**Architecture Document**

**1. Application Architecture**

For our project, we propose a **Microservices Architecture** due to its scalability, flexibility, and ability to support multiple independent services while ensuring smooth integration. Each feature of the platform, such as the chatbot, meditation timer, mood tracker, and planner, can be developed as independent microservices, allowing for easier updates and maintenance.

**1.1 Justification for Microservices Architecture:**

* **Scalability:** Each service can scale independently based on demand.
* **Flexibility:** Allows different teams to work on separate modules without affecting the whole system.
* **Reliability:** Failure in one microservice does not impact the entire application.
* **Technology Diversity:** Enables using different programming languages or databases for different services.

### 1.2 Component Breakdown

Our proposed system integrates multiple components to enhance the efficiency and accuracy of Multiple Sclerosis (MS) detection using deep learning techniques.

* **Lesion Segmentation Module (U-Net)** – This module performs precise segmentation of MS lesions from MRI scans, leveraging an encoder-decoder architecture with skip connections for high spatial accuracy.
* **Classification Engine (EfficientNet)** – A deep learning model optimized for classifying MRI images into Normal, MS Axial, and MS Sagittal categories, ensuring high accuracy with reduced computational complexity.
* **Preprocessing & Augmentation Service** – Implements normalization, resizing, contrast adjustments, and augmentation techniques (rotation, shear, zoom) to enhance model robustness.
* **User Interface & Report Generation** – A web-based platform that allows clinicians to upload MRI scans and receive automated diagnostic reports, assisting in clinical decision-making.
* **Attention Mechanism & Feature Extraction** – Integrates attention-based models to focus on lesion regions, improving classification performance and reducing false positives.
* **Model Optimization & Transfer Learning** – Utilizes pre-trained deep learning models and multi-center datasets to improve generalization and scalability in clinical applications.
* **Performance Evaluation & Validation** – Implements precision, recall, F1-score, and confusion matrix analysis to assess model effectiveness in classification and segmentation tasks.

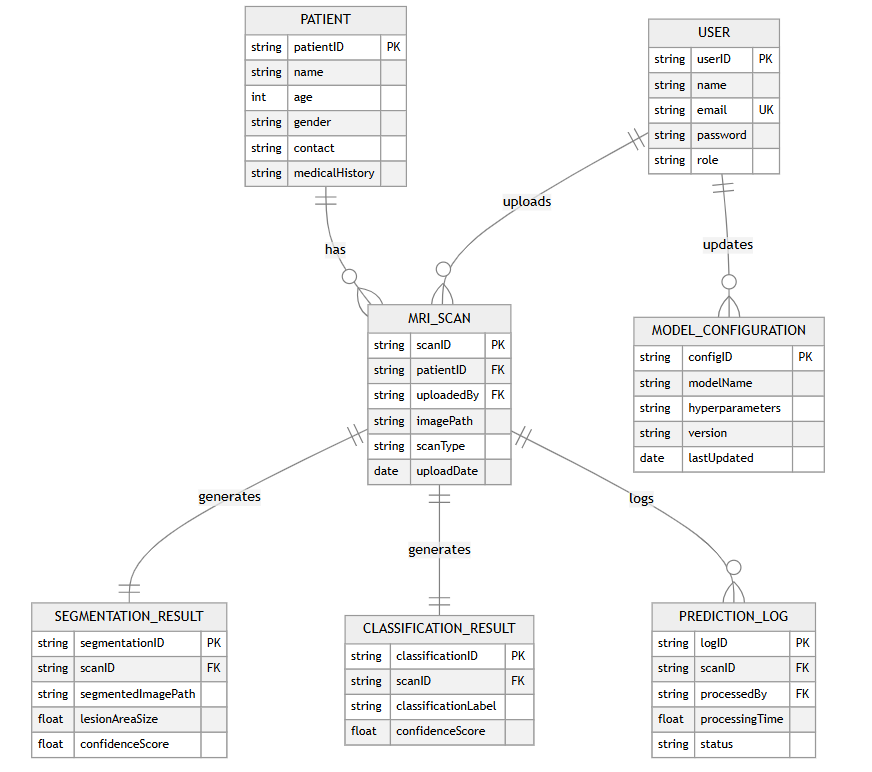
### 2. Database Architecture

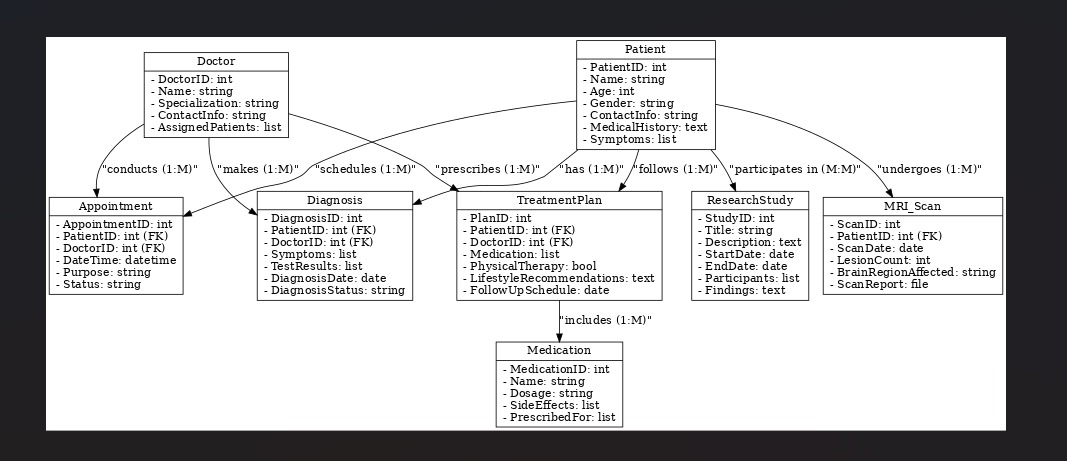
Our system adopts a **hybrid database architecture** to efficiently store and manage structured and unstructured data for MS detection.

* **Relational Database (SQL-based)** – Stores structured medical records, MRI scan metadata, segmentation results, and classification outputs.
* **NoSQL Database (Document-based storage)** – Handles unstructured data such as lesion segmentation masks, chatbot interactions (for AI-assisted diagnosis explanations), and user history tracking.
* **Scalable Cloud Storage** – Ensures efficient handling of large MRI datasets with high availability and security.
* **Real-time Processing Pipeline** – Enables seamless integration between segmentation, classification, and clinical report generation, optimizing diagnostic workflows.

This hybrid approach ensures **high efficiency, scalability, and reliability**, making it suitable for AI-assisted medical imaging application.

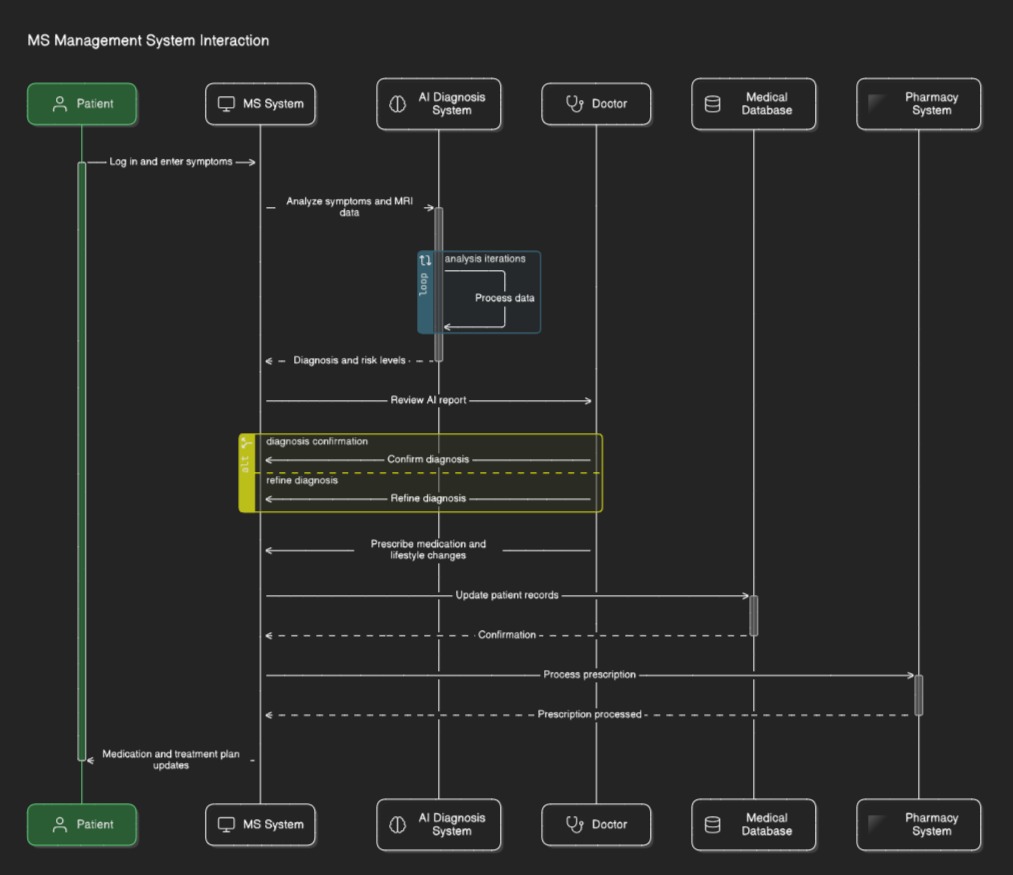
**2.1 ER Diagram**

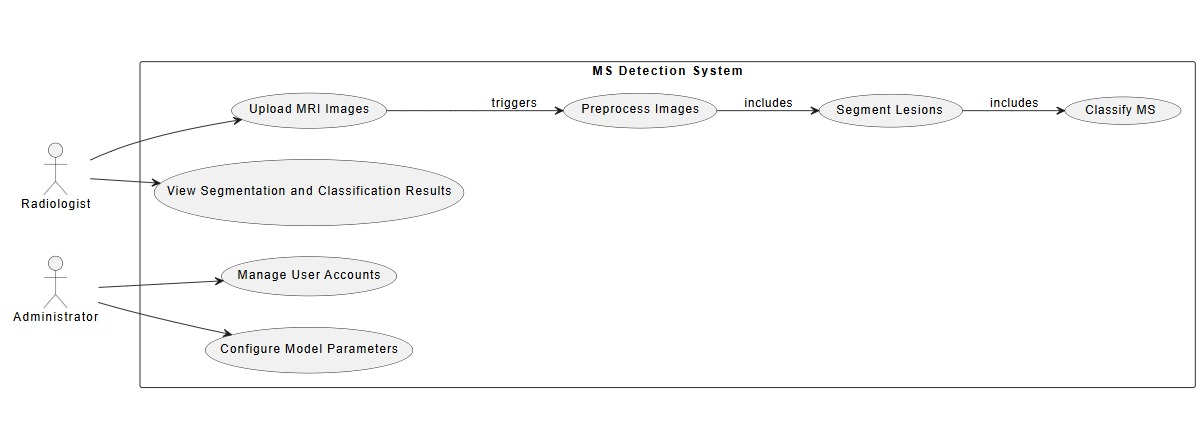
****

**2.2 Class Diagram :**

**2.3 Component Diagram :**

**2.4 Sequence Diagram :**

****

**2.5 Use Case Diagram :   
**

**2.5 Deployment Diagram :**

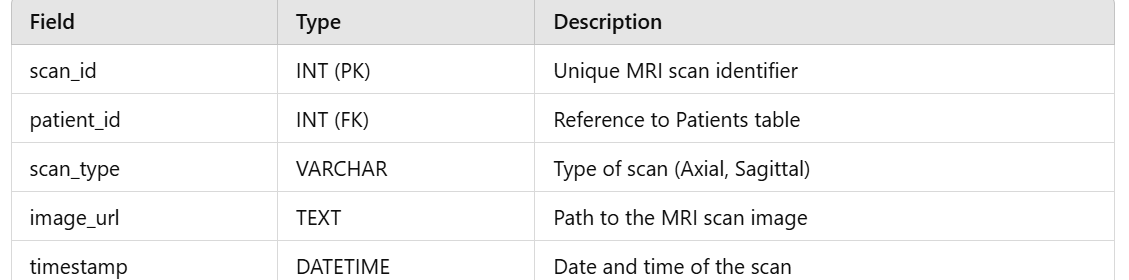
## 2.6 Schema Design

Our system employs a structured database schema to efficiently store, retrieve, and manage patient data related to Multiple Sclerosis (MS) detection. The schema is designed to handle MRI scan metadata, segmentation results, classification outputs, and user interactions with the system.

### 2.6.1 Patients Table (Stores Patient Information)

### 

### 2.6.2 MRI Scans Table (Stores MRI Scan Data for Each Patient)



### 2.6.3 Lesion Segmentation Table (Stores U-Net Segmentation Results)

### 

### 

### 

### 

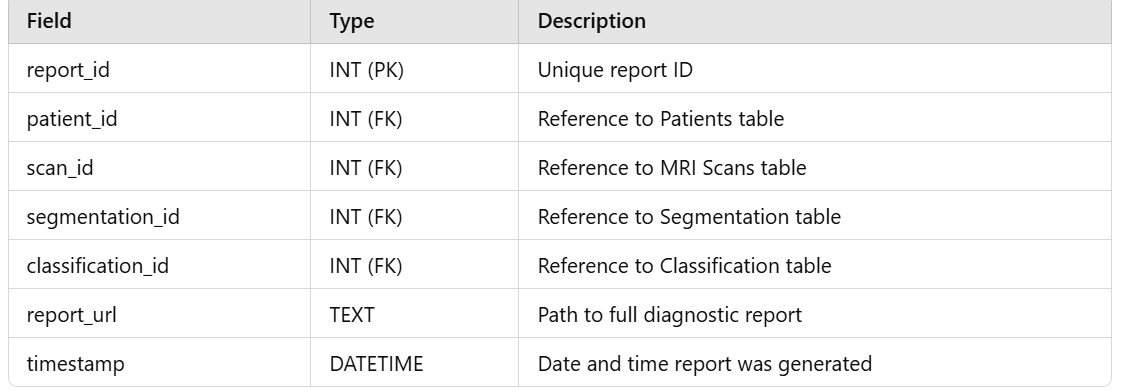
### 2.6.4 Classification Results Table (EfficientNet Classification Output)2.6.5

### 

### 

### 

### 2.6.5 Reports Table (Stores Patient Diagnosis Reports)



## 3. Data Exchange Contract

### 3.1 Frequency of Data Exchanges

This section defines how often different data transactions occur within the system.

#### 1️ Real-Time Updates (Instant Data Exchange)

* **MRI Scan Upload & Processing:**
  + As soon as an MRI scan is uploaded, the system processes it in real time.
  + U-Net generates lesion segmentation masks, and EfficientNet classifies the scan automatically.
  + Results are stored in the database and immediately displayed to the clinician.

#### 2️ Daily/Weekly Updates (Scheduled Data Exchange)

* **Performance Monitoring:**
  + The system periodically evaluates classification accuracy and segmentation performance.
  + Updates are logged for further fine-tuning of the deep learning models.
* **Diagnostic Reports:**
  + Daily reports summarize detected lesions, classification outcomes, and changes in scan results over time.

#### 3️ On-Demand Data Retrieval (As Requested by the Clinician/Patient)

* **MRI Reports:**
  + Doctors can access historical scan reports and compare lesion progression across different timeframes.
* **AI-Assisted Diagnosis:**
  + On request, AI can provide additional insights on MRI scans, including heatmaps and uncertainty scores.

### 3.2 Data Sets

#### 1️ Patient Data (Stored Medical Profiles)

* **Profile Details:** Patient name, email, age, gender, medical history.
* **Scan History:** List of previous MRI scans and corresponding diagnoses.

#### 2️ MRI Scan Data (Stored for Analysis & AI Processing)

* **MRI Images:** Raw scans stored in a secure cloud storage system.
* **Lesion Segmentation Masks:** U-Net-generated masks to highlight affected regions.
* **Classification Labels:** AI-determined MS detection results with confidence scores.

#### 3️ Diagnostic Reports (AI-Generated Clinical Summaries)

* **Summary Reports:** Generated PDF/HTML reports accessible by clinicians.
* **Comparison Data:** Longitudinal tracking of lesion growth over multiple scans.

### 3.3 Mode of Exchanges (How Data is Transferred)

#### 1️RESTful APIs (Frontend-Backend Communication)

APIs enable seamless communication between the web application and the backend system.

##### Example API Endpoints:

* **GET /patient/{id}** → Fetch patient profile & history
* **POST /scan/upload** → Upload MRI scan for processing
* **GET /report/{scan\_id}** → Retrieve AI-generated diagnostic report

#### 2️ WebSockets (Real-Time Scan Processing)

* Used for **real-time lesion segmentation and classification** display.
* Clinicians receive instant updates on scan processing status.

#### 3️ Cloud Storage & Message Queues (Efficient Data Processing)

* **MRI Images & Reports** → Stored in a HIPAA-compliant cloud storage system.
* **AI Model Predictions** → Asynchronously processed and stored for immediate retrieval.
* **Automated Notifications** → Doctors receive alerts for new scan results.