Derrick Fox

CS 214 - Advanced Java

April 20, 2015

Project 9 - Database

Algorithm, UML Diagram and Screen Shots

Environment:

\*\*SPECIAL NOTE\*\*For some reason the GUI will not let you Insert, Update, or View a record unless all the fields have been cleared first. So please click the “Clear” button first before trying any other functions. I will come into for tutoring hours to ask about this bug.

* This program assumes that you have a database already installed on your root directory named “javabook”.
* This program assumes that the proper MySQL JAR files has been included and that the build-path has that JAR file included.
* This program creates a database table called “Staff” and if the table already exists, the program will drop that table and create a new Staff table.
* The “View” feature works by first using the “Clear” button to clear all fields. Then enter an Employee’s ID in the “ID” field on the top of the window and then click the “View” button.
* The program starts with three employees already entered into the table. Their IDs are 1, 2, and 3.

**StaffDatabase.java**: (I am building upon the “AddressBook” file from the previous project but I could not successfully change the name without throwing an exception. So it is still named AddressBookl.java)

StaffDatabase:

1. Declare text fields to enter Employee data.
2. Declare buttons to process commands.
   1. “Insert” button to add an employee record
   2. “Update” button to update an existing employee record.
   3. “View” button to update
   4. “Clear” button clears all text fields but does not save it to the database.

start:

1. Declare a Connection object using the pathway, username, and password.
2. Declare a Statement object from the Connection object.
3. Create a String that holds a SQL query which creates a Staff table in your local database.
4. Declare listeners and event handlers for the buttons.

writeEmployee:

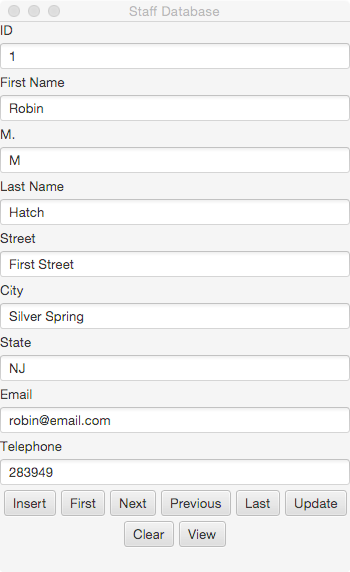
1. Create a Connection object
2. Create a SQL update statement to Insert an employee
3. Run the update statement.

viewEmployee:

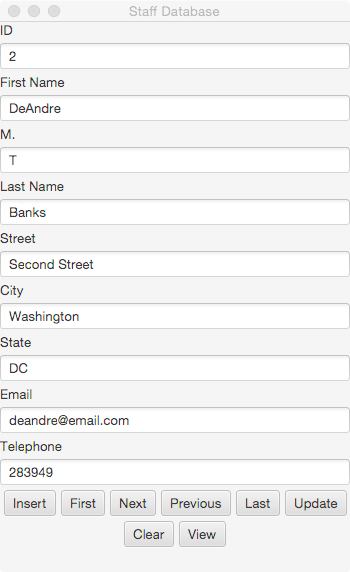
1. Create a Connection object
2. Create a SQL query to View an employee’s information.
3. Run the query.

updateEmployee:

1. Create a Connection object
2. Create a SQL update statement.
3. Run the update statement.

**Screen Shots**

| StaffDatabase |
| --- |
| - id: TextField |
| - lastName: TextField |
| - firstName: TextField |
| - mi: TextField |
| - address: TextField |
| - city: TextField |
| - state: TextField |
| - telephone: TextField |
| - email: TextField |
| - myConn: Connection |
| - myStmt: PreparedStatement |
| + start: void |
| + main: void |
| + clear: void |
| + writeStaff: void |
| + viewStaff: void |
| + updateStaff: void |

****