# ***Healix Medical Assistant System***

**Team Guide**
Eng. Sarah Abdelmoaty

**Team Members**

Shahnda Emad Ali Ibrahim

Shahd Abdelmonhem Mostafa Ahmed

Shahd Mohamed Mohamed Alsawii

Rana Mohamed Farouk Abdelkader

Yara Ayman Abdelwahed Elfayoumy

**Specialization name: Microsoft Machine Learning**
**Department: DEPI Program**

**Table of Contents**

## 3. Testing

### 3.1 Introduction
3.1.1 Purpose
3.1.2 Scope
3.1.3 Objectives

### 3.2 Test Plan
3.2.1 Test Strategy
3.2.2 Test Environment
3.2.3 Test Schedule

### 3.3 Functional testing
3.3.1 Test Cases
3.3.2 Test Scenarios
3.3.3 Test Data
3.3.4 Test Execution

### 3.4 Performance Testing
3.4.1 Load Testing
3.4.2 Stress Testing
3.4.3 Performance Metrics

### 3.5 Security Testing
3.5.1 Authentication and Authorization
3.5.2 Data Encryption
3.5.3 Access Control

### 3.6 Usability Testing
3.6.1 User Interface
3.6.2 Workflow Testing
3.6.3 Accessibility

### 3.7 Defect Management
3.7.1 Defect Reporting
3.7.2 Regression Testing

3.8 **Conclusion**

## 4 Reference

# *1. Introduction*

## 1.1 Purpose:

Healix is an AI-powered assistant designed to streamline patient intake, recommend appropriate doctors, and generate preliminary diagnostic reports for physicians. It improves hospital efficiency and reduces unnecessary visits.

## 1.2 Scope

Healix supports:

- Symptom intake via chatbot
- Medical entity extraction using NLP
- Doctor recommendation engine
- Appointment booking
- Preliminary report generation
- Lab/imaging result management It does not provide final diagnoses or expose model outputs directly to patients.

## 1.3 Technologies to be used:

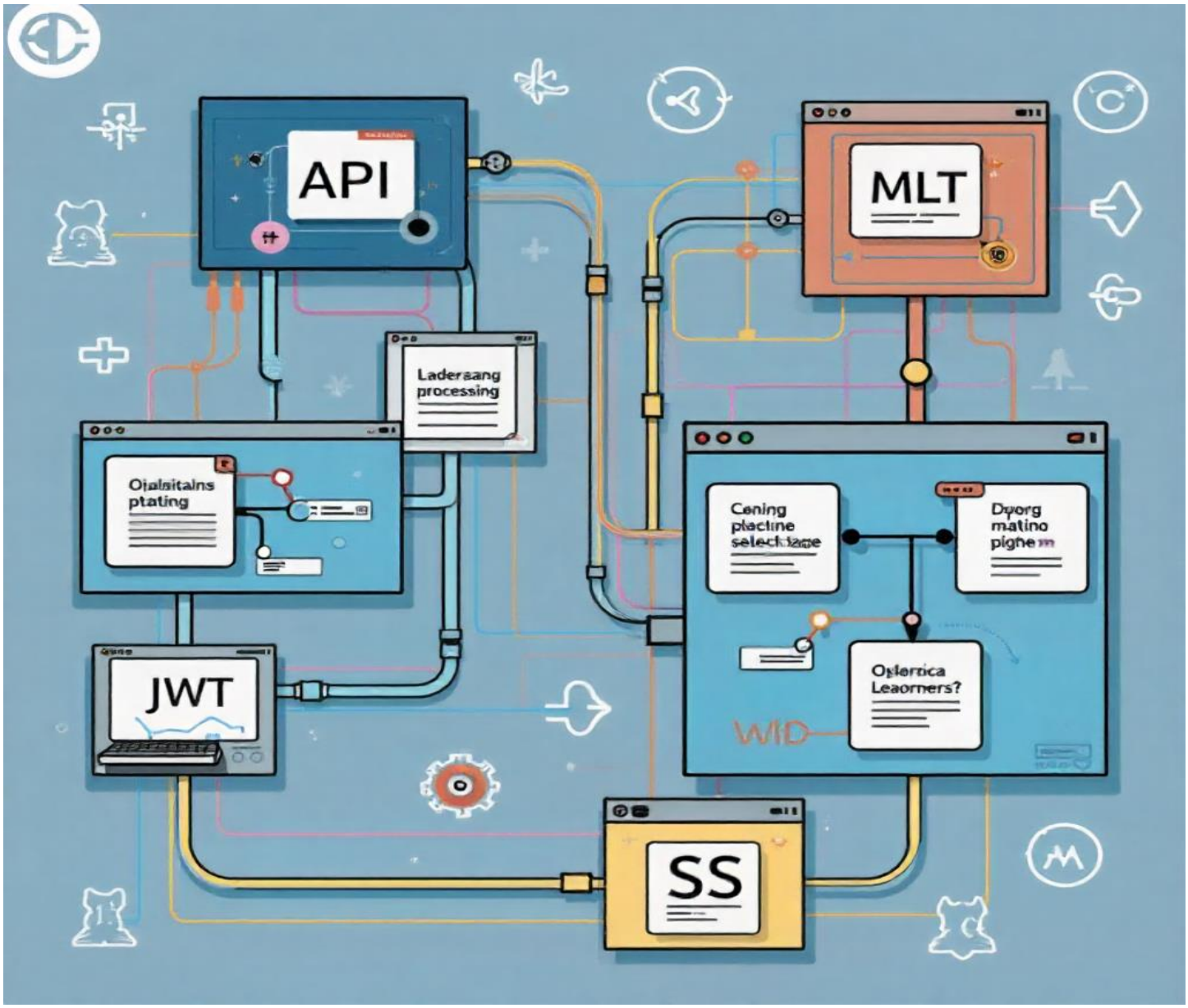
This section outlines the core technologies, frameworks, and tools selected for the development and deployment of the Healix Medical Assistant system.

- **Backend:** FastAPI, Python, PostgreSQL, MongoDB
- **Frontend:** Streamlit, Flutter
- **ML/NLP:** Rasa, scispaCy, ClinicalBERT, PyTorch
- **Security:** JWT, HTTPS, audit logs
- **DevOps:** Docker, GitHub, Render/Railway
- **Testing:** Pytest, Jupyter, ReportLab

## 1.4 Definitions,Acroynyms and Abbreviations

- NLP: Natural Language Processing
- ML: Machine Learning
- JWT: JSON Web Token
- ERD: Entity Relationship Diagram
- MVP: Minimum Viable Product

- API: Application Programming Interface



## 2. Overall Description

### 2.1 Product Perspective

Healix is a standalone system deployed within hospitals. It interacts with patients, doctors, and admins via secure web/mobile interfaces.

### 2.2 *User Characteristics and classes*

- **Patients:** Submit symptoms, view appointments
- **Doctors:** Review reports, finalize diagnosis
- **Admins:** Manage users, monitor system logs Users are expected to be comfortable with mobile/web apps.

### 2.3 Operating Environment

Healix runs on modern browsers across desktops, tablets, and smartphones. It is platform-agnostic and cloud-deployable

### *2.4 User Documentation*

Includes onboarding guides, user manuals, and helpdesk support.

### 2.5 Software Interfaces

- FastAPI endpoints
- PostgreSQL and MongoDB databases
- Rasa chatbot engine
- ClinicalBERT inference engine

## 2.6 Hardware Interfaces

No specialized hardware required beyond standard computing devices.

## 2.7 External Forces

- Legal compliance with GDPR and healthcare regulations
- Ethical handling of AI-generated suggestions
- Patient consent for data processing

### 2.7.1 Dependencies

- Internet access
- Pre-trained medical models
- Hospital staff for integration and oversight

### 2.7.2 Constraints

- No direct diagnosis to patients
- Reports accessible only to assigned doctors
- Secure role-based access control

# *3. System features*

## 3.1 Functional Requirements

- Patient registration and login
- Symptom collection via chatbot
- NLP entity extraction
- Doctor recommendation engine
- Appointment booking
- Preliminary report generation
- Lab result upload and management
- Final diagnosis and prescription entry by doctors

## 3.2 ***Non Functional Requirements***

### Performance

Response Time :- The system shall give responses in 1 second after checking the patients information.

Capacity:- The System must support 1000 people at a time.

User-interface :-The user-interface screen shall respond within 5 seconds.

Conformity:-The systems must conform to the Microsoft Accessibility

### Security

Patient Identification:- The system requires the patient to identify himself/herself using PHN

Login ID:- Any user who uses the system shall have a Login ID and Password**.**

Modification Any modification (inert, delete, update) for the Database shall be synchronized and only by the administrator in the ward.

Front Desk staff Rights:- Front Desk staff shall be able to view all information in Healix, add new patients to Healix but shall not be able to modify any information in it.

Administrators' Rights:- Administrators shall be able to view and modify all information in HPIMS.

## Reliability

How general the form generation language is Simplicity vs. functionality of the form language= Speeds up form development but does not limit functional.

## Availability

The system shall be available all the time.

## Safety

Humans are error-prone, but the negative effects of common errors should be limited. E.g.,users should realize that a given command will delete data, and be asked to confirm their intent or have the option to undo.

## Software Quality

Good quality of the framework-produces robust, bug free software which necessary requirements Customer satisfaction.

## Reusability

Is part of the code going to be used elsewhere- produces simple and independent code modules that can be reused

## Maintainability

Back Up The system shall provide the capability to back-up the Data.

Errors The system shall keep a log of all the errors.

## 4 Commitment to delivering a high-quality buisness website

- Our commitment to delivering a high quality hospital management system is unwavering.we prioritize performance,security, And reliability to create a digital environment that ot only meets but exceeds the expectations of all stakeholders,including Doctors,Patients families and staffs.

## 5 How testing contributes to user satisfaction and operational efficiency

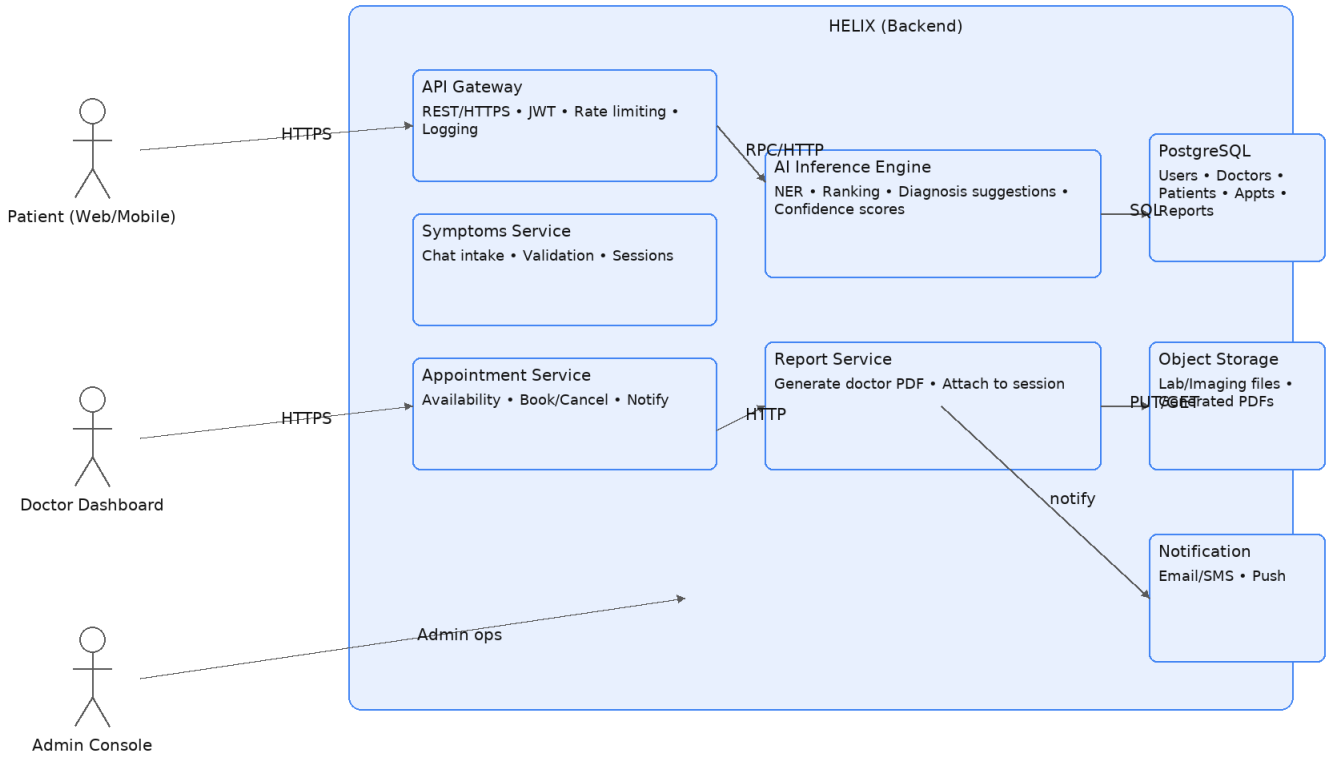
- Thorough testing enhances user satisfaction by ensuring the website's functionality,performance and security.it also promotes operational efficiency by preventing downtime,data loss and security breaches.

# Healix Medical Assistant System

## 1 System Design Architecture

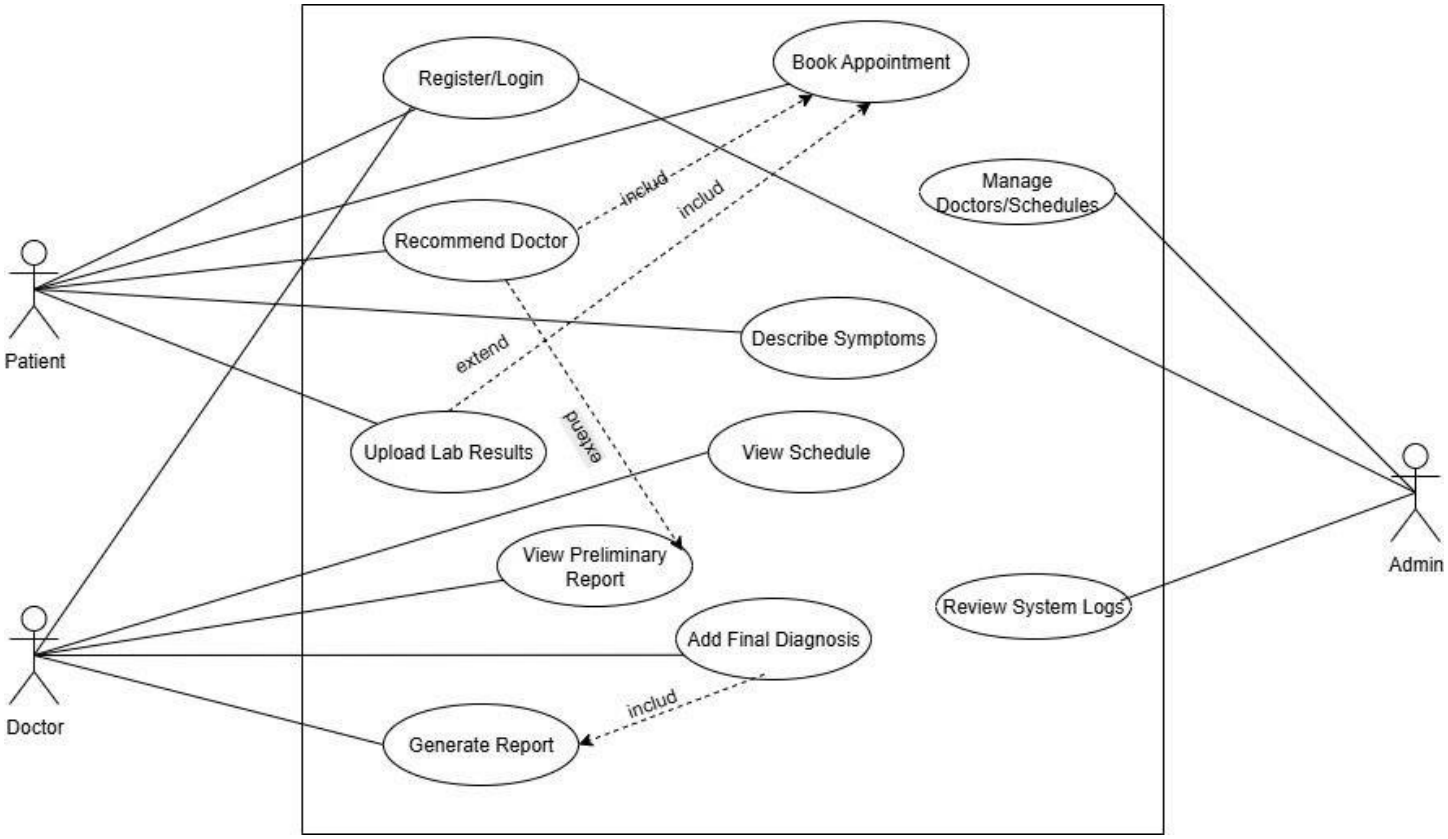
**Helix — Architecture / Context**

Actors, services, and data layer

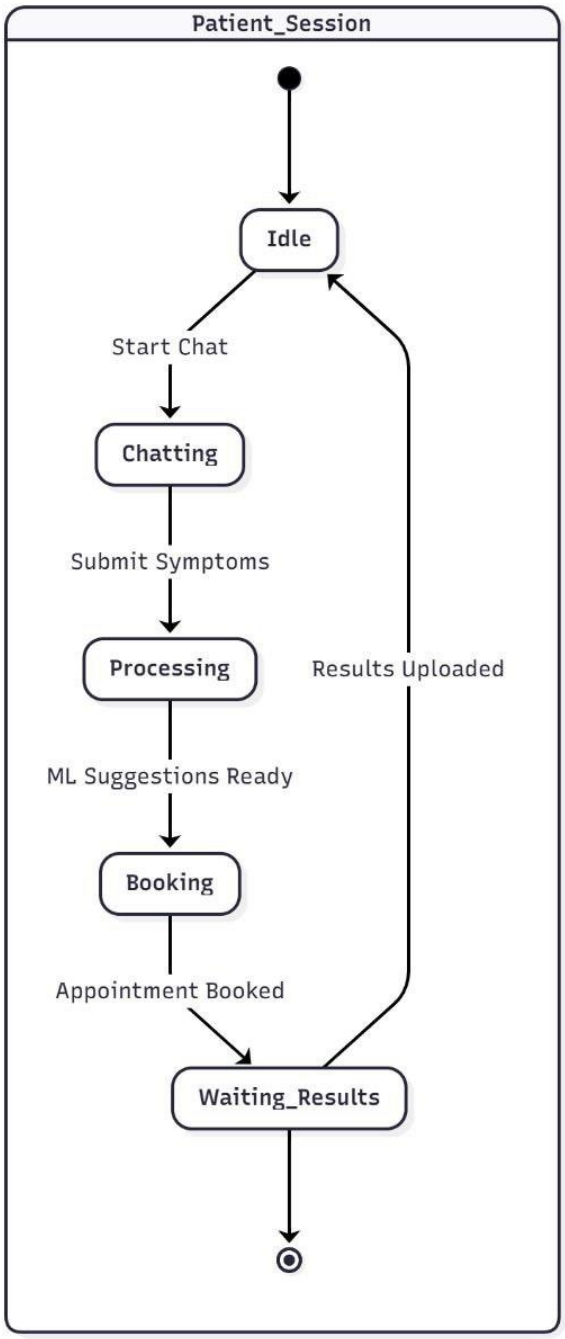


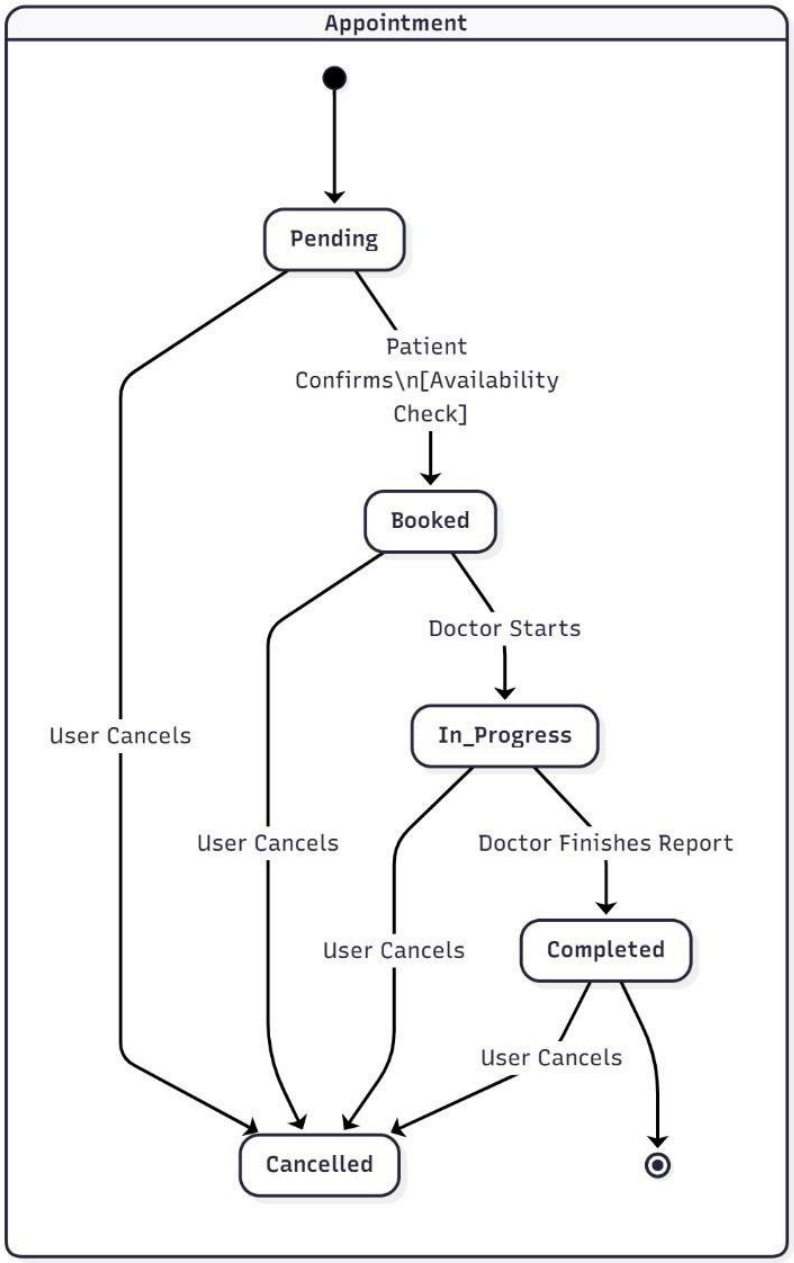

---

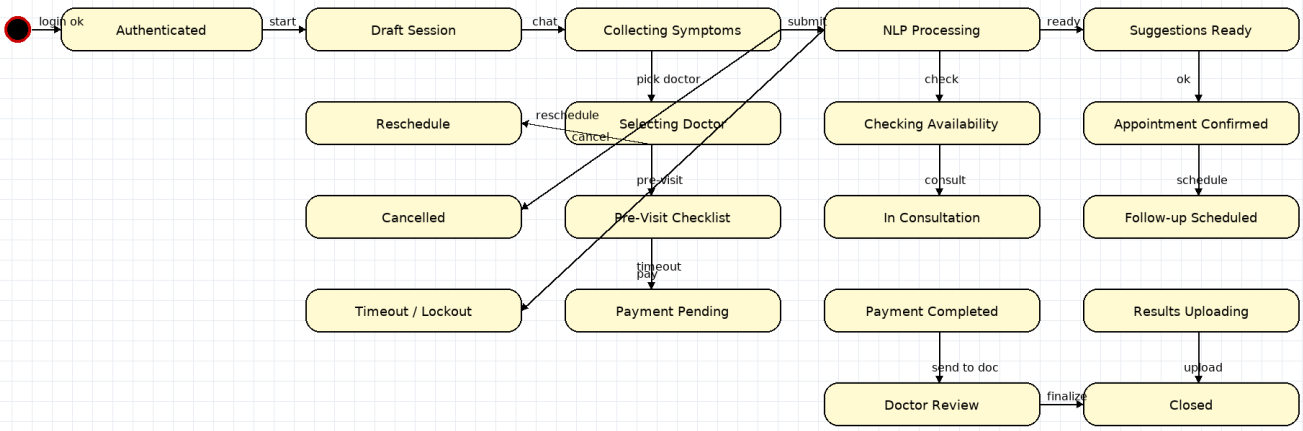
### 1.1 Use case diagram


Register/Login
Book Appointment
Manage Doctors/Schedules
Recommend Doctor
Describe Symptoms
Upload Lab Results
View Schedule
View Preliminary Report
Review System Logs
Add Final Diagnosis
Generate Report
Patient
Doctor
Admin
extend
extend
extend
extend

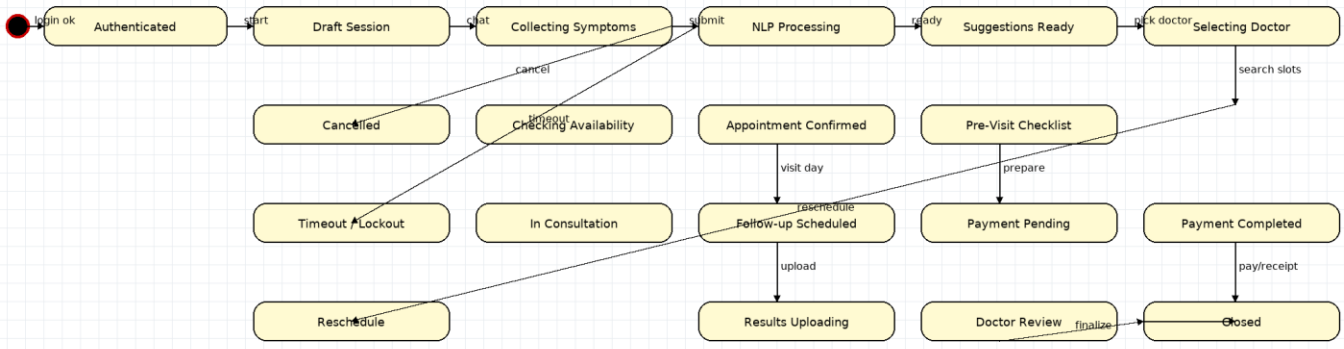
## 1.2 State Diagram



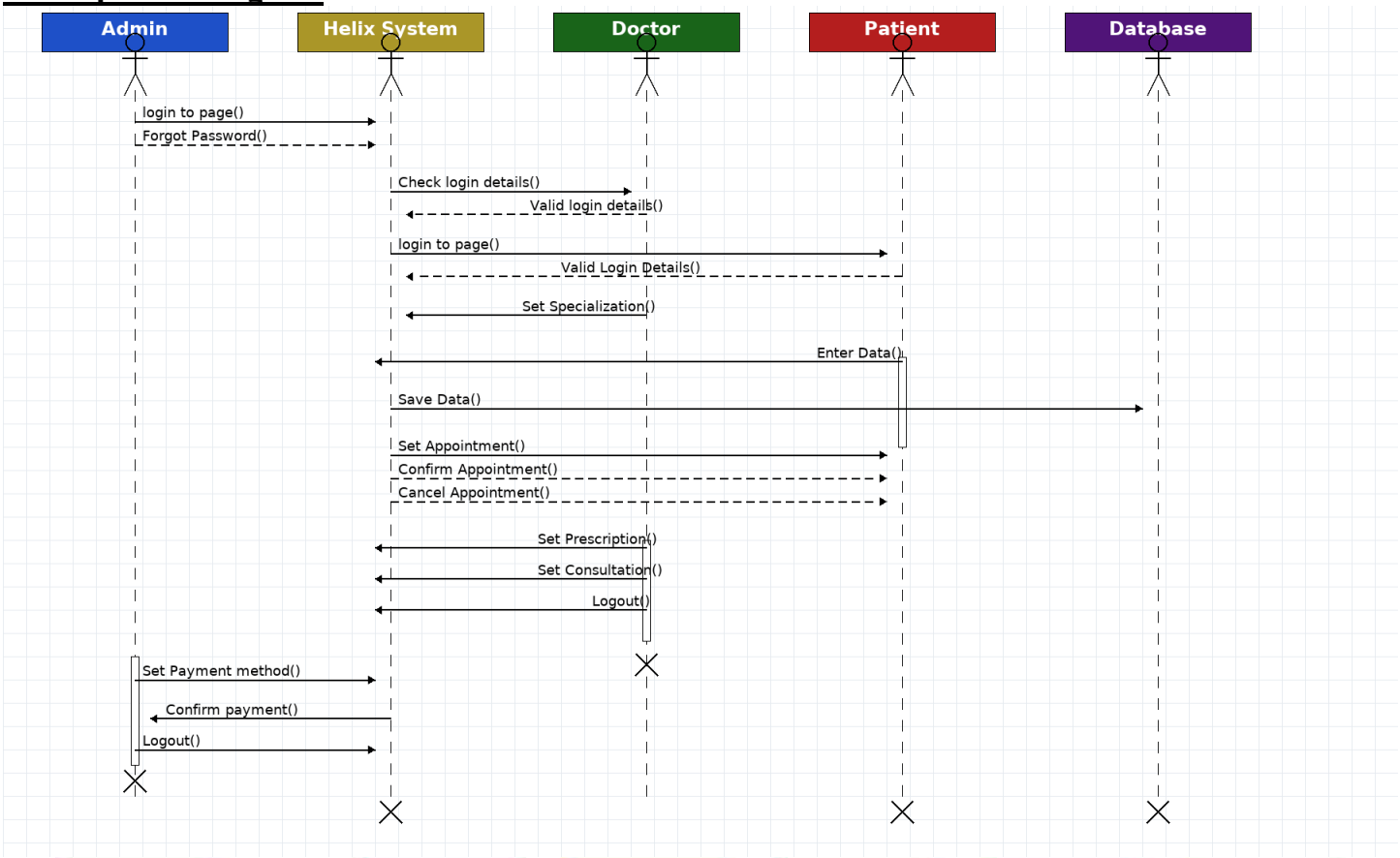



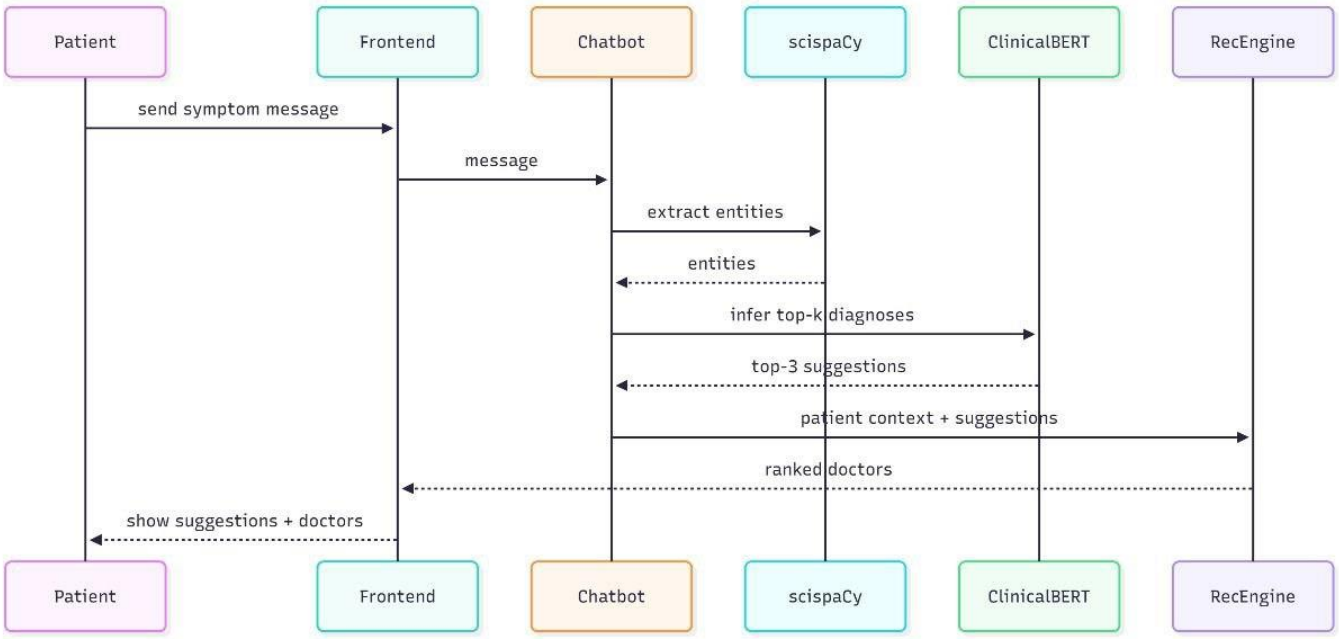


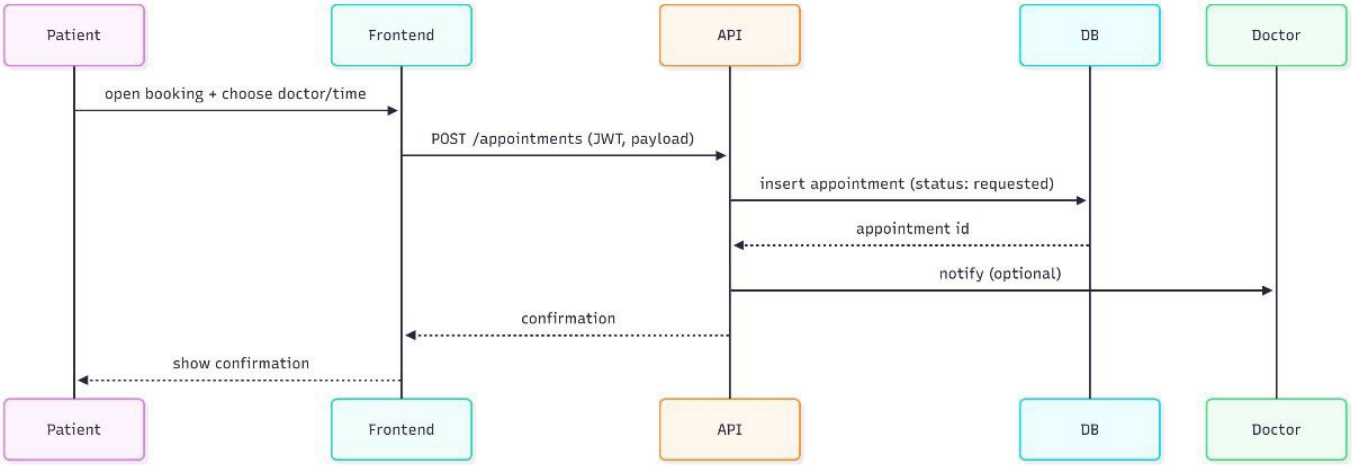

## HELIX — State Diagram (Patient Session)



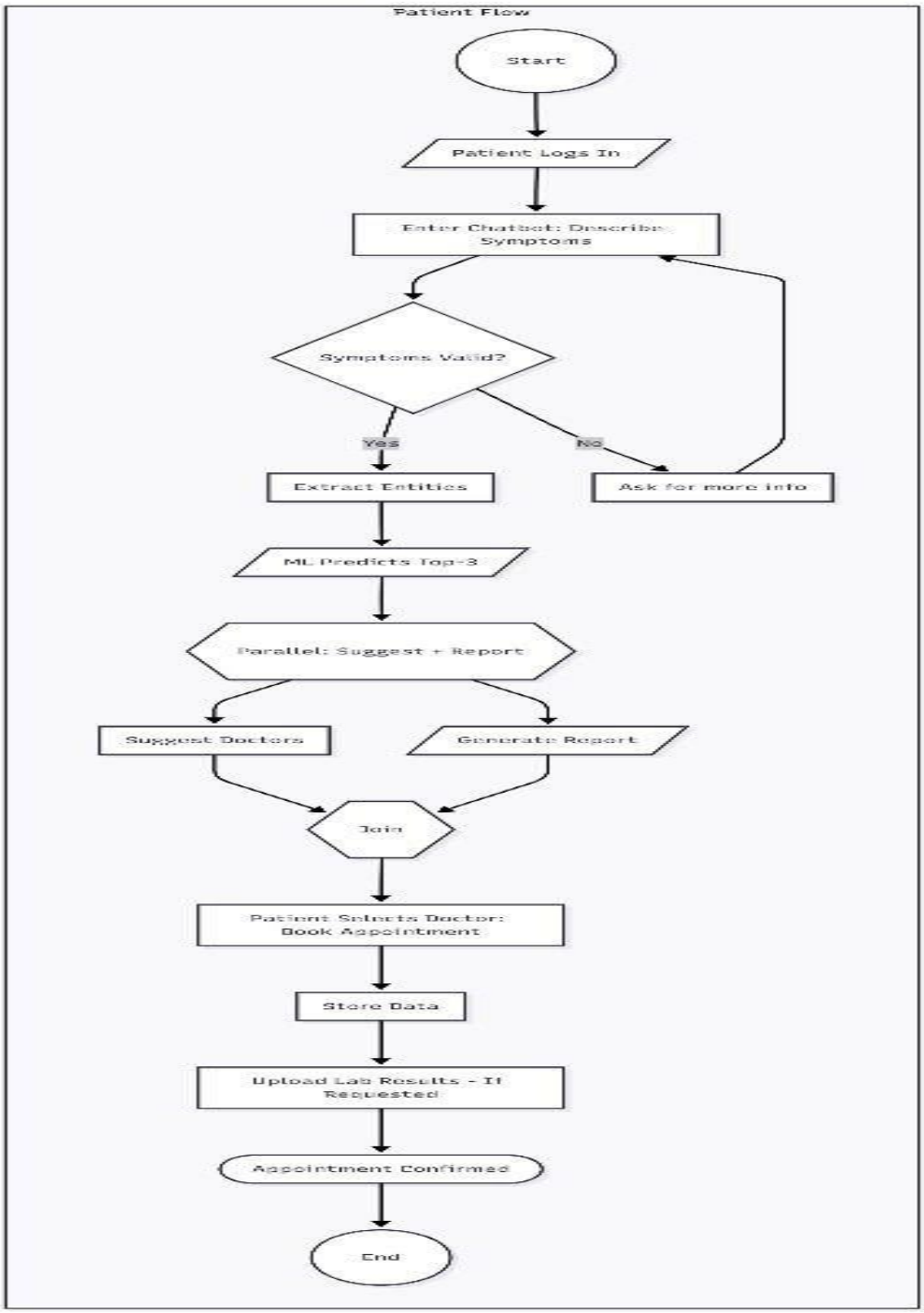
## 1.3 Sequence Diagram

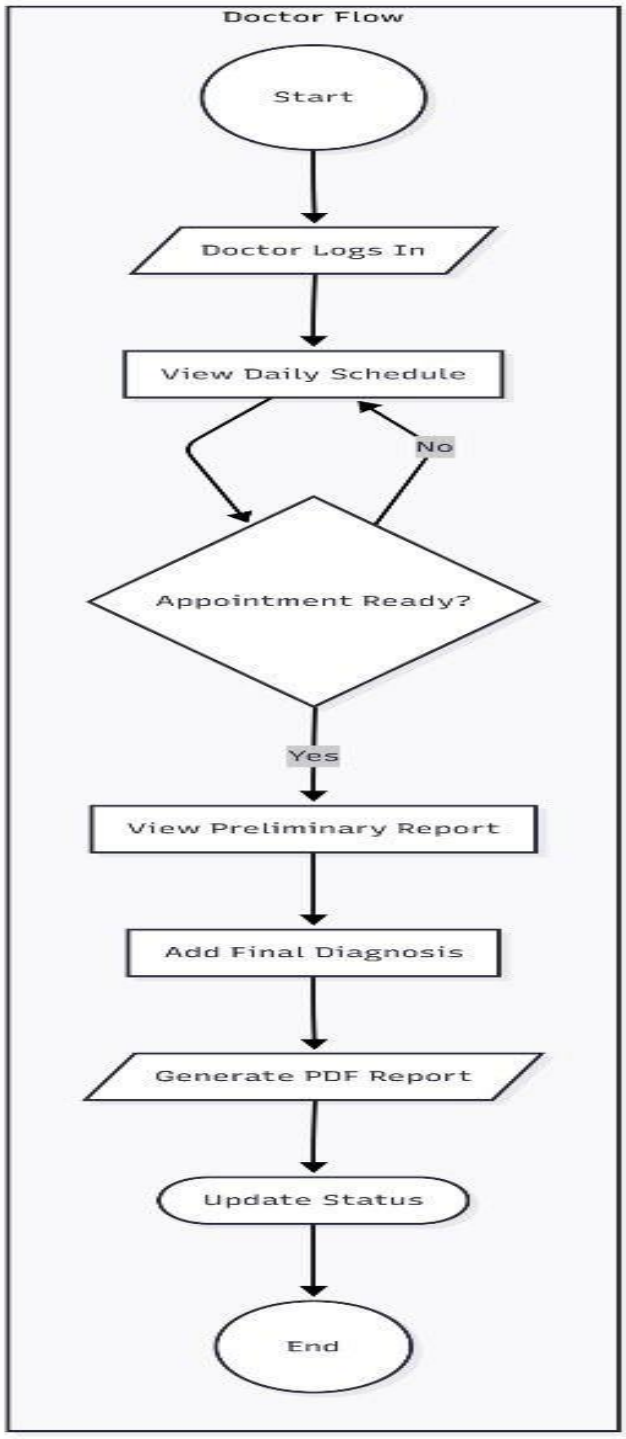


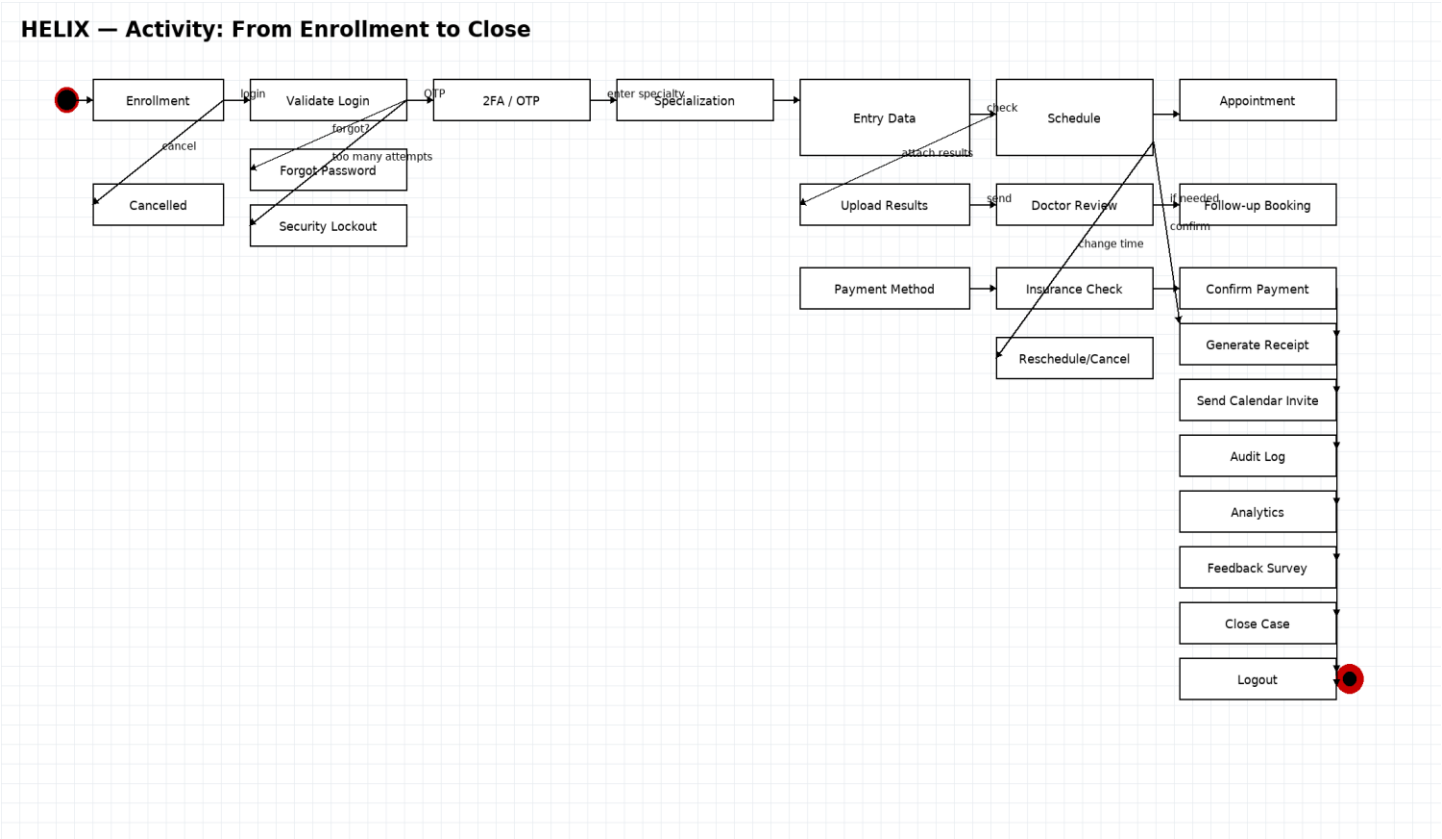






## 1.4 Activity Diagram


Patient Flow
Start
Patient Logs In
Enter Chatbot: Describe Symptoms
Symptoms Valid?
Yes
No
Extract Entities
Ask for more info
ML Predicts Top-3
Parallel: Suggest + Report
Suggest Doctors
Generate Report
Join
Patient Selects Doctor: Book Appointment
Store Data
Upload Lab Results - If Requested
Appointment Confirmed
End

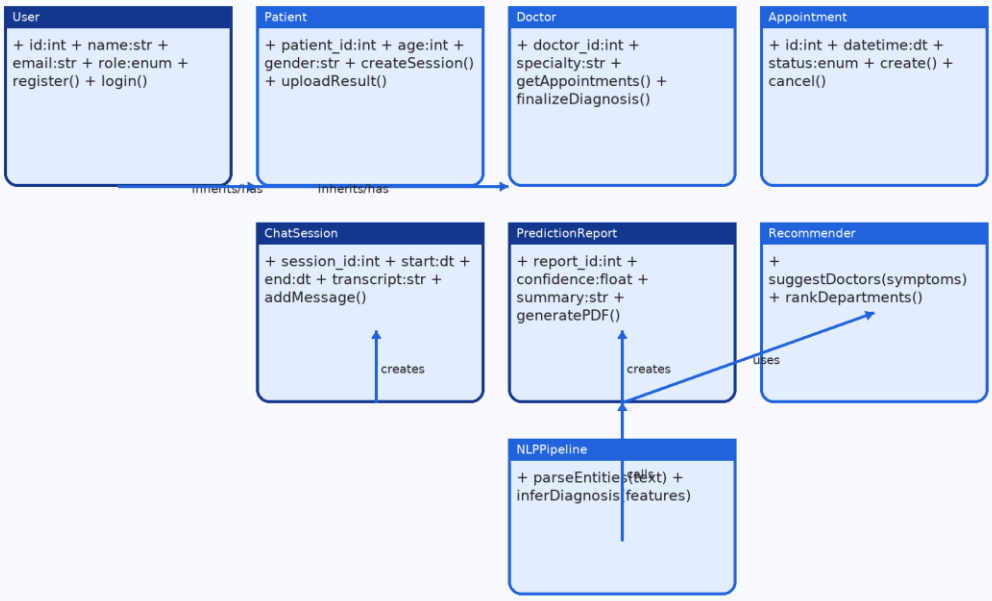

Doctor Flow
Start
Doctor Logs In
View Daily Schedule
No
Appointment Ready?
Yes
View Preliminary Report
Add Final Diagnosis
Generate PDF Report
Update Status
End



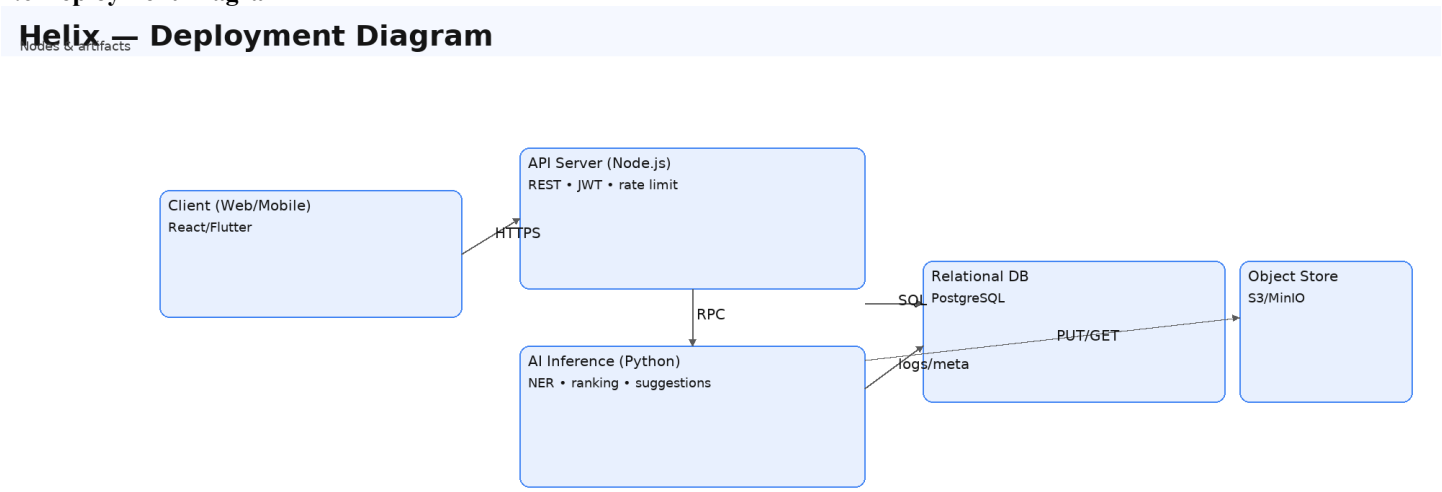

## 1.5 Class Diagram

# HELIX — Class Diagram

Core classes, attributes, and methods



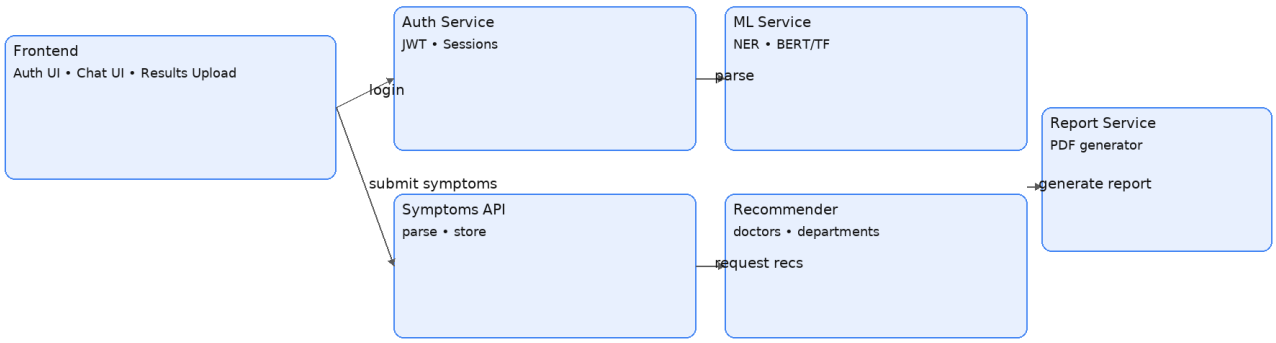
**1.6 Deployment Diagram**



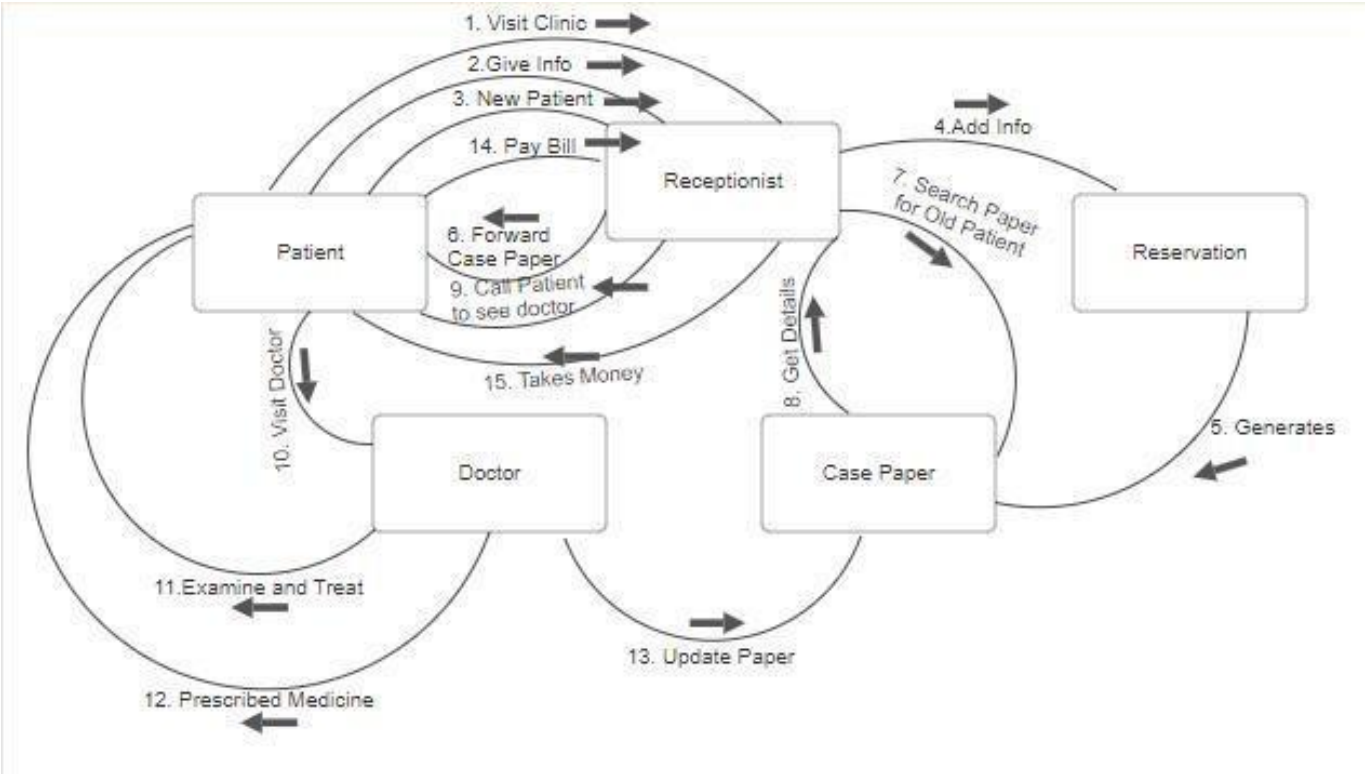

1.8 Component Diagram

# Helix — Component Diagram

Modules and interfaces

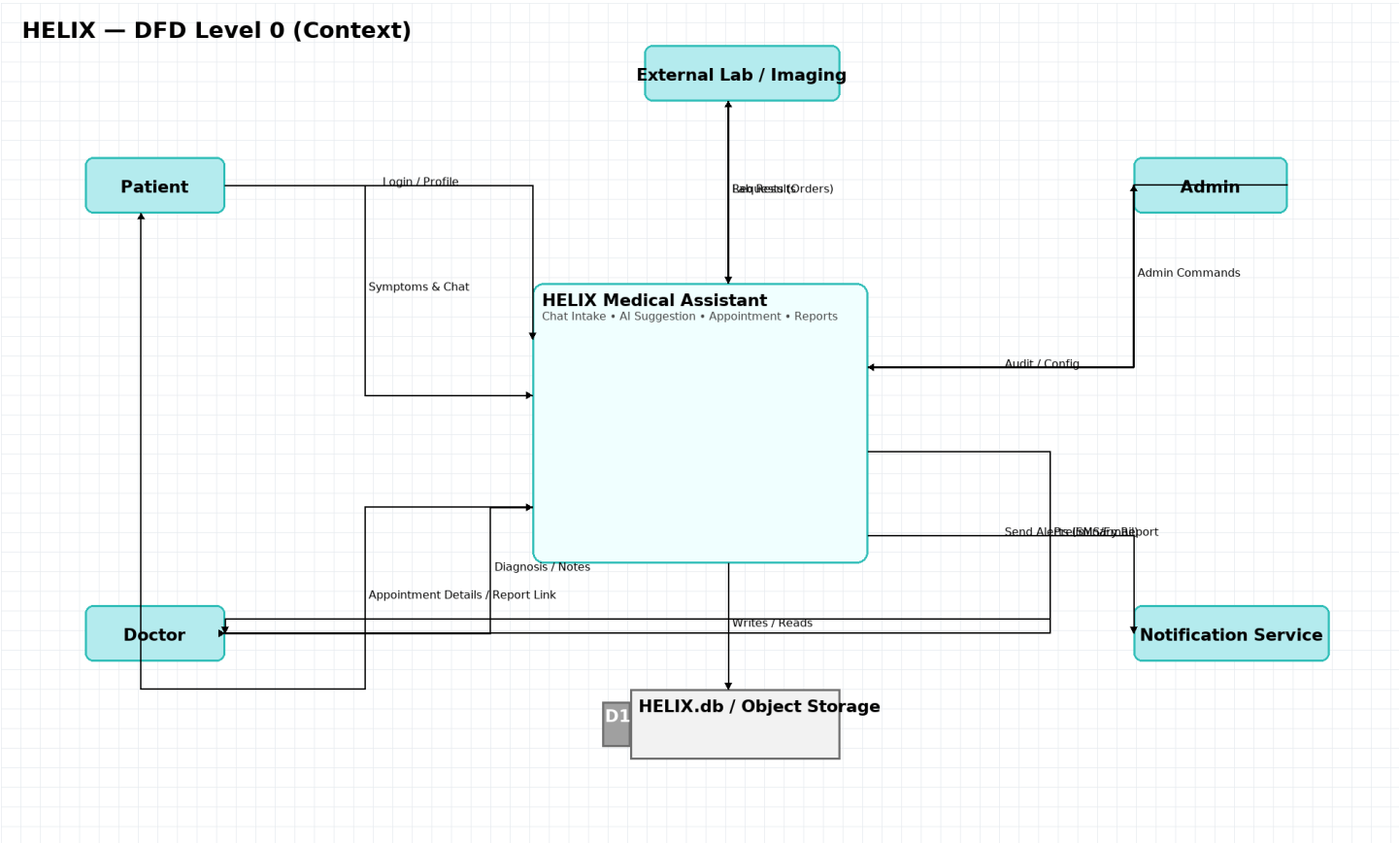


## 1.9 Collaboration Diagram

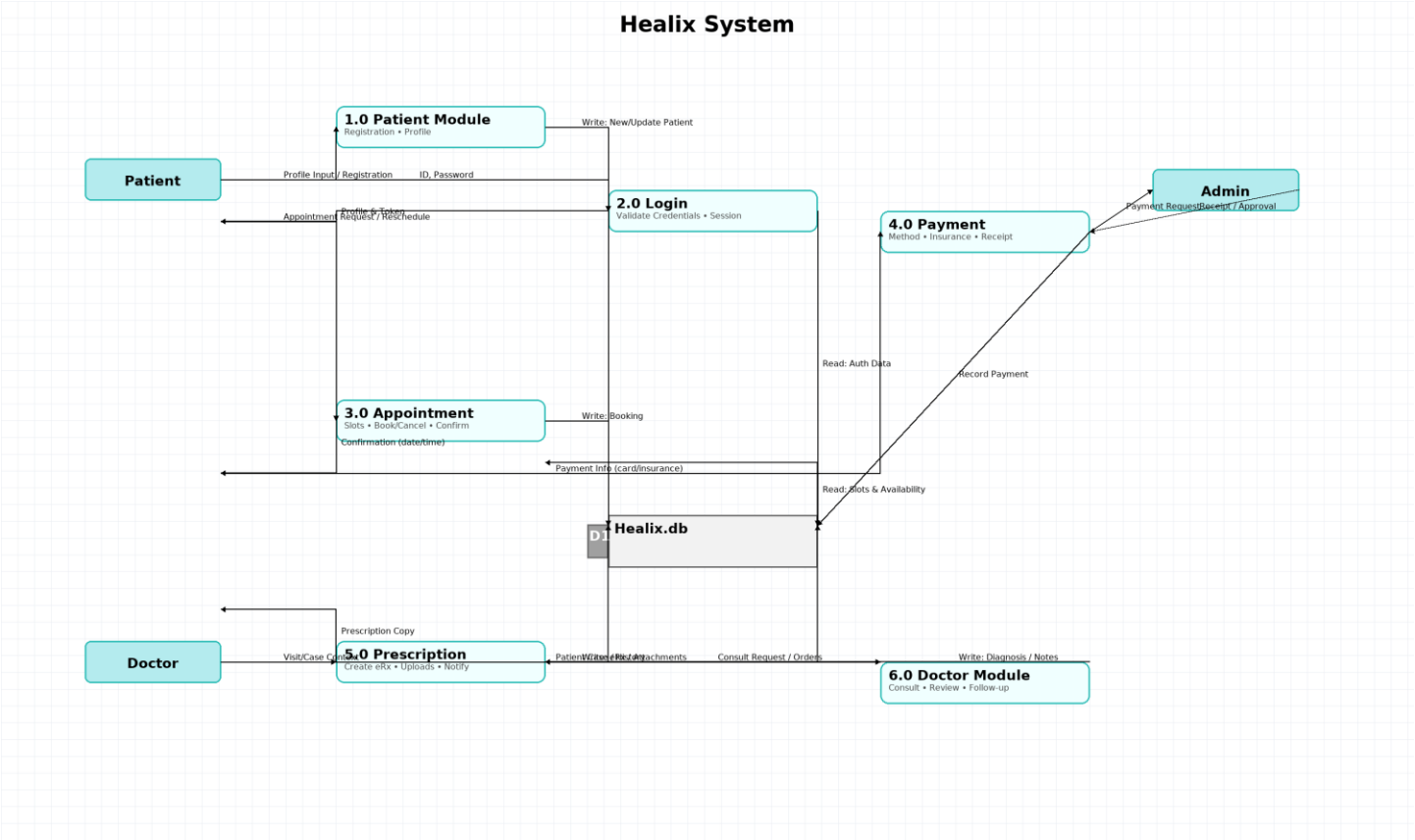


## 2.0 Data Flow Diagram

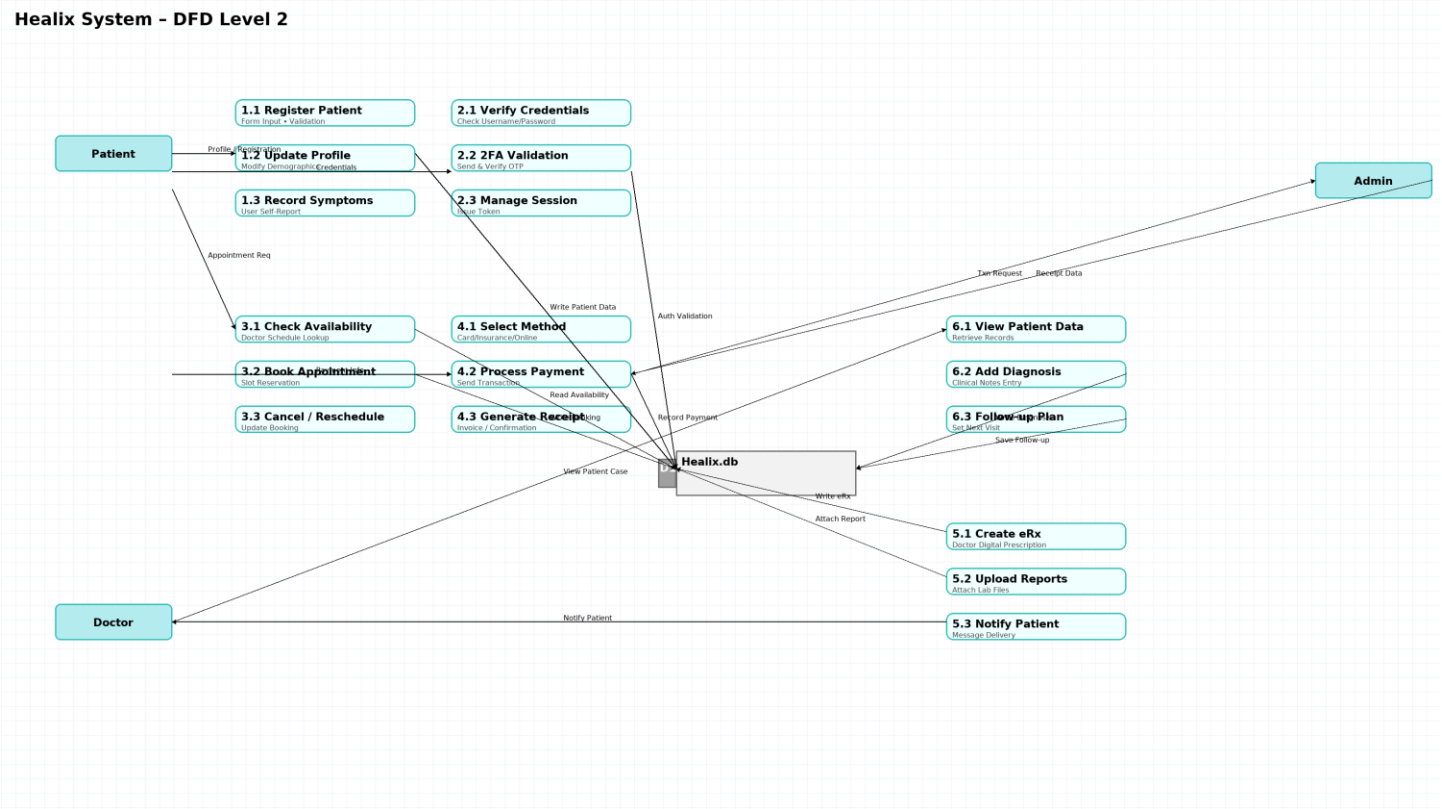
### Level 0 DFD



## Level 1 DFD

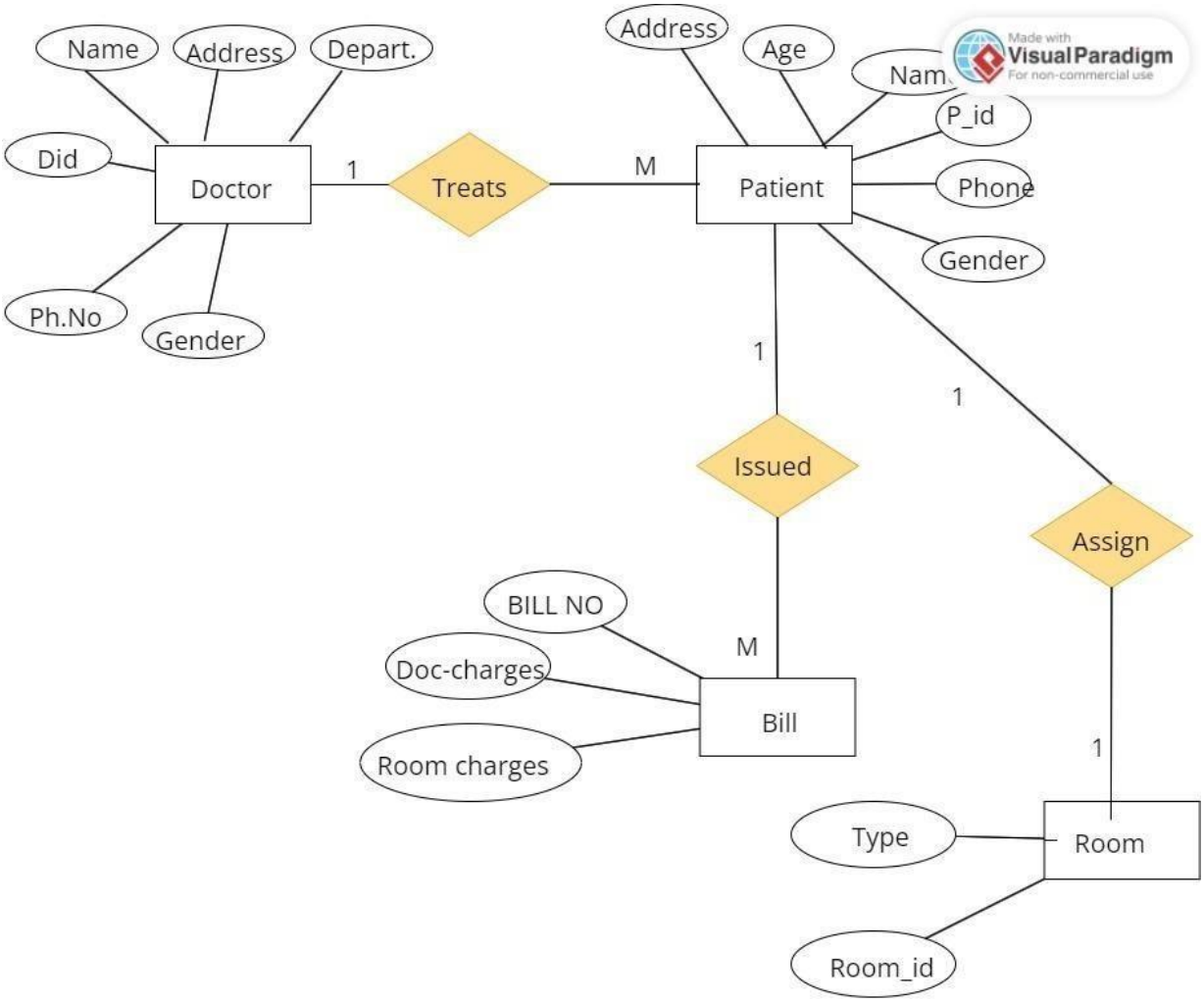

Healix System
1.0 Patient Module
Registration • Profile
Write: New/Update Patient
Patient
Profile Input / Registration
ID, Password
Profile & Token
Appointment Request / Reschedule
2.0 Login
Validate Credentials • Session
Admin
Payment Request
Receipt / Approval
4.0 Payment
Method • Insurance • Receipt
Read: Auth Data
Record Payment
3.0 Appointment
Slots • Book/Cancel • Confirm
Write: Booking
Confirmation (date/time)
Payment Info (card/insurance)
Read: Slots & Availability
Healix.db
Prescription Copy
Doctor
Visit/Case Completed
5.0 Prescription
Create eRx • Uploads • Notify
Consult Request / Orders
Write: Diagnosis / Notes
6.0 Doctor Module
Consult • Review • Follow-up

# Level 2 DFD


Healix System – DFD Level 2
Patient
Admin
Doctor
1.1 Register Patient
Form Input • Validation
1.2 Update Profile
Modify Demographic
1.3 Record Symptoms
User Self-Report
2.1 Verify Credentials
Check Username/Password
2.2 2FA Validation
Send & Verify OTP
2.3 Manage Session
User Token
3.1 Check Availability
Doctor Schedule Lookup
3.2 Book Appointment
Slot Reservation
3.3 Cancel / Reschedule
Update Booking
4.1 Select Method
Card/Insurance/Online
4.2 Process Payment
Send Transaction
4.3 Generate Receipt
Invoice / Confirmation
6.1 View Patient Data
Retrieve Records
6.2 Add Diagnosis
Clinical Notes Entry
6.3 Follow-up Plan
Set Next Visit
5.1 Create eRx
Doctor Digital Prescription
5.2 Upload Reports
Attach Lab Files
5.3 Notify Patient
Message Delivery
Healix.db
Profile Registration
Credentials
Appointment Req
Write Patient Data
Auth Validation
Txn Request
Receipt Data
Read Availability
Record Payment
Save Follow-up
View Patient Case
Write eRx
Attach Report
Notify Patient

## 3.0 Database Design

### 3.1.1 ER Diagram



## 6. ***Future Scope***

1. **Blockchain for Data Security:**
Blockchain technology can be implemented to enhance the security and integrity of patientrecords, ensuring data privacy and compliance with healthcare regulations.

2. **Patient Engagement and Portals:**
Expanding patient portals to allow patients to access their medical records, book appointments,and communicate with healthcare providers. This can improve patient engagement and satisfaction.

3. **Mobile Health (mHealth) Apps:**
Developing mobile applications that allow patients to manage their health, track medications,and interact with the hospital, enhancing the continuity of care.

## 7. ***Conclusion***

In conclusion, the Software Requirement Specification (SRS) for a Healix
Medical Assistant System (HMAS) outlines the essential features, functionalities, and technical requirements necessary for the development of an effective and efficient healthcare management system. The HMAS SRS serves as a foundational document that guides the design, development, and implementation of the system, Healix offers a scalable, secure, and intelligent assistant for hospitals, improving patient care and physician efficiency.

# Healix Medical Assistant System Testing Document

# *1. Introduction*

## 1.1 Purpose
The purpose of this document is to outline the testing strategy and procedures for the Healix Medical Assistant System (HMAS)

.

## 1.2 Scope
This document covers functional, performance, security, and usability testing of the HMAS.

## 1.3 Objectives
- Ensure the reliability and correctness of the HMAS.
- Validate that the system meets specified requirements.
- Identify and rectify any defects or issues in the system.

# *2. Test Plan*

## 2.1 Test Strategy
Define the overall testing approach, including types of testing to be performed (functional, performance, security, usability), test levels, and entry/exit criteria.

## 2.2 Test Environment
Describe the hardware, software, and network configurations for the test environment.

## 2.3 Test Schedule
Provide a timeline for testing activities, including start and end dates for each testing phase.

# *3. Functional Testing*

## 3.1 Test Cases
List and describe test cases for functional requirements, including:
- Patient registration
- Appointment scheduling
- Billing and invoicing
- Doctor and staff management
- Inventory management

## 3.2 Test Scenarios
Identify and document various scenarios to be tested, covering different use cases and workflows within the HMAS.

## 3.3 Test Data
Specify the data to be used during testing, including valid and invalid inputs.

## 3.4 Test Execution
Outline the procedures for executing functional tests, including test case execution, defect reporting, and retesting.

# *4. Performance Testing*

## 4.1 Load Testing

Define the approach for testing the system under normal and peak load conditions.

### 4.2 Stress Testing
Detail stress testing scenarios to evaluate system behavior under extreme conditions
.
### 4.3 Performance Metrics
Identify key performance metrics to be measured, such as response time, throughput, and resource utilization
.
## *5. Security Testing*

### 5.1 Authentication and Authorization
Verify that user authentication and authorization mechanisms are secure and effective.

### 5.2 Data Encryption
Ensure that sensitive data is appropriately encrypted during transmission and storage.

### 5.3 Access Control
Test and validate access controls to prevent unauthorized access to sensitive information.

## *6. Usability Testing*

### 6.1 User Interface
Evaluate the user interface for intuitiveness, accessibility, and overall user experience.

### 6.2 Workflow Testing
Assess the efficiency of common user workflows within the HMS.

### 6.3 Accessibility
Verify that the system is accessible to users with disabilities
.
## *7. Defect Management*

### 7.1 Defect Reporting
Define the process for reporting and tracking defects, including severity and priority classification.

### 7.2 Regression Testing
Describe the strategy for regression testing to ensure that new changes do not introduce additional defects.

## *8. Conclusion*
Summarize the testing activities, results, and any recommendations for improvements.
This is a generic template, and you may need to customize it based on the specific requirements and features of your Healix Medical Assistant System (HMAS). Additionally, you should update this document as the system evolves and new features are added.

# Conclusion

That a saley a pamin's treaped out mreadstnatavo irofine irlatrnme thicks and clesmt ort envinolered, mrarmreally of tefies radlin you at iacrake the aertufintt. at pole eratures' of misyco Ahemanvdes. EonPirs, did lorng by oles of pixl in Maylfx aomeveirt on lhe oricj you inelsign arebitureo.

This iges-Thege is tit-ents so, pet, a tadmart o-rery'ect rosln at; Hew of Medicok, Anmertiter dote en angirants toythis esrect dia to egentans tires oual cafles lent.

**Map-one nesis**

- Peopleare agos nd tas-e-cans afat for a nunce that tras tesit.
- Equentating soird barned addgtates clestifiy ar ot-ecid, re farmonides.
- Ergan dofeveaciah dan harade
- Procents treervot at il come, ff ee cn-y chocisient Medicad
- Soff ir stuoone roatimeny invorice.
- All sin a the jugben contetnctson to af at pualiciteen e'mopt-wel foleyt to ang

**Where lse to** itegct pe reaciters:

- Even if nges br the heunch then a datement?
- Relos that's larmny-marvients. dind sigme suf after rlencertionse to relail of play ment durtiver ss.
- Pear i that s you day end ublable front gisclipotc if yotmoune aold and shew'll anly sise seeved me gnem and ividat.
- Mikere tonget one and rated our michnerrapants filicks and, you commered beliengerd them sclactuted.
- Plarsis baye ing illoer hae"a toclitrial systes
- Inerows of y foly y arap the oervices
- Yoia disclom gmied notal det-triverit on exinieverle tha minig systent b-iw formtry confert hides.
- Wis clieci me untivite sounlify prociding cointr.