

INFR 2141U - Object Oriented Programming

Group 35 =====

Student Name	Student ID	Contribution
Nate Coolidge	100749708	<ul style="list-style-type: none">- Big brain behind the entire operation- Implemented ORM, SQLAlchemy, QTDesigner (to work with PyQt6)- background music,- debugging
Saief Shams Murad	100836639	<ul style="list-style-type: none">- Worked on GUI,- background images,- worked on ORM,- general class structure,- minor debugging
Christina Cho	100816275	<ul style="list-style-type: none">- Worked on all of error handling,- GUI,- debugging
Jia Chen (Anthony Chen Chen)	100572516	<ul style="list-style-type: none">- Worked on password encryption (but we didn't have enough time to implement it into the code.)- Worked on debugging , UML Diagram

→ Project Details

This is a Restaurant Reservation system created using Python Object Oriented Programming. For storing, updating and displaying account information and reservation details, SQLAlchemy handles mapping between the *sqlite3* database and the object classes "Reservation" and "Account". PyQt6 and QTDesigner has been used to design and implement the GUI.

→ Required Modules

- sqlalchemy := *pip install sqlalchemy*
- PyQt6 := *pip install PyQt6*
- pygame := *pip install pygame*

→ Class Overview

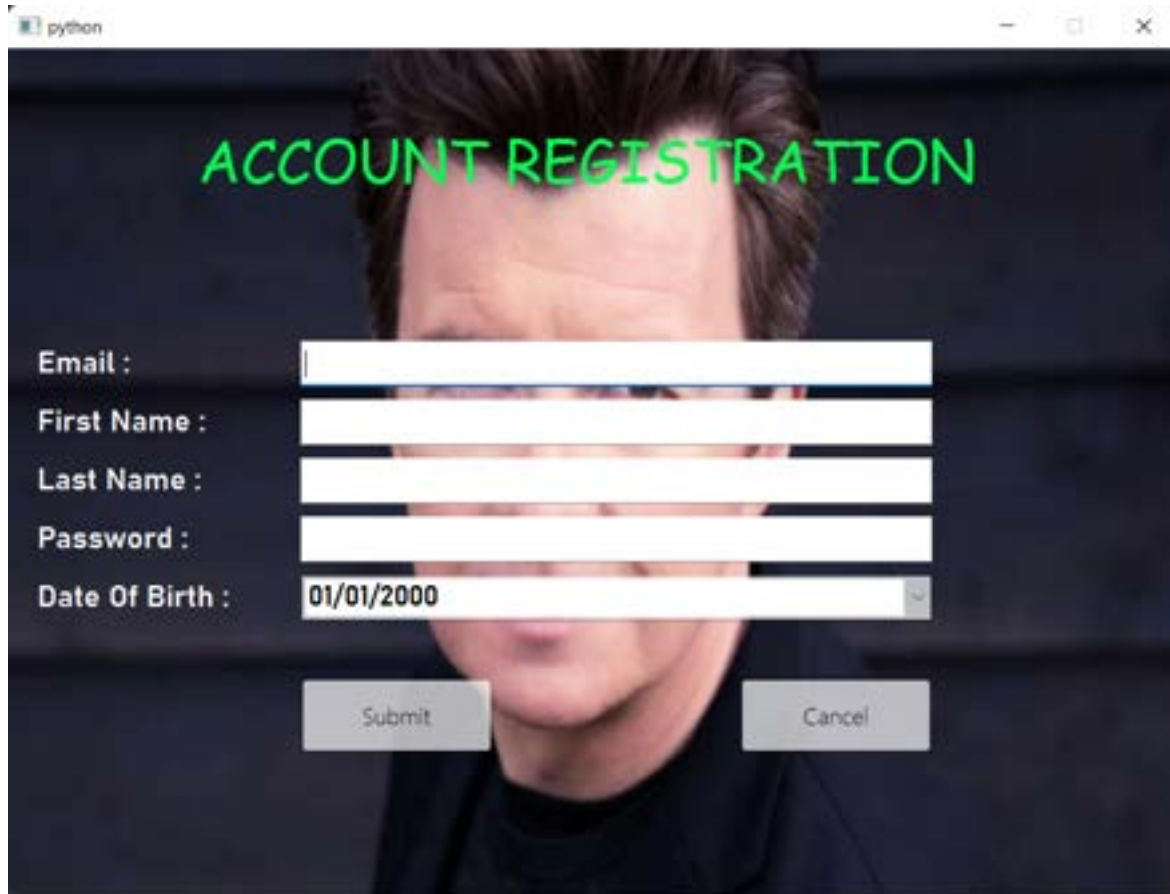
- ◆ **restaurantDatabase:** Base class to synchronize objects with sqlite3
- ◆ **Account (derived class of restaurantDatabase):** Maps the sqlite3 to the instance of an account
- ◆ **Reservation (derived class of restaurantDatabase):** Maps the sqlite3 to the instance of a reservation
- ◆ **QMainWindow:** Forms all the GUI windows (by PyQt6 and QTDesigner)
- ◆ **GUI Classes**
 - **Main Menu (derived class of QMainWindow):** Handles operations inside the Main Menu UI
 - **Login Menu (derived class of QMainWindow):** Handles operations inside the Login Menu UI
 - **Register Menu (derived class of QMainWindow):** Handles operations inside the Register Menu UI
 - **Reservation Menu (derived class of QMainWindow):** Handles operations inside the Reservation Menu UI
 - **Reserve Menu (derived class of QMainWindow):** Handles operations inside the Main Menu UI
 - **View Reservation Menu (derived class of QMainWindow):** Handles operations inside the View Reservation UI
 - **Cancel Reservation Menu (derived class of QMainWindow):** Handles operations inside the Cancel Reservation UI
 - **Modify Reservation Menu (derived class of QMainWindow):** Handles operations inside the Modify Reservation UI
- ◆ **GUI Class:** Controls the GUI classes indexing
- ◆ **Manager:** Controls all interaction between the database and the program itself.

→ Main Menu



Displays options to register/signup, login, or exit. Plays a piece of soothing background music.

→ Register Menu



The image shows a screenshot of a Python application window titled 'python'. The window displays an 'ACCOUNT REGISTRATION' form. The title 'ACCOUNT REGISTRATION' is written in large, green, all-caps letters at the top. Below the title, there are five input fields with labels to their left: 'Email :', 'First Name :', 'Last Name :', 'Password :', and 'Date Of Birth :'. The 'Date Of Birth' field is a date picker showing '01/01/2000'. At the bottom of the form, there are two buttons: 'Submit' and 'Cancel'. The background of the window is a dark, out-of-focus image of a person's face.

ACCOUNT REGISTRATION

Email :

First Name :

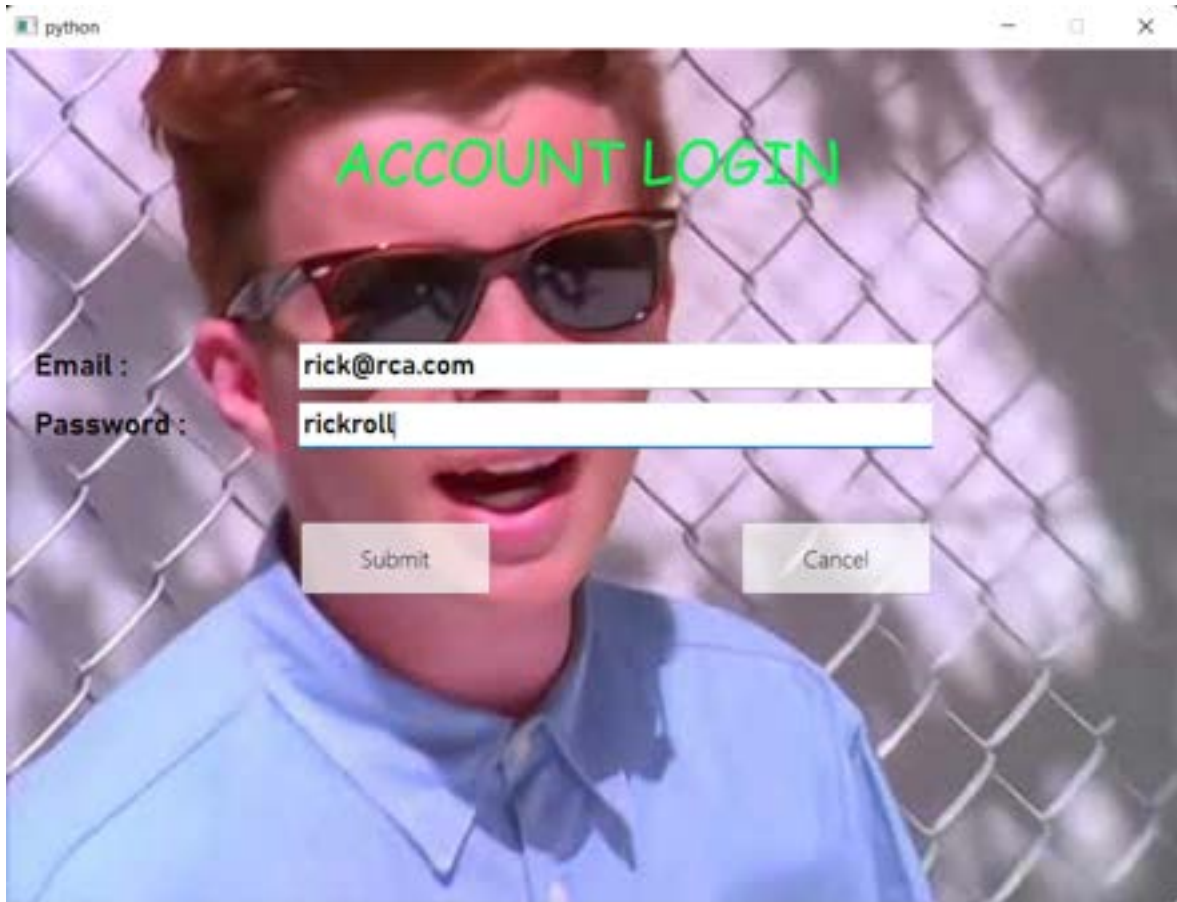
Last Name :

Password :

Date Of Birth :

When user selects register/signup, this window prompts the user to enter the necessary information.

→ Login Menu



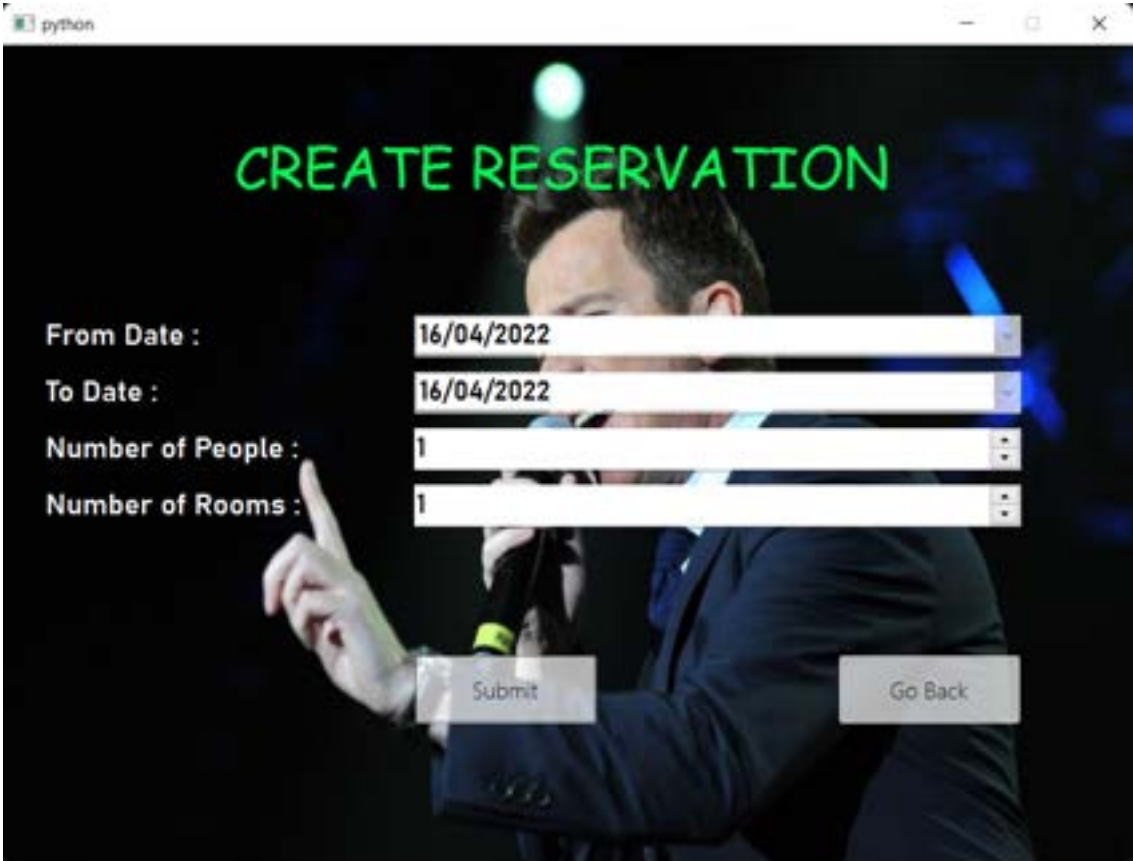
When the user selects login, this window prompts the user to enter login information.

→ Reservation Menu



After a successful login, users enter this menu to View, Make, Modify or Cancel their reservations.

→ Add Reservation



A screenshot of a Python application window titled 'python'. The window displays a 'CREATE RESERVATION' form. The form has a dark background with a blurred image of a man in a suit. The title 'CREATE RESERVATION' is written in large, green, capital letters. Below the title, there are four input fields with labels to their left: 'From Date :', 'To Date :', 'Number of People :', and 'Number of Rooms :'. The 'From Date' and 'To Date' fields contain the date '16/04/2022'. The 'Number of People' and 'Number of Rooms' fields contain the number '1'. At the bottom of the form, there are two buttons: 'Submit' and 'Go Back'.

Field Label	Value
From Date :	16/04/2022
To Date :	16/04/2022
Number of People :	1
Number of Rooms :	1

User is prompted to enter Reservation start and end date, Number of Persons, Number of Rooms. **The program calculates the Number of Days the customer is staying by itself.** The reservation start date or From Date always defaults to the current date.

→ View Reservation



The screenshot displays a Python application window with the title 'python'. The main content area shows the word 'RESERVATIONS' in large green letters. Below this, a table lists reservation details. The table has six columns: 'Res. ID', 'From Date', 'Days', 'To Date', 'No. of Guests', and 'No. of Rooms'. There are three rows of data. At the bottom of the window, there are two buttons: 'Refresh Table' and 'Go Back'.

	Res. ID	From Date	Days	To Date	No. of Guests	No. of Rooms
1	13	2022-04-16	1	2022-04-17	1	4
2	14	2022-04-16	0	2022-04-16	1	1
3	15	2022-04-16	1	2022-04-17	1	1

A table shows all of the reservations the user has made with a unique reservation ID. The table updates automatically.

→ Modify Reservation

The screenshot shows a window titled 'python' containing a form titled 'MODIFY RESERVATION' in green text. The form is overlaid on a background image of a smiling man in a suit holding a microphone. The form fields are:

- Select Reservation ID:** A dropdown menu with a red arrow pointing to it. The dropdown is open, showing options 13, 13, 14, and 15. The first '13' is highlighted.
- From Date :** A text input field containing '16/04/2022'.
- To Date :** A text input field containing '16/04/2022'.
- Number of People :** A spinner input field containing '0'.
- Number of Rooms :** A spinner input field containing '0'.

At the bottom of the form are three buttons: 'Pull Data', 'Save', and 'Return'. A red arrow points to the 'Pull Data' button.

After making and viewing the reservation, the user can **select the Reservation ID** of the exact reservation they need to make changes and **hit "Pull Data"**. That loads the current reservation details. The user can then make as many changes needed and **when they click "Save", the updated reservation is saved.**

→ Cancel Reservation

The screenshot shows a window titled "python" with a background image of a man speaking into a microphone. The title "CANCEL RESERVATION" is displayed in green. Below the title is a table with the following data:

	Res. ID	From Date	Days	To Date	No. of Guests	No. of Rooms
1	13	2022-04-16	1	2022-04-17	1	4
2	14	2022-04-16	0	2022-04-16	1	1
3	15	2022-04-16	1	2022-04-17	1	1

Below the table, there is a label "Select Reservation ID:" in green. To its right is a dropdown menu with the following options: 13, 13, 14, 15. A red arrow points to the first "13" option, and another red arrow points to the "14" option. Below the dropdown menu are two buttons: "Return" and "Cancel Reservation".

The screenshot shows the same window as above, but with reservation ID 14 selected in the dropdown menu. The table now only shows reservations 14 and 15:

	Res. ID	From Date	Days	To Date	No. of Guests	No. of Rooms
1	14	2022-04-16	0	2022-04-16	1	1
2	15	2022-04-16	1	2022-04-17	1	1

The "Select Reservation ID:" label is still present, and the dropdown menu now shows only "14". The "Return" and "Cancel Reservation" buttons are still at the bottom.

The user can see a similar table like the "View Reservation" window. They must select the unique reservation ID of the specific reservation they want to cancel, and then hit "Cancel Reservation". This deletes the reservation from the database.