Team 2

DatabasePros

Vokhob Bazorov - vkb4@pitt.edu

Luiza Urazaeva - lau4@pitt.edu

Nathaniel Stump - nrs70@pitt.edu

**Manual:**

Administrator Feature 1: Erase the database

This feature will allow you to erase all data in the database. The database itself along with all its tables, restrictions, and triggers will remain.

Administrator Feature 2: Load airline information

This feature will allow you to load airline information into the database from external file with tab-separated tuples.

Administrator Feature 3: Load schedule information

This feature will allow you to load schedule information into the database from external file with tab-separated tuples.

Administrator Feature 4: Load pricing information

This feature will allow you to choose between two options.

1. Load pricing information into the database from an external file with tab-separated tuples.

2. Set a new high price and low price for flights between a given departure city and arrival city.

Administrator Feature 5: Load plane information

This feature will allow you to load plane information into the database from external file with tab-separated tuples.

Administrator Feature 6: Generate passenger manifest for specific flight on given day

This feature will print out the salutation, first name, and last name, of every passenger on the given flight on the given day.

Administrator Feature 7: Update the current timestamp

This feature allows you to specify a new timestamp, which will replace the old one.

Customer Feature 1: Add customer

This feature will allow you to add a new customer account to the database.

Customer Feature 2: Show customer info, given customer name

This feature will show all stored information for a given customer, excluding reservations.

Customer Feature 3: Find price for flights between two cities

This feature will show display the price for a flight between two cities given by the user.

Customer Feature 4: Find all routes between two cities

This feature will display all possible routes between two cities given by the user.

Customer Feature 5: Find all routes between two cities of a given airline

This feature will display all possible routes between two cites given by the user, on a specific airline given by the user.

Customer Feature 6: Find all routes with available seats between two cities on given date

This feature will display all possible routes between two cities given by the user, on a date given by the user.

Customer Feature 7: Add reservation

This feature will allow the user to add a reservation by specifying the flight number and date of each leg of the flight the user would like to reserve.

Customer Feature 8: Delete reservation

This feature will allow the user to delete an existing reservation.

Customer Feature 9: Show reservation info, given reservation number

This feature will ask the user for a reservation number, and display the reservation info associated with that reservation number.

Customer Feature 10: Buy ticket from existing reservation

This feature will allow the user to specify a reservation number and buy a ticket for that reservation.

Customer Feature 11: Find the top-k customers for each airline

This feature will allow the user to specify a number *k*, then display the top-k customers who have paid the most money for each airline.

Customer Feature 12: Find the top-k traveled customers for each airline

This feature will allow the user to specify a number *k*, then display the top-k customers who have travelled the most legs for each airline.

Customer Feature 13: Rank the airlines based on customer satisfaction

This feature will print a ranking of the airline companies in order of most customers who have bought tickets.

**Limitations:**

We did not have time to protect against SQL injection, or cover every possibility for incorrect user input. The menus are pretty well protected against bad inputs, but some of the functions are not, especially those that require user data input. Some functions will crash, for instance, if you input the date in the wrong format, or input a string that is the wrong length. To mitigate this, we tried to provide explicit instructions for what to input so that users can avoid incorrect inputs. The program should work fine for a user that intends to use the program correctly and follows the onscreen instructions. However, it is not very well protected against malicious users who are intentionally trying to break the program.

**Possibilities for Improvements**

If we had more time, we would have liked to improve robustness and program security. We could have added protection against SQL injection, and allow the user repeat attempts at inputting valid data in case they make a mistake. We could add better error messages that specifically describe what went wrong, instead of simply printing the stack trace when an exception is thrown.