CS 5200 - Database Management Systems
Project Final Report
MBTA Management System
Group Name: LiklikadceNYuS
Group Members: Nano Liklikadze, Silvia Yu

README

To build and run our project, you will need python, pip, and PyMySQL library installed. If you already have these specifications then skip to the next steps in the next paragraph. The first step is to install python3. This can be done by visiting python.org. Note that to check which version of python you have installed, open the terminal or command line and type "python --version" or "python -V". Next you have to install pip. To do so you would type the following in the command line: "sudo apt install python3-pip". Similarly, to check which version of pip you have installed, use the following command "pip --version". Now, you need to install the PyMySQL library. This can be done by either of the following in the command line: "pip install PyMySQL" or "pip install PyMySQL[rsa]". Note that the latter includes the installation of cryptography which is needed if your MySQL was set up with sha256_password or caching_sha2_password. Then to check if you've correctly installed PyMySQL, use the command "pip list". After running this command there should be a list on the screen. If the list has PyMySQL in it then you have successfully installed the library. Note that to check if you have successfully installed PyMySQL[rsa], the list should include both PyMySQL and cryptography.

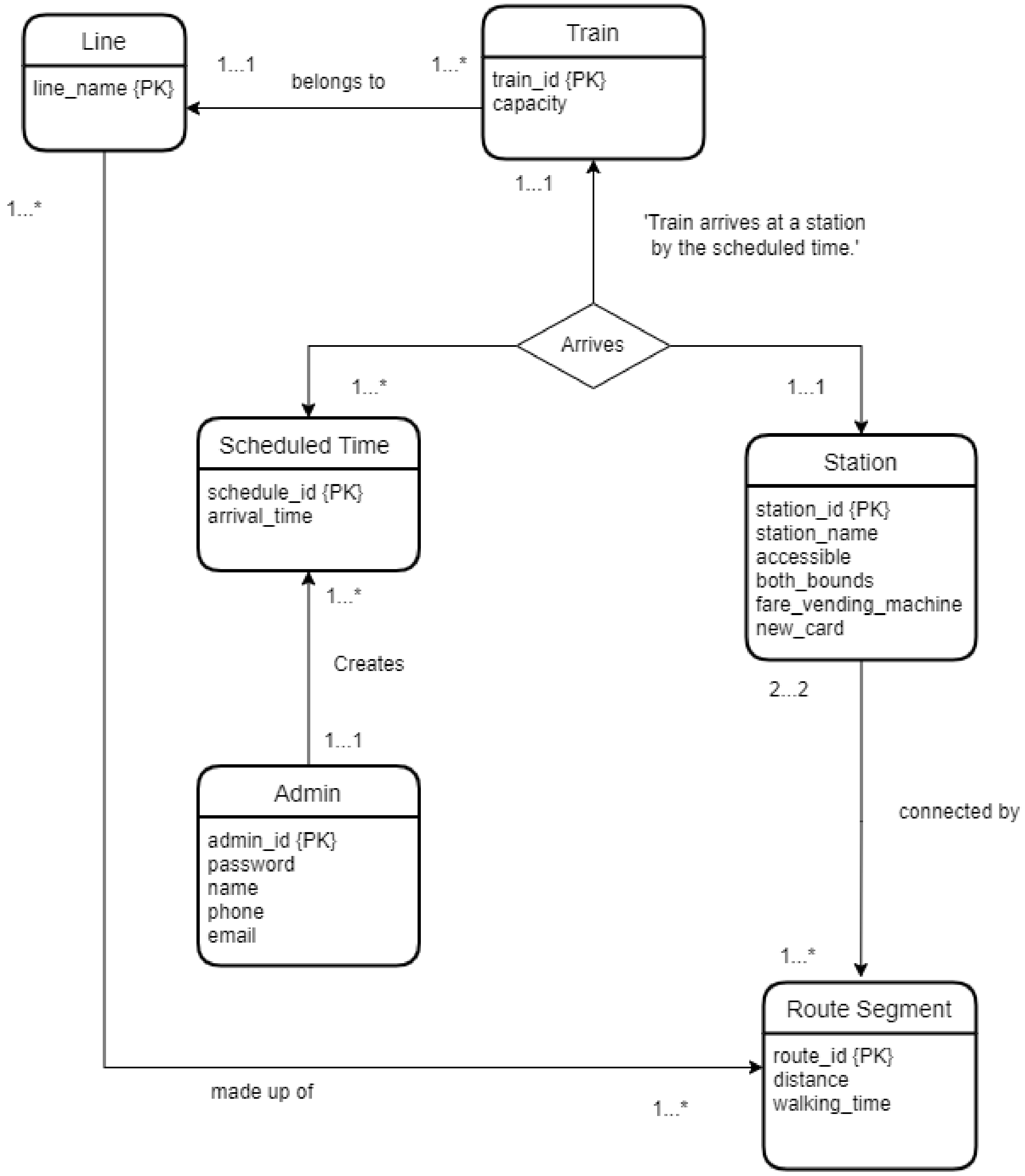
The next step is to import the given self-contained dump file, "LiklikadceNYuS_dump.sql" . To do this open the dump file with your MySQL connection and then run/execute the whole script. Then to check if you have successfully created the database in your workbench, click the refresh button near Schema and find "mbtadb".

The last step would be to run the application code. The application code can be run either using the command line or a python ide of your choice. If you are using the command line make sure you are in the directory where our python application file, "LiklikadceNYuS_app_code.py" , is. Note to navigate to the correct directory use the "cd" change directory command. Once you are in the correct directory, simply type "python LiklikadceNYuS_app_code.py" or "python3 LiklikadceNYuS_app_code.py" depending on which version of python you have.

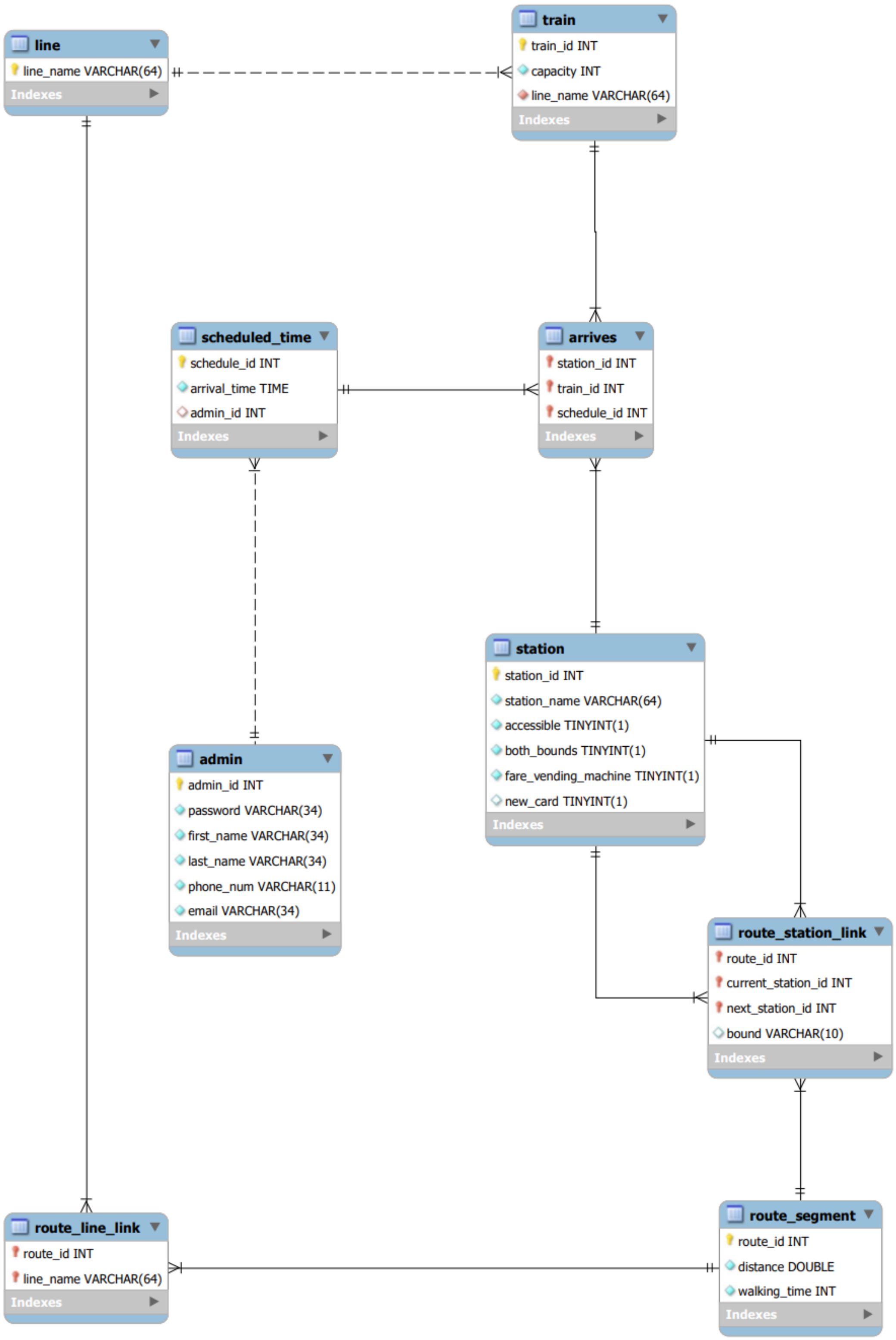
Technical Specifications

For this project, we used SQL and MySQL workbench to create and manage a database. For the front end client application we used python. More specifically, we used the PyMySQL connector library to connect our application to the database. Our project has no known machine restriction.

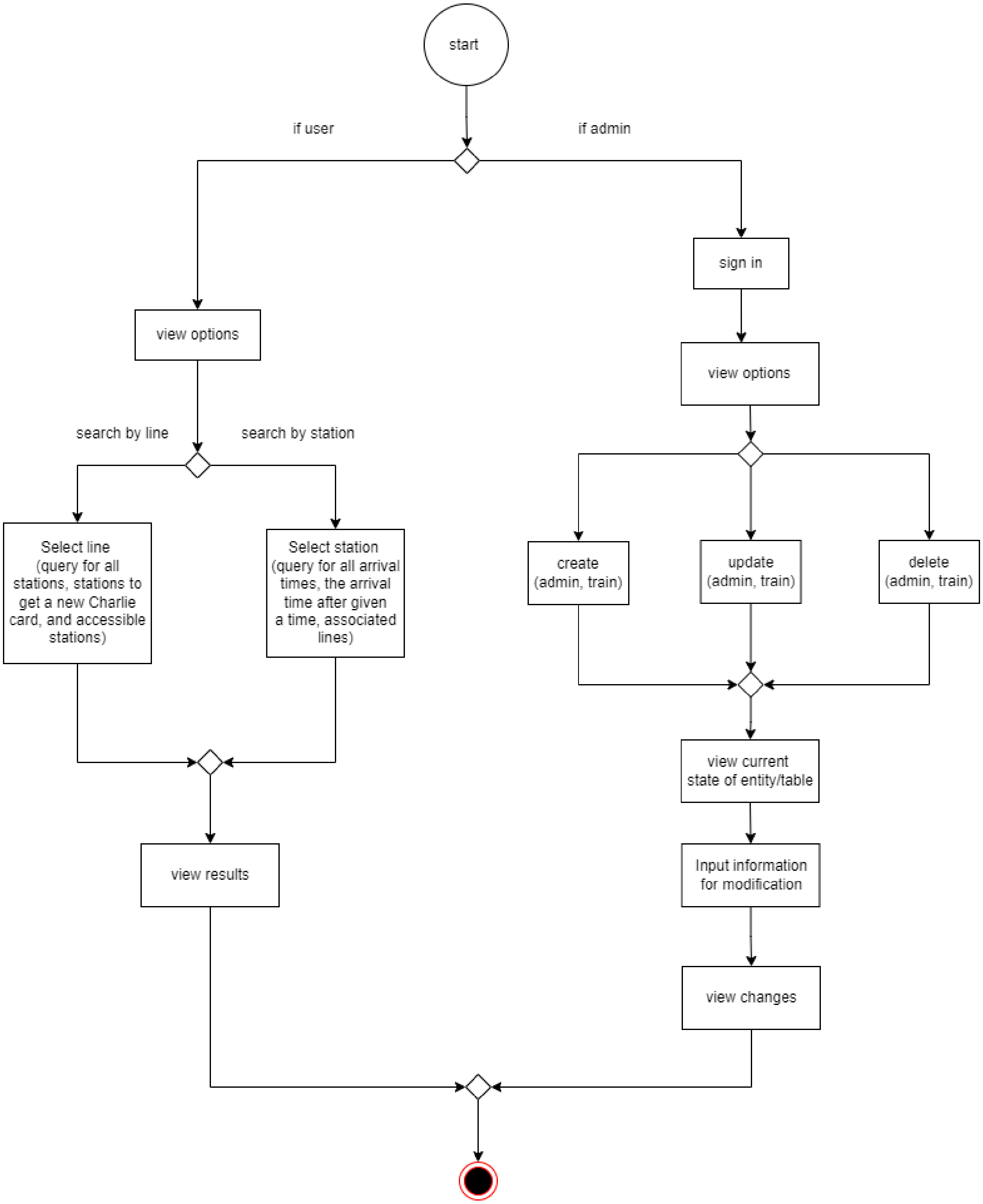
Conceptual Design (UML Diagram)



Logical Design (Reverse Engineer of final schema)



Activity Diagram (user flow of the system)


start
if user
if admin
view options
search by line
search by station
Select line (query for all stations, stations to get a new Charlie card, and accessible stations)
Select station (query for all arrival times, the arrival time after given a time, associated lines)
view results
sign in
view options
create (admin, train)
update (admin, train)
delete (admin, train)
view current state of entity/table
Input information for modification
view changes

Lessons Learned

While working on this project, we solidified and furthered our knowledge of Database Management Systems as we built the entire MBTA database from scratch. We carefully considered the necessary tables and also developed procedures, functions, and triggers. Furthermore, we learned how to connect our database to our front end user application using pymysql. This project provided us with a valuable opportunity to understand the real-world significance of database management systems, and it was rewarding to be able to apply the knowledge we gained in a course to a problem of our interest.

We began working on this project as soon as we received feedback on our project proposal. Our first step was to create the necessary tables and develop our database schema. We started working almost every day after our third exam and divided tasks among ourselves, while regularly checking in with each other after reaching important milestones. Overall, we are pleased with our time management and approach to this task.

Currently, we are using the command line as our client-side interface, and it's working well. However, given more time, we would like to take this project to the next level and develop a user-friendly web application with a clear graphical user interface (GUI). With this, we could enhance the project by adding more procedures and improving CRUD operations.

Lastly, we would like to mention that all our code works as expected so we do not have any code that’s not working.

Future Work

Some of the functionalities that we originally planned to implement have been left for the future because of the time limitation for this project. As mentioned in the section above, given more time we would integrate more of the SQL user defined procedures, functions, and triggers that we have written into the application code and provide more functionalities for the user. These SQL objects can be found in the self contained dump file provided in the project submission. Additionally, some of the entities could be enhanced such as the station entity. Currently, instances of this entity represent just one of the entrances/exits of a particular station. It can potentially be changed by adding other attributes such as address and whether this station leads to all trains or only a certain bound, i.e. inbound train or outbound train. This change will allow for users of this database to query whether a specific entrance of a station will lead to all trains or specific trains.

Another idea that we had for the future was to expand this database to not only include information regarding the MBTA system but also to have other entities such as landmarks and museums. As a city that’s rich in history, Boston has a lot of historical landmarks and museums which might be places that MBTA riders plan to visit so it might be helpful to connect the stations with nearby landmarks and museums. Another expansion could be to include other modes of transportation within the MBTA system such as buses (including the Silver line), the commuter rail, and ferry.

With that said, the current database can be used to query for information such as which stations are accessible, have a fare vending machine, have service agents whom riders can get a new Charlie card from and whether the station leads to all trains or a certain bound. Most of the data in this database accurately represents aspects of the lines (except for the Silver line which is not in our database), trains, and stations in the MBTA system. However, the arrival time of a train at a certain station and all data regarding trains are not accurate. Since the trains do not follow a defined schedule, it would be hard to accurately portray the arrival times. To acquire a relatively accurate portrayal of the arrival times, a potential idea would be to collect past information about the arrival times of trains and analyze it. Note, the commuter rail and ferry have defined schedules that they follow. So, if those entities were added to the database, then it would be easier to accurately portray their arrival and departure times. A potential user for this database could be newcomers to Boston such as college students. It can be used to help them familiarize themselves with the train system.

Lastly, this database can be modified and applied to other transportation systems such as the NYC transit. It would be interesting to see the differences between these two systems and could potentially help improve either systems.