CSE1007 - Java Programming

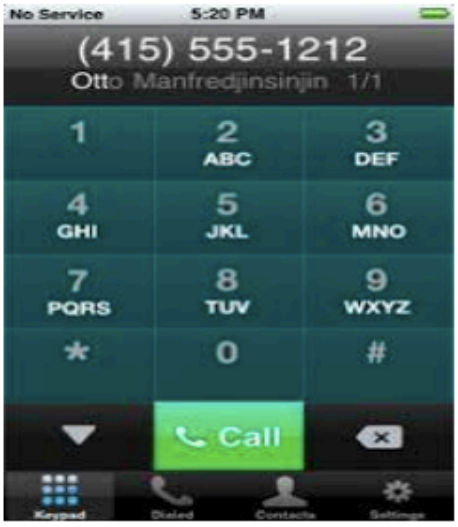
Lab Final Assessment Test

JavaFX with Keypad and Contacts

SANJIT KUMAR
18BCE0715
DR ANITHA A
LAB - L5 + L6
7 JUNE 2021

Question/Task

1.



Develop an Java Fx with 10 buttons for digits 0 to 9 with each digit corresponding to a group of alphabets as in the above smart keypad used for dialling. Store at least a list of 30 names and their corresponding phone numbers in an array. Upon pressing the buttons for dialling, those names that match the letters pressed should be displayed in a list. Assuming 25 names match when only the first button is pressed, all those 35 names should be displayed in the list. When the second button is pressed, assuming only 10 out of those 25 names match, only those 10 names should be displayed in the list. This should continue till an exact match or no match is found. If exact match not found it should display "Number does not exists" information to the user and allowing them to get the entry for the second chance.

Source Code and Screenshots

FXML

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ListView?>
<?import javafx.scene.layout.AnchorPane?>


<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-
Infinity" minWidth="-Infinity" prefHeight="437.0" prefWidth="287.0"
xmlns="http://javafx.com/javafx/16" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.PhoneController">
   <children>
      <Button layoutX="28.0" layoutY="49.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="1" />
      <Button layoutX="108.0" layoutY="49.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="2" />
      <Button layoutX="192.0" layoutY="49.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="3" />
      <Button layoutX="28.0" layoutY="116.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="4" />
      <Button layoutX="108.0" layoutY="116.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="5" />
      <Button layoutX="192.0" layoutY="116.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="6" />
      <Button layoutX="28.0" layoutY="191.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="7" />
      <Button layoutX="108.0" layoutY="191.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="8" />
      <Button layoutX="192.0" layoutY="191.0" mnemonicParsing="false"
prefHeight="41.0" prefWidth="55.0" text="9" />
      <ListView layoutX="33.0" layoutY="259.0" prefHeight="135.0"
prefWidth="205.0" />
   </children>
</AnchorPane>
```

Database Connection

```
package utils;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConnectionUtil {
    Connection conn = null;
    public static Connection conDB()
    {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://
localhost/sanjit_tables", "root", "Luckyusa1");
            return con;
        } catch (ClassNotFoundException | SQLException ex) {
            System.err.println("ConnectionUtil : "+ex.getMessage());
           return null;
        }
    }
    //make sure you add the lib
}
```

Main Class

```
package home;

import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class Main extends Application {

    //define your offsets here
    private double xOffset = 0;
    private double yOffset = 0;

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("/fxml/
keypad.fxml"));
        stage.initStyle(StageStyle.DECORATED);
        stage.setMaximized(false);
```

```
        //grab your root here
        root.setOnMousePressed(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                xOffset = event.getSceneX();
                yOffset = event.getSceneY();
            }
        });

        //sorry about that - Windows defender issue.
        //move around here
        root.setOnMouseDragged(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                stage.setX(event.getScreenX() - xOffset);
                stage.setY(event.getScreenY() - yOffset);
            }
        });
        stage.setTitle("Yarn Trading Inventory Management");
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Controller Code

PhoneController.java

```
package controllers;

//import com.mysql.jdbc.Connection;
//import com.mysql.cj.jdbc.PreparedStatement;
import java.io.IOException;
import java.sql.*;
import java.net.URL;
import java.util.ArrayList;
import java.util.ResourceBundle;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.DatePicker;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableColumn.CellDataFeatures;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.util.Callback;
import utils.ConnectionUtil;

public class PhoneController implements Initializable {
    @FXML
    TableView tblData;

    @FXML
    private Button one;
    @FXML
    private Button two;
    @FXML
    private Button three;
    @FXML
    private Button four;
    @FXML
    private Button five;
    @FXML
    private Button six;
    @FXML
    private Button seven;
```

```
    @FXML
    private Button eight;
    @FXML
    private Button nine;

    PreparedStatement preparedStatement;
    Connection connection;

    public PhoneController() {
        connection = (Connection) ConnectionUtil.conDB();
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        fetColumnList();
        fetRowList();
    }

    private ObservableList<ObservableList> data;
    String SQL = "SELECT * from phone_book";

    //only fetch columns
    private void fetColumnList() {

        try {
            System.out.println("statement executes");
            ResultSet rs =
connection.createStatement().executeQuery(SQL);

            //SQL FOR SELECTING ALL OF CUSTOMER
            for (int i = 0; i < rs.getMetaData().getColumnCount(); i++)
{
                //We are using non property style for making dynamic
table
                final int j = i;
                TableColumn col = new
TableColumn(rs.getMetaData().getColumnName(i + 1).toUpperCase());
                col.setCellValueFactory(new
Callback<CellDataFeatures<ObservableList, String>,
ObservableValue<String>>() {
                    public ObservableValue<String>
call(CellDataFeatures<ObservableList, String> param) {
                        return new
SimpleStringProperty(param.getValue().get(j).toString());
                    }
                });

                tblData.getColumns().removeAll(col);
                tblData.getColumns().addAll(col);

                System.out.println("Column [" + i + "] ");

            }

        } catch (Exception e) {
            System.out.println("Error " + e.getMessage());
```

```
        }
    }

    //fetches rows and data from the list
    private void fetRowList() {
        data = FXCollections.observableArrayList();
        ResultSet rs;
        try {
            rs = connection.createStatement().executeQuery(SQL);

            while (rs.next()) {
                //Iterate Row
                ObservableList row =
FXCollections.observableArrayList();
                for (int i = 1; i <= rs.getMetaData().getColumnCount();
i++) {
                    //Iterate Column
                    row.add(rs.getString(i));
                }
                System.out.println("Row [1] added " + row);
                data.add(row);

            }

            tblData.setItems(data);
        } catch (SQLException ex) {
            System.err.println(ex.getMessage());
        }
    }

    private boolean compare_name(String name, String pattern){
        String patt="";
        name=name.toUpperCase();
        for (int i = 0; i < name.length(); i++){
            char letter = name.charAt(i);
            if (one.getValue()=='A' || one.getValue()=='B' ||
one.getValue()== 'C')
            {
                patt+="2";
            }

            else if(two.getValue()=='D' || two.getValue()=='D' ||
two.getValue()=='D')
            {
                patt+="3";
            }

            else if (three.getValue()=='D' || three.getValue()=='D'||
three.getValue()=='D')
            {
                patt+="4";
            }
            else if (four.getValue()=='D' || four.getValue()=='D' ||
four.getValue()=='D')
            {
```

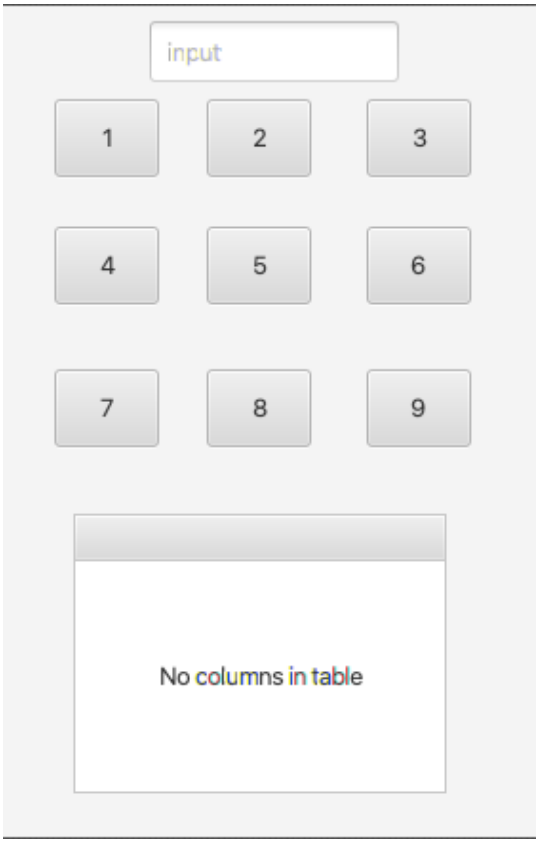
```
                patt+="5";
            }
            else if (five.getValue()=='D' || five.getValue()=='D'||
five.getValue()=='D')
            {
                patt+="6";
            }

            else if (six.getValue()=='D' || six.getValue()=='D'||
six.getValue() == 'S')
            {
                patt+="7";
            }

            else if (six.getValue()=='T' || six.getValue()=='U' ||
six.getValue()== 'V')
            {
                patt+="8";
            }

            else if (six.getValue()=='W' || six.getValue()=='X' ||
six.getValue()== 'Y' || six.getValue()== 'Z')
            {
                patt+="9";
            }
        }
        System.out.println(patt);
        if(pattern.equals(patt.substring(0,pattern.length())))
            return true;

        return false;

    }

}
```

Output

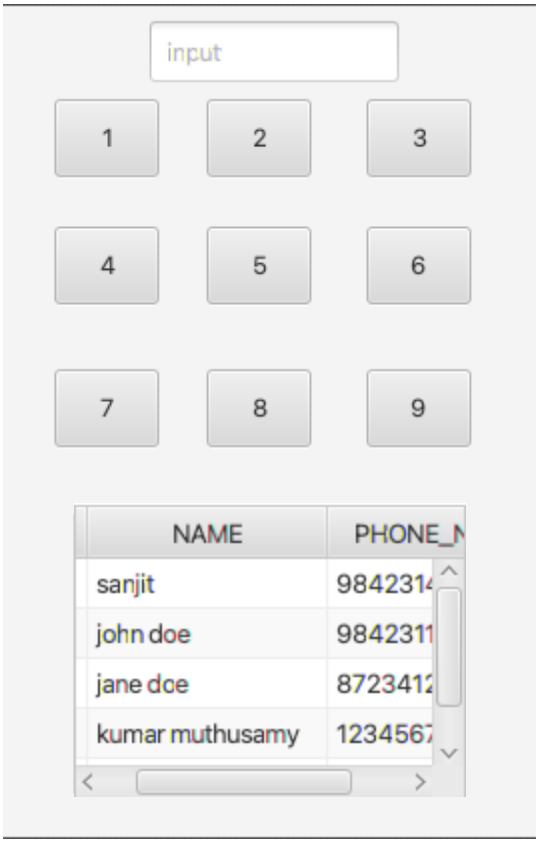
Creation of Database and Insertion of Values in Phone Table

```
~ — mysql -u root -p
Last login: Mon Jun  7 08:34:45 on ttys001
sanjitkumar@Sanjits-MacBook-Air ~ % /usr/local/mysql-8.0.17-macos10.14-x86_64/bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create table phone_book(id int not null primary key auto_increment, name varchar(20) not null, phone_no varchar(20) not null);
ERROR 1046 (3D000): No database selected
mysql> use sanjit_tables;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table phone_book(id int not null primary key auto_increment, name varchar(20) not null, phone_no varchar(20) not null);
Query OK, 0 rows affected (0.07 sec)

mysql> insert into phonebook(name,phone_no) values ("sanjit",9842314733)
    -> ;
ERROR 1146 (42S02): Table 'sanjit_tables.phonebook' doesn't exist
mysql> insert into phone_book(name,phone_no) values ("sanjit",9842314733);
Query OK, 1 row affected (0.01 sec)

mysql> insert into phone_book(name,phone_no) values ("john doe",9842311133);
Query OK, 1 row affected (0.00 sec)

mysql> insert into phone_book(name,phone_no) values ("jane doe",8723412345);
Query OK, 1 row affected (0.01 sec)

mysql> insert into phone_book(name,phone_no) values ("kumar muthusamy",1234567890);
Query OK, 1 row affected (0.00 sec)

mysql> insert into phone_book(name,phone_no) values ("veena kumar",1456234561);
Query OK, 1 row affected (0.00 sec)

mysql>
```

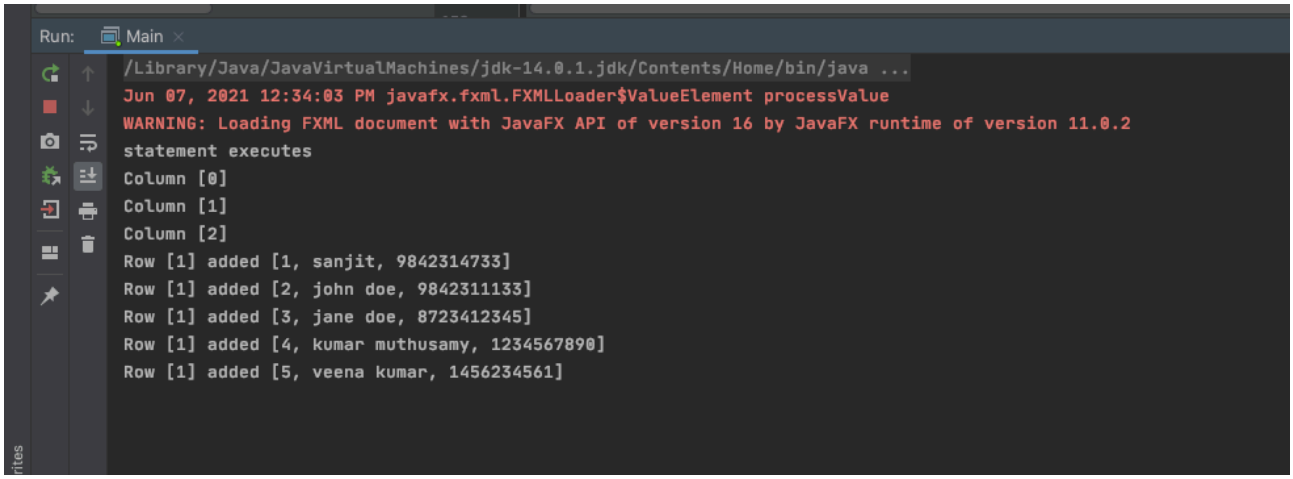
Interface Created with JavaFX



On hitting nothing all the entries are displayed without filtering



Logs printed on retrieval of rows



On searching for a full number we get item from the data base in the TableView component

On searching for partial numbers we get recommendations.