

**Dharmsinh Desai University, Nadiad**  
**Faculty of Technology**  
**Department of Computer Engineering**



**B. Tech. CE Semester – VI**

**Subject: Service oriented Computing**

**Project Title: Hospital Management System using web api**

**Developed By:**

- 1. Kaushik Chaudhary (CE-022)**
- 2. Dhaval Chavda (CE-025)**
- 3. Dhruv Kakadiya (CE-055)**

**Guided by:**

**Prof. Prashant M. Jadav**

**Prof. Apurva A. Mehta**

# Certificate



This is to certify that the project entitled “Hospital Management System” is a Bonafide report of the work carried out by

Mr. Kaushik Chaudhary, Student ID No: 18CEUBS042,  
Mr. Dhaval Chavda, Student ID No: 18CEUBT014,  
Mr. Dhruv Kakadiya, Student ID No: 18CEUOS032

of Department of Computer Engineering, semester VI, under the guidance and supervision for the subject Service Oriented Computing. They were involved in Project training during academic year 2020-21.

1. Prof. Prashant M Jadav  
Department of computer Engineering,  
Faculty of technology,  
Dharmsinh Desai University,  
Date: 17/02/2021
2. Prof. Apurva A. Mehta  
Department of computer Engineering,  
Faculty of technology,  
Dharmsinh Desai University,  
Date: 17/02/2021
3. Prof C.K Bhensdadia  
Head, Department of Computer Engineering,  
Faculty of Technology,  
Dharmsinh Desai University,  
Date: 17/02/2021

## Contents

1. Abstract
2. Introduction
  - Brief introduction
  - Technology and tools
3. Software Requirement Specification
4. Implementation Detail
  - E-R Diagram
  - Class Diagram
  - Use Case Diagram
  - State Diagram
  - Component Diagram
  - Package Diagram
  - Sequence Diagram
  - Activity Diagram
  - Modules and Description
  - Function Prototype
5. Testing
  - Testing the system and Description
6. Screen shots
  - Some major screenshots
7. Conclusion
  - Description of functionality
8. Limitation and future Extension
  - Limitation of project
  - Functionality which was not implemented
  - List of future extensions
9. Bibliography

# **1. Abstract**

- This project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the clinical details of every patient and hospital tests done automatically. It includes a search facility to know the current status of each patient. User can search details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

## **2. Introduction**

### **1. Brief Introduction:**

- Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-today activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully. But keeping track of all the activities and their records on paper is very cumbersome and error prone.
- It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper. Thus, keeping the working of the manual system as the basis of our project.
- We have developed an automated version of the manual system, named as “Hospital Management System”. The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user system interaction and also provides robust & reliable storage and backup facilities.
- Advantages:
  - Save time
  - Save Energy
  - Record of Data
  - 24/7 Availability
  - No waiting in Lines
  - Easy to search

### **2. Tools and Technologies:**

- WCF Service and Web APIs
- C# Programing Language
- ASP.NET Framework

### **3. Software requirements specification (SRS) for “Hospital Management System”**

Prepared by

- 1. Kaushik Chaudhary (CE-022)**
- 2. Dhaval Chavda (CE-025)**
- 3. Dhruv Kakadiya (CE-055)**

Proposed to

**Prof. Prashant M. Jadav**  
**Prof. Apurva A. Mehta**

**01/04/2021**

## **Table of Contents**

### **1. Introduction**

1.1 Introduction

1.2 Document convention

1.3 Consumers, Definitions, Acronyms and Abbreviations

1.3.1 Audience Definitions

1.3.2 Acronyms and Abbreviations

1.4 Technologies to be used

### **2. Overall Description**

### **3. Specific Requirements**

3.1 Functional Requirements

3.1.1 Admin Module

3.1.2 Patient Module

3.2 Non-functional Requirements

3.2.1 Performance Requirements

3.2.2 Safety Requirements

3.2.3 Security Requirements

3.2.4 Error handling

4. Interfaces and Possible Scenarios

4.1 Customer's Interface

4.1.1 Login

4.1.2 Registration Interface

4.1.3 Personal Data Editing

4.1.4 Search

4.1.5 Appointments

4.2 Interface for Administrator

## **1. Introduction:**

Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully. But keeping track of all the activities and their records on paper is very cumbersome and error prone.

It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error prone. It is also not economically & technically feasible to maintain these records on paper. Thus, keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Hospital Management System”. The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user system interaction and also provides robust & reliable storage and backup facilities.

### **1.1 Purpose**

- The Software is for the automation of Hospital Management.
- It maintains two levels of users
  1. patient Level
  2. Admin level
- The Software includes Maintaining Patient details.
- Providing Prescription, Precautions and Health advice.
- Providing and maintaining all kinds of tests for a patient.

### **1.2 Document conventions**

- HMS: Hospital Management System
- LM: Login Module
- RUM: Registered Users Module
- NUM: Normal Users Module
- AM: Administrator Module
- DB: Database
- DM: Doctor module
- ADM: Admin module



## 1.3 Audience Definitions, Acronyms and Abbreviations

### 1.3.1 Audience Definitions

The intended readers of this document are the developers of the site, testers, website owners, managers and coordinators.

### 1.3.2 Acronyms and Abbreviations

Acronym	Meaning
HM	Hospital management System
WCF	Windows communication Foundation
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol

## 1.4 References

- IEEE 830-1998 standard for writing SRS document.
- *Fundamentals of Software Engineering*, 2<sup>nd</sup> ed. by Mr. Rajib Mall.

## 1.5 Technologies to be used

- Programming languages: C#
- WCF Service and Web APIs,
- ASP.NET Framework

### Tools & Development Environment

- SQL Database

## **2. Overall Description**

### **Goals of proposed system:**

- 1. Planned approach towards working:** The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.
- 2. Accuracy:** The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.
- 3. Reliability:** The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
- 4. No Redundancy:** In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
- 5. Immediate retrieval of information:** The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires
- 6. Immediate storage of information:** In manual system there are many problems to store the largest amount of information.
- 7. Easy to Operate:** The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

### **3. Specific Requirements:**

#### **3.1 Functional Requirement**

##### **3.1.1 Admin module:**

###### **R1: Login**

**Input:** Admin-Id and password

**Output:** success or failure message

**Description:** Admin can use this page to login into their admin accounts and will be redirect to the home page of admin module.

**Constraints:** Admin-Id must be contained at least 6 characters and password must contain at least one number, one special character and length of password must be at least 8 character.

**Process:** Check Admin-id and match password in database, if found in database then it will redirect to the admin module pages.

**Precondition:** Admin must be registered into the system.

###### **R2: Add details of doctor**

**Input:** doctor details

**Output:** Success or failure message

**Description:** Admin can use this page to enter details of hospital doctor.

**Constraints:** Admin must be entered required details of respective doctor.

**Process:** After entering all required details of doctor, data of that doctor will be saved in database and show success message.

**Precondition:** Admin must be logged in into the system

###### **R3: Delete details of Doctors**

**Input:** Doctor-Id

**Output:** Success or failure message

**Description:** Admin can use this page to delete details of hospital Doctors.

**Constraints:** Admin must be entered Doctor-Id of respective Doctors.

**Process:** After entering Doctor-Id, data of that Doctor will be deleted in database and show success message.

**Precondition:** Admin must be logged in into the system.

###### **R4: Update Doctor details**

**Input:** doctor details

**Output:** Success or failure message

**Description:** Admin can use this page to update details of hospital doctors.

**Constraints:** Admin must be entered details of Doctor which they have to update.

**Process:** Details of staff will be updated into database then details of doctor will show to admin.

**Precondition:** Admin must be logged in into the system.

### 3.1.2 Patient module:

#### **R1: Signup/Register**

**Input:** Patient Details which are required

**Output:** success or failure message

**Description:** Patient can use this page to signup into their Patient accounts and will be redirect to the home page of Patient module.

**Constraints:** Patient-Id must be contained at least 6 characters and password must contain at least one number, one special character and length of password must be at least 8 character.

**Process:** Check the constraints of Patient-Id and password, then it details of patient will save into database

#### **R2: Login**

**Input:** patient-Id and password

**Output:** success or failure message

**Description:** Patient can use this page to login into their Patient accounts and will be redirect to the home page of Patient module.

**Constraints:** patient-Id must be contained at least 6 characters and password must contain at least one number, one special character and length of password must be at least 8 character.

**Process:** Check patient-id and match password in database, if found in database then it will redirect to the Patient module pages.

**Precondition:** Patient must be registered into the system.

#### **R3: Update details**

**Input:** patient details

**Output:** Success or failure message

**Description:** patient can use this page to update his/her details.

**Constraints:** patient must be entered details of his/her which have to update.

**Process:** Details of patient will be updated into database then details will show to patient.

**Precondition:** Admin must be logged in into the system.

#### **R4: Book appointment**

**Input:** Appointment details

**Output:** Success or failure message

**Description:** patient can use this page to book appointment with doctor.

**Constraints:** patient must be entered details of his/her appointment with any doctor.

**Process:** Details of Appointment will update in database and that shown by doctor and they approve or reject that appointment.

**Precondition:** Patient must be logged in into the system

#### **R4: View Doctor List**

**Input:** Click button

**Output:** List of all doctors

**Description:** patient can use this page to see the list of all doctors.

**Constraints:** patient must be click on the button “Doctor List”.

**Process:** Details of all Doctors will be shown on page.

**Precondition:** Patient must be logged in into the system

## **3.2 Non – Functional Requirement**

### **3.2.1 Performance Requirements**

The system shall accommodate high number of items and users without any fault. Responses to view information shall take no longer than 5 seconds to appear on the screen.

### **3.2.2 Safety Requirements**

System use shall not cause any harm to human users.

### **3.2.3 Security Requirements**

System will use secured database

Normal users can just read information but they cannot edit or modify anything.

System will have different types of users and every user has access constraints.

### **3.2.4 Error handling**

OSS shall handle expected and nonexpected errors in ways that prevent loss in information and long downtime period.

## **4. Interfaces Possible Scenarios**

### **4.1 Patient's Interface:**

#### **4.1.1. Login:**

This interface will consist of two compulsory fields namely, "Email I" and "Password". There will also be options for "New Patient's Registration" which will redirect to "Registration" page and a "Forgot Password" option in case a user forgets the password.

If the password entered is correct the Main User Interface opens up else an error message is displayed.

#### **4.1.2. Registration Interface**

The user will enter his personal details like Name, Email Id, Password, Date Of Birth, Contact Number, Gender, etc.

Users will be warned about any mistakes on data format or any other constraints by validation notes and error messages.

When the button "save" button is clicked, the server will check if the username or email is already taken and alert the user.

If everything is entered correctly and saved a new user will be created.

#### **4.1.2. Personal Data Editing**

If any member wants to change his personal information he can enter his profile by clicking on his name at the top right of the main page and he will be directed to the personal details editing page.

#### **4.1.3. Appointment**

This will be a space for the Patient where he/she can book the appointment he/she to doctor. The user can also remove appointment from appointment prior to conformation. Once the patient decides to book appointment, the patient is directed to the payment page for making payment.

## **4.2 Interface for Administrator**

The administrator will have a different login id using which he can access his account that contains a control panel that allows him to contact each and every aspect of the system.

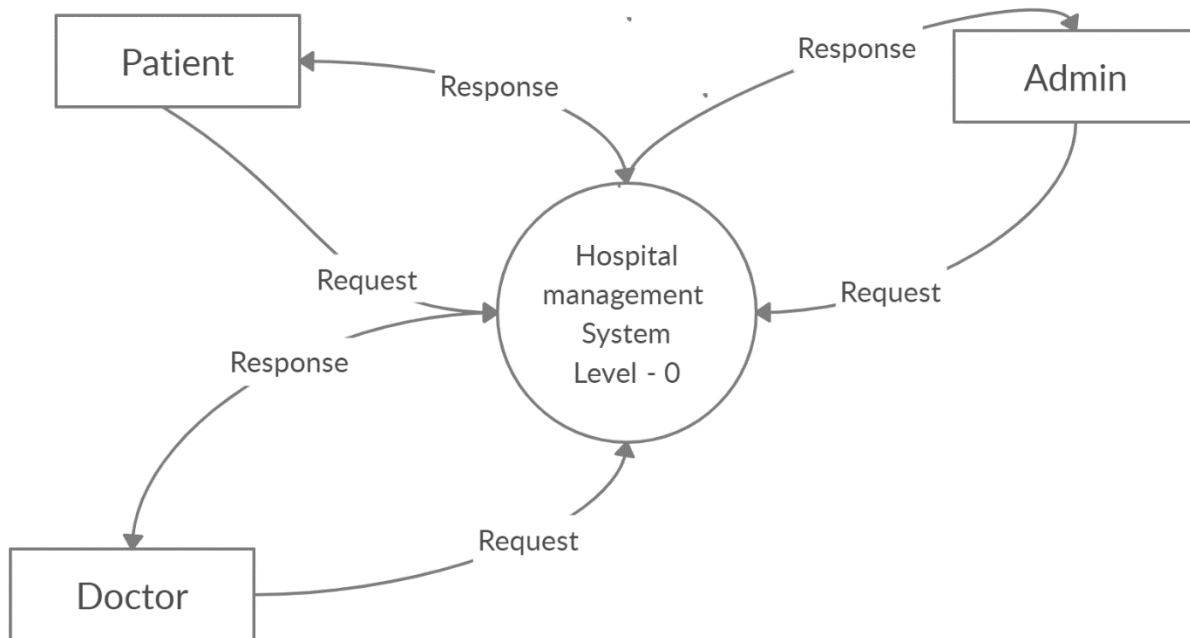
This control panel will allow the administrator to do the following things:

- Access and view the customer database.
- Access and view the doctor's database and manage them.

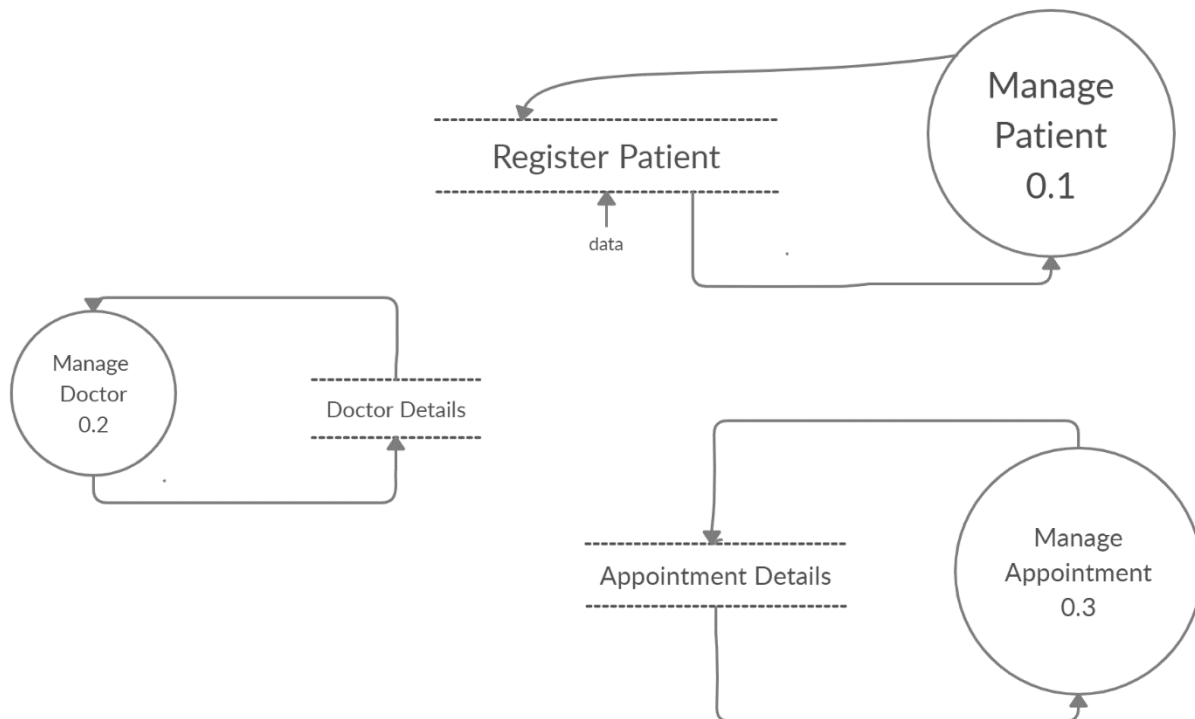
## 4. Implementation Details

### 1. DFD:

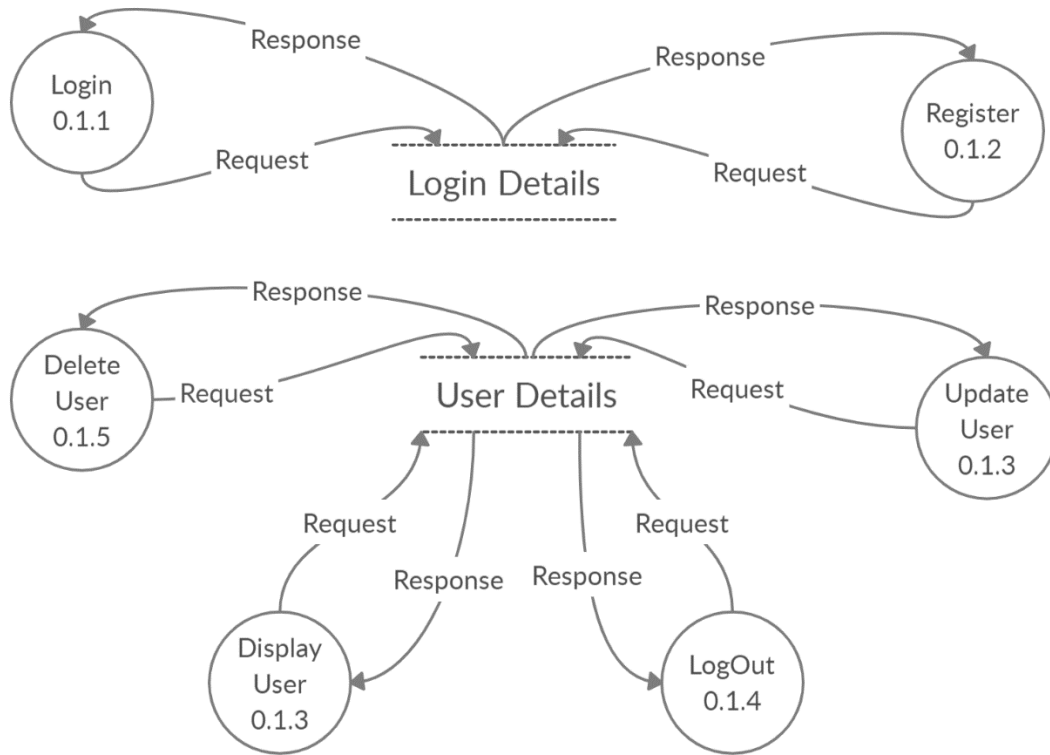
#### Level - 0



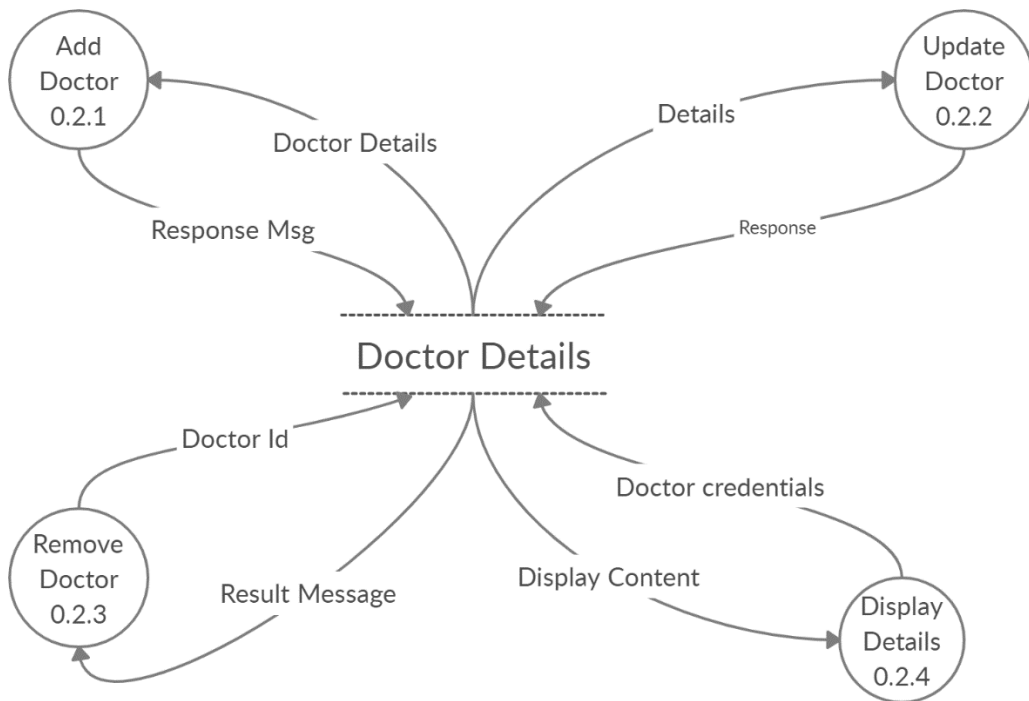
#### Level - 1



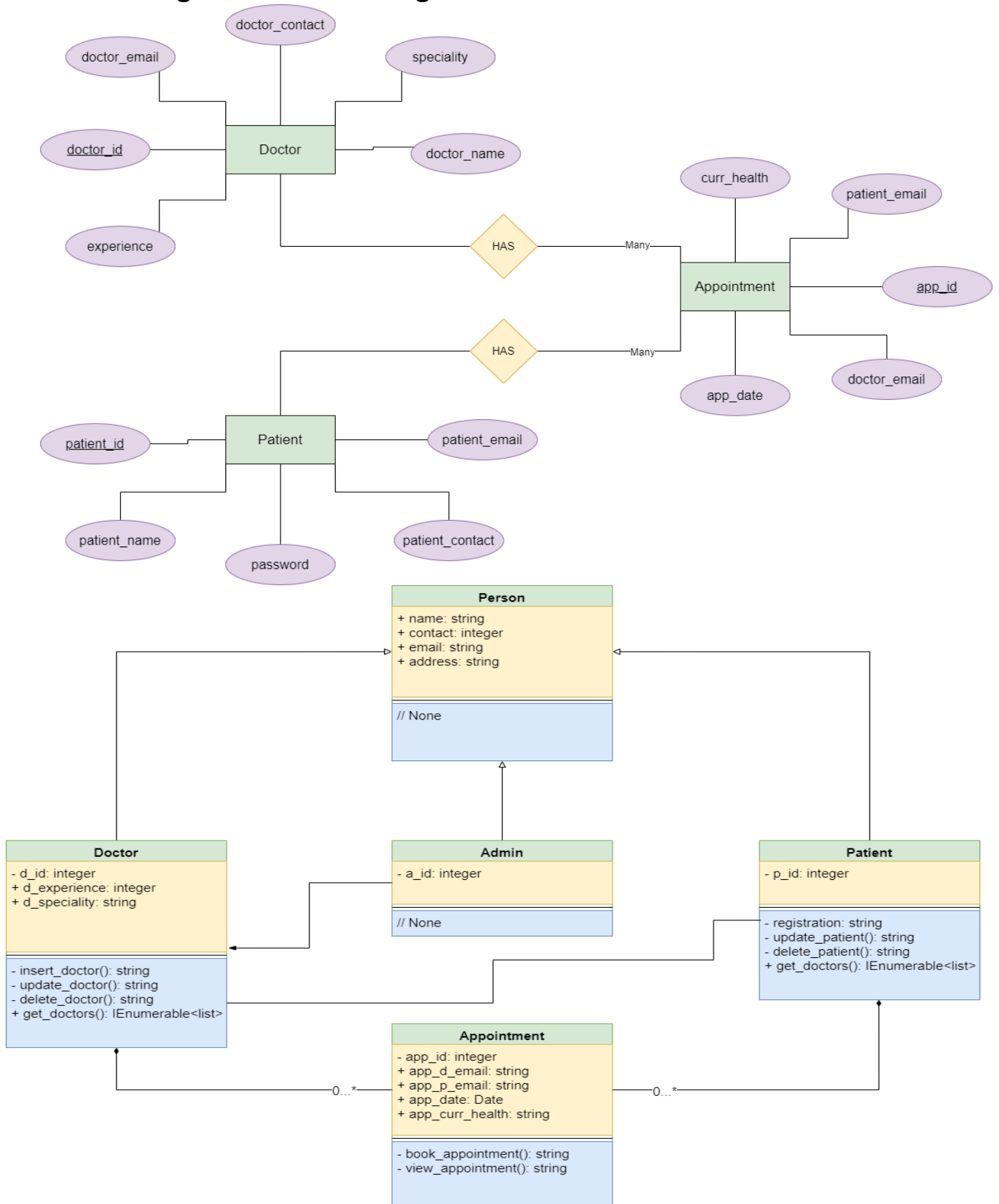




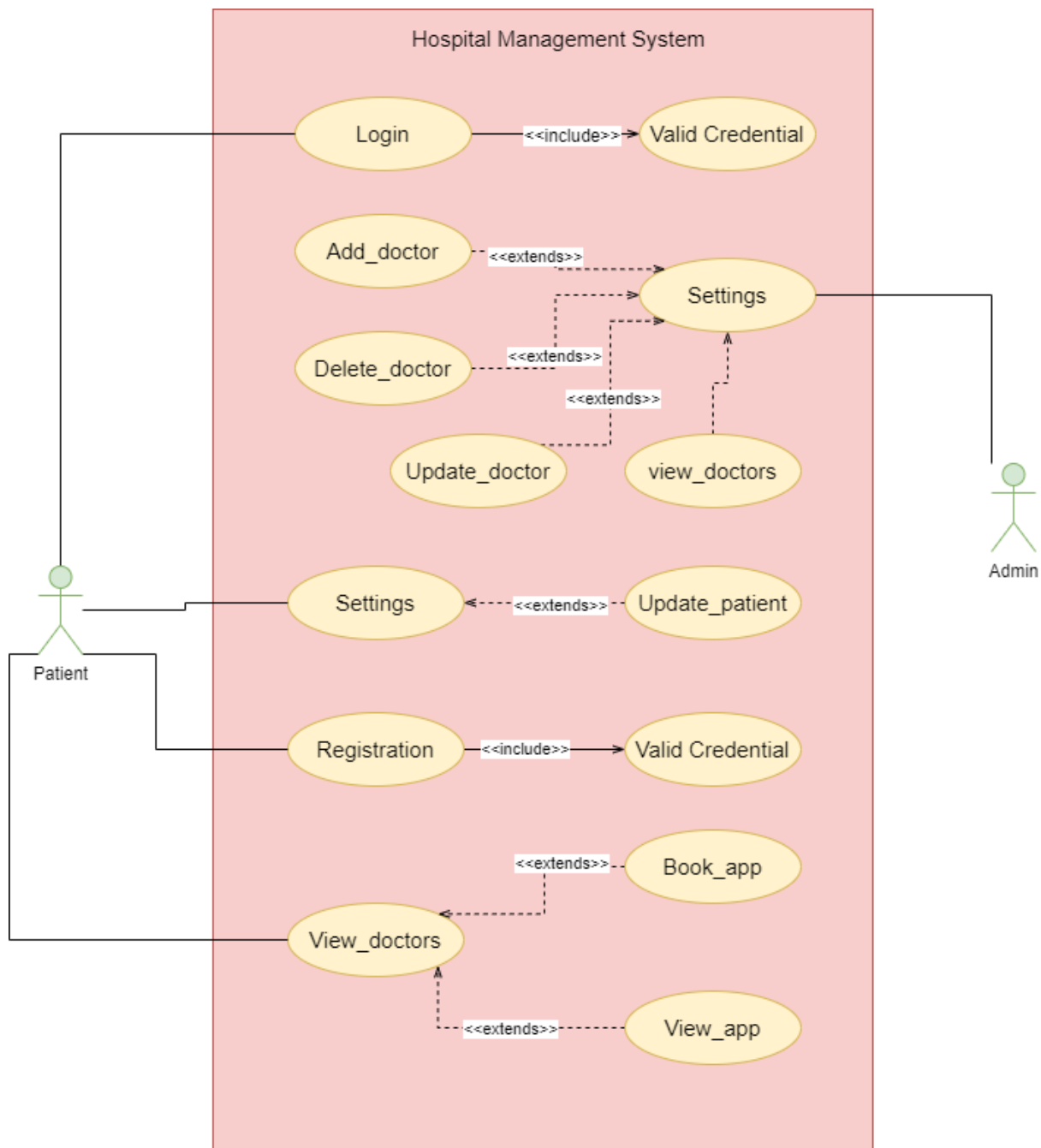
## Level-2



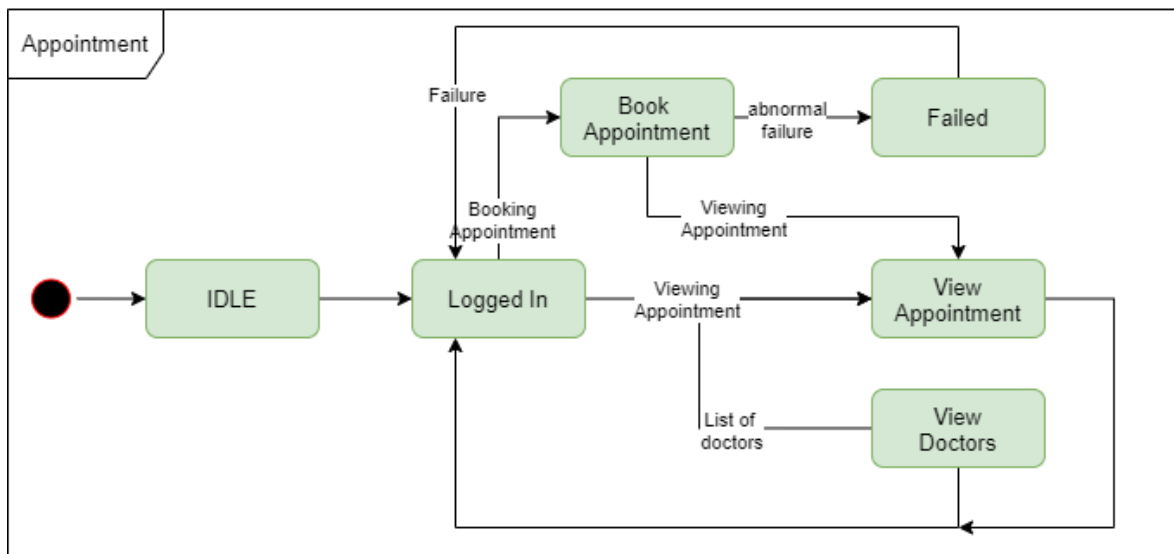
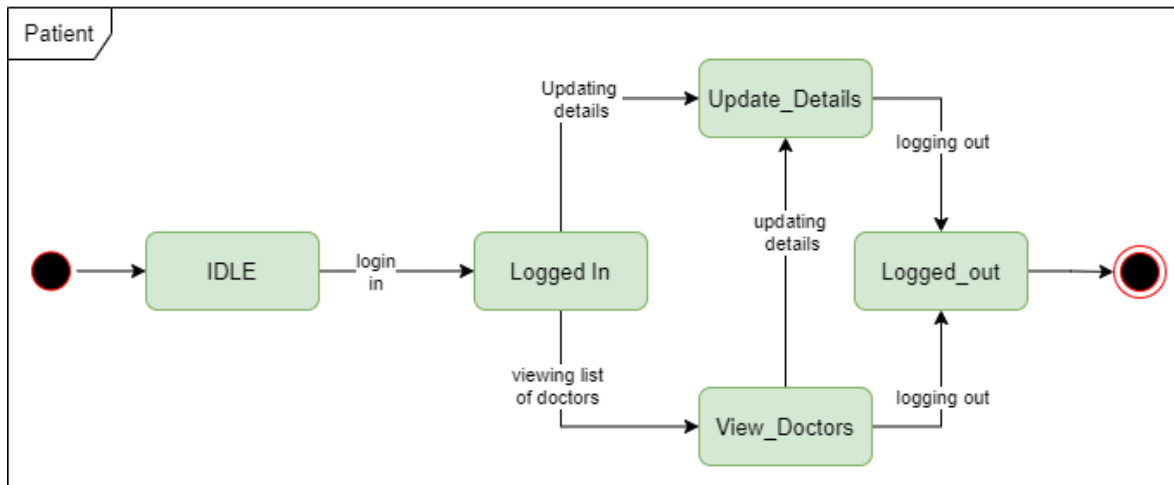
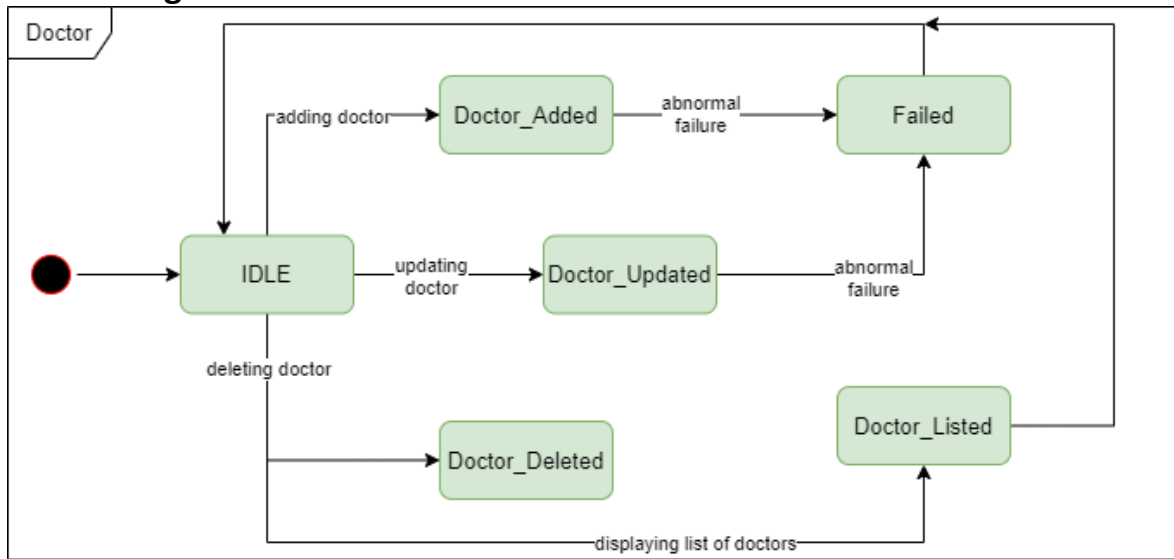
## 2. ER – Diagram and Class Diagram:



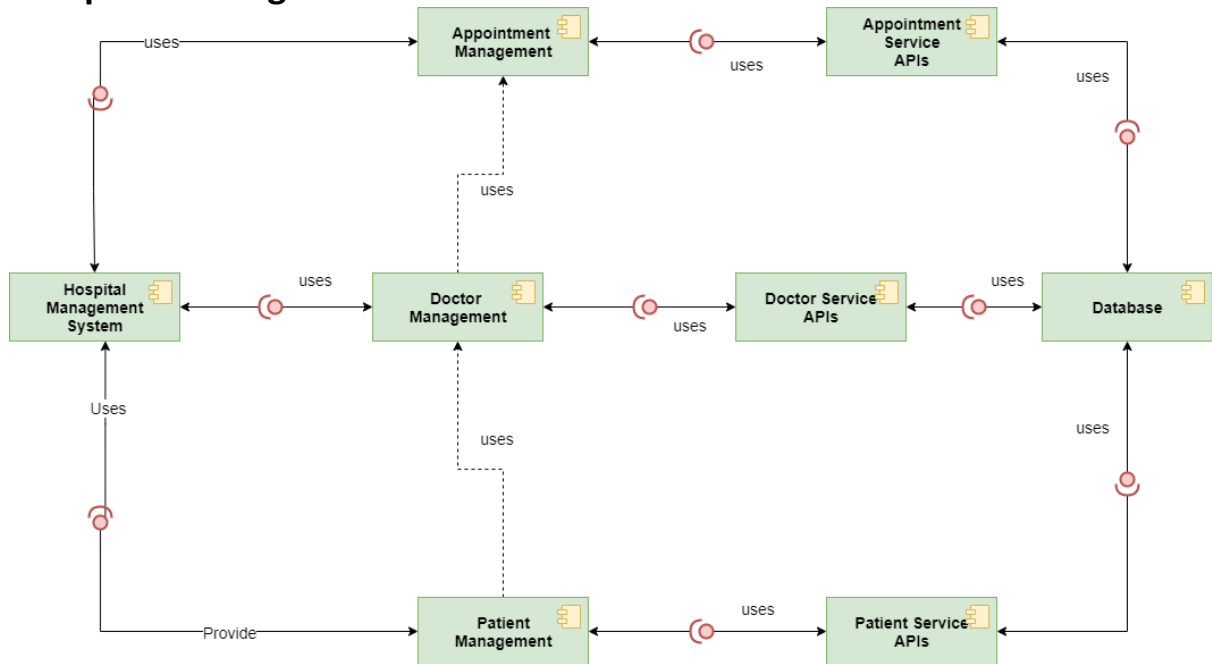
### 3. Use Case Diagram:



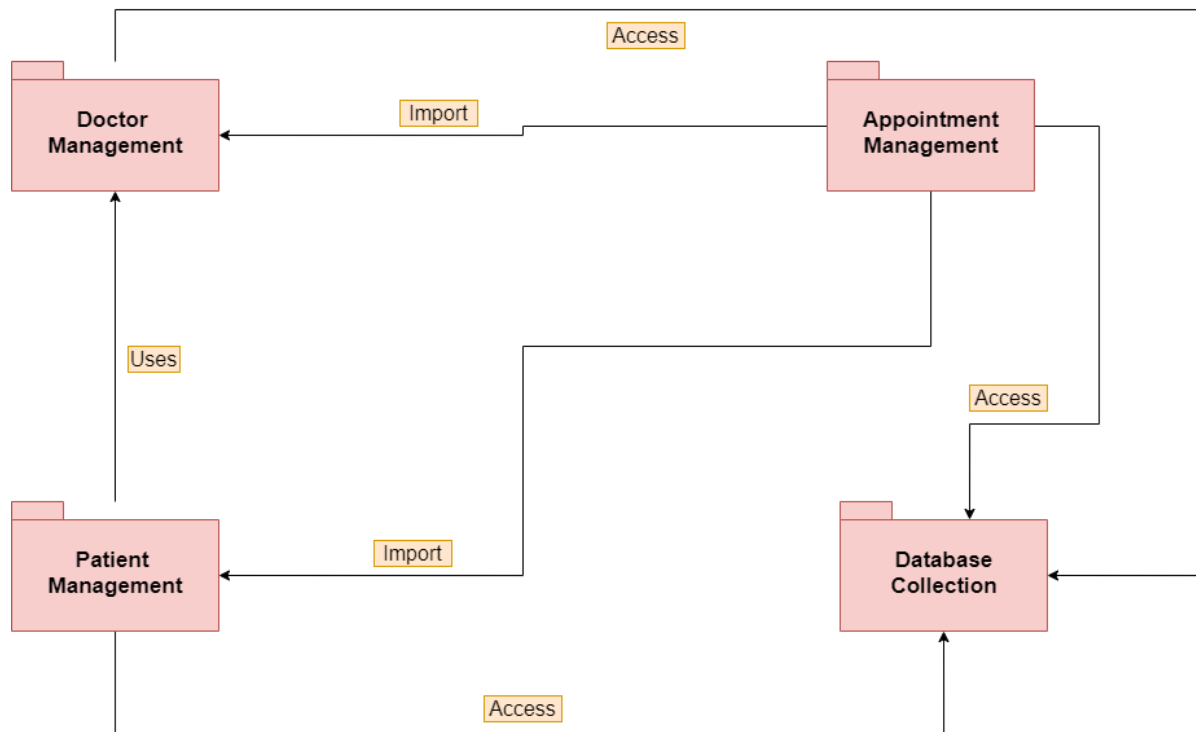
#### 4. State Diagram:



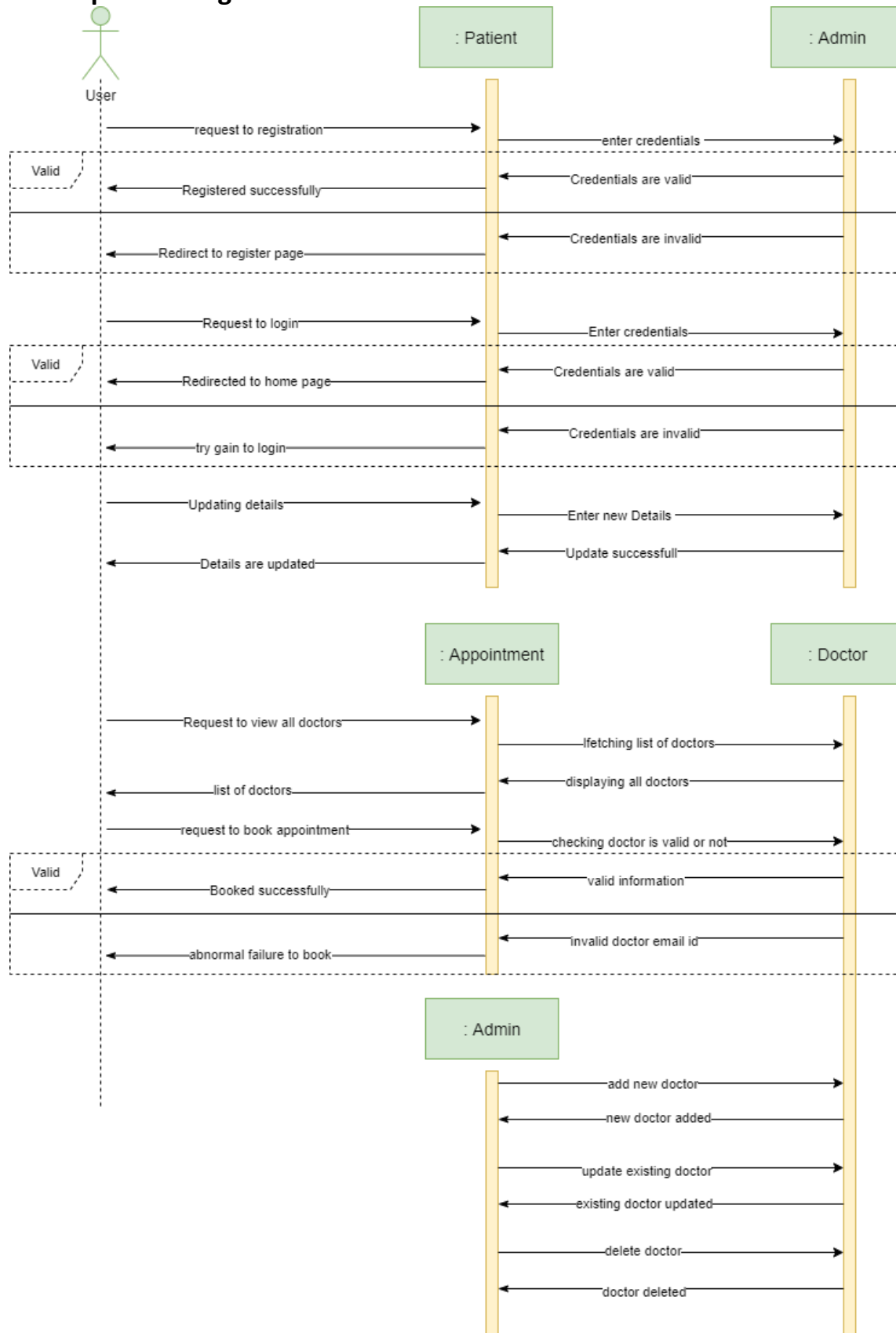
### 5. Component Diagram:



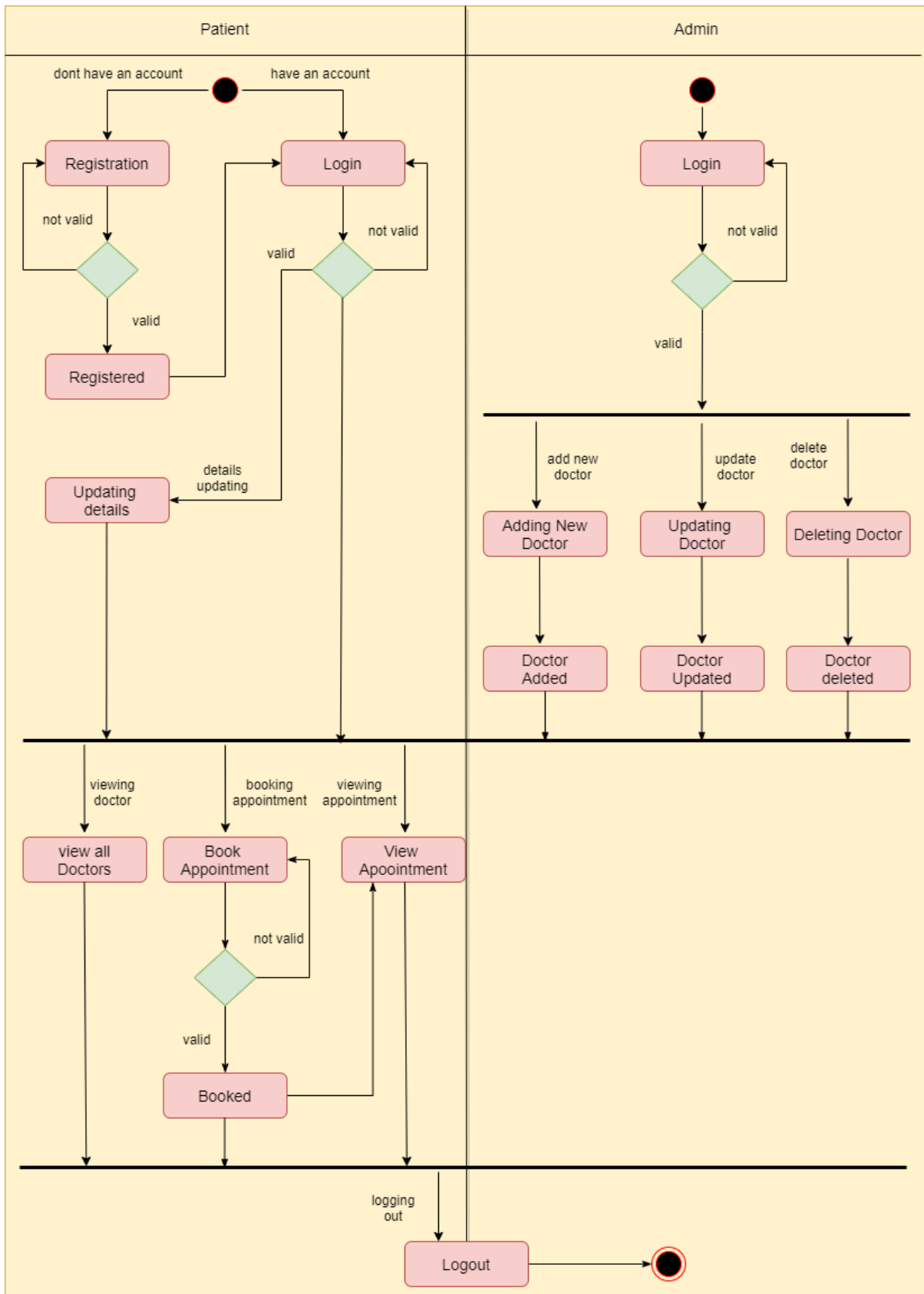
## 6. Package Diagram:



## 7. Sequence Diagram:



## 8. Activity Diagram:



- **Module Description:-**

**1. Login Module:**

- This Module is used to Authenticate the user by its email and password and allow to that user to login itself and give access of all the customer functionality of website.
- Input: User Credential
- Output: Success message
- Processing: Authenticate the user from database and give access.

**2. Registration Module:**

- This Module is used to Register the new guest or visitor to Register itself, All the field in this module contain required validations. User can also navigate to login page if he/she already registered.
- Input: User Information
- Output: Success message
- Processing: validating user data and then storing them into database

**3. Update Module:**

- This Module is update the details of Registered login and if and only if he/she logged in.
- Input: Changes in information
- Output: success message
- Processing: check the value of old password and changes the information of that user.

**4. Doctor Module:**

- All the doctors are in this module
- Input: View doctors
- Output: Show Doctor
- Processing: getting doctor from database

**5. Auth Module:**

- Combination of two modules
- 1. Login Module
- 2. Register Module



## 6. Book Appointment Module:

- This Module is used to book Appointment to doctor by clicking the Book-Appointment button from doctor page, Login user can set appointment date and time by this module.
- Input: click Book-Appointment button
- Output: doctor in appointment page
- Processing: Getting doctor from data source and put it in appointment.

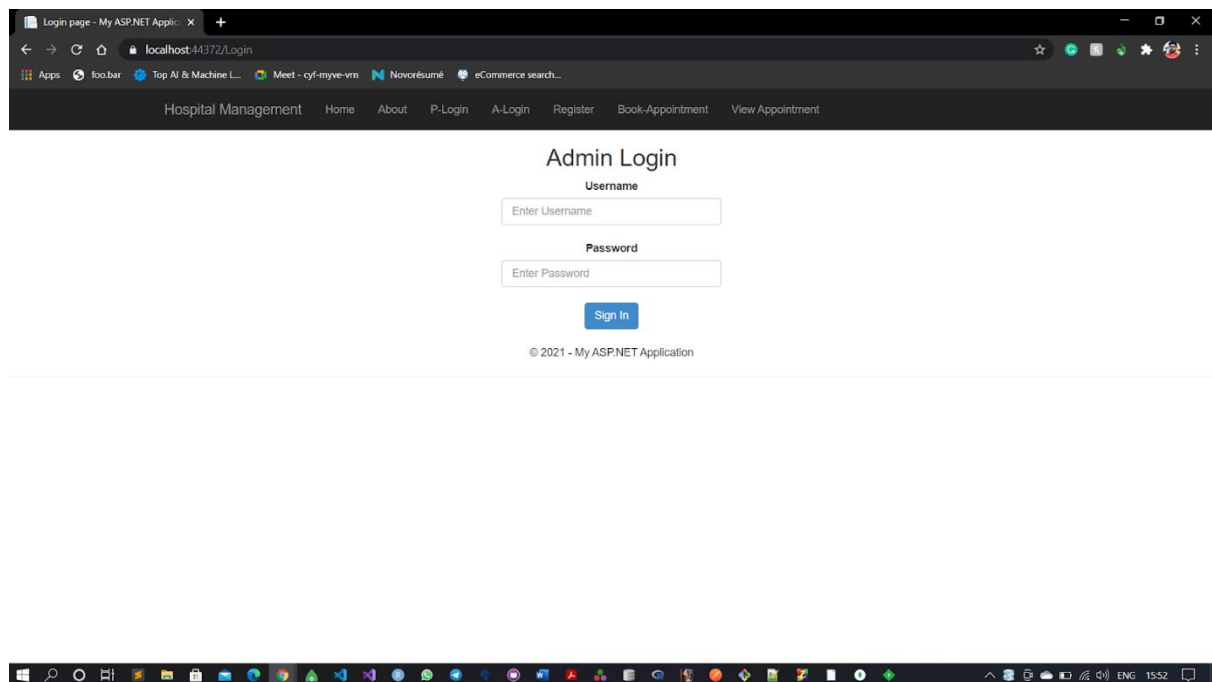
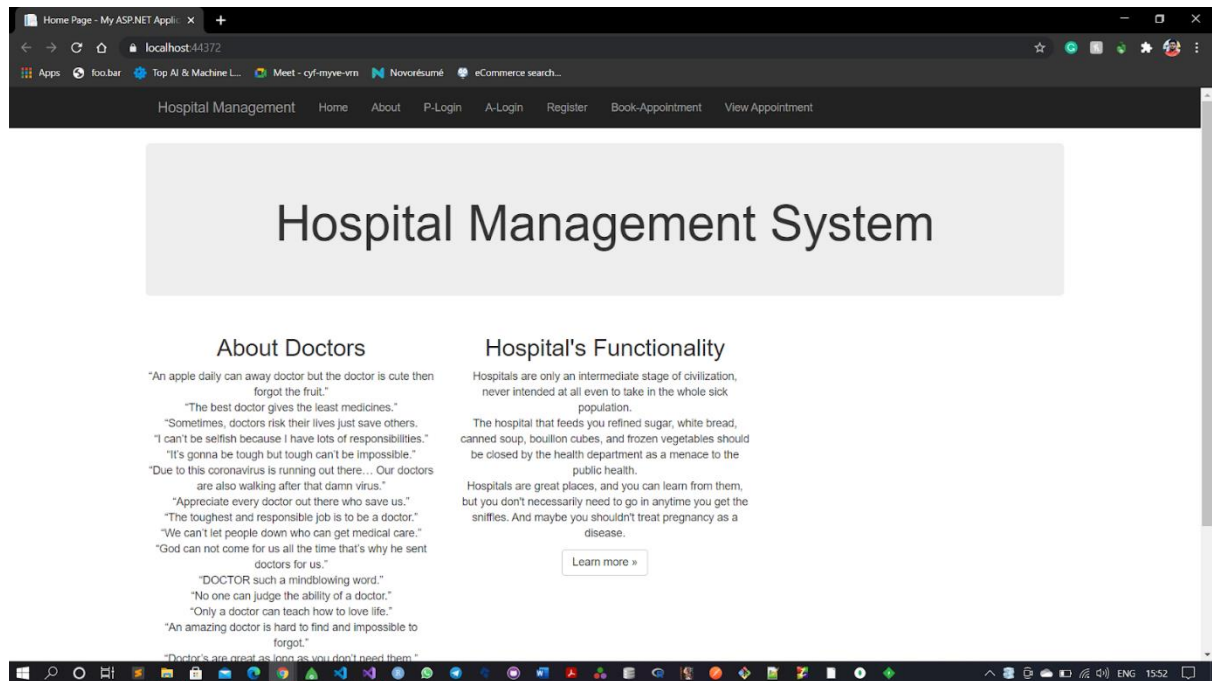
### • Function Prototype:

```
C# prototype.cs
1  // for patient
2  [HttpPost]
3  public HttpResponseMessage insertPatient(PatientModel patient)
4
5  [HttpPut]
6  public HttpResponseMessage updatePatient(PatientModel patient)
7
8  [HttpPost]
9  public HttpResponseMessage validatePatient(AuthModel am)
10
11
12 // For Doctor
13 [HttpGet]
14 public IHttpActionResult getDoctors()
15
16 [HttpPost]
17 public string addDoctor(DoctorModel dm)
18
19 [HttpPut]
20 public HttpResponseMessage updateDoctor(DoctorModel dm)
21
22 [HttpDelete]
23 public HttpResponseMessage deleteDoctor(int did)
24
25
26 // for appointment
27 [HttpPost]
28 public HttpResponseMessage bookAppointment(AppointmentModel bapp)
29
30 [HttpGet]
31 public IHttpActionResult getAllAppointment(string pid)
```

## 5. Testing

- For testing purpose, we use two software:
- 1. Postman
- 2. Fiddler

## 6. Screen-Shots of Project



## All Doctors List

All Doctors

d_id	d_name	d_email	d_address	d_contact	d_experience	d_speciality
8	Elon R Musk	elon@gmail.com	Mars Earth	999999999	35	Cars
11	TheDz	dhruv@gmail.com	USA	333222111	9	Nose

© 2021 - My ASP.NET Application

## Add Doctor

Doctor's Name

Kaushik

Email-ID

kaushik@gmail.com

Contact No

78963214

Address

Ahemdabad, Gujarat, India

Experience

10

Speciality

Heart and Teeth

Add Doctor

Update Doctor

• [Delete Doctor Click Here!](#)

© 2021 - My ASP.NET Application

Add Doctor page - My ASP.NET / X +

localhost:44372/AddDoctor

Apps foobar Top AI & Machine L... Meet - cyf-myye-vm Novorésumé eCommerce search...

Hospital Management Home About Add Doctor All Doctors Logout

### Add Doctor

Doctor's Name

Kaushik

Email-ID

kaushik@gmail.com

Contact No

78963214

Address

Ahemdabad, Gujarat, India

Experience

10

Speciality

Heart and Teeth

Add Doctor Update Doctor

Doctor Added Successfully!

Delete Doctor [Click Here!](#)

© 2021 - My ASP.NET Application

All Doctors page - My ASP.NET / X +

localhost:44372/AllDoctors

Apps foobar Top AI & Machine L... Meet - cyf-myye-vm Novorésumé eCommerce search...

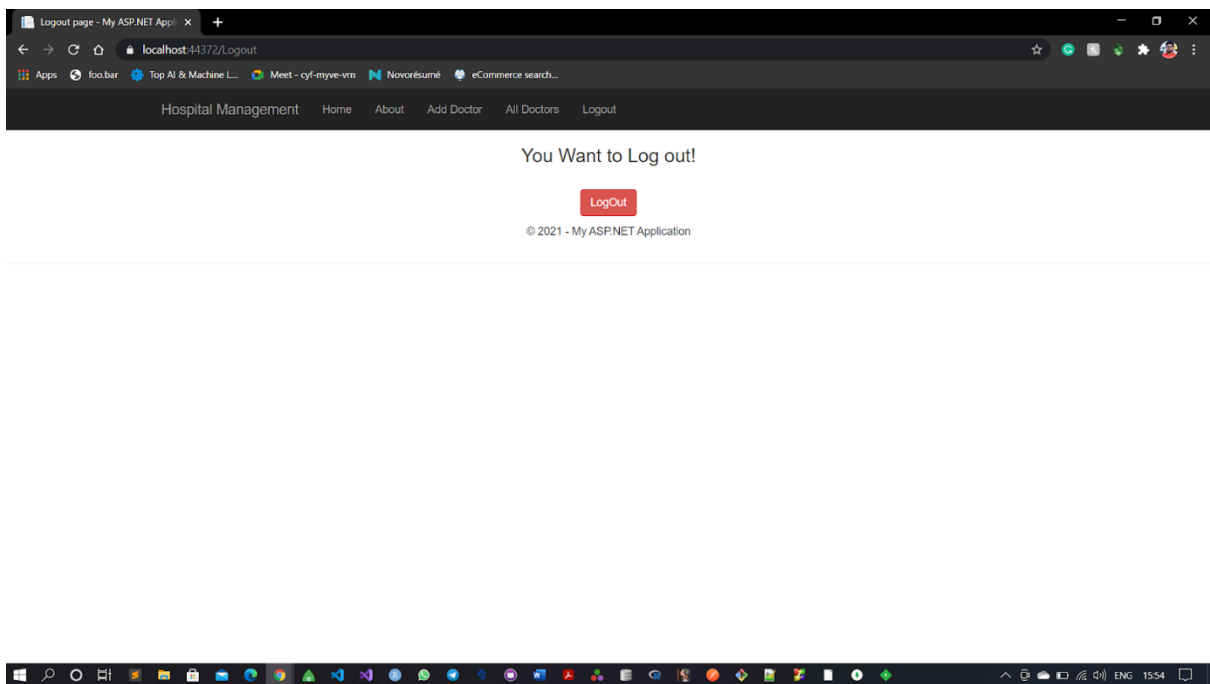
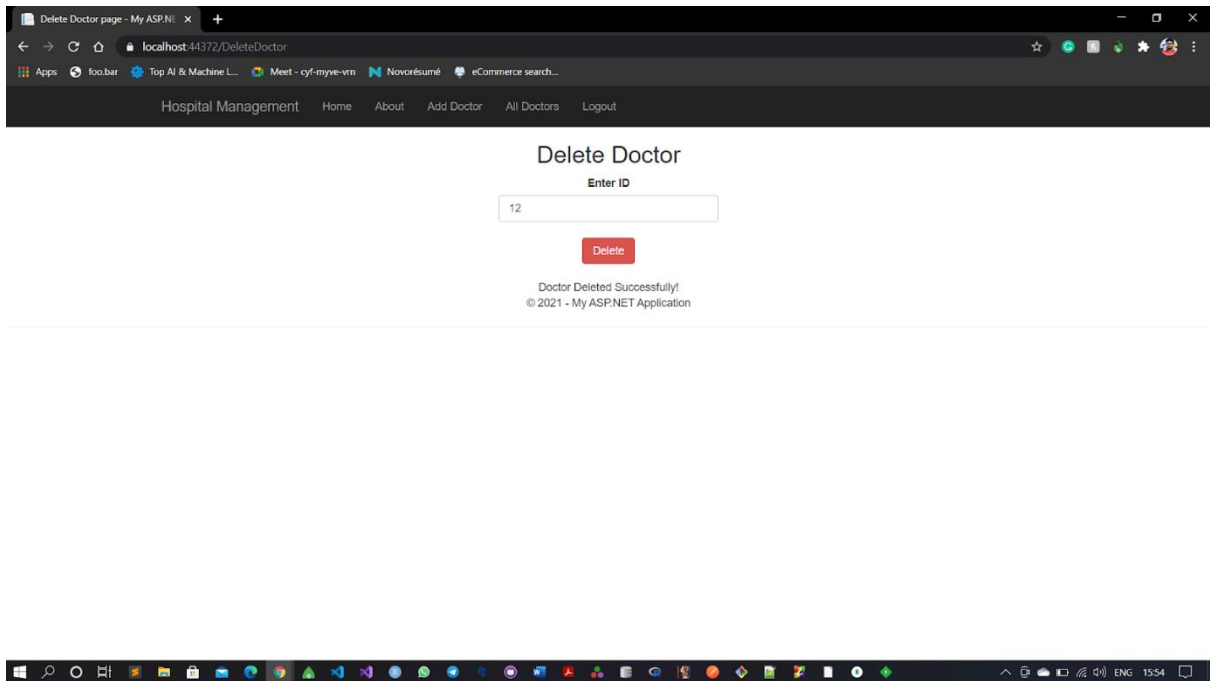
Hospital Management Home About Add Doctor All Doctors Logout

### All Doctors List

All Doctors

d_id	d_name	d_email	d_address	d_contact	d_experience	d_speciality
8	Elon R Musk	elon@gmail.com	Mars Earth	999999999	35	Cars
11	TheDz	dhruv@gmail.com	USA	333222111	9	Nose
12	Kaushik	kaushik@gmail.com	Ahemdabad, Gujarat, India	78963214	10	Heart and Teeth

© 2021 - My ASP.NET Application



Register page - My ASP.NET App X

localhost:44372/Register

Apps toolbar Top AI & Machine L... Meet - cyf-myye-vm Novcrésumé eCommerce search...

Hospital Management Home About P-Login A-Login Register Book-Appointment View Appointment

## Patient's Registration

**Username**

Mr.Chavda

**Email**

dhaval3337@gmail.com

**Contact No**

456123456

**Password**

Enter Password

**Re-Enter Password**

Re-Enter Password

Register

Password Doesn't Match!

© 2021 - My ASP.NET Application

Windows taskbar: File Explorer, Edge, VS Code, etc. System tray: 15:55, ENG

BookAppointment page - My ASP.NET App X

localhost:44372/BookAppointment

Apps toolbar Top AI & Machine L... Meet - cyf-myye-vm Novcrésumé eCommerce search...

Hospital Management Home About All Doctors Book-Appointment View Appointment dhaval3337@gmail.com Logout

## Book Appointment

**Doctor Email**

Enter Doctor's Email

**Patient Email**

Enter Patient's Email

**Current Health**

Enter current health of yours!

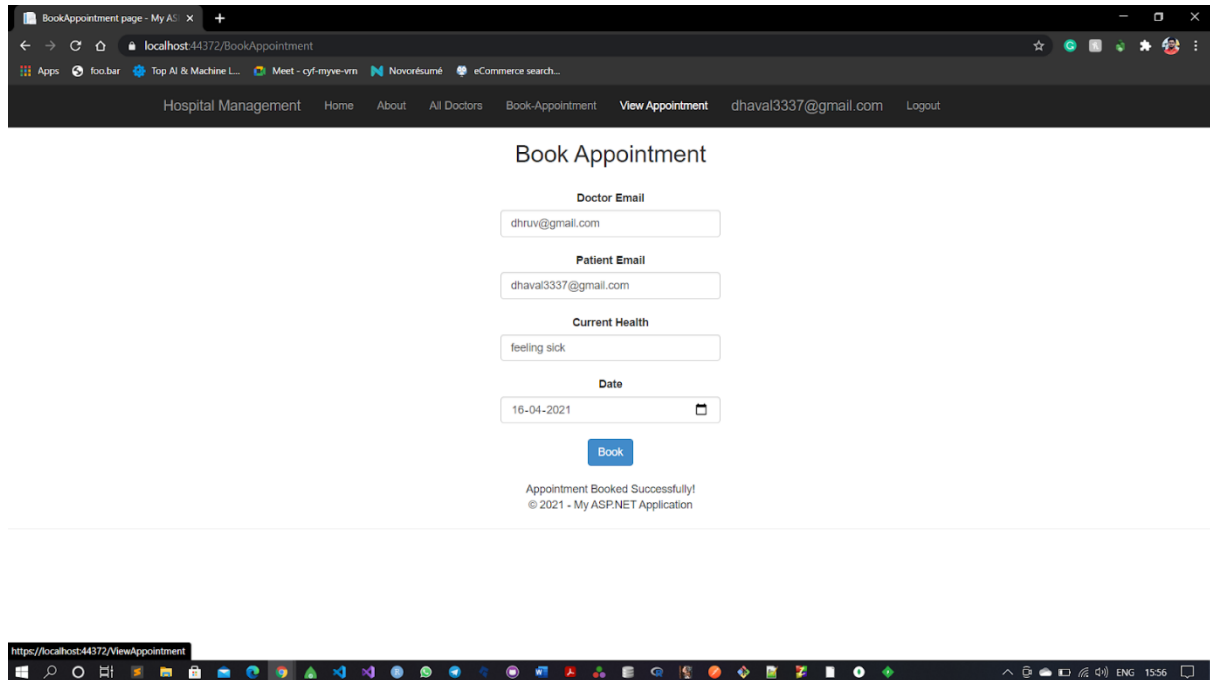
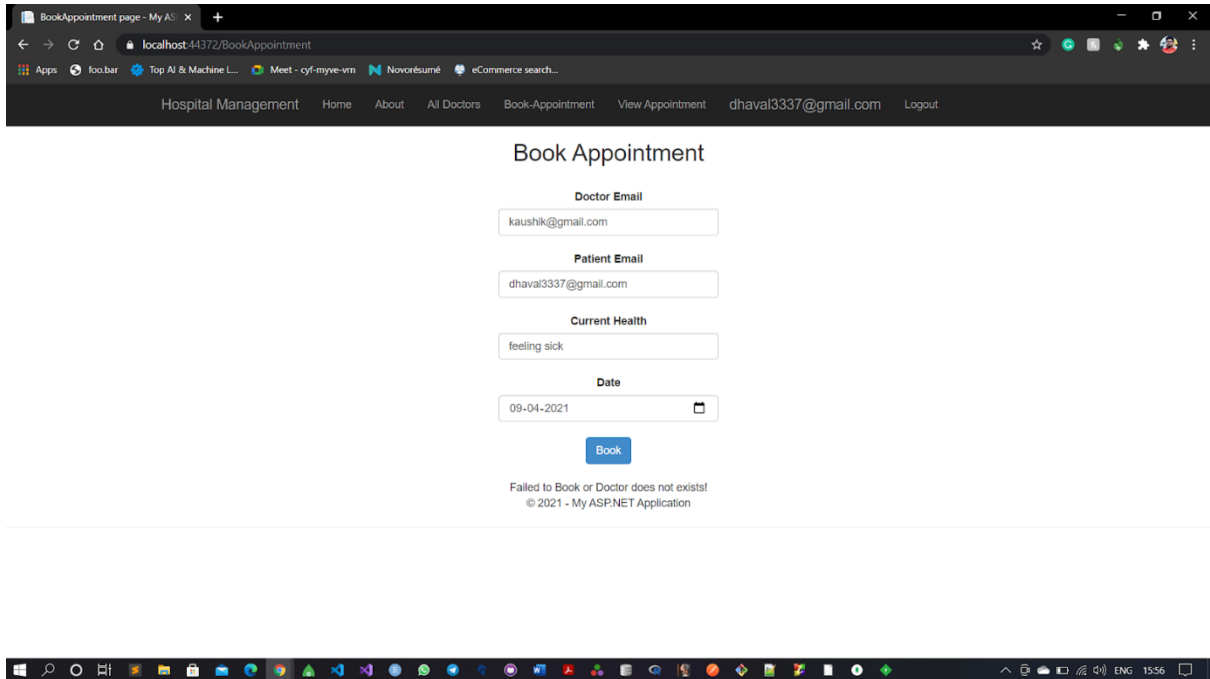
**Date**

dd-mm-yyyy

Book

© 2021 - My ASP.NET Application

Windows taskbar: File Explorer, Edge, VS Code, etc. System tray: 15:55, ENG



## All Appointment

All Appointment

a_id	ap_email	ad_email	a_date	a_desc
7	dhaval3337@gmail.com	dhruv@gmail.com	2021-04-16	feeling sick

© 2021 - My ASP.NET Application

## Patient's Details

Username

DC

Email

dhaval3337@gmail.com

Contact No

7678767878

Password

Enter Password

Re-Enter Password

Re-Enter Password

Update

Profile Updated Successfully!  
© 2021 - My ASP.NET Application



## **7. Conclusion**

This project implemented in the system was done after understanding all the system modules according to the requirements.

Functionality which are successfully implemented in this project are:

- User Authentication
- User Login
- User Registration
- Validations
- User update
- Logout
- Patient Update
- Patient Delete
- Book Appointment
- Doctor Add, Delete and Update

## **8. Limitation and Future Enhancement**

- Limitations:
  - Once User logout then if he/she tries to See appointment details then he/she can't see details.
  - All the photos and videos are available on some local storage or drive.
- Functionality Not Implemented:
  - Payment method
  - Customer care service
  - Admin Services
  - Rating
  - Feedback about review of Item
  - Doctor Services
- Future Extension:
  - Payment method can be added
  - Doctor services
  - Admin services
  - Doctor rating and review

## **9. Bibliography**

<https://www.c-sharpcorner.com/>

for error solving in backend in c-sharp language.

<https://stackoverflow.com/>

for all error solving

<https://www.tutorialspoint.com/index.htm>

for connection and calling APIs.