/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package HomeAffairsPackage;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\*

\* @author byronkraus

\*/

public class AppointmentManager {

private DatabaseManager dbm;//instantiates DatabaseManager class

//instantiates array of type Appointment

private Appointment[] appointmentArr = new Appointment[100];

private int size = 0;//instantiates array counter

private String error;//instantiates error string

public AppointmentManager() {//default constructor method

try {

//retrieves from Access database and creates Appointments

ResultSet rs = dbm.queryDatabase("SELECT \* FROM tblAppointment");

while (!(rs.next())) {

appointmentArr[size] = new Appointment(rs.getString(1),

rs.getString(2), rs.getString(3));

size++;

}

} catch (SQLException ex) {//sends error message

error = "Unable to retrieve information from tblAppointment";

ErrorScreen eng = new ErrorScreen(error);

eng.setVisible(true);

}

}

//adds a new appointment to database and array with retrieved fields

public void addAppointment(String cID, String Cat, String Date) {

appointmentArr[size] = new Appointment(cID, Cat, Date);

size++;

try {

//inserts new appointment into database

ResultSet rs = dbm.queryDatabase("INSERT INTO tblAppointment"

+ "(applicant\_ID, category\_ID, appointmentdatetime) "

+ "VALUES(" + "'cID'" + "','" + "'Cat'"

+ "','" + "'Date')");

//displays error screen + error if error occurs

} catch (SQLException ex) {

error = "Unable to insert into tblAppointment";

ErrorScreen eng = new ErrorScreen(error);

eng.setVisible(true);

}

}

//returns category with matching ID

public String getAppointmentCategory(String id) {

for (int i = 0; i < size; i++) {

if (appointmentArr[i].getID().equalsIgnoreCase(id)) {

return appointmentArr[i].getCategory();

}

}

return "Category not found";

}

//returns category with matching ID

public String getAppointmentDate(String id) {

for (int i = 0; i < size; i++) {

if (appointmentArr[i].getID().equalsIgnoreCase(id)) {

return appointmentArr[i].getAppointmentDateTime();

}

}

return "Date not found";

}

//deletes Appointment from database and array where ID is equal

public void deleteFromDatabase(String id) {

String category = null;

for (int i = 0; i < size; i++) {

if (appointmentArr[i].getID().equals(id)) {

shiftLeft(i);

category = appointmentArr[i].getNumberCategory();

}

}

try {

//deletes from database

ResultSet rs = dbm.queryDatabase("DELETE \* FROM tblAppointment "

+ "WHERE idnumber = 'id' AND category = 'category'");

//catches error and sends error to error screen

} catch (SQLException ex) {

error = "Unable to delete from tblAppointment";

ErrorScreen eng = new ErrorScreen(error);

eng.setVisible(true);

}

}

//removes Applicant at a chosen position from array

public void shiftLeft(int i) {

for (int j = i; j < size; j++) {

appointmentArr[j] = appointmentArr[j + 1];

}

appointmentArr[size] = null;

size--;

}

//deletes Appointment from database and array where ID is equal

public String[] checkArray(String[] array, String date) {

boolean check = false;

String test[] = new String[array.length];

int count = 0;

for (int i = 0; i < array.length; i++) {

check = false;

for (int j = 0; j < size; j++) {

if (appointmentArr[j].getAppointmentTime()

.equalsIgnoreCase(array[i])

&& appointmentArr[j].getAppointmentDate()

.equalsIgnoreCase(date)) {

check = true;

}

}

if (!check) {

test[count] = array[i];

count++;

}

}

return test;

}

}