CS-UY Introduction to Databases Final Project Hand in:

Team Members:

Denizhan Ene
Ivan Sidorenko
Konuralp Sarisozen

- Your source codes. (Details about whether to zip it, etc., will be provided.)

Our source codes are included in the submission.

- A list of the files in your application and what's in each file. (E.g. "homepage.phpscript to generate home page".)

- A separate file that lists all of the use cases and the queries executed by them (with brief explanation). This should be well organized and readable. It should be detailed enough to give readers a good idea of how your application works, without making them dig through all the code.

- For team projects: A summary of who did what.

Denizhan Ene:
Denizhan did the Customer usecases:

1. Purchase Ticket

- **Functionality:** Allows a user to purchase a ticket for a specified flight.
- **Process:** Upon accessing the ticket purchase page, the user is presented with a form to complete the purchase. The form includes flight details like flight number and departure time. When submitted, the user's purchase details are stored in the database.

2. View My Flights

- **Functionality:** Enables users to view a list of flights they have booked that are scheduled for the future.
- **Process:** After verifying the user's login status, the system retrieves flights associated with the user that have a departure time later than the current time. These flights are then displayed on the user's personal flight page.

3. Cancel Ticket

- **Functionality:** Allows users to cancel a ticket they have purchased.
- **Process:** Users can cancel a ticket if the flight departure time is more than 24 hours away. The system checks the departure time, and if it meets the condition, the ticket is deleted from the database. The user is then redirected to the 'My Flights' page with a relevant success or error message.

4. Rate Flights

- **Functionality:** Provides a platform for users to rate and comment on flights they have taken in the past.
- **Process:** The system fetches a list of past flights that the user has completed (flights where the arrival time is in the past) and displays them on a page where the user can submit ratings and comments.

5. Customer Profile

- **Functionality:** Displays the profile information of the logged-in customer.
- **Process:** After verifying the user's login status, the system retrieves and displays the user's profile information, such as email and other personal details, from the `Customer` table.

6. Submit Rating

- **Functionality:** Allows users to submit ratings and comments for flights they have completed.
- **Process:** Users submit their ratings and comments through a form. The system then either inserts a new rating or updates an existing one for the specified flight in the `Ratings` table. Users are redirected back to the rating page with a confirmation message.

7. Logout

- **Functionality:** Logs out the current user from the system.
- **Process:** The system clears the user's session and redirects them to the login page, ensuring a secure log-out process.

Konuralp: Konuralp handled the home page requirements: logging in as a customer, logging in as an airline staff, registering in as a customer, registering in as an airline staff, displaying one way flights information, displaying return tickets information, and cheking flight status.

For logging in, one form with two submit buttons are implemented with different values assigned it to the buttons in the html template. Then, in the code, this value is checked, whether the form is filled for customer or airline staff, and then necessary actions are done: takig username (or email) and password and comparing it with the database. If user does not exist or password is wrong, it is handled with displaying error. Otherwise the logging and session management is implemented.

For registration, it is handled in separate links for customers and airline staff. So, two different templates are used. In these links, all of the attributes are asked to the users to save to the database if their username already not exists. Error is displayed if username exists and error is displayed if we try to register an airline staff that works for an airline that is not in the database (foreign key constraints are handled; the program does not crash.

For viewing flights for oneway tickets and return tickets, again two buttons are used with different values for form submission, and it is checked in the code which button is clicked. Then based on that, sql queries are implemented differently. The search is based on city name but not the airport. SQL needed to handle airport, flight, ticket, airplane tables together so that calculated price can be displayed through tables and necessary checks can be done for the flights for displaying and ensuring the correct handling of data and tables. Error handling is implemented for flights that do not exist in the database when searched.

Flight status is also handled in separate link you can reach from home where you can enter values for the flight and see the status again in the table. Error handling is implemented for values that do not exist in the database when searched.

Ivan: ## 1. Airline Staff Profile

- **Functionality:** Displays the personal profile of the airline staff member logged into the system.
- **Process:** When a staff member accesses their profile, the system first verifies their login status and role as airline staff. Upon successful validation, it retrieves and displays the staff member's details, including their username and associated airline, ensuring only authorized personnel access this information.

2. View Future Flights

- **Functionality:** Allows airline staff to view a list of all flights scheduled by their airline for the upcoming 30 days.
- **Process:** Executes a query to retrieve flights for the staff member's airline that are scheduled to depart within the next 30 days. The result is displayed in a structured format, providing a clear overview of upcoming flights, aiding in operational planning and management.

3. Create New Flight

- **Functionality:** Facilitates the creation and scheduling of new flights by airline staff.
- **Process:** Staff members input details of a new flight via a form, including flight number, departure and arrival times, airport codes, and base price. Upon submission, this data is inserted into the flight schedule in the database. The system then displays the updated list of future flights, including the newly added flight.

4. Change Flight Status

- **Functionality:** Allows staff to update the operational status of a flight (e.g., on time, delayed, canceled).
- **Process:** Staff submits the updated status for a specific flight through a form. The system checks if the flight exists and updates its status in the database. A confirmation is displayed to the staff member, ensuring transparency in flight operations.

5. Add New Airplane

- **Functionality:** Adds a new airplane to the airline's fleet database.
- **Process:** Airline staff enters details of the new airplane, including its ID, number of seats, manufacturer, model number, manufacture date, and age. The system validates the data and inserts it into the fleet database, expanding the airline's operational capacity.

6. Add New Airport

- **Functionality:** Registers a new airport within the airline's operational network.
- **Process:** Staff provides details of the airport, such as its code, name, location, number of terminals, and type. The system first checks for the existence of the airport in the database to avoid duplicates and then adds it, increasing the airline's range of destinations.

7. View Flight Ratings

- **Functionality:** Displays customer ratings and comments for specific flights.
- **Process:** The staff inputs flight details to view its ratings. The system retrieves the average ratings along with individual customer comments from

the evaluations database, providing insights into passenger satisfaction and areas for improvement.

8. Schedule Maintenance for Airplanes

- **Functionality:** Schedules maintenance sessions for airplanes in the fleet.
- **Process:** Staff submits maintenance details, including airplane ID and maintenance time frame. The system verifies the airplane's existence and checks for scheduling conflicts before confirming the maintenance session, ensuring aircraft safety and operational efficiency.

9. View Earned Revenue

- **Functionality:** Calculates and displays the airline's revenue from ticket sales over specified periods.
- **Process:** The system automatically calculates the total revenue from ticket sales for the last month and year. This financial data is crucial for assessing the airline's performance and making informed business decisions.

10. View Frequent Customers

- **Functionality:** Identifies and provides details of the airline's most frequent customers.
- **Process:** Upon inputting a customer's email, the system fetches their flight history and frequency of travel over the past year. This feature helps in recognizing loyal customers and understanding customer travel patterns.

Each use case contributes to a comprehensive airline management system, enabling efficient and effective handling of various aspects of airline operations, from flight scheduling and fleet management to customer relations and revenue tracking.