Project: MBTA_lite

Group name: BishtR

Group Member: Rajat Pratap Singh Bisht

Northeastern University

Khoury College of Computer Science

CS5200-Project Report

December 8, 2023

1. Project Overview

In this project I have worked on MBTA's Subway-Train system. This includes 3 'Metro' lines and 2 'Light rail' lines. Metro lines include 'Blue' line, 'Red' line and 'Orange' line. The Light rail lines include the 'Green' line and 'Ashmont-Mattapan' line. This project will start by implementing the 'Red', 'Blue' and 'Orange' lines first because these subway lines do not contain divergences i.e., splitting of one line into multiple new lines. In the latter part of the project, 'Green' line and 'Ashmont-Mattapan' lines will be considered.

I am submitting an application named "**MBTA_lite**" which can be used in two general modes: a) 'staff-mode', and b) 'guest-mode'. Staff-mode allows the staff members to log in to the system and be able to add, update, delete and read data from the database based on the privileges associated with that particular staff member. Guest-mode is used by a traveler wishing to use MBTA_lite to go from one station to another. Thus, Guest-mode allows the use of application without logging in and provides a connectivity path of all the stations between the queried two stations.

Staff-mode will be used by mangers of subway-lines using their credentials. This will allow the manager of a particular line to make changes to the data associated with that particular line only. Information of other managers and subway lines will also be available, but only in read-only mode. Manager can also keep track of all the trains that run on the managed line. Manager can add new stations or remove condemned stations from the line they oversee.

Guest-mode will provide the guest to query for the path they should take to reach one station from other. In this mode the user will provide the station names as well as the line this station falls on. Guest mode also allows to check for all the stations on a line as well as all the lines available on MBTA_lite application. All the operations from the Guest mode are read-only operations.

2. Database Description

The Rail-network identifies different train routes as color lines. Each lines have a unique name, an associated color, starting and end stations information. There are stations associated with these rail lines. Each line will have at least one station in it and also stores information about its start station and end station (in case of a line with only one station, both can be same). We are also keeping record of the type of lines MBTA offers. Each type of line has a certain speed limit and a unique type_name.

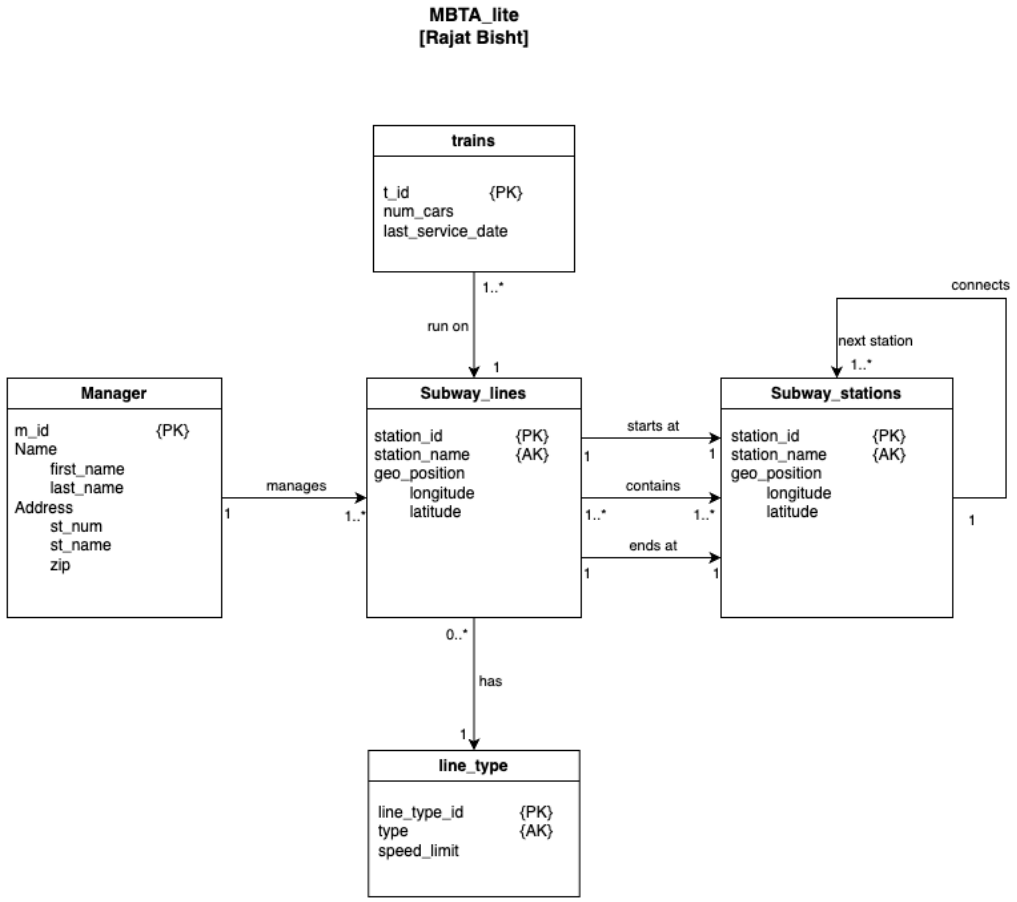
Each station will have a unique id, unique name, longitude, and latitude position. Each station can belong to one-to-many rail-lines since there will be junction stations which are

common to two or more lines. All the stations on these lines follow a sequence of connectivity where each station has a next station on the train-lines.

These subway lines are managed by line managers. A manager will have a unique employee id, name and office location. Each manager can manage multiple subway lines, but a subway line can be managed by only one manager. We are also keeping track of usernames and passwords to keep track of login details. Each username should be unique always

There are also subway-trains that run on these subway lines. Each subway-train has a unique id, number of cars it supports as well as keeps track of its last service date. A train is associated with only one line, but a subway line is supported by multiple trains.

3. UML Diagram



4. Database and Design

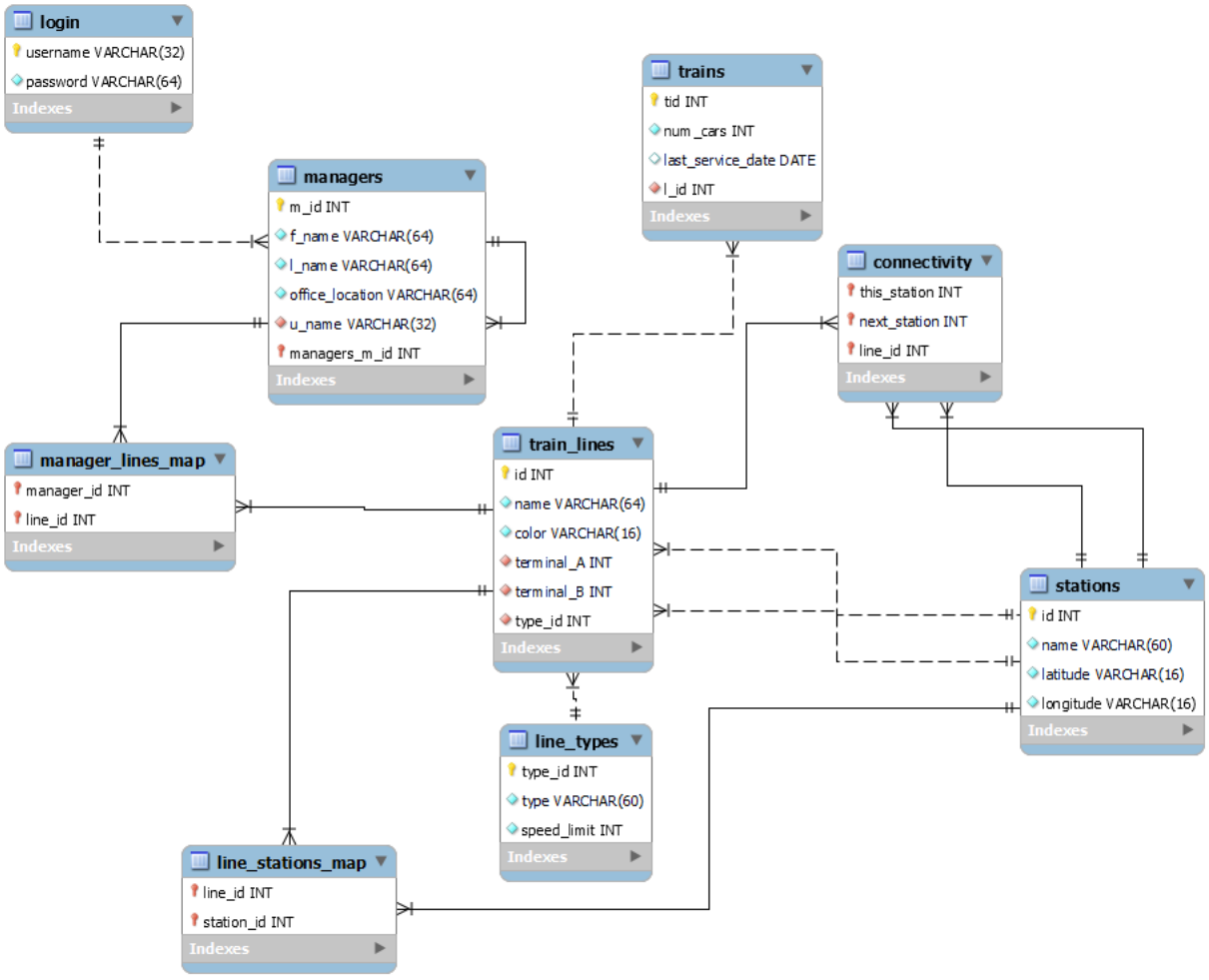
I haves created 'MBTA_lite' database using MSQL relational database management system (RDBMS) to implement the relational data structures presented in the UML diagram above. SQL commands will be used to make queries to and from the database. In the current design, there are 9 total tables:

- Managers table keep track of all the managers that can use it using staff mode.
- We also have login table that keeps login password details. This information is used as token wherever a manager tries to do an operation.
- I also have a train_lines table that keep track of all the information regarding lines.
- Stations are simply a log of unique station names and their geo-locations.
- Since, there is a many to many relation between train_lines and stations table, I have created a mapping table for it. I am also keeping track of the connectivity by adding a connectivity table that keeps track of all the stations in the [this_station, next_station] format. This allows me to use the data as doubly-linked list.
- Train types can increase therefore to maintain 3rd normal form, I have it in a separate table.

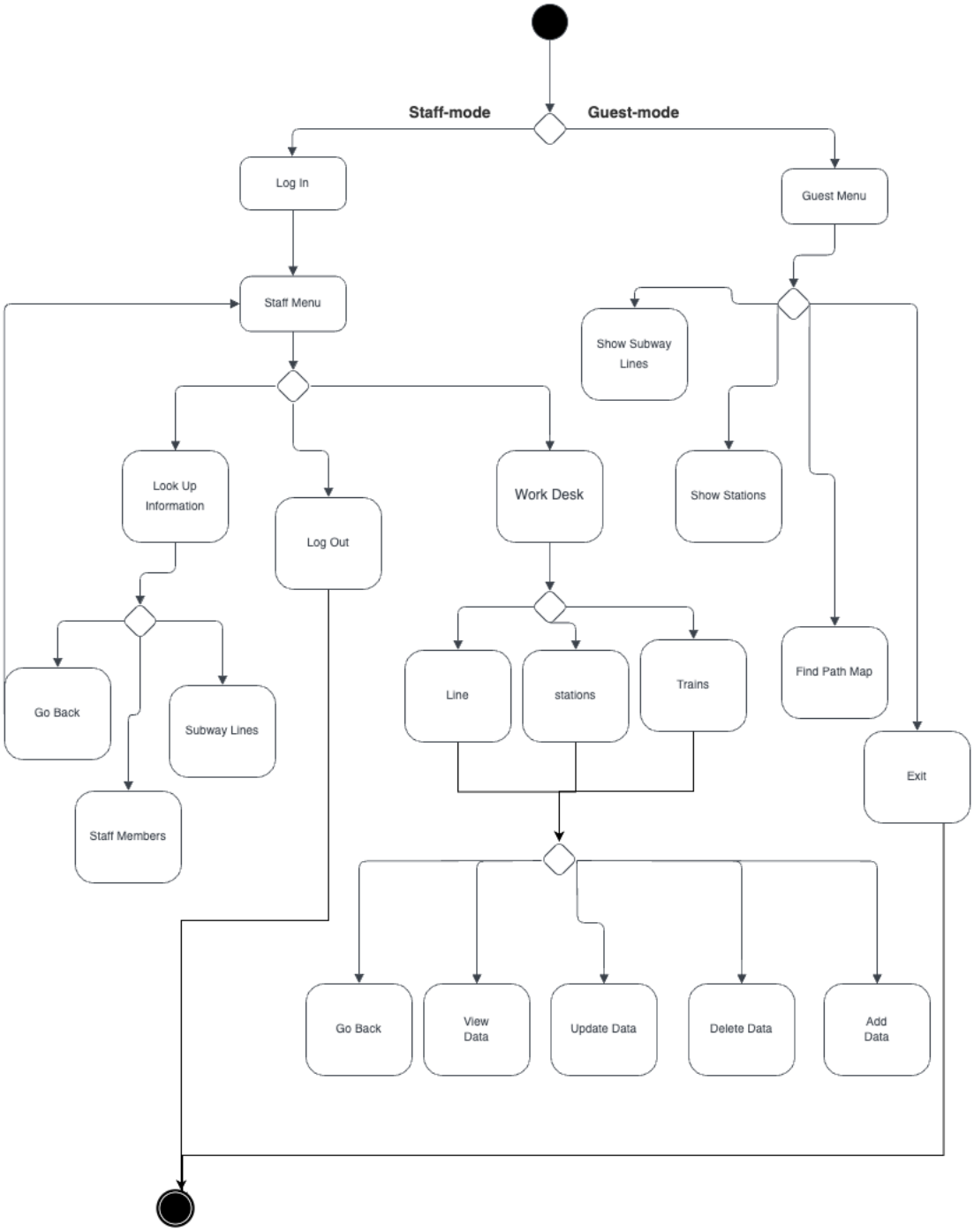
Procedure:

I have made use of procedures to do all the CRUD operations in these tables. This allows me to provide an interface where no user can touch the tables directly, thus making data secure and safe. I have a special procedure "get_path_same_line" that makes use of 'recursive CTE' and generates the path between two stations on the same line from the connectivity table using a similar approach which is used in graph traversals. Most of the procedures are maintain the atomicity of the transaction by getting information from the user and updating required tables as necessary.

Another key point of my design is that no primary key (other than train's tid) is made public to the user of this database. All the procedures make use of the unique names and identifiers to contact the database, where as all the connectivity in between the tables use primary keys as foreign keys,


login
username VARCHAR(32)
password VARCHAR(64)
Indexes
trains
tid INT
num_cars INT
last_service_date DATE
l_id INT
Indexes
managers
m_id INT
f_name VARCHAR(64)
l_name VARCHAR(64)
office_location VARCHAR(64)
u_name VARCHAR(32)
managers_m_id INT
Indexes
connectivity
this_station INT
next_station INT
line_id INT
Indexes
manager_lines_map
manager_id INT
line_id INT
Indexes
train_lines
id INT
name VARCHAR(64)
color VARCHAR(16)
terminal_A INT
terminal_B INT
type_id INT
Indexes
stations
id INT
name VARCHAR(60)
latitude VARCHAR(16)
longitude VARCHAR(16)
Indexes
line_types
type_id INT
type VARCHAR(60)
speed_limit INT
Indexes
line_stations_map
line_id INT
station_id INT
Indexes

5. Application Activity Design



6. User Application

I am proposing a Command Line Interface written in Java to implement the application side of this project. I will be following MVC(Model-View-Controller) design pattern to implement this. MVC model will allow me to implement and encapsulate all the back-end programming which included communicating with RDBMS, credential authentication, permissions, etc. in the Model part.

The Controller has a two jobs in this MVC design. Firstly, it is used to interact with the back end of the application, in which all the features mentioned in the above Activity Design will be implemented. Secondly, it passes on the information retrieved from the back-end model to the View of this application. This will allow modularity in my application design.

View is a simple a Text-based Command line Interface, similar to Bash Terminal. The design choice to implement only Command line interface is due to my limited knowledge in the designing field. Although, I am currently learning how to write a GUI (Graphical User Interface), if time permits, I did **try** to implement a GUI for the final design. But due to lack of time and me being in a one-person team, wasn't able to do GUI in time.

7. Inspiration

Massachusetts Bay Transportation Authority (MBTA) is a Massachusetts State owned public transportation agency, in charge of overseeing Subway-Trains, Buses and Ferries in the state of Greater Boston, Massachusetts. According to the latest stats on MBTA's official website [1], Subway-trains are responsible for 48% of total ridership in the month of September 2023.

I have constantly made use of Subway train public transport service provided by MBTA for more than a year. As a student, it has been an easy, reliable, and inexpensive mode of travel. I was also curious about the huge database management required at an industry level, especially to make such a complex enterprise run smoothly. I am looking forward to this project as learning opportunity since this project will allow me to implement Linked list and graph theory from computer algorithms to a real-world application.

8. README:

To launch the application:

- Use the code.jar file associated with the submission.
- User will be asked to put in mysql username and password.
- Rest of the information is fairly simple and can be seen in the video.

Database dump:

- Sql file is attached in the project.

9. References

[1] https://www.mbta.com/performance-metrics/ridership-the-t