

2022

Costa**EXPRESS**



DATABASE MANAGEMENT SYSTEMS PROJECT

SAMANTHA LAVRINC, EMMA(QINNUO) LI, ANASTASIA MOKHON

DR COSTA – SPRING 2022

Contents

Background:	2
Getting Started:	2
Launch Screen:.....	5
Login Screen:	5
Agent Access:	6
Add Customer:	7
Edit/View Customer:	8
Search Options:	9
Single Route Search:.....	10
Combination Route Search:.....	11
Advanced Search (A):	12
Advanced Search (B):	13
Advanced Search (C):.....	13
Advanced Search (D):.....	14
Advanced Search (E):	14
Advanced Search (F):	15
Advanced Search (G):.....	15
Advanced Search (H):.....	16
Advanced Search (I):	16
Reservation Entry:	17
Obtain Trip Price:	17
Add Reservation:	17
Edit Reservation:	18
Admin Screen:	19
Import:.....	19
Export:.....	19
Delete Database:	20
Final Thoughts:	20

Background:

Project Goal: The goal of the semester group project is to allow groups of three students to become familiar with the development cycle of a database application through conceptual design, logical design, schema evolution, physical design, and implementation of an application.

In this project, groups were given an analysis of the requirements of the supposed travel agency Costa Express, which specializes in organizing train journeys. Groups were asked to conceptually model the described mini-world(project.pdf) and then implement it using the relational database PostgreSQL. Implementation of the application was completed in Java using JDBC.

Getting Started:










To begin the database setup, run the following sql files in datagrip:

- createDB.sql
- getpricefunction.sql
- updatePhase3_v2.sql
- updatePhase3_v3.sql
- updatePhase3_v4.sql

Ensure that all views have been created within the functions of the sql files.

In order to run properly, the program must have the following file structure:

Demo			
Name	Date modified	Type	Size
dbpack	4/20/2022 11:41 PM	File folder	
advancedSearches.sql	4/20/2022 10:20 PM	SQL File	12 KB
AllData.txt	4/15/2022 3:23 AM	Text Document	136 KB
costa-01-01-01.png	4/15/2022 3:23 AM	PNG File	11 KB
createDB.sql	4/15/2022 3:23 AM	SQL File	6 KB
getpricefunction.sql	4/15/2022 3:23 AM	SQL File	2 KB
Makefile	4/20/2022 11:42 PM	File	1 KB
postgresql-42.2.18.jar	4/15/2022 3:23 AM	Executable Jar File	982 KB
README.md	4/19/2022 10:41 AM	MD File	6 KB
result.txt	4/15/2022 3:23 AM	Text Document	854 KB
updatePhase3_v1.sql	4/20/2022 9:50 PM	SQL File	8 KB
updatePhase3_v2.sql	4/16/2022 1:03 AM	SQL File	10 KB
updatePhase3_v3.sql	4/20/2022 10:20 PM	SQL File	7 KB
updatePhase3_v4.sql	4/20/2022 11:34 PM	SQL File	3 KB
user-01-01.png	4/15/2022 3:23 AM	PNG File	3 KB

Demo > dbpack				
Name	Date modified	Type	Size	
 AdminScreen.java	4/20/2022 11:24 PM	JAVA File	12 KB	
 AdvE.java	4/20/2022 10:20 PM	JAVA File	8 KB	
 AdvF.java	4/20/2022 10:20 PM	JAVA File	9 KB	
 SearchDatabase.java	4/20/2022 10:20 PM	JAVA File	9 KB	
 ImportAllText.java	4/20/2022 8:34 PM	JAVA File	18 KB	
 CostaExpress.java	4/20/2022 1:08 AM	JAVA File	4 KB	
 ComboRoute.java	4/19/2022 3:02 PM	JAVA File	20 KB	
 SingleRoute.java	4/19/2022 3:02 PM	JAVA File	16 KB	
 AllData.txt	4/19/2022 10:12 AM	JAVA File	8 KB	

Note that the dbpack file is a package. If running the file returns an error stating it cannot find these classes, that means that your local system is not recognizing dbpack as a package file.

Launch the java file from the root (Demo) directory using the command:

With Windows using chocolatey:

Make

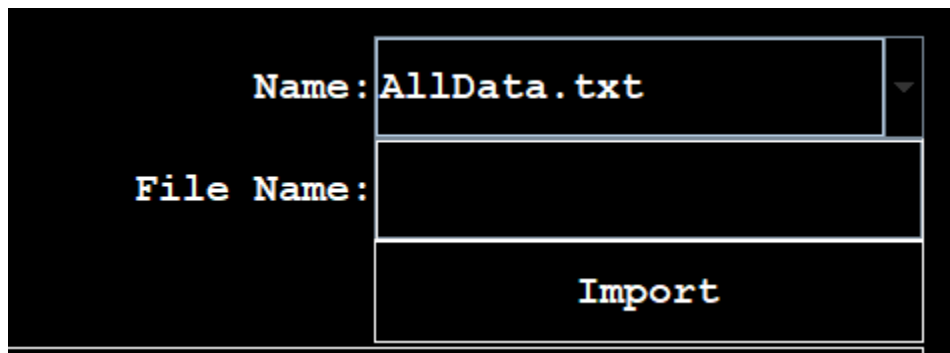
```
java -cp "postgresql-42.2.18.jar;." dbpack/CostaExpress
```

or with Mac/Linux:

Make

```
java -cp postgresql-42.2.18.jar:. dbpack/CostaExpress
```

Select Admin > Login with password > AllData.txt, with no file name, and select 'Import'



The screenshot shows a terminal window with a dark background. On the left, the text 'Name:' is followed by a text box containing 'AllData.txt'. Below this, the text 'File Name:' is followed by an empty text box. At the bottom right, there is a button labeled 'Import'.

The import statement should have generated the demonstration data provided by the project outline which can be verified using select statements in datagrip.

(continued...)

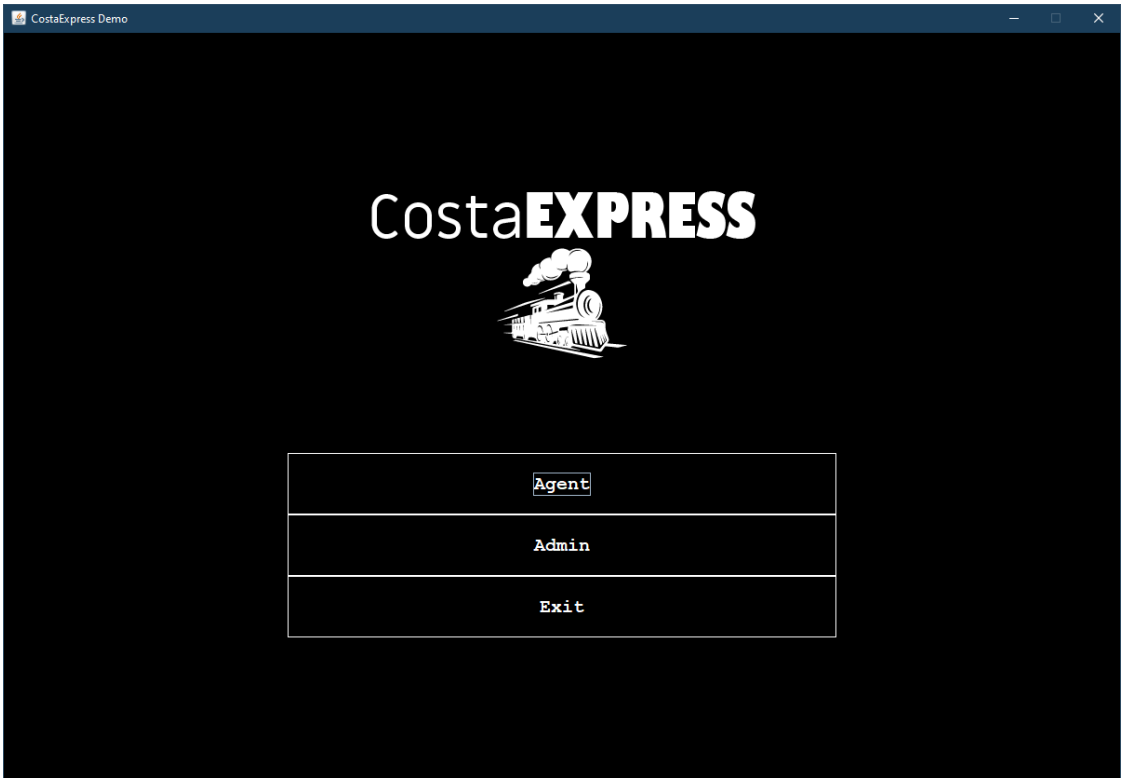
Then, run the following sql file in datagrip:

- updatePhase3_v1.sql

Verify that the most recent sql statement has updates the stop_seatcount table with the command
SELECT * from stop_seatcount;

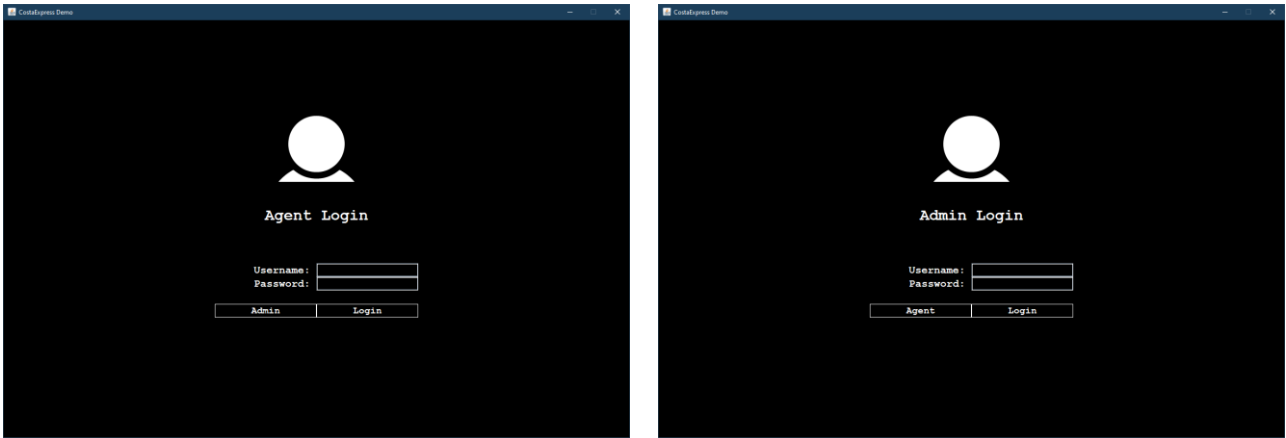
The demo program should now be fully functional, no need for a restart.

Launch Screen:



By clicking 'Agent,' the program is directed to the Agent Login Screen.
By clicking 'Admin,' the program is directed to the Admin Login Screen.

Login Screen:



Note: Username postgres is default and does not need to be entered to obtain access, however a user password must still be added.

Agent Access:



Select one of the menu options.

- Add Customer gives the option to add a Customer to the database.
- Edit/View Customer gives the option to edit and view database Customers.
- Search Database provides a menu of search options including:
 - o Search by Single Route
 - o Search by Combination Route
 - o Advanced Search Options A – I
- Add Reservation gives the option to add a new reservation.
- Update Reservation gives the option to update an existing reservation.
- Logout returns the user to the login menu.

Add Customer:

CostaExpress Demo

Customer Entry

Please enter the following...

First Name:

Last Name:

Email:

Phone Number:

Street:

Town:

Postal Code:

Customer ID: xx

Back

Add

Enter the requested information and select the 'Add' button.
On successful creation, a customer ID is generated and returned into the Customer ID field.

Customer ID:

4

Edit/View Customer:

CostaExpress Demo

Edit Customer

Please enter either (Customer ID)
- OR - (First Name, Last Name and Email)

Customer ID:

-- OR --

First Name:

Last Name:

Email:

Get Customer

Phone Number:

xx

Street:

xx

Town:

xx

Postal Code:

xx

Back

Save

Enter either the (Customer ID) or (First Name, Last Name and Email Address).
Then select 'Get Customer.'
The Customer's information is displayed in the relative fields and can now be updated.
To update, select the 'Save' button.

Customer ID:

100706

-- OR --

First Name:

Last Name:

Email:

Get Customer

Phone Number:

xx

Street:

xx

Town:

xx

Postal Code:

xx

Back

Save

Customer ID:

100706

-- OR --

First Name:

Janise

Last Name:

Heard

Email:

Get Customer

Phone Number:

Street:

1734 la Mancía

Town:

San Antonio

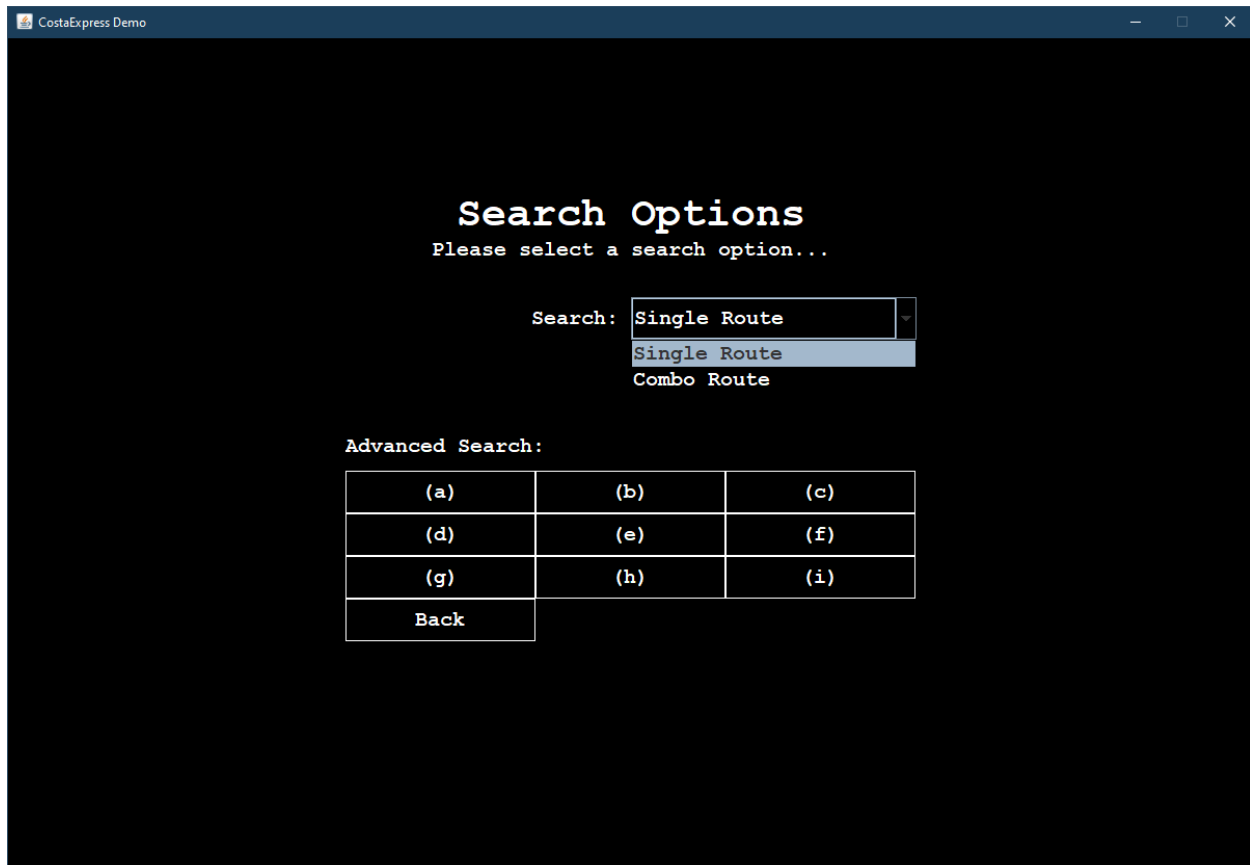
Postal Code:

TX 78258-4547

Back

Save

Search Options:



The Search Option Menu corresponds to the original assignment pdf for search parameters. Additionally, we added a Search Customer Reservation feature.

The Search Selections are as follows:

- Single Route -
- Combo Route -
- Advanced Searches
 - o A – Find all trains that pass through a specific station at a specific day/time combination
 - o B – Find the routes that travel more than one rail line
 - o C – Rank the trains that are scheduled for more than one route
 - o D – Find routes that pass through the same stations but don't have the same stops
 - o E – Find any stations through which all trains pass
 - o F – Find all the trains that do not stop at a specific station
 - o G – Find routes that stop at least at XX% of the stations they visit
 - o H – Display the schedule of a route
 - o I – Find the availability of a route at every stop on a specific day and time
- Back Button – Select to return to the previous screen.

Single Route Search:

CostaExpress Demo

Single Route Search

Route:	Time:	Day:	TrainID:	Seats:	Start:	End:
--------	-------	------	----------	--------	--------	------

Start Station(as ID):

End Station(as ID):

Day:

Order By: Lowest Price

Next >

Back

Search

Enter the ID number for the Start and End station, as well as the day and a sorting option. Clicking the ‘Search’ button populates the result table.

Route:	Time:	Day:	TrainID:	Seats:	Start:	End:
85	13:41:00	Tuesday	244	175	1	2
4772	11:08:00	Tuesday	149	200	1	2
4775	08:34:00	Tuesday	315	150	1	2
114	08:09:00	Tuesday	73	150	1	2
4768	13:46:00	Tuesday	24	150	1	2
2346	03:45:00	Tuesday	145	100	1	2
22	07:29:00	Tuesday	249	75	1	2
22	07:29:00	Tuesday	249	75	1	2
22	07:29:00	Tuesday	249	75	1	2
22	07:29:00	Tuesday	249	75	1	2

Start Station(as ID):

End Station(as ID):

Day:

Order By: Lowest Price

Next >

1

2

Tuesday

Combination Route Search:

CostaExpress Demo

Combination Route Search

Day:	Time:	Rt:	Train:	Start Station:	End Station:	Avail. Seats:

Next >	
Start Station(as ID):	15
End Station(as ID):	20
Day:	Monday
Order By:	Lowest Price
Back	Search

Enter the ID number for the Start and End station, as well as the day and a sorting option. Clicking the 'Search' button populates the result table.

	Day:	Time:	Rt:	Train:	Start Station:	End Station:	Avail. Seats:
1st :	Monday	03:37:00	1029	227	15	40	75
2nd :	Monday	12:21:00	4025	51	40	20	150
1st :	Monday	03:37:00	1029	227	15	12	75
2nd :	Monday	15:55:00	1758	51	12	20	150
1st :	Monday	03:37:00	1029	227	15	8	75
2nd :	Monday	15:55:00	1758	51	8	20	150
1st :	Monday	03:37:00	1029	227	15	28	75
2nd :	Monday	15:55:00	1758	51	28	20	150
1st :	Monday	03:37:00	1029	227	15	8	75
2nd :	Monday	14:55:00	3957	266	8	20	200
1st :	Monday	03:37:00	1029	227	15	8	75
2nd :	Monday	12:21:00	4025	51	8	20	150

The First route is the first 'leg' of the journey, with the end station that functions as a layover station. The Second route is the second 'leg' of the journey, with the start station that functions as a layover station.

Advanced Search (A) :
Finds all trains that pass through a specific station during a specific day/time combination.

CostaExpress Demo

Advanced Search (A)

Find all trains that pass through a specific station at a specific day/time combination.

Train_ID: | Route_ID:

Station(as ID):15

Day:Monday

Time (as 00:00:00):03:37:00

Next >

Back

Search

Enter the Station ID, day and time.

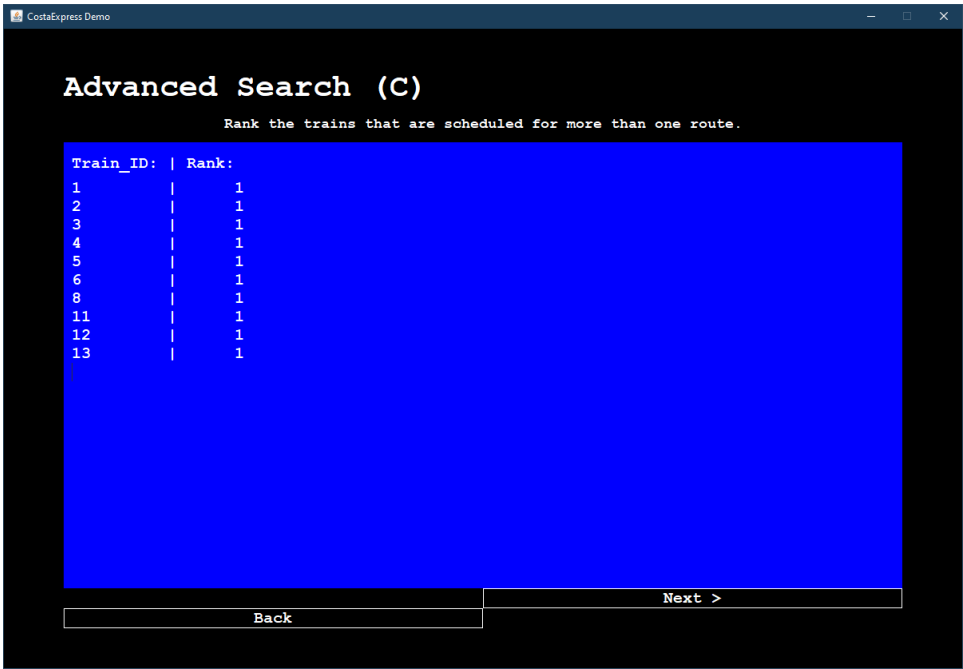
When the 'Search' button is selected, the screen will populate with trains that pass through a specific station during the specified day/time combination.

Train_ID: | Route_ID:
1029 | 227

Advanced Search (B) :
Finds the Routes that travel more than one rail line.

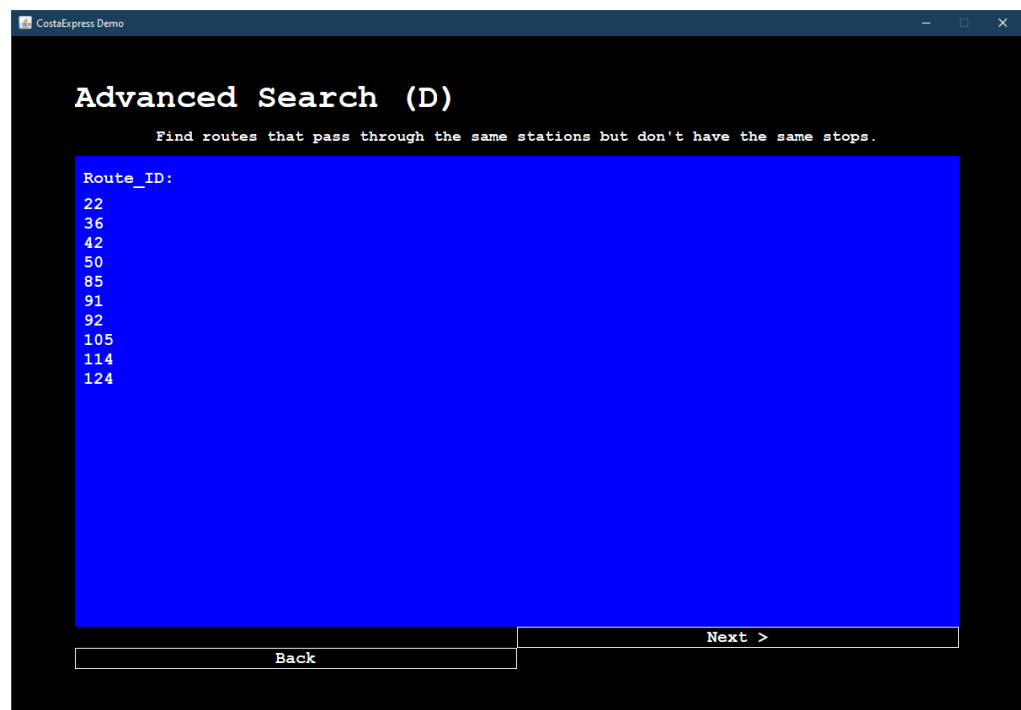


Advanced Search (C) :
Ranks the trains that are scheduled for more than one route.



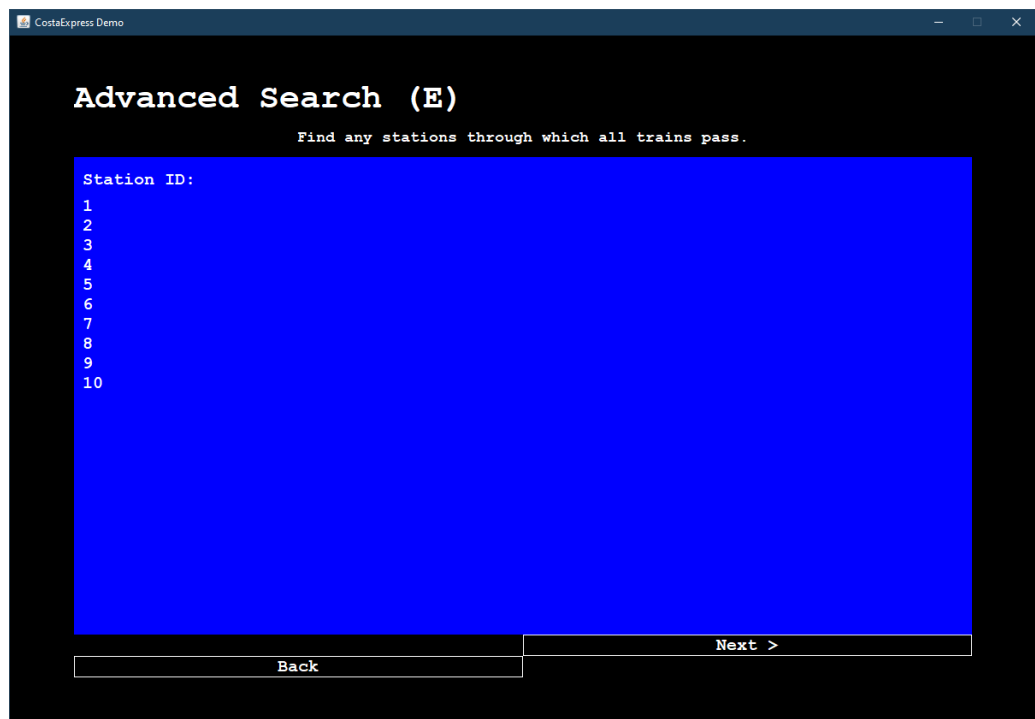
Advanced Search (D) :

Finds the routes that pass through the same stations but don't share the same stops.



Advanced Search (E) :

Finds any stations through which all trains pass.



Advanced Search (F) :

Finds all trains that do not stop at a specific station.

CostaExpress Demo

Advanced Search (F)

Find all trains that do not stop at a specific station.

Train_ID:

Search returns empty results: All trains pass through this station.

Station(as ID):

5

Next >

Back

Search

Enter the Station ID and select 'Search.'

Advanced Search (G) :

Finds routes that stop at least at XX% of the stations they visit, where XX is a number from 10 to 90.

CostaExpress Demo

Advanced Search (G)

Find routes that stop at least at XX% of the stations they visit, where XX number from 10 to 90

Route_ID: | Stops Percentage:

4192 | 0.54

1158 | 0.63

2429 | 0.80

753 | 0.57

477 | 0.73

3925 | 0.72

4674 | 0.53

1091 | 0.54

3726 | 0.63

2080 | 0.71

Percentage (XX):

50

Next >

Back

Search

Enter the Percentage as (XX) or 50 for 50% and select 'Search.'

Advanced Search (H) :

For a specified route, list the days of departure, departure hours and trains that run it.

CostaExpress Demo

Advanced Search (H)

For a specified route, list the days of departure, departure hours and trains that run it.

Route_ID:	Departure_Date:	Departure_Hours:	Trains:
22	Saturday	02:28:00	328
22	Sunday	04:46:00	128
22	Thursday	03:02:00	344
22	Tuesday	07:29:00	249
22	Wednesday	04:05:00	53

Route(as ID):22

Next >

Back

Search

Enter the Route ID and select ‘Search.’

Advanced Search (I) :

Finds the availability of a route at every stop on a specific day and time.

CostaExpress Demo

Advanced Search (I)

Find the availability of a route at every stop on a specific day and time.

Route_ID:	Station_ID:	Day:	Time:
22	1	Saturday	02:28:00
22	2	Saturday	02:28:00
22	6	Saturday	02:28:00
22	7	Saturday	02:28:00
22	9	Saturday	02:28:00
22	11	Saturday	02:28:00
22	37	Saturday	02:28:00
22	40	Saturday	02:28:00
22	20	Saturday	02:28:00

Route(as ID):22

Day:Saturday

Time (as 00:00:00):02:28:00

Next >

Back

Search

Enter the desired Route ID, Day and Time as 00:00:00 and select ‘Search.’

Reservation Entry:

CostaExpress Demo

Reservation Entry

Please enter the following...

Start Station(as ID):

End Station(as ID):

Day:

Time(as 00:00):

Route:

Train:xx

Price:xx

Get Price

Adjustments:Yes

Customer ID:

Amount Paid:

Remaining Balance:xx

Reservation Number:xx

Ticket Number:xx

Back

Add Reservation

Obtain Trip Price:

Begin by entering the Start Station ID, End Station ID, Time, Day and Route which can be obtained from one of the route search options. Select the 'Get Price' button to obtain the price of the trip.

Start Station(as ID):

End Station(as ID):

Day:

Time(as 00:00):

Route:

Train:xx

Price:xx

Get Price

2

4

Saturday

02:28

22

xx

xx

Train:328

Price:800.4

Get Price

Add Reservation:

After obtaining the trip price, select whether the Customer would like to utilize a service to adjust their reservation based on a downed line. Then, enter the Customer ID and Amount Paid. By clicking 'Add Reservation,' the Agent is saving the customer's current balance in the system and if the balance is \$0, will generate a ticket number.

Edit Reservation:

CostaExpress Demo

Edit Reservation

Please enter the Reservation ID number

Reservation Number:

Payment:

Delete

Get Reservation

First Name:

xx

Last Name:

xx

Day:

xx

Time:

xx

Price:

xx

Balance:

xx

Ticket Number:

xx

Back

Save

To update a reservation, it is assumed that the only thing that would need edited about a reservation is the balance field. If a customer wishes to book a reservation with different stops, at a different time or on a different day, an agent may then instead delete the current reservation and create a new reservation with the updates.

Begin by entering the reservation number and selecting 'Get Reservation' to generate the reservation's corresponding data.

- To update the Customer's reservation balance, enter the current payment amount and select the 'Save' option. If the reservation balance is now \$0.00, a new ticket is generated.
- To delete the Customer's reservation, instead select the 'delete' option.

Admin Screen:



Import:

AllData.txt – No file name should be entered. This option uses the AllData.txt file in the root directory to generate demo data.

All other options – These options can import data that was generated *using the program's export option*. Thus, this option should not be used unless Export has generated export .txt files from the database.

If data has been exported using the Export button, select the desired table data to import and add a direct file name to the location of the export data, be sure to fill in the entire path.

EX: C:\... all this stuff too...\CS1555_CostaExpress\export\tickets.txt

Export:

Before using the export function:

Open updatePhase3_v4.sql and update the export file location to your system specific export file location. Run updatePhase3_v4.sql to save these updates.

You may now safely export the database data to the chosen export file. This can be verified with new .txt documents reflecting the name of the table.

Delete Database:

This brings up an option to either confirm the deletion or cancel. Select 'Yes' to delete the data within the database.

Final Thoughts:

Throughout this project, our group was gradually exposed to additional functionality provided by PostgreSQL and JDBC. For most of us, it was our first exposure to SQL and for all of us, it was our first experience writing a fully functioning interactive program from scratch. Approaching this opportunity as a learning experience, many of our early design choices were made with minimal awareness of features like triggers, procedures, views and functions. For example, looking back with the knowledge we obtained from class, it would have been beneficial to create smaller functions to traverse lines and obtain the distance or station numbers from a start station to an end station much earlier in development. It would have also been interesting to add additional functionality and features to the resulting program, perhaps creating a more cohesive display that combined batch results of multiple input parameters. As it currently functions, we have multiple classes for each one of our displays that serve to demonstrate the project requirements, but many of these could have easily been combined, like Add and Edit Customer or Add and Edit Reservation. Or with consideration to use, a program that displays each entity as a searchable/editable object, where each attribute that represents a relationship also functions as a button that reveals a display of the corresponding attributes for the entity.

The next major steps in development would be to implement additional security to the password feature, fully roll out line_disruption trigger implementation, and work on program access to demonstrate the functionality of the reservation_cancel trigger. Additionally, while the basic logic of the sorting functionality within single route search produces results that are sorted based on price, time, stations and stops, because we didn't implement the smaller functions to obtain the distance, or station numbers based on an entry and exit station, neither the single or combination search options display the resulting price, time, stops or stations. The pricing functionality based on distance is implemented on the add reservation screen, so there is a way to obtain the information, but it is not conveniently displayed in the search screen.