UNIVERSITI POLY-TECH MALAYSIA
UPTM

Name: **MUHAMMAD HAZIQ BIN SULAIMAN**		**Section : 02**
ID Number: **AM2304013646**		
Lecturer: **WAN NOR ASNIDA BTE WAN JUSOH**		**Lab group / Tutorial group / Tutor (if applicable):**
Course and Course Code: SWC 4243		**Submission Date:**
Assignment No. / Title: LAB TASK		**Extension & Late submission:** **Disallowed**
Assignment Type:	**% of Assignment Mark**	**Returning Date:**

Penalties:

1. 10% of the original mark will be deducted for every one week period after the submission date.
2. No work will be accepted after two weeks of the deadline.
3. If you were unable to submit the coursework on time due to extenuating circumstances you may be eligible for an extension.
4. Extension will not exceed one week.

Declaration: I/we the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/ academic cheating.

Signature(s): 

Full Name(s): MUHAMMAD HAZIQ BIN SULAIMAN

This section may be used for feedback or other information

Table of Contents

1. Coding

1.1 FXMLDocumentController.java

```java
package genius_kindergarten;

import java.sql.Connection;
import java.sql.PreparedStatement;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import java.time.LocalDate;
import java.sql.ResultSet;
import javax.swing.JOptionPane;


public class FXMLDocumentController {

    @FXML
    private TextField name;
    @FXML
    private DatePicker date;
    @FXML
    private TextField age;
    @FXML
    private TextField address;
    @FXML
    private TextField allergies;
    @FXML
    private TableView<Student> table;
    @FXML
    private TableColumn<Student, String> tname;
    @FXML
    private TableColumn<Student, Integer> tage;
    @FXML
    private TableColumn<Student, LocalDate> tdate;
    @FXML
    private TableColumn<Student, String> taddress;
    @FXML
    private TableColumn<Student, String> tallergies;

    private ObservableList<Student> studentList = FXCollections.observableArrayList();

    int index = -1;
    Connection conn =null;
    ResultSet rs = null;
    PreparedStatement pst = null;


    @FXML
    public void initialize() {
        tname.setCellValueFactory(new PropertyValueFactory<>("name"));
        tage.setCellValueFactory(new PropertyValueFactory<>("age"));
        tdate.setCellValueFactory(new PropertyValueFactory<>("birthDate"));
        taddress.setCellValueFactory(new PropertyValueFactory<>("address"));
        tallergies.setCellValueFactory(new PropertyValueFactory<>("allergies"));

        studentList = mysqlconnect.getDatausers();
        table.setItems(studentList);
    }

    @FXML
  private void addStudent() {
    conn = mysqlconnect.ConnectDb();
    String nameText = name.getText();
    LocalDate birthDate = date.getValue();
    int ageValue;

```

```
67      try {
68          ageValue = Integer.parseInt(age.getText());
69      } catch (NumberFormatException e) {
70          return;
71      }
72
73      String addressText = address.getText();
74      String allergiesText = allergies.getText();
75
76
77      String sql = "INSERT INTO students (name, birth_date, age, address, allergies) VALUES (?, ?, ?, ?, ?)";
78
79      try {
80          pst = conn.prepareStatement(sql);
81          pst.setString(1, nameText);
82          pst.setDate(2, java.sql.Date.valueOf(birthDate));
83          pst.setInt(3, ageValue);
84          pst.setString(4, addressText);
85          pst.setString(5, allergiesText);
86
87          pst.executeUpdate();
88          studentList.add(new Student(nameText, birthDate, ageValue, addressText, allergiesText));
89          clearFields();
90          table.refresh();
91
93      } catch (Exception e) {
94      }
95  }
96
97      @FXML
98      private void updateStudent() {
99          Student selectedStudent = table.getSelectionModel().getSelectedItem();
100         if (selectedStudent != null) {
101             selectedStudent.setName(name.getText());
102             selectedStudent.setBirthDate(date.getValue());
103             selectedStudent.setAge(Integer.parseInt(age.getText()));
104             selectedStudent.setAddress(address.getText());
105             selectedStudent.setAllergies(allergies.getText());
106             table.refresh();
107             clearFields();
108         }
109     }
110
111     @FXML
112     private void deleteStudent() {
113         Student selectedStudent = table.getSelectionModel().getSelectedItem();
114         if (selectedStudent!= null) {
115             try {
116                conn = mysqlconnect.ConnectDb();
117                String query = "DELETE FROM students WHERE name = '" + selectedStudent.getName() + "';";
118               pst = conn.prepareStatement(query);
119                pst.executeUpdate();
120              studentList.clear();
121              studentList = mysqlconnect.getDatausers();
122              table.setItems(studentList);
123               clearFields();
124             }
125
126             catch (Exception e) {
127             }
128         }
129     }
```

```
130     @FXML
        private void selectStudent() {
132             Student selectedStudent = table.getSelectionModel().getSelectedItem();
133             if (selectedStudent != null) {
134                 name.setText(selectedStudent.getName());
135                 date.setValue(selectedStudent.getBirthDate());
136                 age.setText(String.valueOf(selectedStudent.getAge()));
137                 address.setText(selectedStudent.getAddress());
138                 allergies.setText(selectedStudent.getAllergies());
139             }
140         }
141
142         @FXML
143         private void clearFields() {
144             name.clear();
145             date.setValue(null);
146             age.clear();
147             address.clear();
148             allergies.clear();
149         }
150
151         @FXML
            private void resetFields() {
153             clearFields();
154         }
155
156         @FXML
            private void searchStudent() {
158             String nameText = name.getText().toLowerCase();
159             LocalDate birthDate = date.getValue();
160             String ageText = age.getText();
161             String addressText = address.getText().toLowerCase();
162             String allergiesText = allergies.getText().toLowerCase();
163
164             ObservableList<Student> filteredList = FXCollections.observableArrayList();
165
166             for (Student student : studentList) {
167                 boolean matches = true;
168
169                 if (!nameText.isEmpty() && !student.getName().toLowerCase().contains(nameText)) {
170                     matches = false;
171                 }
172                 if (birthDate != null && !student.getBirthDate().equals(birthDate)) {
173                     matches = false;
174                 }
175                 if (!ageText.isEmpty() && student.getAge() != Integer.parseInt(ageText)) {
176                     matches = false;
177                 }
178                 if (!addressText.isEmpty() && !student.getAddress().toLowerCase().contains(addressText)) {
179                     matches = false;
180                 }
181                 if (!allergiesText.isEmpty() && !student.getAllergies().toLowerCase().contains(allergiesText)) {
182                     matches = false;
183                 }
184
185                 if (matches) {
186                     filteredList.add(student);
187                 }
188             }
189
190             table.setItems(filteredList);
191             table.refresh();
192             clearFields();
193
194
195         }
196     }
```

1.2 Genius_kindergarden.java

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/javafx/FXML.java to edit this template
 */
package genius_kindergarten;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

/**
 *
 * @author MSI
 */
public class Genius_kindergarten extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }

}
```

1.3 Student.java

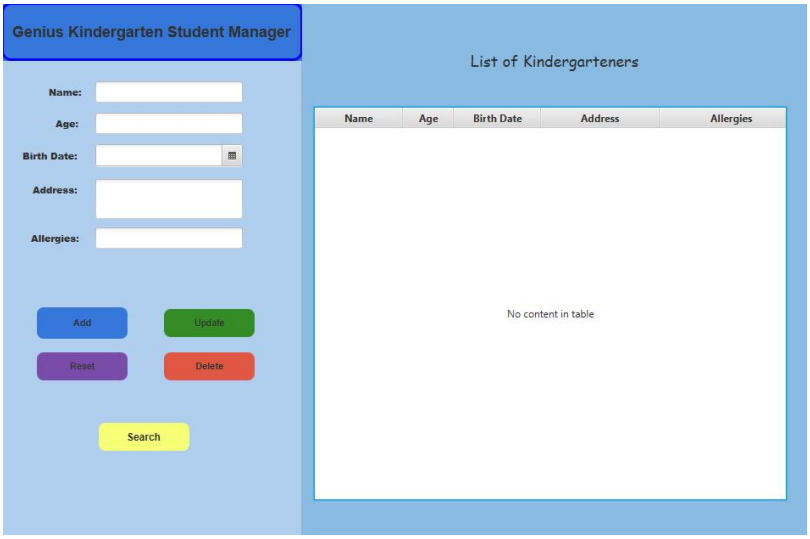
```java
package genius_kindergarten;

import java.time.LocalDate;
import javafx.scene.control.DatePicker;
import javafx.scene.control.TextField;

public class Student {
    private String name;
    private LocalDate birthDate;
    private int age;
    private String address;
    private String allergies;

    public Student(String name, LocalDate birthDate, int age, String address, String allergies) {
        this.name = name;
        this.birthDate = birthDate;
        this.age = age;
        this.address = address;
        this.allergies = allergies;
    }

    Student(TextField name, DatePicker date, TextField age, TextField address, TextField allergies) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }

    // Getters and Setters

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public LocalDate getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(LocalDate birthDate) {
        this.birthDate = birthDate;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getAllergies() {
        return allergies;
    }

    public void setAllergies(String allergies) {
        this.allergies = allergies;
    }

    int getId() {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }


}
```

1.4 mysqlconnect.java

```
package genius_kindergarten;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javax.swing.JOptionPane;

/**
 *
 * @author msi
 */
public class mysqlconnect {

    Connection conn = null;
    public static Connection ConnectDb(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/gk_students","root","");
           //JOptionPane.showMessageDialog(null, "Connection Established");
            return conn;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
            return null;
        }

    }

    public static ObservableList<Student> getDatausers(){
        Connection conn = ConnectDb();
        ObservableList<Student> list = FXCollections.observableArrayList();
        try {
            PreparedStatement ps = conn.prepareStatement("select * from students");
            ResultSet rs = ps.executeQuery();

            while (rs.next()){
                list.add(new Student(
    rs.getString("name"),
    rs.getDate("birth_date").toLocalDate(),
    rs.getInt("age"),
    rs.getString("address"),
    rs.getString("allergies")
));
            }
        } catch (Exception e) {
        }
        return list;
    }

}
```

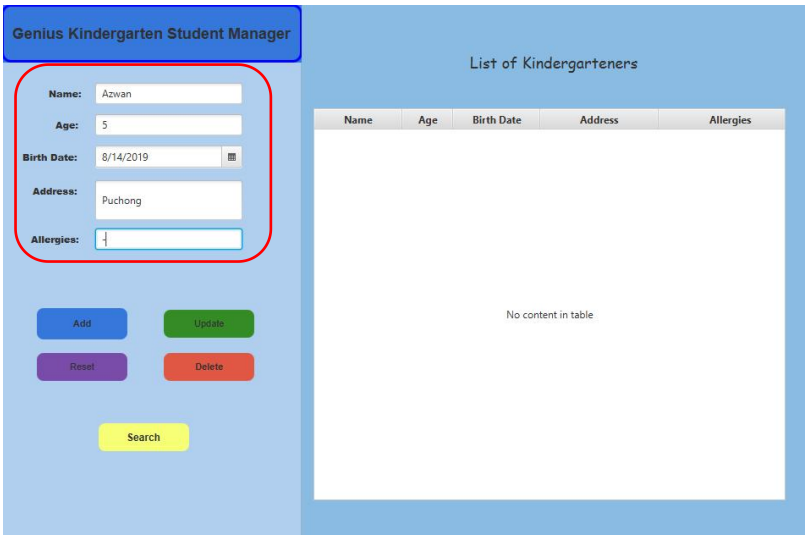
2. Output

2.1 Display Output

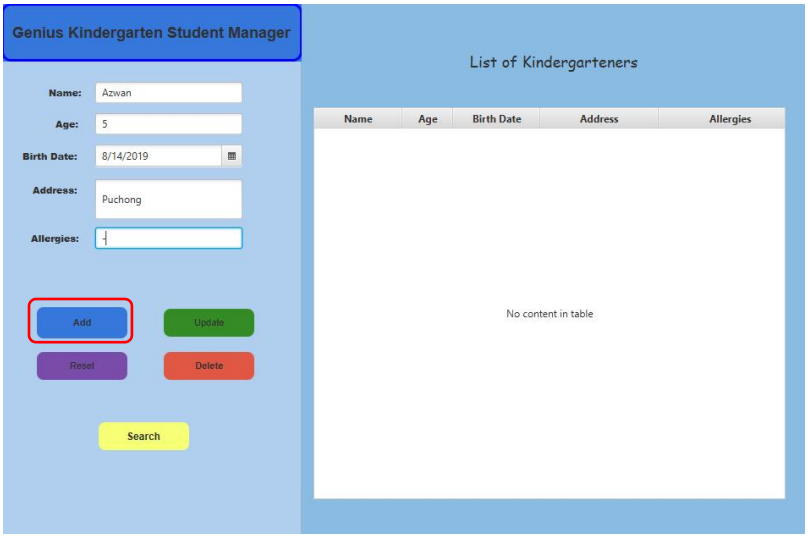


2.2 Add Function Output

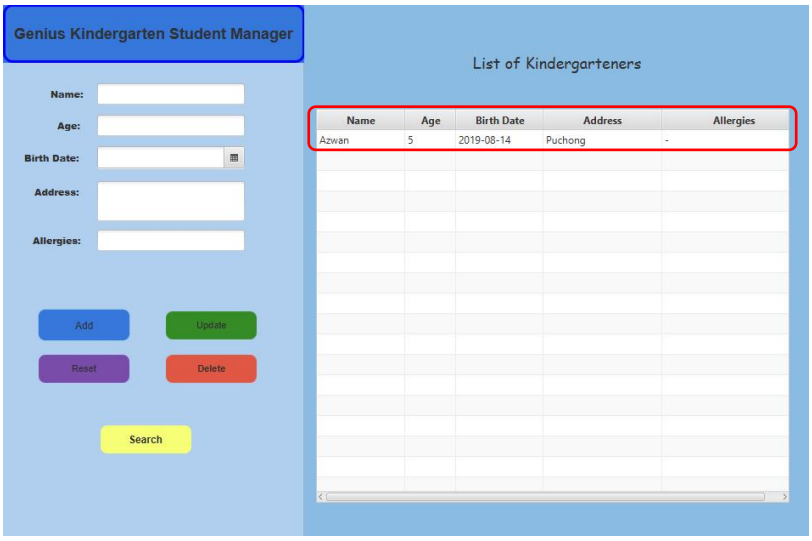
1. First, fill the box.



2. Second, click add button.

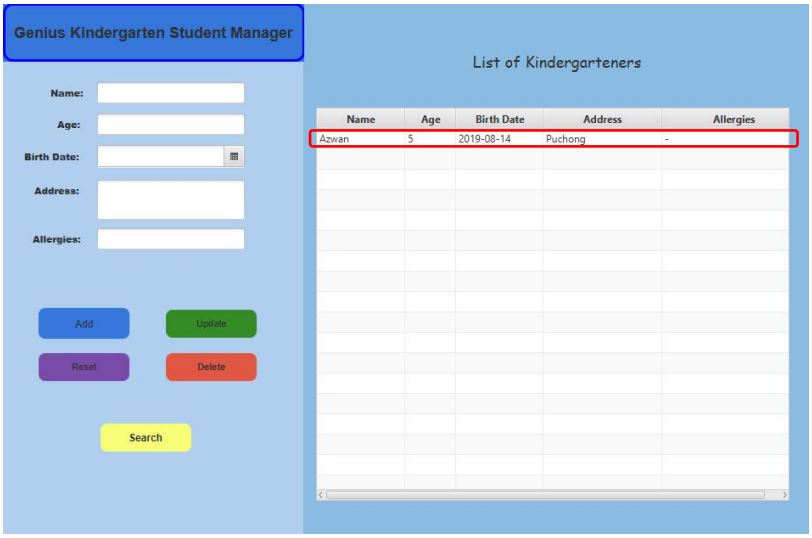


3. It will show at list.

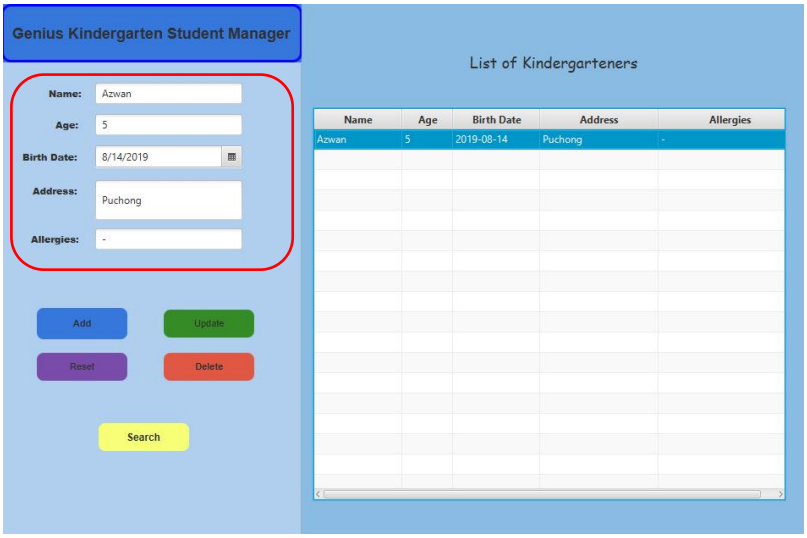


2.3 Update Function Output

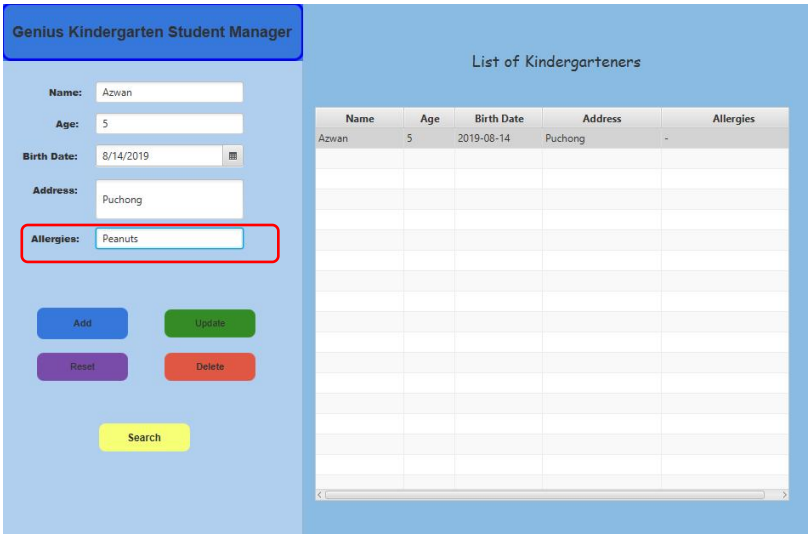
1. First, click at the name that want to change the data.



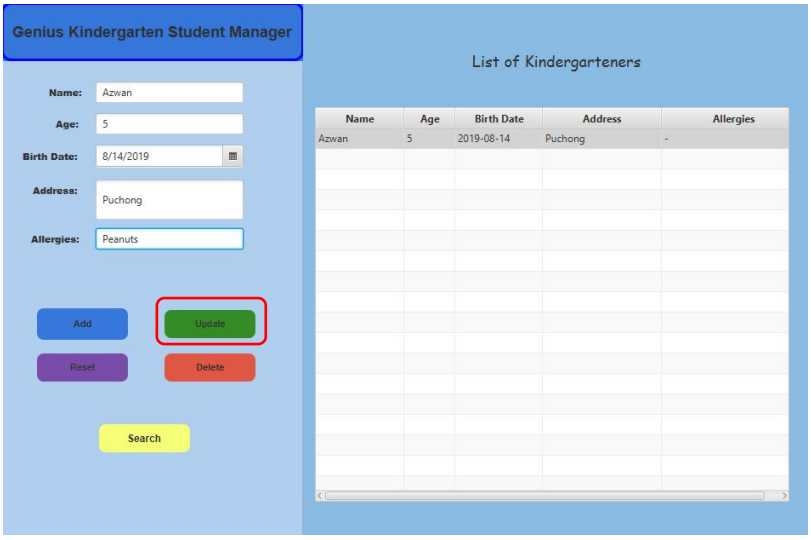
2. It will show the data in the box.



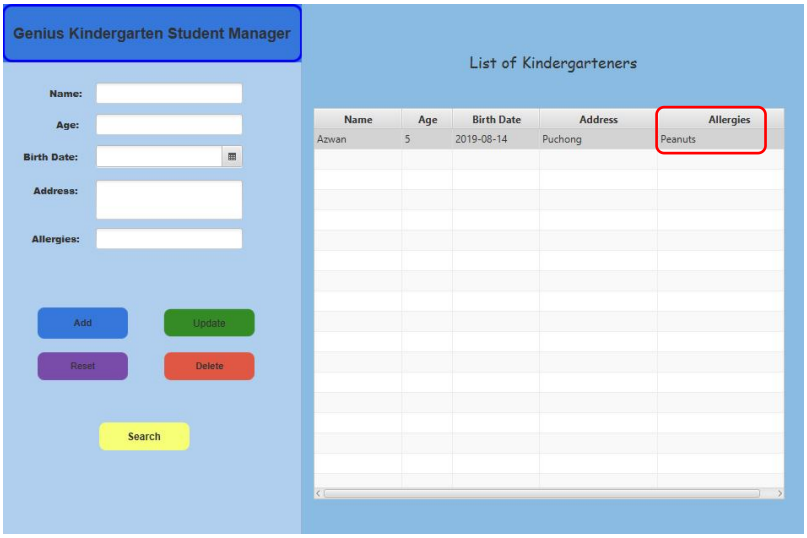
3. Then, change the data that want to update.



4. Click the update button.

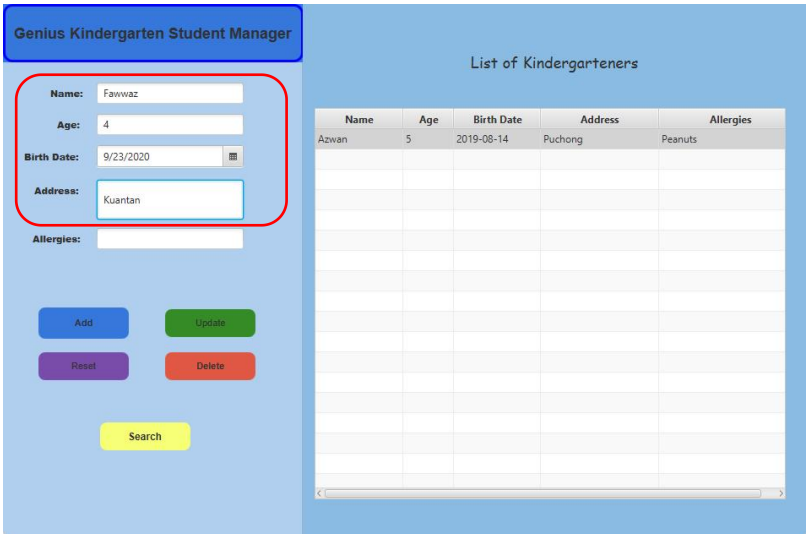


5. The list will automatically update the data that have been changed.

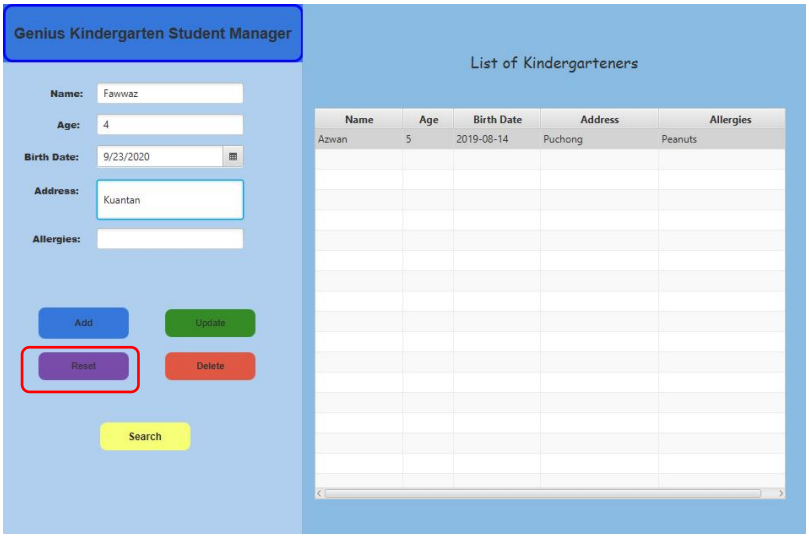


2.4 Reset Function Output

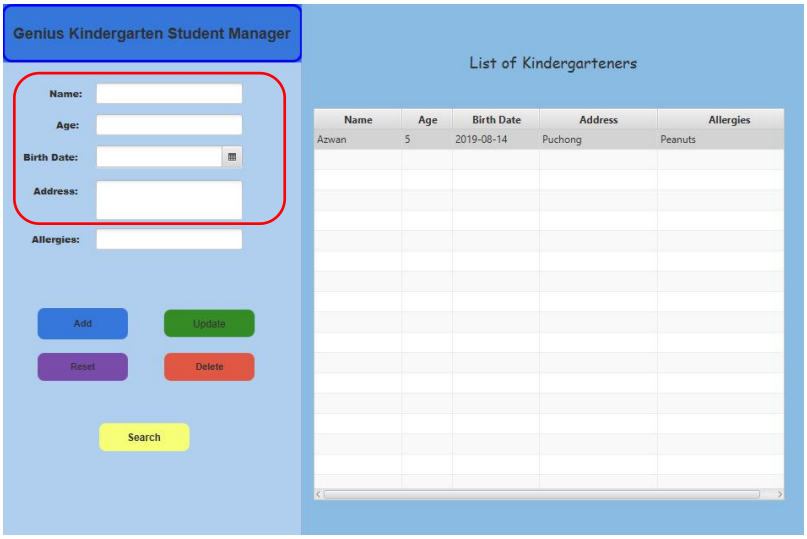
1. If want to clear the filled box.



2. Click button reset.

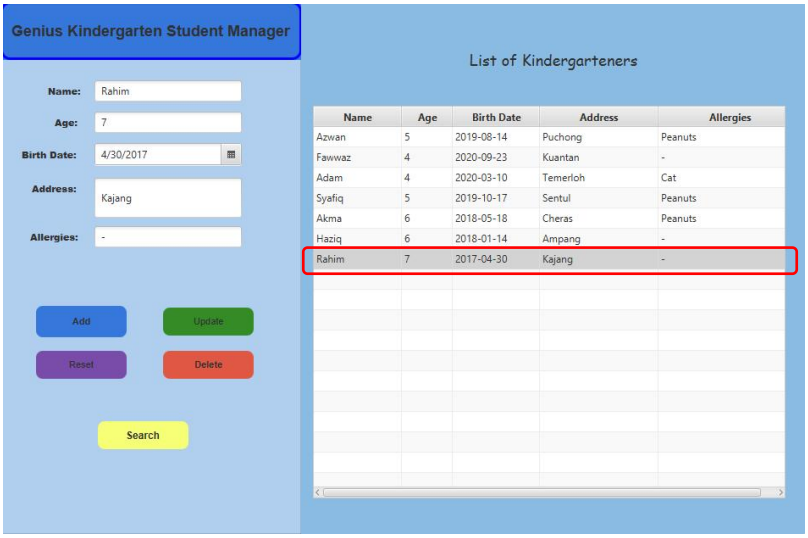


3. It will clear the box.

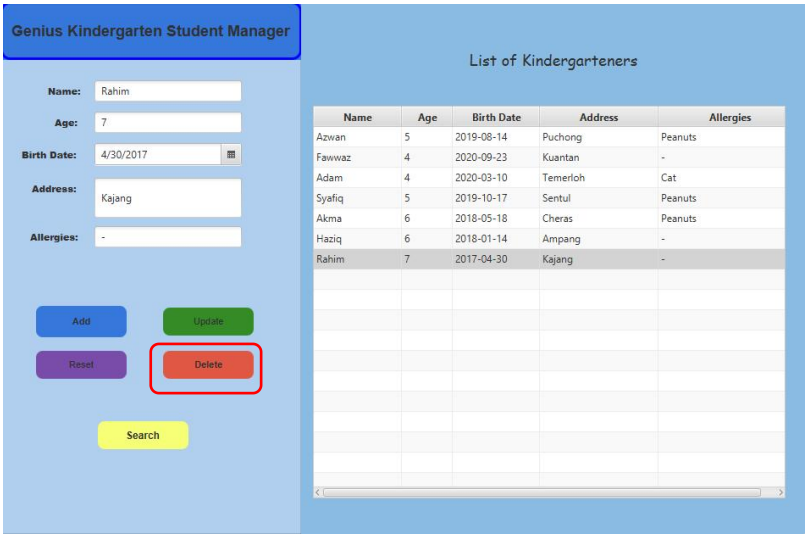


2.5 Delete Function Output

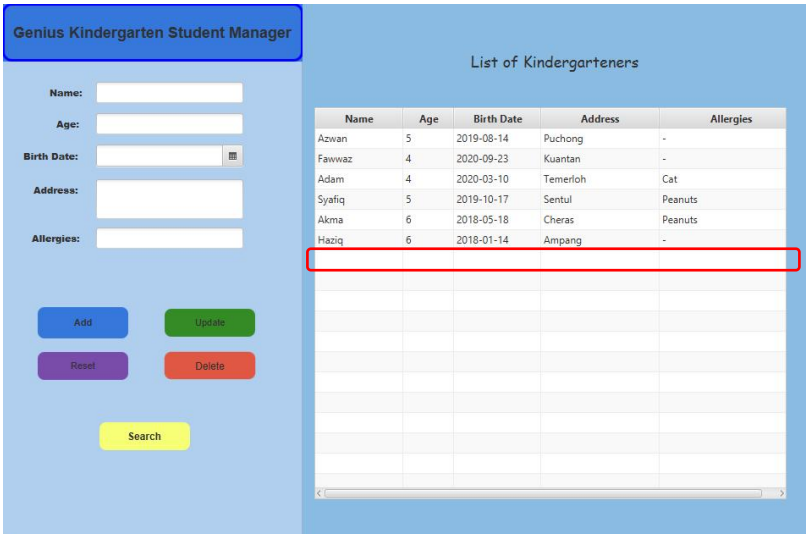
1. Click the data from the list that you want to delete.



2. Then, click button delete.

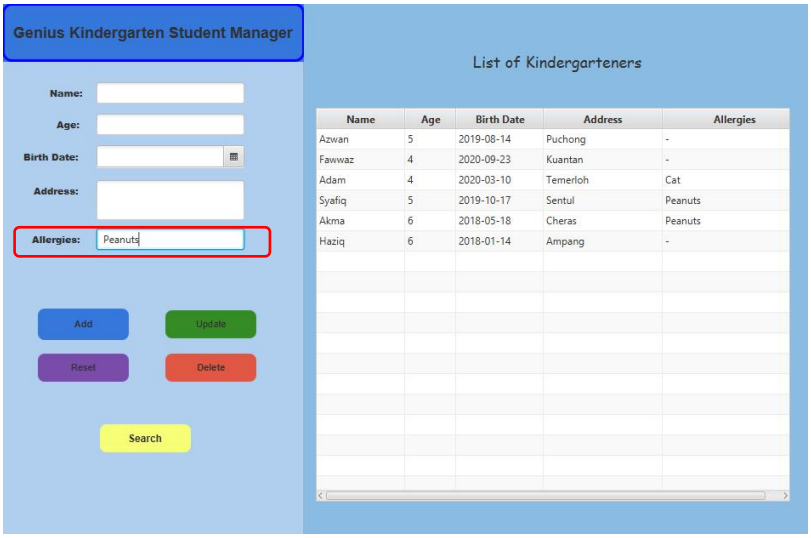


3. It will remove the selected data from the list.

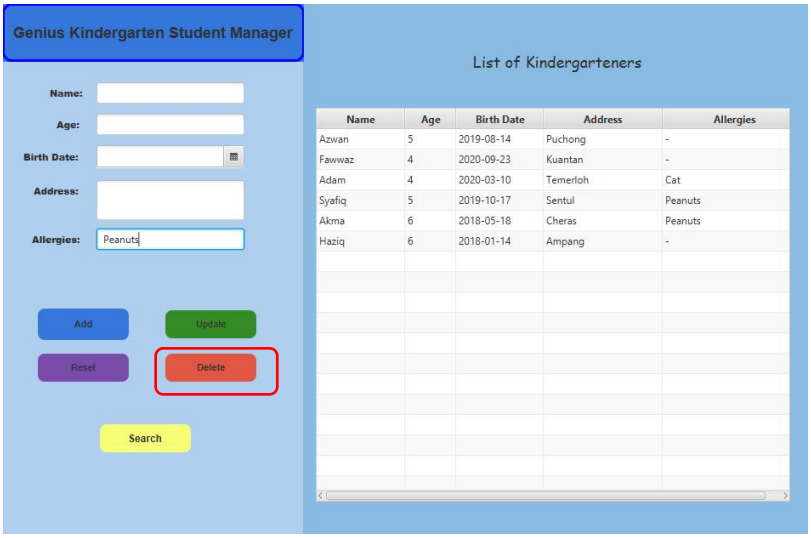


2.6 Search Function Output

1. Fill the box that want to search.



2. Click button search



3. It will show at list based on filled box.