

Reservation system

Alex Mecklin 907666, second year automation and robotics, 16.02.2023

General description:

I am making a reservation system for a hotel, where guests can book different kind of rooms for different amounts of time. The customer should be able to add some optional comments when they are booking. The reservation system should have a graphical user-interface and I am aiming for the medium-hard difficulty.

Use case description:

The user-interface should look something like this:

The sketch is titled "Book a room" and is divided into several sections by horizontal lines. The first section is labeled "Type of room:" followed by a rectangular input box. Below this is a box labeled "Calendar view" with the text "choose start and end date from here" inside. The next section is labeled "personal information" and contains three input boxes labeled "Name:", "Email address:", and "phone number:". Below these is a box labeled "Additional comments:" followed by another input box. The next section is labeled "Check reservations" and contains two rows of input fields. The first row has "By time interval:" followed by a checkbox, "start date:" followed by an input box, and "End date:" followed by an input box. The second row has "By guest:" followed by a checkbox, and "Guest name:" followed by an input box. At the bottom is a box labeled "Prints all reservations in the time interval".

Book a room

Type of room:

Calendar view
choose start and end date from here

personal information

Name:

Email address:

phone number:

Additional comments:

Box for printing
output to the user

Check reservations

By time interval: ☐ start date: End date:

By guest: ☐ Guest name:

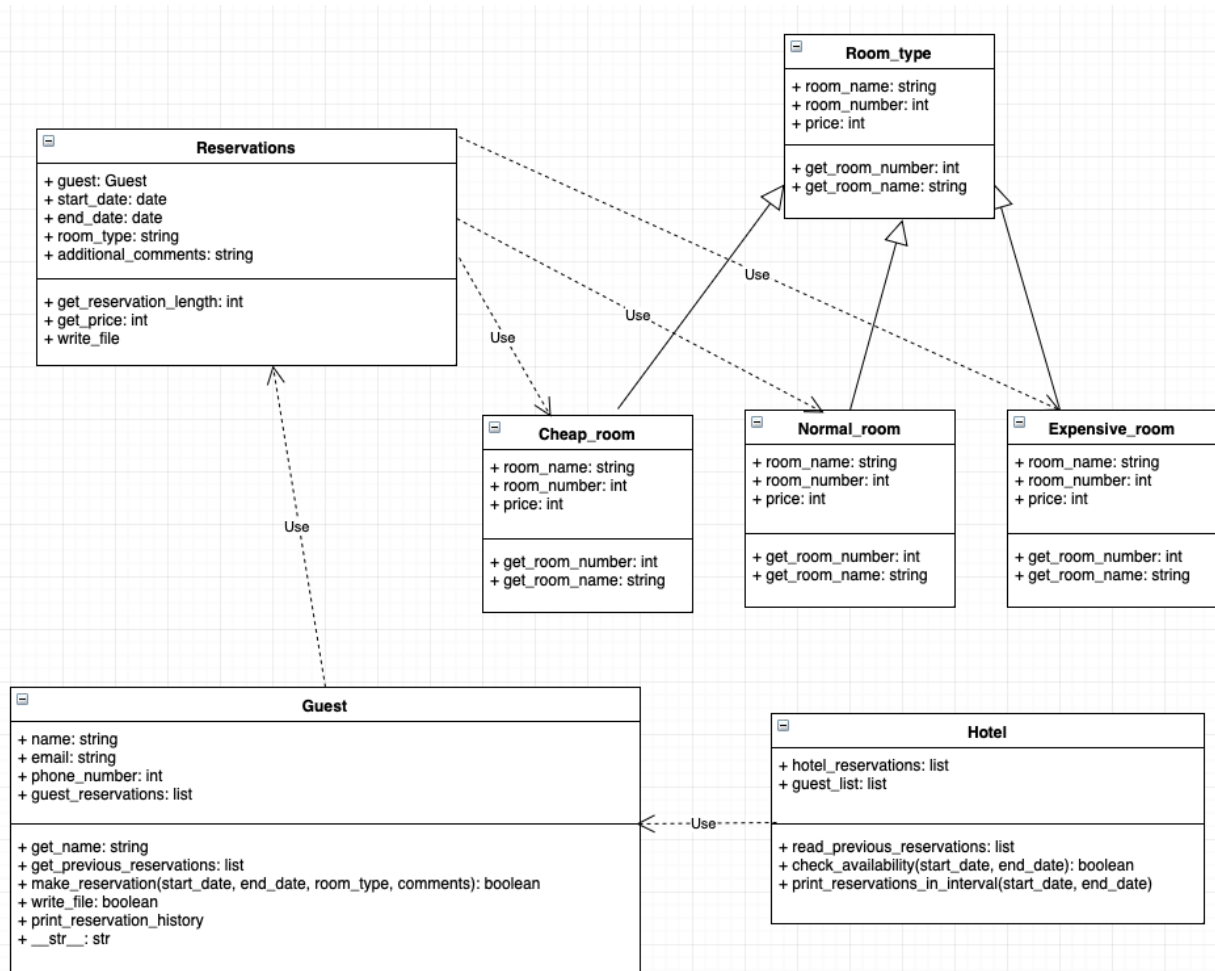
Prints all reservations in the time interval

The user can choose from different room types, and then the dates which when they want to stay at the hotel from the calendar view. If the room is booked in the chosen dates the program should tell that to the user. After everything is filled out the information is saved in a text file and the price of the stay is printed to the user. Output is shown to the user in the box at the bottom.

The user can also check all the reservations between a start and end date at the bottom of the user-interface. The reservations are then printed in the box below that. To do that the program should read from the text-file and print the output.

The program should also save customers historical data, so that you can see when a customer has visited before.

Program's structure plan



Since I don't yet know how to make an UI I have not incorporated that in the plan yet, I will add that when I have learned how it works. The hotel class has all the reservations and guest stored in lists, and it can also read previous reservations from a list. When a reservation is made a guest object is created if that guest does not already exist, and then a reservation is made in the reservations class. All reservations are saved in a list and the guests all reservations are saved in separate lists for every guest. The room classes are used to get room numbers and prices for the rooms. All reservations are stored in the Hotel class.

Data structures

Lists and probably dictionaries will be used.

Files and file formats

The program uses text files. One text file for all the hotels reservations and one for each guests reservation. One reservation should be one line in the text file.

Algorithms

The program will have some kind of loop to iterate through every line in the text files to read all the reservations that have already been done, and then save them into the lists to make the execution of the program easier. Alternatively, the program don't need to save everything into the lists every time you start the program, the program could also iterate through the files each time something needs to be checked.

Testing plan

The testing should test the core parts of the program, like the ability to correctly read from files, write to files, and make reservations. The testing should test that a reservation cannot go through if the room is already booked those dates. If all these things work then the core of the program is working. Especially the edge cases should be tested.

Libraries and other tools

I am planning on using just the standard libraries listed in A+.

Schedule

I will try to be a little bit ahead of the schedule so that I have time to implement everything and get everything to work without stressing. By the first checkpoint I will try to have the core of the program working and by the second checkpoint the UI somewhat working. I will add test along the way to test the different parts of my program. I will not have any micromanaged schedule, but I will do work when I have time.

Literature references and links

I will use Google and ChatGPT in my project, mostly to get examples and to help with debugging.