ADDIS ABABA UNIVERISITY

COLLEGE OF TECHNOLOGY AND BUILT ENVIRONMENT

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

PROJECT TITLE: HEALTH NET

SOFTWARE DESIGN SPECIFICATION

SECTION: 1 **GROUP: 4**

Name	ID No
1. Asanti Oluma	UGR/8165/16
2. Bekalu Addisu	UGR/9538/16
3. Fita Alemayehu.	UGR/7071/16
4. Lemi Gobena.	UGR/1589/16
5. Martha Tegegne	UGR/4457/16
6. Misganaw Habtamu.	UGR/1707/16
7. Olit Oljira.	UGR/8925/16
8. Selamawit Mulat	UGR/1033/16

ADVISOR: Mrs. Nuniyat Kifle

Date: December 18 ,2025

Table of Contents

Contents **Pages**

List of Tables

List of Figures

Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
SDS	Software Design Specification- A document that describes the architecture, components, and detailed design of a software system.
EHR	Electronic Health Record - A digital version of a patient's health information.
HealthNet	The proposed National Electronic Health Record system for Ethiopia.
UPI	Unique Patient Identifier – A unique code assigned to each patient within HealthNet.
QR Code	Quick Response Code – Used for quick patient identification and emergency data access.
RBAC	Role-Based Access Control – A security model restricting system access based on user roles.
UI	User Interface – The visual and interactive part of the system.
API	Application Programming Interface – Enables communication between software components.
JWT	JSON Web Token – A secure token used for authentication and authorization.
SRS	Software Requirements Specification – Document detailing system requirements.
HTTPS	Hypertext Transfer Protocol Secure – Secured communication protocol.
SQL	Structured Query Language – Used for database management.
UML	Unified Modeling Language – A standardized modeling language for system design.
WBS	Work Breakdown Structure – A project management tool for task decomposition.
CPM	Critical Path Method – A scheduling method for project management.

1. Introduction

1.1. Purpose

The purpose of this Software Design Specification (SDS) document is to translate the functional and non-functional requirements specified in the Software Requirements Specification (SRS) into a comprehensive technical blueprint for the HealthNet system. This document provides detailed architectural, structural, and behavioral models that will guide the development team in implementing a secure, scalable, and interoperable National Electronic Health Record (EHR) system for Ethiopia.

1.2. General Overview

HealthNet is a web-based, centralized EHR platform designed to unify patient health records across all levels of healthcare in Ethiopia—from rural health posts to specialized urban hospitals. The system will replace fragmented paper-based records with a secure digital ecosystem, featuring:

- ➢ Unique Patient Identifier (UPI) linked to QR codes for emergency access.
- ➢ Role-based dashboards for Admins, Doctors, Lab Technicians, and Patients.
- ➢ Emergency Access Module allowing first responders to retrieve critical patient data via QR scan.
- ➢ Secure data management with encryption, audit trails, and RBAC.

The system follows a client-server architecture with a *React.js* frontend, *Node.js* backend, and *PostgreSQL* database. It is designed to comply with Ethiopia's eHealth Strategy 2025 and WHO digital health guidelines.

1.3. Development Methods and Contingencies

The HealthNet project adopts an integrated object-oriented methodology using UML for modeling and waterfall-inspired phased development for structured progress. Key methods include:

- ❖ Object-Oriented Analysis and Design (OOAD): Used for modeling system structure and behavior.
- ❖ Unified Modeling Language (UML): For class, sequence, state, and deployment diagrams.
- ❖ Agile-inspired prototyping: For iterative UI/UX and feature validation.
- ❖ Waterfall phases: Requirements → Design → Implementation → Testing → Deployment.

Contingencies:

- ❖ Internet connectivity issues in remote areas may require offline data synchronization features.
- ❖ Data privacy law delays may affect certain compliance features; the system will be designed to be adaptable.
- ❖ Integration with existing health programs (e.g., HIV/TB databases) may be deferred to later phases.
- ❖ User resistance to digital systems will be mitigated through training and pilot feedback loops.
- ❖ Technical resource constraints may lead to prioritized roll-out of core features first.

2. System Architecture

2.1. Subsystem decomposition

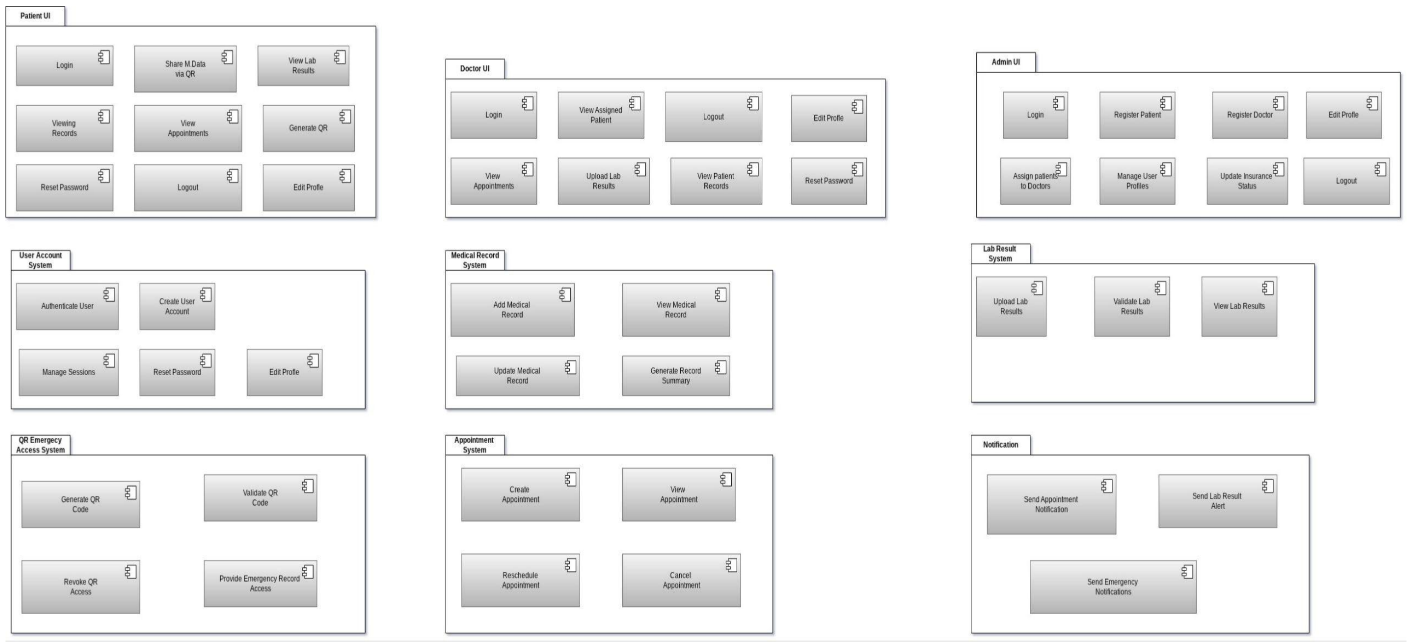


Figure 2-1: Layer-1

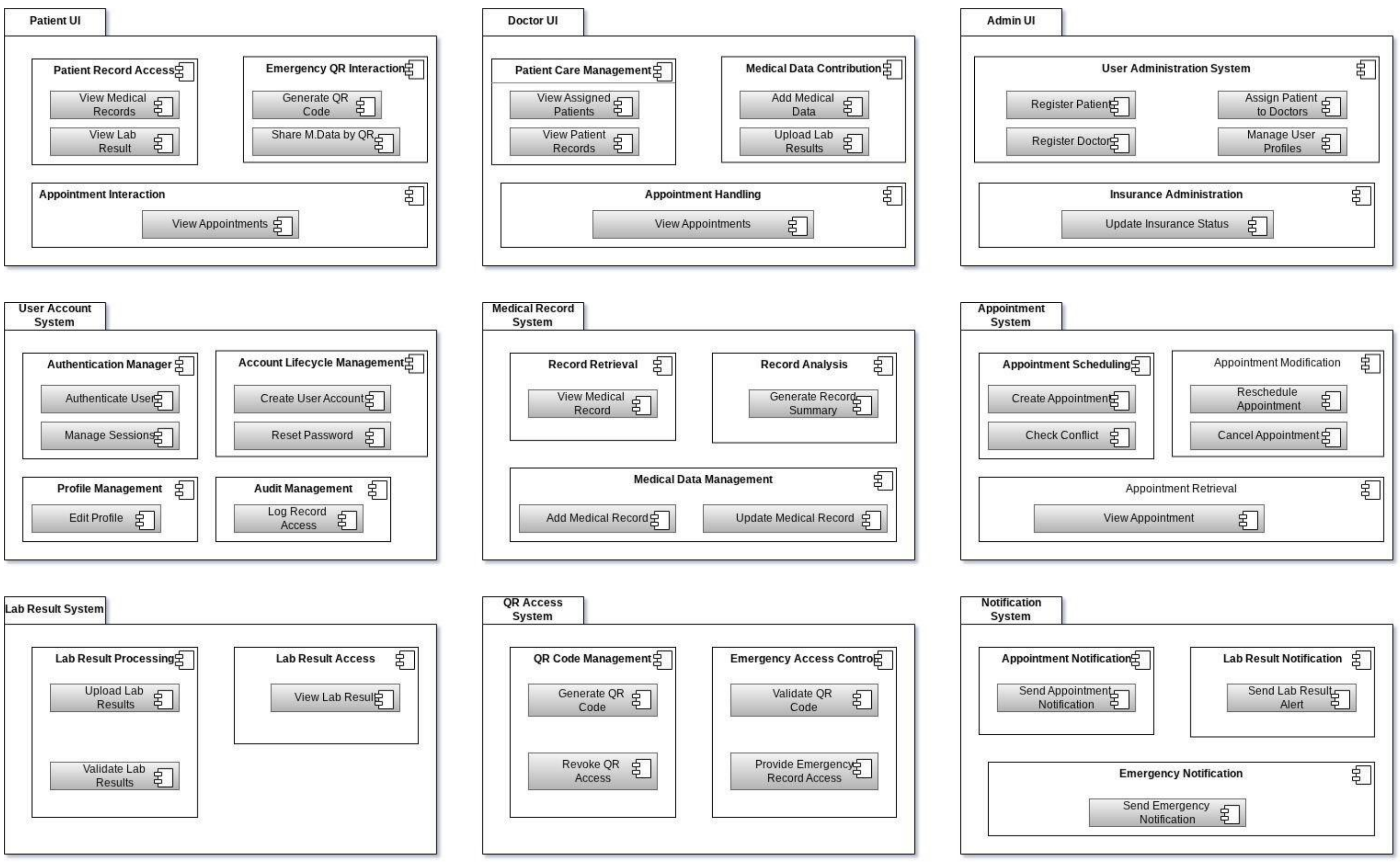


Figure 2-2: Layer-2

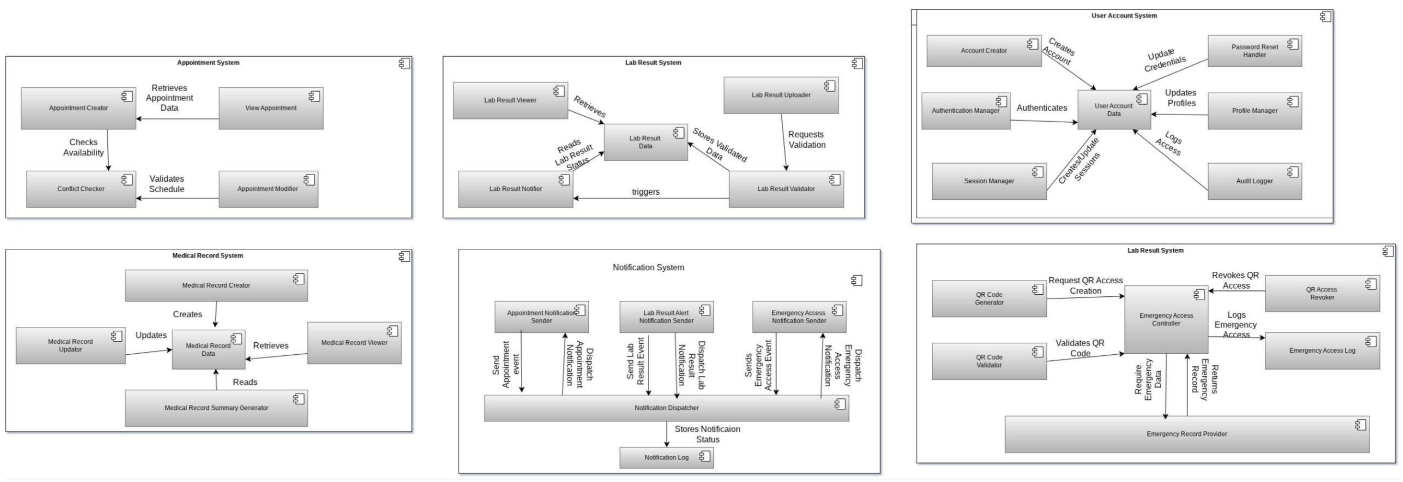


Figure 2-3: Layer-3

2.2. Hardware/software mapping

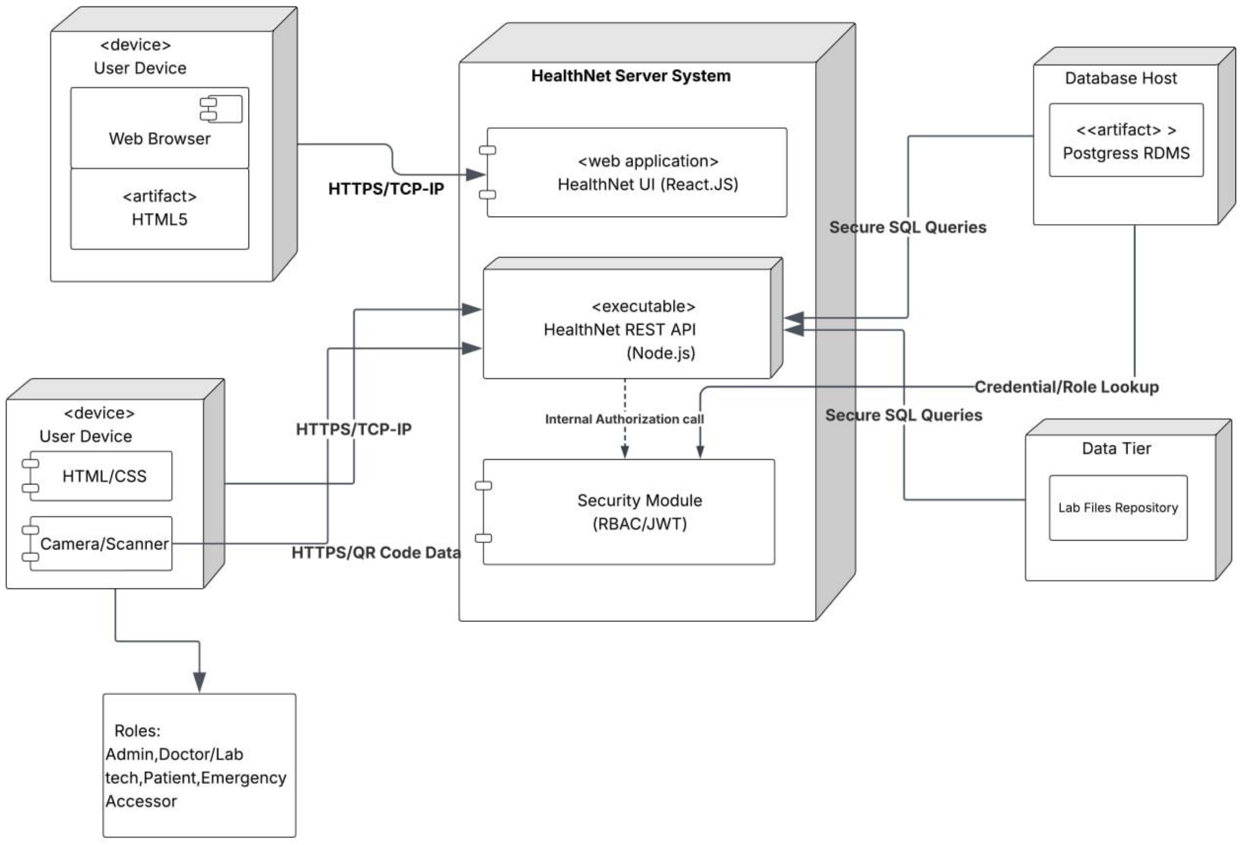


Figure 2-3: UML Deployment Diagram

2.3. Access control

The HealthNet system implements a Role-Based Access Control (RBAC) model to ensure secure and appropriate access to sensitive patient data and system functionality. Access permissions are assigned based on user roles, each with defined privileges aligned with their responsibilities.

Table 1: Role-Based Access Control (RBAC) model

Role	Permissions / Access Rights	Restrictions
Administrator	✓Create, update, deactivate user accounts ✓Assign patients to doctors ✓Manage facilities and system settings ✓View system logs and audit trails	Cannot view or modify patient medical records unless explicitly granted emergency override.
Doctor	✓View assigned patient records ✓Add diagnoses and treatment plans ✓Upload lab results ✓Create and manage appointments ✓View patient emergency info	Cannot access records of unassigned patients; cannot modify user roles or system configurations.
Patient	✓View own medical records and lab results ✓Update personal and emergency contact information ✓Download personal QR code for emergency access ✓Request and reschedule appointments	Cannot access other patients' data; cannot modify medical entries made by doctors.
Lab Technician	✓Upload lab result files ✓View patient lab history (for assigned tests) ✓Confirm receipt of lab samples	Cannot add diagnoses or modify patient profiles; access limited to lab-related modules.
Emergency Responder (QR Scanner)	✓Scan patient QR code to retrieve emergency information (blood type, allergies, emergency contact)	Access is time-limited (token expires after 5 minutes); only pre-authorized fields are visible.

3. Object Model

3.1. Class Diagram

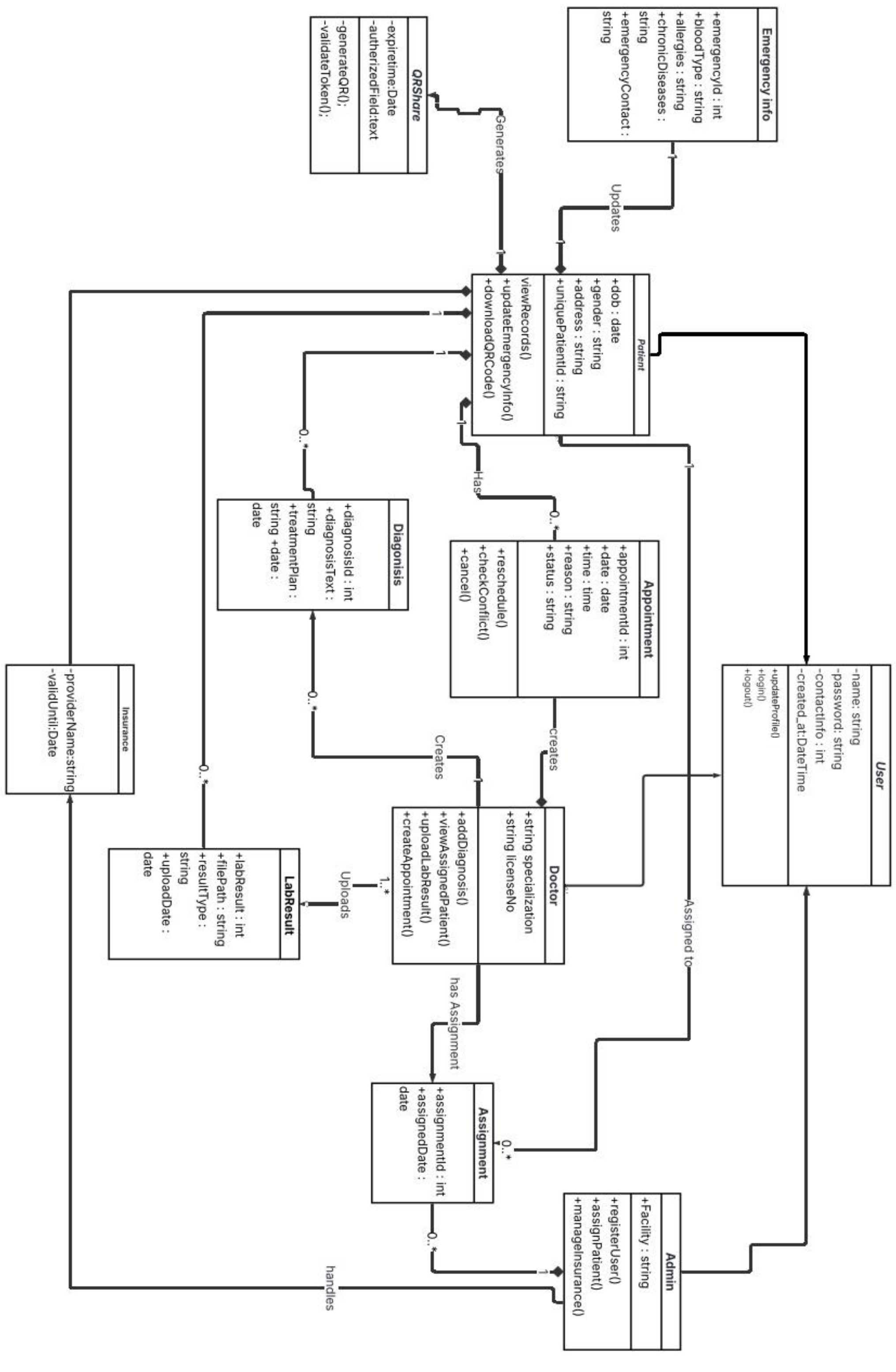


Figure 3-1: Class Diagram

3.2. Sequence Diagram

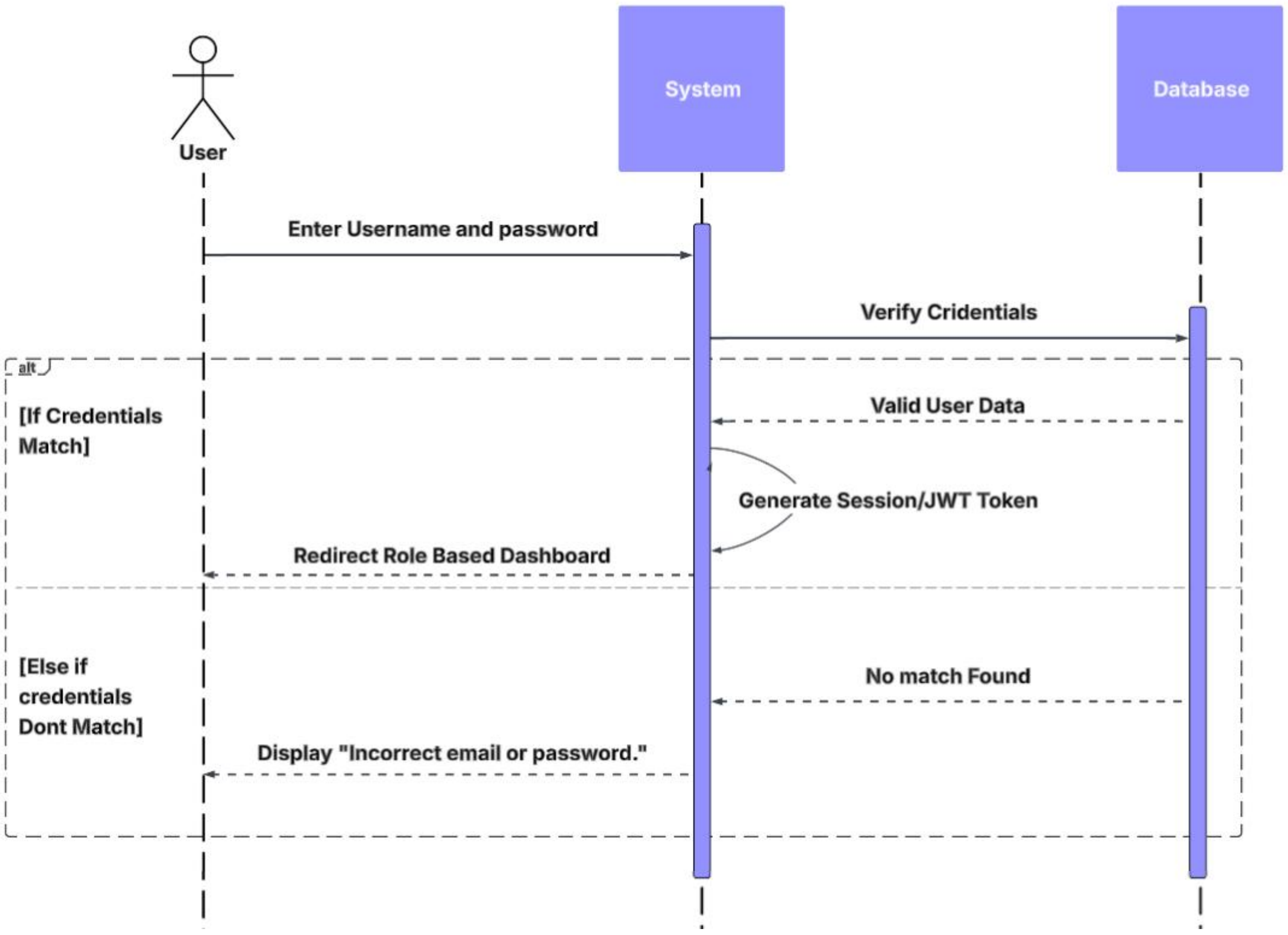

System
Database
User
Enter Username and password
Verify Cridentials
alt
[If Credentials Match]
Valid User Data
Generate Session/JWT Token
Redirect Role Based Dashboard
[Else if credentials Dont Match]
No match Found
Display "Incorrect email or password."

Figure 3-2: User Login

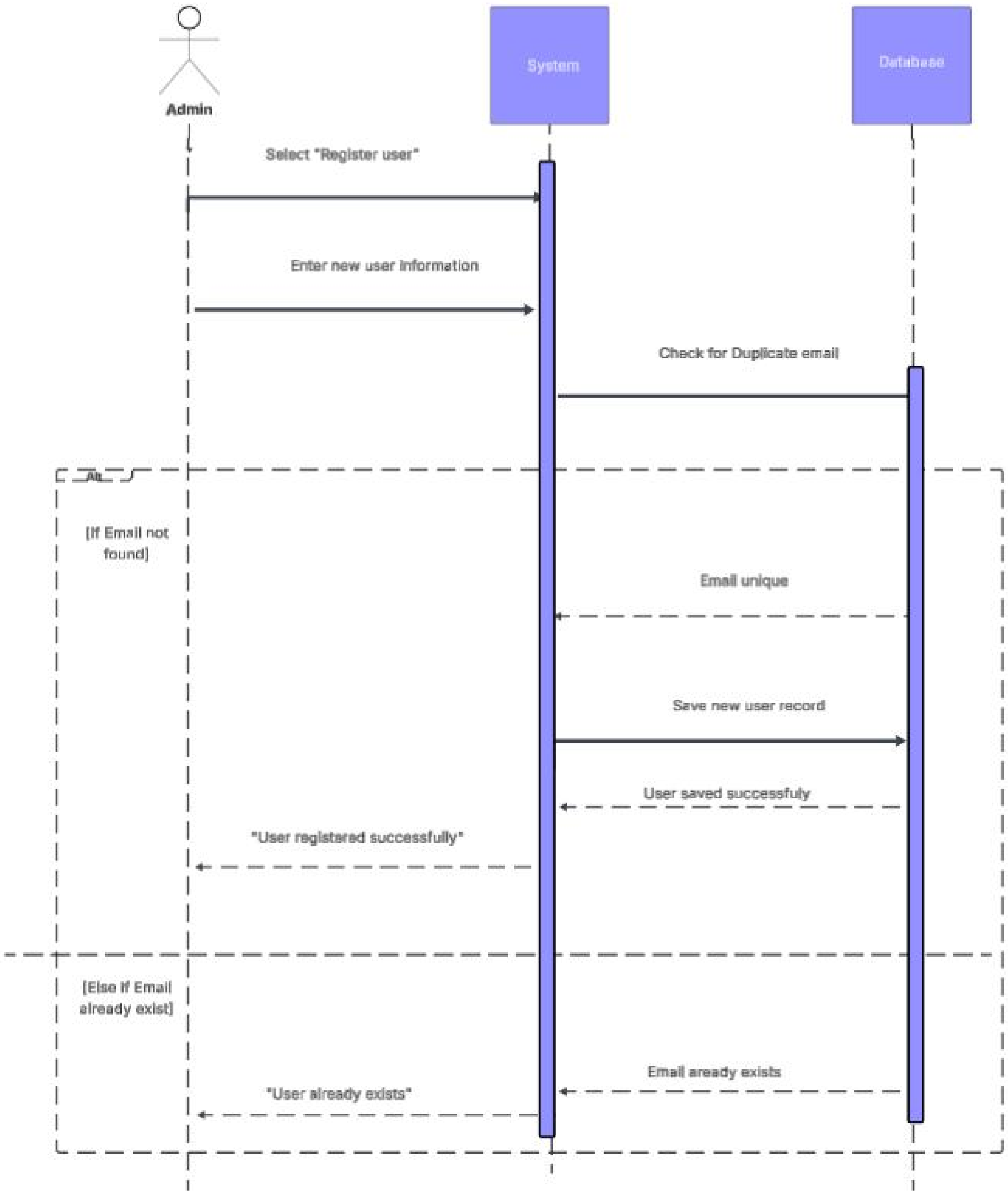
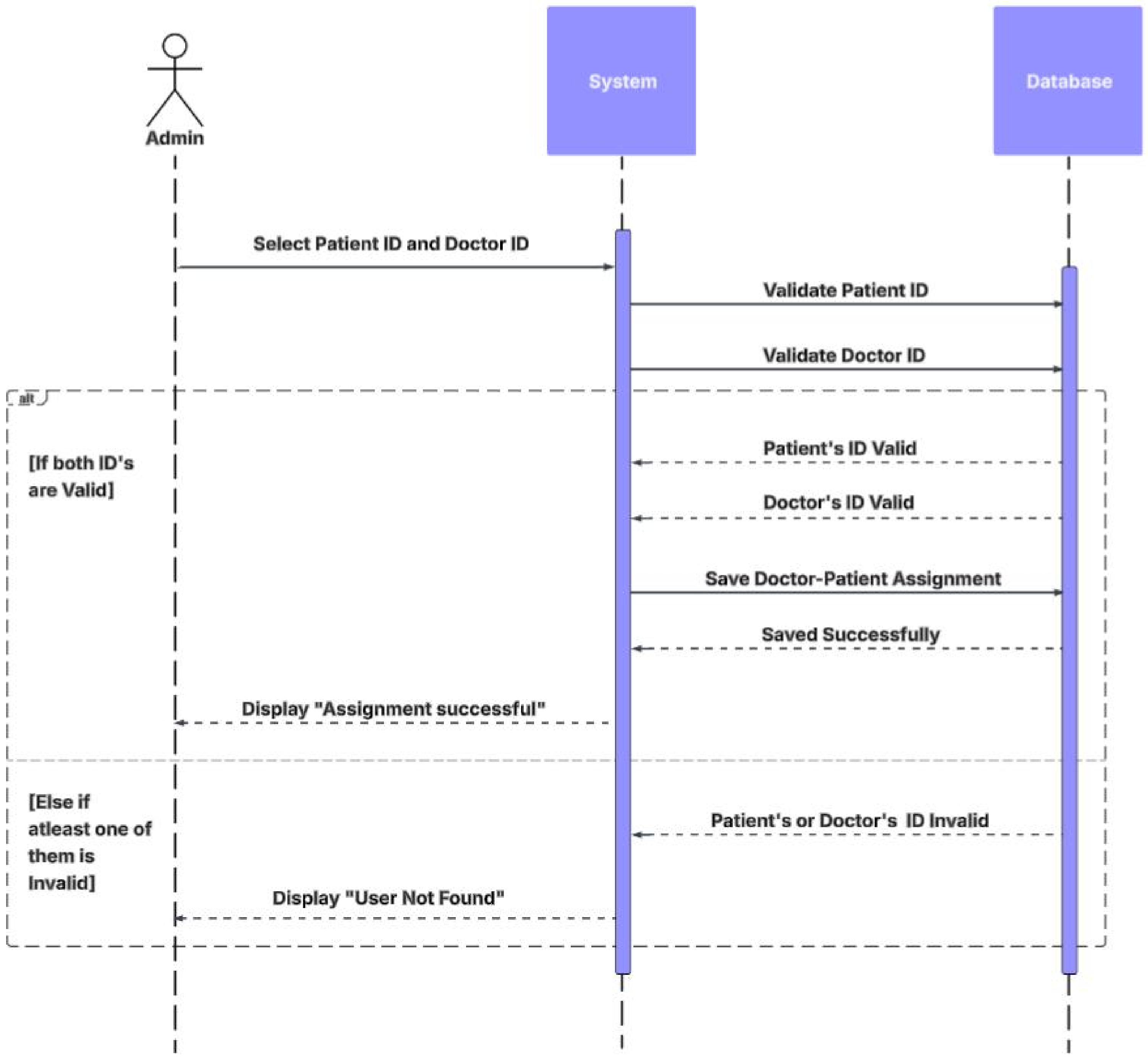


Figure 3-3: Register User



Admin
System
Database
Select Patient ID and Doctor ID
Validate Patient ID
Validate Doctor ID
alt
[If both ID's are Valid]
Patient's ID Valid
Doctor's ID Valid
Save Doctor-Patient Assignment
Saved Successfully
Display "Assignment successful"
[Else if atleast one of them is Invalid]
Patient's or Doctor's ID Invalid
Display "User Not Found"

Figure 3-4: Assign Patient to Doctor

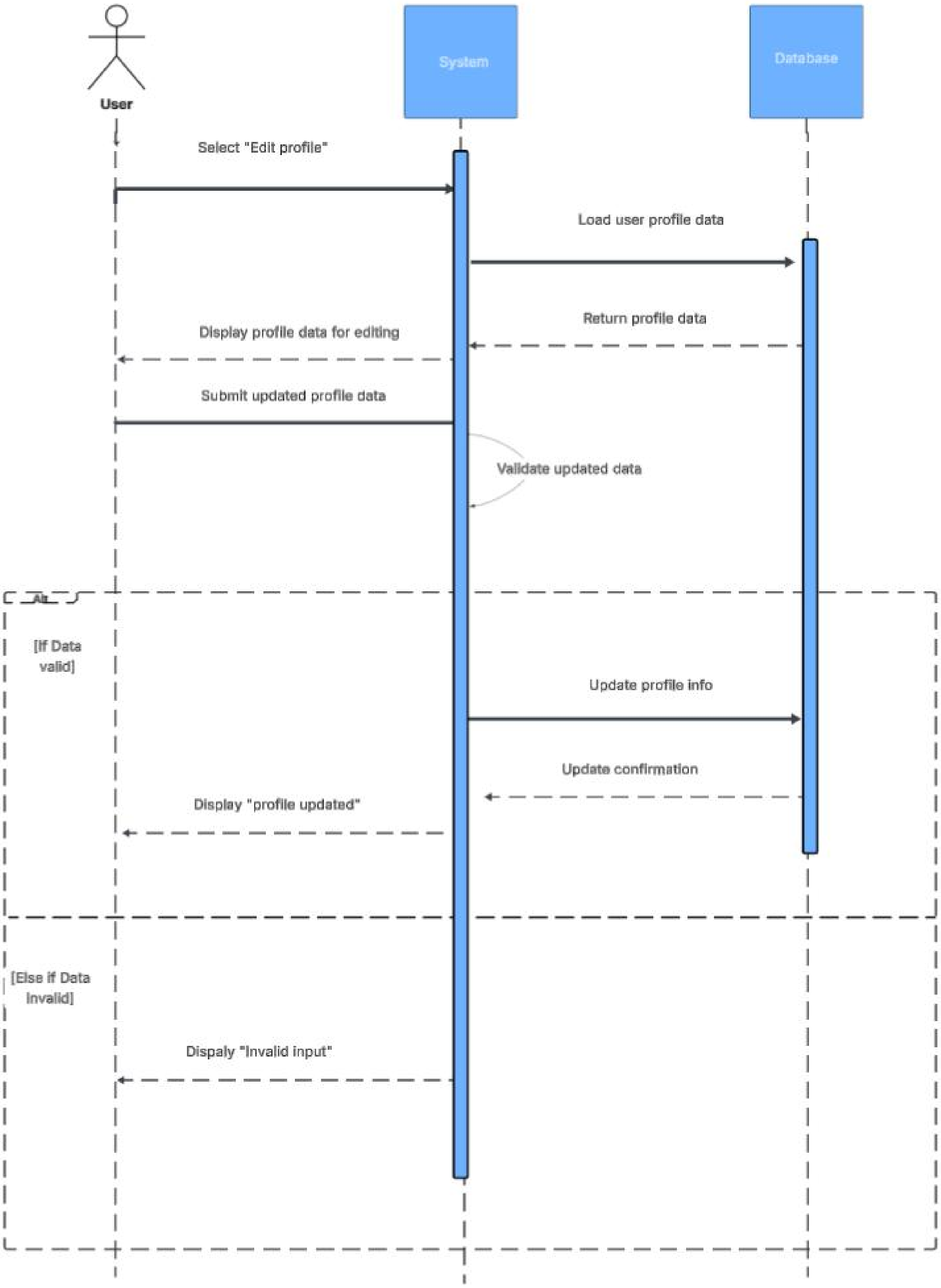


Figure 3-5: Manage User Profiles

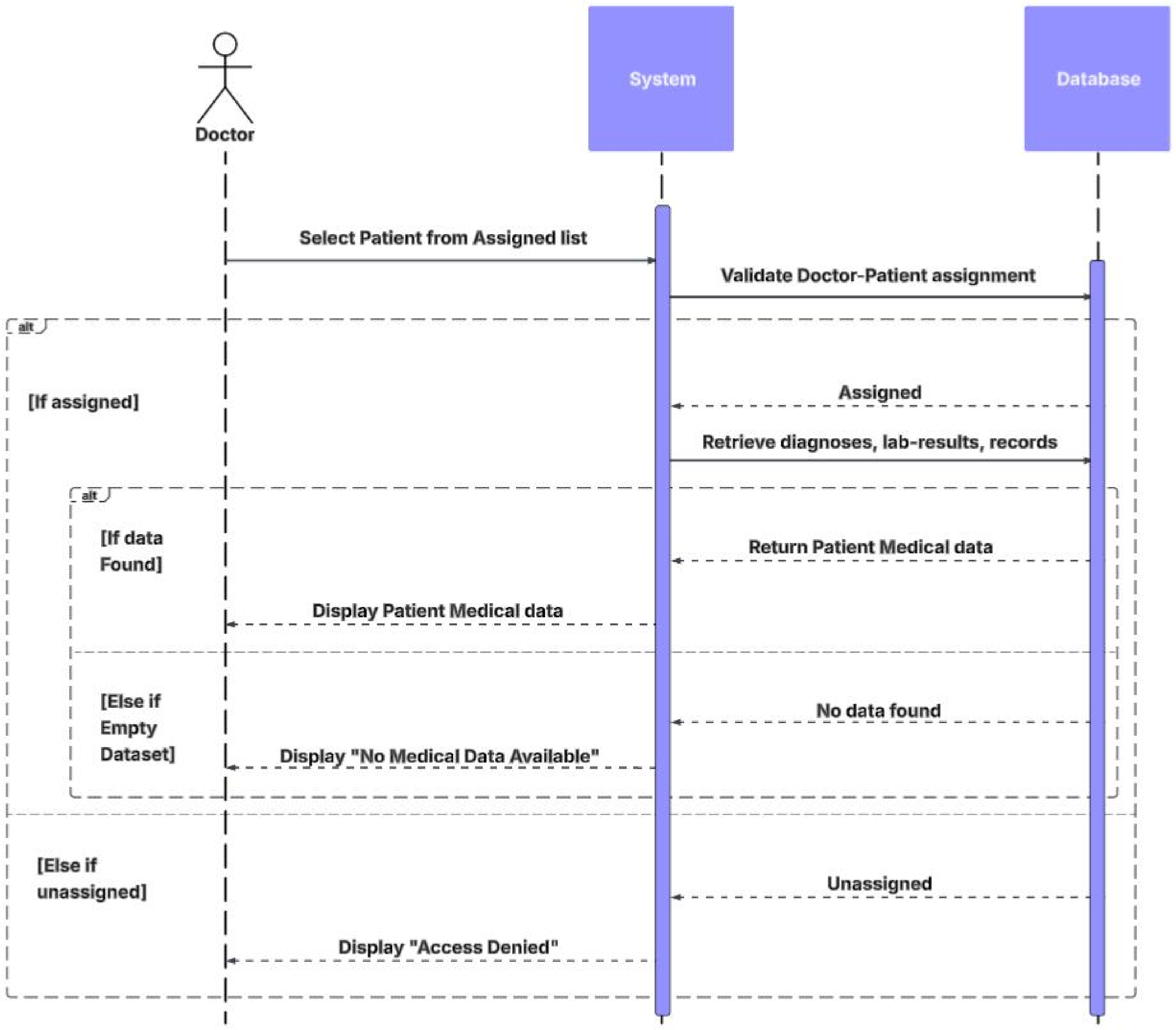

Doctor
System
Database
Select Patient from Assigned list
Validate Doctor-Patient assignment
alt
[If assigned]
Assigned
Retrieve diagnoses, lab-results, records
alt
[If data Found]
Return Patient Medical data
Display Patient Medical data
[Else if Empty Dataset]
No data found
Display "No Medical Data Available"
[Else if unassigned]
Unassigned
Display "Access Denied"

Figure 3-6: View Patient Record

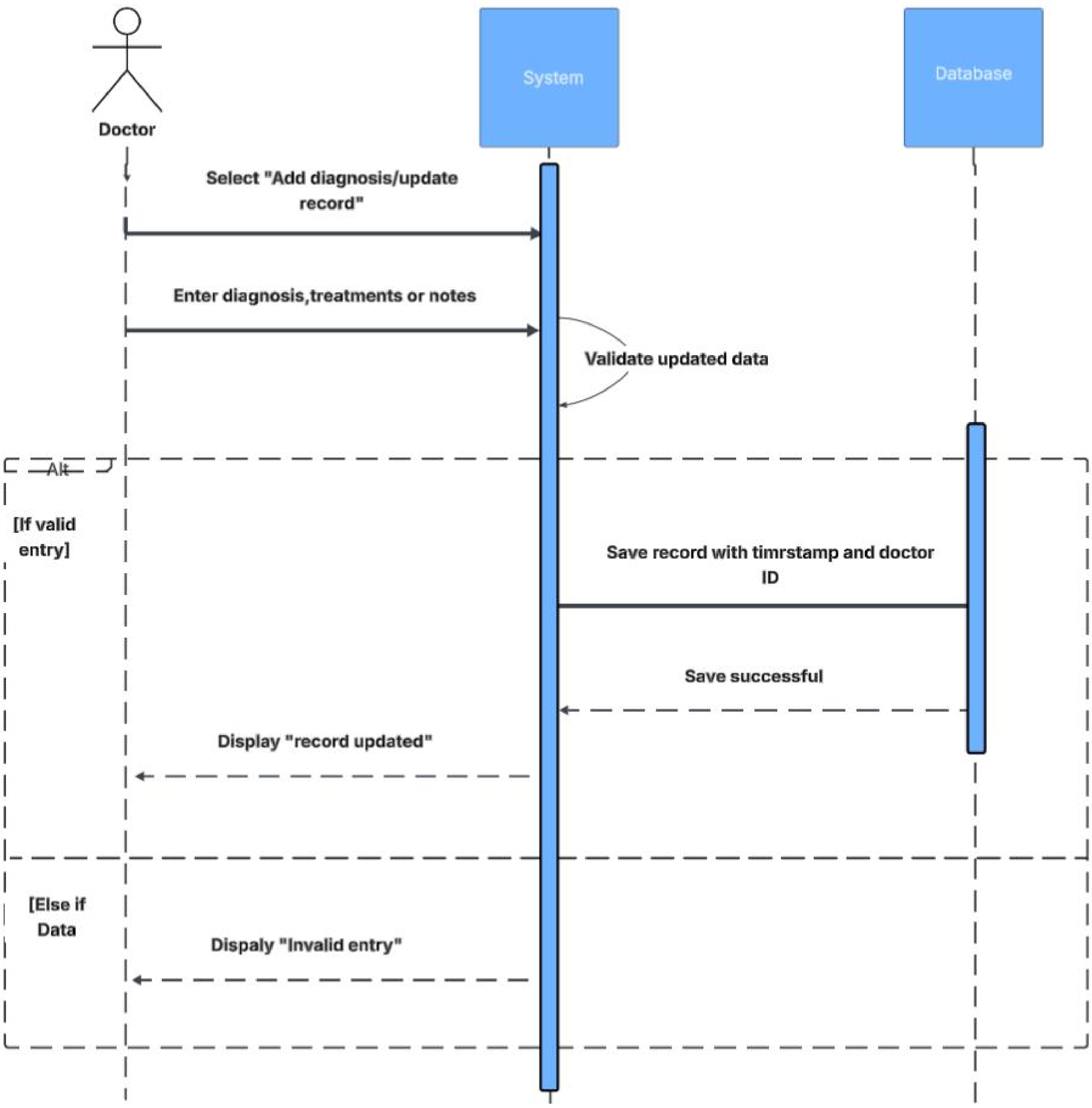


Figure 3-7: Update Patient Record

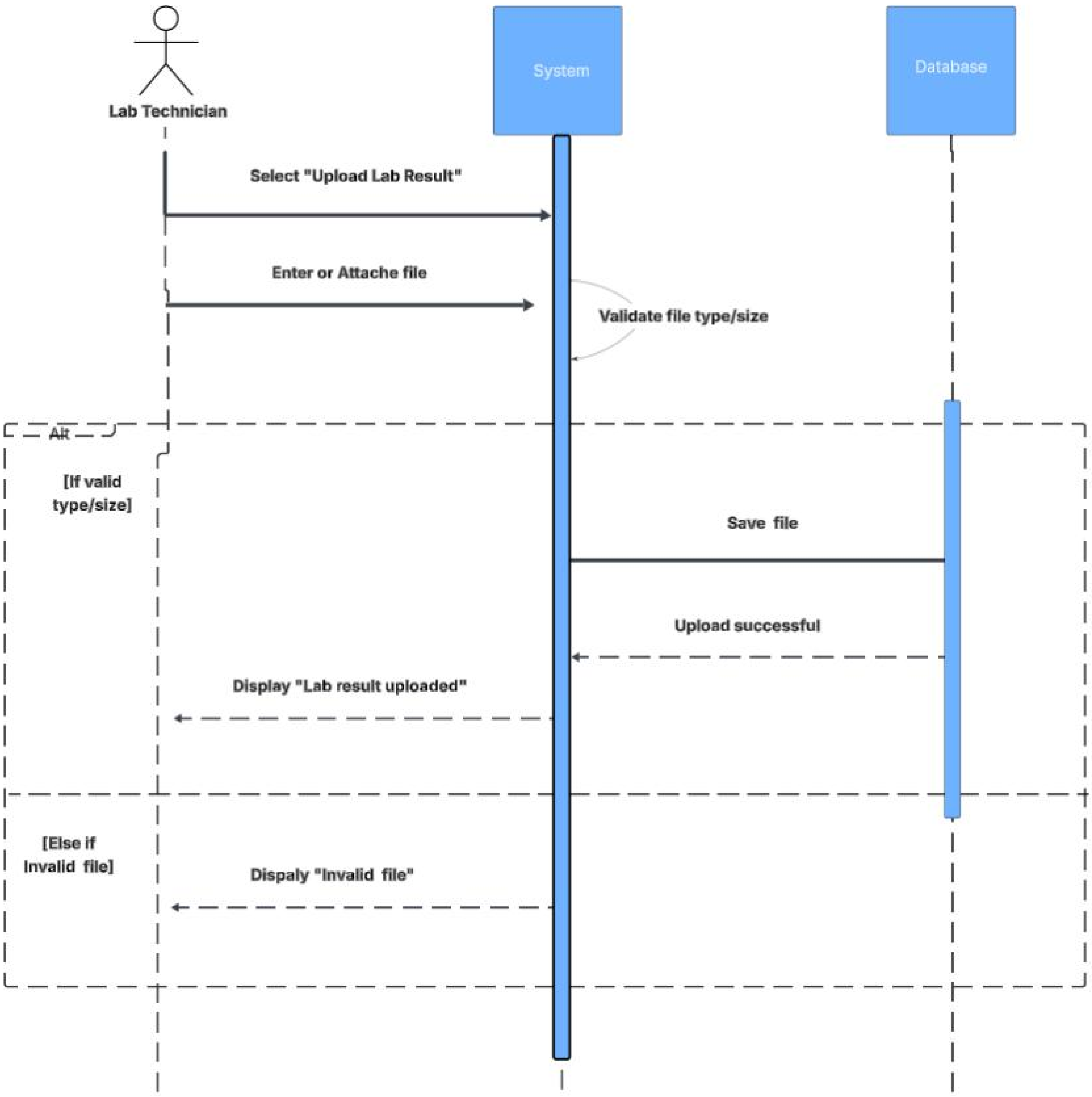


Figure 3-8: Upload Lab Result

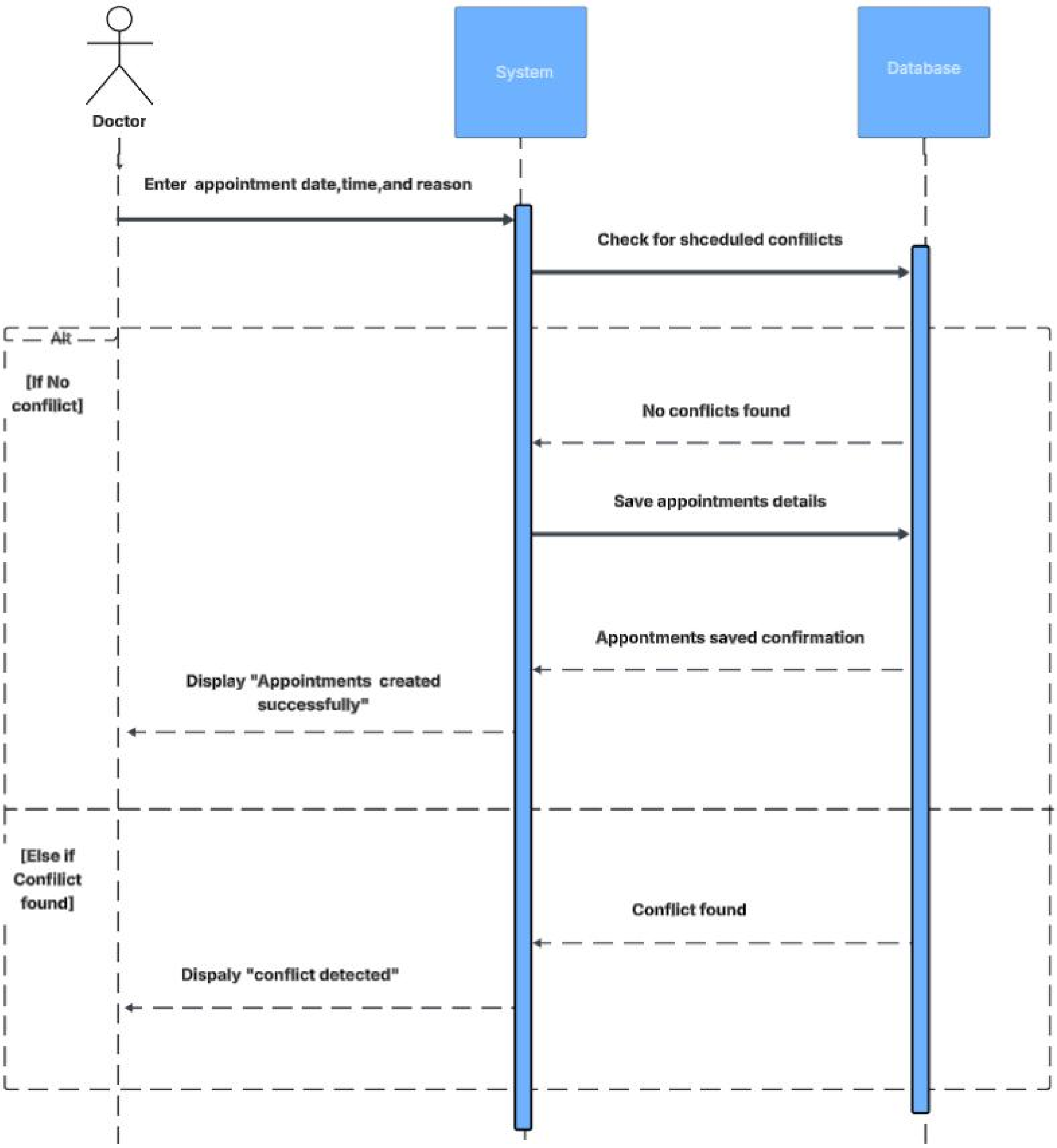

Doctor
System
Database
Enter appointment date,time,and reason
Check for shceduled confilicts
Alt
[If No confilict]
No conflicts found
Save appointments details
Appontments saved confirmation
Display "Appointments created successfully"
[Else if Confilict found]
Conflict found
Dispaly "conflict detected"

Figure 3-9: Create Appointment

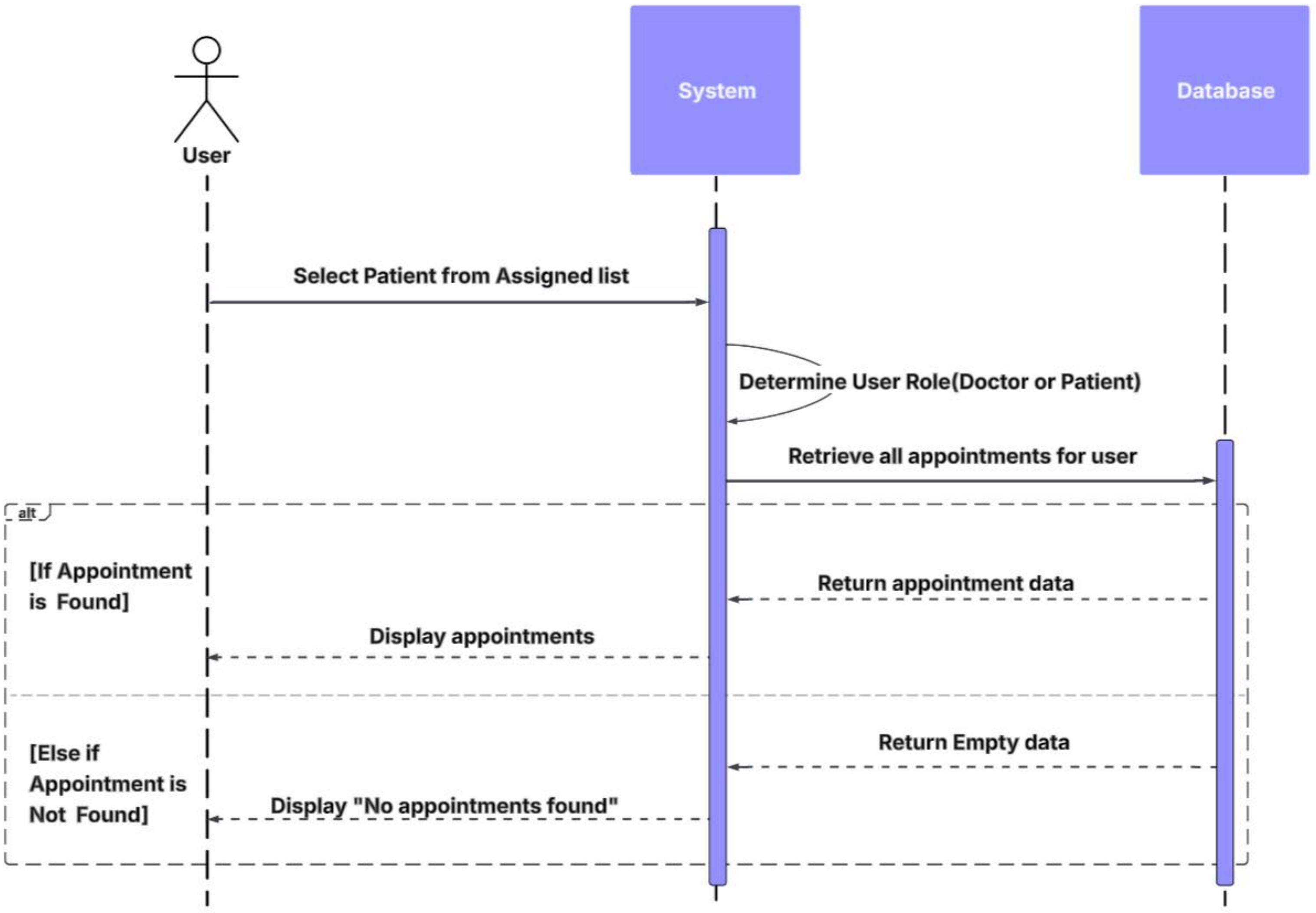
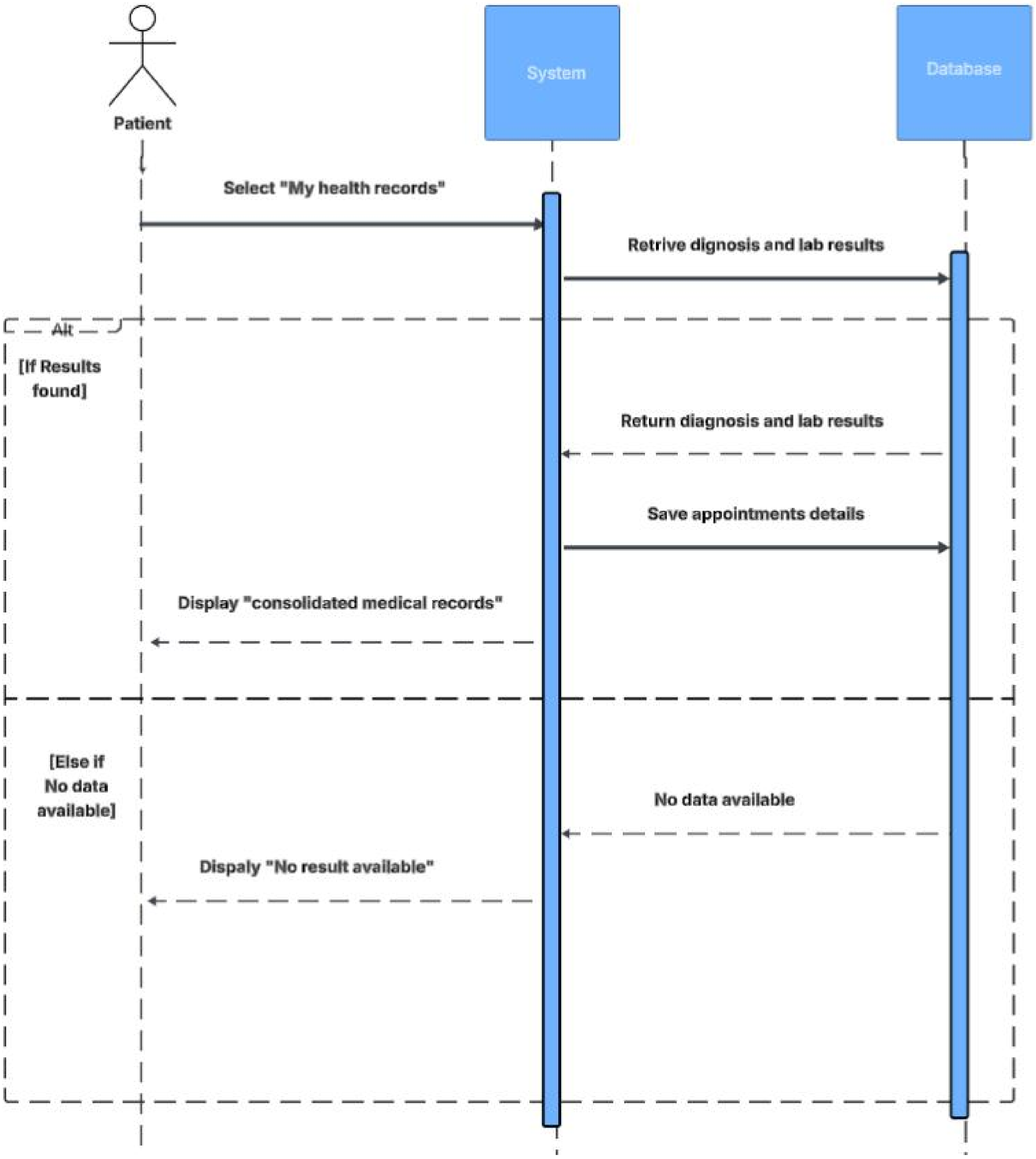

User
System
Database
Select Patient from Assigned list
Determine User Role(Doctor or Patient)
Retrieve all appointments for user
alt
[If Appointment is Found]
Return appointment data
Display appointments
[Else if Appointment is Not Found]
Return Empty data
Display "No appointments found"

Figure 3-10: View Appointments



Patient
System
Database
Select "My health records"
Retrive dignosis and lab results
Alt
[If Results found]
Return diagnosis and lab results
Save appointments details
Display "consolidated medical records"
[Else if No data available]
No data available
Dispaly "No result available"

Figure 3-11: View Lab Results and Diagnosis

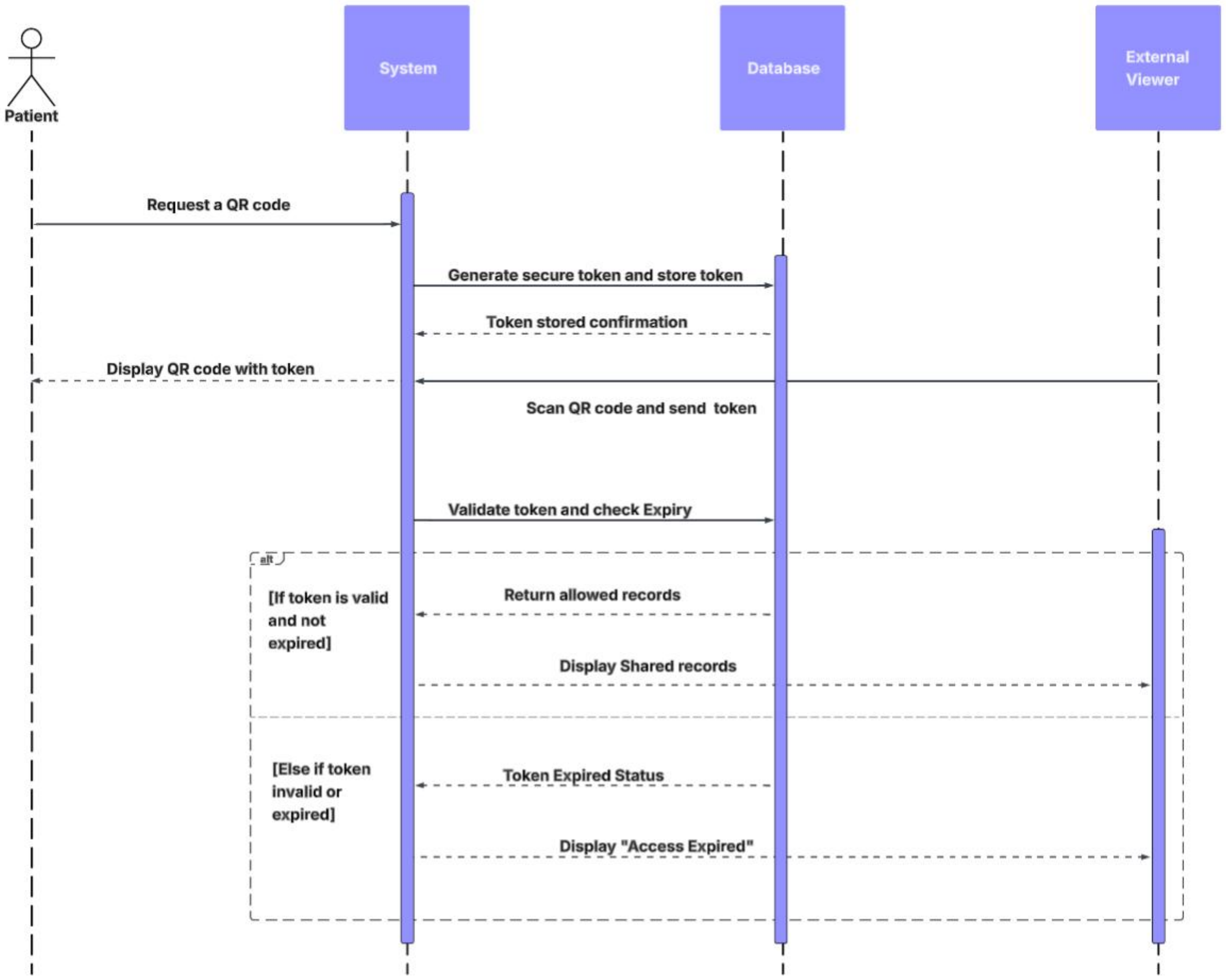

Patient
System
Database
External Viewer
Request a QR code
Generate secure token and store token
Token stored confirmation
Display QR code with token
Scan QR code and send token
Validate token and check Expiry
alt
[If token is valid and not expired]
Return allowed records
Display Shared records
[Else if token invalid or expired]
Token Expired Status
Display "Access Expired"

Figure 3-12: QR-Based Record Sharin

3.3. State chart Diagram

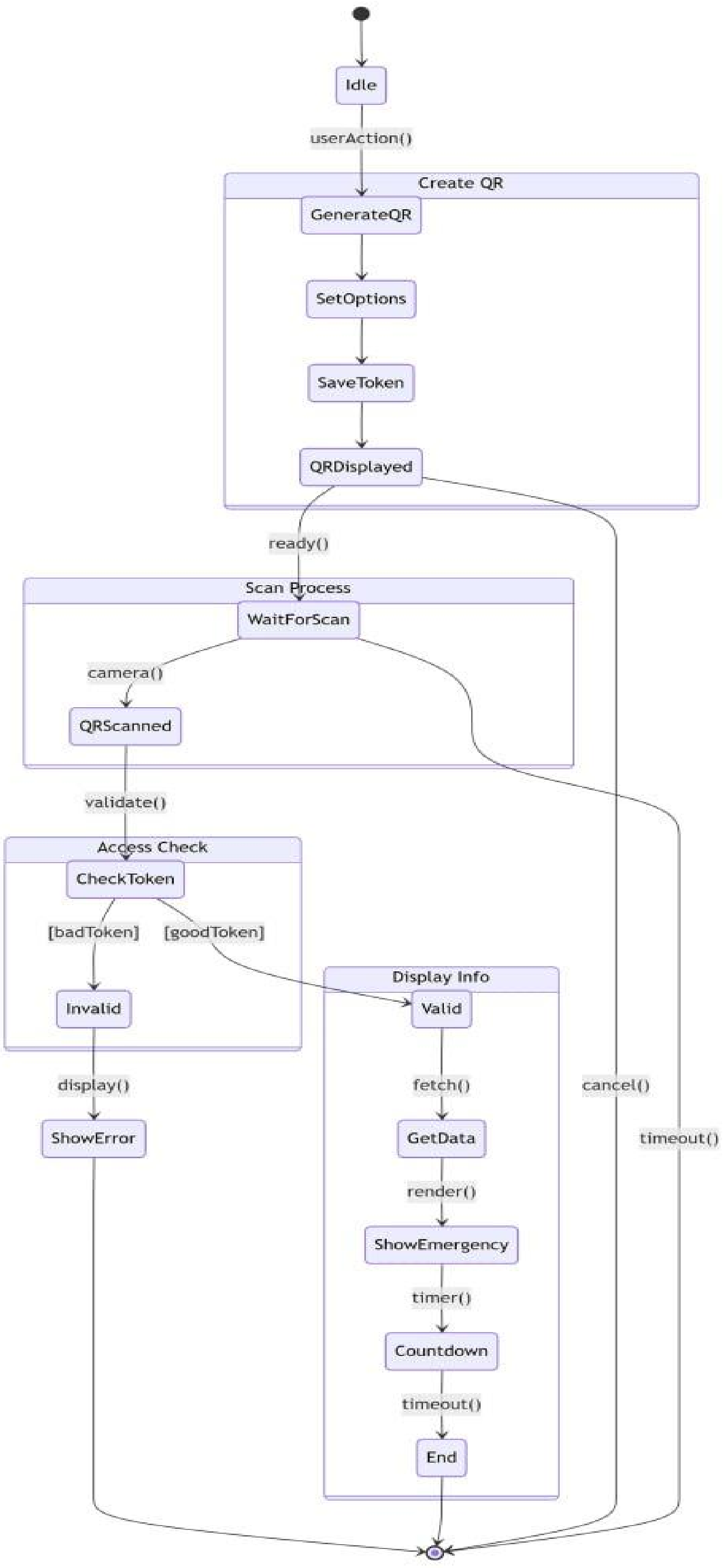


Figure 3-13: Emergeny Chart

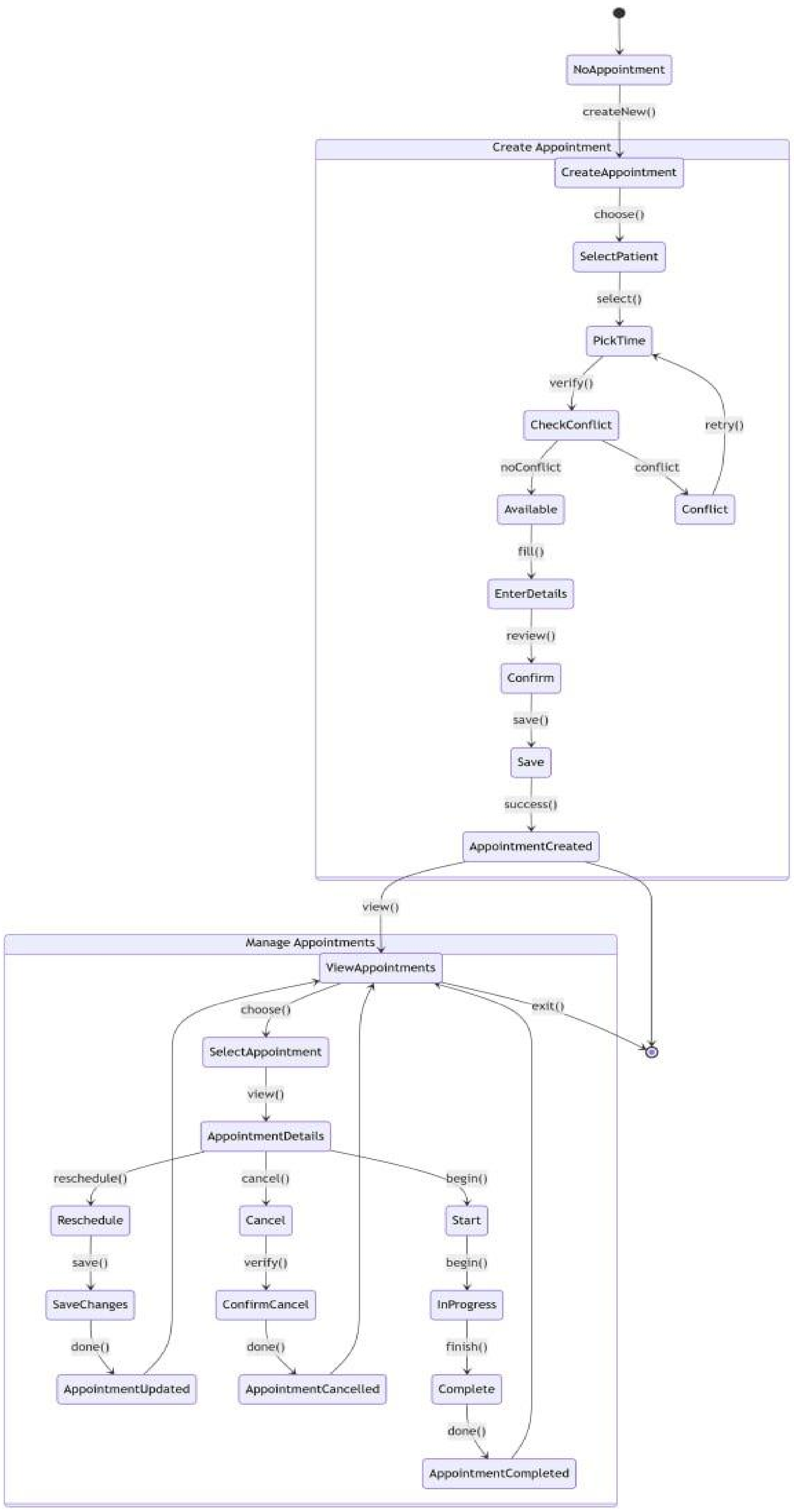


Figure 3-14: Appointments Chart

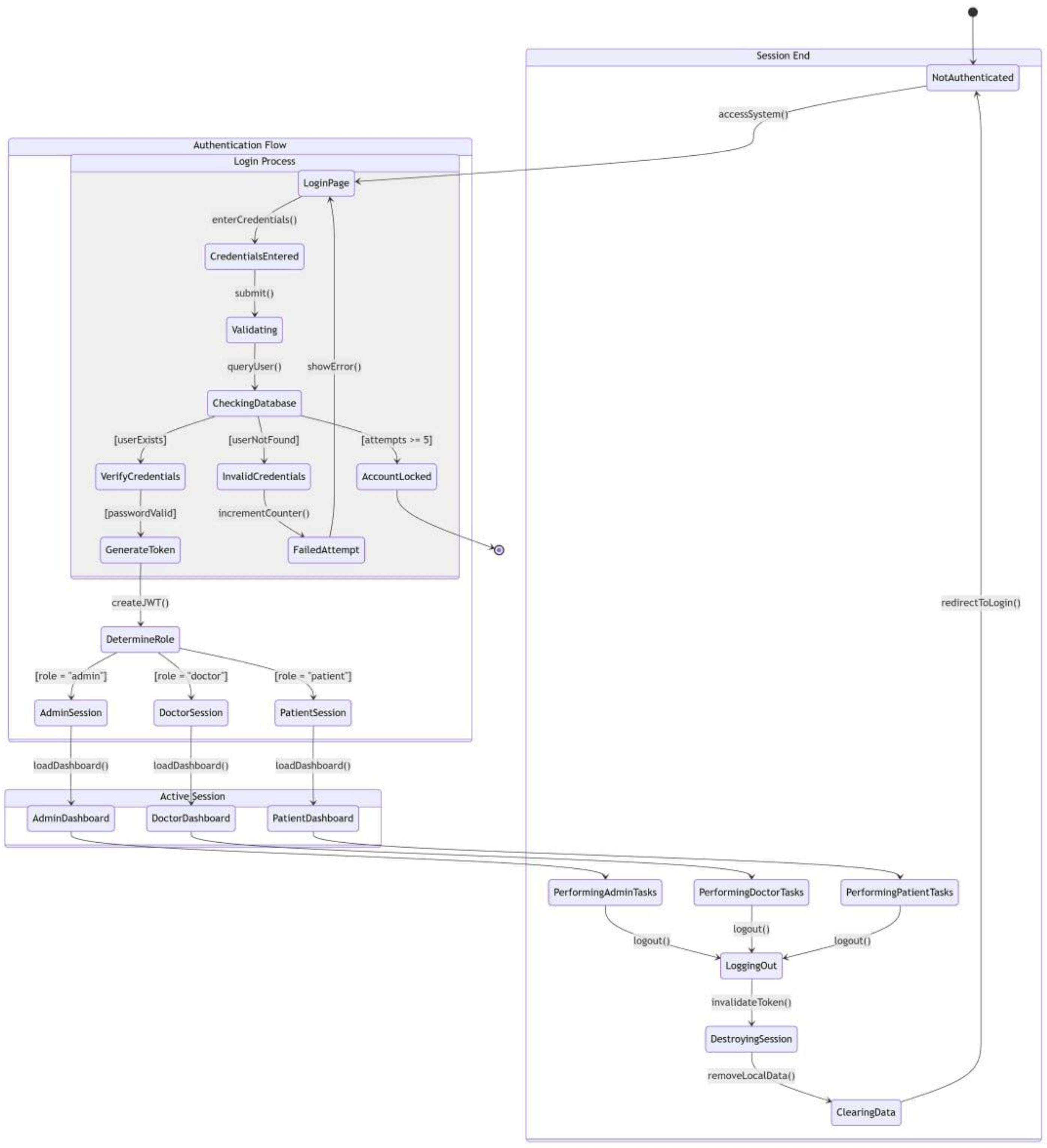


Figure 3-15: User Authentication Chart

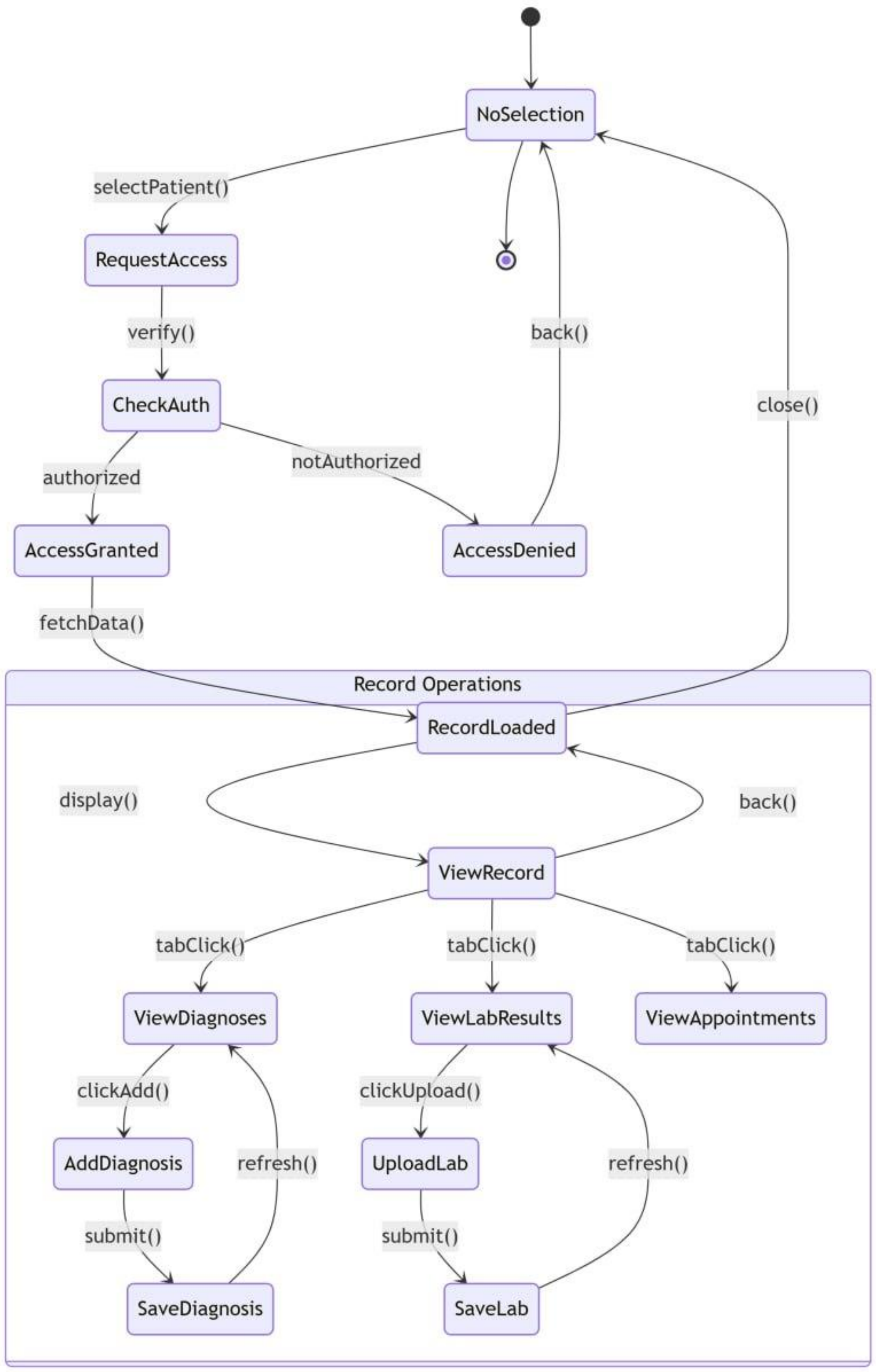


Figure 3-16: Patient Medical Record Chart

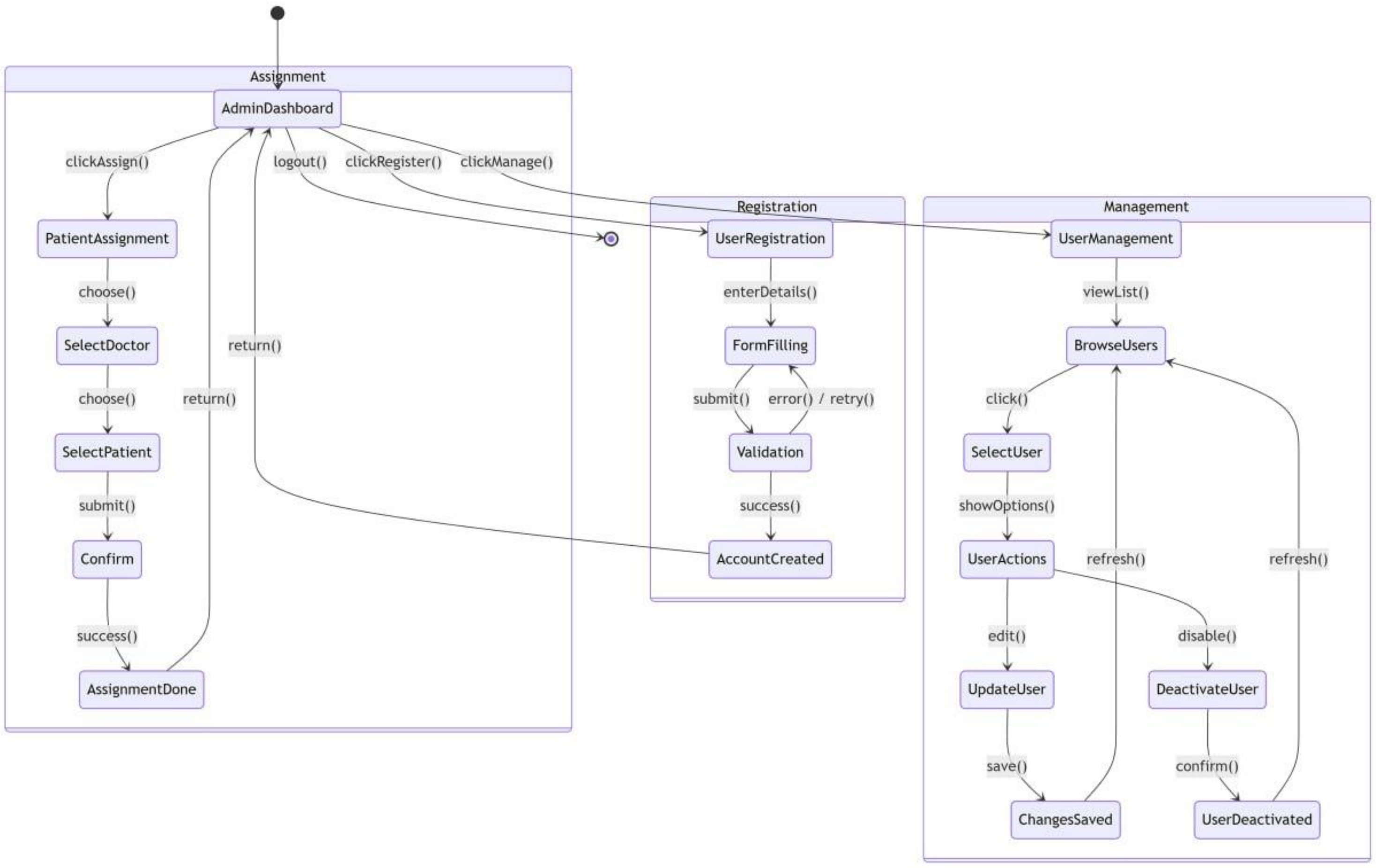


Figure 3-17: Adminstration Management Chart

4. Detailed Design

4.1. User Class (Foundation)

Table 2: Attributes Description for USER Class

Attribute	Type	Visibility	Invariant (Validation Rules)
name	String	Private	name <> NULL. Should conform to standard naming conventions.
password	String	Private	password <> NULL. Stored as a hash (e.g., bcrypt). Must meet complexity requirements (e.g., min. 8 chars).
contactinfo	Integer	Private	Must be a valid phone number format or unique identifier.
created_at	DateTime	Private	Must be less than or equal to the current date/time. Automatically set on creation.

Table 3: Operation Description for USER Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
updateProfile	Public	void	New Profile Data	User is authenticated (logged in).	Non-critical profile information is validated and updated.
login	Public	void	Credentials (name/ password)	Account exists and is active.	User is authenticated; a session token (JWT) is issued based on their Role.
logout	Public	void	--	User is currently authenticated.	User's session token is invalidated, and the user is logged out.

4.2. Admin Class

Table 4: Attributes Description for ADMIN Class

Attribute	Type	Visibility	Invariant (Validation Rules)
Facility	String	Public	Facility <> NULL. Must reference a valid facility within the system configuration.

Table 5: Operation Description for ADMIN Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
registerUser	Public	User Object	User Details (Role, Name, Password)	Admin is authenticated and authorized to create new accounts.	A new User account is created and stored in the database.
assignPatient	Public	Boolea n	PatientID, DoctorID	Patient and Doctor IDs are valid.	An Assignment relationship is created between the specified Patient and Doctor.
manageInsur ance	Public	Insuran ce Object	Provider Details	Admin is authenticated.	An Insurance entry is created, updated, or deleted.

4.3. Doctor Class

Table 6: Attributes Description for DOCTOR Class

Attribute	Type	Visibility	Invariant (Validation Rules)
specialization	String	Public	specialization <> NULL. Must be chosen from a predefined, controlled list.
license No	String	Public	license No <> NULL. Must be unique and validated against an external or internal registrar.

Table 7: Operation Description for DOCTOR Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
addDiagnosis	Public	Diagnosis Object	PatientID, Diagnosis Text, Treatment	Doctor is authenticated. Patient record is accessible.	A new Diagnosis entity is created, linked to the patient, and an audit log is created.
viewAssignedPatient	Public	List of Patients	--	Doctor is authenticated.	Returns a list of all patients linked via the Assignment relationship.
uploadLabResult	Public	LabResult Object	PatientID, File Path, Result Type	Doctor is authenticated. File data is available.	A new LabResult entity is created, and the file is stored in the Data Tier.
createAppointment	Public	Appointment Object	PatientID, Date, Time	Doctor's schedule has no conflict at the specified time.	A new Appointment is scheduled with the initial status set to 'Scheduled'.

4.4. Patient Class

Table 8: Attributes Description for PATIENT Class

Attribute	Type	Visibility	Invariant (Validation Rules)
dob	Date	Public	dob <> NULL. Must be a valid date in the past.
gender	String	Public	Must be chosen from a defined list ('Male', 'Female', 'Other').
address	String	Public	Should be a valid geographical address string.
unique PatientId	String	Public	unique PatientId <> NULL. Must be a unique identifier for the patient, system-generated.

Table 9: Operation Description for PATIENT Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
viewRecords	Public	List of Records	--	User is authenticated as the patient.	A list of all historical records and diagnoses is displayed.
updateEmergencyInfo	Public	Void	New Info Data	Patient is authenticated.	The linked Emergency Info entity is updated with the new details.
reschedule	Public	Void	AppointmentID, New Date/Time	Appointment exists and the new slot is available.	The time and date of the specified Appointment are updated.
downloadQRCode	Public	QRShare Object	--	Patient is authenticated.	The QRShare entity generates a token, and the QR code image is returned to the user.

4.5. Emergency Info Class

Table 10: Attributes Description for EMERGENCY INFO Class

Attribute	Type	Visibility	Invariant (Validation Rules)
emergencyId	Integer	Public	emergencyId <> NULL. Primary Key.
bloodType	String	Public	Must be one of the recognized blood groups (A, B, AB, O, +/-).
allergies	String	Public	List of known substances causing an allergic reaction.
chronic Diseases	String	Public	List of long-term medical conditions.
emergency Contact	String	Public	Must be a valid, reachable phone number or name.

Table 11: Operation Description for EMERGENCY INFO Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
updateEmergencyInfo	Public	Void	All Attributes	User with update privileges (e.g., Patient or Doctor) is logged in.	The data in the Emergency Info entity is validated and saved.

4.6. Appointment Class

Table 12: Attributes Description for APPOINTMENT Class

Attribute	Type	Visibility	Invariant (Validation Rules)
appointmentid	Integer	Public	appointmentid <> NULL. Primary Key.
date	Date	Public	Must be a date in the future.
time	Time	Public	Must be a valid time (e.g., within facility operating hours).
reason	String	Public	Must be provided by the patient or doctor.
status	String	Public	Must be one of: 'Scheduled', 'Completed', 'Canceled', 'Pending'.

Table 13: Operation Description for APPOINTMENT Class

Operations	Visiblity	Return type	Argument	Precondition	Post-condition
checkConflict	Public	Boolean	Date, Time, DoctorID	Doctor's schedule is available.	Returns TRUE if the time slot is free, FALSE otherwise.
cancel	Public	Void	AppointmentID	Appointment exists.	Appointment status is updated to 'Canceled'.
reschedule	Public	Void	AppointmentID, New Date/Time	New Date/Time slot is available (checked via checkConflict).	Appointment date and time are updated.

4.7. Diagnosis Class

Table 14: Attributes Description for DIAGNOSIS Class

Attribute	Type	Visibility	Invariant (Validation Rules)
diagnosisid	Integer	Public	diagnosisid <> NULL. Primary Key.
diagnosisText	String	Public	diagnosisText <> NULL. Textual description of the finding.
treatmentPlan	String	Public	Details of care, medication, or procedure.
date	Date	Public	Must be less than or equal to the current date.

Table 15: Operation Description for DIAGNOSIS Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
addDiagnosis	Public	Diagnosis Object	PatientID, Data	User (Doctor) is authenticated and authorized to create records.	A new Diagnosis entity is persisted and linked to the patient.

4.8. Lab Result Class

Table 16: Attributes Description for LAB RESULT Class

Attribute	Type	Visibility	Invariant (Validation Rules)
labResult	Integer	Public	labResult <> NULL. Primary Key.
filePath	String	Public	Must be a valid, accessible path/URL on the file storage system.
resultType	String	Public	Must be selected from a predefined list of test types.
uploadDate	Date	Public	Must be less than or equal to the current date.

Table 17: Operation Description for LAB RESULT Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
uploadLabResult	Public	LabResult Object	File Data, PatientID, Result Type	User (Doctor or Lab Tech) is authenticated.	The file is stored, and a new LabResult entity is created and linked to the patient.

4.9. QRShare Class

Table 18: Attributes Description for QRShare Class

Attribute	Type	Visibility	Invariant (Validation Rules)
expiretime	DateTime	Private	expiretime <> NULL. Must be in the future (e.g., 5-minute window from generation).
autherized Field	Text	Private	Must be a comma-separated list of sensitive fields accessible with the token.

Table 19: Operation Description for QRShare Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
generateQR	Private	String (Token)	PatientID	Patient is authenticated and requests a QR code download.	A unique, time-limited token is created, persisted, and encoded into a QR image for display.
validateToken	Private	Boolean	Token String	A First Responder or authorized scanner provides the token.	Checks if the token is valid, not expired, and matches an active record. Returns TRUE or FALSE.

4.10. Assignment Class

Table 20: Attributes Description for ASSIGNMENT Class

Attribute	Type	Visibility	Invariant (Validation Rules)
assignmentId	Integer	Public	assignmentId <> NULL. Primary Key.
assignedDate	Date	Public	Must be less than or equal to the current date. Automatically set upon creation.
DoctorID (FK)	Integer	Protected	Must reference a valid UserID with the 'Doctor' role.
PatientID (FK)	Integer	Protected	Must reference a valid UserID with the 'Patient' role.

Table 21: Operation Description for ASSIGNMENT Class

Operations	Visibility	Return type	Argument	Precondition	Post-condition
assign	Public	Assignment Object	DoctorID, PatientID	Both IDs are valid and not already linked by an active assignment.	A new Assignment entity is created, linking the Doctor and Patient.
terminate	Public	Void	AssignmentID	Assignment exists.	The Assignment entity is marked as inactive or deleted, ending the relationship.

Reference

1. **Software Requirements Specification (SRS) for HealthNet**
   Authors: HealthNet Team(Group-4)
   Advisor: Mrs. Nuniyat Kifle
   Date: December 2025
   Version: 1.0
2. **Ethiopian Ministry of Health – National eHealth Strategy 2025**
   Publisher: Federal Democratic Republic of Ethiopia, Ministry of Health
   Year: 2022
3. **Role-Based Access Control (RBAC) – NIST Standard**
   Publisher: National Institute of Standards and Technology (NIST)
   Document: NIST Special Publication 800-207
   Year: 2020
4. **Sommerville, Ian. Software Engineering, 9th Edition**
   Publisher: Pearson Education
   Year: 2011
5. **Lucidchart – Diagramming Software**
   Publisher: Lucid Software Inc.
   Used for: Creating all UML diagrams (Class, Sequence, State, Deployment)
   Website: https://www.lucidchart.com