

CS 336 -- Principles of Information and Data Management

Fall 2021

Requirements Specification for the Database Programming Project

Introduction

In this project, you will design and implement a relational database system to support the operations of an **online travel reservation system**. You will use HTML for the user interface, MySQL for the database server, and Java, JDBC for connectivity between the user interface and the database server.

You will have to install your own web server that will host your web application (Tomcat) as well as a MySQL server. Everything will be local. Many resources will be provided about how to do everything so don't worry. ☺☺

You are to work in **teams of four**.

Project Specification

As you probably know, there is a multitude of online travel reservation systems on the web. Some popular ones are [expedia](#), [skyscanner](#) and [kayak](#). I suggest that you visit these web sites to get an understanding of the look-and-feel of a travel web site and how such a system is supposed to function.

The basic idea behind your on-line travel reservation system is that it will allow customers to use the web to browse/search the contents of your database (at least that part you want the customer to see) and to make flight reservations over the web. Your web site should allow users to make both domestic and international reservations. It should also allow users to query the database for available flights (direct or indirect) between a pair of cities for a given date.

Actual travel sites allow you to do a lot more than simply make flight reservations. For example, you can book a rental car or a hotel room. Due to time limitations, we will stick to flight reservations.

You will first create an E-R diagram of your online travel reservation system before developing your relational model according to the functional requirements described below.

You are responsible for arranging the data items into tables, determining the relationships among tables and identifying the key attributes. Finally, you should specify and enforce integrity constraints on the data, including referential integrity constraints.

1. Functional Requirements

The users of your system will be the customers (passengers) that use your system to make a flight reservation, customer representatives who provide customer-related services, and the site's admin. You should assume that the computer knowledge of the users is limited, and thus your system must be easy to access and operate.

Every airline company owns a number of aircrafts and it is associated with a number of airports from where it operates. Each airline has a two-letter ID from which it is being identified uniquely. For example, the ID for American Airlines is AA, and the ID for United Airlines is UA. Similarly, each airport has a three-letter ID. For example, EWR, LGA, and JFK are well known local airport codes.

A flight is operated by an airline and a specific aircraft that has specific number of seats, and operates on a given set of days of the week (e.g. every Monday, Wednesday). Flights can either be domestic or international. For every flight, it must be recorded its flight number (unique only within that airline), the departure and destination airports, as well as the departure and arrival time.

When accessing your online reservation system, customers should be able to search for specific flights by providing information about the departure and arrival airport as well as the date they wish to fly. They should also be able to search for one-way or round-trip flights, and they should be able to set if they are flexible about flight dates (+/- 3 days).

Customers should then be able to make flight reservations (buy flight tickets) for the flights they are interested in. Once they buy it, they should be able to see all the information for their flight ticket. If there are not available seats for a specific flight, they should be able to get into the flight's waiting list. A flight ticket has a unique number and is for just a single passenger. Each ticket is associated with a sequence of flights. For example a ticket might be associated with just one flight if it is one-way and direct or with 2 flights if it is direct and round-trip or more than 2 if it has stops (either for one-way or round-trip). Each ticket must include all the associated flights and include: from-airport, to-airport, flight numbers (along with its airline), departure date and time, seat number, and class (economy/business/first). It also has the following attributes: first and last name of the passenger, ID number, total fare, and date and time when ticket was purchased. In case the class of the ticket is economy, the customer should not be able to change/cancel their ticket unless a fee is paid. For business/first class, customers should be able to change their ticket with no fee. Finally, a flight ticket also has an associated booking fee, which is how your company makes money.

A customer may partake in any number of flight transactions and s/he is associated with one account (customers have to register in the system and they also have to be logged in to make a reservation) which includes a reservation portfolio, indicating all the flight history held in this account (past flights and upcoming).

Your online reservation system should have the following functionality:

Customer-Level Functionality

Customers should be thought of as online airline ticket buyers and should be able to easily browse your online travel reservation system on the web and buy flight tickets. In particular, they should be able to first search about the following type of flights:

- One-Way on a specific date
- Round-Trip on specific dates
- One-Way/ Round-Trip on flexible dates (+/- 3 days)

A customer should also be able to:

- browse the available resulting flights
- sort flights by different criteria (price, take-off time, landing time, duration of flight)
- filter the list of flights by various criteria (price, number of stops, airline, take-off times, landing times)
- make flight reservations
- cancel their flight reservations (if it is business or first class)
 - if there are people in the waiting list, they should receive an alert that there is an available seat, and be able to proceed with paying their ticket and complete the reservation.
- enter the waiting list if the flight is full
- view all the past reservations with their details
- view all the upcoming reservations with their details
- be able to browse and send questions to customer representatives
- be able to search questions and answers by keywords

Admin-Level Functionality

The admin should be able to:

- Add, Edit and Delete information for a customer representative or customer
- Obtain a sales report for a particular month
- Produce a list of reservations by flight number or by customer name
- Produce a summary listing of revenue generated by a particular flight, airline or customer
- Determine which customer generated most total revenue
- Produce a list of most active flights (most tickets sold)

Customer-Representative-Level Functionality

Customer Representatives should be thought of as reservation agents and should be able to:

- Make flight reservations on behalf of users

- Edit flight reservations for a customer
- Reply to user's questions
- Add, Edit, Delete information for aircrafts, airports and flights
- Retrieve a list of all the passengers who are on the waiting list of a particular flight
- Produce a list of all flights for a given airport (departing and arriving flights)

2. User Access Control

Your database system should provide controlled access to the data by distinguishing between the different types of users: admin, customer representatives, and customers.

- Customer Representatives should not be able to perform manager-level transactions;
- A customer should not be allowed access to other customers' account information, or to any employee information.

3. User Interface

HTML and its successors provide facilities for creating pop-up and pull-down menus, value lists, input/output forms, labels and customized reports. You should make use of all of these capabilities, and in the process come up with a system that caters to users with only limited computer knowledge. We don't care about having a beautiful UI. It only has to work!

Good luck! ☺

Final Submission Files

You should submit the following 6 files, the file names of which should be preceded by your group number (e.g. Group15_projectCode.zip). You should submit only once, under one person's sakai submission.

- 1) #projectCode.zip: a tar/zip file of all the project code the group has written. Please submit all your eclipse project from your workspace (all your .java, .jsp, .html files etc.) NOT your .war file.
- 2) #schema.sql: the sql file for your DB schema (you have to export it from your MySQL workbench)
- 3) #ER diagram: your final ER diagram
- 4) #ProjectChecklist.docx: This file is under your project resources.
- 5) #README.txt: please provide the admin and customer representative credentials. In addition you should mention anything you want us to know about your application or group.
- 6) A demo video where you show step by step all the functionality you have implemented (you can use any screen recording video application you like).

IMPORTANT: In case things are unclear or we have concerns about your project, you should be ready to demo your application live for us.

DEADLINE: Sunday, December 5 at 11:59pm

Good luck! 😊

IMPORTANT DATES

October 13: ER diagram due date

October 24: Relational schema due date

November 7: Login/Register page due date

December 5: Final project due date