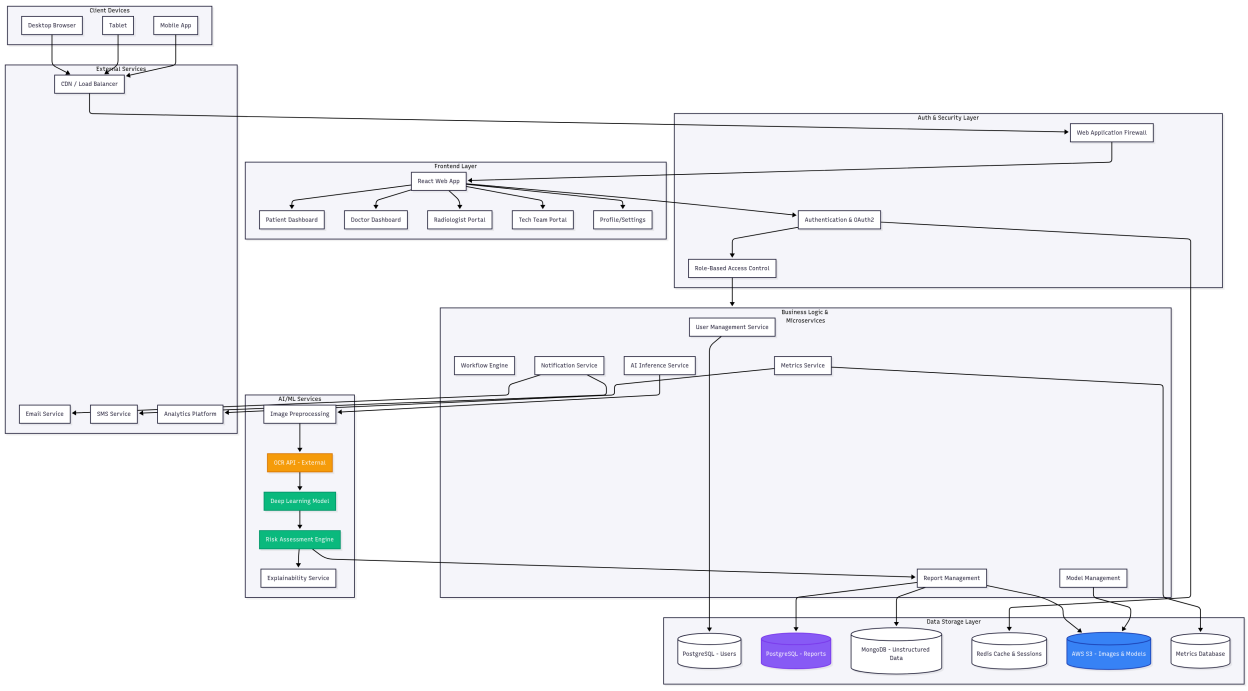
Updated Wireframes and First Draft of Backend

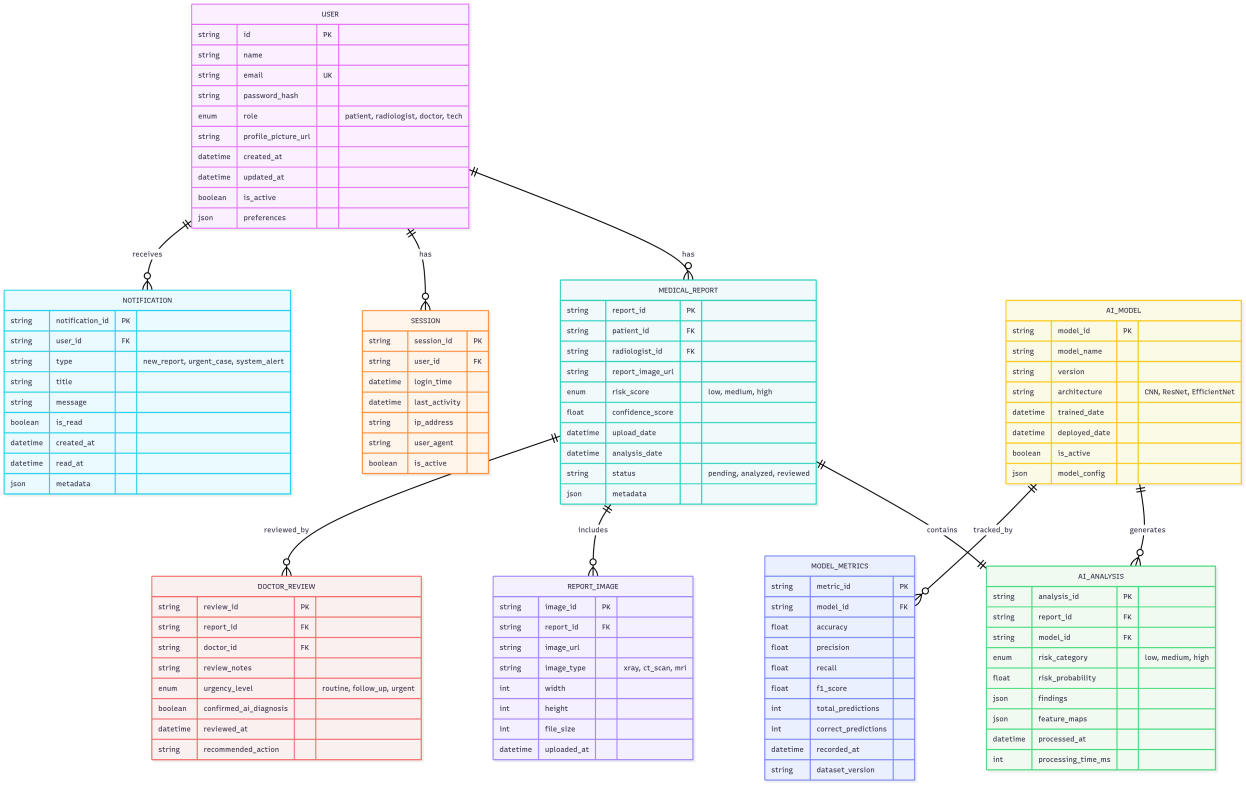
1.Updated Wireframes

Figma Link

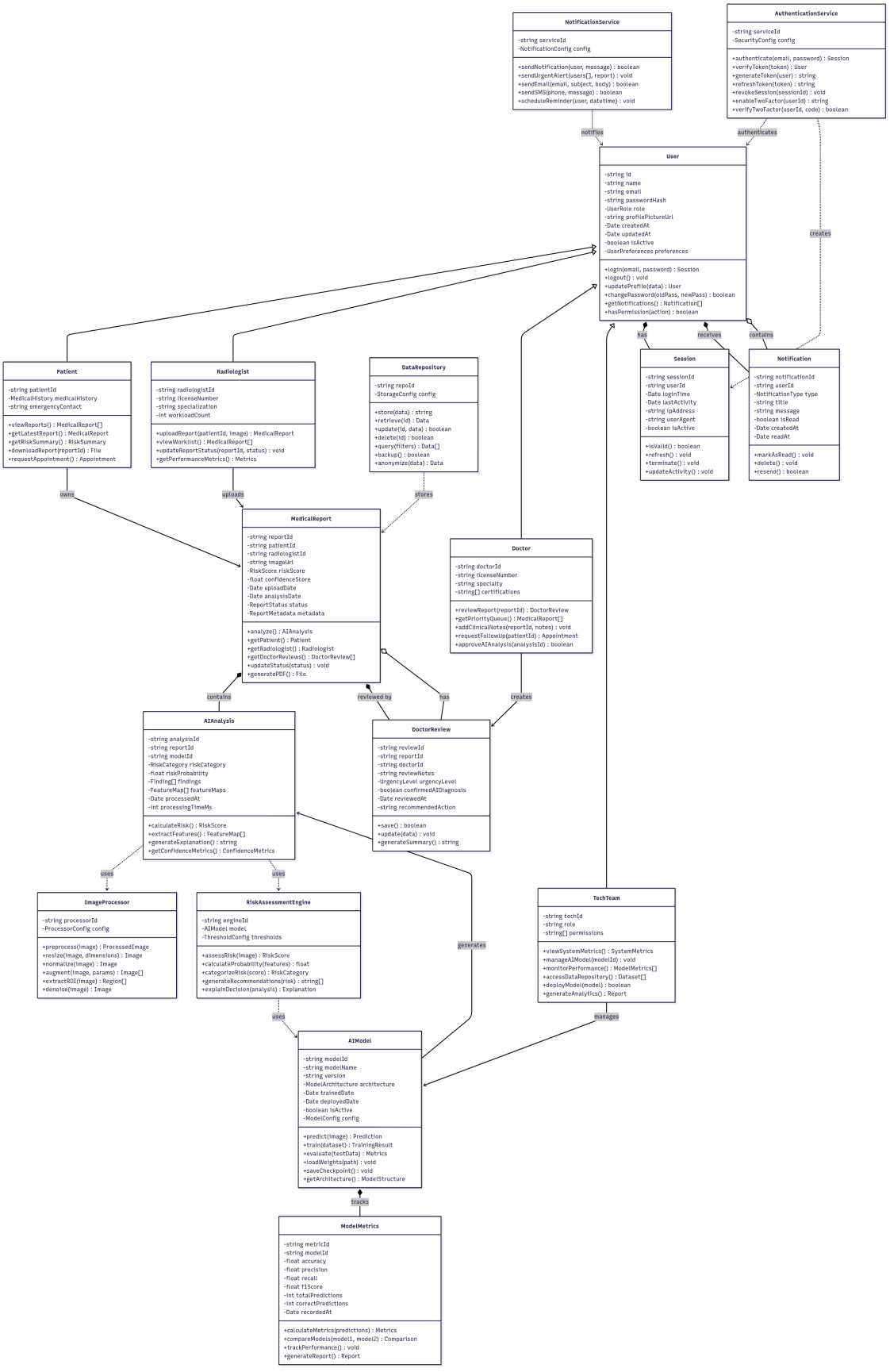
2.Overall Architecture Diagram



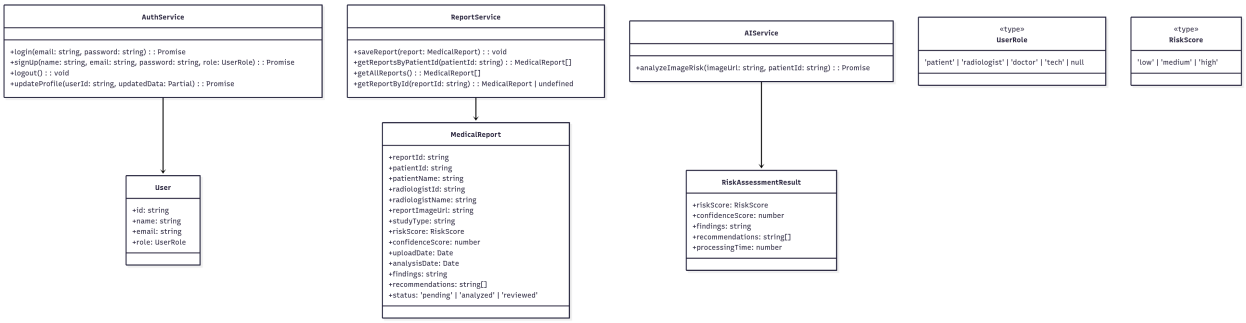
3.Entity-Relation Diagram



4.UML Diagram

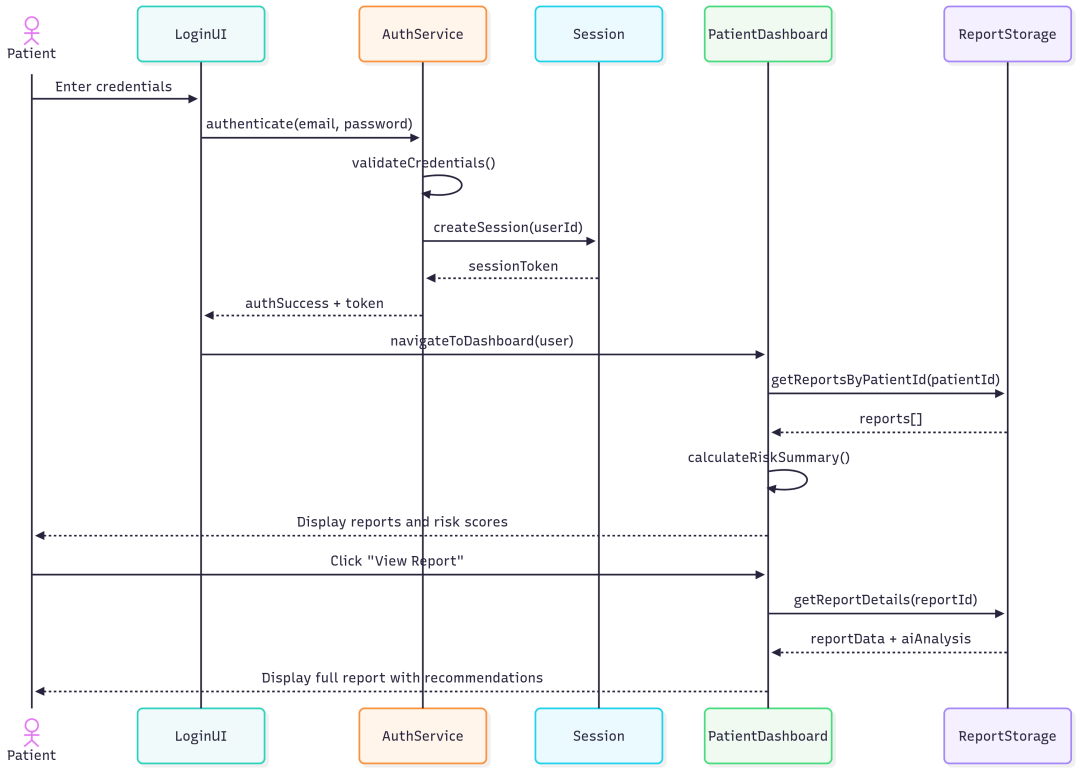

NotificationService
-string serviceId
-NotificationConfig config
+sendNotification(user, message) : boolean
+sendUrgentAlert(users[], report) : void
+sendEmail(email, subject, body) : boolean
+sendSMS(phone, message) : boolean
+scheduleReminder(user, datetime) : void
AuthenticationService
-string serviceId
-SecurityConfig config
+authenticate(email, password) : Session
+verifyToken(token) : User
+generateToken(user) : string
+refreshToken(token) : string
+revokeSession(sessionId) : void
+enableTwoFactor(userId) : string
+verifyTwoFactor(userId, code) : boolean
notifies
authenticates
User
-string id
-string name
-string email
-string passwordHash
-UserRole role
-string profilePictureUrl
-Date createdAt
-Date updatedAt
-boolean isActive
-UserPreferences preferences
+login(email, password) : Session
+logout() : void
+updateProfile(data) : User
+changePassword(oldPass, newPass) : boolean
+getNotifications() : Notification[]
+hasPermission(action) : boolean
creates
has
receives
contains
Session
-string sessionId
-string userId
-Date loginTime
-Date lastActivity
-string ipAddress
-string userAgent
-boolean isActive
+isValid() : boolean
+refresh() : void
+terminate() : void
+updateActivity() : void
Notification
-string notificationId
-string userId
-NotificationType type
-string title
-string message
-boolean isRead
-Date createdAt
-Date readAt
+markAsRead() : void
+delete() : void
+resend() : boolean
Patient
-string patientId
-MedicalHistory medicalHistory
-string emergencyContact
+viewReports() : MedicalReport[]
+getLatestReport() : MedicalReport
+getRiskSummary() : RiskSummary
+downloadReport(reportId) : File
+requestAppointment() : Appointment
Radiologist
-string radiologistId
-string licenseNumber
-string specialization
-int workloadCount
+uploadReport(patientId, image) : MedicalReport
+viewWorklist() : MedicalReport[]
+updateReportStatus(reportId, status) : void
+getPerformanceMetrics() : Metrics
DataRepository
-string repoId
-StorageConfig config
+store(data) : string
+retrieve(id) : Data
+update(id, data) : boolean
+delete(id) : boolean
+query(filters) : Data[]
+backup() : boolean
+anonymize(data) : Data
owns
uploads
stores
MedicalReport
-string reportId
-string patientId
-string radiologistId
-string imageUri
-RiskScore riskScore
-float confidenceScore
-Date uploadDate
-Date analysisDate
-ReportStatus status
-ReportMetadata metadata
+analyze() : AIAnalysis
+getPatient() : Patient
+getRadiologist() : Radiologist
+getDoctorReviews() : DoctorReview[]
+updateStatus(status) : void
+generatePDF() : File
Doctor
-string doctorId
-string licenseNumber
-string speciality
-string[] certifications
+reviewReport(reportId) : DoctorReview
+getPriorityQueue() : MedicalReport[]
+addClinicalNotes(reportId, notes) : void
+requestFollowUp(patientId) : Appointment
+approveAIAnalysis(analysisId) : boolean
contains
reviewed by
has
creates
AIAnalysis
-string analysisId
-string reportId
-string modelId
-RiskCategory riskCategory
-float riskProbability
-Finding[] findings
-FeatureMap[] featureMaps
-Date processedAt
-int processingTimeMs
+calculateRisk() : RiskScore
+extractFeatures() : FeatureMap[]
+generateExplanation() : string
+getConfidenceMetrics() : ConfidenceMetrics
DoctorReview
-string reviewId
-string reportId
-string doctorId
-string reviewNotes
-UrgencyLevel urgencyLevel
-boolean confirmedAIDiagnosis
-Date reviewedAt
-string recommendedAction
+save() : boolean
+update(data) : void
+generateSummary() : string
uses
uses
ImageProcessor
-string processorId
-ProcessorConfig config
+preprocess(image) : ProcessedImage
+resize(image, dimensions) : Image
+normalize(image) : Image
+augment(image, params) : Image[]
+extractROI(image) : Region[]
+denoise(image) : Image
RiskAssessmentEngine
-string engineId
-AIModel model
-ThresholdConfig thresholds
+assessRisk(image) : RiskScore
+calculateProbability(features) : float
+categorizeRisk(score) : RiskCategory
+generateRecommendations(risk) : string[]
+explainDecision(analysis) : Explanation
generates
TechTeam
-string techId
-string role
-string[] permissions
+viewSystemMetrics() : SystemMetrics
+manageAIModel(modelId) : void
+monitorPerformance() : ModelMetrics[]
+accessDataRepository() : Dataset[]
+deployModel(model) : boolean
+generateAnalytics() : Report
uses
manages
AIModel
-string modelId
-string modelName
-string version
-ModelArchitecture architecture
-Date trainedDate
-Date deployedDate
-boolean isActive
-ModelConfig config
+predict(image) : Prediction
+train(dataset) : TrainingResult
+evaluate(testData) : Metrics
+loadWeights(path) : void
+saveCheckpoint() : void
+getArchitecture() : ModelStructure
tracks
ModelMetrics
-string metricId
-string modelId
-float accuracy
-float precision
-float recall
-float f1Score
-int totalPredictions
-int correctPredictions
-Date recordedAt
+calculateMetrics(predictions) : Metrics
+compareModels(model1, model2) : Comparison
+trackPerformance() : void
+generateReport() : Report

5.API Signature Diagram

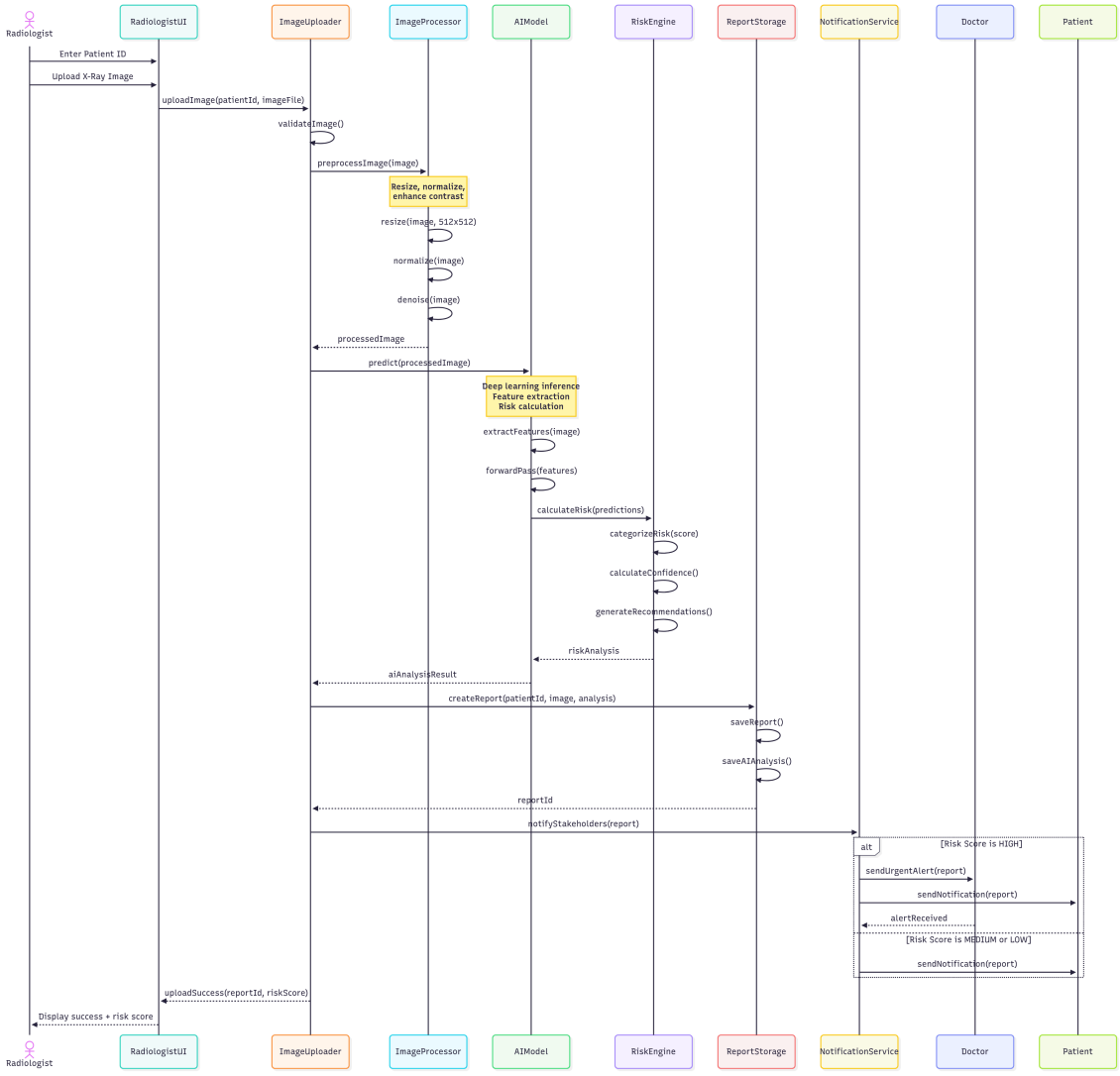


6.Sequence Diagram

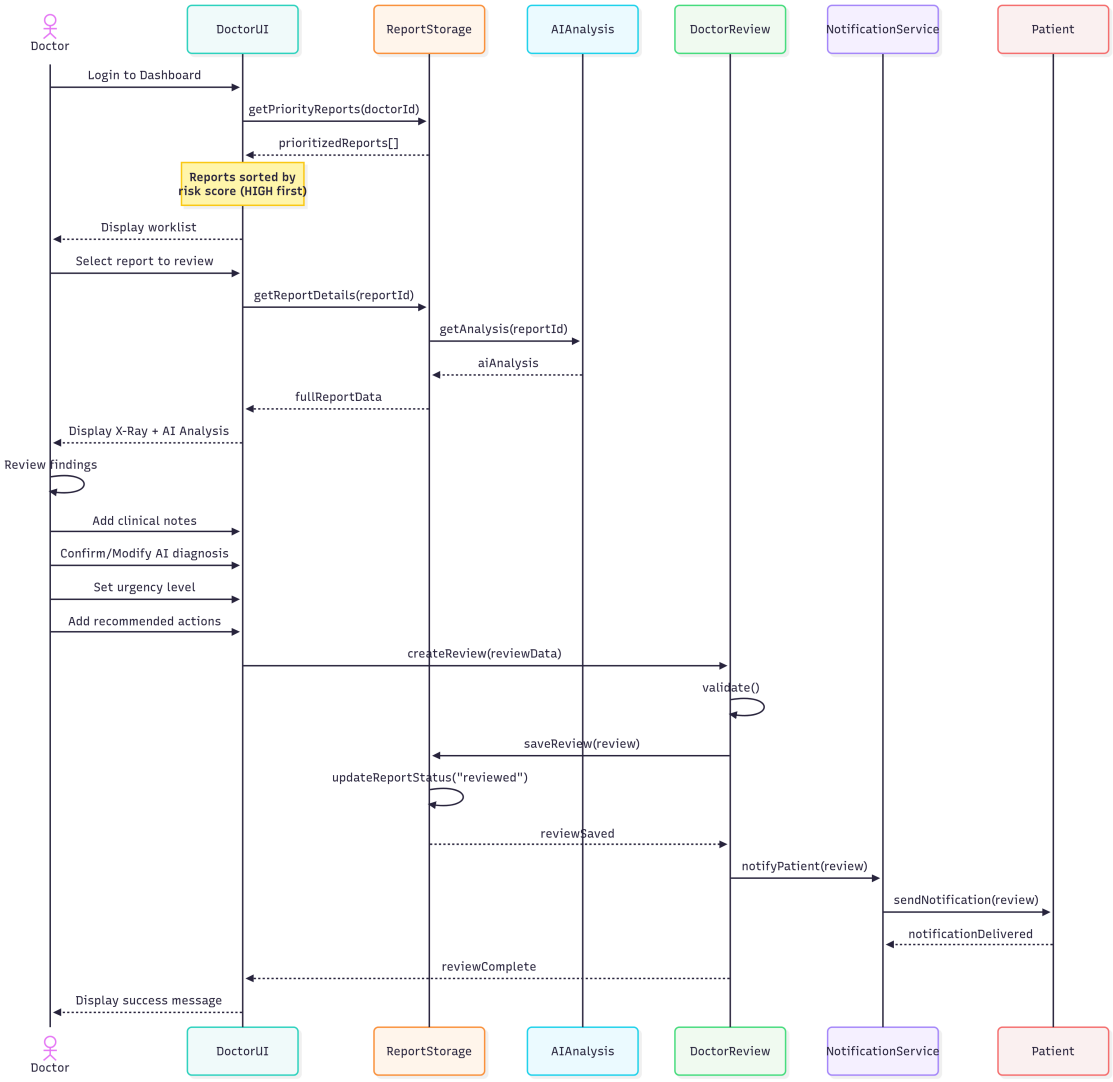
1. Patient Login and View Reports



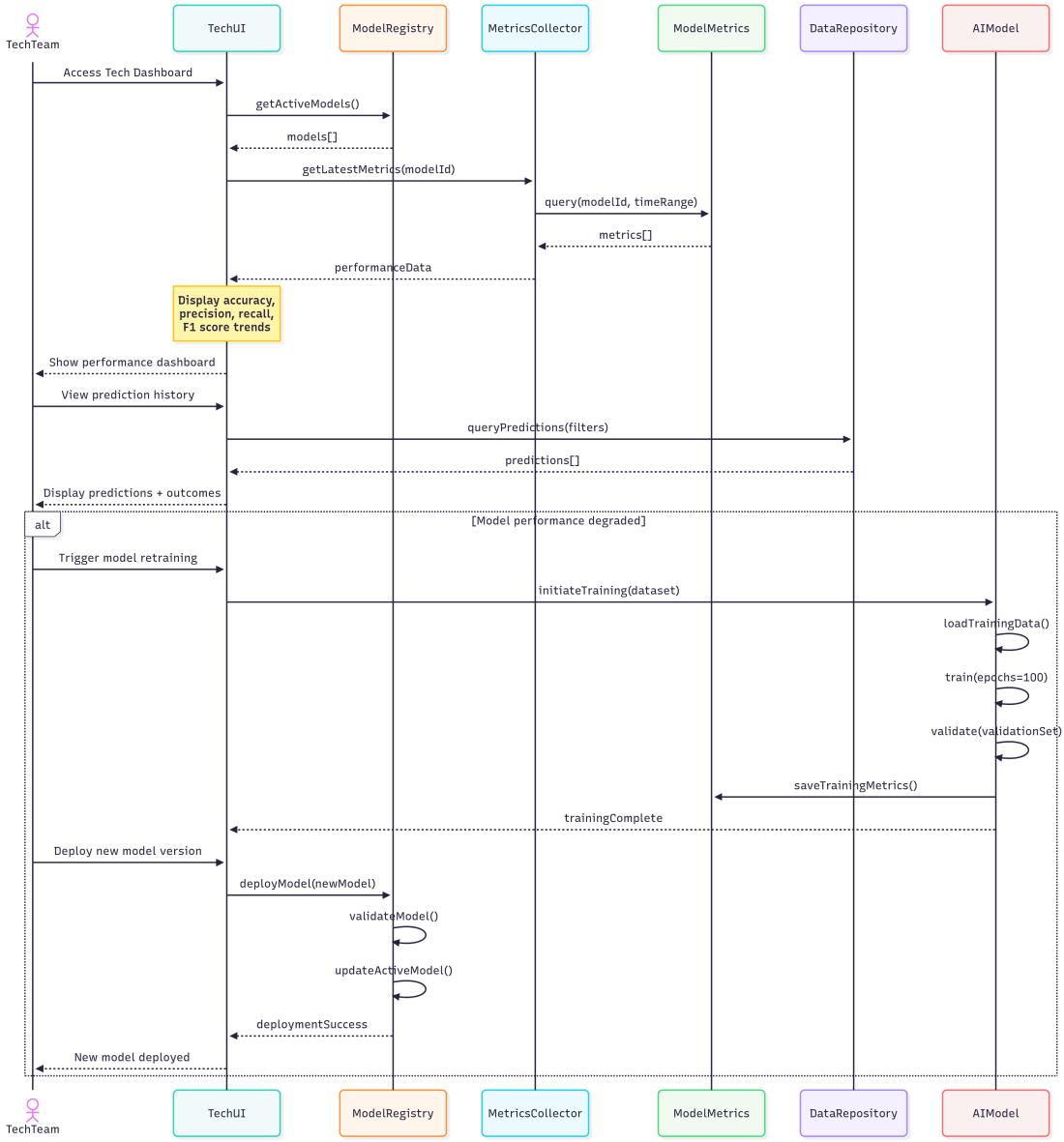
2. Radiologist Upload X-Ray Report (AI-Powered)



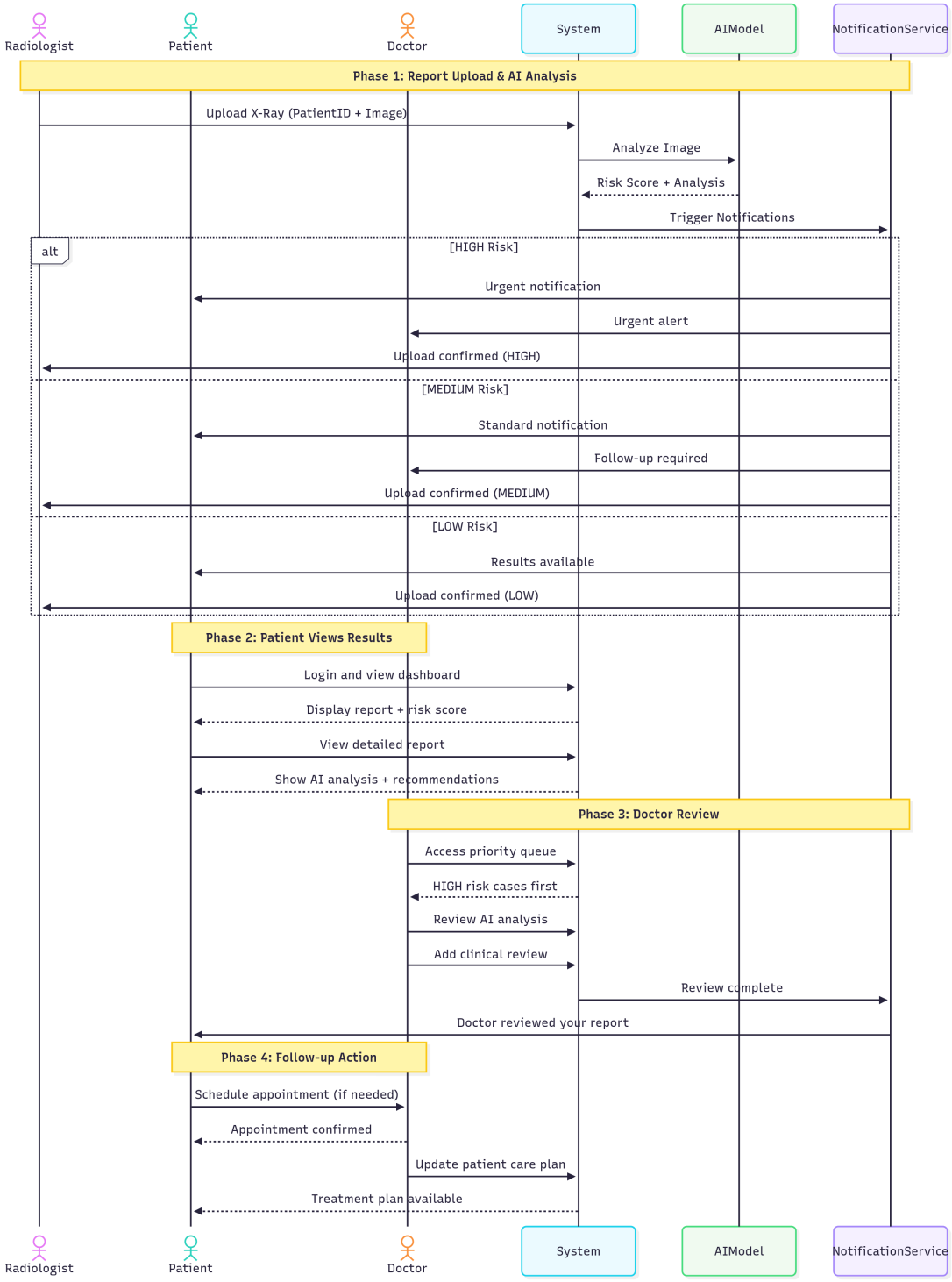
3. Doctor Review AI-Generated Report



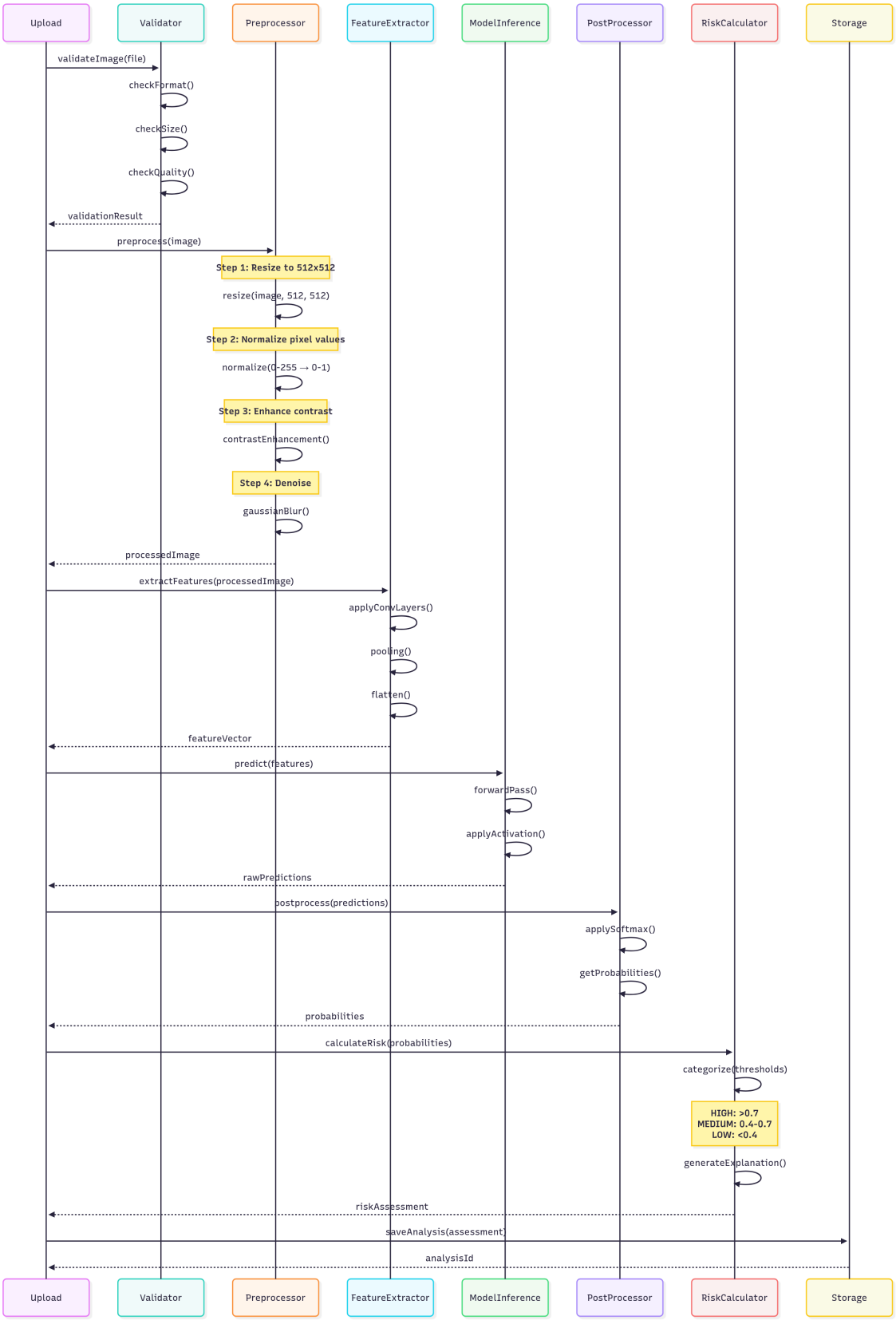
4. Tech Team Monitor AI Model Performance



5. Complete User Journey: From Upload to Treatment



6. AI Model Prediction Pipeline (Detailed)


Upload
Validator
Preprocessor
FeatureExtractor
ModelInference
PostProcessor
RiskCalculator
Storage
validateImage(file)
checkFormat()
checkSize()
checkQuality()
validationResult
preprocess(image)
Step 1: Resize to 512x512
resize(image, 512, 512)
Step 2: Normalize pixel values
normalize(0-255 → 0-1)
Step 3: Enhance contrast
contrastEnhancement()
Step 4: Denoise
gaussianBlur()
processedImage
extractFeatures(processedImage)
applyConvLayers()
pooling()
flatten()
featureVector
predict(features)
forwardPass()
applyActivation()
rawPredictions
postprocess(predictions)
applySoftmax()
getProbabilities()
probabilities
calculateRisk(probabilities)
categorize(thresholds)
HIGH: >0.7
MEDIUM: 0.4-0.7
LOW: <0.4
generateExplanation()
riskAssessment
saveAnalysis(assessment)
analysisId

7.Database schema V1.0(PostgreSQL 14+)

```sql
-- ================ USERS & AUTH ================
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    role VARCHAR(20) CHECK (role IN ('patient','radiologist','doctor','tech')) NOT NULL,
    profile_picture_url TEXT,
    phone VARCHAR(20),
    date_of_birth DATE,
    is_active BOOLEAN DEFAULT TRUE,
    is_email_verified BOOLEAN DEFAULT FALSE,
    preferences JSONB DEFAULT '{}',
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE sessions (
    session_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    token VARCHAR(512) NOT NULL UNIQUE,
    refresh_token VARCHAR(512) NOT NULL UNIQUE,
    ip_address INET,
    user_agent TEXT,
    login_time TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    last_activity TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    expires_at TIMESTAMPTZ NOT NULL,
    is_active BOOLEAN DEFAULT TRUE,
    logout_time TIMESTAMPTZ
);

-- ================ MEDICAL REPORTS ================
CREATE TABLE medical_reports (
    report_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    radiologist_id UUID REFERENCES users(id),
    study_type VARCHAR(100),
    report_image_url TEXT,
    image_metadata JSONB,
    risk_score VARCHAR(10) CHECK (risk_score IN ('low','medium','high')),
    confidence_score DECIMAL(5,4),
    findings TEXT,
    recommendations JSONB,
    status VARCHAR(20) CHECK (status IN ('pending','analyzing','analyzed','reviewed','archived')) DEFAULT 'pending',
    upload_date TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    analysis_date TIMESTAMPTZ,
    patient_age INT,
    patient_gender VARCHAR(20),
    medical_history JSONB,
    symptoms JSONB
);

CREATE TABLE report_images (
    image_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    report_id UUID NOT NULL REFERENCES medical_reports(report_id) ON DELETE CASCADE,
    image_url TEXT NOT NULL,
    image_type VARCHAR(50),
    format VARCHAR(20),
    width INT,
    height INT,
    file_size BIGINT,
    dicom_metadata JSONB,
    display_order INT DEFAULT 0
);

-- ================ AI & ANALYSIS ================
```

```sql
-- ================ AI & ANALYSIS ================
CREATE TABLE ai_models (
    model_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    model_name VARCHAR(255) NOT NULL,
    version VARCHAR(50) NOT NULL,
    architecture VARCHAR(100),
    framework VARCHAR(50),
    model_file_url TEXT,
    trained_date TIMESTAMPTZ,
    deployed_date TIMESTAMPTZ,
    baseline_accuracy DECIMAL(5,4),
    precision DECIMAL(5,4),
    recall DECIMAL(5,4),
    f1_score DECIMAL(5,4),
    is_active BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE ai_analysis (
    analysis_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    report_id UUID NOT NULL REFERENCES medical_reports(report_id) ON DELETE CASCADE,
    model_id UUID REFERENCES ai_models(model_id),
    risk_category VARCHAR(10) CHECK (risk_category IN ('low','medium','high')),
    risk_probability DECIMAL(5,4),
    findings JSONB,
    feature_maps JSONB,
    attention_maps JSONB,
    processed_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    processing_time_ms INT
);

CREATE TABLE model_metrics (
    metric_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    model_id UUID NOT NULL REFERENCES ai_models(model_id) ON DELETE CASCADE,
    accuracy DECIMAL(5,4),
    precision DECIMAL(5,4),
    recall DECIMAL(5,4),
    f1_score DECIMAL(5,4),
    confusion_matrix JSONB,
    total_predictions BIGINT DEFAULT 0,
    correct_predictions BIGINT DEFAULT 0,
    auc_roc DECIMAL(5,4),
    average_processing_time_ms INT,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- ================ DOCTOR REVIEWS ================
```

```sql
-- ================ DOCTOR REVIEWS ================
CREATE TABLE doctor_reviews (
    review_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    report_id UUID NOT NULL REFERENCES medical_reports(report_id) ON DELETE CASCADE,
    doctor_id UUID NOT NULL REFERENCES users(id),
    review_notes TEXT,
    clinical_findings TEXT,
    confirmed_ai_diagnosis BOOLEAN,
    ai_accuracy_rating INT CHECK (ai_accuracy_rating BETWEEN 1 AND 5),
    doctor_risk_assessment VARCHAR(10) CHECK (doctor_risk_assessment IN ('low','medium','high')),
    urgency_level VARCHAR(20) CHECK (urgency_level IN ('routine','follow_up','urgent','critical')),
    recommended_action TEXT,
    treatment_plan TEXT,
    medication_prescribed JSONB,
    follow_up_required BOOLEAN DEFAULT FALSE,
    follow_up_date DATE,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- ================ NOTIFICATIONS ================
CREATE TABLE notifications (
    notification_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    type VARCHAR(50),
    priority VARCHAR(10) CHECK (priority IN ('low','normal','high','urgent')) DEFAULT 'normal',
    title VARCHAR(255),
    message TEXT,
    action_url TEXT,
    is_read BOOLEAN DEFAULT FALSE,
    sent_via_email BOOLEAN DEFAULT FALSE,
    sent_via_sms BOOLEAN DEFAULT FALSE,
    sent_via_push BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    expires_at TIMESTAMPTZ
);

-- ================ AUDIT & COMPLIANCE ================
CREATE TABLE audit_logs (
    log_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id),
    action VARCHAR(100) NOT NULL,
    entity_type VARCHAR(50),
    entity_id UUID,
    ip_address INET,
    user_agent TEXT,
    old_values JSONB,
    new_values JSONB,
    status VARCHAR(20) CHECK (status IN ('success','failure','error')),
    error_message TEXT,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- ================ PATIENT HISTORY ================
CREATE TABLE patient_medical_history (
    history_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID NOT NULL UNIQUE REFERENCES users(id) ON DELETE CASCADE,
    conditions JSONB,
    allergies JSONB,
    medications JSONB,
    family_history JSONB,
    lifestyle JSONB,
    previous_surgeries JSONB,
    hospitalizations JSONB,
    immunizations JSONB,
    last_updated TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- ================ SYSTEM SETTINGS ================
CREATE TABLE system_settings (
    setting_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    setting_key VARCHAR(100) UNIQUE NOT NULL,
    setting_value JSONB NOT NULL,
    description TEXT,
    is_sensitive BOOLEAN DEFAULT FALSE,
    updated_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);
```