import java.io.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

// Student Class

class Student implements Serializable {

private String studentId;

private String name;

private String rollNo;

private String className;

private double marks;

private String address;

public Student(String studentId, String name, String rollNo, String className, double

marks, String address) {

this.studentId = studentId;

this.name = name;

this.rollNo = rollNo;

this.className = className;

this.marks = marks;

this.address = address;

}

public String getStudentId() {

return studentId;

}

public String getName() {

return name;

}

public String getRollNo() {

return rollNo;

}

public String getClassName() {

return className;

}

public double getMarks() {

return marks;

}

public String getAddress() {

return address;

}

public String toString() {

return &quot;ID: &quot; + studentId + &quot;, Name: &quot; + name + &quot;, Roll No: &quot; + rollNo +

&quot;, Class: &quot; + className + &quot;, Marks: &quot; + marks + &quot;, Address: &quot; + address;

}

}

// StudentDatabase Class

class StudentDatabase {

private static final String FILE\_NAME = &quot;students.dat&quot;;

public void createDatabase(List&lt;Student&gt; students) {

try (ObjectOutputStream oos = new ObjectOutputStream(new

FileOutputStream(FILE\_NAME))) {

for (Student student : students) {

oos.writeObject(student);

}

System.out.println(&quot;Database created successfully.&quot;);

} catch (IOException e) {

e.printStackTrace();

}

}

public List&lt;Student&gt; displayDatabase() {

List&lt;Student&gt; students = new ArrayList&lt;&gt;();

try (ObjectInputStream ois = new ObjectInputStream(new

FileInputStream(FILE\_NAME))) {

while (true) {

Student student = (Student) ois.readObject();

students.add(student);

}

} catch (EOFException ignored) {

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

return students;

}

public void deleteRecord(String studentId) {

List&lt;Student&gt; students = displayDatabase();

students.removeIf(student -&gt; student.getStudentId().equals(studentId));

saveUpdatedDatabase(students);

System.out.println(&quot;Record deleted successfully.&quot;);

}

public void updateRecord(Student updatedStudent) {

List&lt;Student&gt; students = displayDatabase();

for (int i = 0; i &lt; students.size(); i++) {

if (students.get(i).getStudentId().equals(updatedStudent.getStudentId())) {

students.set(i, updatedStudent);

break;

}

}

saveUpdatedDatabase(students);

System.out.println(&quot;Record updated successfully.&quot;);

}

public Student searchRecord(String studentId) {

List&lt;Student&gt; students = displayDatabase();

for (Student student : students) {

if (student.getStudentId().equals(studentId)) {

return student;

}

}

return null;

}

private void saveUpdatedDatabase(List&lt;Student&gt; students) {

try (ObjectOutputStream oos = new ObjectOutputStream(new

FileOutputStream(FILE\_NAME))) {

for (Student student : students) {

oos.writeObject(student);

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

// Main Class

public class Main {

public static void main(String[] args) {

StudentDatabase db = new StudentDatabase();

Scanner scanner = new Scanner(System.in);

// Create sample data

List&lt;Student&gt; students = new ArrayList&lt;&gt;();

students.add(new Student(&quot;1&quot;, &quot;Alice&quot;, &quot;101&quot;, &quot;10th&quot;, 95.5, &quot;123 Main St&quot;));

students.add(new Student(&quot;2&quot;, &quot;Bob&quot;, &quot;102&quot;, &quot;10th&quot;, 88.0, &quot;456 Oak St&quot;));

students.add(new Student(&quot;3&quot;, &quot;Charlie&quot;, &quot;103&quot;, &quot;10th&quot;, 92.5, &quot;789 Pine St&quot;));

// Create database

db.createDatabase(students);

// Display database

System.out.println(&quot;Displaying Database:&quot;);

List&lt;Student&gt; studentList = db.displayDatabase();

studentList.forEach(System.out::println);

// Update a record

System.out.println(&quot;\nUpdating Bob&#39;s record...&quot;);

db.updateRecord(new Student(&quot;2&quot;, &quot;Bob&quot;, &quot;102&quot;, &quot;10th&quot;, 90.0, &quot;456 Oak St

Updated&quot;));

// Search for a record

System.out.println(&quot;\nSearching for Alice&#39;s record...&quot;);

Student searchResult = db.searchRecord(&quot;1&quot;);

if (searchResult != null) {

System.out.println(&quot;Found: &quot; + searchResult);

} else {

System.out.println(&quot;Record not found.&quot;);

}

// Delete a record

System.out.println(&quot;\nDeleting Charlie&#39;s record...&quot;);

db.deleteRecord(&quot;3&quot;);

// Display database again

System.out.println(&quot;Displaying Database after deletion:&quot;);

studentList = db.displayDatabase();

studentList.forEach(System.out::println);

scanner.close();

}

}