VIT-AP UNIVERSITY

Name	:	Siddireddy Pramod Reddy
Reg No.	:	23BCE8565
Course Code	:	CSE2005
Slot	:	L31+L32+L43+L44
Faculty	:	Prof. Sibi Chakkaravarthy S

Student Management System – Java Project

Overview

The Student Management System is a console-based Java application that provides CRUD (Create, Read, Update, Delete) operations for managing student records. The data is persistently stored in a file using CSV format.

Features

- - Add Students: Register students with ID, name, and age
- - View Students: Display all registered students
- - Search by ID: Find students using their ID
- - Delete Students: Remove students from records
- - Persistent Storage: Saves data to a file (students.txt)

System Design

Classes

Student

Fields: id, name, age

Methods: toString(), toCSV(), fromCSV(String)

StudentManagementSystem

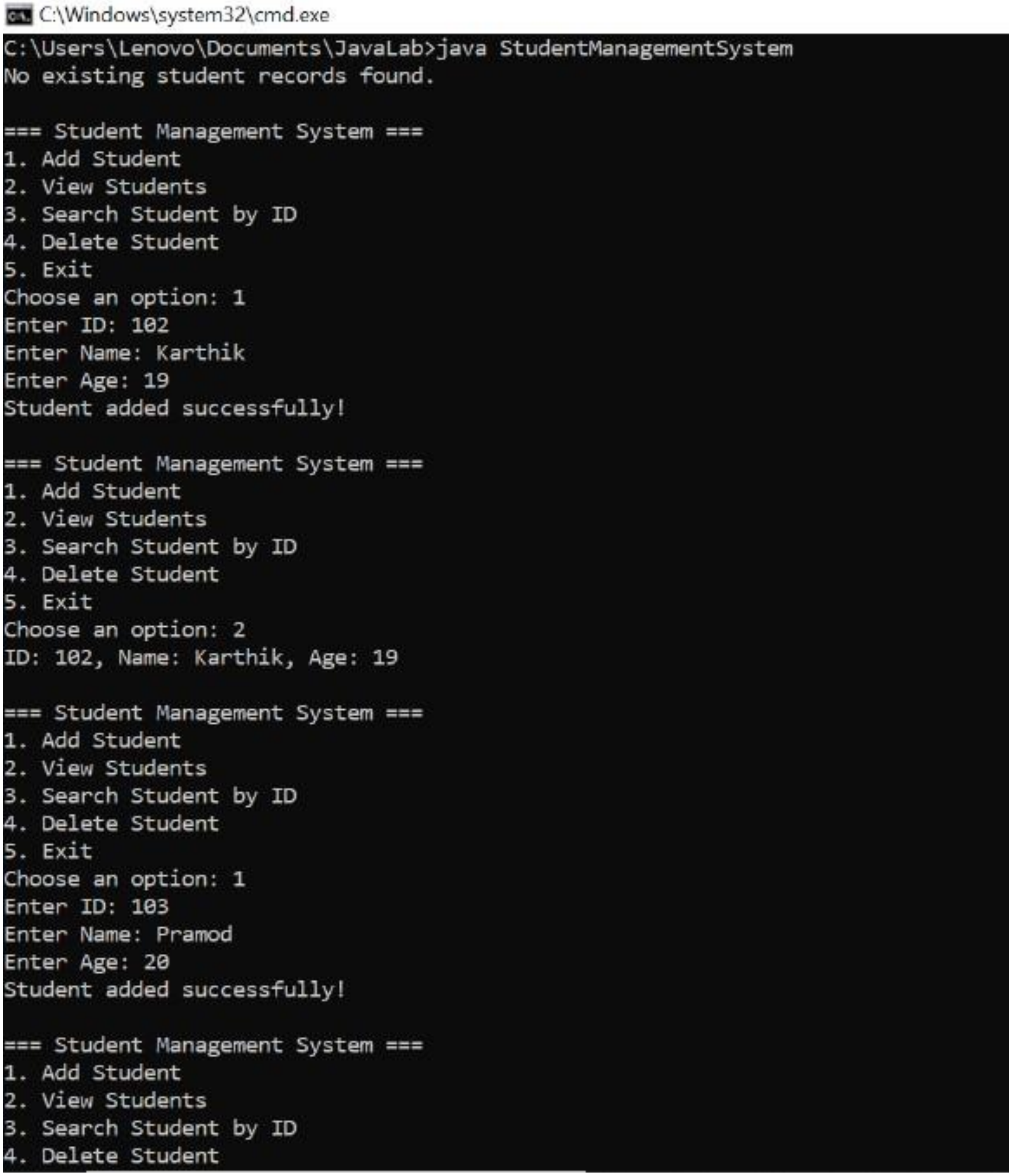
Handles user interface, logic, file I/O, and student management operations

Code

```java
import java.io.*;
import java.util.*;

class Student {
    int id;
    String name;
    int age;

    Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Age: " + age;
    }

    public String toCSV() {
        return id + "," + name + "," + age;
    }

    public static Student fromCSV(String line) {
        String[] parts = line.split(",");
        return new Student(Integer.parseInt(parts[0]), parts[1],
Integer.parseInt(parts[2]));
    }
}

public class StudentManagementSystem {
    static Scanner scanner = new Scanner(System.in);
    static List<Student> students = new ArrayList<>();
    static final String FILE_NAME = "students.txt";

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

        while (true) {
            System.out.println("\n=== Student Management System ===");
            System.out.println("1. Add Student");
            System.out.println("2. View Students");
            System.out.println("3. Search Student by ID");
            System.out.println("4. Delete Student");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();
            switch (choice) {
                case 1 -> addStudent();
                case 2 -> viewStudents();
                case 3 -> searchStudent();
                case 4 -> deleteStudent();
                case 5 -> {
                    System.out.println("Exiting...");
                    saveStudentsToFile();
                    return;
                }
                default -> System.out.println("Invalid choice!");
            }
        }
    }
```

```
static void loadStudentsFromFile() {
    try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
        String line;
        while ((line = reader.readLine()) != null) {
            students.add(Student.fromCSV(line));
        }
    } catch (IOException e) {
        System.out.println("No existing student records found.");
    }
}

static void saveStudentsToFile() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME))) {
        for (Student s : students) {
            writer.write(s.toCSV());
            writer.newLine();
        }
    } catch (IOException e) {
        System.out.println("Error saving student data.");
    }
}

static void addStudent() {
    System.out.print("Enter ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Age: ");
    int age = scanner.nextInt();

    students.add(new Student(id, name, age));
    saveStudentsToFile();
    System.out.println("Student added successfully!");
}

static void viewStudents() {
    if (students.isEmpty()) {
        System.out.println("No students to display.");
    } else {
        for (Student s : students) {
                System.out.println(s);
            }
        }
    }

    static void searchStudent() {
        System.out.print("Enter ID to search: ");
        int id = scanner.nextInt();
        for (Student s : students) {
            if (s.id == id) {
                System.out.println(s);
                return;
            }
        }
        System.out.println("Student not found.");
    }

    static void deleteStudent() {
        System.out.print("Enter ID to delete: ");
        int id = scanner.nextInt();
        boolean removed = students.removeIf(s -> s.id == id);
        if (removed) {
            saveStudentsToFile();
            System.out.println("Student deleted successfully.");
        } else {
            System.out.println("Student not found.");
        }
    }
}
```

Output:



```
C:\Users\Lenovo\Documents\JavaLab>java StudentManagementSystem
No existing student records found.

=== Student Management System ===
1. Add Student
2. View Students
3. Search Student by ID
4. Delete Student
5. Exit
Choose an option: 1
Enter ID: 102
Enter Name: Karthik
Enter Age: 19
Student added successfully!

=== Student Management System ===
1. Add Student
2. View Students
3. Search Student by ID
4. Delete Student
5. Exit
Choose an option: 2
ID: 102, Name: Karthik, Age: 19

=== Student Management System ===
1. Add Student
2. View Students
3. Search Student by ID
4. Delete Student
5. Exit
Choose an option: 1
Enter ID: 103
Enter Name: Pramod
Enter Age: 20
Student added successfully!

=== Student Management System ===
1. Add Student
2. View Students
3. Search Student by ID
4. Delete Student
```