CMPE 275 Section 2 Fall 2017

Term Project

Last updated: 12/18/2017 (status: published)

California Ultra-Speed Rail (CUSR) is train transport system with 26 stations, A-Z, lining up linearly from north to south in alphabetical order, where A is the northernmost station and Z is the southernmost station. The rails are doubled such that trains going southbound and northbound are on their own rails without sharing. The goal of the project is to build a ticket booking and management system for CUSR. For simplicity, all times in this handout are based on PST, and we do not take daylight saving time into consideration.

Train Capacity and Schedules

A southbound train departs from Station A every 15 minutes, from 6AM to 9PM, inclusive. These trains are respectively numbered SB0600, SB0615, SB0630, …, SB2100. Symmetrically, a northbound train departs from Station Z every 15 minutes too, also from 6AM to 9PM, inclusive, numbered NB0600 through NB2100. For simplicity, it always takes 5 minutes for a train to travel between two adjacent stations, (e.g., E to F). If a trains stops at a station, it stops for 3 minutes for passengers to get off and on.

Each train can accommodate N passengers (N defaults to 1000; please see the Test Assistance section for other cases).

Express vs Regular Trains

Trains departing from Station A (or Z) at whole hours, i.e., {S | N}B{06-21}00, are express trains in that they only stop every 5 stations. For example, SB0900 starts from Station A at 9AM, and will only stop at Station F, K, P, U, and Z. All other trains are regular trains and stop at every station along their way.

Ticket Pricing

The pricing of tickets is based on the travel distance, and in unit of whole dollars. It starts at $1 for regular trains, covering up to 5 stations. For example, a ride on a regular train from Station B to Station M passes 11 stations, thus will be charged with a fare of $3, and a regular train from Station A to Station F is priced at $1. Express train pricing doubles regular train; e.g., suppose

one rides an express train from A to F, and then regular train from F to H, his fair will be $2+$1=$3.

User Registration

A passenger must register for an account first using his email address before he can use the ticket booking system. You must support signing in with Facebook or Google (Oauth) at the user's choice. Feel free to use Firebase Authentication for this purpose.

If you use Oauth, you can either request an auth scope that provides access to the passenger's email address, or ask the user to provide his email address directly.

Search for Trains

You must support the following parameters for ticket search.

Number of passengers:

Between 1-5, inclusive. All the tickets purchased in one transaction must have the same itinerary; i.e., the passengers can stay together through the whole trip.

The default value is 1.

Departure time

The *earliest* date and time to depart. It is OK to use dropdown to select the departure time for a given station and day. Please provide an option (e.g., checkbox with the label "Exact Time") for the user to express that intention to only consider trains departing at the given time.

The passenger is ***not*** allowed to search or buy tickets for a train that departs within 5 minutes from now.

From Station

The starting station of the trip.

To Station

The destination station of the trip.

Ticket Type

Regular, Express, or Any

- Regular: only regular trains are considered.
- Express: at least one of the trains in the itinerary needs to be an express train. This needs to true for the return trip as well if the ticket is for a roundtrip.

- Any: any combination is fine. This should be the default choice.

Number of Connections

- Any: this is the default value. It actually means the trip can make up to 2 connections.
- None: no connections can be made; i.e., this passenger does not need to switch trains for the trip.
- One: up to one connection is allowed.

Please note in the case of roundtrips, the above apply to both the departure and return trips. E.g., if "One" is chosen, the passenger is OK to make up to one connection in the departure trip, and up to another one in the return trip.

Roundtrip

Round trip (or NOT): By default, the trip should be one way. If the chooses to buy a round trip, the departure time for the return trip needs to be provided. The "Exact Time" option for the departure trip will automatically apply to the return trip, so does the ticket type.

The from and to stations for the return trip are the reverse of the departure trip; basically, the options for the departure trip will apply to the return trip whenever possible.

The search results are ordered by the time of arrival, with the earliest arrival showing up first. If there are multiple options with the same arrival time, then short them by the total travel time (the time between boarding the first train and arrival at the destination). You only need to show need to show up to the top five options of itineraries, based on the sorting described above.

Further simplification on train search:

- When a passenger makes a connection, it is NOT allowed for him to wait for more than 2 hours.
- The departure time of the return trip must be within 7 days of the departure time of the departure trip.
- It is OK if you only consider going forward, and not backward when making connections. The capability to search trains in an opposite direction when making connections will get up to 1 bonus point.

In order to purchase the ticket(s), the customer needs to select one of the presented options in the search results, and proceed with the purchase.

You server cannot take too long to serve a ticket search request. Any search request that takes longer than 10 seconds should be treated as a failure from grading (or customer's) perspective.

Purchase of Tickets

In addition to the fare price, there is a $1 transaction fee. Round trip counts as one transaction.

Each purchased ticket needs to contain the following info:

Name(**s**) of the passenger

The # and departure time of each train

The arrival time of the trip (both departure and return trips, in the case of roundtrips)

Purchase Notification

Upon ticket purchasing, an email confirmation with the details of the ticket(s) needs to be send to the customer, the person who purchased the ticket(s). The minimal information to be included in the email includes itinerary, price, and passenger name(s).

Ticket Cancellation

- A ticket can be cancelled ***one*** hour before its boarding time. If multiple tickets are bought in the same transaction, they need to be cancelled together.

Train Cancellation

When a train is cancelled, we need to rebook each passenger (unless there are not enough tickets available meeting the original searching criteria, in which case we notify the ticket purchaser about the cancellation), and give the ticket purchaser the opportunity to cancel one hour before the boarding time. The cancellation of a train and the automated rebooking must happen no later than 3 hours before the original train starts. It is not acceptable to rebook tickets only for some of the passengers with the same original purchase; i.e., the rebooking needs to be one transaction for all passengers as well, or it is a cancellation only.

System Reports

You need to provide the following reports, each of which can be requested separately. The reports need to be directly displayed in the UI. It is not required to make reports downloadable.

- Per Train Reservation Rate: for a given day in the future (including today), provide the occupancy rate for each train based on reservation, including both southbound and northbound. The occupancy rate for a given train is calculated as

    - Average of the occupancy rate for each of the 25 road segments, where the rate for each segment is the number of reserved seats divided by the total number of seats. For example, if all seats are reserved throughout the 26 stations for a train, its reservation rate is 100%. If only one passenger has reserved a seat on that train with capacity of 100, which is is from Station A to C, then the reservation rate is (1/100+1/100)/25=0.08%.
- Daily System Reservation Rate: For a given date range, provide the daily reservation rate for the system for each day within the given rate. The daily system reservation rate is the average of the reservation rate of each train in that day.
- Daily ticket reservation stats: for a given day, provide the following numbers
    - The total number of ticket search requests
    - The percentages of search requests with connection numbers as NONE, ONE, and ANY.
    - For each of the three connection number mentioned above, provide the average server side latency (in milliseconds), from the time the request arrives at the server to the time the result (successful or not) leaves the server.

Non Functional Requirements

You *must* use either a relational database, or Datastore if you choose AppEngine.

If any feature described in this handout is unclear or ambiguous, and you fail to get a clear answer from the instructor or TA, you can use your best judgement to interpret and add the missing details, provided that you clearly document and explain your reasoning in the product report.

Testing assistance

For ease of testing and grading, you need to provide the capability to reset the system, including all booking history and past/future bookings. The only thing not to reset is the user registration; i.e., there is no need for a user to register again.

When resetting the system, the user needs to be provided with the option to set the capacity of trains, in the range of 5-1000, inclusive.

Grouping

This project is group based, with group size up to four people. Once the project plan is submitted, group membership cannot be changed.

Source Code Management

You are recommended to use a Source Control Management (SCM) system to manage your team's source code. This can be a private Bitbucket repository or your local git. During the grading of the term project, you may be asked to provide commit history or any other document to help evaluate each team member's contribution.

Cheating Policy

Your app must be built by yourself, and cannot be based on the code base of any existing app. If you used any code not written by yourself, it must be clearly documented in your README.TXT file, unless it is part of publicly available libraries. *If your app is already used to serve the requirements of any other class, it will not be accepted by this class*. In the case any form of cheating is confirmed, your will get an F grade for this class.

Deliverables and Grading

The project is worth 25 points in total. The actual *due dates* of the deliverables will be specified in Canvas.

Project Presentation and Demo (5 points)

To be presented in class.

- The presentation should cover introduction, high level design, and major features with screenshot. Time limit: 3 minutes.
- You must also do a live demo. The guideline for how to do demos is to be added. Time limit: 5 minutes.
- Grading will be based on successfulness of the demo, the content and clarity of the slides, and the delivery of the presentation

The presentation slides must be submitted through Canvas as a PDF file.

Project Report (5 points)

The report needs to cover the following topics.

1. Motivation and introduction of your app
2. High level and component level design
3. Technology choices

4. Description of features with final screenshots
5. Testing plan executed and results
6. Lessons learned and possible future work

You are recommended not to exceed 20 pages, but you will not be penalized just because the report is too long or too short, as long as the level of coverage for the required topics is reasonable and clear. The report must be submitted through Canvas as a PDF file.

Project App (15 regular points)

Note: the instruction for submission is still *subject to change*.

1. Your must submit all your source code / resource files through Canvas
2. Features correctness, stability, performance, choice of technology and implementation are worth 12 points
3. User interface and user experience are worth 3 points
4. You need to keep your app live for at least a week before we finish grading
5. README.TXT, including
    a. The names, email IDs, and students IDs of the members
    b. The URL to access your app
    c. Any other instruction necessary for the TA to grade the app
    d. Build instructions