Collaboration – not permitted

For this test, you are NOT permitted to perform work with or communicate with others in any form. This includes sharing your code or helping classmate's trouble shoot code. Violation of this will result in an automatic grade of 0 and an academic misconduct being filed.

GitHub/GitHub Classroom

1. At the start of the test, you must create a private GitHub repository and make PriyanshT a collaborator.
   OR
2. Accept the Test on GitHub Classroom and start working on the repository created for you. (You do NOT need to add me as a collaborator in this case)

You need to submit your code at the end of every challenge until you submit your final work. Once you have submitted the GitHub link in Blackboard you cannot make any changes to your GitHub repo. The test is considered submitted at that time of submission to Blackboard. If the timestamp on GitHub is later than the timestamp in Blackboard, your paper will not be marked. Tests with a GitHub timestamp after 1:00 pm will not be accepted.

Database commands – Java MidTerm.sql

You are provided a file called Java MidTerm.sql. This should be run on your AWS MySQL server (same as assignment 1).

If the connection to AWS is not working, you can use a local DB connection. (The connection string in your DBUtility will be different if in case of localhost)

The .sql file will create a database and a table of employees. The contents of that table will be used to populate your GUI.

GUI behavior

Check the Screenshots at the end to view the different phases of the GUI application after all steps are completed.

Files for the test

You can get started and access the files using **ANY ONE** of the following methods.

GitHub Classroom (recommended and easiest)	Click the classroom link and accept the assignment. Your repository will be generated. Open using IntelliJ or command prompt and start working.
GitHub repo - Duplicating/Downloading my repo	Download the files from the GitHub link. Create your own repository on GitHub. Add PriyanshT as collaborator.
Zip file - Downloading zip file from the Blackboard	You are given a .zip file for the test. Extract this to a place that is easy to find on your computer. You can open it using IntelliJ by going to file -> open and navigate to the folder you stored it in. Create and upload to your own repository on GitHub. Add PriyanshT as collaborator.
Create your own project and copying the content.	If none of the above work, create a new JavaFX project as we have done in the classroom. Create MVC files and copy paste the contents of view and controller. Create and upload to your own repository on GitHub. Add PriyanshT as collaborator.

This is a project that has a Main method to run the program, a view and a controller. The view and associated controller already have fxid's and methods defined, so you do not need to add those.

Details of each class/methods are defined below. You are welcome to have additional classes and methods, but these ones are mandatory.

ComboBox

The ComboBox used in the test is the same as most other JavaFX objects. It contains an ObservableList which is configured to hold String objects. Here some useful snippets to save you time looking them up.

- ***Add a single String to the ComboBox***, areaCodeComboBox.getItems().add("what ever");
- ***Add a collection to the ComboBox***, areaCodeComboBox.getItems().addAll(collectionOfStrings);
- ***Get the String selected*** from the ComboBox, areaCodeComeboBox.getSelectionModel().getSelectedItem();

1. The Employee (model) class

The Employee class has the following attributes (minimum): employee id, first name, last name, address, city, province, and phone no. The class should have a constructor that validates all the arguments, get, and set methods.

Validation rules

a. The employee id should be greater than 200465000.
b. First and last names must be more than 1 character.
c. Phone should match the North American dialing plan.
d. Address must be more than 5 characters.
e. Province should be in the list of
"AB","BC","MB","NB","NL","NS","NT","NU","ON",”PE","QC","SK”, “YT"
f. City must be more than 3 characters.

Commit all changed/new files to GitHub with the commit message “Question 1 complete”

Marks: 5 (see rubric for breakdown)

2. Controller Class and fxml file

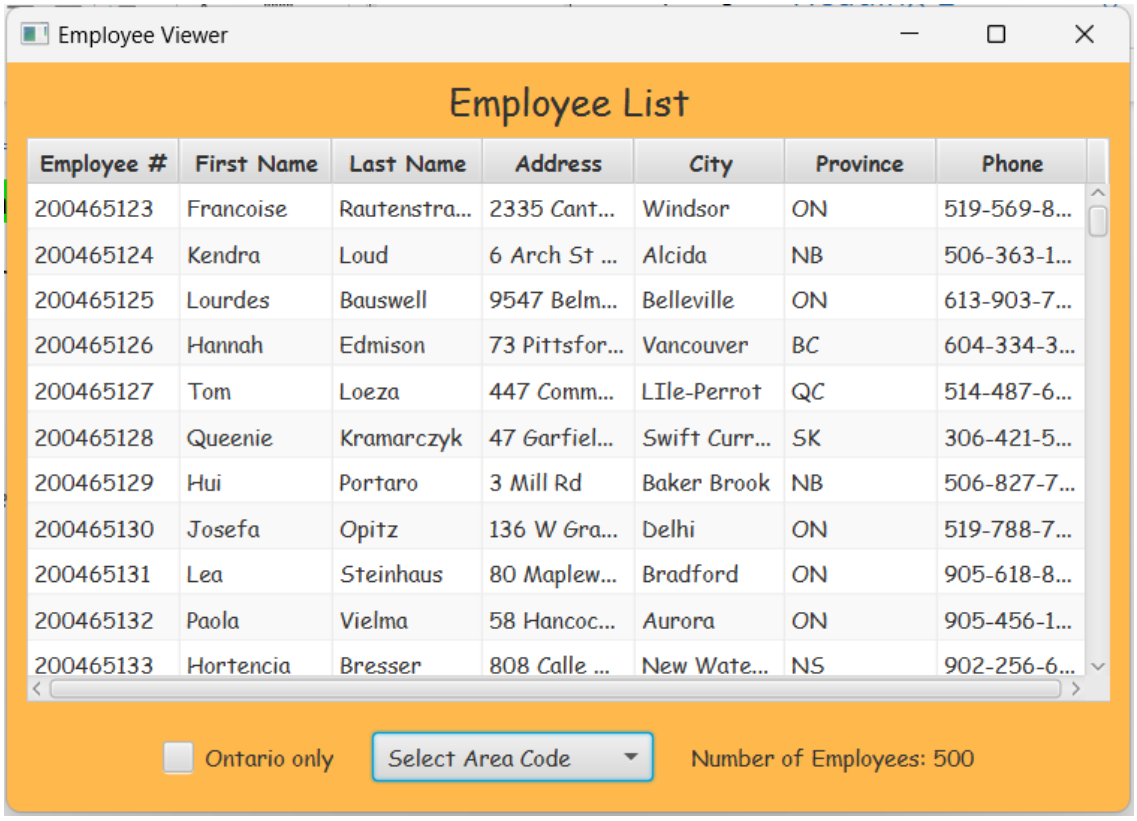


Figure 1-Initial View when application launches

Requirements – Initial Load

a. When the application launches, it should look visually the same as Figure 1. The program should instantiate 500 Employee objects based on a query to the database. Hint: be sure to use Maven to get all necessary sql classes. Upload your code to GitHub with the commit message "Question 2a complete"

b. The ComboBox for selecting the area code should be populated with a sorted list of distinct area codes from the employee's phone numbers. The area code is the first 3 digits of their phone number. Think about what data structure would both order the area codes AND prevent duplicates. Hint: substring() could be useful for getting the area codes from a telephone number. areaComboBox.getItems().addAll() can be used to pass in a collection. If you cannot figure out how to dynamically get the area codes, hard code some in the areaCodeComboBox. Upload your code to GitHub with the commit message "Question 2b complete"

c. The numOfEmployeesLabel should indicate how many employees are in the table. Note: do NOT hard code this to say "Number of Employees: 500", have the code derive it from the TableView. Upload your code to GitHub with the commit message "Question 2c complete"

Requirements – Filter

The ontarioOnlyCheckBox is connected to a method called "ontarioOnlyCheckBox_OnClick()". areaCodeComboBox is connected to a method called "areaCodeComboBox_OnClick ()".

If the user selects/deselects from any of them, that method will be called. Your task is to update the GUI with the following capabilities.

Here a couple of hints:

- TableView objects contain an ObservableList. You can empty the list with the method clear()
- If you want to use regex to search for an area code, you should add .* to the end of the area code. For example, if looking for "416", the regex would be "416.*"

d. If the ontarioOnlyCheckBox is selected, the TableView should be updated such that only Employees from Ontario are showing. The province field should only have employees from 'ON'. The number of employees label is updated to show the number of Ontario employees (175). If the Ontario only check box is deselected, the tableview should no longer be filtered based on province and the number of employees label should be updated to the number of employees in the table (500 if no other filters are on). Upload your code to GitHub with the commit message "Question 2d complete"

e. When an area code is selected from the areaCodeComboBox, the TableView should be updated to only have employees with that area code in their phone number. The number of employees label should be updated to reflect the number of employees in the table. If the user selects "All", the TableView should not have any filter based on area code.

- The area code is the first 3 digits of the phone number.
- ***areaCodeComboBox.getSelectionModel().getSelectedItem();*** will return the String selected in the comboBox
- Upload your code to GitHub with the commit message "Question 2e complete"

f. The logic / order of applying the filters should not impact the results. For example, on the initial load, there should be 500 employes. If Ontario was selected as a filter, it would drop to 175 employees. If the user then selects the 519 area code, it should **DESELECT** ontarioOnlyCheckBox, and it would contain 37 employees.

   (Hint: ontarioOnlyCheckBox.setSelected(false) will change the value via program)

   Upload your code to GitHub with the commit message "Question 2f complete"

g. If the user now selects ontarioOnlyCheckBox, the selected areaCode. i.e., 519 should go back to say **"All"** and display 175 employees.

   (Hint: areaCodeComboBox.getSelectionModel().clearAndSelect(0); will change the value to the value at first location in comboBox via program)

   Upload your code to GitHub with the commit message "Question 2g complete"

Rubric

Question	Level 0	Level 1	Level 2	Level 3
1-validation	Constructor does not validate inputs	Constructor validates inputs, but does not use set methods	Level 2, plus validation for a, b, d and f are correct	Level 2 plus questions c & e are fully functioning
1-structure/format	Code does not follow coding best practices (i.e., instance variables are not private, set methods are using different validation from the constructor)	Mostly good, however, there is at least 1 indenting/format issue	Everything looks clean and professional	
2.a) Initial load	The TableView object is not populated with Employee Objects, or an exception is triggered	There are Employee objects in the table, but it is not populated from the DB	Level 1, plus the Employee objects are created by querying the DB	Level 2 plus all objects used to access the DB are closed

Question	Level 0	Level 1	Level 2	Level 3
2.b) Area code combo Box	The combobox is not populated with any area codes	The combobox is populated with area codes, but they are "hard coded", not dynamically generated from the list of Employee's	The list of Employee objects is used to derive all the area codes	The area codes are sorted with no duplicates because the user used a smart data structure.
2.c) # of employees Label	The label is not updated, or it is simply hard coded	The TableView is used to determine how many Employee objects are in the table		
2.d) Ontario only CheckBox	Selecting/deselecting the checkbox does not update the Employee objects in the table or triggers an exception	The tableView is updated to show only employees from Ontario (ON)	Level 1, plus if the checkbox is deselected, the tableview updates to hold all employees irrespective of their province	Level 2, plus the number of employees label is updated each time to show how many employees are in the table.
2.e) Area code filter	Selecting an area code from the combobox does not adjust the number of Employee objects and/or triggers an exception	The tableView is updated to show only employees with the selected areacode.	The number of employees label is updated to show how many employees are in the table	There is a selection available at the top of the combobox titled "All", this will effectively remove the filter and update the number of employees label.
2.f) Order of OntarioOnly filter	Filters trigger an exception under any circumstance			The checkBox objects can coordinate with comboBox and get the correct results
2.g) Order of areaCode filter	Filters trigger an exception under any circumstance			The comboBox objects can coordinate with checkBox and get the correct results

Question	Level 0	Level 1	Level 2	Level 3
GitHub	The URL provided for the repo is incorrect or PriyanshT is not added as a collaborator	GitHub is updated after most (approx. 75% of the questions)	GitHub is updated after EVERY question	

Submitting your work

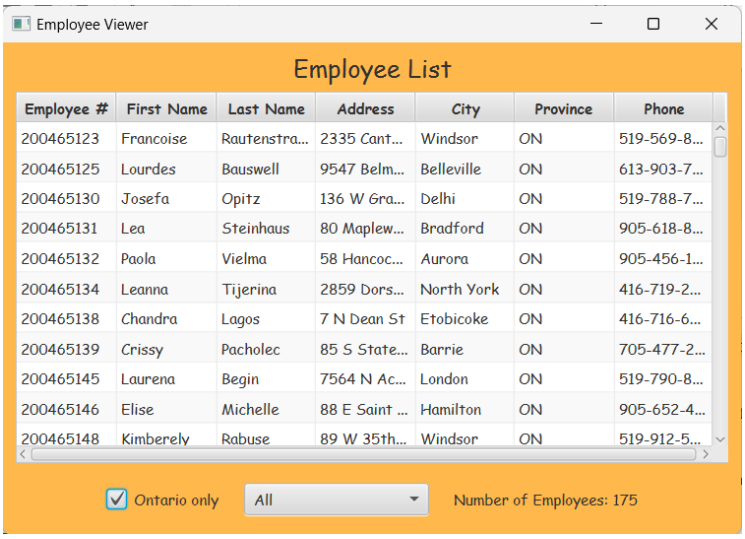
As noted above in the GitHub section, your work needs to be captured in a private GitHub repository with PriyanshT as a collaborator. (No collaborator if using classroom). Your code should be uploaded to GitHub at the end of each challenge.

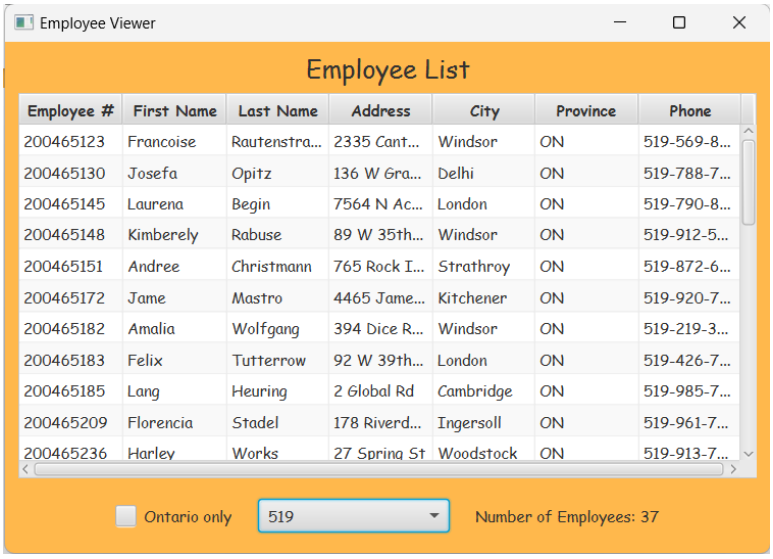
Once you are confident that you have completed the test with quality, submit the link to your private GitHub repository into Blackboard.

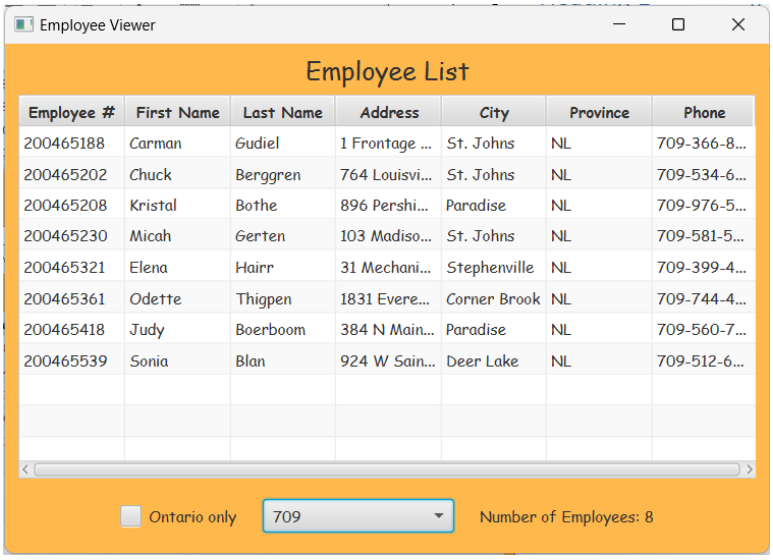
You MUST submit your work prior to 1:00 pm or your test will not be marked. So, manage your time carefully and be ready to submit prior to the deadline.

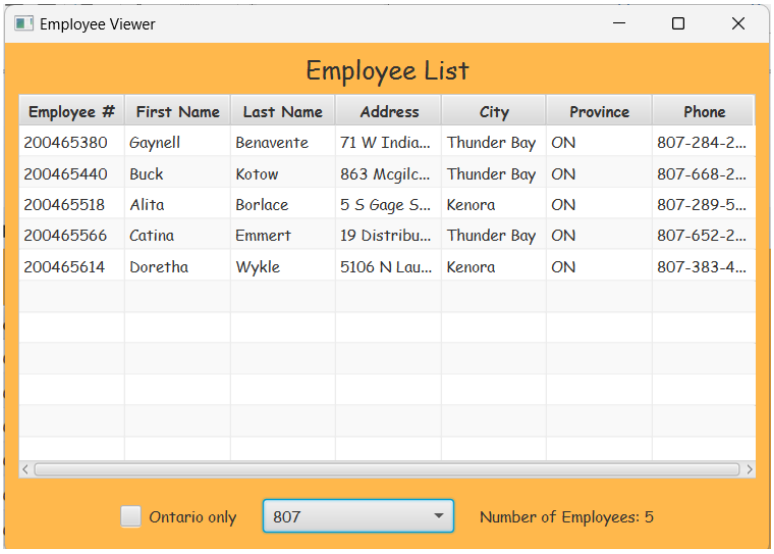
All work on this test must be your own.

Screenshots




Employee Viewer
Employee List
Employee # First Name Last Name Address City Province Phone
200465123 Francoise Rautenstra... 2335 Cant... Windsor ON 519-569-8...
200465130 Josefa Opitz 136 W Gra... Delhi ON 519-788-7...
200465145 Laurena Begin 7564 N Ac... London ON 519-790-8...
200465148 Kimberely Rabuse 89 W 35th... Windsor ON 519-912-5...
200465151 Andree Christmann 765 Rock I... Strathroy ON 519-872-6...
200465172 Jame Mastro 4465 Jame... Kitchener ON 519-920-7...
200465182 Amalia Wolfgang 394 Dice R... Windsor ON 519-219-3...
200465183 Felix Tutterrow 92 W 39th... London ON 519-426-7...
200465185 Lang Heuring 2 Global Rd Cambridge ON 519-985-7...
200465209 Florencia Stadel 178 Riverd... Ingersoll ON 519-961-7...
200465236 Harley Works 27 Spring St Woodstock ON 519-913-7...
Ontario only
519
Number of Employees: 37


Employee Viewer
Employee List
Employee # First Name Last Name Address City Province Phone
200465188 Carman Gudiel 1 Frontage ... St. Johns NL 709-366-8...
200465202 Chuck Berggren 764 Louisvi... St. Johns NL 709-534-6...
200465208 Kristal Bothe 896 Pershi... Paradise NL 709-976-5...
200465230 Micah Gerten 103 Madiso... St. Johns NL 709-581-5...
200465321 Elena Hairr 31 Mechani... Stephenville NL 709-399-4...
200465361 Odette Thigpen 1831 Evere... Corner Brook NL 709-744-4...
200465418 Judy Boerboom 384 N Main... Paradise NL 709-560-7...
200465539 Sonia Blan 924 W Sain... Deer Lake NL 709-512-6...
Ontario only
709
Number of Employees: 8


Employee Viewer
Employee List
Employee # First Name Last Name Address City Province Phone
200465380 Gaynell Benavente 71 W India... Thunder Bay ON 807-284-2...
200465440 Buck Kotow 863 Mcgilc... Thunder Bay ON 807-668-2...
200465518 Alita Borlace 5 S Gage S... Kenora ON 807-289-5...
200465566 Catina Emmert 19 Distribu... Thunder Bay ON 807-652-2...
200465614 Doretha Wykle 5106 N Lau... Kenora ON 807-383-4...
Ontario only
807
Number of Employees: 5