Hospital Management System

Prepared by

Harish Babu Alle (2091129)

Sravan Kumar Adula(2091226)

Anurag Rao Beerelli (2091164)

Under the supervision of

Prof. Yash Sheth

A Report Submitted to

Cegep de la Gaspesie et des Iles

for Partial Fulfilment of the Requirements for the

Diploma in Mobile Application Development

in Software Development Tools

MAD321 Software Development Tools Group Project-IV

Submitted at

Cégep de la Gaspésie et des Îles

CAMPUS DE MONTRÉAL

MOBILE APPLICATION DEVELOPMENT

At: 577 Boul Henri-Bourassa E, Montréal, QC H2C 1E2

June 2020

ABSTRACT

Life is becoming too busy to get medical appointments in person and to maintain a proper health care. The main idea of this work is to provide ease and comfort to patients while taking appointment from doctors and it also resolves the problems that the patients will face while making an appointment. The web application Hospital Management System acts as a client whereas the database containing the doctor's details, patient's details and appointment details is maintained by admin and website that acts as a server.

ACKNOWLEDGEMENT

No project or an initiative step to solve a real life problem, can stand without and stand without a support and guidance, and same is the case with this one. It is not single handedly completed project but it is a cooperative and correlative team work of hands together.

It is an honour to acknowledge the sincere help and guidance from all our teachers. We would like to thank the department of Mobile Application Development for giving us an opportunity for doing a project of our choice.

We are equally grateful to the entire faculty team of Department of Mobile Application Development that poured confidence in project that hasn’t not only enhanced our professional and technical skills but also enabled us to become more competitive in this field.

We are grateful to our Prof. Yash Sheth and other faculties of our department for their support and guidance.

Table of Contents

List of Figures

LIST OF ABBREVATIONS

PHP	**Hypertext Preprocessor.**
SQL	**Structural Query Language**
GUI	**Graphical User Interface**
IDE	**Integrated Development Environment**
HTML	**Hypertext Markup Language**
ANSI	**American National Standards Institute**
UML	**Unified Modelling Language**
XAMPP	**Cross-platform Apache server MySQL PHP Perl**
SHA	**Secure Hash Algorithm**

1 INTRODUCTION

If anyone is ill and wants to visit a doctor for check-up, he or she needs to visit the hospital land waits until the doctor is available. The patient also waits in a queue while getting appointment.

There is much book keeping work in this regard. An intelligent agent based appointment system has been proposed in which a scheduling system is provided for patients and searching doctors so they can get proper treatment on time. Mainly the input contains User's information, Knowledge base (symptoms and output of reasoning service. Our Web application provides various functionalities including personnel information, to trace position of actual user in real-time. Another study consists of an online database for the monitoring of patient. The proposed work in this paper is a Hospital Management System application that uses a PHP platform that makes the task of making an appointment from the doctor easy and reliable for the users. Web Application based "Hospital Management System" contains three modules. One module is the application designed for the patient that contains a login screen. The patient must register himself before logging in to the application. After logging in, the patient can select a Specialty and preferred doctor. The patient has the option of selecting a doctor from the list of doctors and can view the doctor's details.

1.1 Patient module:

Patient can read specialty information of doctors. Patient can view history of old appointments. Patient can Signup/Login. They can book an appointment at the flexible time and date. They can cancel an appointment through their user page.

1.2 Admin module:

The second module is the admin module that is designed on the website. The admin views all details of doctors and all appointments by the patient. The admin can add doctor, view patient's details and doctor's details. All the doctors of the specific Specialty are registered by the admin. Doctors cannot register themselves.

1.3 Doctors module:

Doctors can have their own profile page. Doctors can confirm or decline an appointment on their convenience. Doctors can register by consulting admin.

1.4 Objectives:

The objective of building this Hospital Management System is:

• To develop a system that allows Admin and Doctors to

manage business.

• To help Hospital Management to effectively manage their data.

• To reduce unnecessary paperwork for hospital in maintaining patients

information.

2 SYSTEM ANALYSIS

2.1 Introduction

The requirements are very important for the implementation of the project. One should clearly specify the requirements clearly in order to get the desired output.

System Requirements

2.1.1 Functional Requirements

- Admin: Admin adds doctor, view patients details and doctors' details.
- Doctor: Doctor can confirm or decline an appointment on their convenience.
- Patient: Book an appointment or cancel an appointment.

2.1.2 Non-functional requirements:

- **Security:** Only authorized user can access the system with username and password.
- **Performance:** Easy tracking of record and updating can be done.
- **User friendly:** This system is very interactive.

2.2 Overall description

The hospital management system allows authorized members to access the records of registered patients or doctors. It can be used in various hospitals across the globe and simplifies working of hospitals.

- ➢ Product perspective

The proposed system shall be developed using XAMPP server architecture and be compatible with Microsoft windows operating system.

- **User interfaces:**
  - Login: to allow the entry of only authorized users through valid login id and password.
  - Admin: Maintains Doctors details and patient's details.
  - Doctor: Generate the invoice and upload the file.
  - Patient: Login/Signup and book an appointment.
- **Software interfaces:**
  - MS-windows operating system.
  - MS SQL server 2000.
  - Platform: PHP, HTML, Java Script.

- **Product functions**

The HMS will allow access only to authorized users with specific roles (system administrator, doctor, and patient).

Depending upon the user's role, he/she will be able to access only specific modules of the system a summary of major functions that the user will perform.

- Login facility for enabling only authorized access to the system.
- Admin will be able to add doctor, view patient's details and doctor's details.
- Patients can view history of old appointments, signup/login and book or cancel an appointment.

- Doctors can maintain their own profile page, confirm, or decline an appointment.

Below are the Database, Web Application and Server details.

Database: MySQL

Web Application: PHP, JAVASCRIPT

Server: XAMPP

3 DESIGN

3.1 UML Diagrams:

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. UML provides variety of documents in addition raw executable codes.

3.1.1 Use Case Diagram:

A Use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which the specific roles are played by the actors within and around the system.

The relationships between and among the actors and the use cases.

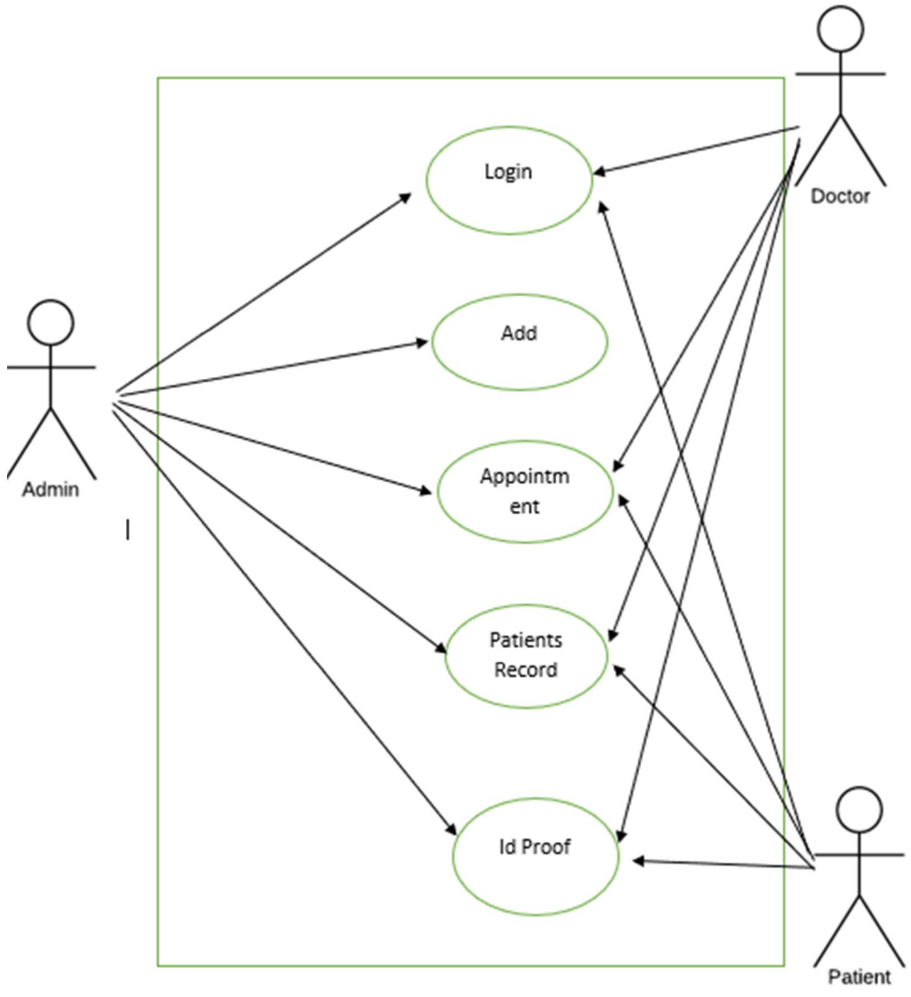


Figure 1 Use Case

3.1.2 Class Diagram:

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods ,and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and cantered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned, and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned, and the first letter is lowercase.

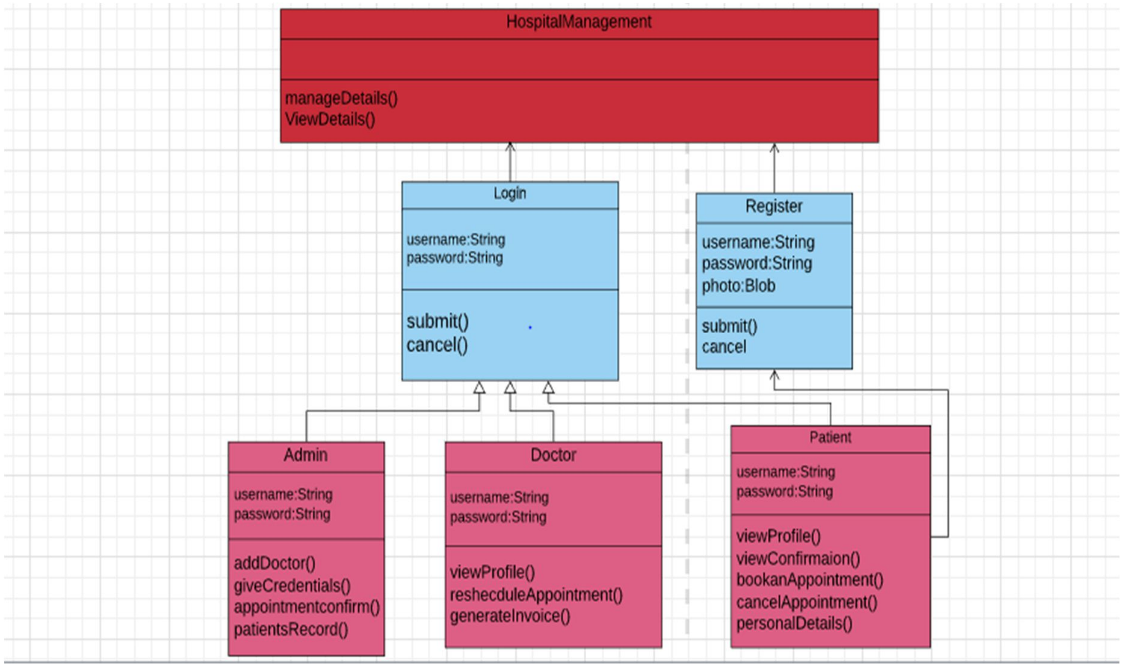


Figure 2 Class Diagram

Admin: This module has the entire access to all other modules. Admin used to add the doctor’s details.

Patient: Login/Signup and book an appointment.

Doctor: Doctor can confirm or decline an appointment on their convenience.

3.1.3 Sequence Diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (*lifelines* , different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

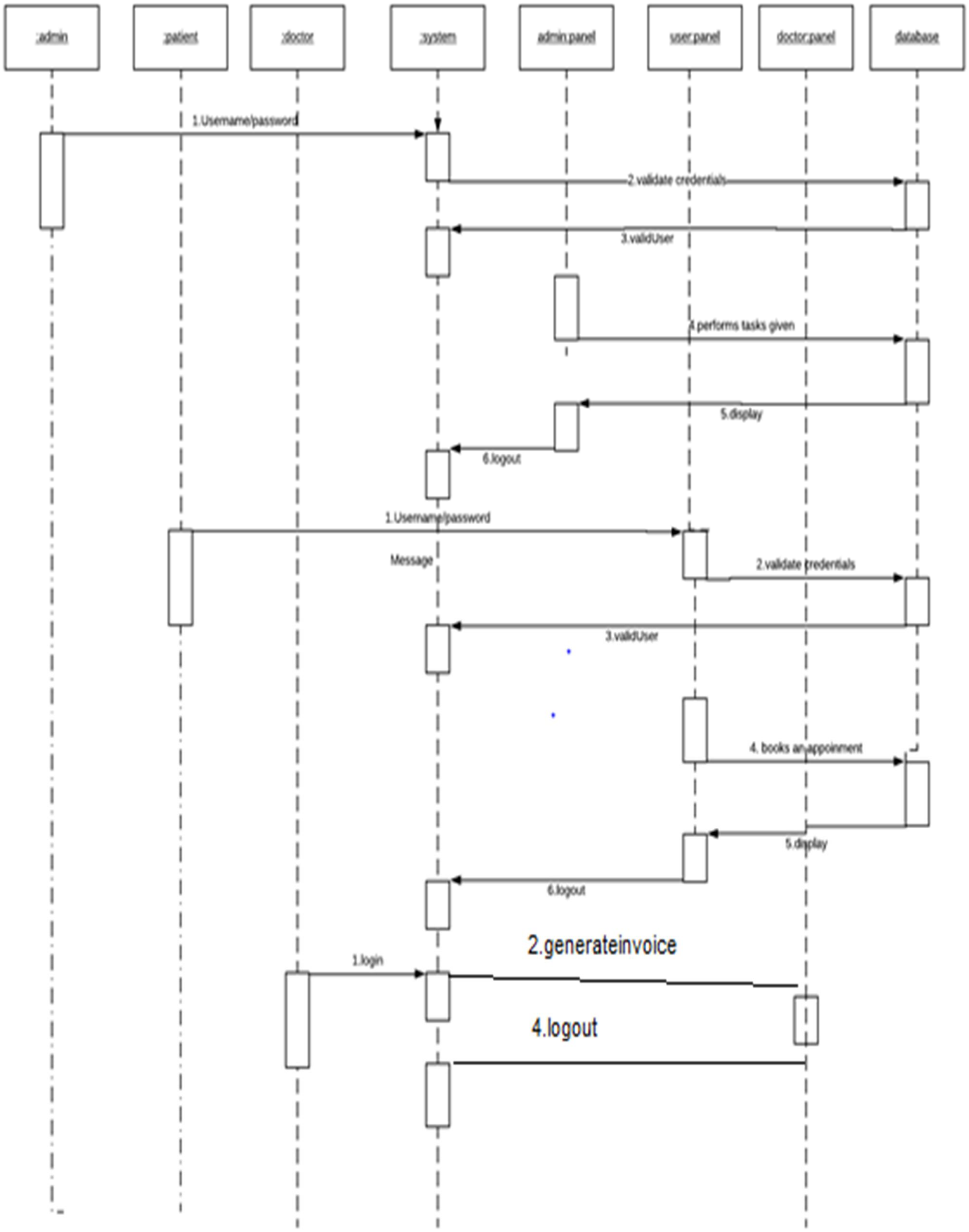


Figure 3Sequence Diagram

3.1.4 State chart Diagram:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

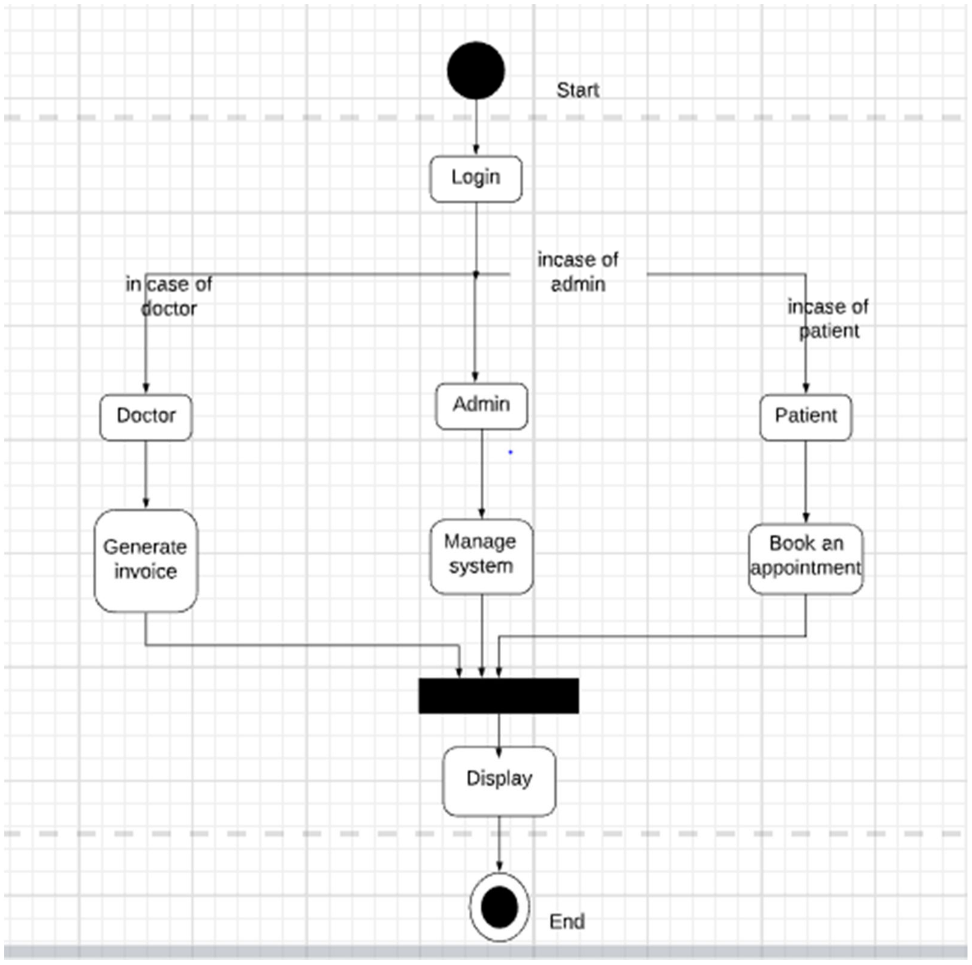


Figure 4 State Chart Diagram

3.1.5 Collaboration Diagram:

A collaboration diagram resembles a flow chart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

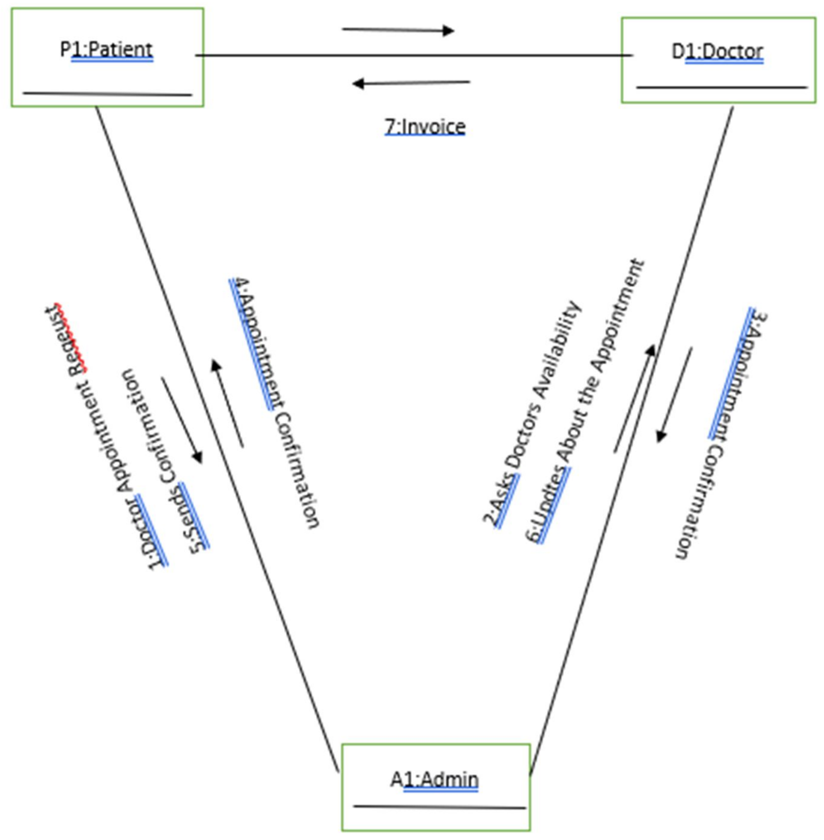

P1:Patient
D1:Doctor
7:Invoice
1:Doctor Appointment Request
5:Sends Confirmation
4:Appointment Confirmation
2:Asks Doctors Availability
6:Updtes About the Appointment
3:Appointment Confirmation
A1:Admin

Figure 5 Collaboration Diagram

3.2 Module Design and Organization Modules

Admin: This module has the entire access to all other modules. Admin used to add the doctor’s details.

Patient: Login/Signup and book an appointment.

Doctor: Doctor can confirm or decline an appointment.

4 IMPLEMENTATION

4.1 HTML

HTML stands for Hypertext Mark-up Language, is the predominant mark-up language for web pages. It provides a means to describe the structure of text-based information in a document by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags , surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behaviour of web browsers and other HTML processors. HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML descended form (such as XHTML 1.0 and later or its form descended directly from SGML.

Hyper Text Mark-up Language

Hypertext Mark-up Language (HTML) , the languages of the Worldwide Web (WWW) , allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Mark-up Language , but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A mark-up language is simply a series of elements, each delimited with special characters that define how text or other items enclosed with the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document. HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. HTML provides tags (special codes to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, colour, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

4.1.1 Basic HTML Tags:

<! -- --> specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels (1– 6)

<HEAD>...</HEAD> Contains tags that specify information about a Document.

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags.

<META>...</META> Provides meta19 information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

4.1.2 Advantages

1. A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
2. HTML is platform independent.
3. HTML tags are not case-sensitive.

4.2 JavaScript:

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser's display accordingly Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client-side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags.

<SCRIPTS>..</SCRIPT>.

<SCRIPT LANGUAGE = "JavaScript">

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript:

1. Validate the contents of a form and make calculations.

2. Add scrolling or changing messages to the Browser's status line.

3. Animate images or rotate images that change when we move the mouse over them.

4. Detect the browser in use and display different content for different browsers.

5. Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

4.3 Database:

A database management system (DBMS is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerge as a fairly standard part of any company back office.

4.3.1 Description

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database.

A DBMS includes:

1. A modelling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.

The four most common types of organizations are the hierarchical,network, relational and object models. Inverted lists and other methods are also used. A given database management system may provide one or more of the four models.

- The optimal structure depends on the natural organization of the application's data, and on the application's requirements which include transaction rate (speed , reliability, maintainability, scalability, and cost) .
- The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.

2. Data structures (fields, records, files and objects optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory.

3. A database query language and report writer to allow users to interactively interrogate the database, analyse its data and update it according to the users privileges on data. It also controls the security of the database.

Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called subschemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data. If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization.

These controls are only available when a set of application programs are customized for each data entry and updating function. A transaction mechanism that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control , and faults (fault tolerance) . It also maintains the integrity of the data in the database. The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints for example, no two customers with the same customer numbers (key fields can be entered into the database. See ACID properties for more information (Redundancy avoidance . The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data.

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system. Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators. Database servers are specially designed computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with RAID disk arrays used for stable storage. Connected to one or more servers via a highspeed channel, hardware database accelerators are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. Sometimes DBMSs are built around a private multitasking kernel with built-in networking support although nowadays.

4.4 SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model. In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

Data definition: Defining tables and structures in the database (DDL) used to create, alter and drop schema objects such as tables and indexes .

Data manipulation: Used to manipulate the data within those schema objects (DML) Inserting, Updating, Deleting the data, and Querying the database.

A schema is a collection of database objects that can include: tables, views, indexes and sequences.

List of SQL statements that can be issued against an Oracle database schema are:

- ❖ **ALTER** - Change an existing table, view or index definition (DDL)
- ❖ **AUDIT** - Track the changes made to a table (DDL)
- ❖ **COMMENT** - Add a comment to a table or column in a table (DDL)
- ❖ **COMMIT** - Make all recent changes permanent (DML – transactional)
- ❖ **CREATE** - Create new database objects such as tables or views (DDL)
- ❖ **DELETE**- Delete rows from a database table (DML)
- ❖ **DROP** - Drop a database object such as a table, view or index (DDL)
- ❖ **GRANT** - Allow another user to access database objects such as tables or views (DDL)
- ❖ **INSERT** - Insert new data into a database table (DML)
- ❖ **No AUDIT** - Turn off the auditing function (DDL).
- ❖ **REVOKE** - Disallow a user access to database objects such as tables and views (DDL).
- ❖ **ROLLBACK** - Undo any recent changes to the database (DML – Transactional).
- ❖ **SELECT** - Retrieve data from a database table (DML).
- ❖ **TRUNCATE** - Delete all rows from a database table cannot be rolled back (DML).
- ❖ **UPDATE**- Change the values of some data items in a database table (DML).

5 Output Screens

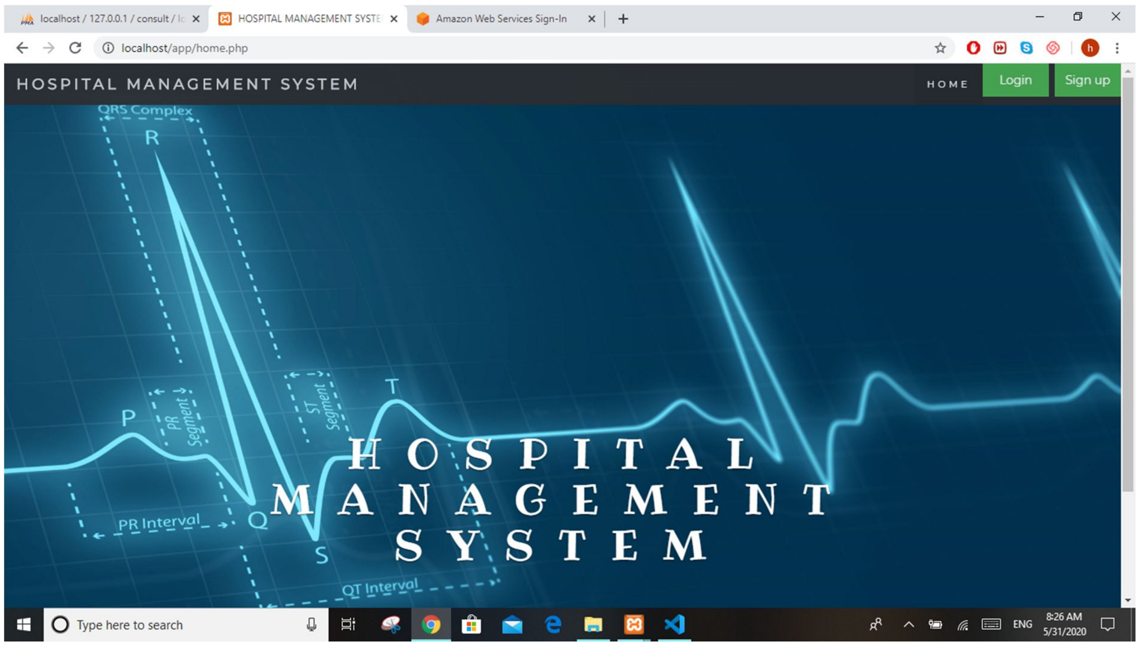

localhost / 127.0.0.1 / consult / lo
HOSPITAL MANAGEMENT SYSTE
Amazon Web Services Sign-In
localhost/app/home.php
HOSPITAL MANAGEMENT SYSTEM
HOME
Login
Sign up
HOSPITAL MANAGEMENT SYSTEM
Type here to search
8:26 AM
5/31/2020

Figure 6 5.1 Home Page

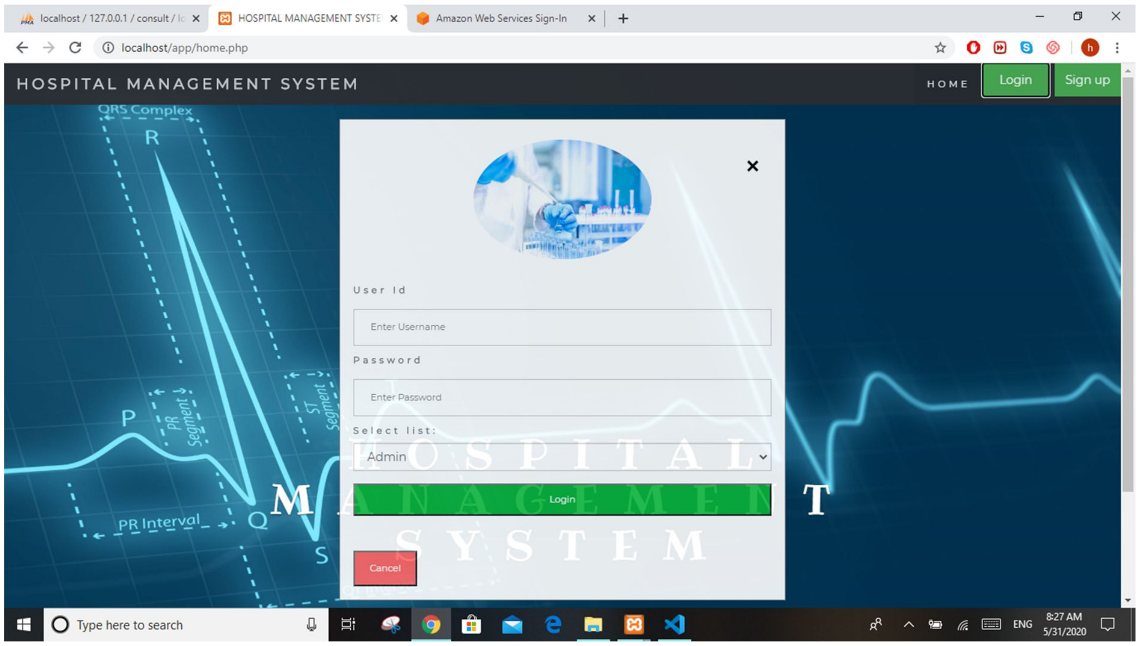


Figure 7 5.1.2 Login Page

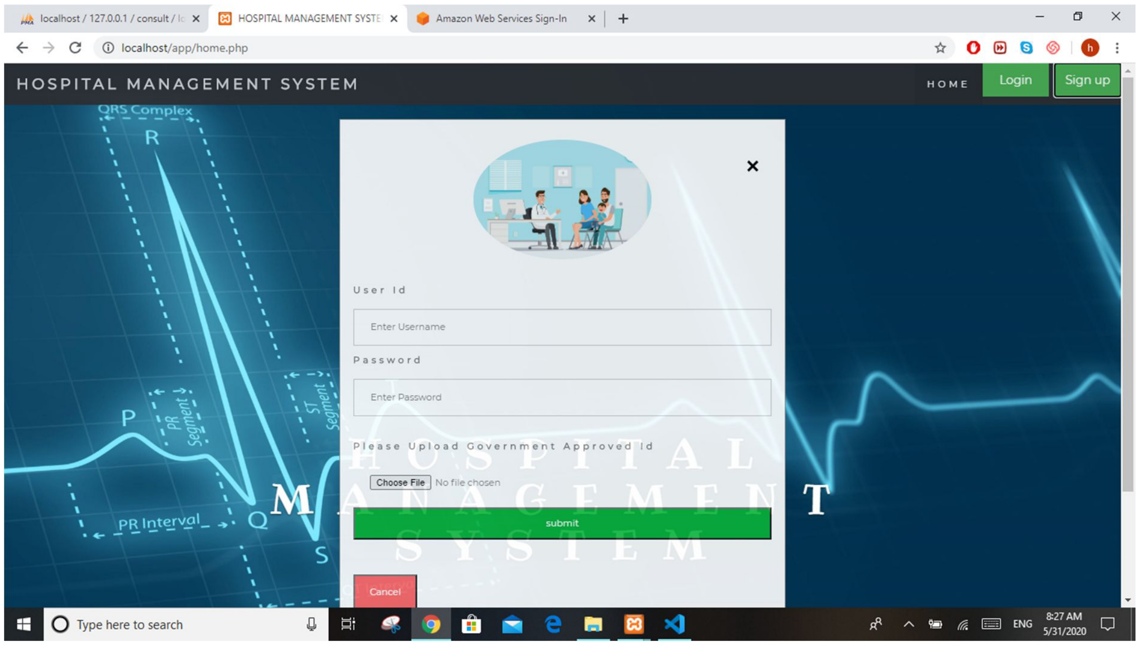
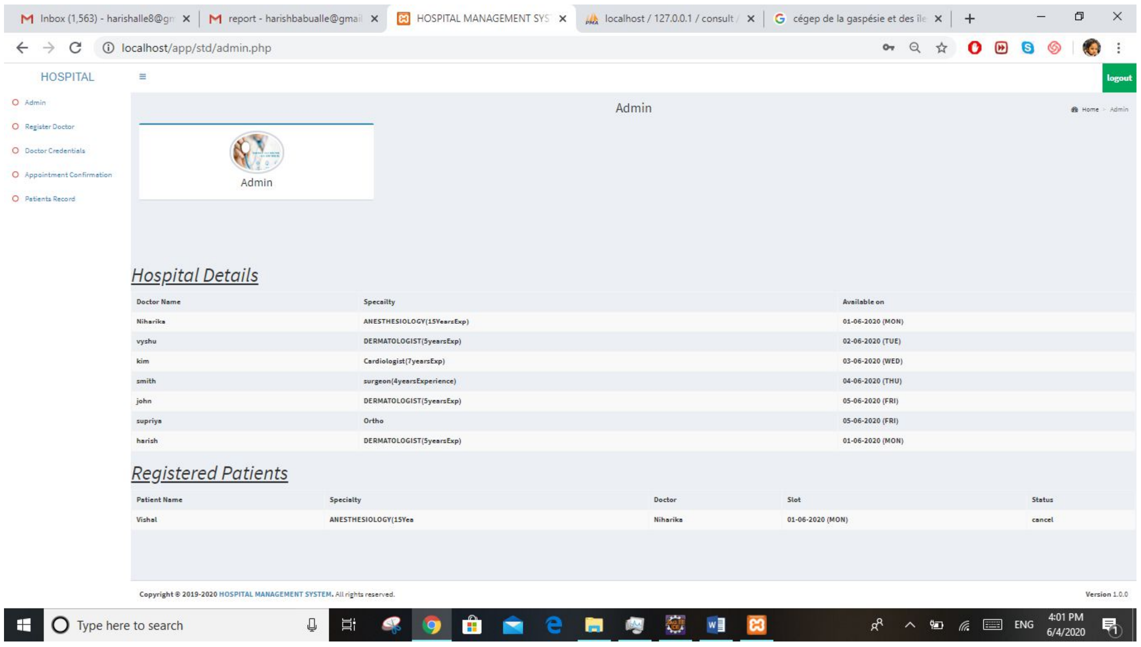


Figure 8 5.1.3 Sign up Page



Hospital Details

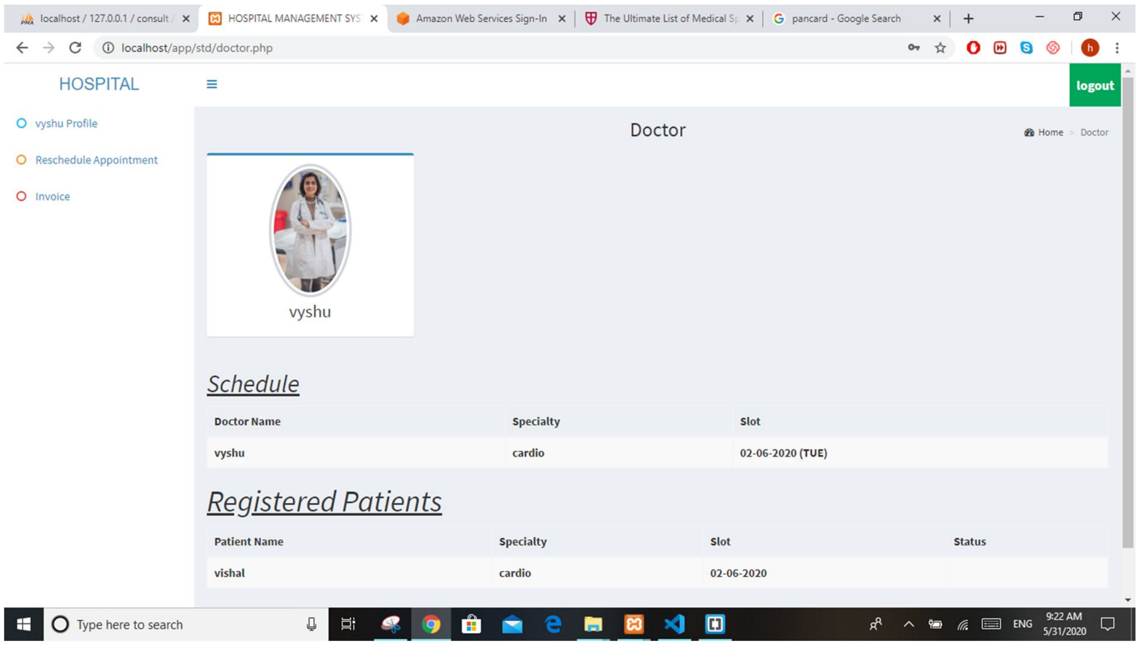
Doctor Name	Specailty	Available on
Niharika	ANESTHESIOLOGY(15YearsExp)	01-06-2020 (MON)
vyshu	DERMATOLOGIST(5yearsExp)	02-06-2020 (TUE)
kim	Cardiologist(7yearsExp)	03-06-2020 (WED)
smith	surgeon(4yearsExperience)	04-06-2020 (THU)
john	DERMATOLOGIST(5yearsExp)	05-06-2020 (FRI)
supriya	Ortho	05-06-2020 (FRI)
harish	DERMATOLOGIST(5yearsExp)	01-06-2020 (MON)

Registered Patients

Patient Name	Specialty	Doctor	Slot	Status
Vishal	ANESTHESIOLOGY(15Yea	Niharika	01-06-2020 (MON)	cancel

Copyright © 2019-2020 HOSPITAL MANAGEMENT SYSTEM. All rights reserved. Version 1.0.0

Figure 9 5.1.4 Admin Module



Doctor

vyshu

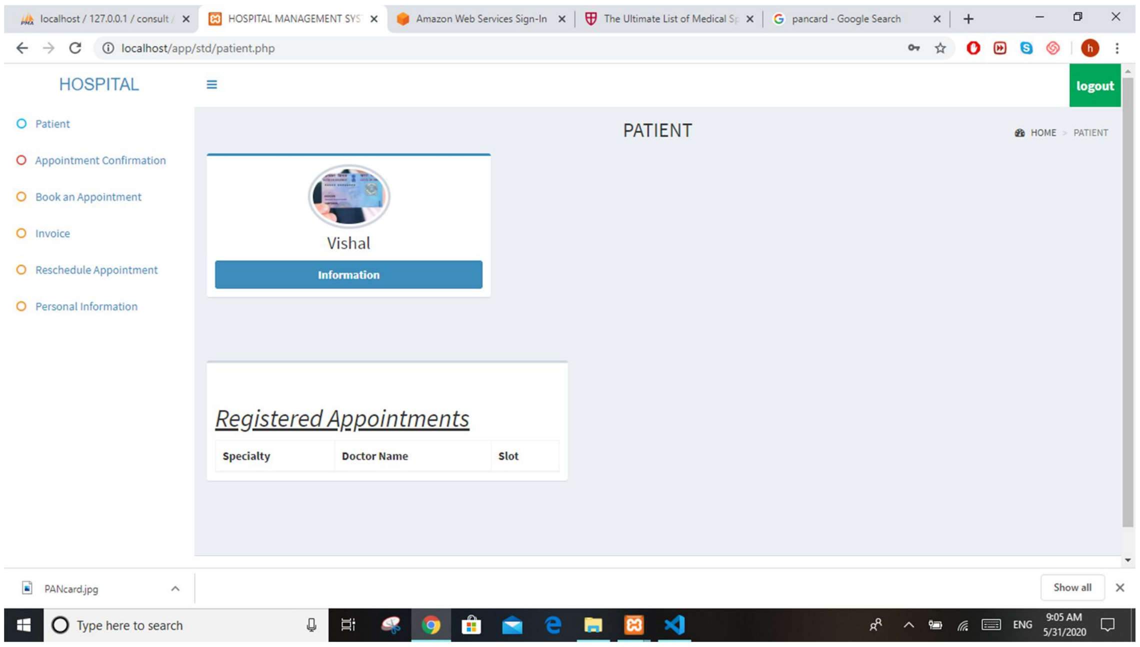
Schedule

Doctor Name	Specialty	Slot
vyshu	cardio	02-06-2020 (TUE)

Registered Patients

Patient Name	Specialty	Slot	Status
vishal	cardio	02-06-2020	

Figure 10 5.1.5 Doctor Module



PATIENT

Vishal

Registered Appointments

Specialty	Doctor Name	Slot

Figure 11 5.1.6 Patient module

6 Task Division

6.1 Work Distribution

- Harish Babu Alle (2091129): Worked with PHP, JAVASCRIPT and Database.
- Sravan kumar Adula(2091226): Worked with HTML,CSS,Bootstrap.
- Anurag Rao (2091164): Worked with HTML.CSS, Bootstrap

7 CONCLUSION

The aim of this project is to design and develop a system that could give easy management to the Hospital Management System. This chapter is about the early view of what will be from the developer's view. Project background describe the introduction for the project and includes the problem has occurs from previous system, the aim of this system, the target user, module, the benefit of this system and the expected result from this project.

The problem statement describes the problem related to why should this system be developed. By replacing and improving the management system, less manual work will be required in the process of collecting, handling and maintaining of the data. The scope of the projects explains the module and target user for this system. While the project significance delivers what the system can provide based on developer's side. The expected output explains the system ability from user view.

8 FUTURE SCOPE

A payment or some amount may be charged to the users/patients while making an appointment to avoid the unethical users. As many users only register themselves just for fun and has no concern by making an appointment.

Some more future directions are the improvements in the patient's module which includes setting reminders for the appointments and saving the appointment date to the calendar and patient can register for one or more doctors.

9 Bibliography

References

- w3schools.com
- tutoiralspoint.com
- stackoverflow.com