

All the related diagram of the project

1. ER Diagram (Entity Relationship Diagram)

Entities and their relationships in the database.

Entities:

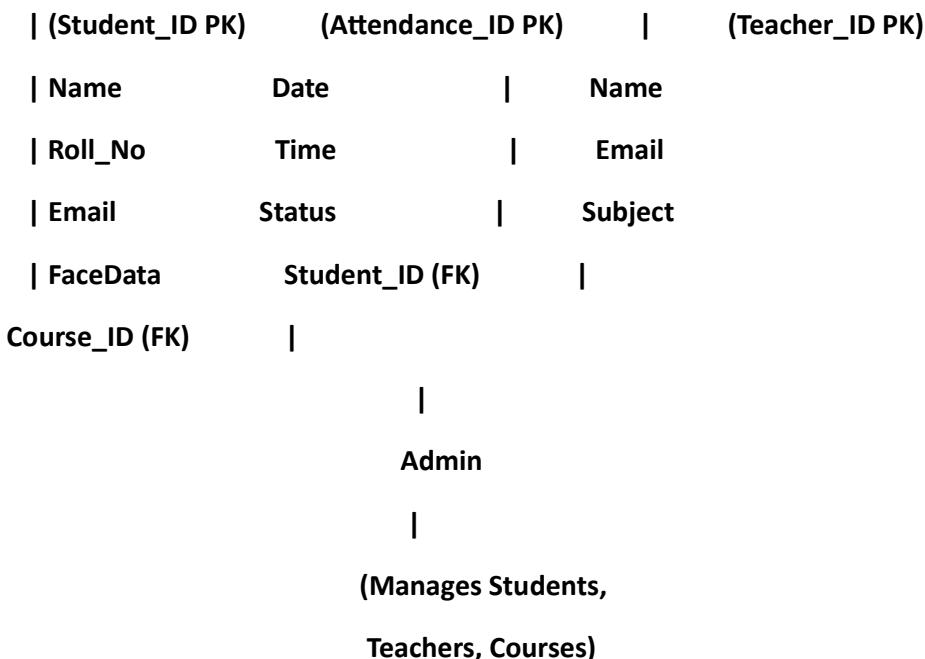
- Student (Student_ID, Name, Roll_No, Email, FaceData)
- Teacher (Teacher_ID, Name, Email, Subject)
- Admin (Admin_ID, Name, Email, Role)
- Attendance (Attendance_ID, Date, Time, Status, Student_ID, Course_ID)
- Course (Course_ID, Course_Name, Teacher_ID)

Relationships:

- A **Student** is enrolled in one or many **Courses**.
- A **Course** is taught by one **Teacher**.
- **Attendance** is recorded for each **Student** in a **Course**.
- **Admin** manages Students, Teachers, and Attendance.

📌 ER Diagram (Text Sketch):

Student -----< Attendance >----- Course >---- Teacher



2. Use Case Diagram

Actors: Admin, Teacher, Student, Camera System (AI model)

Use Cases:

- Admin → Register Student, Manage Database, Generate Reports
- Teacher → View Attendance, Validate Attendance
- Student → View Attendance Status
- Camera System → Detect & Recognize Faces, Mark Attendance

📌 Use Case (Text Sketch):

[Admin] ----- (Register Student)

| ----- (Generate Reports)

| ----- (Manage Database)

[Teacher] ----- (View Attendance)

| ----- (Validate Attendance)

[Student] ----- (View Attendance Status)

[Camera System] ----- (Detect Face) ----> (Mark Attendance in DB)

3. Data Flow Diagram (DFD)

Level 0 (Context Level):

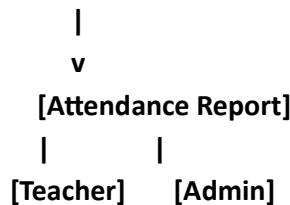
- Camera → Face Recognition System → Database → Reports

Level 1:

- Student Face Data captured → AI Model → Attendance Marked → Stored in DB
- Teacher retrieves report from DB
- Admin manages student/course data

📌 DFD (Text Sketch):

Student -----> [Camera Input] -----> [AI Face Recognition System] -----> (Attendance DB)



4. File Structure Diagram

```
SmartAttendanceSystem/
|
|   |-- backend/
|   |   |-- src/
|   |   |   |-- controllers/      # Java classes for handling HTTP requests
|   |   |   |-- models/          # Java POJOs representing DB entities
|   |   |   |-- services/        # Business logic and face recognition service
|   |   |   |-- repositories/    # Database access logic
|   |   |   |-- utils/           # Utilities such as image processing
|   |   |-- resources/
|   |   |   |-- application.properties # DB config, server properties
|   |   |-- pom.xml or build.gradle # Build configuration
|
|   |-- frontend/
|   |   |-- css/
|   |   |-- js/
|   |   |-- images/
|   |   |-- index.html         # Student login page
|   |   |-- faculty.html       # Faculty dashboard
|   |   |-- attendance.html    # Attendance marking page
|
|   |-- database/
|   |   |-- schema.sql         # SQL script to create DB schema
|
|   |-- documentation/
|   |   |-- project-report.docx  # Full documentation report
|   |   |-- module-diagrams.png  # Architecture and module diagrams
|   |   |-- user-manual.pdf     # Instructions for users and faculty
```

5. Sequence Diagram

-  Shows the step-by-step flow of attendance marking.

Actor(Student) ---> Camera ---> AI Face Recognition ---> Database ---> Teacher/Admin

- Student : Appears before camera
- Camera : Captures face
- AI Model : Detects & verifies student
- Database : Records attendance (Date, Time, Status)
- Teacher/Admin : Fetches attendance repository.