Java Assignment - Data Explorer

Completed by Eoghan Byrne (C17315336)
12th April 2019
eoghan.byrne4@mydit.ie (mailto:eoghan.byrne4@mydit.ie)

Index

Overview

The program is designed to allow a user to launch the application, import their cleaned data of the TFI Bus Stops (https://data.gov.ie/dataset/b61d8abf-efd1-4476-a29b-afc8c2edd6ba/resource/6db74b2d-c7d3-4faf-a922-851c042715ba) and see filtered views with row counts on each query
The user can also configure their required settings to connect to their own database. (***Note there is no remote database connection available for demo***)

Link to Video Explanation (https://drive.google.com/file/d/1ELxCeChoeXsOi_z0bUUY_ENndT6KawCp/view?usp=sharing)

Link to Github (https://github.com/C17315336/Java-Assignment/)

Classes

Control

This class was used to initiate the program by creating a new GUI

```
new GUI();
```

GUI

This class is where the user interacts with the program
It primarily operates off of;

- 11 Buttons for various actions
- 1 Table to display the users content and queries
- 2 Text fields/areas for user input and system output
- 1 Checkbox for additional usability

Each attribute is declared and all names are easily understandable

```
private JLabel lbTitle;
private JTable tbTable;
private JTextField tfFilter;
private JButton btFilter;
private JButton btClearDB;
private JButton btPullDB;
private JButton btSelectFile;
private JButton btImportFile;
private JCheckBox cbHeader;
private JButton btQuit;
private JButton btAdmin;
private JButton btStop;
private JButton btNameLocal;
private JButton btLocality;
private JButton btName;
private JTextArea taOutput;
```

Each attribute is placed in the GUI through the use of the `GridBagLayout`
Then each button has an `ActionListener` that will call on other classes to trigger a function

Admin

This class is where the user interacts with the database configurations of the program The user simply has access to configure;

- Domain
- Port
- Username
- Password
- Database (*Name*)
- Table (*Name*)

Once the user is happy they can save the settings and close the window
The program will then generate the correct connection details and will dynamically alter an *SQL* queries to match the settings eg

```
connect = DriverManager.getConnection(jdbc:mysql://localhost:8889/java, Eoghan, 1
etmein);
```

ConnectionInfo

This class is primarily a holding point for all the *"global variables"* if you will
This is just where the individual variables of admin settings are held

EraseData

This class is used to erase the contents of the database entirely
It does this by connecting to the database and executing the line of *SQL*

```
DELETE FROM `assignment` WHERE `StopNumber` > 0
```

This removes everything in the database as there are no stop numbers 0

Filter

This class takes the users input from the text field and applies it to the `setRowFilter` which will then update the table's content

LoadData

This class will take the values loaded from the CSV file and insert them into the table view

PullData

This class will pull the values from the database and populate them into the table view

SaveData

This class will take the contents of the CSV file and insert them into the database the application is connected to

More specifically the *SaveData* class is only called by *btImportFile* which can only be enabled once a CSV file has been selected and loaded onto the table view, the class can then be accessed where it converts the contents of the table into SQL *INSERT* statements through a while loop

```
String sql = "INSERT INTO " + ConnectionInfo.getDbtable()
                            + " (StopNumber,NamewithoutLocality,Locality,Name,Easting,N
orthing) " +
                            "VALUES ('" + StopNumber + "','" + NamewithoutLocality + "
','" +
                            Locality + "'" + ",'" + Name + "','" + Easting + "','" + N
orthing + "') ";
```

SortDataAs & SortDataDs

These classes simply sort the table view by using a `SortKey` and the `TableRowSorter` along with a column *ID* specified by the associated button action

There are separate classes to allow the user to **RIGHT Click** for Ascending and **LEFT Click** for Descending

Additional Information

Database Configuration

There is a `.sql` file which will build the required table of:

```
+------------+---------------------+----------+------+---------+----------
+
| StopNumber | NamewithoutLocality | Locality | Name | Easting | Northing |
+------------+---------------------+----------+------+---------+----------
+
```

External JARs

There are 2 associated JARs which must be configured in the Build Path

CSV Data Cleaning

As there is plain English which has been converted to a `.csv` format, it has been required to change the ASCII character from **'** to **^** and additionally goes for the ASCII character of **,** which is now **..**

*Note - The general downloadable CSV of TFI Bus Stops has a row count of over 17,000, my demo CSV file will import 3,500 rows

Brief

2. Data Explorer
Note: I will cover SQL database connection (briefly) in class.

Data analysis is a major area within Computer Science. Apart from the Big Data generated by social media and internet generally – there is an insatiable desire on behalf of companies to analyse their data in order to reveal "knowledge" hidden in the data. For example, an Optical chain analysed their sales data to determine that sales of high-end high-profit glasses peaked on Friday afternoons – so they made sure that they had enough staff to service this demand.

The Irish Government have put 1000s of datasets into public use at a portal site: https://data.gov.ie/data (https://data.gov.ie/data)

This has data about a whole plethora of public interests and government control information e.g. about crime rates, hospitals, schools, transport, environment, energy use and so on.

The purpose of this project is to take **ONE** of these datasets – and build a tool that shows interesting facts from the dataset. The dataset formats include Comma Separated (CSV) which is probably the easier to work at – so these datasets are at: https://data.gov.ie/data/search?res_format=CSV (https://data.gov.ie/data/search?res_format=CSV)

You don't have to use the full dataset if it is too big. But to query it, you will need to LOAD the dataset, or a subset of it, into a relational database yourself – and get a connection working between your java code and the database (using JDBC).

You could also just read in the file and do simple operations on it (e.g. how many of ??).. but searching is very limited if you stick just to file format without a database.

Your project will need to have a GUI that allows query parameters to be put in.

Extras

- Ability to see the results through the GUI too.
- Flexible queries – not just one or two hardcoded
- Whatever else you decide might enhance the application