

MEMBER 1 – STUDENT MANAGEMENT (JAVA DEVELOPMENT)

Name: E.N.RACHANA

Overall Role

Member 1 is **responsible for the core student data of the entire system**. Every other module (attendance, marks, results, backlogs) **depends on the student information created by this member**.

If student data is wrong → the whole system breaks.
So this member builds a **strong, reliable Java foundation**.

1. Create Student-Related Java Classes

This member designs the **basic Java structure** for students.

What they code:

- Student class
 - Fields:
 - Roll Number
 - Name
 - Branch
 - Year
 - Section
 - Status (Active / Inactive)
- StudentService class
 - Contains all **logic** related to student operations

Purpose:

To represent each student as a **Java object** that can be stored, updated, and shared with other modules.

2. Student Registration Logic

This member writes Java code to **add new students into the system**.

Responsibilities:

- Accept student details from input (UI or console)
- Create a new Student object
- Save the student data (in database or in-memory collection)

Outcome:

Every student gets **officially registered** in the system.

3. Store Student Academic Information

This member ensures the system correctly stores:

- Roll Number (unique identifier)
- Name
- Branch (CSE, ECE, etc.)
- Year (1st, 2nd, 3rd, 4th)
- Section (A, B, C)

Why this matters:

- Attendance uses roll number
- Marks depend on year and section
- Results depend on correct student mapping

4. Update Student Academic Status

Students don't stay the same forever.

This member handles:

- Promoting students to the next year
- Marking students as:
 - Active
 - Inactive (left college, detained, graduated)
- Updating section or branch if needed

Example:

- Year 2 → Year 3 promotion

- Student marked “Inactive” after graduation

5. Validate Student Data (Very Important)

This member **prevents wrong data from entering the system.**

Validation rules they implement:

- Roll number must be **unique**
- Name cannot be empty
- Branch, year, and section are mandatory
- Invalid year or section should be rejected

Purpose:

- Avoid duplicate students
- Maintain clean and reliable data

6. Provide Student Data to Other Modules

This member makes student data **accessible to other team members.**

What they provide:

- Methods to fetch student details using roll number
- Student lists for:
 - Attendance module
 - Marks module
 - Result module

Example:

- Attendance module asks:
“Give me student details for roll number 21CSE045”
- Student Management module responds with correct data