Entity-Relationship Diagram (ERD) and **Data Flow Diagrams (DFD Level 0 and 1)**:

---

1. Entity-Relationship Diagram (ERD)

Entities:

1. **User**:
   - `id` (Primary Key)
   - `email` (Unique)
   - `password`
2. **Profile**:
   - `id` (Primary Key)
   - `fname` (First Name)
   - `lname` (Last Name)
   - `gender`
   - `dob` (Date of Birth)
   - `height`
   - `weight`
   - `bloodGroup`
   - `parentId` (Foreign Key referencing `User.id`)
3. **XrayImage**:
   - `id` (Primary Key)
   - `filePath` (Path to saved image)
   - `uploadedAt` (Timestamp)
   - `profileId` (Foreign Key referencing `Profile.id`)
4. **Report**:
   - `id` (Primary Key)
   - `fractureDetails` (Details of the analysis)
   - `fracturePercentage` (Severity percentage)
   - `createdAt` (Timestamp)
   - `xrayImageId` (Foreign Key referencing `XrayImage.id`)
   - `profileId` (Foreign Key referencing `Profile.id`)
5. **ChatHistory**:
   - `id` (Primary Key)
   - `message` (User's or chatbot's message)
   - `timestamp` (Time of message)
   - `profileId` (Foreign Key referencing `Profile.id`)

Relationships:

1. **User ↔ Profile**:
   - A **User** can have many **Profiles** (1:N).
   - Each **Profile** belongs to a single **User**.
2. **Profile ↔ XrayImage**:
   - A **Profile** can have many **XrayImage** entries (1:N).
   - Each **XrayImage** is associated with a single **Profile**.
3. **Profile ↔ Report**:
   - A **Profile** can have many **Reports** (1:N).
   - Each **Report** belongs to a single **Profile**.
4. **XrayImage ↔ Report**:

    - Each **XrayImage** is associated with exactly one **Report** (1:1).
    - Each **Report** references a single **XrayImage**.
5. **Profile ↔ ChatHistory**:
    - A **Profile** can have many **ChatHistory** entries (1:N).
    - Each **ChatHistory** belongs to a single **Profile**.

Diagram Layout:

- **User** at the top, connected to multiple **Profiles**.
- **Profile** in the center, connecting:
    - To multiple **XrayImage** entries.
    - To multiple **Reports**.
    - To multiple **ChatHistory** entries.
- **XrayImage** connected to a single **Report**.

Would you like a **visual representation** of the ERD, or are these details sufficient for now?

DFD Level 0 (Context Diagram)

This represents the high-level overview of your system.

Processes:

- **AI-Based X-Ray Analysis App**:
    - Accepts X-ray image uploads from users.
    - Performs image analysis to generate reports.
    - Manages user profiles and links uploaded images to reports.
    - Provides chat assistance and report sharing functionality.

External Entities:

1. **User**:
    - Inputs: X-ray image, credentials, profile data, and chat inquiries.
    - Outputs: Reports, chat responses, and profile management options.
2. **Medical Specialist**:
    - Inputs: Shared reports for professional consultation.
    - Outputs: Feedback to users (optional).

Data Stores:

1. **User Database**:
    - Stores user credentials and profile details.
2. **Xray Image Database**:
    - Stores uploaded X-ray images and their metadata.
3. **Report Database**:
    - Stores generated reports and links them to uploaded images.
4. **Chat Database**:
    - Stores chat histories.

Diagram Components:

- User → [AI-Based X-Ray Analysis App] → Report Database
- Medical Specialist → [AI-Based X-Ray Analysis App] → Shared Reports
- User ↔ User Database (Login/Registration, Profile Data)
- User ↔ Chat Database (Chat Assistance)
- User ↔ Xray Image Database (Image Uploads)

DFD Level 1

This expands on the key processes in Level 0, showing the detailed workflows.

Processes:

1. **User Authentication and Profile Management**:
    - Input: User credentials or profile data.
    - Output: Authentication success/failure or updated profile details.
    - Data Store: User Database.
2. **X-ray Image Upload and Storage**:
    - Input: X-ray image file from the user.
    - Output: Stored image and confirmation message.
    - Data Store: Xray Image Database.
3. **Fracture Detection and Report Generation**:
    - Input: X-ray image ID (from storage).
    - Output: Generated report.
    - Data Store: Report Database.
4. **Chat Assistance**:
    - Input: User concerns and queries.
    - Output: Chat responses and linked report references.
    - Data Store: Chat Database.
5. **Report Sharing**:
    - Input: Selected report and recipient details.
    - Output: Shared report with the medical specialist.
    - External Entity: Medical Specialist.

Data Flows in Level 1:

1. **User → App**:
    - Credentials, profile data, X-ray images, chat queries.
2. **App → Databases**:
    - Stores user data, uploaded images, chat history, and generated reports.
3. **App → User**:
    - Provides access to uploaded images, reports, chat responses, and profile details.
4. **App → Medical Specialist**:
    - Sends shared reports for consultation.
