

CS3380 SPRING 2017 Final Group Project: RAILROAD CAR SYSTEM

Groups: Should be composed of 5-6 students

Objectives:

- ✓ Create a website for establishing a railroad car system. Your group should agree on a standard look-and-feel and theme for the site, to which all members will adhere, and deploy the site on tc.rnet in the s17cs3380g## account, where ## is your group number.
- ✓ Design a database schema (MYSQL-specific) for the backend of your website.
- ✓ Work together as a team to complete the project.
- ✓ Seamlessly integrate various group members' work as well as different aspects of the web application into a single cohesive web application.
- ✓ Utilize PHP code for relative sequencing and automating web logging activities.
- ✓ Test individual components of the site by populating the database with test data.
- ✓ Produce a coherent report that details the full design, including ERD, user-interaction, multiple user login credentials (at least one for each type of user), algorithms used, and a basic user's manual.

Brief Introduction

The railroad industry in the US is huge. There are over 1.1 million rail cars in the country. In this project you will design a system for a railroad named Missouri Rail. This is a new start up company and you will have to read the requirements of the system (including business rules) and develop a fully functional system complete with documentation. I will provide some "high level" requirements but it is YOUR job, as a team, to ask any questions of me to make sure your project meets MY specific requirements. Read these "business rules" and design your database and application accordingly:

- ✓ Employees – there are 3 types: Administrator, Engineer and Conductor. You MUST use a hierarchical (super/sub) entity for this.
- ✓ Customers – only one type of customer. They make rail car reservations
- ✓ Equipment – rail cars have different attributes including load capacity (weight), assigned serial numbers, types (box car, grain hopper, coal car, caboose, locomotive, flat bed, etc.), location, manufacturer, etc.
- ✓ Trains – these consist of a train number (Train number 303, etc.), they must have a locomotive and cars assigned, they must have 2 engineers, at least one conductor, a departure city (FROM) and a destination city (TO).

Specifications

The web application should support four different user types (administrator, engineer, conductor, customer) and allow access accordingly. The functionalities for these roles are defined below. Only three of these will require authentication (administrator, engineer, conductor).

“CUSTOMER” USERS

Customers should be able to browse/search the global directory of available cars or search/sort by

- a) type of car
- b) location
- c) Price (\$0 to \$999, etc.)

Selecting criteria from the above should send the customer to a list page that displays all the cars in the database that satisfy their selection criteria and the individual information for each car including type, capacity, location, price. When a customer selects a car they must enter company ID, type of car needed, number of cars requested, type of cargo. The price for the cars will be calculated as the price in the database PLUS 5% sales tax. This final information should be displayed to the customer and they have one final chance to approve or discard the reservation. Once they “approve” then the customer record is created and all appropriate tables are updated.

“ENGINEER” / “CONDUCTOR” USERS

1. Engineers should, of course, register to use the web application and must provide their username and password to access the system. You should store all the basic information about an engineer including status (Active or Inactive), total hours of traveling, rank (senior engineer or junior engineer). There should be some means provided to allow an engineer to edit their information. Engineers should also be able to view a log of all their past and current trains they have run.
2. Conductors should have the same access as engineers but different information such as rank (senior or junior). They do NOT need total hours of traveling.

“ADMINISTRATOR” USERS

Administrators will have full access to the site. They should be able to edit information on any user (engineer, conductor, customer), reset passwords, monitor all trains, add engineers, conductors, equipment and trains to the master list when needed.

AUTHENTICATION

You should write a generalized module to perform user authentication against a database table of “userIDs” and encrypted passwords. This module should handle form-layout, logins, logouts, and session management of “userIDs.”. Engineers and conductors will need to be authenticated.

WEB LOGGING

The site should have extensive logging and log utilities for ALL users. All logging should be in a single log table. Logs should contain information like IP address, date/time of action, what SPECIFIC action was taken, who performed the action, etc. Administrators will be able to see the entire log, while engineers, conductors and customers will only see items in the log related to them. The page that allows a user to view the log should include means to filter the log so that specific information may be located quickly (for example, by date of action, by type of action [car update, train, etc.], number of log entries to display, etc.).

Notes and Other Requirements

1. Authentication tables MUST store the passwords in ENCRYPTED form. You will lose 15% of the total points for storing passwords as clear text in the database. The authentication table should be separate from the user table and should contain ONLY userIDs, passwords, and roles. You must provide protection against simple SQL injection.
2. All your webpages should use a common color scheme. Do NOT use plain black text on a white background.
3. All programming or script code should be modular, to increase the readability and to maximize the reuse of code fragments. You should write functions to perform specific tasks. You should also use the PHP “include” and/or “require” directives for attaching PHP code from other files.
4. The PHP scripts must be deployed and working on tc.rnet, and the database must exist in the MYSQL database on tc.rnet.
5. Anything not specified is up to your discretion. What information would you like to see in a site like this?

6. Be creative! START EARLY!!!

7. Regular backups of both code and the database are suggested, for your group's protection.

EQUIPMENT

Equipment refers to the rail cars themselves. They have a freight capacity (measured in pounds), they have an ID number, they have a type (grain car, coal car, hopper, flat bed, etc.), and a location. You also need to know if they are already reserved to prevent 2 customers from reserving the same car.

TRAINS

Trains have a unique number assigned to them, are made between 2 cities, have at least one engine and possibly more assigned to them, an engineer(s) assigned to them, conductor(s) assigned to them, certain days they are run (Mon/Wed/Thu/Sat). Trains are assigned cars based on the "FROM" and "TO" values for the train and any cars. Customers reserve a freight car(s) and are shown the day/time the car will leave and arrive at the destination. Think about it! This will be the most complicated relationship in the system.

OTHER BASIC INFORMATION

- ✓ Administrators can add, change and delete any records in the system for engineers, conductors, customers, trains and equipment.
- ✓ Administrators are solely responsible for assigning engineers and conductors to trains.
- ✓ Engineers can add, change or delete their own information.
- ✓ Conductors can add, change or delete their own information.
- ✓ Customers can make car reservations but CANNOT change them when they have been made.

Milestone 1 (Submission Deadline: Monday, March 6th, 2017 emailed before the beginning of class as attachments. Put your group number in the subject line of the email!).

1. Each team should submit an **ERD** (***ERD_G##_ver1.png where ## is your group number***) electronically (email) in **image** format for the entire final project. This ERD should include any entities/relationships/attributes necessary for the above-mentioned tasks. Comments (feedback) on your **initial** ERDs will be handed out in class on Wednesday, March 8th, 2017.
2. Each team should also write up a project planning document (***ProjectPlanning_G###.txt where ## is your group number***). This should include information detailing which team members are responsible for which parts of the project. The delegation of work should be fairly detailed. You should also include a **detailed** schedule of deadlines for completion of the project to keep your group on track for the duration of the semester. I would also like to see specifics on how your team will handle communication during the project with ideally a regularly scheduled group meeting to make sure team members are keeping up with their individual tasks.

Milestone 2 (Submission Deadline: Wednesday, March 15th, 2017 emailed before the beginning of class as attachments. Put your group number in the subject line of the email!).

1. Complete ERD for the final project (with revisions if necessary).
2. A text file (***CreateTables_G###.sql***) with CREATE TABLE statements (for the MYSQL DBMS) corresponding to your ERD.
3. Updated project planning document.

Milestone 3 (Submission Deadline: Wednesday, April 5th, 2017 emailed before the beginning of class as attachments. Put your group number in the subject line of the email!).

Each group should write up a group progress report that gives the current status of the project (relative to the project planning document) including:

1. Specific tasks that are complete.
2. Specific tasks that are on schedule.
3. Specific tasks that are ahead of schedule.
4. Specific tasks that are behind schedule (as well as how the group plans to catch up).
5. Any group difficulties or anticipated issues.
6. An update, detailed schedule for ensuring the project will be completed by the deadline.

Group Meeting

Each group will meet with the instructor during the week of April 24th, 2017 to discuss your final project and demonstrate it with a live presentation in my office. As we get closer to that date I will publish a schedule of dates/times for each group to stop by and perform this important final task.

Deliverables

1. **Website:** A usable and visually pleasant web application that has all the functionality described in the above-mentioned tasks, including user authentication and web application logging.
2. **Final Report:** The organizational layout is below.
3. **Files/DB Dump:** A copy of all your files (code, CSS, images, etc) and a dump of your database on a thumb drive. All file references in your code should be relative to the current directory so your application can be ported elsewhere on tc.rnet and still function. Your database should be populated with data.

Final Report (due at the Group Meeting on April 24th).

The report must follow the following organizational layout:

1. Introduction, restatement of the problem/design-goals:
 - a. Include links to group project site
 - b. Include multiple engineer credentials and at least one admin credential set
2. Delegation of project tasks. Include the members of the group and what was accomplished by each group member (be specific).
3. Data model (ERD), appropriately documented.
4. **One page (minimum) per major task**, include short description of solution and separate enumerated algorithms for PHP. This should be written by the team member that did the work. Brag about what you accomplished/learned.
5. Usage instruction/user manual. Instructions should be simple enough to allow a computer novice to learn your system.
6. Issues, frustrations, areas of difficulty. What did you collectively learn or get out of the group project? What would you do differently next time?
7. Summary

Grading Breakdown (150 points):

- 10 Milestone 1: Initial ERD, Project planning document
- 15 Milestone 2: Revised/final ERD, create table statements
- 15 Milestone 3: Group Progress Report
- 50 Functionality
- 30 Report

10 My Discretion*

10 Readability/Commenting of Code

10 Proper/Efficient usage of SQL

*These points will be given based on overall design and integration of various scripts into a cohesive application and user friendliness.

Peer Evaluations

You will be asked to evaluate yourself and your group several times during the course of this project. These will be completed and submitted individually. **BE SURE TO PULL YOUR OWN WEIGHT!** Though grades are given on a team basis, individual grades will be adjusted appropriately for poor evaluations and/or issues with individual progress reports. These evaluations **MUST** be turned in to receive a grade on the final project! There are **NO** makeup evaluations if you miss class on that day!

Teamwork Issues

An objective of the class and specifically the group project is to develop your skills at working in teams. Keep communication lines as open as possible, as this will be essential to a successful project. You should keep your teammates aware of your progress and difficulties and ask for help when necessary. Try to resolve any group conflicts among yourselves first. If there are issues that cannot be resolved within the group, bring them to the instructor. Do not wait until near the deadline to start addressing conflicts, as it will probably be too late at that point.

Summary of Due Dates

1. Milestone 1:

- a. What: ERD, Project Planning document
- b. Due: Monday, March 6th emailed to me before class.

2. Milestone 2:

- a. What: Revised ERD, CREATE TABLE statements
- b. Due: Wednesday, March 15th emailed to me before class.

3. Milestone 3:

- a. What: Group progress report
- b. Due: Wednesday, April 5th emailed to me before class.

4. Final Project:

- a. What: Fully functioning website, files, DB dump, and final report.
- b. Due: week of April 24th at our group project presentation meeting.

Start early and have fun!