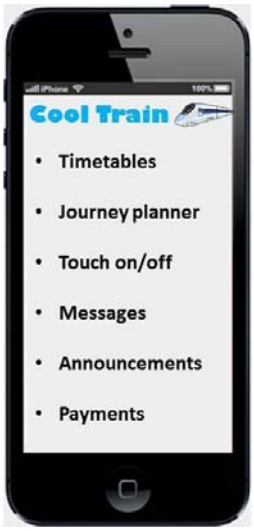
Assignment One – Data Modelling

INFO90002 2016 sem1

The *Cool Train* app



Users of public transport in Melbourne can use smartphone apps to plan their journeys: see for example http://ptv.vic.gov.au/getting-around/mobile-apps/ . You are part of a software startup who are building a new app called *Cool Train* which will offer extra features designed to excite technology-savvy public transport users.

Your programmers will write the software for this new system. As the data modeller, your job is to design a database that can handle the system's storage requirements. The required software and data storage features are detailed below.

To narrow the scope while the new app undergoes acceptability testing, the first version (which are you working on) will only cover trains, not buses or trams.

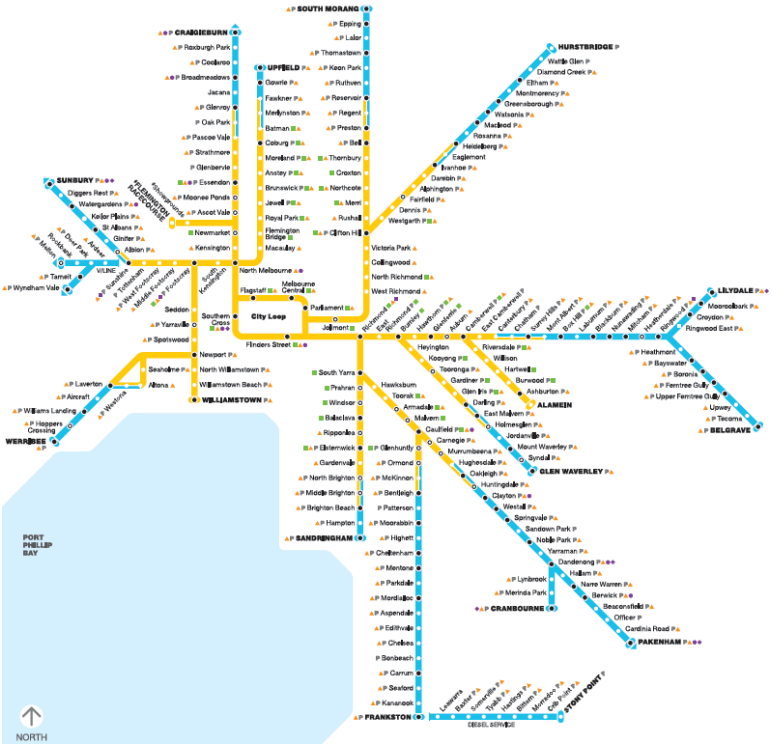
About Melbourne's train system

Melbourne has 218 train stations on 15 train lines in two zones. Some stations belong to more than one train line. Stations are in Zone 1 or Zone 2 or in a small overlap zone (see http://ptv.vic.gov.au/tickets/zones/). Train stations each have a name (e.g. "Melbourne Central"), a street address, and a geographical location used when calculating a user's nearest station. Locations are expressed as a pair of numbers representing a latitude and longitude in decimal degrees, at a resolution of 13 decimal places. For example the location of the Melbourne Central station is (-37.8099387667386, 144.962593535096).

A train service begins at one station and ends at another station, stopping at some or all stations between. There are about ~~10~~70,000 train services per ~~day~~week, each with a unique service number (1, 2, 3, etc.) and a name like "Tues 15:30, Frankston to Flinders St". For each service, we need to store which stations it stops at and at which times.

The schedule of services is different from day to day, but is the same every week; thus we only need to store one week's timetable in the database.

The train system map can be found at http://ptv.vic.gov.au/assets/PDFs/Maps/Network-maps/PTV-Metro-Train-Network-Map.pdf .



Note that the actual train system has been simplified slightly for the purposes of the assignment.

Trains

There are about 500 trains in the Melbourne fleet. Each train is assigned an identifying number, and about each we record its model and the date it was commissioned.

For each train we store the service it is currently operating – this is recorded when the train begins a journey and removed when the journey ends. Trains detect stations as they pass through them, so we also record for each train its current station. We do not need to keep historical records of a train's services and stations: just the current data is enough.

Software features

Cool Train will allow users to:

Display timetables

The app can display timetables for any station or line. When a user selects this feature, they then click on a station or line, the app sends their request to the server, the server fetches the appropriate timetable data from the database, and returns it to the app for display. There is an example timetable below and others can be found at https://ptv.vic.gov.au/timetables/ .

Plan a journey

The app knows its current location from the phone's GPS receiver, and allows the user to indicate their desired destination on a map. These two pairs of coordinates are sent to the server, which calculates the best start and end stations for the user's journey. The server then finds the next train to catch, and sends this data back to the app for display.

Make a journey

Users 'touch on' at the start ~~station~~ and 'touch off' at the end ~~station~~ of their journey, ~~by holding their~~ as their phone passes near a scanner ~~at~~ inside the ~~station gate~~door of the train. This works like existing keycard services such as *Myki*, *Oyster* and *Octopus*, but uses the near field communication (NFC) feature of modern smartphones. Each scan is sent to the server to be recorded. (Trains knows which station they are stopped at, so we record the station id with the scan.) At touch-off, the cost of the journey is calculated, and the customer is asked to rate their journey on a scale of 1 to 5 stars. We record the details of each journey taken, including the fare charged. During the journey, we temporarily store in the customer's record the id of their train they are riding.

Pay for journey

When a journey is completed (the customer touches off), the cost of the journey is calculated and a charge is made to the user's account. The cost of a journey depends on how many zones the customer passes through (see http://ptv.vic.gov.au/tickets/zones/).

~~Customers sometimes forget to touch off: when this happens, 2 hours later the system automatically assumes that their journey has ended, and assigns a default fare.~~

Each ~~user~~ customer has one account, from which their journey fares are deducted. They must keep this account in credit to avoid being fined. Customers ~~Users~~ can add money to ("top-up") their account via online payments. When customers top-up their account, our bank sends our system a top-up notification, saying that a particular user paid a certain amount from a certain bank account and bank-state-branch code (BSB) at a certain time. We need to record these top-up transactions and keep the user's balance up to date.

Social media

To promote a friendly travel experience and drive engagement with public transport, our app will allow customers to interact socially with each other while riding a train.

Some of our social media features make use of a social graph; that is, a system that allows ~~users~~ people to nominate which other ~~users~~ people are their friends. Any customer can befriend any other customer – this involves sending a friend request, which if accepted creates a friendship between this pair of customers. (Friend-requests may be rejected.) Staff are not part of the social graph.

When a customer looks at their app they can see if any of their friends are currently making a ~~train~~ journey. ~~If a friend has touched on but has not yet touched off, the service they are riding is displayed to the user.~~ If a friend is currently riding a train, the station they last passed through is displayed to the user.

Customers traveling on a train can send short anonymous text messages that are visible to all the other passengers on the same train, whether they are friends or not. Example messages might include: “Good morning everyone, the train seems to be running well today” or “I see people carrying umbrellas – is it going to rain?” Each train therefore acts like a chat room for the people currently travelling on it.

Any user – customer or employee – can click “Like” on a message. We must record these Likes, so that the number of Likes a message has received can be displayed under the message. If someone clicks Like they may later click Unlike, in which case we simply delete their Like record from the database.

To preserve decorum, employees can delete any message that is felt to be unsuitable. If a message is “deleted” it is not actually removed from our database, but is simply marked not to be displayed, with a record of which employee marked it so, and when.

Employees do not send ~~or receive~~ messages, but they can send text announcements, which are broadcast to all app users regardless of whether, or on which train, they are currently travelling.

* These social media ideas are inspired by work by Marcus Foth and Ronald Schroeter at QUT – see the following paper, available via Google Scholar and on LMS:

> Foth, M., & Schroeter, R. (2010). Enhancing the experience of public transport users with urban screens and mobile applications. In Proceedings of MindTrek (pp. 33-40).

Other data requirements

Everyone who uses the *Cool Train* ~~app~~ system must register as a user. About each user we record their username (by which they are identified in the system) and password (which they use to authenticate while logging in), their email address, and their first (mandatory) and last (optional) name(s).

There are two types of users of our system - customers and employees. Each user is either a customer or an employee, and cannot be both. ~~(We want to be able to add more user types later without changing our data model.)~~ There are about three million customers and a thousand employees: these numbers are likely to grow. About a million journeys are made per day.

For employees we also record the department they work for. There are several departments including Operations, Human Resources, Accounts, Public Relations and Information Technology. We need to be able to add new departments later without changing the data model.

We must record all customer activity. This means recording each touch-on and touch-off, and each journey~~. We record when and~~ , including at which station ~~usage~~ these events occur.

We also record all usage of our app. We need to record every login attempt, and also every "click"; that is, every time a user (of any type) triggers a request to the server. Because we want to able to analyse this data in sophisticated ways, we will record it in the database rather than in log files. For each click, we record which user clicked, the time of the click, and a character string representing the function called. For login events, we store the username and password entered, the time of the attempt, whether it was successful, and if it was successful, which user logged in.

Extra information

Example timetable

For your convenience, an example timetable is shown here that shows 9 trains leaving Upfield for Parliament. Other timetables may be found at https://ptv.vic.gov.au/timetables/ .

Morning (am)/Afternoon (**pm**) >	am	am	am	am	am	**pm**	**pm**	**pm**	**pm**
Upfield Station (Coolaroo)	10:24	10:44	11:04	11:24	11:44	**12:04**	**12:24**	**12:44**	**1:04**
Gowrie Station (Glenroy)	10:28	10:48	11:08	11:28	11:48	**12:08**	**12:28**	**12:48**	**1:08**
Fawkner Station (Fawkner)	10:30	10:50	11:10	11:30	11:50	**12:10**	**12:30**	**12:50**	**1:10**
Merlynston Station (Coburg North)	10:31	10:51	11:11	11:31	11:51	**12:11**	**12:31**	**12:51**	**1:11**
Batman Station (Coburg North)	10:33	10:53	11:13	11:33	11:53	**12:13**	**12:33**	**12:53**	**1:13**
Coburg Station (Coburg)	10:35	10:55	11:15	11:35	11:55	**12:15**	**12:35**	**12:55**	**1:15**
Moreland Station (Coburg)	10:37	10:57	11:17	11:37	11:57	**12:17**	**12:37**	**12:57**	**1:17**
Anstey Station (Brunswick)	10:39	10:59	11:19	11:39	11:59	**12:19**	**12:39**	**12:59**	**1:19**
Brunswick Station (Brunswick)	10:41	11:01	11:21	11:41	**12:01**	**12:21**	**12:41**	**1:01**	**1:21**
Jewell Station (Brunswick)	10:42	11:02	11:22	11:42	**12:02**	**12:22**	**12:42**	**1:02**	**1:22**
Royal Park Station (Parkville)	10:44	11:04	11:24	11:44	**12:04**	**12:24**	**12:44**	**1:04**	**1:24**
Flemington Bridge Station (North Melbourne)	10:46	11:06	11:26	11:46	**12:06**	**12:26**	**12:46**	**1:06**	**1:26**
Macaulay Station (North Melbourne)	10:48	11:08	11:28	11:48	**12:08**	**12:28**	**12:48**	**1:08**	**1:28**
North Melbourne Station (West Melbourne)	10:51	11:11	11:31	11:51	**12:11**	**12:31**	**12:51**	**1:11**	**1:31**
Flagstaff Station (Melbourne City)	10:54	11:14	11:34	11:54	**12:14**	**12:34**	**12:54**	**1:14**	**1:34**
Melbourne Central Station (Melbourne City)	10:56	11:16	11:36	11:56	**12:16**	**12:36**	**12:56**	**1:16**	**1:36**
Parliament Station (Melbourne City)	10:58	11:18	11:38	11:58	**12:18**	**12:38**	**12:58**	**1:18**	**1:38**

General Transport Feed Specification

The timetable section of our data model can be thought of as a simplification of Google's GTFS model, which public transport operators use to send timetable data to Google Maps. You may wish to read about GTFS in order to find out more about how transport apps work. However the model you submit for your assignment should reflect the specification in this document.

You can read about GTFS at https://developers.google.com/transit/gtfs/ .

You can download Public Transport Victoria's GTFS data from https://www.data.vic.gov.au/data/dataset/ptv-timetable-and-geographic-information-2015-gtfs .

Near Field Communication (NFC)

Read about the NFC capabilities of smartphones at: http://www.techradar.com/au/news/phone-and-communications/what-is-nfc-and-why-is-it-in-your-phone-948410 .

Payment by phone is available in several public transport systems: see http://letstalkpayments.com/7-amazing-transport-based-mobile-payment-systems-not-to-be-left-unnoticed .