INFR 2141U - Object Oriented Programming

Group 35 ==========

Student Name	Student ID	Contribution
Nate Coolidge	100749708	- Big brain behind the entire operation - Implemented ORM, SQLAlchemy, QTDesigner (to work with PyQt6) - background music, - debugging
Saief Shams Murad	100836639	- Worked on GUI, - background images, - worked on ORM, - general class structure, - minor debugging
Christina Cho	100816275	- Worked on all of error handling, - GUI, - debugging
Jia Chen (Anthony Chen Chen)	100572516	- 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

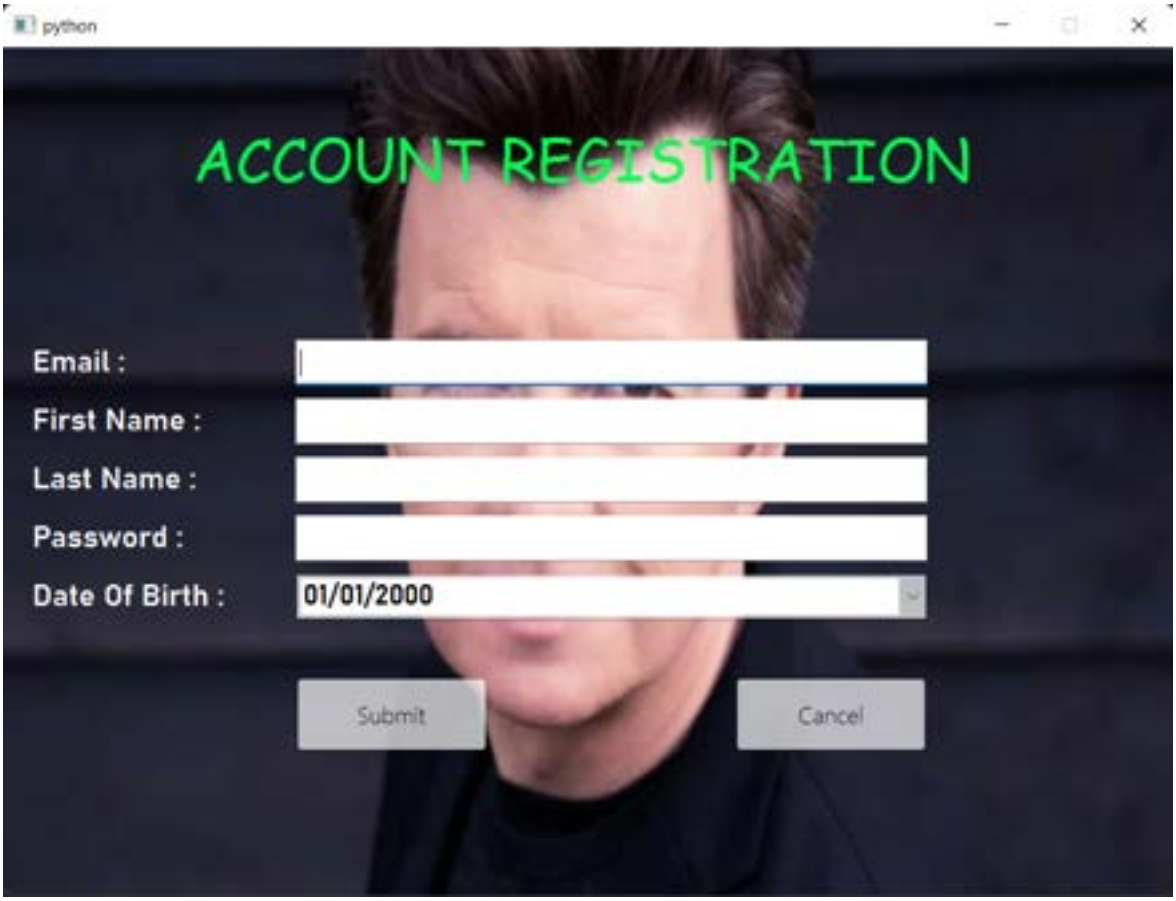
- **GUI Class:** Controls the GUI classes indexing

- **Manager:** Controls all interaction between the database and the program itself.

➔ Main Menu

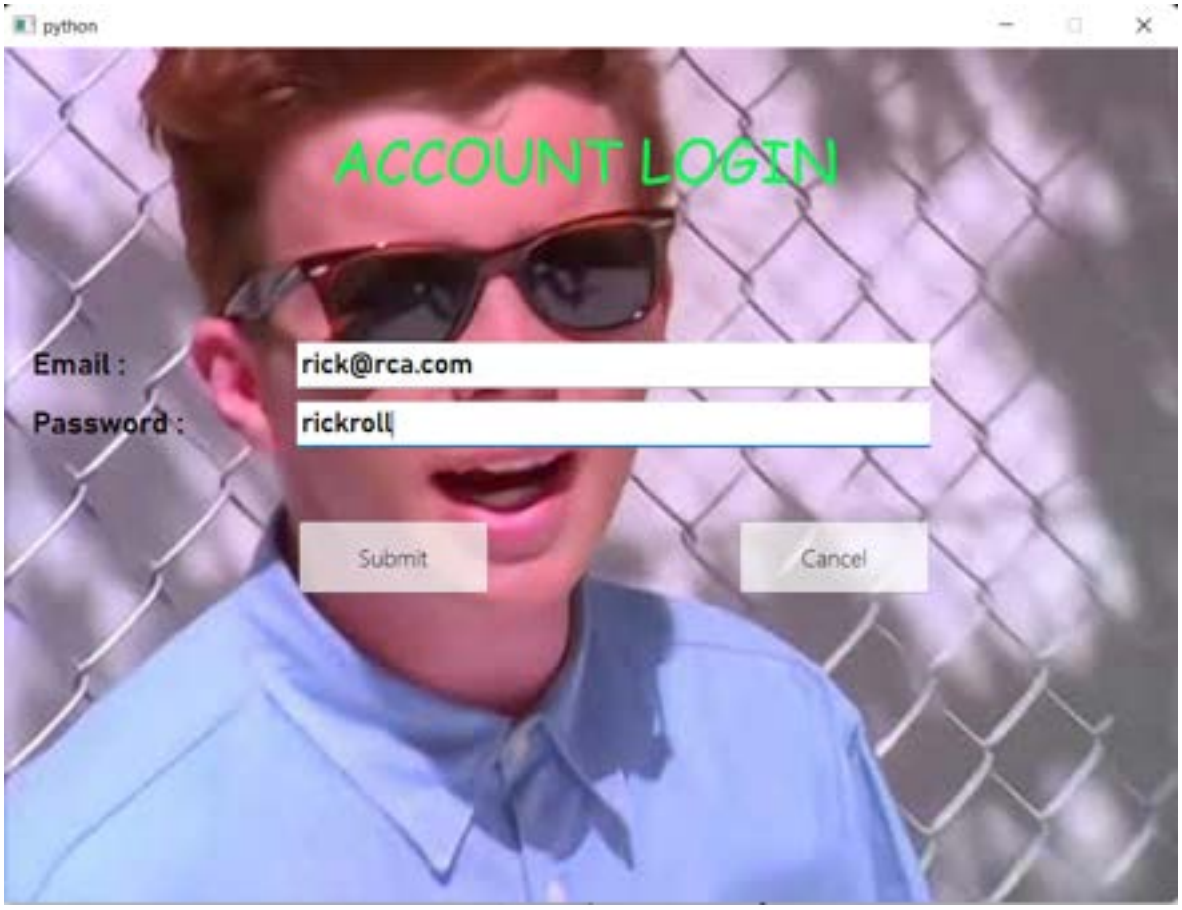
Displays options to register/signup, login, or exit. Plays a piece of soothing background music.

➔ Register Menu



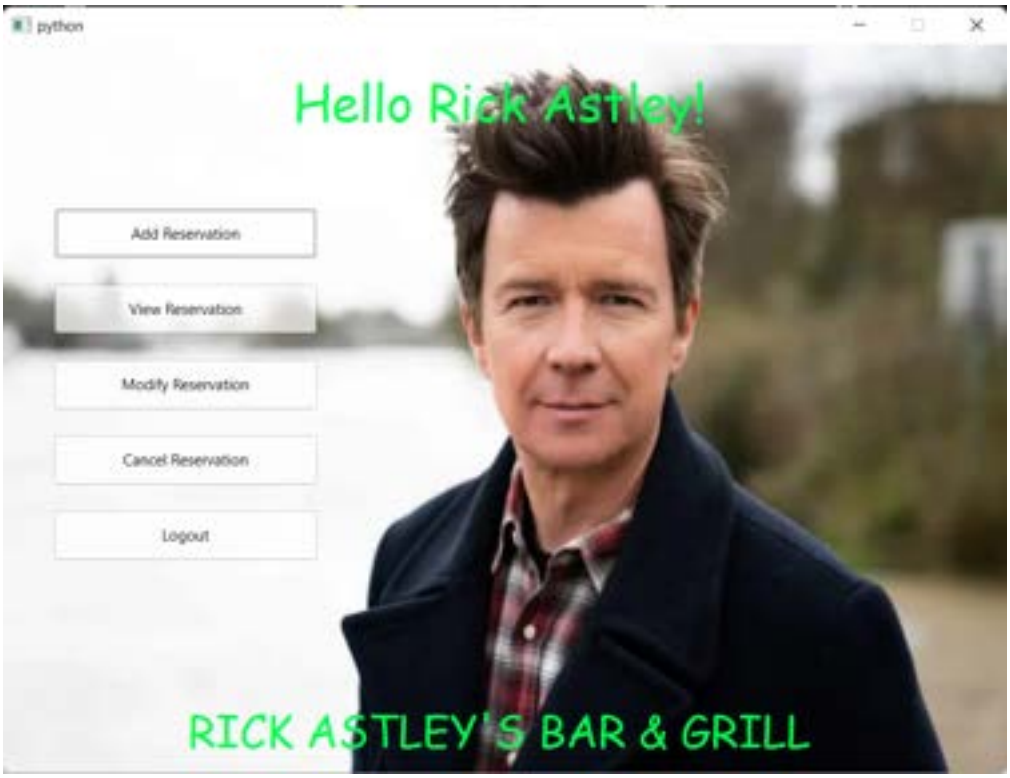
When user selects register/signup, this window prompts the user to enter the necessary information.

➔ Login Menu



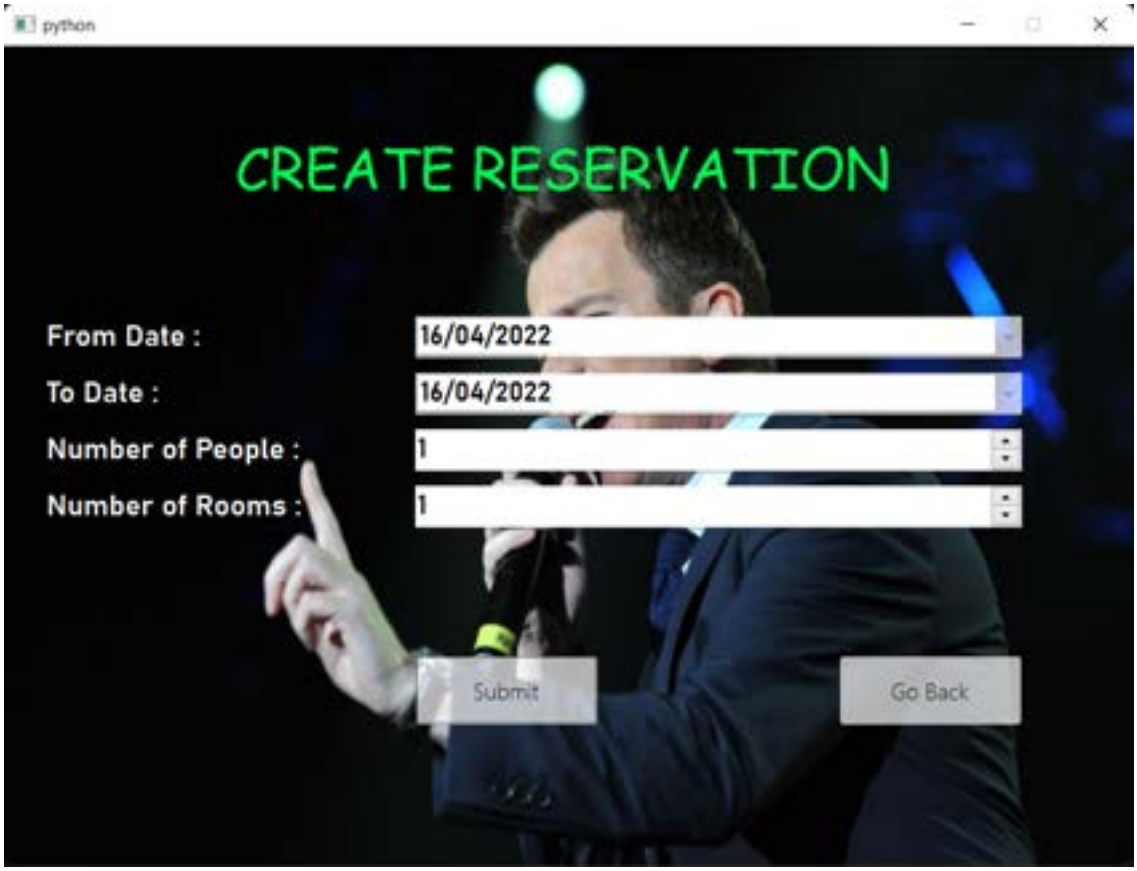
When the user selects login, this window prompts the user to enter login information.

➔ Reservation Menu



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

➔ Add Reservation



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

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

➔ Modify Reservation

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 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.