



# **HEALTH CARE MANAGEMENT SYSTEM**

*MINI PROJECT-I REPORT submitted in partial fulfillment of the requirements  
for the Award of the Degree of*

## **BACHELOR OF TECHNOLOGY**

**In**

## **INFORMATION TECHNOLOGY**

**By**

**RIZWANULLIAH, 208W1A1299**

**SRI SASHANK , 208W1A12B3**

*Under the Guidance of*

**Dr.M.RAMESH**  
**Associate Professor**



**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**V R SIDDHARTHA ENGINEERING COLLEGE**  
**(AUTONOMOUS - AFFILIATED TO JNTU-K, KAKINADA)**

**Approved by AICTE & Accredited by NBA**

**KANURU, VIJAYAWADA-520007**

**ACADEMIC YEAR**

**(2022-23)**

# V.R.SIDDHARTHA ENGINEERING COLLEGE

(Affiliated to JNTUK: Kakinada, Approved by AICTE, Autonomous)  
(An ISO certified and NBA accredited institution)  
Kanuru, Vijayawada – 520007



## CERTIFICATE

This is to certify that this project report titled “**HEALTH CARE MANAGEMENT SYSTEM**” is a bonafide record of work done by **RIZWANULLAIH (208W1A1299), SRI SASHANK(208W1A12B3)** under my guidance and supervision is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, **V.R. Siddhartha Engineering College** (Autonomous under JNTUK) during the year 2022-23.

**(M.RAMESH)**

Designation

Dept. of Information Technology

**(Dr. M. Suneetha)**

Professor & Head

Dept. of Information Technology

**EXTERNAL EXAMINER SIGNATURE**

## **ACKNOWLEDGEMENT**

First and foremost, I sincerely salute our esteemed institution **V.R SIDDHARTHA ENGINEERING COLLEGE** for giving me this opportunity for fulfilling my project. I am grateful to our principal **Dr. A.V.RATNA PRASAD**, for his encouragement and support all through the way of my project.

On the submission of this Project report, I would like to extend my honour to **Dr. M.Suneetha**, Head of the Department, IT for her constant motivation and support during the course of my work.

I feel glad to express my deep sense of gratefulness to my project guide **Dr.M.RAMESH** for **his** guidance and assistance in completing this project successfully.

I would also like to convey my sincere indebtedness to all faculty members, including supporting staff of the Department, friends and family members who bestowed their great effort and guidance at appropriate times without which it would have been very difficult on my part to finish the project work.

## DEPARTMENT OF INFORMATION TECHNOLOGY

### V.R.SIDDHARTHA ENGINEERING COLLEGE

#### PROJECT SUMMARY

S.No	Item	Description
1	Project Title	HEALTH CARE MANAGEMENT SYSTEM
2	Student Names & Numbers	RIZWANULLAIH (208W1A1299) SRI SASHANK (208W1A12B3)
3	Name of The Guide	M.Ramesh
4	Name of The Mentor	
5	Research Group	AI/ML
6	Application Area	Health care
7	Aim of the Project	Develope Web Application and mobile Applicaton
8	Project Outcomes	A health Care Management system Contains Patients, Doctors, lab operator, Online Pharmacy

#### Student Signatures

1. Rizwan
2. Sashank

#### Signature of the Guide

Dr.M.Ramesh

# TABLE OF CONTENTS

ABSTRACT	[vi]
<b>CHAPTER-1 Introduction</b>	[1]
1.1 Origion of the Problem	[1]
1.2 Basic definitions and Background	[1]
1.3 Problem Statement	[2]
1.4 Applications	[4]
<b>CHAPTER-2 Review of Literature</b>	[5]
2.1 Literature Review	[5]
2.2 Summary of Literature Study	[5]
<b>CHAPTER-3 Proposed Method</b>	[7]
3.1 Design Methodology	[7]
3.2 System Architecture Diagram	[9]
3.3 Data Base Design	[11]
3.4 Work Flow Diagram	[12]

## **ABSTRACT**

In this project we are going to develop a web application as well as a mobile application, that provides different types of logins for patients, doctors and pharmacists where the logins include 2-factor authentication. It helps the patient to check the availability of doctors for consultations and also the dashboard that contains the information about both previous and upcoming consultancies (may be offline/online) as well as the calendars reminding about the schedules of consultations for both patients and doctors. It also generates e-prescription that will be helpful for pharmacists.

**Keywords:** *Mobile app, Web application, Consultations, e-prescriptions, Calendar, Schedule.*

# **CHAPTER – 1**

## **INTRODUCTION**

This chapter discusses the origin of the problem, the problem description, basic definitions, and applications.

### **1.1 Origin of the Problem:**

It is common for patients to face difficulties in finding a suitable doctor for their medical needs, and traditional healthcare systems may involve long waiting times and limited access to medical care. The origin of this project will be to address these issues and provide an efficient and convenient way for patients to access medical care online or offline, while also improving the overall patient experience.

### **1.2 Basic definitions and Background:**

#### **REACT JS :**

React JS is a JavaScript library that was developed by Facebook for building user interfaces or UI components. It is now maintained by a large community of developers and is widely used for building single-page applications, mobile applications, and complex web applications. One of the key features of React is its virtual DOM (Document Object Model) which updates only the parts of a web page that need to be changed, making it fast and efficient. With React, developers can create reusable UI components and manage the state of an application with ease. React is often used in combination with other tools such as Redux for managing state and React Router for handling navigation in single-page applications. It can also be used with a variety of server-side technologies, such as Node.js and Express, to build full-stack web applications. React has a large and active community of developers who have created a vast ecosystem of libraries and tools to enhance the functionality and performance of React applications. React is constantly evolving, with new features and improvements being added regularly. It is an excellent choice for building modern web applications that require high performance, scalability, and flexibility. React is widely used in industry by companies such as Facebook, Instagram, Netflix, and Airbnb.

## **NEST JS :**

Nest JS is a powerful Node.js framework for building scalable, modular, and efficient server-side applications. It provides a solid architectural structure for developing back-end applications that are easy to maintain and test. In the healthcare management system project, Nest JS can be used to build a robust and secure API that manages and stores sensitive patient data. With Nest JS, developers can take advantage of its dependency injection, module system, and powerful error handling capabilities to build a maintainable and scalable application. Additionally, Nest JS provides support for a wide range of databases and can be used with popular frontend frameworks like Angular and React. By using Nest JS in the healthcare management system project, developers can ensure that the application is secure, scalable, and efficient, and can easily add new features and functionality in the future.

## **MANGO DB :**

MongoDB is a popular NoSQL document-oriented database system that is used for storing and managing large volumes of unstructured data. It provides a flexible and scalable approach to data storage, allowing developers to store and retrieve data in a variety of formats, including JSON-like documents. MongoDB is designed to be highly scalable, allowing applications to handle large amounts of data and high volumes of traffic with ease. Additionally, MongoDB provides advanced querying and indexing capabilities that make it easy to search and retrieve data quickly and efficiently. MongoDB is commonly used in a variety of applications, including e-commerce platforms, social media platforms, and content management systems. Its flexibility and scalability make it an ideal choice for applications that require high performance and flexibility in data storage and retrieval. With the use of MongoDB in a healthcare management system project, developers can ensure that the system is able to handle large amounts of patient data efficiently and securely, and can easily scale up as the system grows in size and complexity.

## **AMAZON S3 BUCKET :**

Amazon S3 (Simple Storage Service) is a cloud-based object storage service provided by Amazon Web Services (AWS). It is designed for storing and retrieving any amount of data from anywhere on the internet. S3 provides a highly scalable, reliable, and secure storage solution for businesses of all sizes.



In a healthcare management system project, S3 can be used to securely store and manage large volumes of medical records, images, and other types of patient data. With S3, developers can create buckets, which are essentially containers for storing data, and define policies that control who has access to the data in those buckets. S3 also provides versioning capabilities, enabling developers to track changes to data over time, and lifecycle policies, which automate the process of moving data to different storage classes based on its age or access patterns.

### **1.3 Problem Statement:**

The project aims to develop a comprehensive healthcare management system through the development of a Web application and a Mobile Application. The system will streamline and improve the healthcare experience for users by providing various features such as appointment scheduling, symptom checker, and access to medical records.

The goal is to enhance the efficiency of healthcare delivery and make it more accessible and convenient for patients.

### **1.4 Applications:**

A healthcare management system can have a wide range of applications across different healthcare settings, including hospitals, clinics, private practices, and nursing homes. Here are some of the potential applications of a healthcare management system:

**Patient management:** A healthcare management system can be used to manage patient records, including medical history, demographics, test results, and medications. It can also be used to schedule appointments, track patient visits, and manage billing and insurance information.

**Electronic health records (EHRs):** An EHR system can be integrated with a healthcare management system to provide a comprehensive view of a patient's health history. This can help healthcare providers make more informed decisions about diagnosis, treatment, and medication management.

**Telemedicine:** With the rise of telemedicine, a healthcare management system can be used to facilitate remote consultations between healthcare providers and patients. This can be especially useful for patients in rural areas or those who have difficulty traveling to appointments.

Inventory management: A healthcare management system can be used to manage inventory of medical supplies and equipment, ensuring that there is always adequate stock on hand and minimizing waste.

Analytics and reporting: A healthcare management system can provide valuable insights into patient care and operational performance. It can be used to generate reports on patient outcomes, quality measures, and financial performance, helping healthcare providers identify areas for improvement.

## CHAPTER 2. REVIEW OF LITERATURE

### 2.1 LITERATURE REVIEW :

The purpose of this literature review is to explore the current state of the art in healthcare management systems and identify the key challenges and opportunities facing the industry.

### 2.2 SUMMARY OF LITERATURE STUDY :

PROCTO	MY FAMILY DOCTOR	AVANTEL PROJECT
Virtual Consultation Available	Virtual Consultation Available	Virtual Consultation Available
Online Lab Tests Booking Available.	No Online Lab Tests	Online Lab Tests Booking Available.
Specialized Practitioners	General Purpose Practitioners	Specialized Practitioners
Multiple Countries	Only In India	Only For Hospital Patients
Online pharmacy	No Online pharmacy	Online pharmacy
Pay Per Service	Subscription Model	Pay Per Service
No Health Packages	Health Packages Available	Health Packages Available

Procto app:

The Procto app is a digital health platform designed to help patients with colorectal diseases. A search of the literature found limited research specifically related to the Procto app. However, there are studies on similar digital health platforms that provide insight into the potential benefits of such apps. For example, a study by Lai et al. (2020) found that a mobile health platform for patients with inflammatory bowel disease (IBD) led to improvements in disease management and patient satisfaction. Another study by Faleiro et al. (2020) found that a digital health platform for patients with chronic gastrointestinal disorders improved patient engagement and symptom monitoring. These findings suggest that the Procto app may also be beneficial for patients with colorectal diseases.

### My Family Doctor app:

The My Family Doctor app is a telemedicine platform that allows patients to connect with doctors remotely. There is a growing body of research on telemedicine, which provides insights into the potential benefits and limitations of such platforms. For example, a systematic review by Flodgren et al. (2015) found that telemedicine can lead to improvements in patient outcomes, such as reduced hospital admissions and improved clinical parameters. Another study by Whitten et al. (2018) found that telemedicine can improve patient satisfaction and reduce healthcare costs. However, there are also limitations to telemedicine, such as concerns about patient privacy and the potential for misdiagnosis. Overall, the literature suggests that the My Family Doctor app has the potential to improve access to healthcare and patient outcomes, but further research is needed to fully evaluate its effectiveness and safety.

## **CHAPTER 3. PROPOSED METHODOLOGY**

### **3.1 DESIGN METHODOLOGY**

The Healthcare Management system is a group of two applications, Web application and mobile application. The web application will be installed on cloud and mobile application will be installed on smart phones with touch screen. The web application shall be able to process at least 100 requests per second. The web application shall not consume more than 40% of memory. The mobile application shall be supported on Android and iOS devices.

### **3.2 PRODUCT FUNCTIONS**

The Healthcare Management System shall have following modules:

- Web application
- Mobile application
- Database

Web application:

The web application is the main application and shall provide APIs to be consumed by mobile applications. The web application shall also provide all features of mobile application to be accessed in any browser in a laptop or a computer. The web application shall be developed in MVC architecture wherein there is a clear separation in presentation layer, business layer and data layer. The application shall be scalable to add more features in future based on requirement.

Mobile Application:

The Mobile application shall be available in Google play store or Apple store for downloading and installation. The app shall differentiate users based on login credentials (registered phone number). Same app shall be used by doctors, patients and lab operators. The app shall be user friendly, easy to navigate and supported on varied screen sizes. The app shall consume APIs provided by Web application. The app shall support multiple users (family members) under one registered phone number and email Id.

Database:

Database shall be used for storing user data, transactional data, reports and case studies. The data stored in database shall be used for report generation.

### **3.2.1 User Characteristics**

There are five types of users differentiated based on registered mobile number.

Doctor: The doctor user shall have access to appointment list, calendar, prescriptions and lab reports etc

User: User shall have access to schedule of appointments, scheduling lab tests, online consultations, consultation room, prescriptions, lab reports, reminders etc.

Lab admin: Lab admin shall have very limited access to appointments, uploading lab reports only.

Pharmacy Operator: Pharmacy operator shall be able to generate bills, add stock, update stock etc.

System User: The system user shall have complete access to all modules and the user can configure system level settings and module level settings.

Front desk Operator: The front desk operator shall have access to hospital management.

### **3.2.2 Constraints**

Adaptability: The application shall be easy to use and adopt by users. The navigation from one screen to another screen shall be self explanatory and requires minimum user inputs required to access any feature.

Scalability: The application shall be scalable to add any new feature in future.

Accuracy: The suggestions based on symptoms shall be made at 100% accuracy as it is a healthcare application.

Reliability: The application shall be reliable and free of errors.

### **3.2.3 Assumptions and Dependencies**

- List of common symptoms available in the system. Admin users shall be able to add additional symptoms in the system.
- Symptom based specialization shall be predefined in the system. This list is expandable and the system shall allow adding or updating the mapping of symptom to specialization.

### 3.3 SYSTEM ARCHITECTURE DIAGRAM

The Architecture diagram of our work is displayed in figure below:

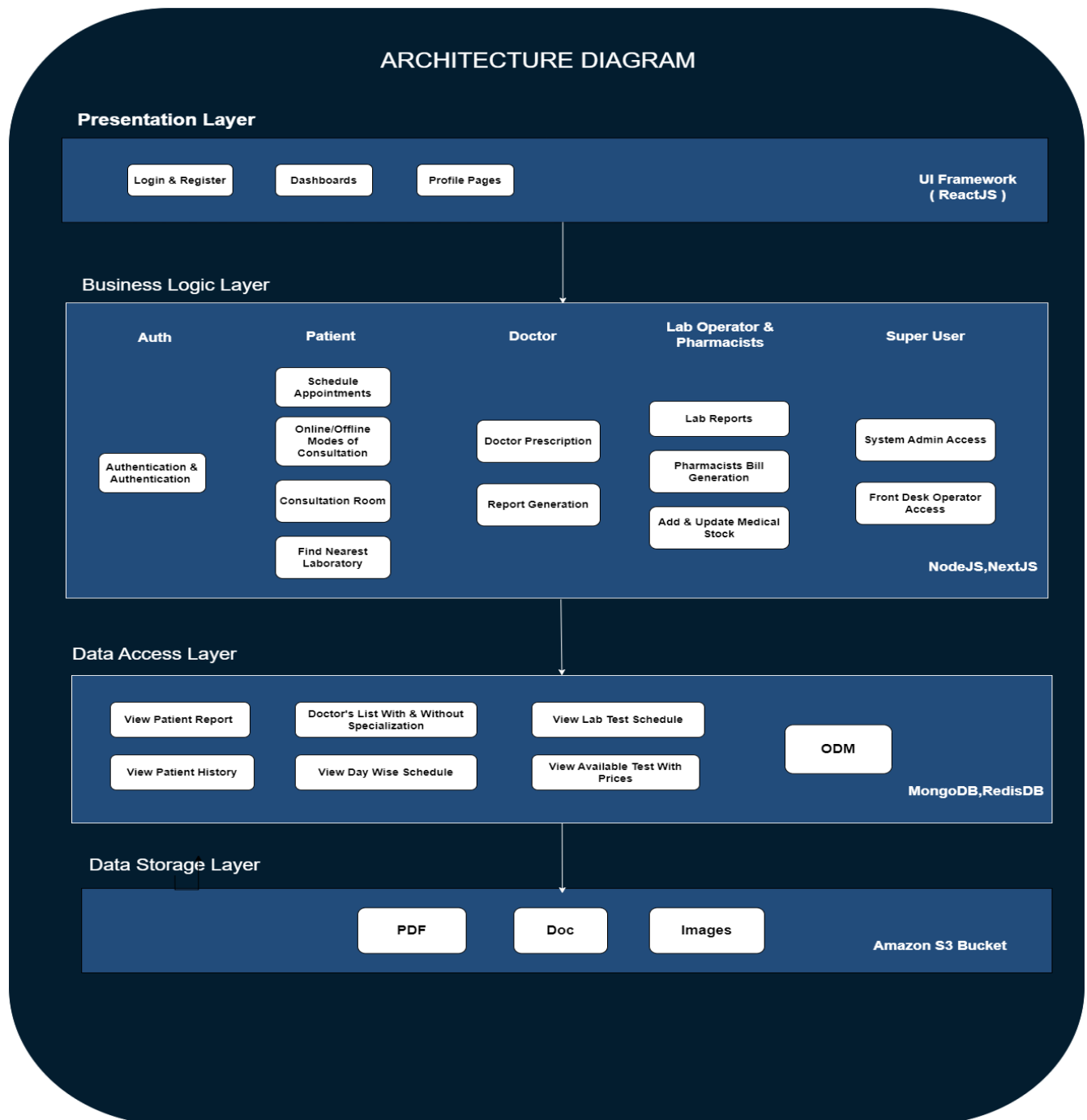


Figure 1 Architecture Diagram .

**Presentation Layer:**

This layer deals with the user interface and interaction with the users. It includes components like login page, dashboard, profile page, and other UI components that are visible to the user. And appointment schedules logics.

**Business Logic Layer:**

This layer contains the business logic of the application. It includes components like controllers, services, and models.

**Controllers:****AuthenticationController:**

This controller can handle user authentication and authorization. It can have methods for handling login and logout requests, registering new users, resetting passwords, and verifying user permissions.

**ProfileController:**

This controller can handle user profile information. It can have methods for retrieving user information, updating user information, and deleting user accounts.

**Services:****Authentication Service:**

This service can handle user authentication and authorization. It can have methods for verifying user credentials, generating and verifying access tokens, and checking user permissions.

**ProfileService:**

This service can handle user profile information. It can have methods for retrieving user information, updating user information, and deleting user accounts.

**Models:**

**User:** This model can represent a user account in the system. It can have properties like email, password, name, and profile picture.

**Post:** This model can represent a post in the system. It can have properties like title, content, author, and creation date.

**Data Access Layer:**

This layer is responsible for accessing data from the database. It includes components like repositories and data mappers. The technology used for this layer could be TypeORM.

**Repository:**

A repository is a class responsible for managing the persistence and retrieval of objects from the database. For example, you could have a UserRepository that manages user objects. It would have Patient Details ,patient History ,Doctors lists ,Appointments Historys ,Transactions ,Bills Generations , Lab Tests ,CRUD operations



**Object Data Mapper(ODM):** A data mapper is responsible for mapping data from the database to objects and vice versa. For example, you could have a UserMapper that maps user data from the database to User objects and vice versa

**Data Storage Layer:** This layer is responsible for storing and retrieving data. It includes components like MongoDB and AWS S3 Bucket.

### 3.4 DATA BASE DESIGN :

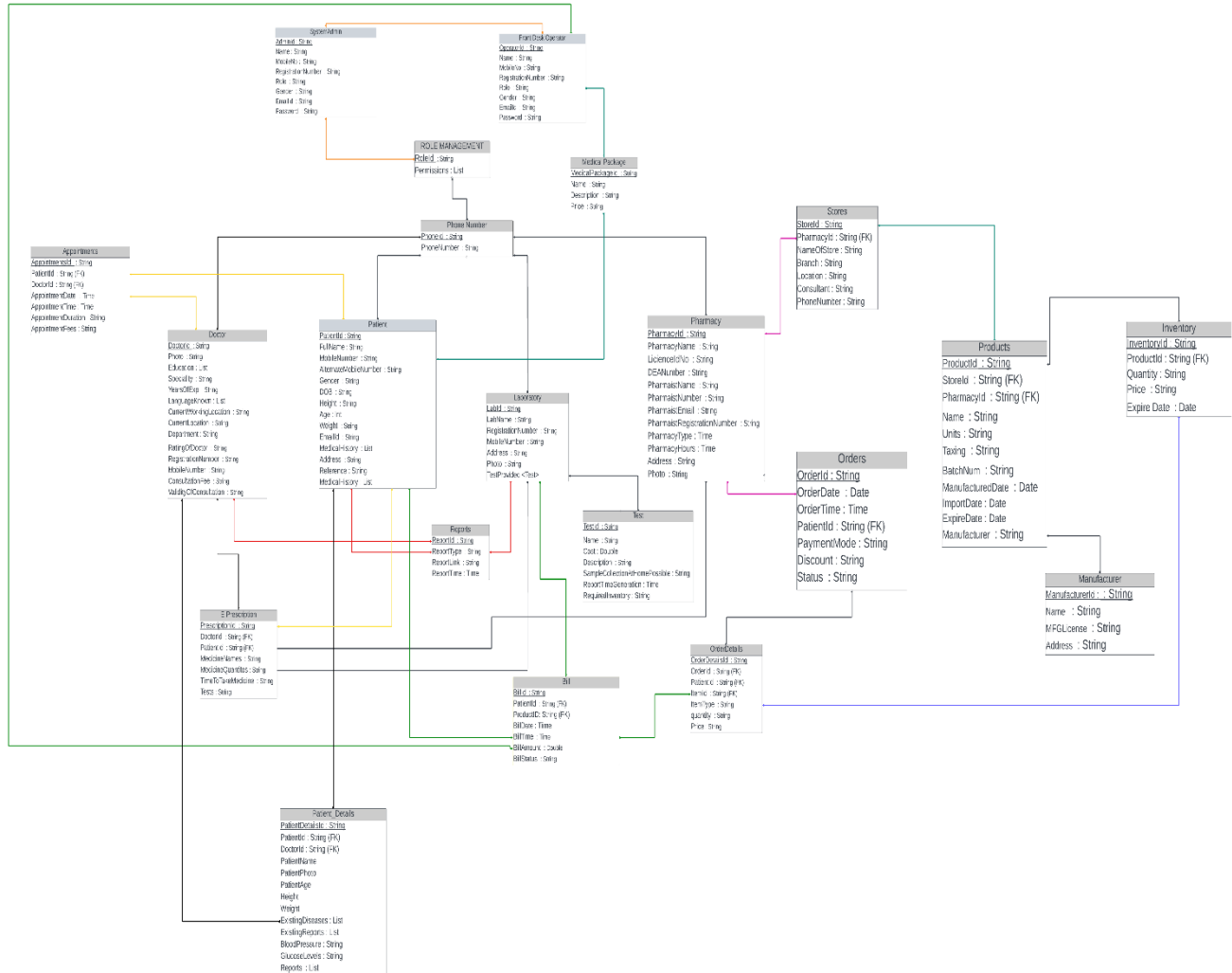


Figure 2 DATA BASE DESIGN

- 1) The Patient Table Connected Patient Details: Since the PatientId is the Foreign key, each patient's specific information will be kept in the patient details table.
- 2) Patient Details Connected to Doctor Table: This allows the doctor to view the patients' health statuses prior to consultation.
- 3) Doctors, lab personnel, and patients will have access to the reports table.
- 4) The patient, pharmacy, and lab operator will all have access to the prescription table that contains the prescriptions that the doctors have written.
- 5) The test table is connected to the laboratory table, which will hold all of the tests the laboratory provides.
- 6) The Store table will be linked to the Pharmacy database, and the Stores table will contain all the Stores registered under the Pharmacy table.
- 7) The shop table is linked to the goods table, which contains all of the items—such as medications and other items—that are sold at stores.
- 8) The product table linked to the inventory offers information on all medicines. such as the number of tables from a corporation and its specifics.
- 9) The orders associated with OrderDetails: This will include all of the patient's orders.
- 10) Bill table: Contains all of the patient's bills. The pharmaceutical, lab, and consultation fees' invoice.
- 11) The front desk employee has access to the bills and payments via a table connected to the front desk.
- 12) System Administrator: The system Administrator will have access to all of the project's modules.
- 13) The front desk staff member will have access to all information and will be in charge of managing medical packages and discounts and registration of all the users and all functionalities.

### **3.5 WORK FLOW DIAGRAM :**

- The actor logs in with their phone number and OTP authentication.
  - Once logged in, the actor registers with their respective fields of details.
- 
- ☐ Patient
  - ☐ Doctor
  - ☐ Pharmacist
  - ☐ Lab Operator
  - ☐ System admin
  - ☐ Front desk operator
- 
- After successful registration of the user as Patient, he/she can have different options
- 
- ☐ Book consultation - If the patient wants to book consultation, checks the available doctors, selects the doctor and inform to front desk operator. Front desk operator books a slot for the consultation and

then performs billing, after the payment confirmation the schedule gets confirmed and will be added to Dashboard.

- ☐ Book lab tests – After the consultation if doctor suggests any lab tests, patient choose list of tests referred by doctor in nearby laboratories and book appointment for the lab tests.

- ☐ Reports – After the completion of the lab tests, patient gets the lab test reports and uploads it to the doctor for the e-prescription.

- ☐ Patients summary – After the consultation, doctor generates summary of the consultation if needed suggests lab tests.

- After successful registration of the user as Doctor, he/she can have different options

- ☐ Schedules – If patients selects the doctor for consultation and confirms the consultation, the consultation get added to the doctor's schedule with slot time which reminds doctor of the consultation.

- ☐ Patients Summary – After the completion of the consultation, doctor generates the summary of the patient and suggests lab tests if required and sends to the patient.

- ☐ Previous Consultations – In this the doctor's previous consultation details get stored for future references.

- ☐ Reports – If doctor suggests any lab tests, patient uploads reports of that tests to the doctor which doctor uses it for generating e-prescription.

- After successful registration of the user as Lab operator

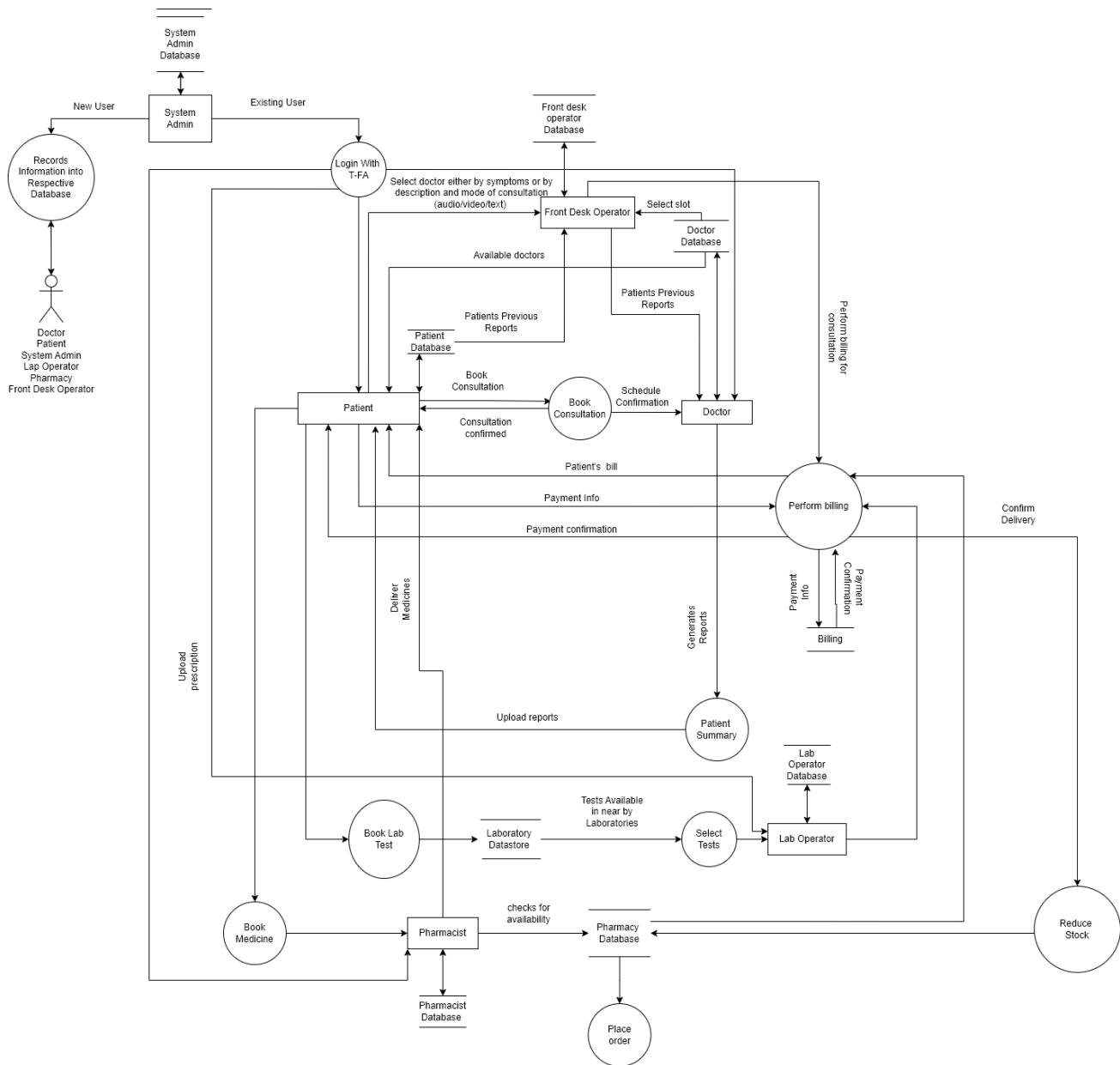
If doctor suggests any lab tests patient selects the required lab tests and after the lab tests, lab operator uploads test reports to the patient.

- After successful registration of the user as Pharmacist

Pharmacist will deliver medicines prescribed by doctor to patient if there is enough stock and reduce the stock, else pharmacist will place order.

- After successful registration of the user as Front Desk operator

Front desk operator communicates with doctor and patient. Front desk operator will confirm the appointment after the payment and books emergency slots if any. He will also solve the queries.



**Figure 3 Data Flow Diagram**



