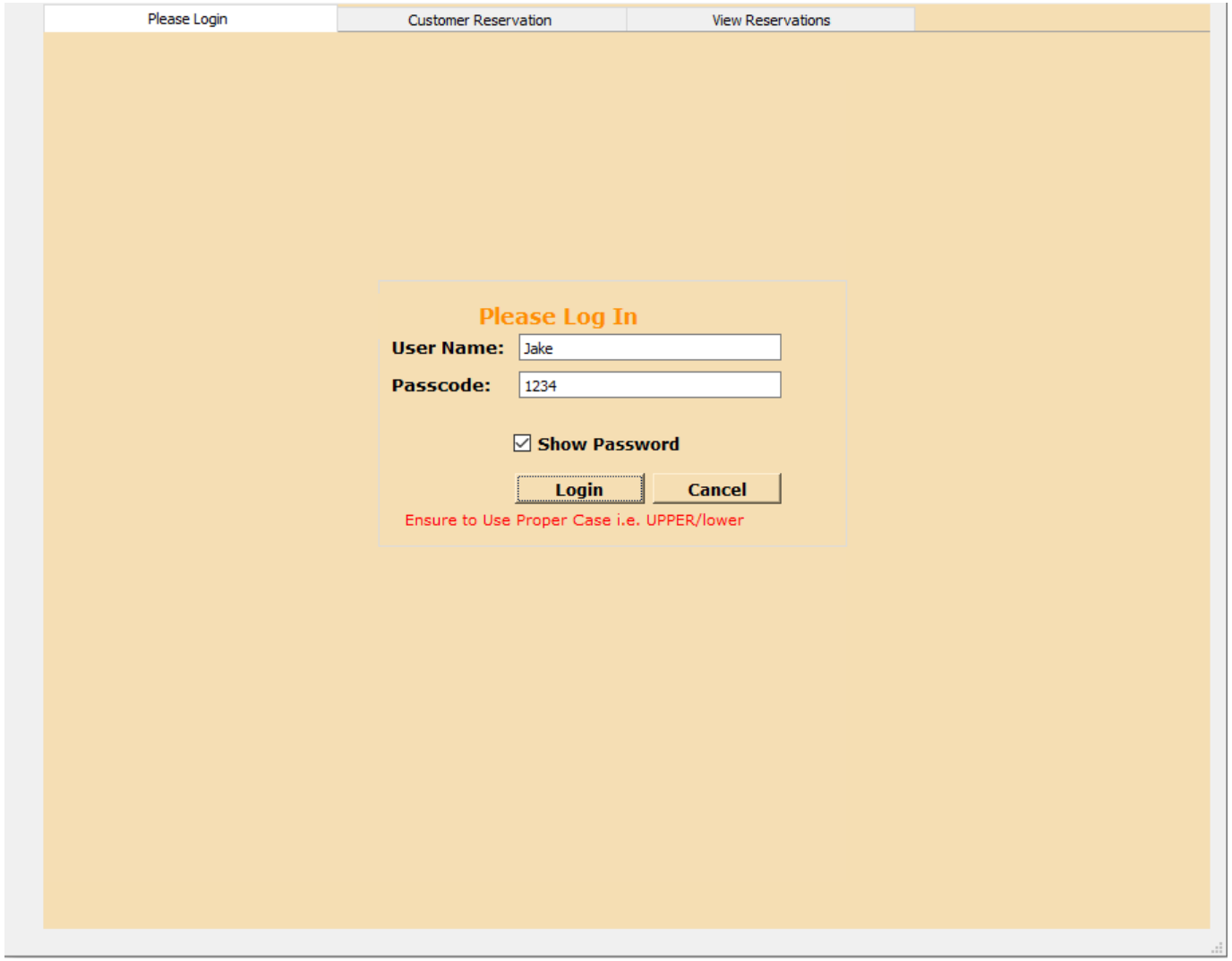
Application has built in mechanisms to ensure user login is undertaken. If for example due to resource unavailability such as a missing authentication file, the program terminates and informs the user of the reason for termination

User Authentication: SubWindow



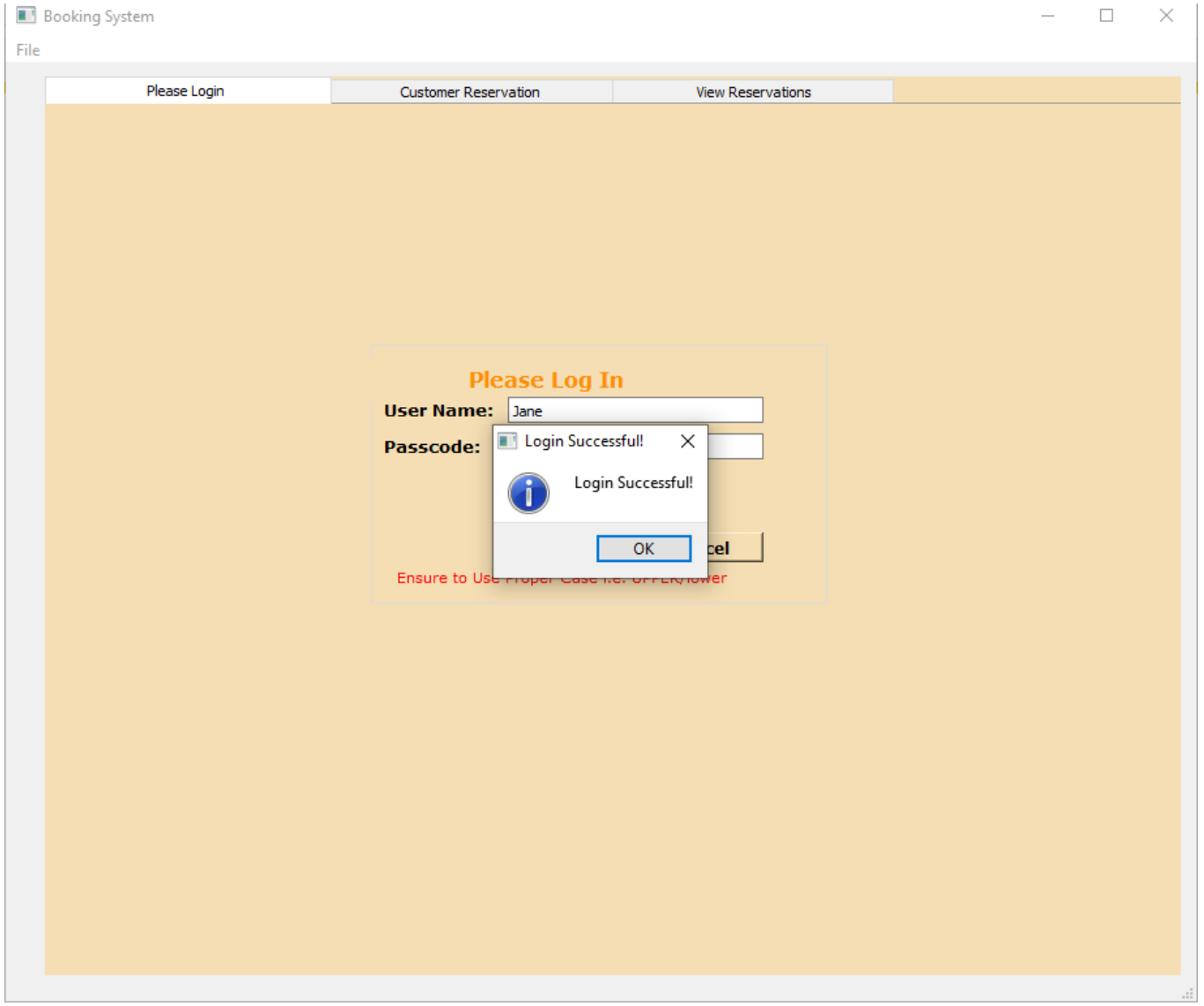
Application provides defensive mechanisms against unauthorized users. App validates user details by comparing end user's entered user name and passcode to those store in file named **authUsers1.txt** .
If the credentials are incorrect a message is shown

User Authentication: SubWindow



Application provides clues for end users On how to avoid data entry errors during login attempt when keying in credentials (i.e. use proper letter cases: upper case or lower case)

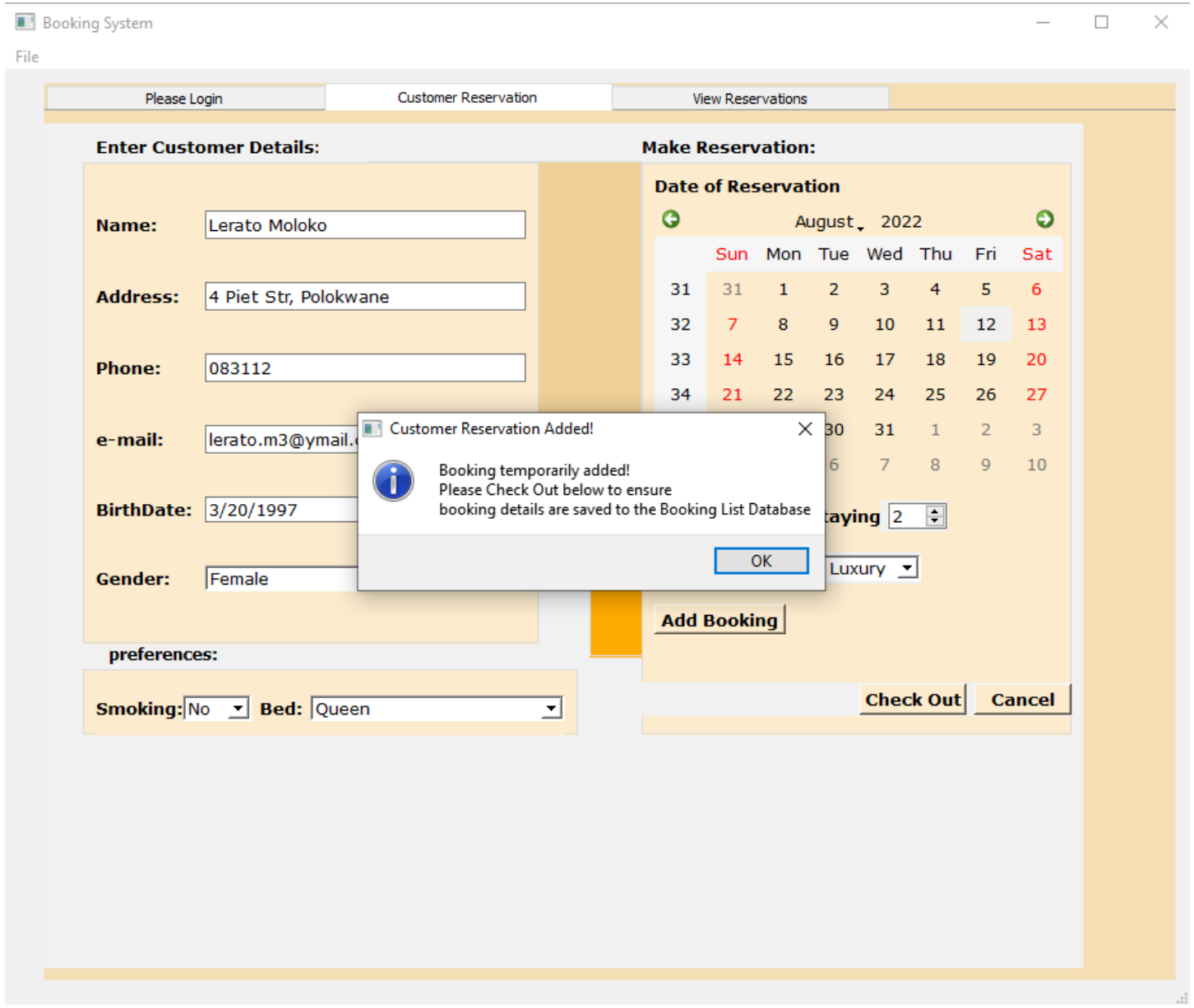
User Authentication: SubWindow



If the credentials are correct the application notifies the user of a successful log in, enabling user to access the next subwindow(s)

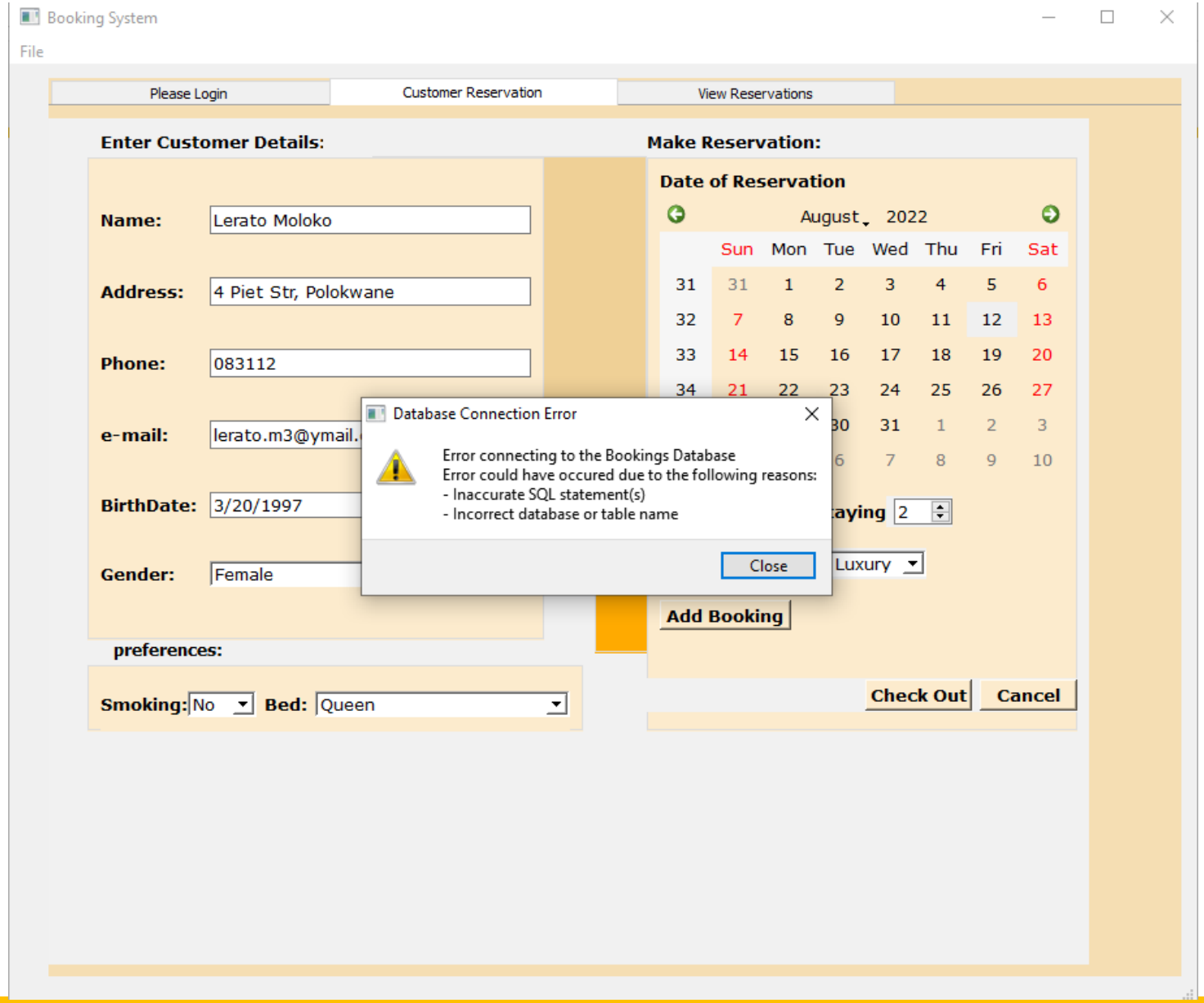
Customer Reservations: Screen Shots

Customer Reservation: SubWindow



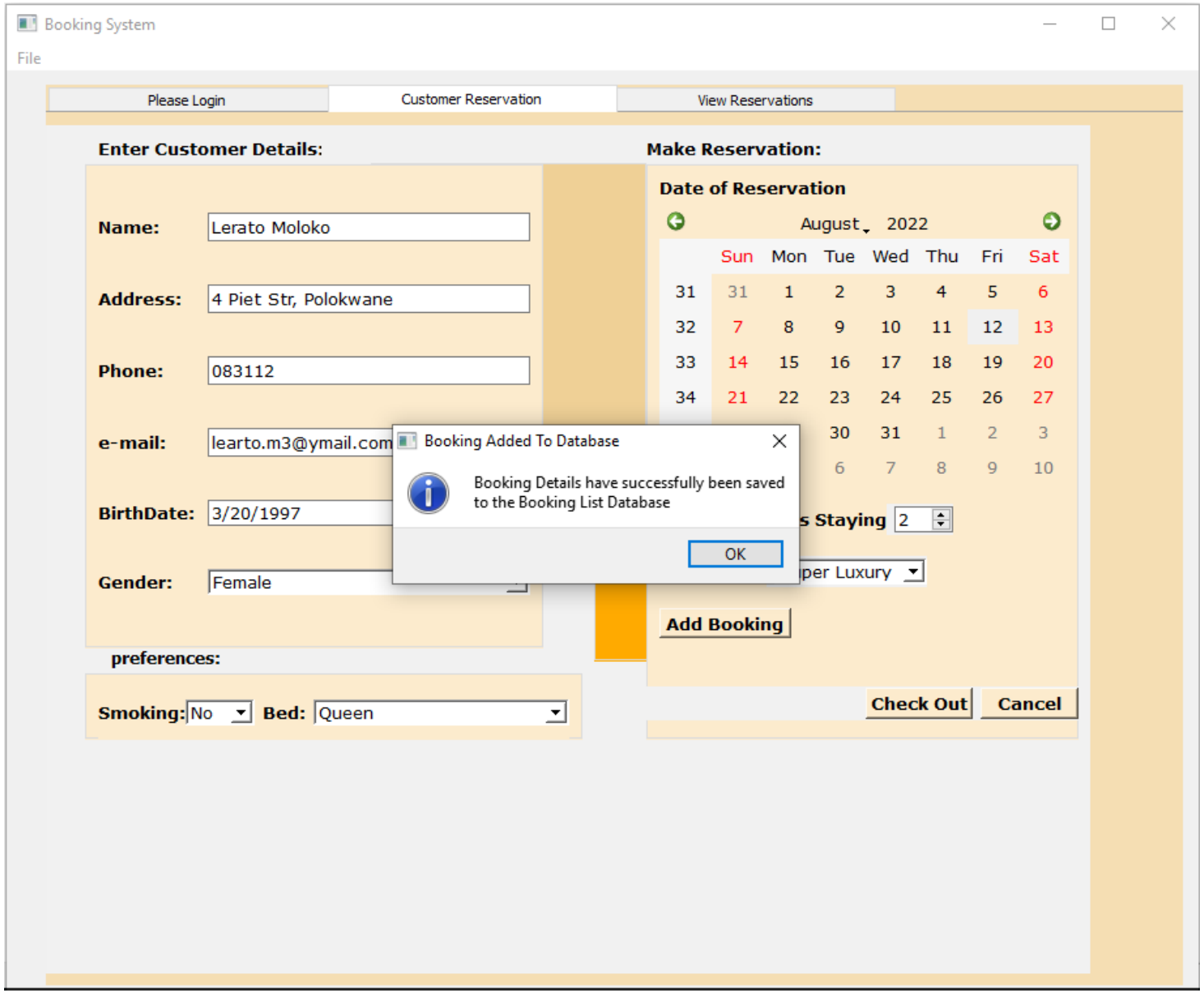
Booking is temporarily stored, once the application has successfully connected to the **Bookings** database the booking will be committed / saved to the **bookingList** table (depending on accuracy of SQL statement)

Customer Reservation: SubWindow



If connection to the **Bookings** database is unsuccessful or there is an inaccurate SQL statement in the application, the application notifies the user of possible errors signaling to the user that the booking was not added to the database

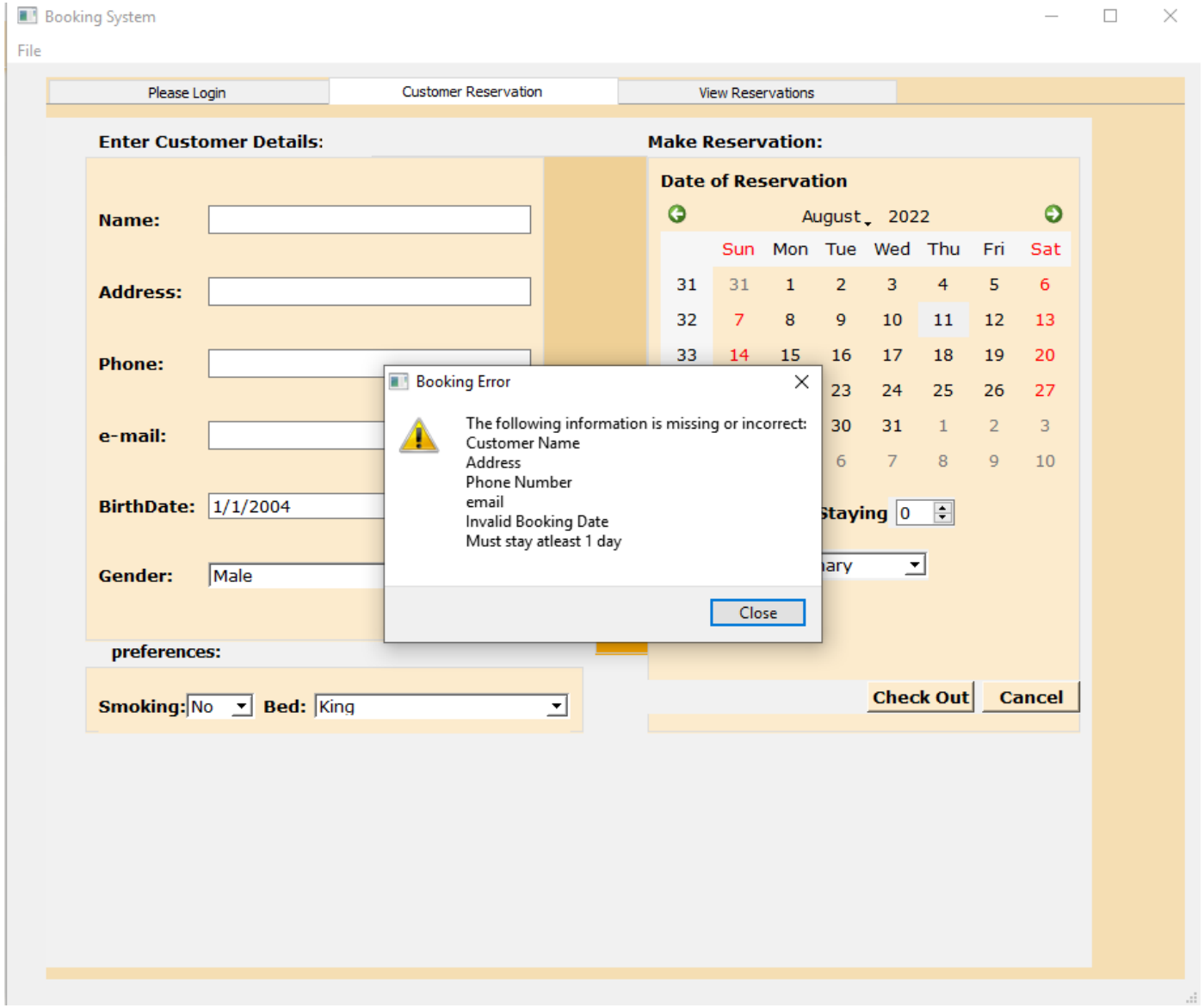
Customer Reservation: SubWindow



Once there is a successful connection to the **Bookings** database and the accuracy of SQL statement is valid application notifies the user of the booking being successfully committed / saved to the **bookingList** table

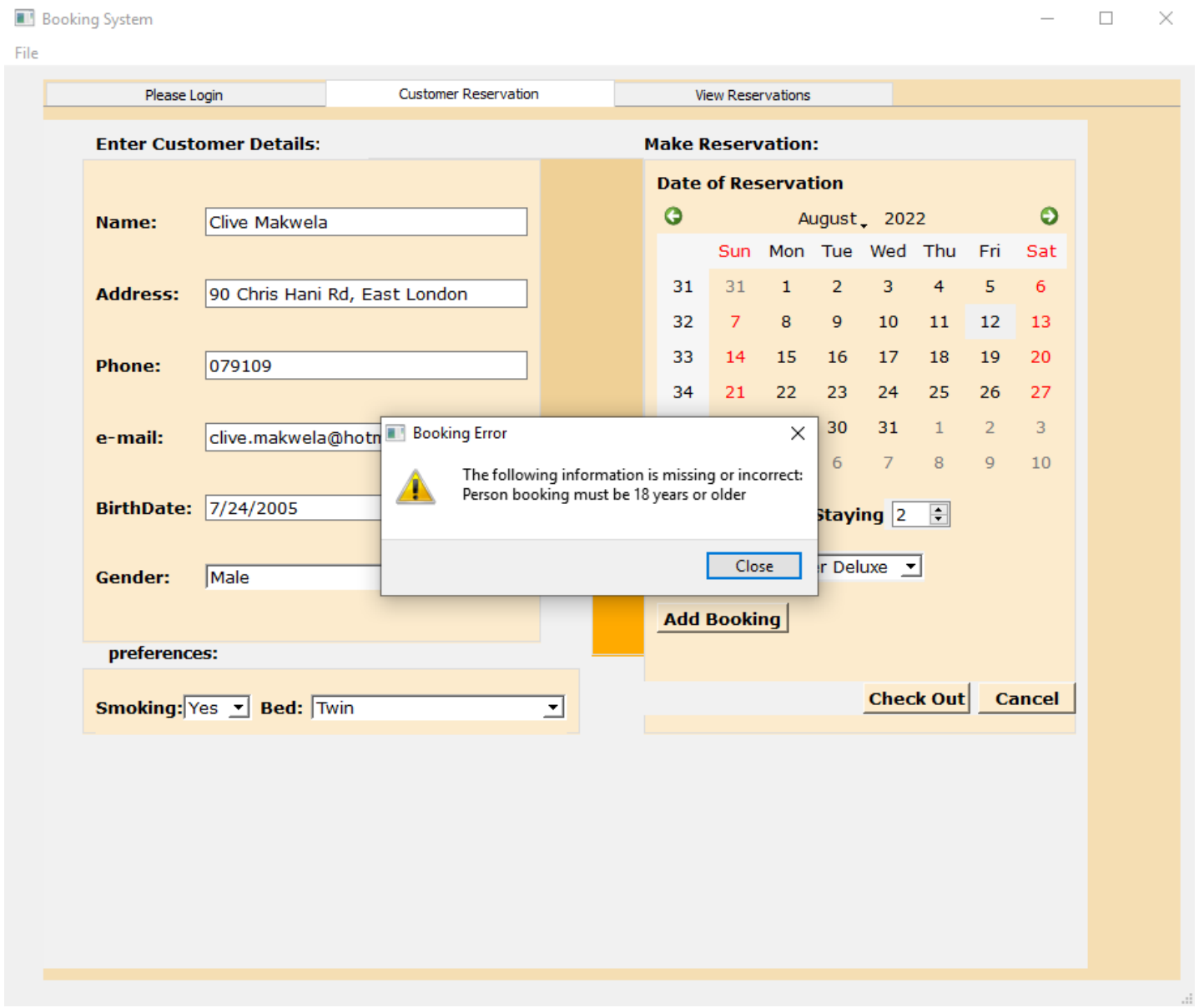
Customer Reservations: Business Rules Adherence

Customer Reservation: SubWindow (EXCEPTION HANDLING)



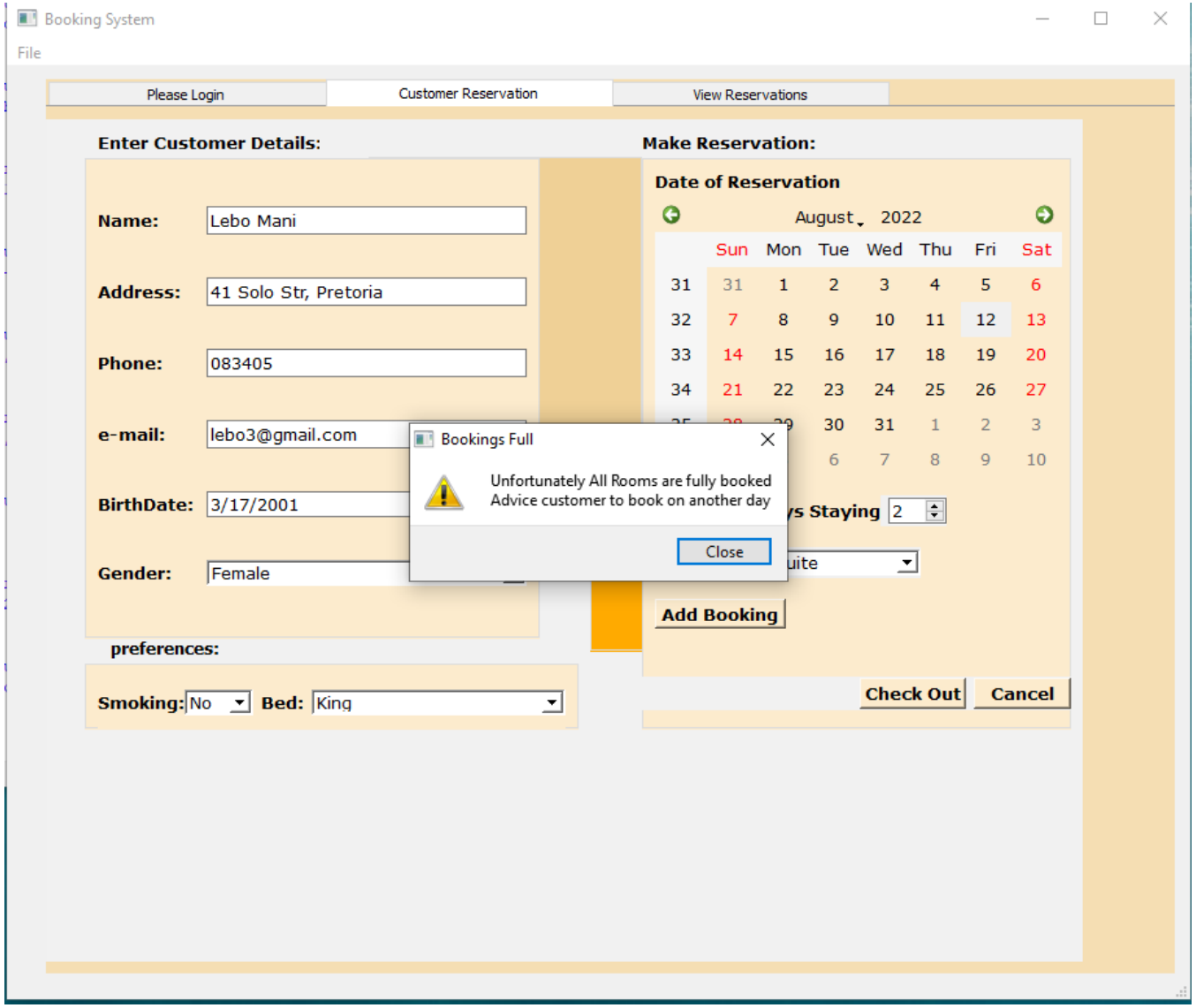
Customer Reservation subwindow has error validation and checking mechanisms in place to detect erroneous end user inputs. Various blank inputs and non selections are checked to see if user has supplied valid inputs and if not a error message is invoked informing user of missing or incorrect data

Customer Reservation: SubWindow (EXCEPTION HANDLING)



Apart from handling erroneous inputs, a business rule that an underage customer making a booking is not allowed can be enforced by the application, thus rejecting the reservation

Customer Reservation: SubWindow (EXCEPTION HANDLING)



When rooms are fully occupied the application informs the end user of the failure to make the booking

Customer Reservation: Console/Shell Out (EXCEPTION HANDLING)

Output from testDrive.py

```
>>>
1
Booking Number  1  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Super Deluxe

2
Booking Number  2  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Deluxe

3
Booking Number  3  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Ordinary

4
Booking Number  4  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Suite

5
Booking Number  5  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Super Luxury

6
Booking Number  6  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Ordinary

7
Booking Number  7  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Suite

8
Booking Number  8  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Super Luxury

9
Booking Number  9  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Ordinary

10
Booking Number  10  added, details are:
Arrival date: 12-08-2022
Days staying: 14-08-2022
Room type: Ordinary

Booking Failed
>>>
```

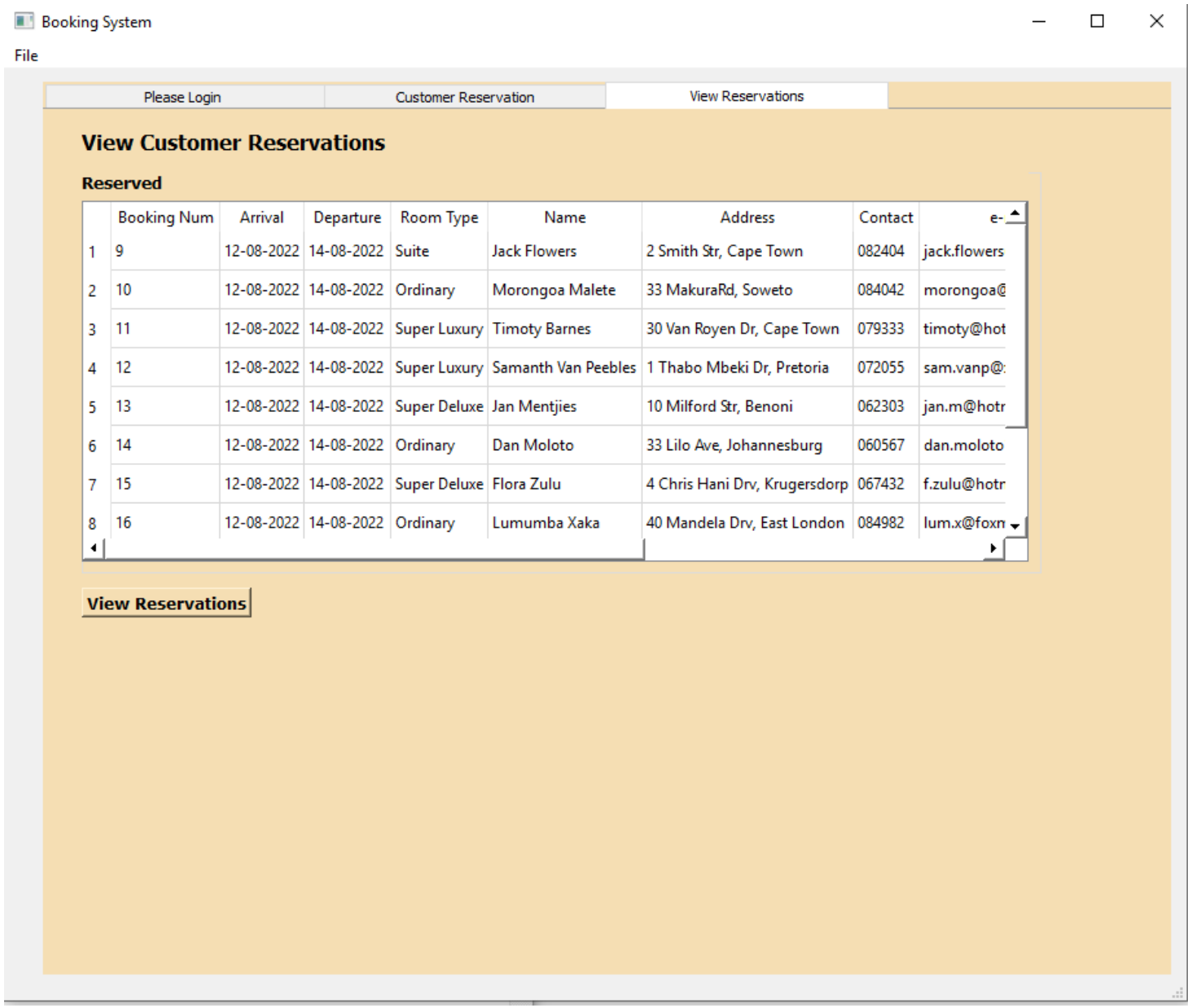
Using the shell/console output: we could create a console based application (see **testDrive.py**) to check the Booking System logic of verifying if the system will reject over-bookings from occurring (e.g. if maxRoom is set to 10 rooms, and 11 customers attempt to book for 2 days stays)

See output on the left

The 11th booking fails as it returns an error message: "Booking Failed"

View Reservations: Screen Shots

These Records are retrieved from the Database,

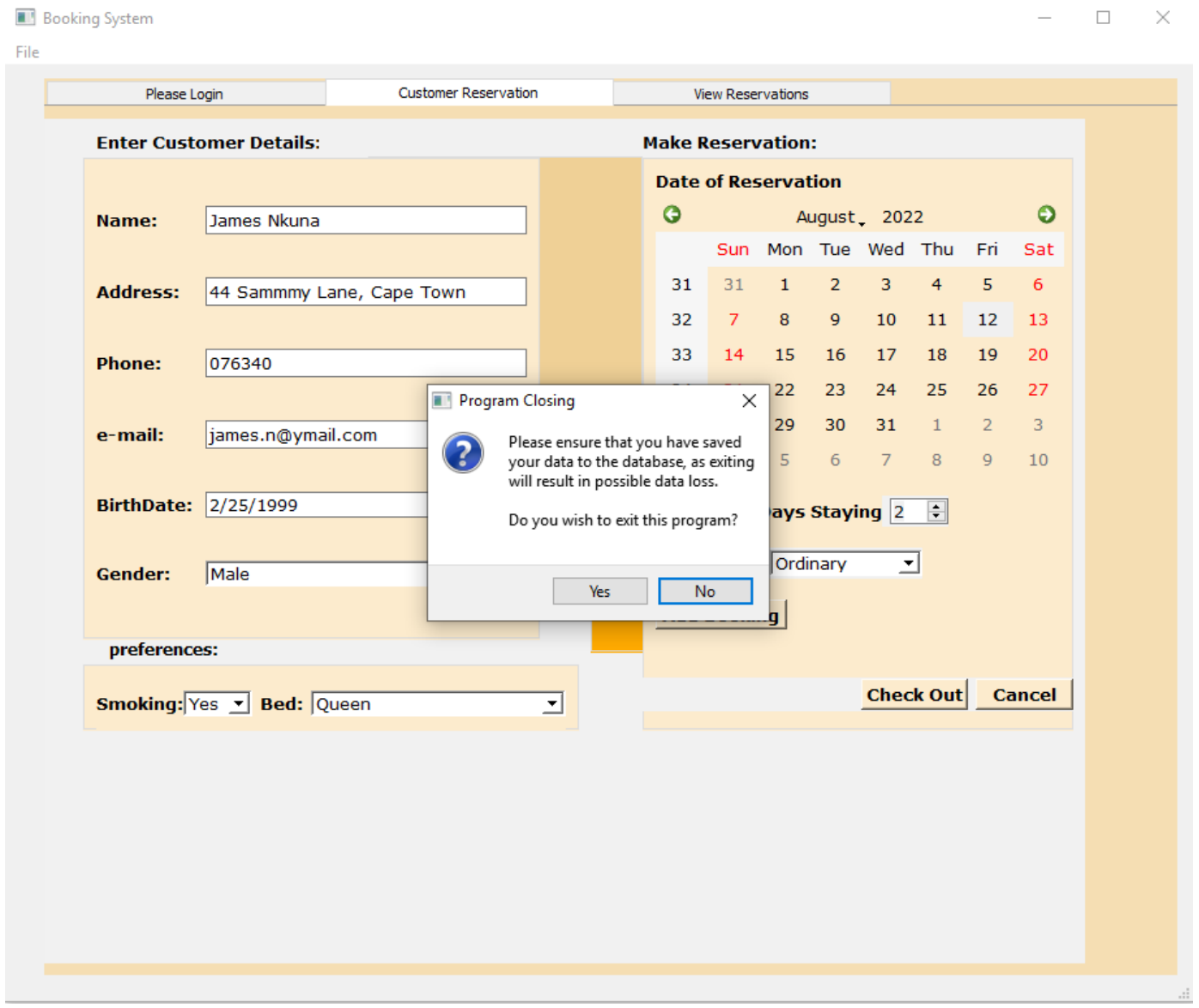


	Booking Num	Arrival	Departure	Room Type	Name	Address	Contact	e-
1	9	12-08-2022	14-08-2022	Suite	Jack Flowers	2 Smith Str, Cape Town	082404	jack.flowers
2	10	12-08-2022	14-08-2022	Ordinary	Morongoa Malete	33 MakuraRd, Soweto	084042	morongoa@
3	11	12-08-2022	14-08-2022	Super Luxury	Timoty Barnes	30 Van Royen Dr, Cape Town	079333	timoty@hot
4	12	12-08-2022	14-08-2022	Super Luxury	Samanth Van Peebles	1 Thabo Mbeki Dr, Pretoria	072055	sam.vanp@:
5	13	12-08-2022	14-08-2022	Super Deluxe	Jan Mentjies	10 Milford Str, Benoni	062303	jan.m@hotr
6	14	12-08-2022	14-08-2022	Ordinary	Dan Moloto	33 Lilo Ave, Johannesburg	060567	dan.moloto
7	15	12-08-2022	14-08-2022	Super Deluxe	Flora Zulu	4 Chris Hani Drv, Krugersdorp	067432	f.zulu@hotr
8	16	12-08-2022	14-08-2022	Ordinary	Lumumba Xaka	40 Mandela Drv, East London	084982	lum.x@foxn

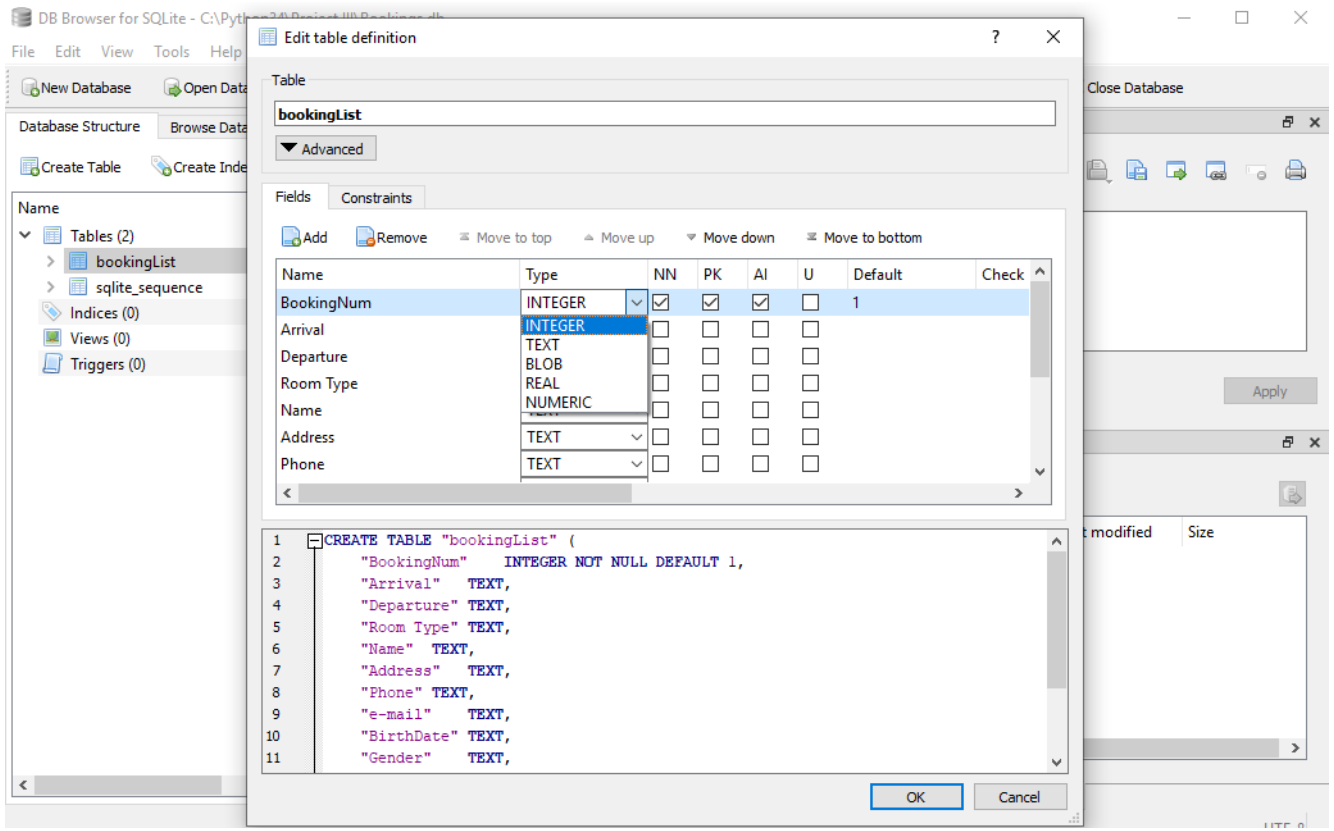
The View Reservation subwindow simply has a feature to view the successfully added Bookings.
It connects to the Bookings database and retrieves (accesses) the records and displays them in a tabular format, where the end user can see the customer who have booked rooms in the Guest House on particular days.

Other: Screen Shots

Safely Exiting Program



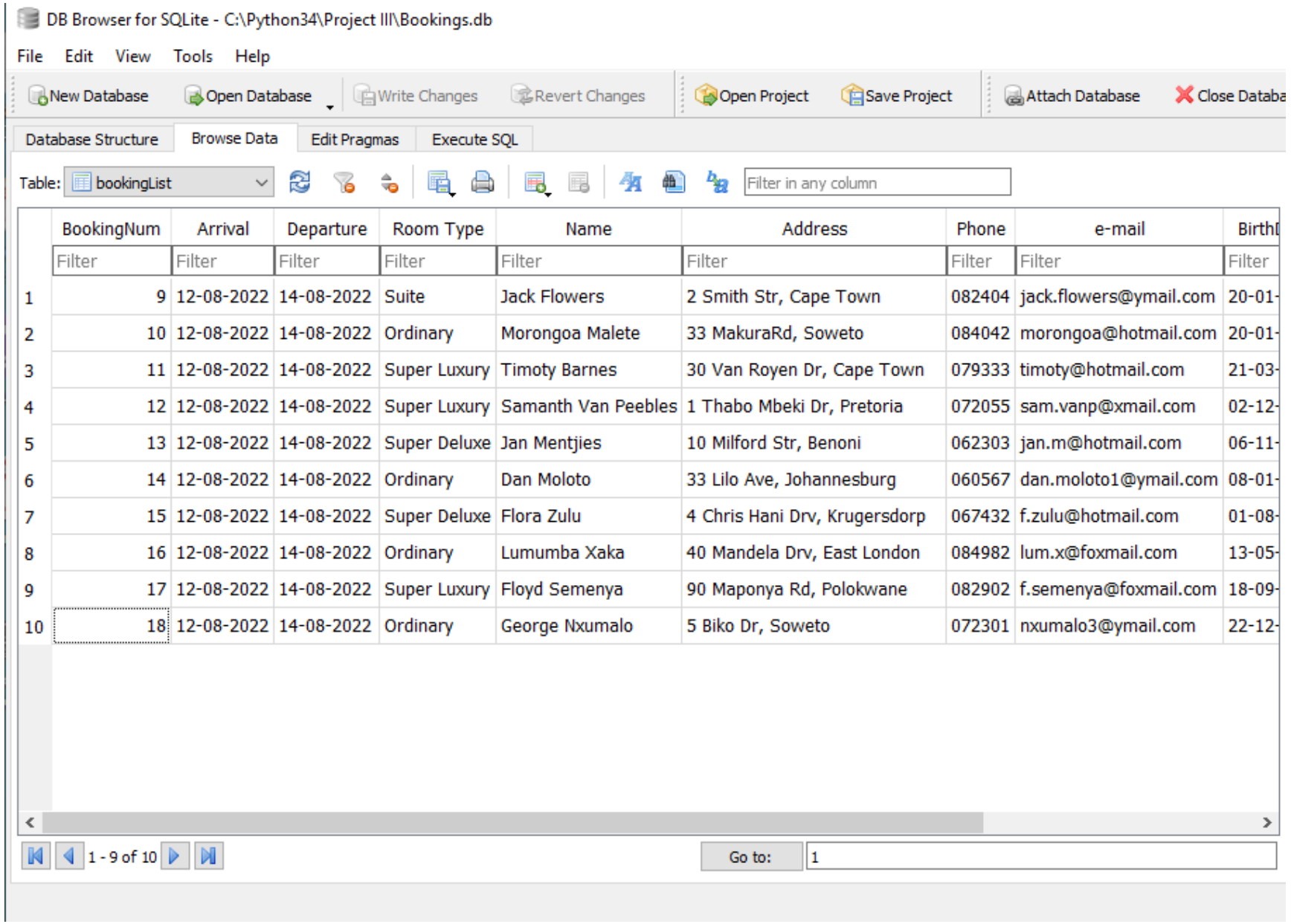
Database Data Types



Please Note that some data types
Such as Date/DateTime, VARCHAR
Are not available. That is why date's
Were converted to strings to and stored
in the table as TEXT data types.

The available Data Types in DB Browser
Are (i) Integer, (ii) Text, (iii) BLOB,
(iv) Real and (v) Numeric

Database stored records


DB Browser for SQLite - C:\Python34\Project III\Bookings.db

File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Databa

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: bookingList | Filter in any column

	BookingNum	Arrival	Departure	Room Type	Name	Address	Phone	e-mail	Birth
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	9	12-08-2022	14-08-2022	Suite	Jack Flowers	2 Smith Str, Cape Town	082404	jack.flowers@ymail.com	20-01-
2	10	12-08-2022	14-08-2022	Ordinary	Morongoa Malete	33 MakuraRd, Soweto	084042	morongoa@hotmail.com	20-01-
3	11	12-08-2022	14-08-2022	Super Luxury	Timoty Barnes	30 Van Royen Dr, Cape Town	079333	timoty@hotmail.com	21-03-
4	12	12-08-2022	14-08-2022	Super Luxury	Samanth Van Peebles	1 Thabo Mbeki Dr, Pretoria	072055	sam.vanp@xmail.com	02-12-
5	13	12-08-2022	14-08-2022	Super Deluxe	Jan Mentjies	10 Milford Str, Benoni	062303	jan.m@hotmail.com	06-11-
6	14	12-08-2022	14-08-2022	Ordinary	Dan Moloto	33 Lilo Ave, Johannesburg	060567	dan.moloto1@ymail.com	08-01-
7	15	12-08-2022	14-08-2022	Super Deluxe	Flora Zulu	4 Chris Hani Drv, Krugersdorp	067432	f.zulu@hotmail.com	01-08-
8	16	12-08-2022	14-08-2022	Ordinary	Lumumba Xaka	40 Mandela Drv, East London	084982	lum.x@foxmail.com	13-05-
9	17	12-08-2022	14-08-2022	Super Luxury	Floyd Semenya	90 Maponya Rd, Polokwane	082902	f.semenya@foxmail.com	18-09-
10	18	12-08-2022	14-08-2022	Ordinary	George Nxumalo	5 Biko Dr, Soweto	072301	nxumalo3@ymail.com	22-12-

1 - 9 of 10 | Go to: 1

See committed/saved records of the bookingList Table which where booked For a specific date

Thank You