In order to build the database “update_data.bat” must point to the postgresql data folder and be run.

“build_jar.bat” will build a portable jar executable that “start.bat” will open a console window for.

The .class file will error if asked to rebuild queries/functions as it will look for files in the jar.

Username “exit” will exit from the login screen

Any other username/password including blank will work

In most cases there is no error catching for input as it is expected, as this is not a customer facing program, that you will enter only valid options

Main Menu:

1. Customer operations:
    1. Add customer: takes customer data and returns a CID if successful
    2. Edit customer data: takes a CID, a column name, and a value
        a. If you don’t have access to the column names it is viewable by going to Database operations -> Import data to db -> select passenger
    3. View customer data: takes a cid, returns customer data
2. Trip search/Reserve:
    1. Single route search: simple search on one route from a station to a station on a day, various sort options given:
        a. Order by stops asc gives a list of routes ordered by the number of stops between the 2 stations, excluding the first, ascending
        b. Order by stations desc gives a list of routes ordered by the number of stations between the 2 stations, excluding the first, descending
    2. Combination route search: (not working right now) search on one or more routes between 2 stations on a day, sortable by same options as single route search.
    3. Reserve seats: (for simplicity’s sake books passenger along entire route instead of between 2 stations) takes a CID, route id, and the starting time of the whole route in the schedule. If the time required is not known it is viewable by going Other searches -> Find origin time from station and departure time (which does exactly what it says). The same cid can be reserved on multiple routes without returning to the menu.
3. Other searches:
    1. Find trains at station at time and day: lists trains that will be at a station at the given time and day
    2. Find routes that use > 1 rail line: does what it says
    3. Routes with similar but not same stops: returns a list of routes with the same stations as, but not the same stops as, another route
    4. Find stations visited by all trains: does what it says

5. Find trains that do not visits a station: lists trains that do not stop at the given station
6. Find routes that stop at greater than X% of stations: lists routes that stop at more than the given percentage of stations. Percentage 0-100
7. Display schedule of route: does what it says
8. Find available seats on route at day and time: takes origin time and day and a route number and returns the number of available seats. If origin time is not known, the next submenu item (3.9) will show, or (3.7)
9. Find origin time from station and departure time: gives time at first station of a route given a station on the route and the time at that station, and the day.

4. Database operations:
    1. Import data to db: imports data from file to database, assuming file is a csv (can be not commas). Will ask for columns present in data to import along with their order.
    2. Export data from db: exports csv with or without header to an absolute path. Does not allow to choose which columns are exported, exports all.
    3. Delete all data: deletes all data in table, does not delete data itself
    4. Delete all functions: deletes all functions from db
    5. Reset tables: rebuilds tables from expressRailway.sql
    6. Update/reset queries: rebuilds functions from expressRailwayQueries.sql

       Note: both sql files are included in the jar at compile time, so updates after compilation to the files outside the jar will not be read.
5. Exit: returns to the login screen, username exit to end program