Dharmsinh Desai University,Nadiad

Faculty of Technology

Department of Computer Engineering

B.Tech CE Semester – VI

Subject: System Design Practice

A

Project Report

On

Health-card

Submitted By:

Aayush J Dakwala	CE020	18CEUOD015
Tanuj V Doshi	CE031	18CEUOD008
Bhavik M Mojidra	CE071	18CEUBD001

Guided By:

Prof.Pandav K Patel

Dharmsinh Desai University, Nadiad

Faculty of Technology,Department of Computer Engineering

CERTIFICATE

This is to certify that System Design Practice Project entitled "**Health-Card**" is the bonafied report of work carried out by

Aayush J Dakwala(18CEUOD015)

Tanuj V Doshi(18CEUOD008)

Bhavik M Mojidra(18CEUBD001)

Of Department of Computer Engineering ,Semester VI , academic year 2019-2020, under our supervision and guidance.

Guide	HOD
Prof. Pandav K. Patel Associative Professor of Department of Computer Engineering, Dharmsinh Desai University, Nadiad	**Dr. C. K. Bhensdadia** Head of Department of Computer Engineering, Dharmsinh Desai University, Nadiad

Content

Abstract

We have Aadhar card, pan card, Voter id, driving licence like identity proofs for our day to day transactions, but we have major missing that is our health card, which manages our day to day health history like disease, allergy etc. And same card is used everywhere in private and public hospitals and clinics, medical store, laboratory to track your health-related data.

A big responsibility to maintain health related data can be resolved by the help of this system. because of this system user does not have to standing at queue of hospitals to take new case every time. User just need to carry his/her health card.

Introduction

2.1 Brief Introduction

- Purpose: -

  The purpose of the Health card is to provide an easy, responsive web interface for the patient who want to store health related data online

- Scope: -

  Health card System provides information of the Patient all medical related data, and lets the government use that data and get all statistic analysis of patient health data area wise .

- Objectives: -

  Health card System aims The application reduces lot of work load for government and can be helpful to all patient.

2.2 Tools/Technologies used

Technology: - HTML 5

CSS3

BOOTSTRAP 4

ANGULAR CLI

NODE

TYPESCRIPT

MONGODB

Platform: - Google chrome, Mozilla Firefox

Tools: - Visual Studio Code, Mongo DB Compass

3 SOFTWARE REQUIREMENT SPECIFICATION

3.1 Types of Users:

1.Doctor

2. Laboratorian

3. patient

3.2 Functional Requirement: -

R.1 Doctor:

R.1.1 Registration
Description: - If Doctor Wants To Register To Health-card system he/she can register by giving valid proofs.
Input: – Doctors Details like licence number, email, licence expiration date etc
output: – sign in page

R.1.2 Login
Description: - Doctor can login to system by giving valid email id and password.
Input: - doctor credential
output: - home page

R.1.3 View Previous health reports

Description: Doc tor shall check health reports and health statistics of the patient.

Input: User Selection

Output: Displays Selected Health report and statistics

R.1.4 Write a Prescription

Description: Doctor shall write a Prescription of a medicine for the required treatment.

Input: Prescription

Output: Generation of a Prescription

R.2 Laboratorian

R.2.1 Login: -
Description: Laboratorian Login to the system by entering valid email id and password
input: Laboratorian credential
output: home page

R.2.2 Add Lab Report: -
Description: Laboratorian Can add lab report of respective patient
input: - Report Selection
output: - Prompts whether reports added or not

R.2.3 Search Report
Description: Laboratorian can search reports by patient name or date
input: search input
output: search related data

R.3 Patient:

R.3.1 Register: -
Description: Patient can register itself by giving valid data.
input: patient basic details including blood group of patient
output: sign in page

R.3.2: -view and download health card
Description: patient can view as well as download his/her health card
input: - patient login credential
output: downloaded health card in pdf format

R.3. View Prescriptions
Description: Patient can view Prescription written by Doctor
Input: User Selection

Output: Display selected prescription

3.3 Other non-functional requirement

1)performance: -

Performance The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. In case of opening App components, of popping error messages and saying the settings or sessions there is delay much below 3 seconds.

2)safety: -

User details should be securely to be the server. The main security concern is for user account hence proper login mechanism should be used to avoid hacking.

3)Reliability: -

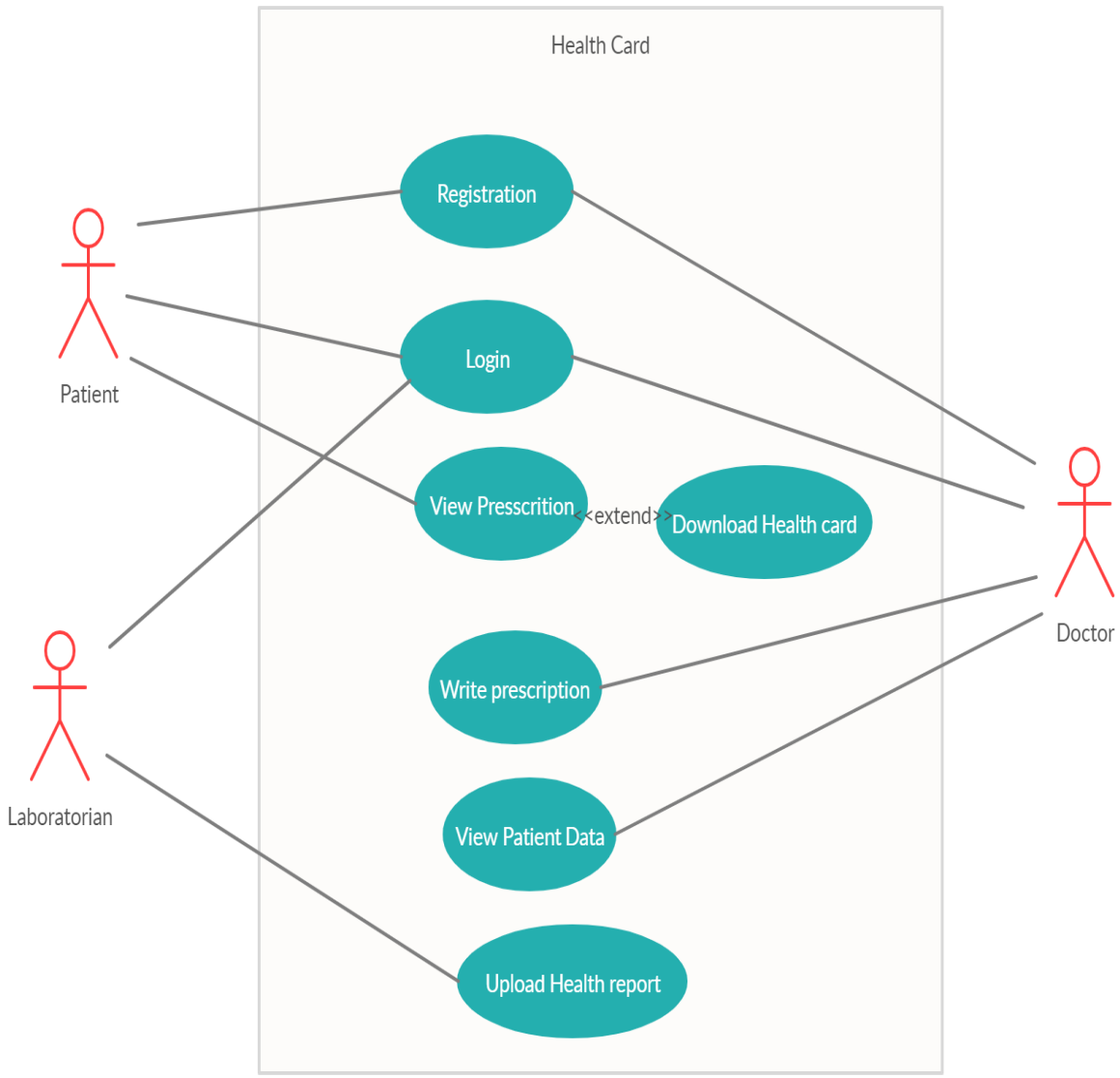
As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

4)Database: -

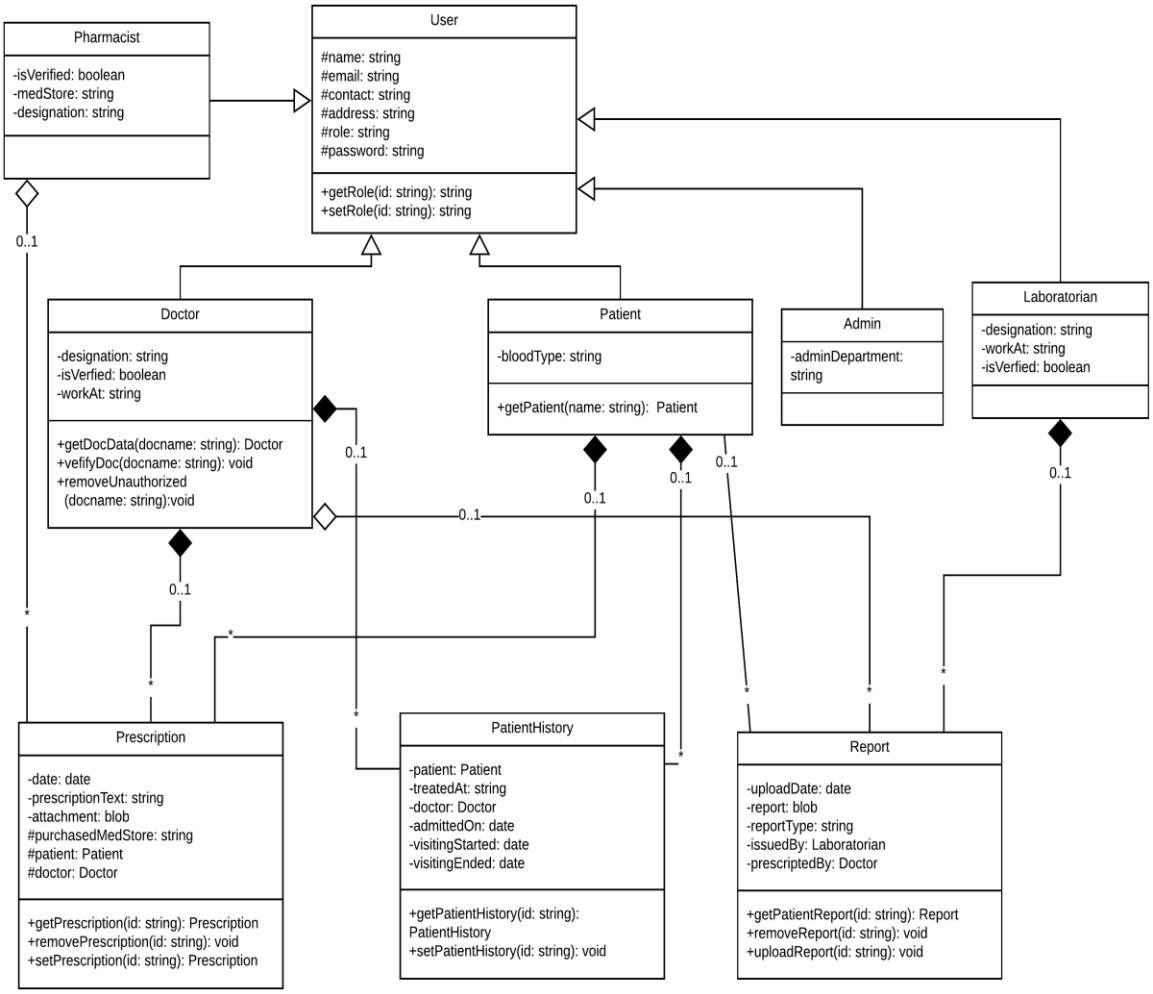
System requires to access users data fast to maintain the performance

4 Design

4.1 Use case Diagram

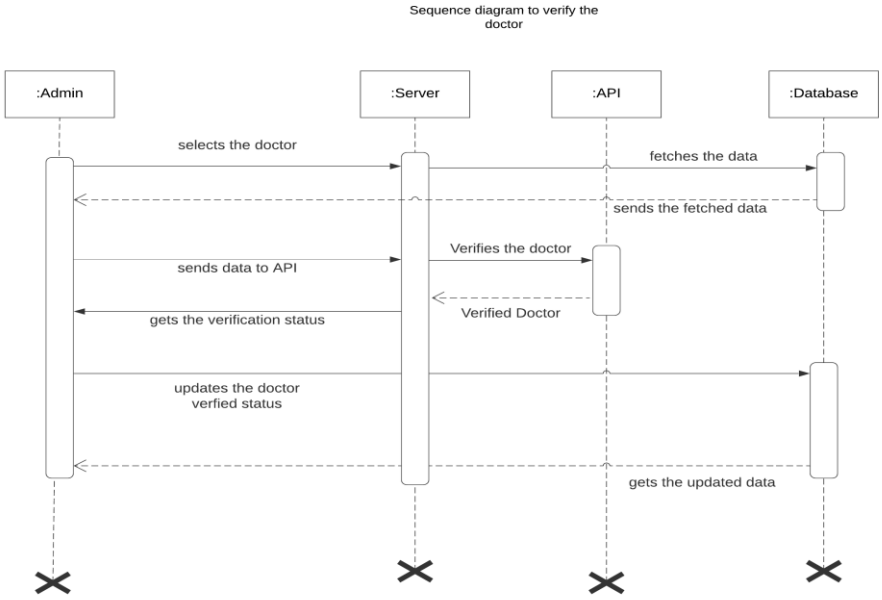


4.2 Class Diagram

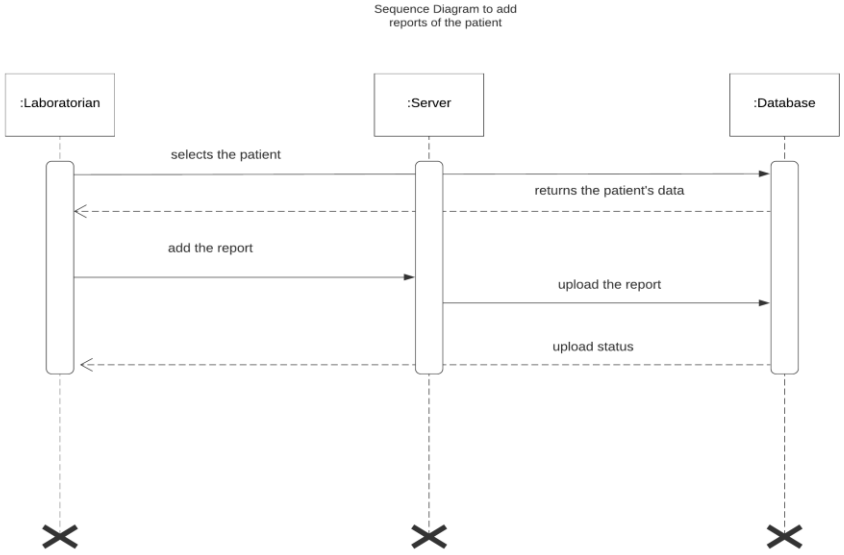


4.3 Sequence Diagram

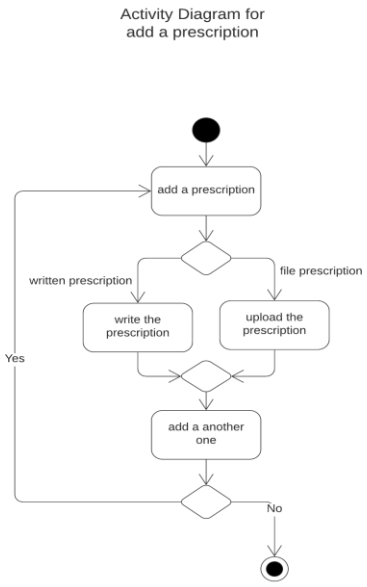
1)

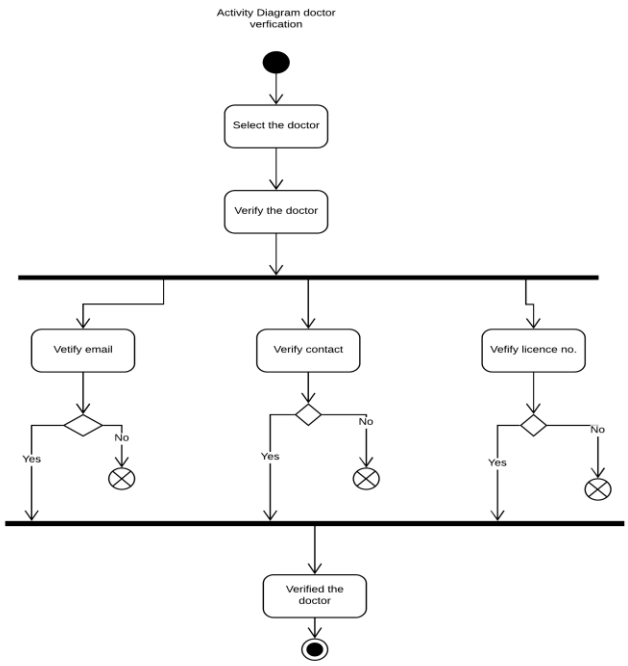


2)

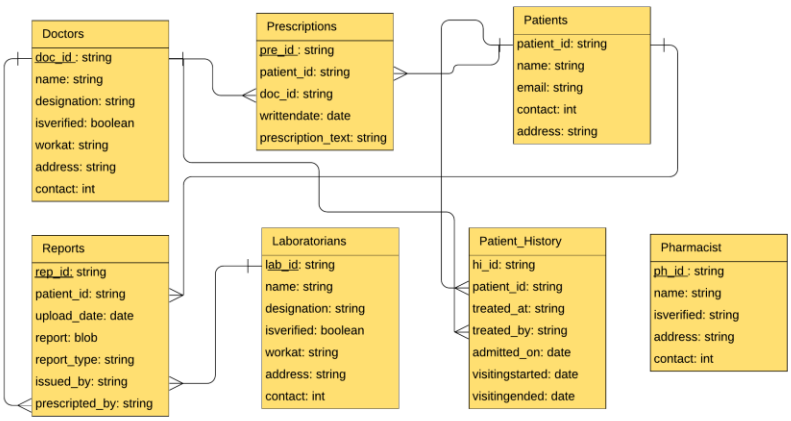


4.4 Activity Diagram





4.4 E-R Diagram



4.5 Data Dictionary Table

1) Doctors				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
Doctor_id	Number	PK		
Name	Number			
designation	Varchar2			
isvarified	Boolean			
Work-at	Varchar2			
address	Varchar2			
Contact	Number			

2)users				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
user_id	Number	PK		
Name	Varchar2			
Email	Varchar2			
Contact	Number			
Address	Varchar2			

3)Reports				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
Rep_id	Number	PK		
user_id	Number	FK	user	
Upload_date	date			
Report	blob			
Report_type	Varchar2			
Issued_by	Varchar2			
Prescripted_by	Varchar2			

4)prescription				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
Prescription_id	Number	PK		
user_id	Number	FK	user	
Doc_id	Varchar2			
Written_date	date			
Prescripton_type	Varchar2			

5)prescription_details				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
Prescription_detail_id	Number	PK		
Prescription_id	Number	FK	prescription	
Drug_name	Varchar2			
Drug_duration	Varchar2			
Morning	Boolean			
Afternoon	Boolean			
Evening	boolean			
After/before meal	Boolean			

6)Laboratorians				
Field Name	**Data Type**	**PK/Fk**	**Referenced Table**	**Description**
Lab_id	Number	PK		
Name	Varchar2			
Designation	Varchar2			
isVarified	Boolean			
Workat	Varcahr2			
Address	Varchar2			
Contact_no	Number			

5.Implemantation Details

5.1 Modules Description

- Registration Module**:**

  This Module is used to store user data to the database and enables the user to login to the system if he /she is authorized. All the fields in this module contain require fields. user can also navigate to login page if he/she has already registered.

- Login Module:

  This Module takes user credential and then verifies with registered user, if user is nor registered the invalid credential is shown else if they match with database and then login user.

- Upload report Module:

  This module only accessed by authenticated laboratian. It is users upload report page, in which lab report of individual patient has been uploaded.

- Prescription Module:

  This Module is accessed only by authenticated doctors, in which doctors can add prescription of particular patient by entering valid details of patient.

- View prescription Module:

  In this module patient can view prescription or his/her history which is main goal of project to manage that. Patient can view his/her prescription with all details

5.2 Function prototype

1)Register User: -

```
this.http
  .get(
    this.UserUrl+"/getUserId/" +
      fname +
      "/" +
      lname +
      "/" +
      user +
      "/" +
      dob
  )
  .subscribe((response: any) =>
```

2)Login

```
login(uname, password) {
  this.http
    .post(this.LoginUrl+"/login", { uname, password })
    .subscribe((response: any) => {
```

3)Setting Session: -

```
sessionStorage.setItem("uid",response.userId);
```

4)Authentication: -

```
{
  path: "Prescription/Add",
  component: AddPrescriptionComponent,
  canActivate: [DoctorGuard]
},
```

6 Testing

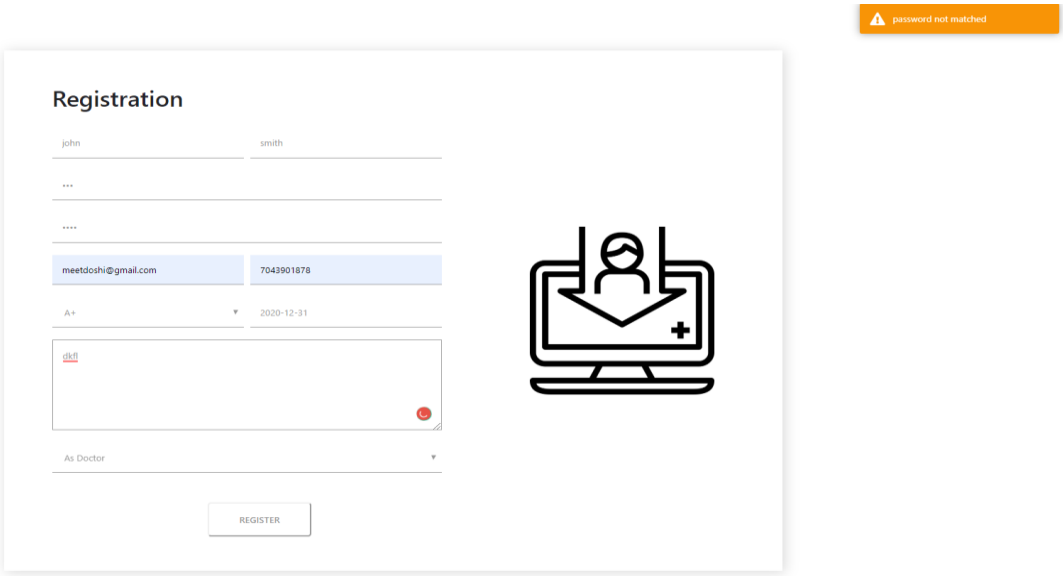
6.1 Testing Method:

We have Performed Black box testing For the testing purpose

6.2 Test Cases:

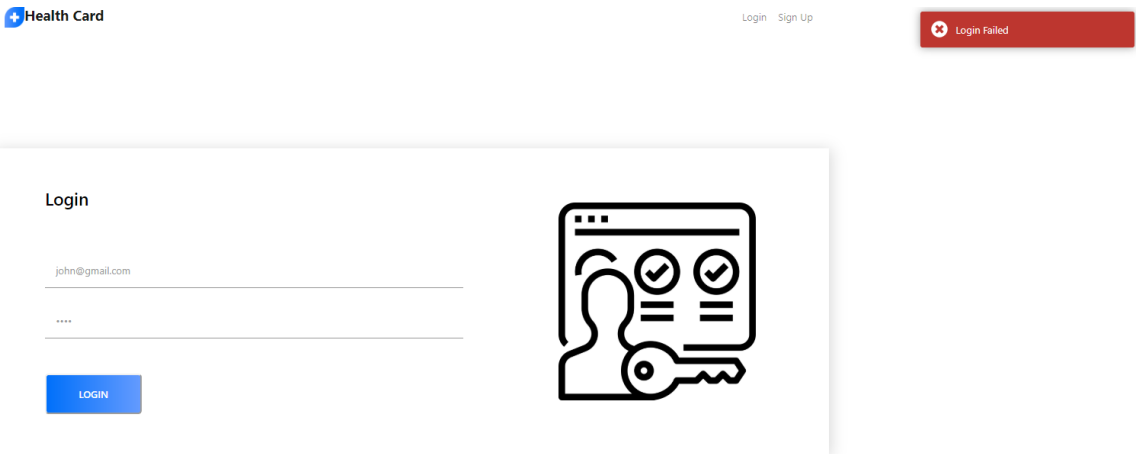
For Registration:

Password and confirm password must be same



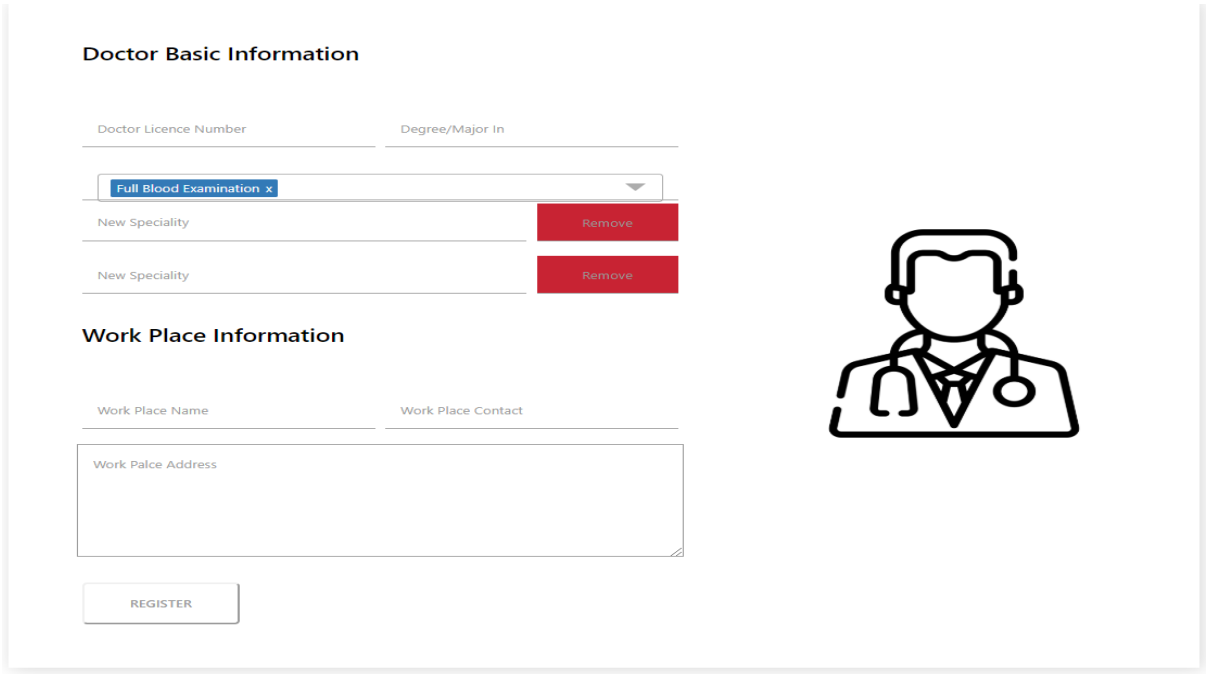
Login:

If credentials Are Invalid Error Message will be shown



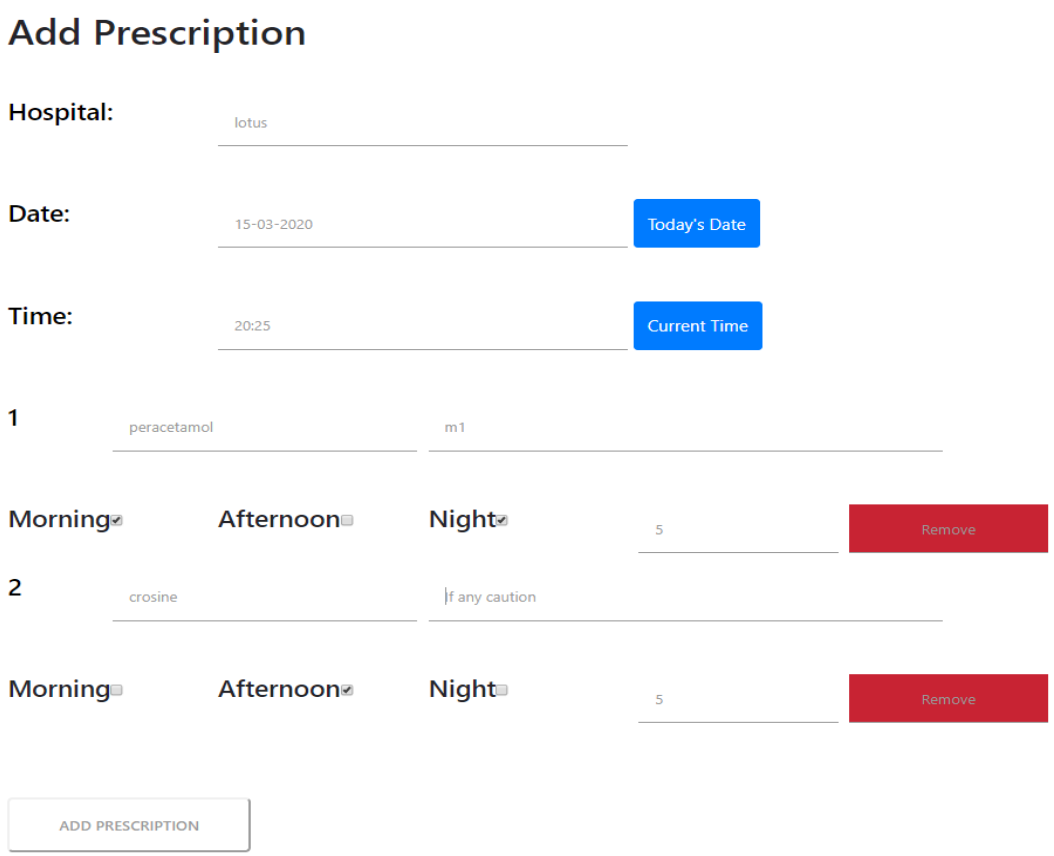
7 Screenshots

1)Doctor Registration

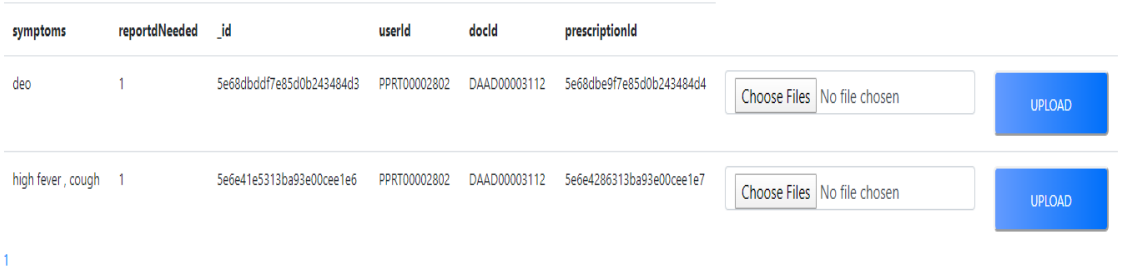


2)Add new Diagnosis

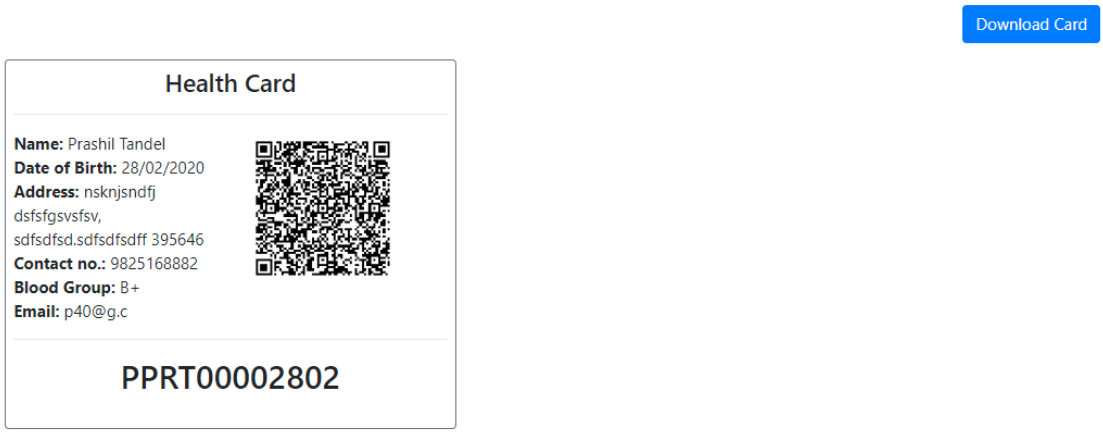
3)Add prescription



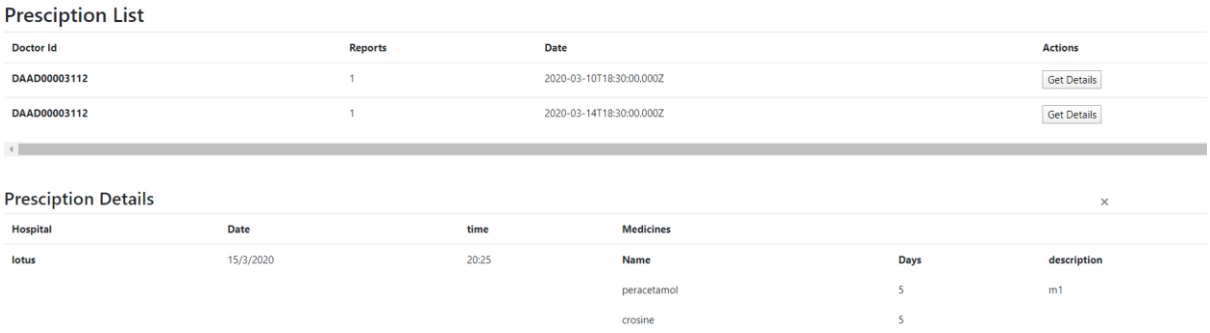
4)Upload Report



5)Patient Health card:



6)patient view prescription:



Conclusion

The functionality implemented in the system was done after understanding all the system modules according to requirements.

Functionality that are successfully implemented in the system are:

- Patient/Doctor/Lab Registration
- Login
- Session management
- Upload lab reports and searching
- Add diagnosis
- View / download health card
- View prescription of patients
- Log out

After the implementation and coding of system , comprehensive testing was performed on the system to determine the loopholes and possible flaws in the system

Limitations and Future Enhancements

- **Limitations**
  - Every time when doctor add prescription, he/she needs to add drug name
  - Not verified doctors are also allowed to use system

- **Functionality not implemented**
  - Laboratorian cannot view uploaded reports
  - User cannot change his profile name once Registered

- **Future Extension**
  - Making module of medical shop owner
  - By gathering all health data train, a model and analysis of allergies area wise
  - Registration of patient via Aadhar card

Bibliography

➢ Websites:-

1.https://www.w3schools.com–For Html,CSS,Bootstrap,XML,Javascript.

2 https://mongoosejs.com/ - For mongoose database

3.https://angular.io –For Understanding Angular.

4.https://stackoverflow.com –For Solving problems.

➢ Useful Links

1. https://nodejs.org/en/ -For download and setup of NodeJS.

2. https://fonts.google.com –For getting fonts families.

3 https://www.tutorialspoint.com/index.htm