# Assignment 1

## Overview

The goal of assignment 1 is to create a graphical user interface that will allow users to read information from a MySQL database and display it as chart data. The information should be ***anything you are interested in***. For example, it could be comparing aspects of video games, weather data, processor capabilities, etc… Each student will need to register a unique data set prior to building their program.

The MySQL database should be remotely accessible on your AWS platform. Your program must be built using Intellij and stored in a PRIVATE GitHub repository.

When the application is launched, it should show a graph of information on a styled JavaFX application.



Figure - Initial launch of project shows a graph

The application must support at least 2 different graphs and/or change to a scene with a TableView object that displays all the data from the database.

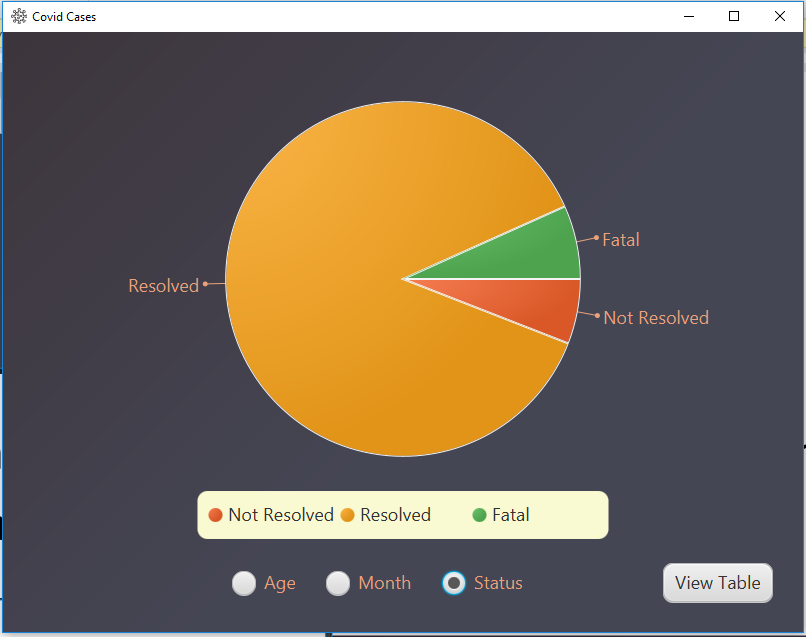


Figure -Project showing 2 different graphs

OR

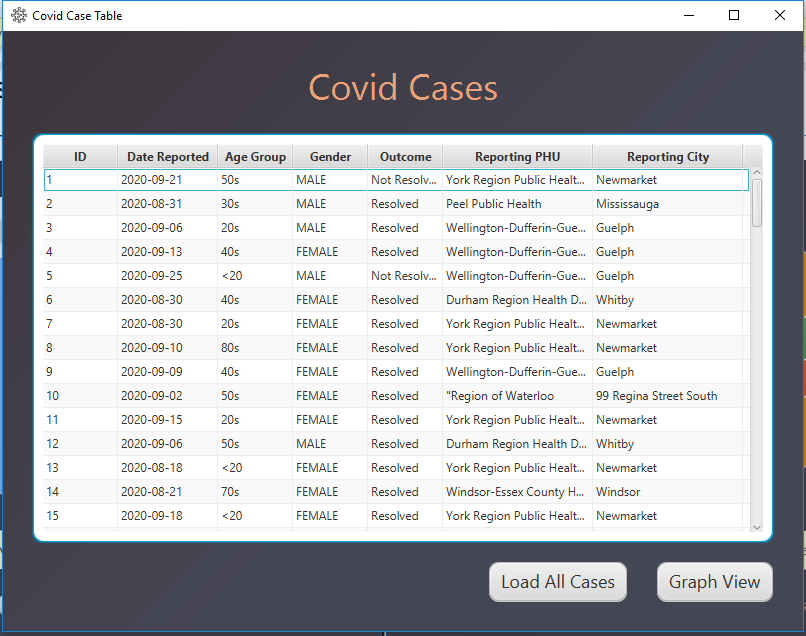


Figure - A graph and a TableView Scene

## What to Do:

1. Register your unique project idea by going to <https://fall20projects.azurewebsites.net/>, find the appropriate section of COMP1011 and select “Submit Idea”. You can see what other students are planning to build. No duplicate projects will be allowed, so do not invest significant energy in your project until you receive approval from me.
2. Carefully review, sign and send me the integrity pledge (included in assignment on line).
3. When the application launches, it should display a scene with a Chart object populated from information in a MySQL database.
4. There should be a utility to either change scenes (such as a button) or to select a different graph (I used RadioButton’s for this in my example below).
5. All scenes should be styled using CSS. Do not use the default grey background for everything. Add some colour, round some corners, add images to buttons, change some fonts… have fun with it!! Check out <http://www.jfoenix.com/> for some material design tools if you want to push some limits.
6. The icon on the stage should be something unique.

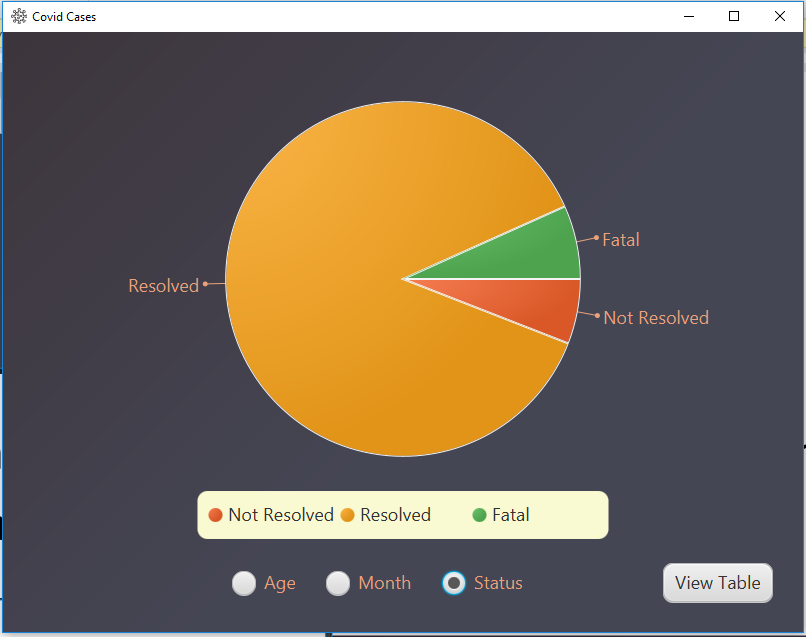


Figure - Image showing changed icon

## Grading

All of your marks will be based on the rubric defined below (and visible in Blackboard).

| Criteria | Level 0 | Level 1 | Level 2 | Level 3 |
| --- | --- | --- | --- | --- |
| Code Style | The code does not follow typical Java programming style. I.e. a capital letter to start all class names, lower case letters start variable and method names | The code is indented and has proper upper case/lower case conventions | All of level 1, plus each method has a Javadoc style of comment prior to the method describing what it does | Level 2 plus different directories/packages are used for the model, view and controller to keep everything organized |
| User Experience | When the program launches, an exception is thrown or the Chart is not populated. | The program launches and the chart launches with some data, but the overall look and feel does not have a professional look to it. i.e. objects are not aligned or extreme colour choices are made. | Level 1 plus a professional look and feel is apparent. |  |
| When launched, a Chart object is populated with data | The first scene does not contain a graph. | The first scene has a Chart object, but it is not populated from a MySQL DB. | Level 1 plus it is populated based off a DB query. | Level 2 plus the labels and legend are easy to read/visible. If labels/legend are visible, but a DB query is not used, it is level 2. |
| Change to TableView or a different Chart scene | There is no utility to change from the first graph. | There is a functional object (i.e. button or radio button) that will change the scene to show a new graph or TableView object |  |  |
| CSS styling | There is no CSS stylesheet and/or it is not connected to the view object. | The CSS is connected and styles up to 3 elements. | Level 1, plus it styles up to 6 elements. | Level 2, plus it styles more than 6 elements. |
| Change the icon | The default icon is showing | There is a new icon. | The new icon is somewhat related to the graph content. |  |
| The Model class(es) is well structured | The model class(es) is either not present or is not used. | The model class has poor naming conventions and/or poor instance variable data types. All instance variables must be private. | Level 1 plus all data types are logical. All method and variable names are logical and follow camelcase style.  The constructor uses the set methods | Level 2 plus all “set” methods contain ***useful*** validation. Do not use isBlank() (or equivalent) for validation in all set methods. Validate against known lists, number ranges, etc… |
| Database | There is no remote database connectivity in the project and/or the necessary SQL was not provided for a local DB | There is database access, but it is local and the necessary SQL statements are missing | There is local DB access and the SQL statements are provided OR a remote DB with data is used and all necessary info is provided (i.e. URL, user name and password work) |  |
| Database Query | A DB query does not exist or creates an error | The DB query returns a valid ResultSet, but it is not used to populate a Chart or TableView | Level 1 plus it is used to populate a Chart or TableView object | Level 2 plus queries are used to populate either 2 charts or a chart and a TableView object. The Connection, Statement and ResultSet are closed at the completion of the method. |
| Submission | A link to your private GitHub account was not submitted on Blackboard. | A private Github link was sent. JaretWright is listed as a collaborator AND all project files including the build info are present for Intellij. | Level 1 plus there are at least 2 commits per week and the source URL for the data series is included. | Level 2 plus there are a total of over 6 commits with meaningful changes. In other words, do not submit a series of commits at the last minute with updates to comments. I want to see you working on your project over time. |

## Bonus-2 marks

Create a utility that can convert a text based csv file into valid sql to insert records in your table. You must supply the URL you down loaded the csv file from. Alternatively, use a JSON feed to retrieve records and upload them to your MySQL DB.

## Reverse Rubric

In the event that the project submitted does not follow Java programming best practices, 1 mark will be deducted for EACH infraction noted below:

* -1 mark for any class name that does not start with a capital letter and follow camel case practices.
* -1 mark for any variable or method name that starts with a capital letter.
* -1 mark for each line of code that must be change to make the project compile. A maximum of 10 lines will be changed at which point assessment of the project will stop
* There is a 20% penalty per calendar day for late submissions

## What to submit and when

|  |  |
| --- | --- |
| Lab week 4 | Integrity pledge, register your idea on <https://fall20projects.azurewebsites.net/> and submit your private GitHub URL for assignment 1. |
| Lab week 5 | DBUtility class with methods that can return information that can be used in your chart (displayed in the console) |
| Lab week 6 | Code should be able to launch a JavaFX scene with a working Chart object |
| Full Submission (week 7) | All requirements as defined by this document |