Reservation System

Personal Information:

Name: **Beatriz Glaser**

Student number: **1021033**

Study program: **Digital Systems and Design - Bachelor's Program**

Study year: **one**

Date: **23.02.2022**

General Description:

My plan is to complete my project with the hard-level requirements.

I want to create a reservation system used on a bowling alley. The basic idea of the system is to book and keep track of the (un)reserved lanes. The program allows the user to book a bowling lane, as well as change or cancel a reservation. In addition, the system will save/recognize users based on a username and password so that clients can "log in" and check their own bookings.

As part of the requirements, part of the reservation process includes adding/providing detailed information (e.g.: how many people are going to be playing on the lane, their names and shoe sizes) and additional options (e.g.: beverage and snacks deals). The system will then calculate the total price of the booking based on what was ordered.

I also want to add a user "bonus" category, that would allow returning costumers to receive better deals/promotions.

Use Case Description and User Interface:

The UI can be divided into 3 important windows (with maybe extra ones to add details). The first one concerns the user "sign-up" or "log-in" processes. Both contain text boxes, so that the user can add the necessary information. The second one is the main reservation area. It should provide the user with a clear visual of a calendar and the available times and lanes. In this window, the user should also be able to press the desired time and execute the booking. Here, an extra window could be added for a clearer reservation process (this window could include extra information and the possibility of adding extra services). The third and last main window consists of a "profile" page. Here, the user would be able to see their own information and confirmed bookings.

Let us walk through a concrete use case. A client is using the system for the first time, so they enter the program and choose the "sign-in" option on the first window. They are then presented with a series of text boxes where they should add their name, email address, password, birthdate, phone number (and maybe shoe size). Once they press "confirm", the system first checks through a file list whether that email has already been registered or not, if so, a message appears notifying the user the email address has already been added to the program and offering for it to be redirected to a “log-in” page, where they would have to provide a password. If the user is in fact a first-time client, the program then saves their information on a file which contains all registered users.

Once logged in, the user is redirected to the second main reason where they scroll through the following dates and press their desired date and time. They are then given the option to add extra services to the booking. Once that is done, the program calculates and prints the price for their booking. The system will also mark that lane as occupied and not allow other users to book it for that specific time.

The user then wants to confirm whether or not his reservation is correct, so they press a “profile” icon and are directed to the third main window, their own page. There they can scroll through it and check their own reservation and information. There are also buttons to edit the personal information (redirected to the “signup” window) or one of their reservations (redirected to the reservations window).

Structure plan:

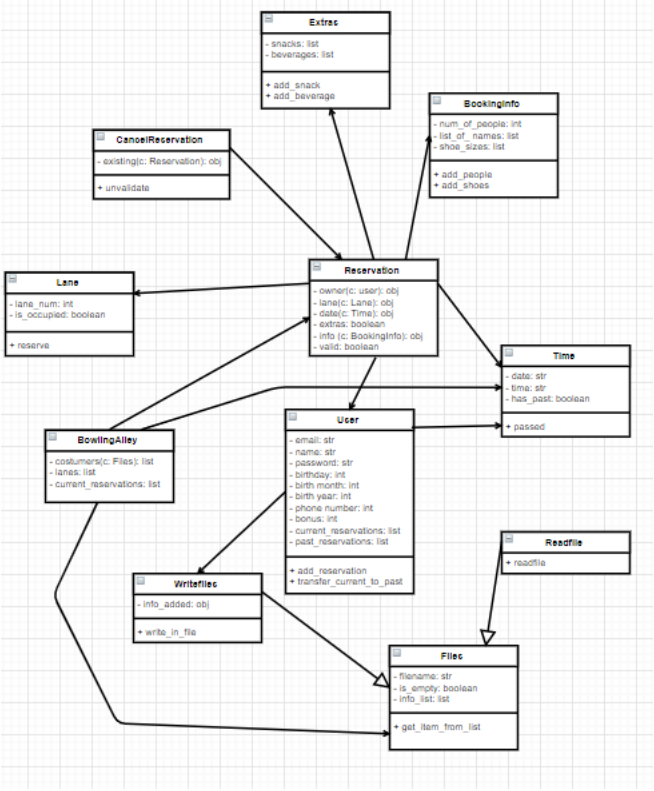

Figure 1 - UML graph.

The figure above shows a UML sketch of the classes I plan on coding. Their relationships are straight forward and described on this plan’s program description and use case. The biggest “connected chain” is with the reservation class, as it is the main function of this program.

All the bowling alley information is stored on separate files, as previously described. The BowlingAlley class contains a list of files (each for a costume). And for each user, a file is created.

Data Structures:

I will not do anything complicated for the data structure. All information will be stored on lists (of objects and files.

Files and file formats:

For now, my plan is to only use text files. A file will contain all users registered to the system, and there will be a file for each costumer which would store their personal information and past bookings.

There would also be a single file containing the general bookings for each day. The information for each booking includes user email and phone number, date and time of the booking, lane number and number of people playing.

I do not have any prior experience with creating graphical interfaces, so I am not still sure if I will need to use files with images or not. However, I believe my program specifically will not require any images.

Algorithms:

I will not be using any complicated mathematical calculations on my program. The only operations used will be addition and subtraction. The program will add total prices of each reservation and subtract possible promotions and bonus points.

Testing Plan:

I plan on doing unittests for some part of the program, and process tests throughout the development process. I want to code unittests, specifically for the first part of my project where I will be doing the basic code to the system. I want the basic part to be thoroughly tested.

Then, I plan on doing a lot of tests once the program is completed, in order to check the complete user experiences and guarantee every feature works.

Libraries and tools:

I will only be using PyQt5, as recommended by this course.

Schedule:

As this is my first programming project, it is hard to estimate and allocate the correct amount of time for each phase. However, I would ideally like to complete the project by first focusing on the main and most important features and then adding extra ones along the process. By doing the project in small iterations, I believe I will be a better control of the overall performance of the system. As I have been planning the project, I have noticed that my ideas and implementable features are endless and I have very little notion of the time in hand. Hence, I will hopefully add as much features as timely possible, focusing on keeping the program clean and easy to operate.

Firstly, I will focus strictly on the "background" code. I will establish all the classes, and basic requirements: the memory system that saves all users and reservations, and the reservation system itself (booking available times). I would like to have a functioning system with these features by the first checkpoint, together with some unittests to verify those features.

Secondly, I will focus on the graphical interface. This part is even harder for me to estimate the time needed as I still do not know much about it. A lot of research will be required. I want to complete a graphical interface that covers all previously coded basic requirements. Once that is done, I will start adding extra features and their representation on the graphical interface. My plan it to have at least 1/3 of the extra features added by the second checkpoint.

After checkpoint 2, I will continue to add on features and run final tests on the whole program. I plan on continuously working on the project until the final demo presentation.

Literature references and links:

3.10.2 Documentation. (n.d.). Python Documentation. Retrieved February 24, 2022, from https://docs.python.org/3/

Qt for Python — Qt for Python. (n.d.). Pqty5 Documentation. Retrieved February 24, 2022, from https://doc.qt.io/qtforpython/