

CSC207 Project

Team

- Brian (Brian Tat Quach)
- Loic (Loic Nassif-Lachapelle)
- Dan (Dhanraj Ashwinkumar Patel)
- Danya (Danya Canivet Lette)

Events file (`events.txt`)

The file `events.txt` is comprised of a sequence of commands that can be issued to the transit fare manager. There are three types of commands: Admin, Customer, and Card.

Admin commands

- Calculate total revenue: `Admin: Revenue, Total`
- Calculate revenue on a specific day: `Admin: Revenue, DD/MM/YYYY`
- See all trips taken: `Admin: Trips, Total`
- See all trips taken on a specific day: `Admin: Trips, DD/MM/YYYY`
- See all stations reached on a specific day: `Admin: Stations, DD/MM/YYYY`
- See all routes: `Admin: Routes`
- See all customers: `Admin: Customers, Total`
- Create customer: `Admin: Customers, Create, <customerName>, <customerEmail>`
- See all cards: `Admin: Cards`

Customer commands

- See overview of customer account: `Customer: <customerId>: Details`
- Update customer name: `Customer: <customerId>: Update Name, <newName>`
- List this customer's cards: `Customer: <customerId>: Cards, View`
- Request new card for this customer: `Customer: <customerId>: Cards, New`
- Get average transit cost per month: `Customer: <customerId>: Average Cost`
- View three most recent trips: `Customer: <customerId>: Recent Trips`

Card commands

- See overview of card: `Card: <cardId>: Details`
- See three most recent trips: `Card: <cardId>: Recent Trips`
- Add money (10, 20, or 50 only) to card: `Card: <cardId>: Add Funds, <moneyAmount:10,20,50>`
- See balance: `Card: <cardId>: Balance`

- Deactivate: `Card: <cardId>: Deactivate`
- Tap in at a station: `Card: <cardId>: Tap In, <routeType:Bus,Subway>, <routeName>, <stationName>, DD/MM/YYYY HH:mm`
- Tap out at a station: `Card: <cardId>: Tap Out, <routeType:Bus,Subway>, <routeName>, <stationName>, DD/MM/YYYY HH:mm`

Map configuration file (`map.txt`)

The map configuration file is in `map.txt`.

There are three types of commands in this file: Route, Station, and Hub.

The commands must appear in that order, i.e. all "Route" commands must appear before all "Station" commands; all "Station" commands must appear before all "Hub" commands.

Route Commands

Use a route command to create a new route on this map.

The commands are structured as follows: `Route: <route type>, <route name>`, where route type must be either `Subway` or `Bus`.

In order to create a subway line or a bus route in this map, add a command to the "Route" block of `map.txt`. A command starting with "Route, Subway" will create a new subway line. A command starting with "Route, Bus" will create a new bus line.

Example

To create a subway line called "University", followed by a bus line called "College", write

```
Route, Subway, University
Route, Bus, College
```

Station Commands

Use the station command to create a new station on a route on this map.

The commands are structured as follows: `Station: <route type>, <route name>, <station name>`.

In order to create a station station or a bus stop on this map, add a command to the "Station" block of `map.txt`. Recall that all "Station" commands must appear *after* all "Route" commands.

Creating Hubs

Use this command to make a set of stations adjacent to one another.

In practice, one station being adjacent to another just means that it is permissible to transfer between those two stations on one trip. Recall that all "Hub" commands must appear *after* all "Station" commands.

The commands are structured as follows: `Hub: <route type>, <route name>, <station name> | ...`
`| <route type>, <route name>, <station name>` A hub may have any number of stations.