CS537 Final Project: A Flight Ticket Biding System

Group Members: William Sun (wjs45), Liu Wang (jx38), Xinyu Hong (xh83), Yang Zhang (yz598)

1. Basic Ideas

In the airline industry, there are always mismatches between supply and need. On one hand, many customers want to buy low fare flight tickets, but they cannot afford the full price of some expensive flight tickets. On the other hand, no major airlines can sell all their tickets, but they are reluctant to sell the spare tickets at a lower price, worrying that selling tickets at a relatively low price will hurt their brand. In order to address this issue, we made a pilot airline ticket biding system. It takes the spare tickets from the airlines, and then allocate them between customers interested in buying those tickets. In our specific project, we use the British Airways as an example. We modeled in the flight information of British Airways into our system, and then built a bidding system on that. Customers can enter our system, and input their biddings into the system, and then we will select the winner of the tickets.

2. Work Split

At the beginning of the project, we brain stormed and fixed the basic ideas of the database together. We contributed to the implementation in the following ways

William

- Tested and debugged the SQL tables, queries, and data
- Restructured SQL queries to support their functions and created documentation
- Wrote SQL scripts to automate database installation & synchronization across the project
- Created an HTML GET/REQUEST API
- Set up a remote database on Heroku for future deployment
- Tried to host a remote database on local machines

Liu

- Drew the final E-R diagram based on our discussion and an E-R diagram draft we drew together
- Converted the diagram to relational schemas
- Created the SQL tables
- Wrote the SQL queries for selecting a single winner and the matching tickets. The schemas and queries were modified and tested by others.

Xinyu

- Modeled half of the relationships (flight, schedule, time and bid)
- Generated data for the above relationships
- Checked, modified and made sure the data and the schemas are consistent
- Tested the SQL script on Linux and in Zoo environment and wrote a bash script to automatically draw 10 winners.
- Tried to write PHP scripts to access our database, but it is not included in our project.

Yang

- first proposed the prototype of our project—the idea and framework of bidding based on his knowledge and experience in economics
- developed the core formula for our system which computes the lowest acceptable ticket price
- Modeled and generated the flight and client information part of the database

- checked, modified and made sure the data and the schemas are consistent, and tested the SQL script on Mac.

3. Functions Supported

The E-R diagram of the data base is showed in graph-1 and graph-2 in the following pages, and the database has 11 tables in total:

Airport (airport_code, airport_name, city, state)

Driving_distance (start_airport, end_airport, miles)
Foreign-key start_airport, end_airport reference airport_code in relation Airport.

Route (route_code, departure_airport, arrival_airport, miles)
Foreign-key departure_airport, arrival_airport reference airport_code in relation Airport

Flight (flight_number, airline, standard_price, route_code)
Foreign-key route_code references route_code in relation Route

Schedule (flight_number, date, seat_sold, aircraft_id, time_id)
Foreign-key flight_number references flight_number in relation Flight
Foreign-key aircraft_id references aircraft_id in relation Aircraft
Foreign-key time_id references time_id in relation Time

Time (time_id, departure_time, arrival_time)

Aircraft (aircraft_id, model, number_of_seats, manufacturer)

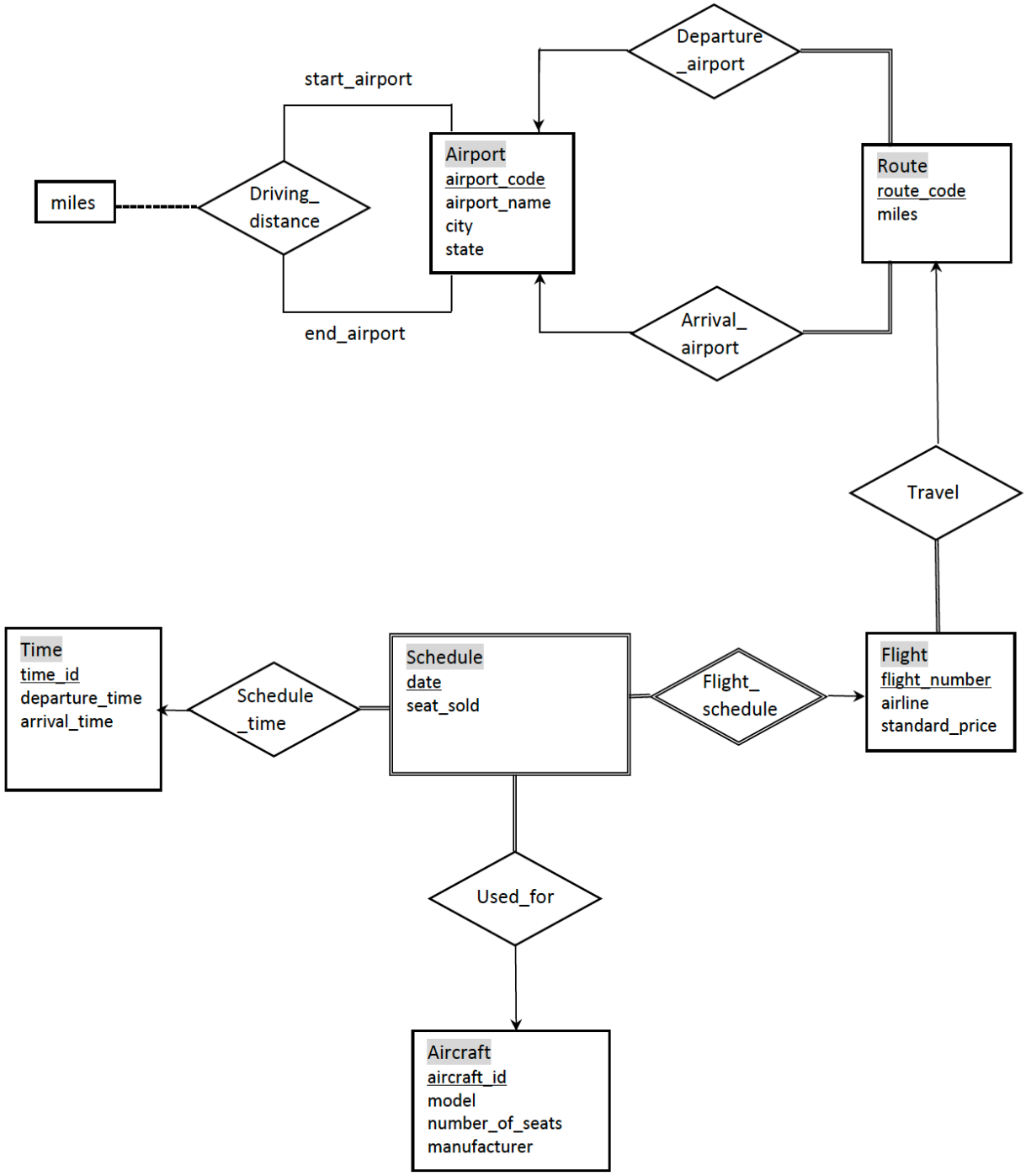
Client (client_id, first_name, last_name, date_of_birth)

Bid (client_id, bidding, willing, departure_airport, arrival_airport, date)
Foreign key client_id references client_id in relation Client

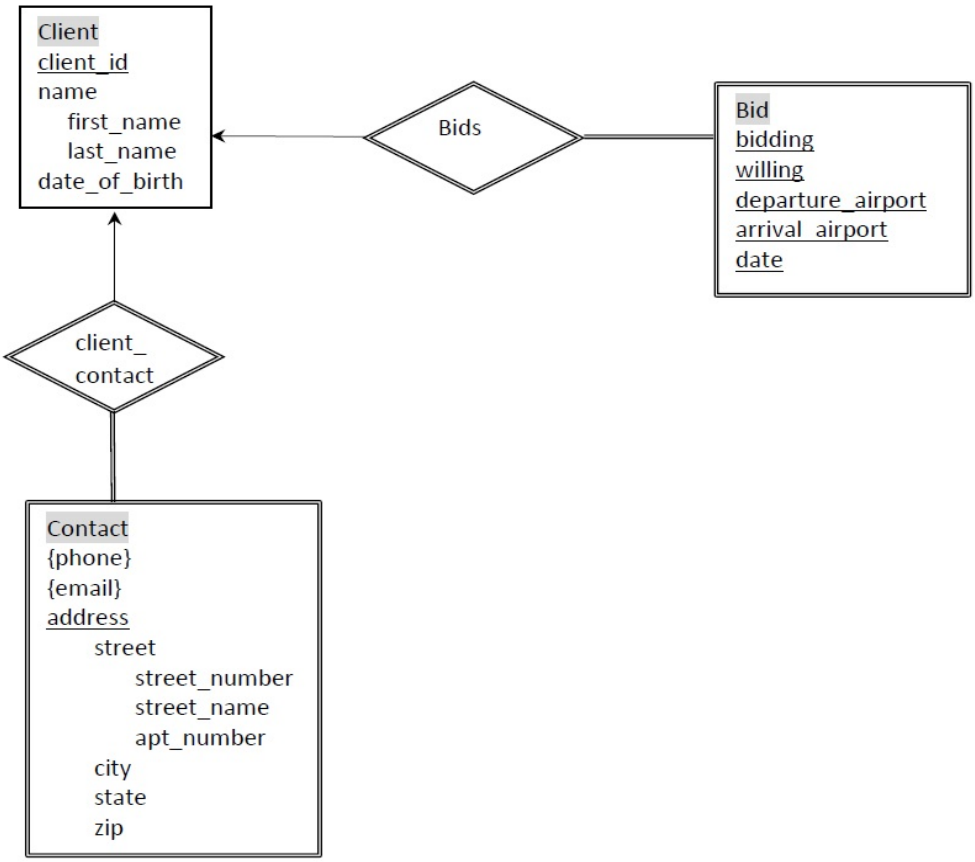
Contact (client_id, phone, email, street_number, street_name, city, state, zip)
Foreign key client_id reference client_id in relation Client.

Our system allows users to make their biddings in the system. In the meantime, they put in their personal and contact information in the system. For every fixed time period, for example a week, we run the winner selection process and allocate the flight ticket. After that, we send out the result to every customer who participate in the bidding. The selection process is illustrated in graph-3.

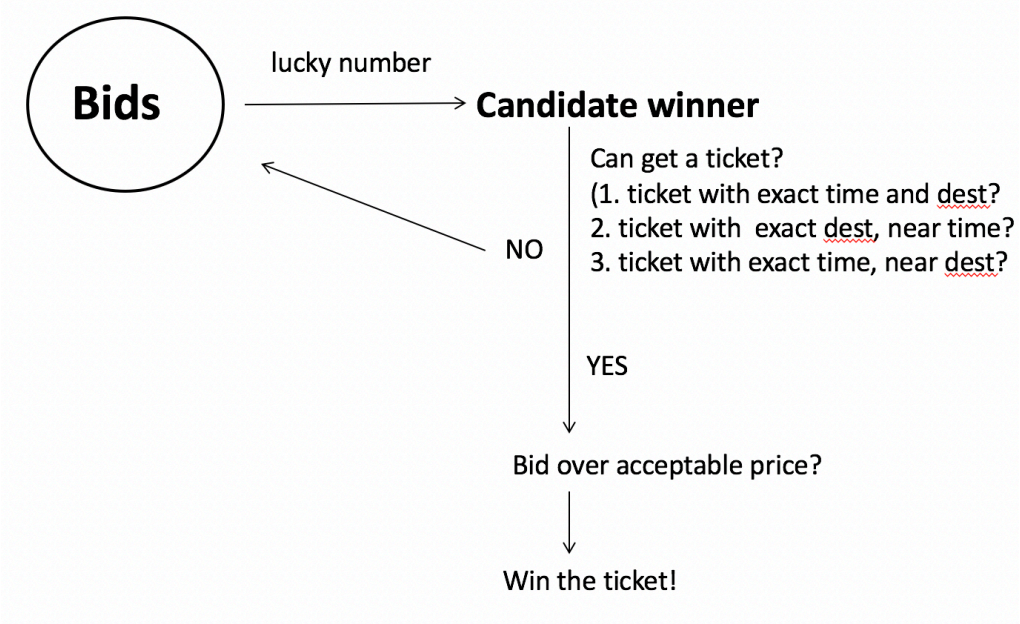
Graph-1 The E-R Diagram of the Flight Information



Graph-2 E-R Diagram of the Bidding System



Graph-3 Winner Selection Process



4. Major Limitations

First, the database only has limited amount of flight information, and therefore we can only select limited number of winners using the data in our system, but we believe it won't be a problem in reality. Second, for people who want to participate in the bidding process, there could be some learning curve since the ticket allocation mechanism is a little bit complicated.

5. Major Issues

The major issue we faced in the project is how to send the results from server to the front end after receiving a client's request and executing the queries. We can build a front end and collect information from a remote client, then access the database and process the returned table at the back end, but we didn't succeed sending the results to the front end.

Another issue is keeping consistency. Our group members use different operating systems: Windows, Mac and Linux. So we ended up testing the codes and checking the formats of data on all three different systems. This also made collaboration difficult. It often happens that something works on one computer but not on another and we had to spend a lot of time figuring out what the problem was.

6. Major Contributions

As discussed in the beginning of this report, there are lots of planes flying with many empty seats every day, meanwhile many people want to fly but couldn't afford the regular tickets. Our bidding system can offer a solution to this issue. This kind of system is exactly what many air companies and ticket agents are looking for, while the low prices are appealing to customers. It would be a trend soon as more and more air companies have realized that selling the empty seats through this way will bring them more profits while regular-priced tickets won't be affected.

7. Selling Point

In our project, we implemented lots of skills and knowledge obtained from the course. For example, we used the SQL queries to choose the winner based on 'willing' and 'lucky number', find alternative routes and augment winner, find flights with exact destination and flexible time (or the exact time combined with flexible destination), and compute the lowest acceptable price. We used the E-R diagram to design and model the whole database, and etc.