Student Management System – Full Project Manual

1. Introduction

The Student Management System is a Java-based console application designed to help beginners understand core Java concepts, including Object-Oriented Programming (OOP), File Handling, Collections, and modular programming. It supports adding, updating, deleting, searching, and listing student records with persistent storage using CSV files.

2. Project Objectives

- Understand practical implementation of OOP concepts.
- Build a functional CRUD-based console application.
- Learn data persistence using CSV files.
- Handle user input and validation.
- Apply core Java concepts in real-world logic.

3. System Requirements

Software:

- Java JDK 8 or above
- Code Editor (VS Code, IntelliJ, Eclipse, Notepad++)

Hardware:

- Any computer capable of running Java programs.

4. Technologies & Concepts Used

4.1 Java Concepts

- Data Types
- Variables
- Operators
- Control Statements (if, switch, loops)
- Methods
- Exception Handling

- Classes and Objects
- Constructors
- Encapsulation
- ArrayList
- Scanner Input
- File I/O (BufferedReader, PrintWriter)

4.2 Technologies

- Java SE
- CSV File Storage
- ReportLab (PDF generation for documentation)

5. Functional Requirements

Must Support:

1. Add Student
2. Update Student
3. Delete Student
4. Search by Name
5. List All Students
6. Save Data to CSV
7. Load Data from CSV Automatically

6. Non-Functional Requirements

- Easy to use
- Error-free execution
- Handles invalid inputs
- Data should persist between program sessions
- Modular code structure

7. System Flow (Step-by-Step)

1. Application starts
2. CSV data loads

3. Menu displayed
4. User enters a choice
5. Program executes selected operation
6. Returns to menu
7. Until user exits

8. Program Flowchart

Start → Menu → Input → Add/Update/Delete/Search/List → Save CSV → Loop → Exit

9. Module Description

9.1 Student Module

Stores:

- ID
- Name
- Age
- Course

9.2 Data Handling Module

Responsible for:

- Reading CSV
- Writing CSV
- Managing ArrayList

9.3 UI Module

Displays menu and handles user interaction.

10. CSV File Format

id,	name,	age,	course
1,	John Doe,	19,	Computer Science
2,	Anita Rao,	20,	IT

11. Running the Program

Compile:

```
javac StudentManagement.java
```

Run:

```
java StudentManagement
```

CSV file generates automatically.

12. Interview Questions

Java Basics

1. What is OOP?
2. Explain Encapsulation.
3. What is a Class? Object?
4. Difference between Array and ArrayList?
5. What is a Constructor?

File Handling

1. How do you read files in Java?
2. Difference between FileReader and BufferedReader?
3. What is a CSV file?

Project Related

1. Explain your Student Management project.
2. How do you implement search functionality?
3. How do you persist data?

13. Future Enhancements

- Multiple file structure
- GUI using Swing / JavaFX
- Database integration using JDBC
- Web version using Spring Boot
- Sorting & Pagination

14. Complete Project Description (For Resume)

Designed and developed a Java-based Student Management System implementing CRUD operations.

The system uses Object-Oriented Programming principles and persists data in CSV format using Java

File I/O. Implemented menu-driven interface, input validation, search functionality, and modular code

structure. Demonstrates knowledge of Java basics, collections, error handling, and file management.

15. Conclusion

This project is ideal for beginners transitioning from Java basics to practical application development. It

builds strong fundamentals for advanced technologies such as JDBC, Servlets, Spring Boot, and

full-stack development.