# Airline Management System

CS 5200 Final Report

## Info

Group Name: BasangoudarARongaE
Members of Group: Amey Basangoudar, Elisa Ronga

## README

To start the database and run the application perform the below steps:

Database:

- Use MySQL workbench and import the schema and data dump that is included in the file “dump_airline_dbms.sql”.

Application:

- Preferably use Windows OS (MacOS may have compatibility issues with the GUI)
- Use VS code or PyCharm as IDE
- Install mysql DB connector using the command "python -m pip install mysql-connector-python" on the terminal.
- Install tkinter using the command “pip install tkinter” on the terminal
- Tkinter documentation: https://docs.python.org/3/library/tkinter.html
- Run the file “app.py”. You will be prompted to enter the username and password for your database.
- When the python script is run the GUI will appear on top of the IDE.
  May face the error: mysql.connector.errors.NotSupportedError: Authentication plugin 'caching_sha2_password' is not supported
  To resolve this issue, you can either upgrade to a newer version of the MySQL connector library that supports the 'caching_sha2_password' authentication plugin, or you can change the default authentication plugin used by your MySQL server to one that is supported by your current version of the MySQL connector library.

# Technical Specifications

The project uses a **relational database (SQL)** for storage and persistence on **MySQL Workbench**.
The user application uses a **python programming** language script that generates a Graphical User Interface (**GUI)** using the **tkinter library**, in order to make the experience to the user more intuitive and interactive.
The project was implemented on Windows OS hardware.

# Conceptual Design

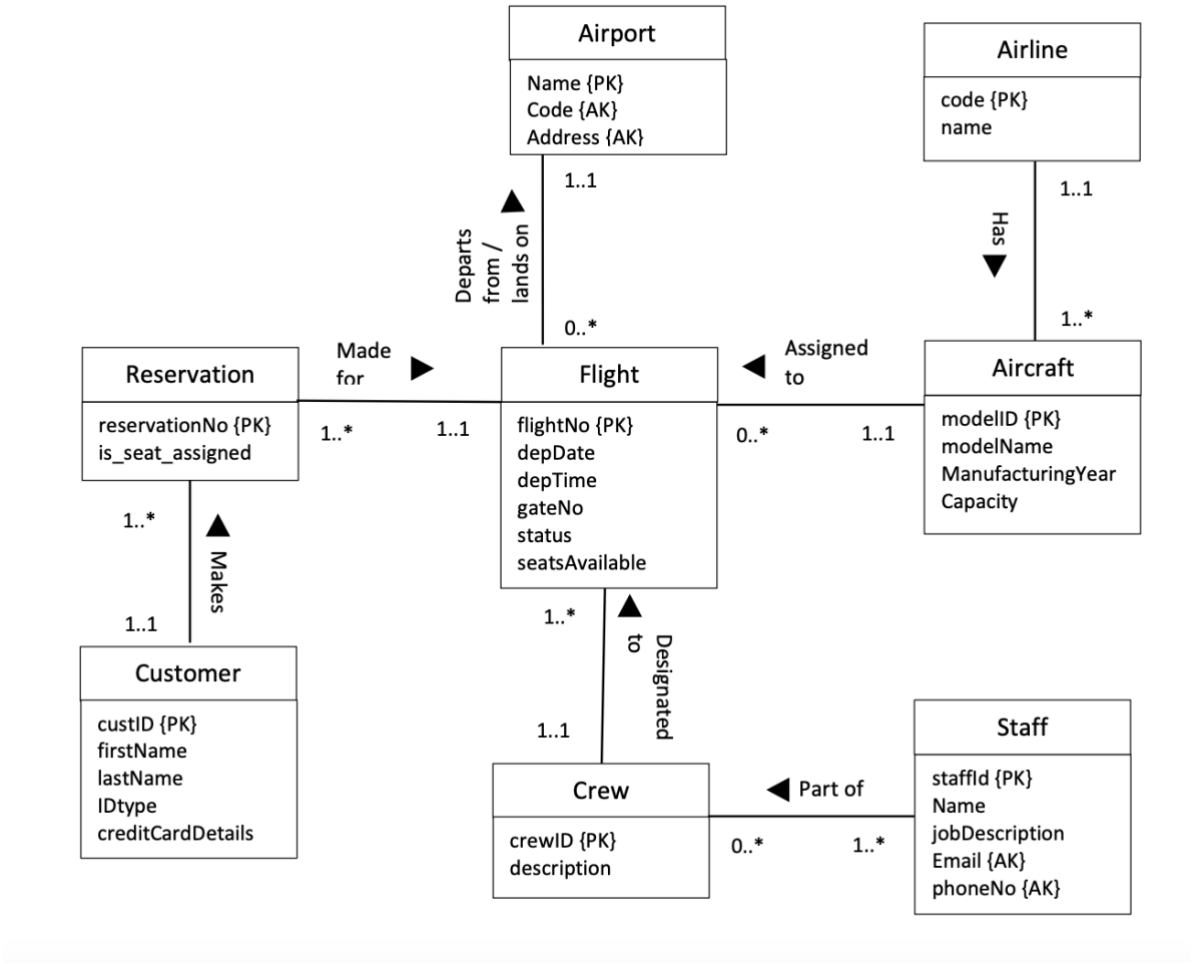


Figure 1: Conceptual Diagram using UML notation

# Logical Design

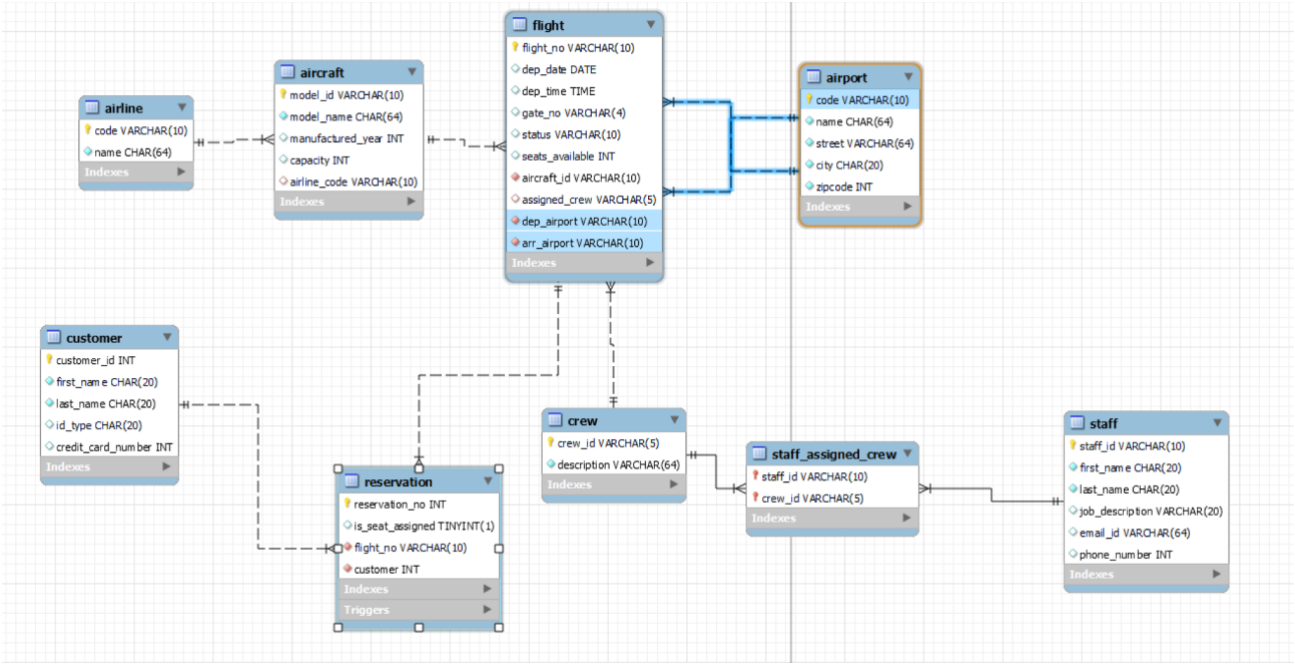


Figure 2: Logical Design using crow's foot notation

# User Flow

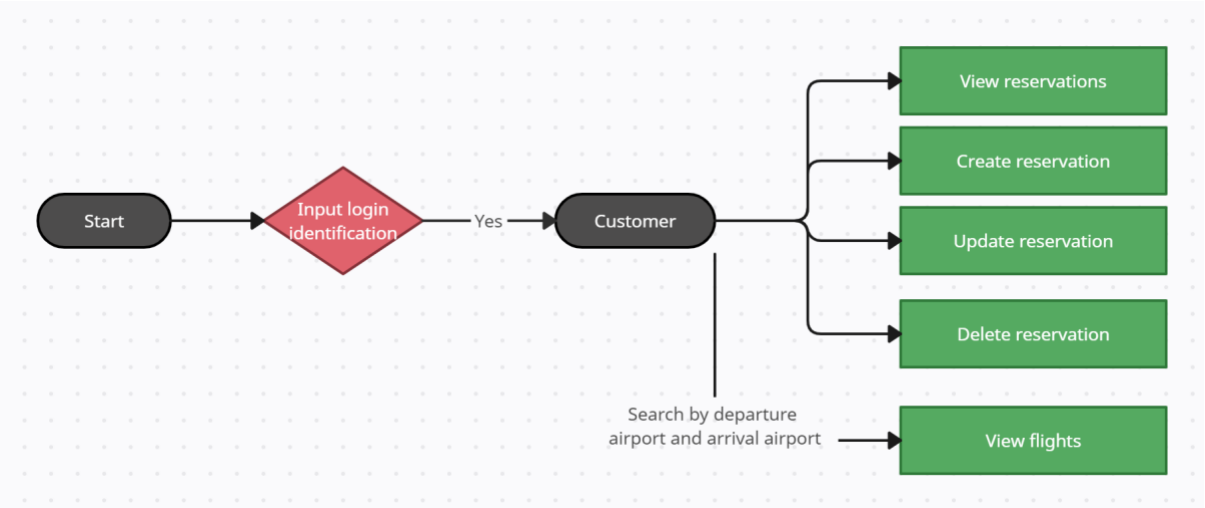


Figure 3: The user flow above represents how the GUI will proceed for the customer

The customer upon starting the GUI will have the option to view all the flights that are available by inputting the departure and arrival airport. (A table is given to show all the airports that are in the database).

**Creating a reservation:**

The customer upon choosing the desired flight number, will next have to create a reservation by entering the first name, last name, flight number, whether a seat should be assigned (0 or 1), ID provided and credit card number. Once the customer presses the “create reservation” button, the reservation can be viewed by entering the reservation number and clicking on the “view reservation” button.

**Updating a reservation:**

The customer can update the flight number and whether a seat should be assigned. To do this the customer needs to input the new flight number, the new value of seat assigned (0 or 1) and the existing reservation number. (It is assumed that the reservation number is sent to the customer via email or text when the reservation was first created). Next the “update reservation” button is pressed to update the reservation. To view the updated values, the customer needs to enter the reservation number and click on “view reservation”.

**Deleting a reservation:**

The customer can delete a reservation by entering the reservation number and clicking on the “delete reservation” button.

# Lessons Learned

One of the lessons learned was how important the conceptual design is to the implementation of the process. Having a well thought-out conceptual design and delegating tasks accordingly was essential for our team to work asynchronously. Also our understanding of how procedures, functions and triggers enabled a smoother flow of the functionalities in the database, was enhanced when we implemented it for a real application based GUI.

We also learnt the importance of building a solid database for airlines, since any minute failure in one of the functionalities can cause a cascade of failures thus compromising the entire database.

We initially planned to include a functionality that allows a staff member to view and perform CRUD operations on the crew tables and flight tables but we felt that the customer side of the application need to work well before more functionalities are added for the staff.

# Future Work

Future work consists of creating CRUD functionality for other tables including flights, as well as implementing security protocols. The latter especially includes creating isolation for each customer's account in order to protect their private information. The database can be further improved by introducing more entities like customer account login info and notifications to enable better customer experience.