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

RIZWANULLAH MD, 208W1A1299

SRI SASHANK P, 208W1A12B3

Under the Guidance of

M.RAMESH
Asst 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.

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.

NEXT JS :

Next 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, Next JS can be used to build a robust and secure API that manages and stores sensitive patient data. With Next JS, developers can take advantage of its dependency injection, module system, and powerful error handling capabilities to build a maintainable and scalable application. Additionally, Next JS provides support for a wide range of databases and can be used with popular frontend frameworks like Angular and React. By using Next 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.

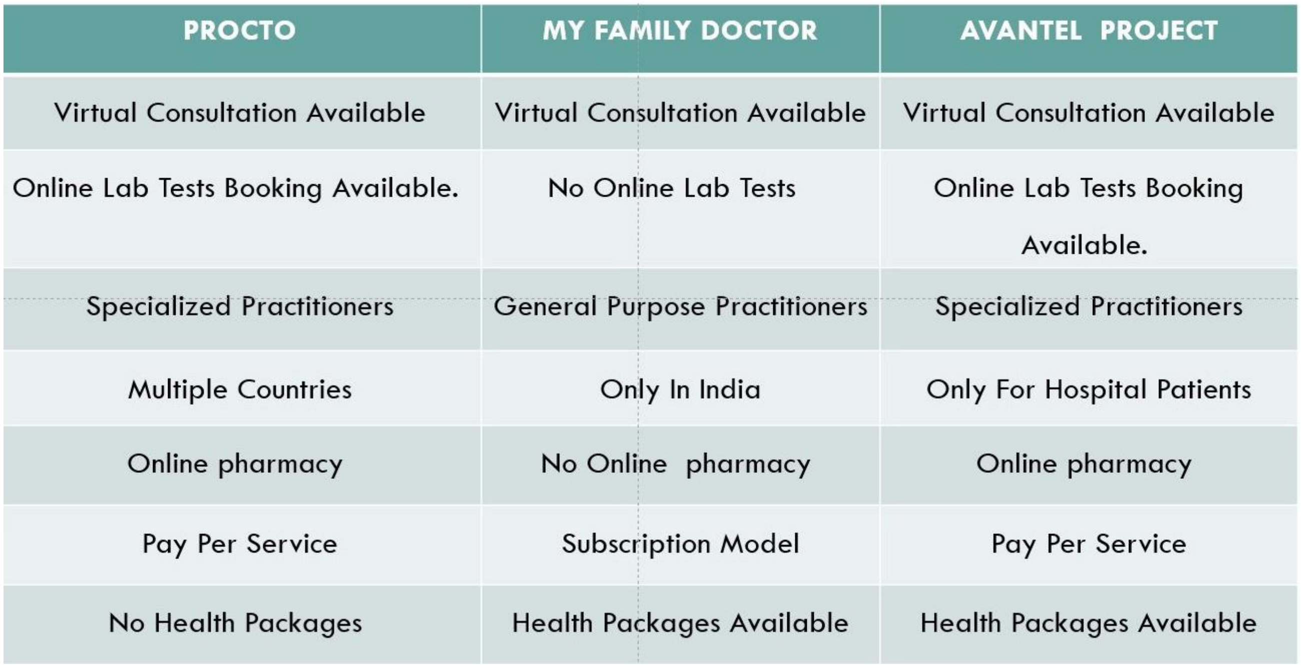
Analytics and reporting: A healthcare management system can provide valuable insights into patient care and operational performance. It can be used to generate reports on patient outcomes, quality measures, and financial performance, helping healthcare providers identify areas for improvement.

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 [1] 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[2] 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 could 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.

3.1.1 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.

3.1.2 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.

3.1.3 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.

3.1.4 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.

3.2 SYSTEM ARCHITECTURE DIAGRAM

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

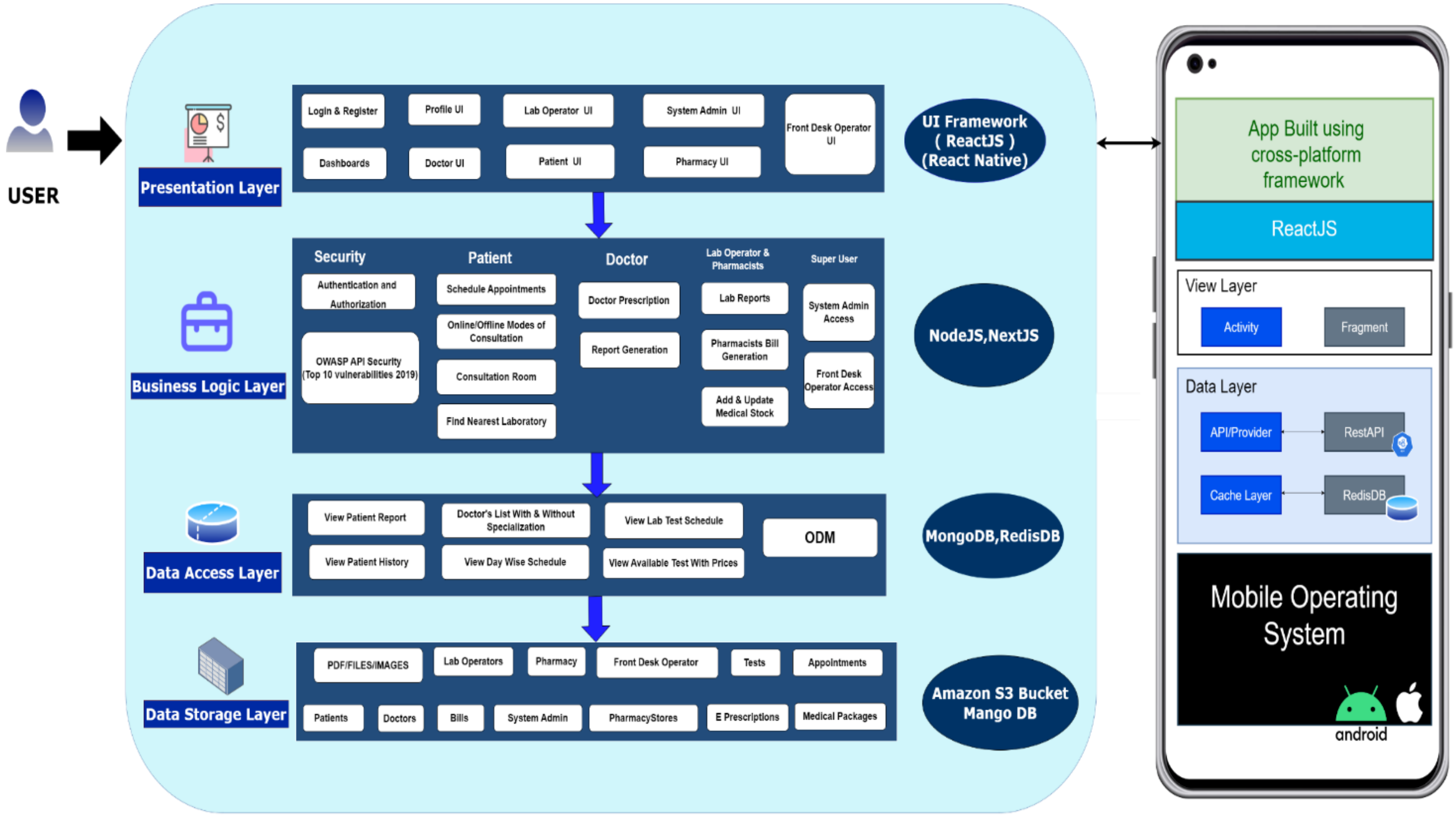


Figure 1 Architecture Diagram .

3.2.1 Description - Layered Architecture

Presentation Layer:

This layer deals with the user interface and interaction with the users. It includes components like User Registration, User Dashboards, User Profiles, Appointment / Schedule logic and other UI components that are need to be visible to the user.

Business Logic Layer:

This layer contains the business logic of the proposed application. It includes components like controllers, services, and models to handle security, user operations and healthcare services.

Controllers :

Controllers in HCMS Proposed Layered Architecture are of two types :

- Authentication Controller

- Profile Controller

Authentication Controller is to manage every user authentication and authorization. Its operations include user login request, user logout, registering new users, resetting passwords, and controlling user access.

Profile controller is to maintain user information. Supporting methods need to be for retrieving user information, updating user information, and deleting user accounts.

Services

Services in HCMS Proposed Layered Architecture are of two types :

- Authentication Service
- Profile Service

Authentication Service:

This service is to handle user authentication and authorization at all times of user entry/exit of the developing system. Various necessary methods include authenticate user credentials, generating and validating access tokens and checking user permissions.

Profile Service:

This service deals with user profile information. Methods relevant here include retrieving user information, updating user information and deleting user accounts.

Models

User Model: This model represents all types of users in the system. Its characteristics include email, password, username, and profile picture.

Post Model: This model represents a post in the system. Its properties comprise title, content, author, and date.

Data Access Layer : The data access layer is responsible for managing the interaction between the application and the underlying database. In this layer, it is included several functionalities, such as:

- ➢ **View Patient Report:** This functionality allows authorized users to view patient reports, which may include medical history, diagnosis, medications, and other related information.

- **Doctor's List With & Without Specialization:** This functionality provides a list of all doctors in the healthcare facility, along with their respective specialties.
- **View Lab Test Schedule:** This functionality allows authorized users to view the schedule of lab tests for patients, which includes the date, time, and location of the tests.
- **View Patient History:** This functionality enables authorized users to view a patient's complete medical history, including past treatments, medications, and test results.
- **View Day Wise Schedule:** This functionality provides a day-wise schedule of appointments for doctors, along with the names of the patients and their appointment times.
- **View Available Test With Prices:** This functionality enables authorized users to view the available tests in the healthcare facility, along with their prices.
- **ODM:** This stands for Operational Data Model, which is a standard format for exchanging healthcare data between systems. This functionality helps in the efficient exchange of data between the application and the underlying database

- **Data Storage Layer :** The data storage layer in a healthcare management system is responsible for storing and managing data for various entities in the system. In this layer, it is included with the following entities:
- **-PDF/FILES/IMAGES:** This entity stores all the PDF files, images, and other documents related to the healthcare management system.
- **Lab Operators:** This entity stores data related to lab operators, including their personal information, work schedules, and assignments.
- **Pharmacy:** This entity stores data related to the pharmacy, including medication inventory, prescription history, and other related information.
- **Front Desk Operator:** This entity stores data related to front desk operators, including their personal information, work schedules, and assignments.
- **Tests:** This entity stores data related to lab tests, including test names, descriptions, and pricing.
- **Appointments:** This entity stores data related to appointments, including appointment times, patient names, doctor names, and other related information.
- **Patients:** This entity stores data related to patients, including personal information, medical history, appointment history, and other related information.
- **Doctors:** This entity stores data related to doctors, including their personal information, specialties, work schedules, and assignments.

- **Bills:** This entity stores data related to bills generated for patients, including bill amounts, payment status, and other related information.
- **System Admin:** This entity stores data related to system administrators, including their personal information, access levels, and other related information.
- **Pharmacy Stores:** This entity stores data related to pharmacy stores, including their locations, inventory, and other related information.
- **E-Prescriptions:** This entity stores data related to electronic prescriptions, including prescription details, patient information, and other related information.
- **Medical Packages:** This entity stores data related to medical packages, including package names, descriptions, and pricing.

The data storage layer uses AWS S3 BUCKET for storing PDF/files/images and Mango DB technology for storing the rest of the data. AWS S3 BUCKET is a cloud-based storage service that provides secure, scalable, and durable storage for objects like files and images. Mango DB is a NoSQL database technology that provides flexible and scalable data storage capabilities for complex data structures like JSON documents.

3.3 High Level Architecture :

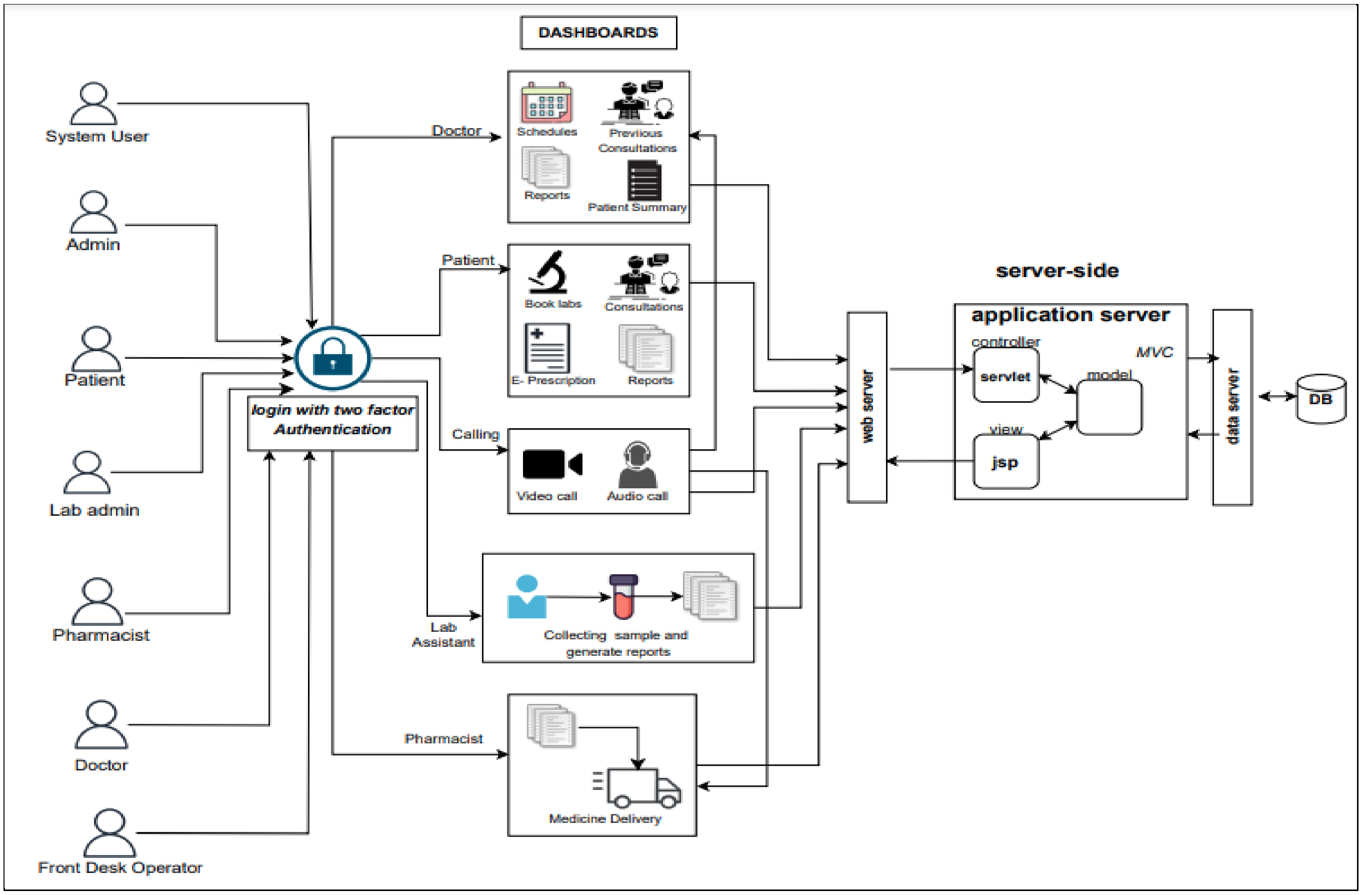


Figure 3.2 Architetcure of the Overall Proposed System

3.4 DATA BASE DESIGN :

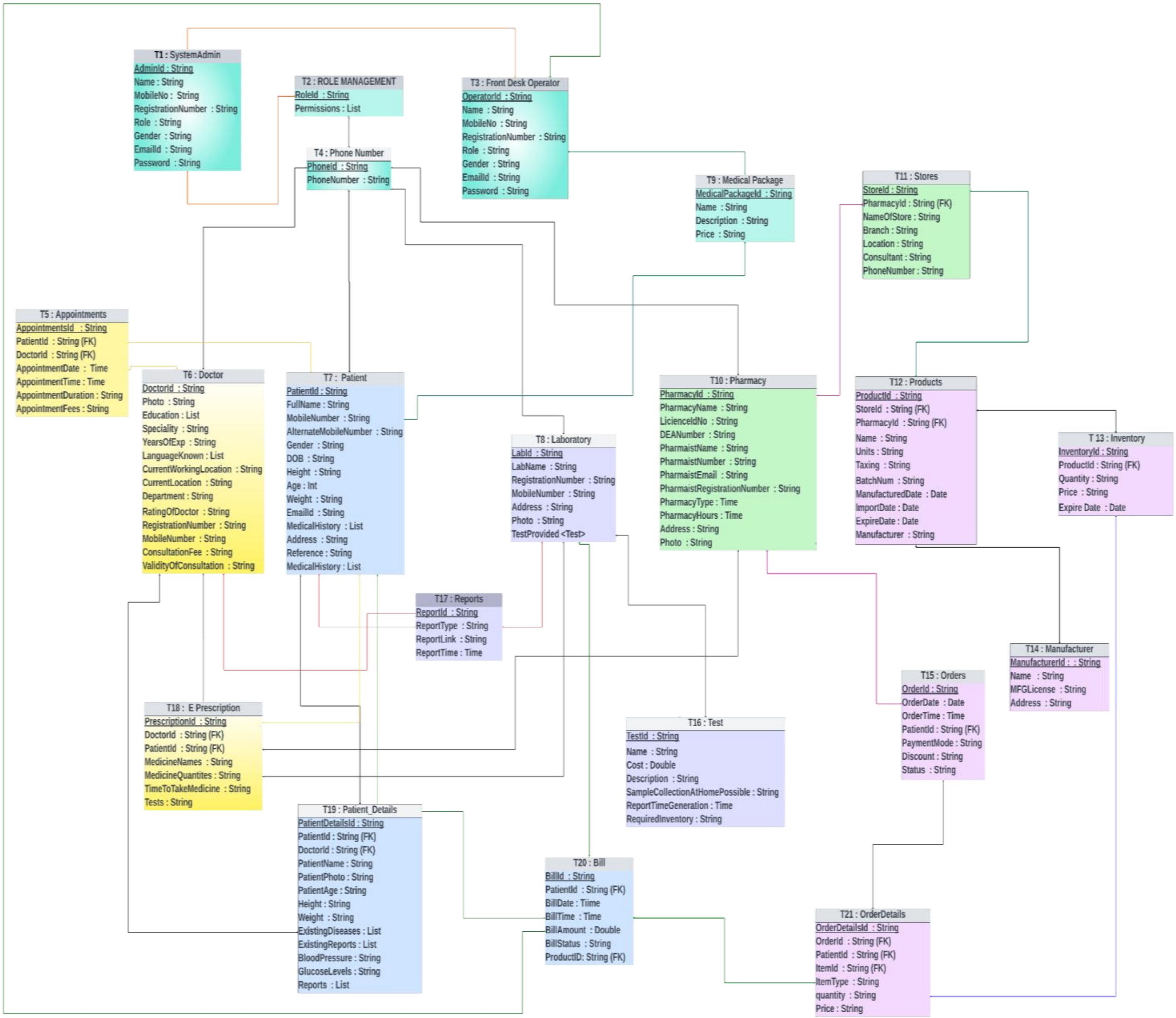


Figure 2 DATA BASE DESIGN

3.4.1 DATABASE DESIGN DESCRIPTION :

T1 : SystemAdmin : **AdminId (PK) ,Name , MobileNo , RegistrationNumber , Role , Gender , EmailId, Password** : This is the super user in the proposed system. Admin will have access to all the Modules

T2 : ROLE MANAGEMENT : **RoleId (PK), Permissions : This table is used for permissions**

T3 : Front Desk Operator **: OperatorId (PK), Name , MobileNo , RegistrationNumber , Role , Gender, EmailId , Password :** Bills and payments can also be accessed by the Front desk operator. The front desk operator will have access to all and medical packages and discounts will be managed by front desk operator .

T4 : Phone Number **: PhoneId (PK), PhoneNumber** : It will store all the phone numbers.and with accordinly give access to the particular user

T5 : Appointments : **AppointmentsId (PK), PatientId (FK), DoctorId (FK) ,AppointmentDate, AppointmentTime , AppointmentDuration , AppointmentFees :** The appointemnts booking will be stored in this table with reference to the patinetsid and doctorsID .

T6 : Doctor : **DoctorId (PK), Photo , Education ,Speciality , YearsOfExp ,LanguageKnown, CurrentWorkingLocation ,CurrentLocation , Department ,RatingOfDoctor RegistrationNumber , MobileNumber ,ConsultationFee ,ValidityOfConsultation** : The individual doctor details will be stored here and Doctorid is the Foreign key. It has a relationship with E Precription table.

T7 : Patient : **PatientId , FullName , MobileNumber , AlternateMobileNumber , Gender , DOB , Height , Age : Int Weight , EmailId , MedicalHistory , Address , Reference , MedicalHistory ,**

T8 : Laboratory **: LabId , LabName , RegistrationNumber , MobileNumber , Address , Photo , TestProvided <TEST >** : Apart from laboratory details, this table is connected to Test table

T9 : Medical Package : **MedicalPackageId , Name , Description , Price :** The medical packages which are dialy updated by the front desk operator will be stored in this and will be accessed by the patients and they can use the packages .

T10 : Pharmacy : **PharmacyId , PharmacyName , LicienceIdNo , DEANumber , PharmaistName , PharmaistNumber , PharmaistEmail , PharmaistRegistrationNumber , PharmacyType PharmacyHours ,Address , Photo** : Maintains pharmacy operations. It has connections to the Store table.

T11 : Stores : **StoreId , PharmacyId , (FK) NameOfStore , Branch , Location , Consultant , PhoneNumber :** The stores contains all the stores which are registred under the pharmacy table. The store table is connected to the products table

T12 : Products : **ProductId , StoreId , (FK) PharmacyId , (FK) Name , Units , Taxing , BatchNum , ManufacturedDate , ImportDate , ExpireDate , Manufacturer** : Contains all the Products like medicines etc., available in the stores. This table has relationship with inventory table.

T13 : Inventory : **InventoryId , ProductId , (FK) Quantity , Price , Expire Date** : Comprises all medicines details. Like how many tables from a company and its details . This has relationship with ordered details table.

T14 : Manufacturer : **ManufacturerId : , Name , MFGLicense , Address** : The manufacturer details of the medicines will be stored In this .

T15 : Orders : **OrderId , OrderDate , OrderTime ,PatientId , (FK) PaymentMode , Discount , Status** , : Ther orders which are received from the patient will be stored in this .

T16 : Test : **TestId , Patient ID , (FK) Name , Cost :** Double Description , SampleCollectionAtHomePossible , ReportTimeGeneration ,Required Inventory : The test table will contain all the tests provided by the laboratory.

T17 : Reports **: ReportId ,ReportType ,ReportLink,ReportTime** : will be accessed by Doctors, lab operator, patient

T18 : E Prescription : **PrescriptionId (PK), DoctorId (FK) ,PatientId (FK), MedicineNames MedicineQuantites , TimeToTakeMedicine , Tests** : Contains the prescription given by the doctors which will be accessible by the Patient, Pharmacy and Lab operator.

T19 : Patient_Details : **PatientDetailsId ,PatientId (FK), DoctorId (FK), PatientName , PatientPhoto , PatientAge , Height , Weight , ExistingDiseases, ExistingReports , BloodPressure , GlucoseLevels ,Reports :** It has relationship with Doctor Table. By this the doctor can view the health conditions of the patients before consultation.

T20 : Bill : **BillId (PK), PatientId (FK), BillDate ,BillTime , BillAmount , BillStatus , ProductID, (FK) :** Contains all the billing information. The bills belong to patient, pharmacy, laboratory and consultation Fees. It is connected to front desk operator table.

T21 : OrderDetails : OrderDetailsId (PK), OrderId (FK) ,PatientId (FK), ItemId (FK) ,ItemType , quantity,Price : This contains all the orders received from the patient.

Patient Module : Registration, Online consultation, Follow-up consultation, Schedule lab test, Previous consultations, Profile management

Doctor Module : Registration, Today's consultations, Previous consultations, Calendar, View Scheduled Appointments ,Receive Notifications, View lab reports, Prescribe lab tests, Prepare Prescription

Pharmacy Module : Generate invoice, Payments, Stock availability etc.

Lab operator Module : Today's appointments, Previous appointments, Upload reports

System Admin Module : Access to all the Modules

Front desk Operator Module : Controlled by system admin and access accordingly, Managing other information like payments, medical packages, discounts

3.5 Work Flow Design

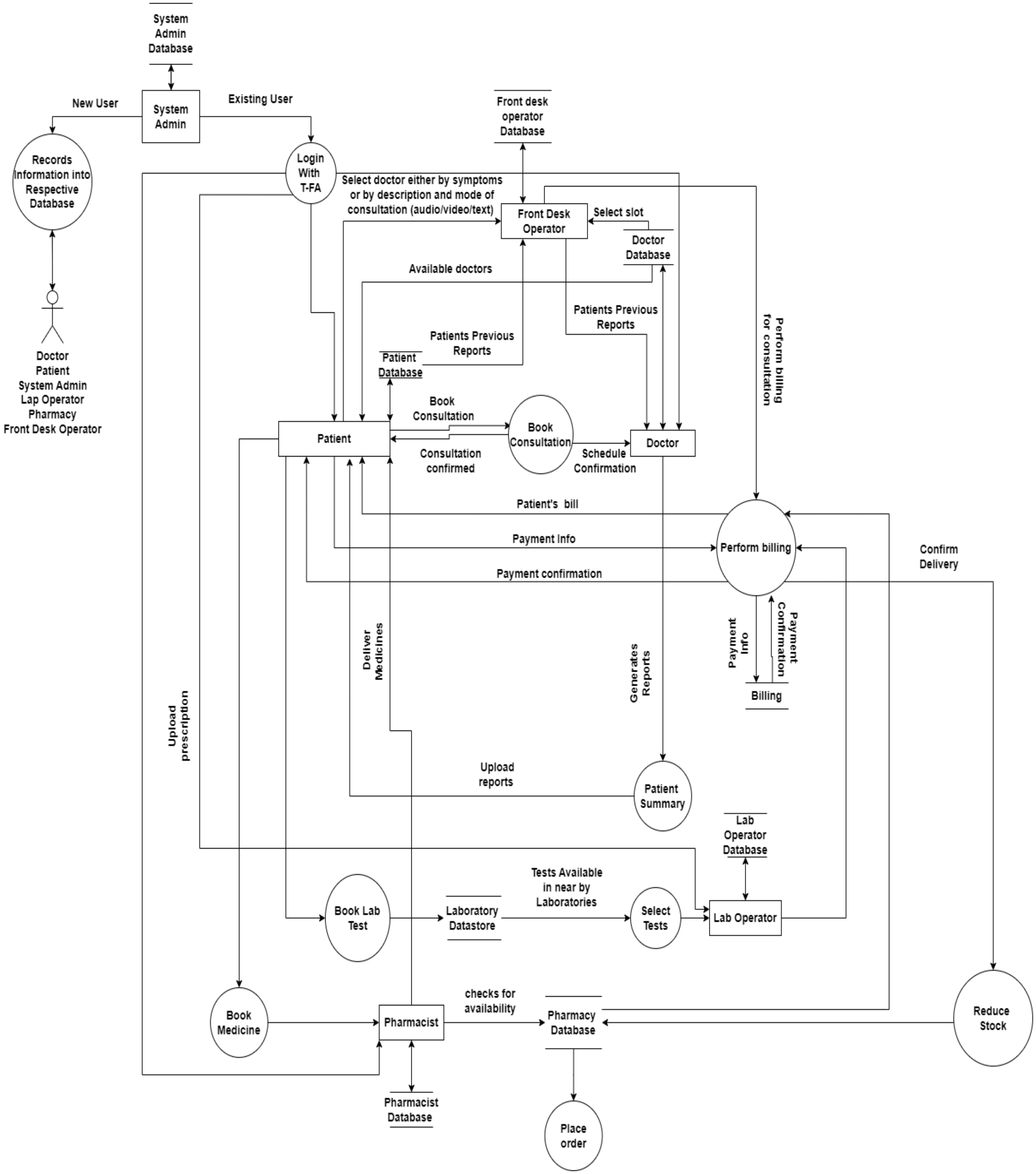


Figure 3.3 Data Flow Diagram of the Proposed System

3.5.1 DESCRIPTION OF DATA FLOW

• The user/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

CHAPTER 4 CONCLUSION AND FUTURE STUDY

CONCLUSION :

FUTURE STUDY :

PHASE-1:

Developing the Web Application for the Health Care Management System.

PHASE-2:

Developing Mobile Application for the Health Care Management System.

PHASE-3:

Apply Authentication & Security in Web application and Mobile application. Identify and handle OWASP API Security the Top 10 Vulnerabilities 2019 in APIs exposed by the web application.

Implement TLS v1.2 in APIs for securing data transfer between web server and mobile/web application. Implementation of logging and registration for all users in the proposed system. Design registrations, schedules, calendars, e-prescription through dashboard**.**

REFERENCES

[1] https://nextjs.org/

[2] https://react.dev/

[3] https://www.mongodb.com/

[4] https://www.practo.com/

[5] https://www.clickittech.com/devops/web-application-architecture/

[6] https://soft-builder.com/bank-management-system-database-model/