Healthcare Appointment Management System
Phase 1 Documentation

Abdelrahman Amr (ID: 120210335)
Salma Amin (ID: 120210389)

May 1, 2025

Contents

Chapter 1

Project Proposal

1.1 System Overview

This web-based system addresses critical challenges in healthcare appointment management through:

- Secure patient-doctor matching using JWT authentication
- Intelligent schedule optimization algorithms
- HIPAA-compliant medical record integration

1.2 Key Objectives

- Reduce appointment no-show rates by 30% via automated SMS/email reminders
- Improve doctor utilization efficiency by 25% through smart scheduling
- Maintain 99.5% system availability with AWS cloud deployment

Chapter 2

Software Requirements Specification

2.1 Functional Requirements

User Authentication and Authorization

- Secure login for Patients, Doctors, and Admins
- Role-based access control
- Password reset via email
- Session timeout after 15 minutes of inactivity

Patient Management

- Register new patients with personal information
- View and update profile details
- Access appointment and medical history

Doctor Management

- Manage doctor profiles (specialization, contact info, availability)
- Assign doctors to departments
- Access patient records and appointment history

Appointment Management

- Book, cancel, and reschedule appointments
- View doctor availability in real-time
- Prevent double-booking and log changes
- Send confirmation and reminder notifications

Schedule Management

- Doctors and Admins set available time slots
- Patients can only book during available periods

Department Management

- Organize doctors under specific departments
- Display department descriptions and services

Medical Records

- Create and view diagnosis, prescriptions, and notes
- Link records to patients and appointments

Reporting and Analytics

- Generate reports for appointments, utilization, and peak hours
- Exportable summaries for management decisions

2.2 Non-Functional Requirements

Performance

- Page load time under 2 seconds
- Support at least 100 concurrent users
- Fast database query responses

Security

- AES-256 encryption for sensitive data
- HIPAA-compliant audit logs
- Penetration testing and activity monitoring

Usability

- Mobile-responsive design
- Accessible UI for patients and staff
- No technical training required

Reliability

- 99.5% uptime target
- Daily backups
- Error tracking and logging

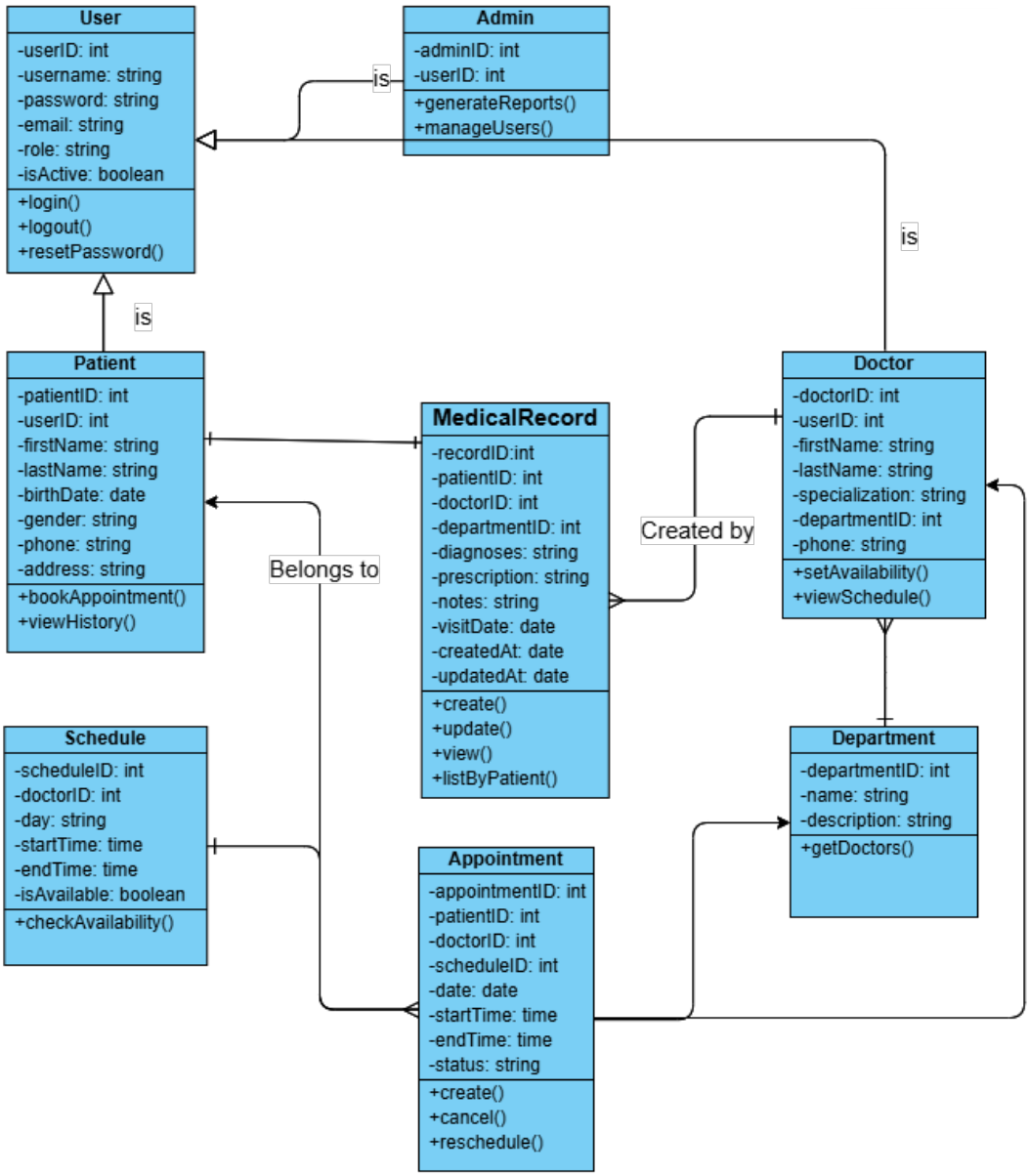
Scalability

- Architecture supports horizontal and vertical scaling
- Flexible schema for future growth

Chapter 3

System Design

3.1 Class Diagram



3.2 Entity Relationship Diagrams

3.2.1 Basic ERD

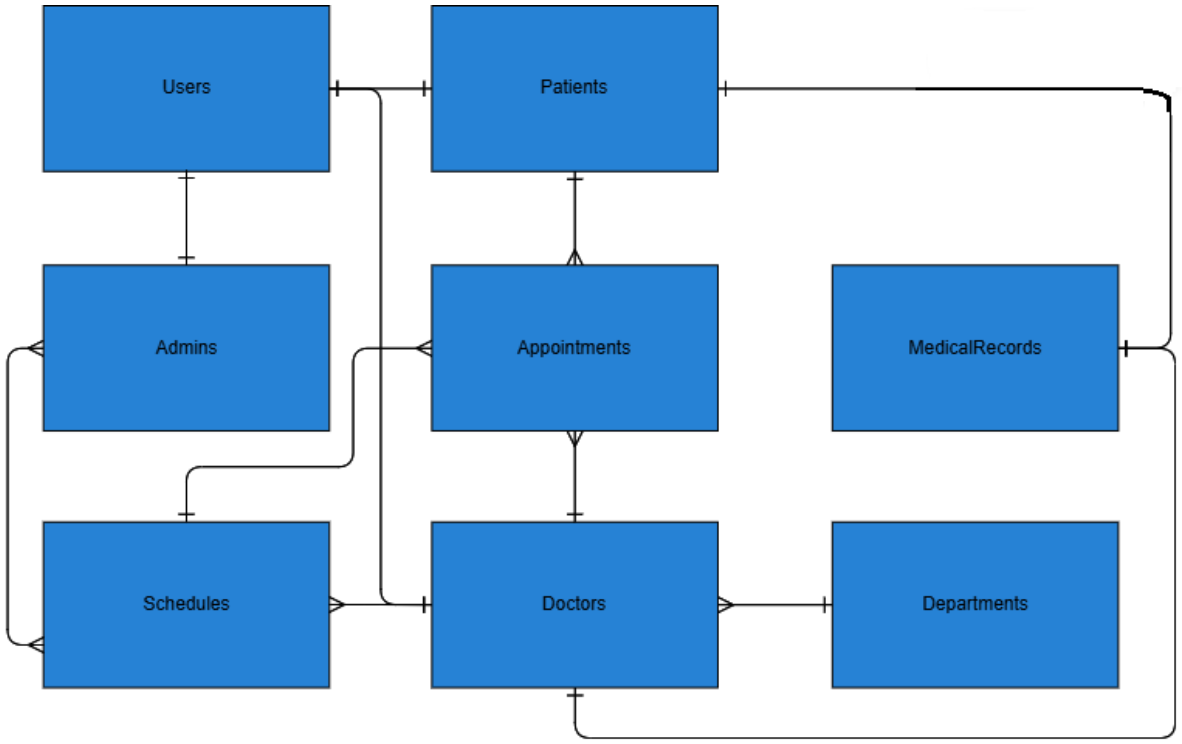


Figure 3.2: Basic Entity-Relationship Diagram (ERD)

3.2.2 Enhanced ERD

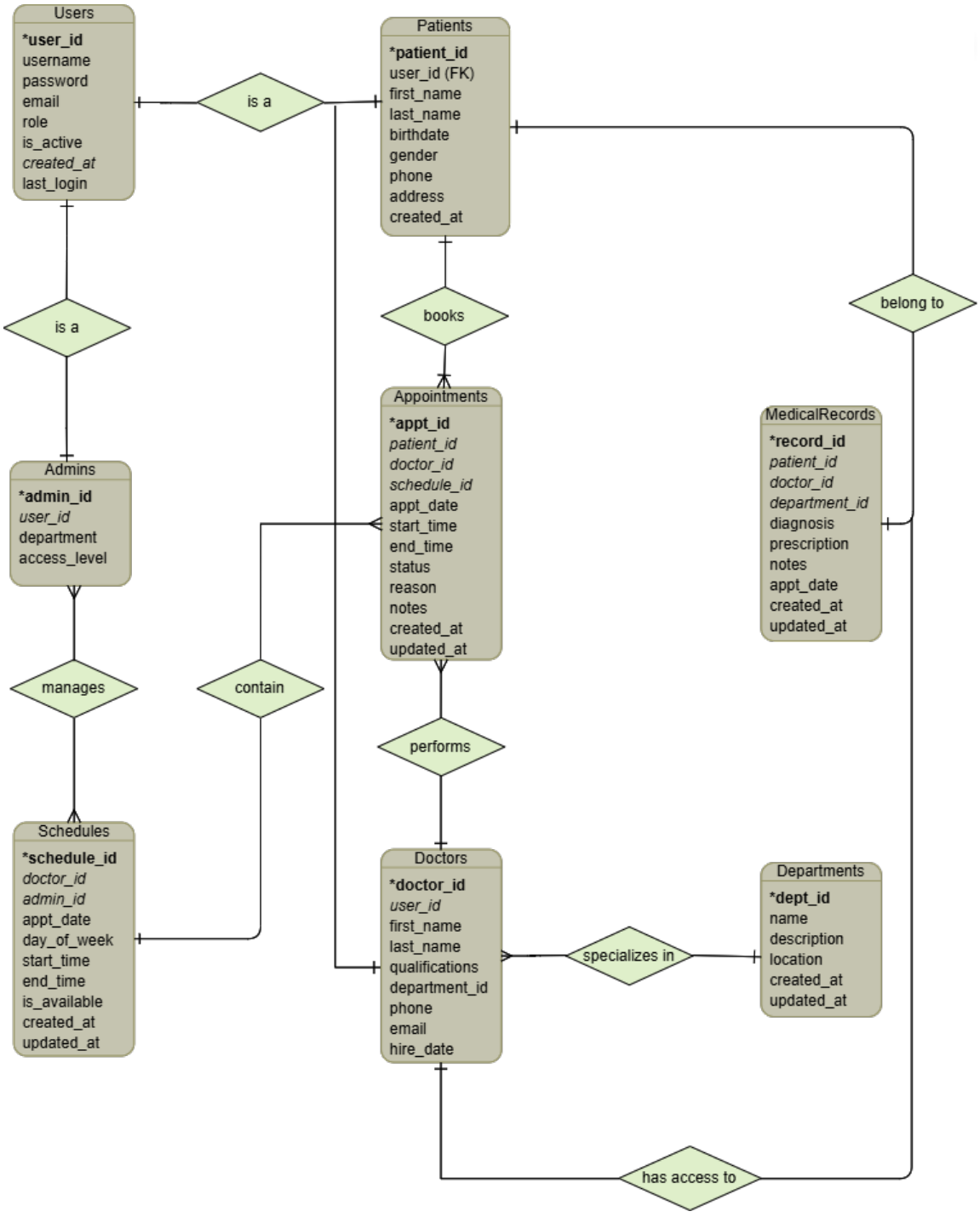


Figure 3.3: Enhanced Entity-Relationship Diagram (EERD)

Chapter 4

Risk Analysis

Risk	Impact	Mitigation
Data Breach	High	Implement AES-256 encryption Regular security audits
System Downtime	Medium	AWS auto-scaling Load-balanced architecture

Chapter 5

Project Plan

5.1 8-Week Timeline

Week	Deliverables
1-2	Database schema implementation (PostgreSQL)
3-4	Core API development (Django REST Framework)
5-6	Frontend development (React.js components)
7-8	Testing & Deployment (AWS EC2)

5.2 Technology Stack

- **Frontend**: React 18 + Material-UI
- **Backend**: Django 4.2 + Django REST Framework
- **Database**: PostgreSQL 15
- **DevOps**: Docker + GitHub Actions

Bibliography

[1] U.S. Department of Health & Human Services. (2023). *HIPAA Security Rule*. Retrieved from https://www.hhs.gov/hipaa

[2] Django Software Foundation. (2023). *Django Documentation*. Retrieved from https://docs.djangoproject.com

[3] Meta Platforms. (2023). *React Documentation*. Retrieved from https://react.dev

Appendix

Deliverables Checklist

Software Requirements Specification

Class Diagram (Fig. 3.1)

ERD Diagram (Fig. 3.2)

Risk Analysis Matrix

Project Timeline