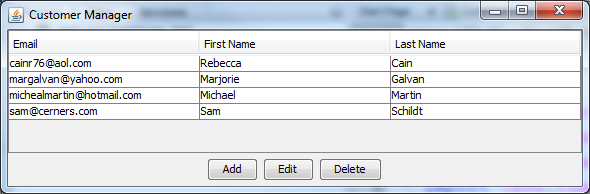
**Customer data Management System (GUI)**



## Operation

* This application begins by displaying a table of customer data.
* If the user clicks the Add button, the application allows the user to add customer data to the table (and the underlying database).
* If the user selects a customer row and clicks the Edit button, the application allows the user to update the data for the selected customer row in the table (and the database).
* If the user selects a customer row and clicks the Delete button, the application deletes the selected customer row from the table (and the database).

## Specifications

* Create a **table** in the mma database described in **chapter 19** to store the necessary data. To do that, you can use the SQL script stored in the create\_customer\_table.sql file that’s supplied.
* Create a class named **Customer** that stores data for the **user’s id**, **email address**, **first name**, and **last name** as **instance** variables. Include a **constructor** for the **Customer** class which takes these four values as parameters to create an object. Create get and set methods for each of the four instance variables.
* Create a class named **CustomerDB** that contains the methods necessary to get an **array list of Customer objects**, to get **a Customer object** for the customer with the specified id, and to **add**, **update**, or **delete** the specified customer. Other classes to support the data layer functions are allowed.
* Create a **CustomerManagerFrame** class like the one shown above. This frame should display a table of customer data as well as the Add, Edit, and Delete buttons. This class must use the Customer and CustomerDB classes to work with the customer data.
* Create a **CustomerTableModel** class for the **JTable** control used in the **CustomerManagerFrame** class.
* Create a **CustomerForm** class that allows the user to add or edit customer data.

Create a GUI Java Customer Management program with NetBeans that satisfies the specifications above. Projects are cumulative and can require the use of material from any of the previous chapters covered in the textbook.

Put **General Comments** at the beginning of the project that includes (1) your name, (2) the project name, (3) the date, and (4) a description of the project.

Create **screen shots** of the GUI output by using **Alt-Print Screen**. This will capture the active window of the GUI only, not the entire screen. Then use Paste to copy the GUI image into the **Word document**. This will be the documentation for testing the program. Show multiple screen shots of the program to verify that the program works properly.

Turn in a ZIP file of the final version of the program. The **Word document** must be inside the ZIP file.

In the **Comments** **section** on the **Assignment** webpage, report (A) an **estimate** of the **time** it took to complete the project. Report a single value in minutes, and (B) a **single** **rating** of the project, on an ordinal scale, as either (1) Easy, (2) Moderate, (3) Hard, OR (4) Challenging.

USE mma;

DROP TABLE IF EXISTS Customer;

CREATE TABLE Customer

(

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

EmailAddress VARCHAR(50)

);

INSERT INTO Customer VALUES

(2, 'Michael ', 'Martin', 'michealmartin@hotmail.com'),

(3, 'Marjorie ', 'Galvan', 'marjgalvan@yahoo.com'),

(4, 'Rebecca', 'Cain', 'cainr76@aol.com'),

(5, 'Sam', 'Schildt', 'sam@cerners.com');