CS 336 -- Principles of Information and Data Management

Fall 2020
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 railway booking system**. You will use HTML for the user interface, MySQL for the database server, and Java, Javascript, and JDBC for connectivity between the user interface and database server.

You will have to install your own virtual machine with a web server that will host your web application as well as a MySQL server. Everything will be under the AWS (Amazon Web Services). Many resources and recitations 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 railway booking systems on the web. The basic idea behind your on-line railway booking 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 train reservations over the web. It should also allow users to query the database for available train schedules (direct or indirect) between a pair of cities for a given date and "approximate" time.

Your database system must be based on the specifications and requirements that follow.

1 System Users

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

2 Required Data

The data items required for the train reservation database can be classified into six categories: *trains*, *stations*, *train schedules*, *reservations*, *customers* and *employees*.
The above classification does not imply any particular table arrangement. 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.

You will first create an E-R diagram of your online railway reservation system before developing your relational model. Details of this assignment will be forthcoming.

2.1 Train Data

Your system should store information about trains. Every train has a unique four-digit number id.

2.1 Station Data

A station has a unique id, a name, the name of the city this station is, and the state.

2.3 Train Schedule Data

The basic information related to this data is:

1. Transit line name (e.g. Northeast Corridor)
2. Train
3. Origin
4. Destination
5. Stops
6. Departure datetime
7. Arrival datetime
8. Travel time
9. Fare

All trains perform a specific route based on a unique transit line throughout the day. Depending on the transit line, the trains start from a specific origin and arrive at a specific destination station. For example, the Northeast Corridor transit line has origin the New York Penn station and destination the Trenton station. Every transit line starts from its origin on a specific time and arrives at its destination at another specific time. It also has a number of stops, with each stop having its own arrival and departure times. For example, the Northeast Corridor line with train #3806 starts at 3:48am from Trenton, and arrives at NY Penn Station at 5:21am with a total travel time of 93 minutes. While performing this route it stops to 10 stations which are Princeton, New Brunswick, Edison, Metuchen, …, etc. and each of these stations have their own departure/arrival time. A route has also an associated fare which depends on the type of the route (if it is one way, or round trip), as well as fare discounts for children, seniors and disabled.. You can assume that every transit line has a fixed fare from origin to destination and then the individual stops have a fare=fare/number of stops in this line. For example, the Northeast Corridor line (which goes from Trenton to NY Penn) has a fare of $50. Then if we assume that it has 10 stops in between (e.g. Trenton, Princeton, New Brunswick, Edison, Metuchen, etc.), the fare between two consecutive stops would be $50/10 = $5, between two stops (e.g. NB-> Metuchen) $10 etc. Children have a

discount of 25%, seniors 35% and disabled 50%. Round trip tickets have double fare.

2.4 Reservation Data

This category of data should include the following items:

1. Reservation Number
2. Date
3. Passenger
4. Total Fare

A **reservation** has a unique number and is for a single passenger. Each reservation has an associated origin station, destination station and transit line name (along with its train number), departure date and time. A reservation also has the following attributes: total fare and the date when reservation was made. For example, Mr. John Smith makes a reservation on 7/25/2020 for the Northeast Corridor train #2345 that goes from New Brunswick station on 8/10/2020 11:00am to New York Penn Station 8/10/2020 11:59am and costs $80.

2.5 Customer Data

The items required for this category include:

1. Last Name
2. First Name
3. E-mail Address
4. Username
5. Password

A customer may partake in any number of reservation transactions. Associated with each account is a reservation portfolio, indicating which reservations are held in that account (past and future).

2.6 Employee Data

This category of data should include the following:

1. Social Security #
2. Last Name
3. First Name
4. Username
5. Password

3 User-Level Functionality

3.1 Manager/Admin-Level Functionality

The manager should be able to:

- Add, Edit and Delete information for a customer representative
- Obtain a sales report per month (total revenue per month)
- Produce a list of reservations by transit line or by customer name
- Produce the total revenue generated by a particular transit line, or customer
- Determine which customer generated most total revenue
- Produce a list of the 5 most active transit lines (most reservations per month) **3.2**

Customer-Representative-Level Functionality

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

- Edit and Delete information for train schedules
- Reply to customer questions
- Produce a list of all customers who have reservations on a given transit line and date
- Produce a list of all schedules for a given station (both as origin and as destination) (e.g. list of train schedules that have New Brunswick as origin, or list of train schedules that have NB as destination)

3.3 Customer-Level Functionality

Customers should be thought of as online train ticket buyers and should be able to easily browse your online travel reservation system on the web, search for train schedules given an origin, destination and date, and make train reservations. In particular, they should be able to make the following types of reservations:

- One-Way
- Round-Trip

A customer should also be able to cancel an existing reservation and they should be able to retrieve the following information:

- A history of all current and past reservations they have made
- Travel itinerary for a given reservation

A customer should also be able to ask questions to the customer service, where the customer representatives should be able to reply (like a forum functionality).

4 User Access Control

Your database system should provide controlled access to the data by distinguishing between the different types of users: manager/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.

5 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. The information you provide to customers should look professional and inviting.

Implementation requirements

The system is to be accessed through a web interface, programmed in jsp, accessing a mySQL database. Details of the technical set up, which uses the Amazon Web Services (AWS), will be covered in recitation. At the end, you will need to provide us with a link, where we can invoke and test your program. This means that if you choose to develop the system on your own machines, you are responsible for making it available at some URL! In order to be fair to everyone in the class, please do not use frameworks or other tools that make programming easier. (If we had time to teach everyone such a framework, we would.)

Good luck! ☺

PROJECT PARTICIPATION RULES:

If someone is **not participating properly** in the project, by not carrying their weight, not attending meetings, not submitting things, it is suggested that the others send emails cc-ing the project grader (Shreyans), so we can keep an eye out when apportioning credit for the project.