LIFECARE

A Project Report

submitted in partial fulfillment for the requirements of the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

PAHUL KALRA (1702913069)

RAKSHIT SHARMA (1702913086)

RAHUL CHOUDHARY (17020913082)

RAHUL MISHRA (1702913083)

Supervised by

PROF. AMAR SINGH

DEPARTMENT OF INFORMATION TECHNOLOGY

KIET GROUP OF INSTITUTIONS, GHAZIABAD, UTTAR PARDESH

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)

Session 2020-21

DECLARATION

We declare that

a. the work contained in this report is original and has been done by us under the guidance of our supervisor.
b. the work has not been submitted to any other institute for any degree or diploma.
c. We have followed the guidelines provided by the institute to prepare the report.
d. We have conformed to the norms and guidelines given in the ethical code of conduct of the institute.
e. wherever We have used materials (data, theoretical analysis, figures and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references.

Signature of the student:
Name: Pahul Kalra
Roll number: 1702913069

Signature of the student:
Name: Rakshit Sharma
Roll number: 1702913086

Signature of the student:
Name: Rahul Choudhary
Roll number: 1702913082

Signature of the student:
Name: Rahul Mishra
Roll number: 1702913083

Place: KIET Group of Institutions, Ghaziabad
Date:

CERTIFICATE

This is to certify that the project Report entitled, **"LIFECARE"** submitted by **Pahul Kalra, Rakshit Sharma, Rahul Choudhary** and **Rahul Mishra** in the Department of Information Technology of KIET Group of Institutions, Ghaziabad, affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India, is a record of bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Information Technology of the Institute.

Signature of Supervisor:
Supervisor Name: PROF. AMAR SINGH
Date: 10thJuly, 2021

List of Figures

List of Tables

List of Acronyms

1. WWW -	World Wide web
2. HTML -	Hyper Text Markup Language
3. GUI -	Graphical User Interface
4. DFD -	Data Flow Diagram
5. ERD -	Entity Relationship Diagram
6. API -	Application Programming Interface
7. CSS -	Cascading Style Sheets
8. DOM -	Document Object Model
9. IDE -	Integrated Development Environment
10.JSON -	JavaScript Object Notation
11.REST -	Representational State Transfer

Abstract

1. What is Lifecare ?

This project underwent development solely for one purpose and that is to **Prevent Epidemic Rebound.** At Lifecare, we care about people and we always try our best to keep you update to latest health related news, you can contact your doctor sitting at home, track your surroundings for COVID numbers.

Services provided by us -

1.1 Medical And Health News -

In this service, we provide Medical and Health related news from WHO and other genuine sources so that you and the people that matters to you can be safe from any future pandemic and as a whole, benefit the whole society.

1.2 COVID Tracking –

Now that COVID has been spread already, we should take necessary protection you can take so that you can become immune to it.

1.3 Online Doctor Consultation –

At the comfort of your home, now you can consult any doctors near you or far from you so that you don’t have to go outside your home in these pandemic situations. We provide high quality video calls to help you with this.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

It is a Web based system where patient can consult and get appointment of doctor easily. The appointment module is an electronic paper less application designed with high flexibility and ease of usage, implemented in single clinics and polyclinics. The most important thing is to do all these in just few clicks on internet at home or through mobile. This system will take care of your health and doctor consultation and appointment process gets easier & convenient. This is a web based application that overcomes the issue of managing and booking appointments according to user’s choice or demands. The data are well protected for personal use and makes the data processing very fast. Towards this achievement the computerization of the Lifecare will help greatly in maintaining of proper information about the out patients who are eligible for the free services and the patients who are not eligible for the free services and patients records. Lifecare is designed for multi speciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

1.2 Problem Definition

Keeping the current scenario in mind, we are still feeling the wrath of COVID-19 and it has created a situation where people are scared just to go out and consult doctors even when their health are deteriorating.

People are even scared to go out and buy medicines. Some people are just too busy to go to a doctor and wait in long queues for an appointment and during this pandemic we now knew surely that people are aware of technology and use it. They are not a layman anymore and can adapt to change.
In our project people can book an appointment with a doctor nearby him/her or even at distance and can have a high-quality video chat with him/her. This takes care of any physical meeting or waiting in queues at a doctor's premises for hours, or just to travel to his/her premises.
Also to prevent this in the coming future, our platform also informs the user about the ongoing outbreaks in different areas and other diseases that may be spreading at a fast rate. Our platform tackle this by providing alerts and proper precautions that one may take so that he/she is immune from it.

1.3 Project Overview

Lifecare a place where patient can consult and get appointment of doctor easily. The appointment module is an electronic paper less application designed with high flexibility and ease of usage, implemented in single clinics and polyclinics. The most important thing is to do all these in just few clicks on internet at home or through mobile. This system will take care of your health and doctor consultation and appointment process gets easier & convenient. This is a web based application that overcomes the issue of managing and booking appointments according to user's choice or demands.

1.4 Objectives

Smart medical consult and booking system that provides patients or any user an easy way of booking a doctor's appointment online. This system will take care of your health and doctor consultation and appointment process gets easier & convenient.

Alert the user for ongoing diseases that are most active in the environment so user get aware of symptoms and preventions of that disease. This application suggests the list of the doctors that are available and user can consult on virtual meet as well as physical if needed.

We provide Medical and Health related news from WHO and other genuine sources so that you and the people that matters to you can be safe from any future pandemic and as a whole, benefit the whole society.

COVID TRACKING -Now that COVID has been spread already, we should take necessary protection you can take so that you can become immune to it.

1.5 Motivation

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

CHAPTER 2

FEASIBILITY STUDY

2.1 Exposed System

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

2.2 Proposed System

The Lifecare is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

2.3 Feasibility Study

Feasibility Study is a preliminary study undertaken to determine and document a project's viability. The term feasibility study is also used to refer to the resulting document. These results of this study are used to make a decision whether to proceed with the project, or table it. If it indeed leads to a project being approved, it will – before the real work of the proposed project starts – be used to ascertain the likelihood of the project’s success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative.

2.3.1 Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customised products have to be purchased.

2.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

2.3.3 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 3

SYSTEM ANALYSIS & DESIGN

3.1 Requirement Analysis

3.1.1 Introduction

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as(computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

3.1.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HP), especially in case of operating systems. An HP lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Table no -1 : Hardware Requirements

Processor	Intel Core i3
Ram	4 GB
Hard Disk	500 GB

3.1.3 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Table no -2 : Software Requirements

Operating System	Windows 7/8/10
Front End	Html, CSS, java script. Server SIDE
Script	Node Js
Database	Mongo Db

3.2 Design

3.2.1 Introduction to UML

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

· Visualizing

· Specifying

· Constructing

· Documenting

Visualizing - Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying - Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing - UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML

Documenting - The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

3.2.2 UML Diagrams

Use Case Diagram:

A use case diagram in the Unified Modelling Language(UML) is a type of behavioural diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases) and any dependencies between those use cases.

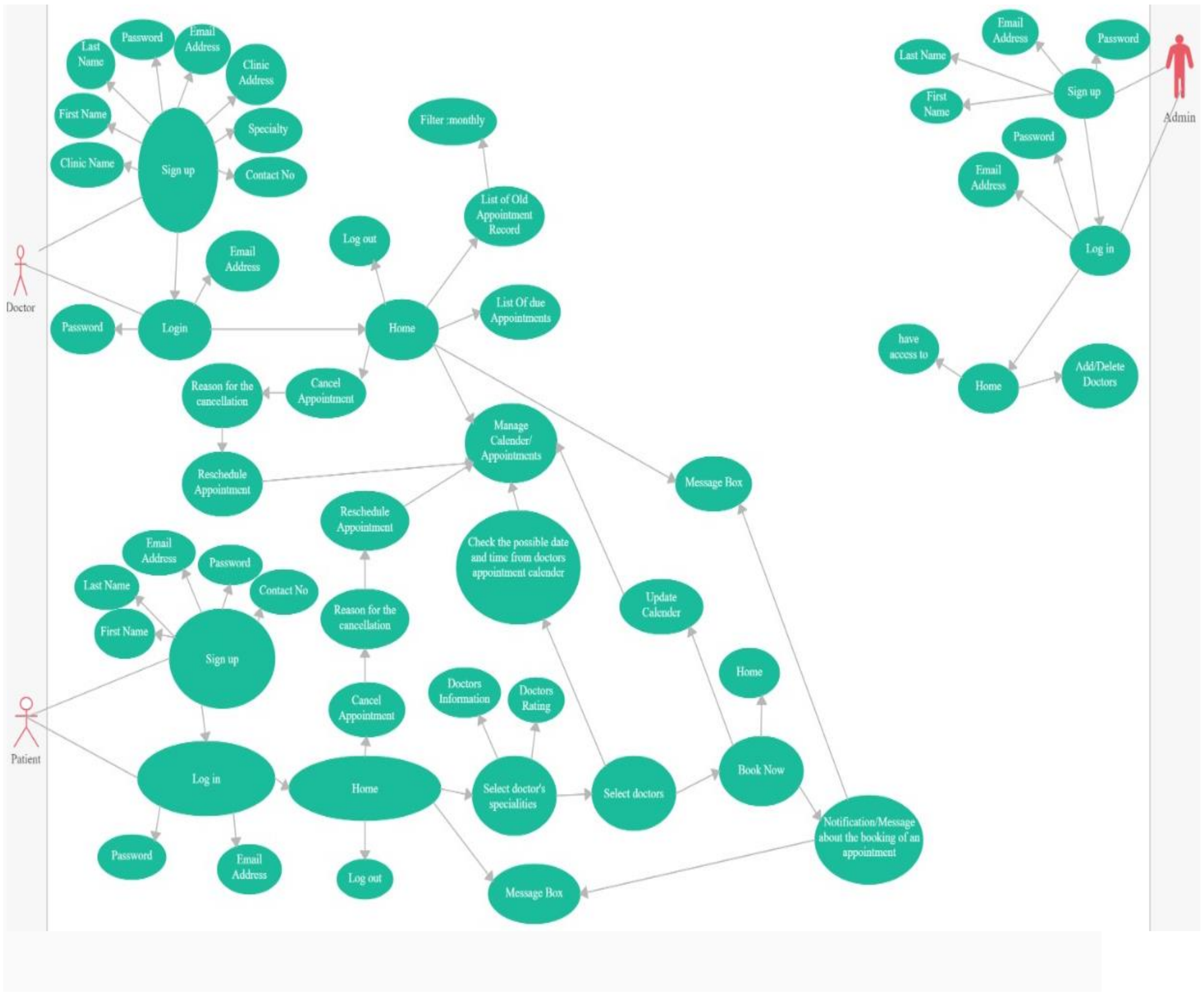


Fig. 1: Use case Diagram

Class Diagram:

A Class is a category or group of things that has similar attributes and common behaviour. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provides the representation that developers work from. Class diagrams help on the analysis side, too.

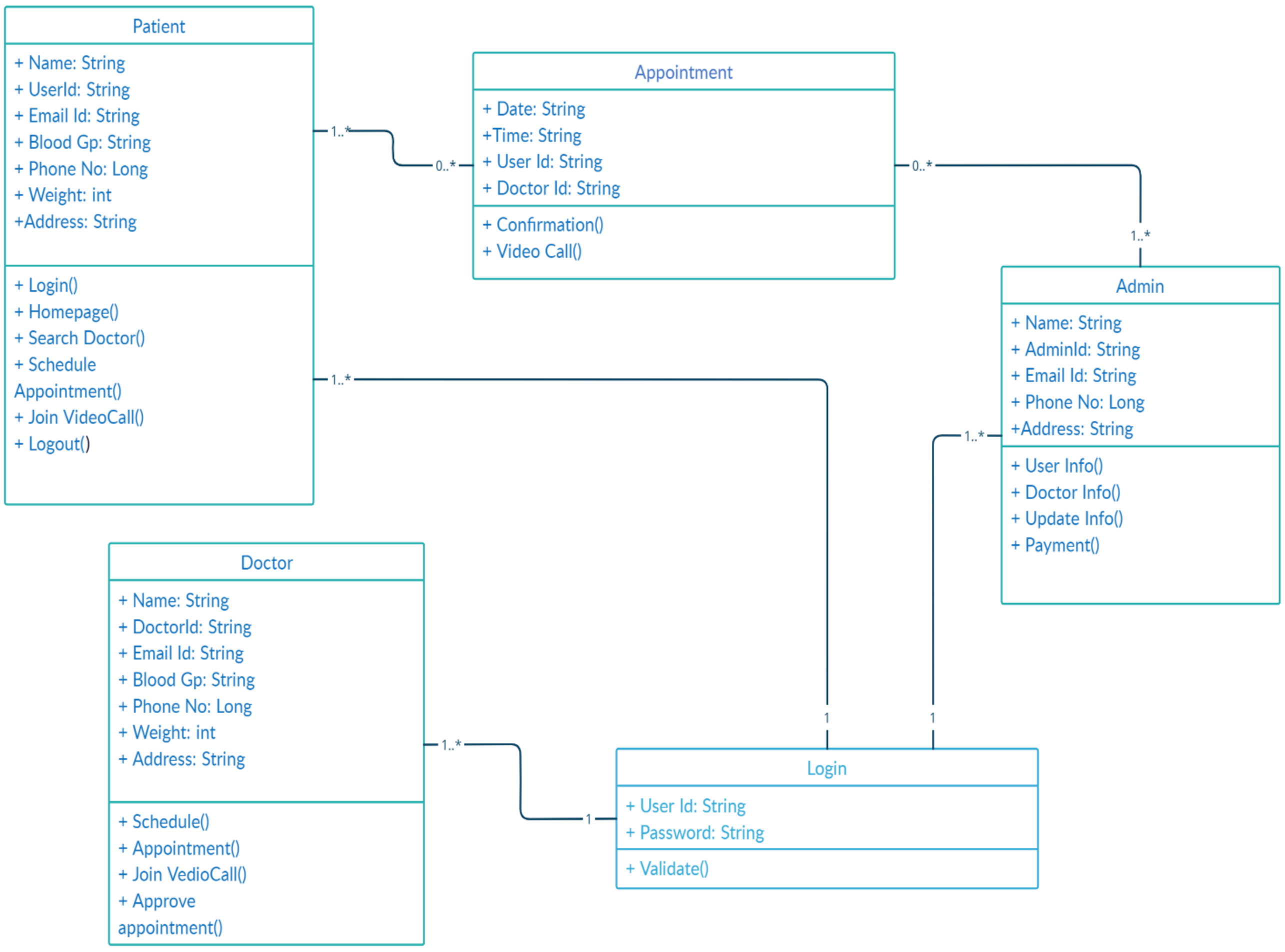


Fig. 2: Class Diagram

Activity Diagram:

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

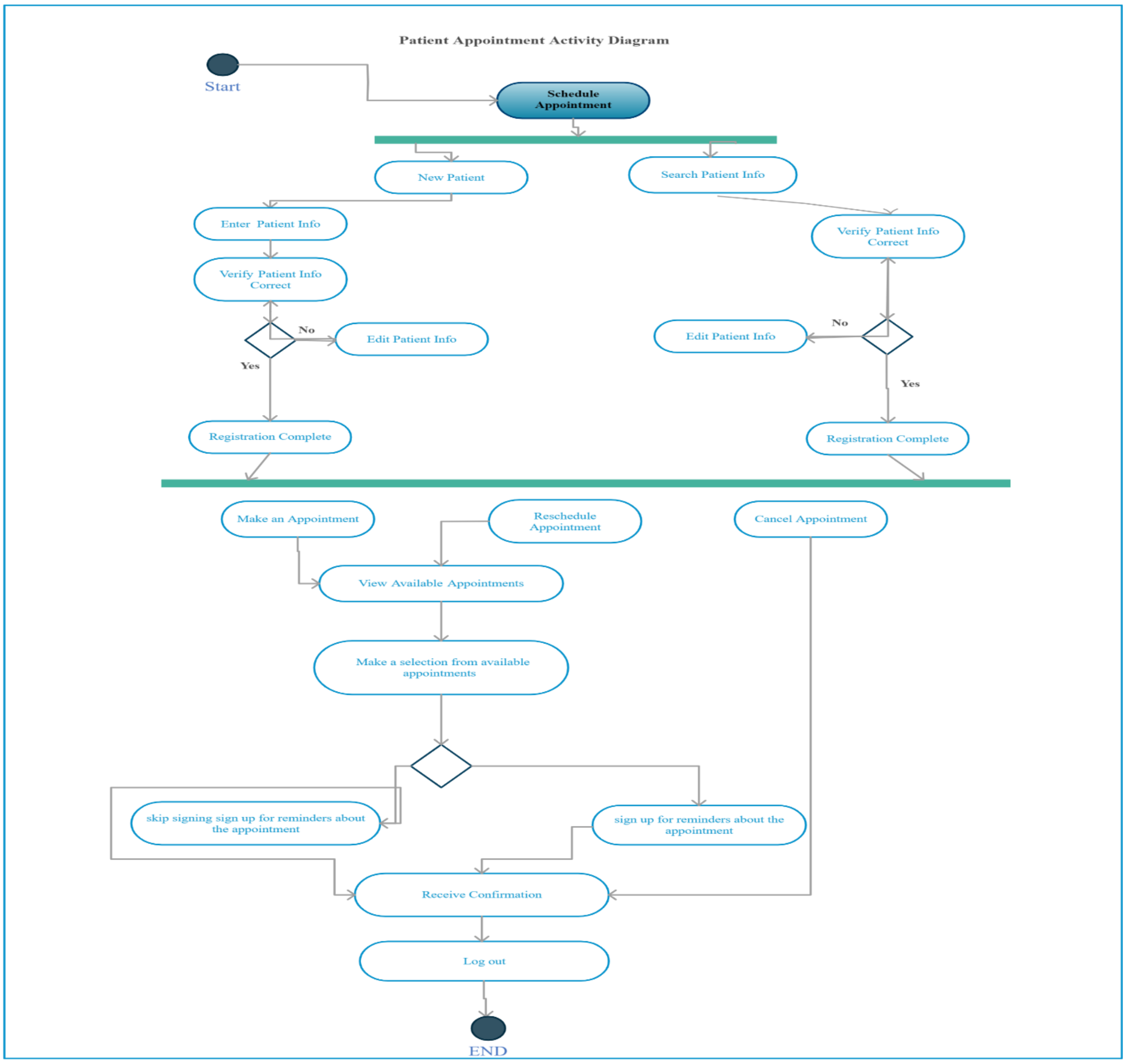

Patient Appointment Activity Diagram
Start
Schedule Appointment
New Patient
Search Patient Info
Enter Patient Info
Verify Patient Info Correct
No
Edit Patient Info
Yes
Registration Complete
Verify Patient Info Correct
No
Edit Patient Info
Yes
Registration Complete
Make an Appointment
Reschedule Appointment
Cancel Appointment
View Available Appointments
Make a selection from available appointments
skip signing sign up for reminders about the appointment
sign up for reminders about the appointment
Receive Confirmation
Log out
END

Fig. 3: Activity Diagram

Data Flow Diagram:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Level 0: DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analysed or modelled.

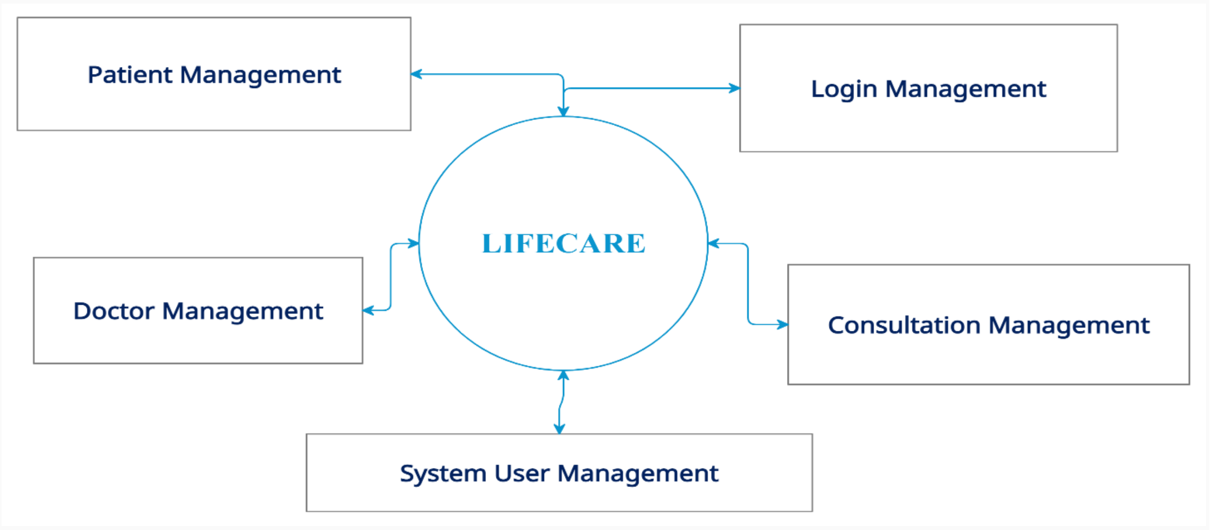


Fig. 4.1: Level 0 DFD

Level 1: DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

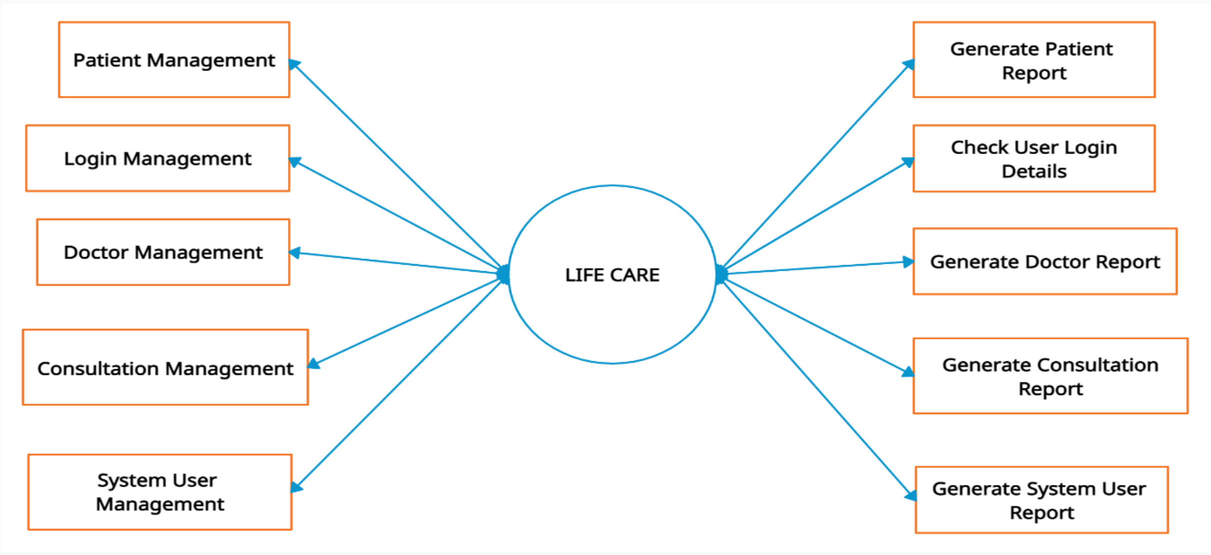


Fig. 4.2: Level 1 DFD

Level 2: DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system's functioning.

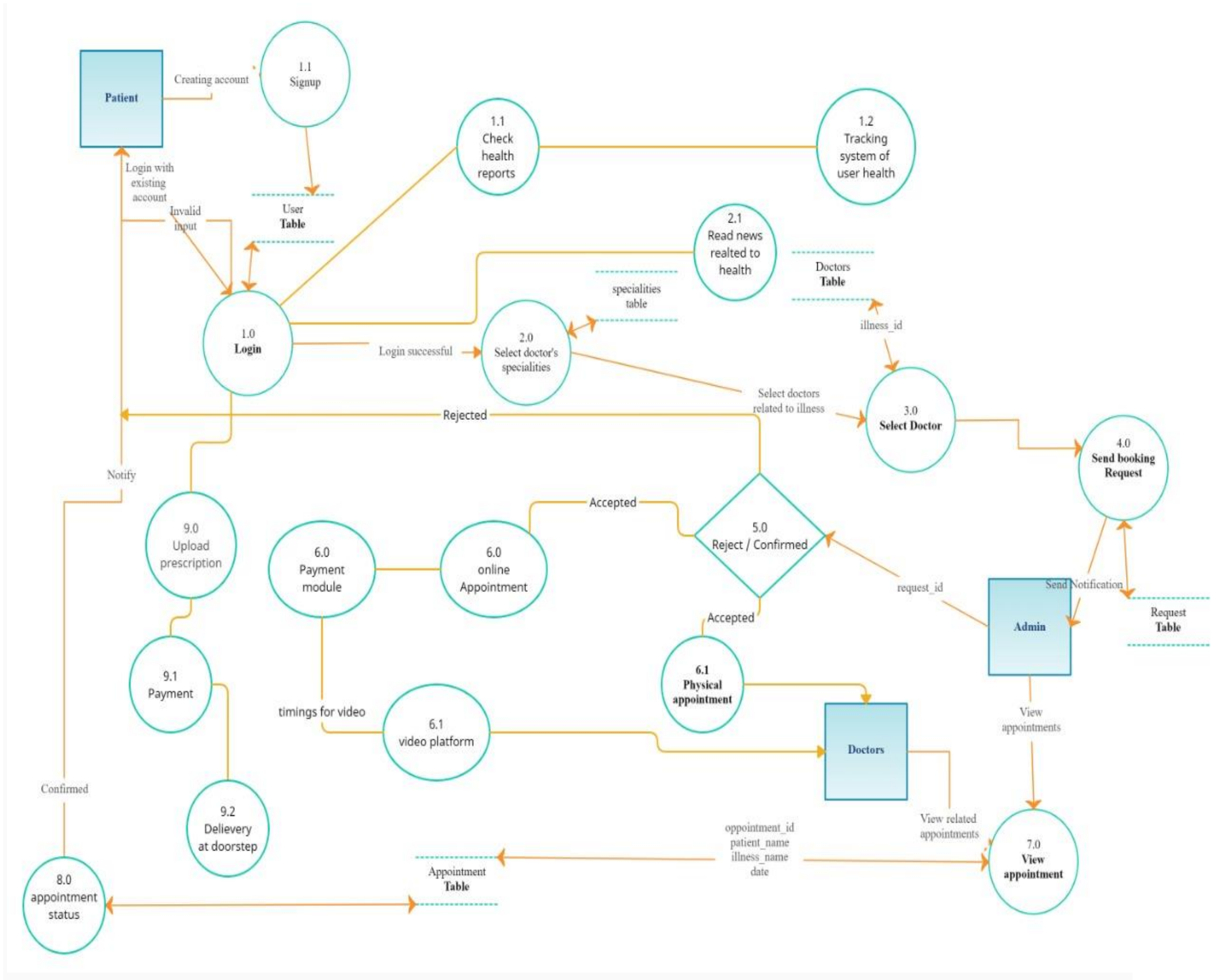


Fig. 4.3: Level 2 DFD

3.4 Software Specification

HTML:

Hypertext Markup Language is the standard markup language used to create web pages. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not

display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites.
HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

CASCADING STYLE SHEETS (CSS):
It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.
CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colour, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

JAVASCRIPT:
JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

Properties are the values associated with a JavaScript object.
A JavaScript object is a collection of unordered properties.
Properties can usually be changed, added, and deleted, but some are read only.

NODE.JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

MONGODB:

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database. MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model. MongoDB Atlas is the multi-cloud database service for MongoDB available on AWS, Google Cloud, and Azure. Best-in-class automation and built-in proven practices provide continuous availability, elastic scalability, and support with regulatory compliance.

CHAPTER 4

SYSTEM IMPLEMENTATION & TESTING

4.1 Project Description

4.1.1 User Interface

Login Page

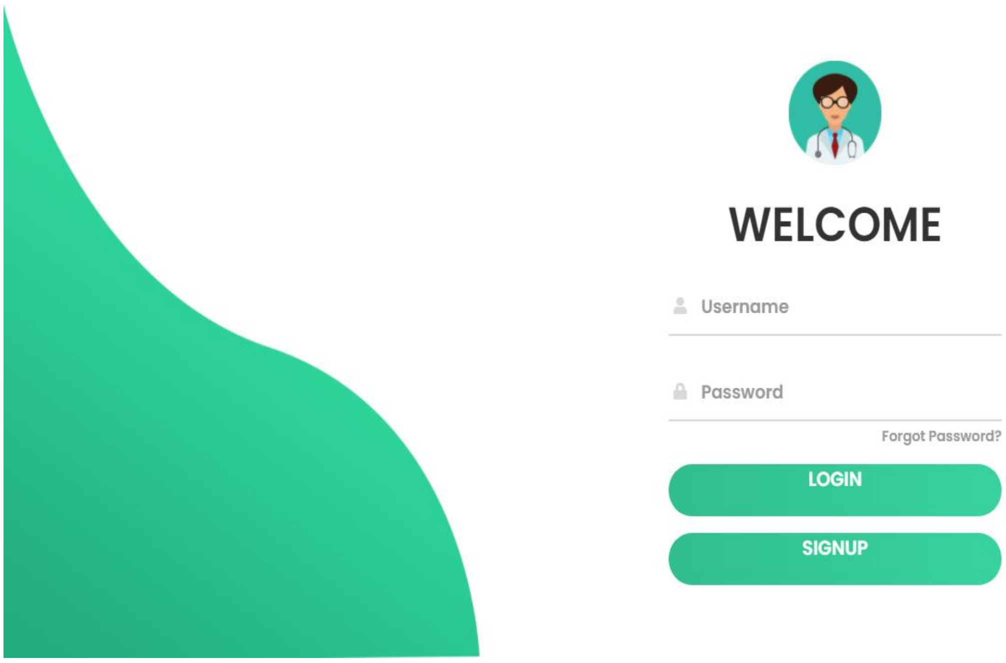


Fig. 5.1: Login Page

Sign Up Page

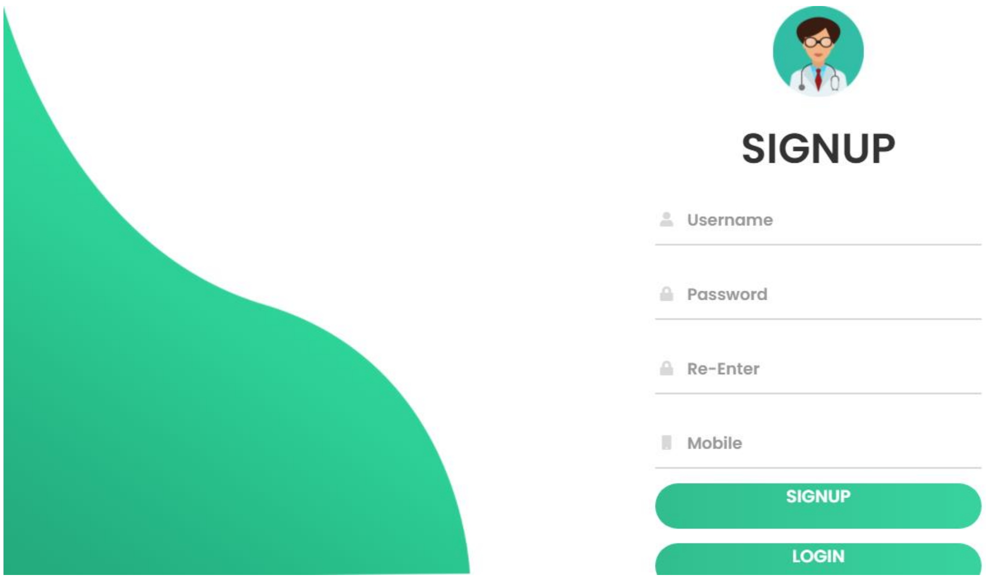
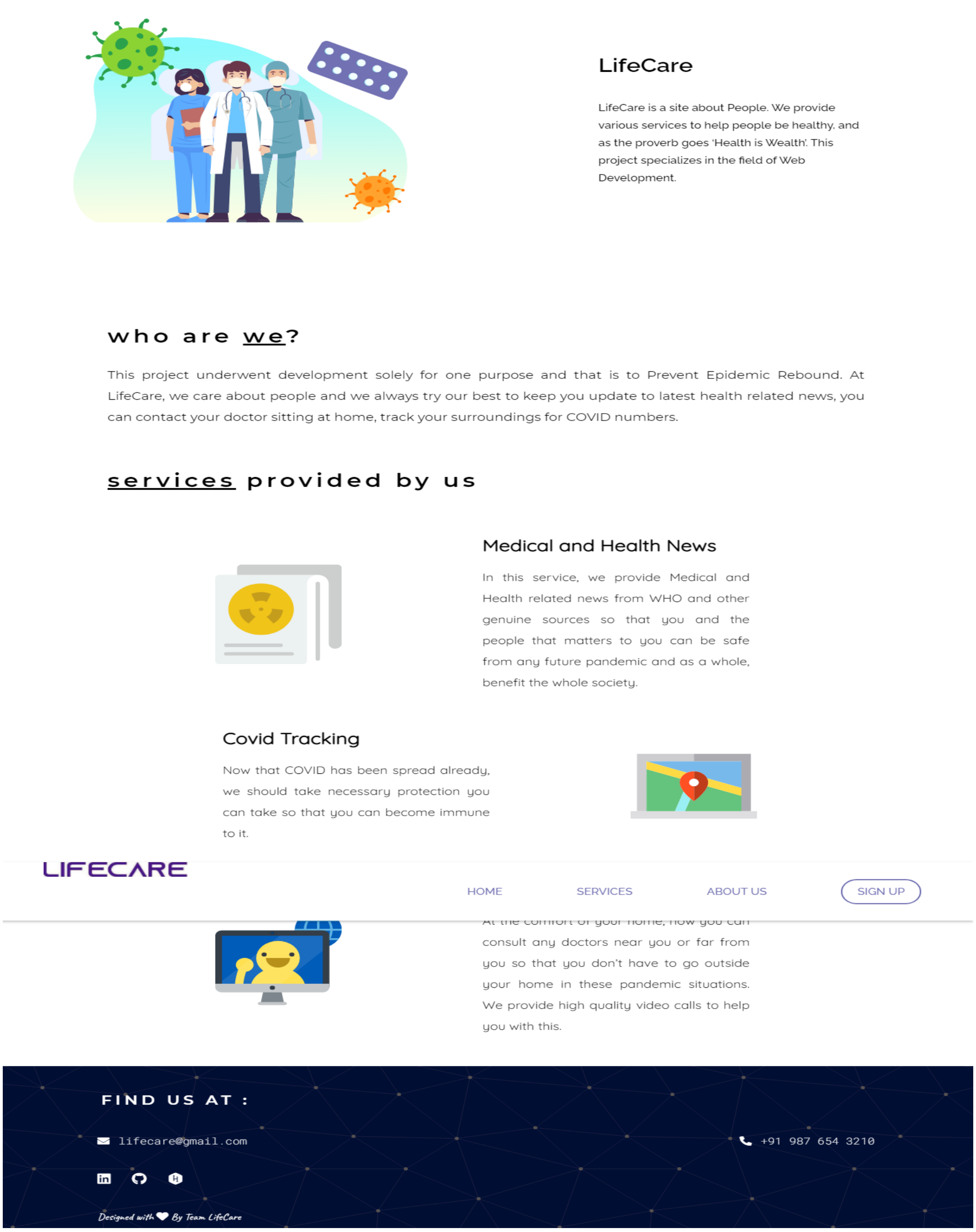


Fig. 5.2: Sign up Page

Home Page



LifeCare

LifeCare is a site about People. We provide various services to help people be healthy, and as the proverb goes 'Health is Wealth'. This project specializes in the field of Web Development.

who are we?

This project underwent development solely for one purpose and that is to Prevent Epidemic Rebound. At LifeCare, we care about people and we always try our best to keep you update to latest health related news, you can contact your doctor sitting at home, track your surroundings for COVID numbers.

services provided by us

Medical and Health News

In this service, we provide Medical and Health related news from WHO and other genuine sources so that you and the people that matters to you can be safe from any future pandemic and as a whole, benefit the whole society.

Covid Tracking

Now that COVID has been spread already, we should take necessary protection you can take so that you can become immune to it.

At the comfort of your home, now you can consult any doctors near you or far from you so that you don't have to go outside your home in these pandemic situations. We provide high quality video calls to help you with this.

FIND US AT :

lifecare@gmail.com

+91 987 654 3210

Designed with ♥ By Team LifeCare

Fig. 5.3: Home Page

Doctors Page

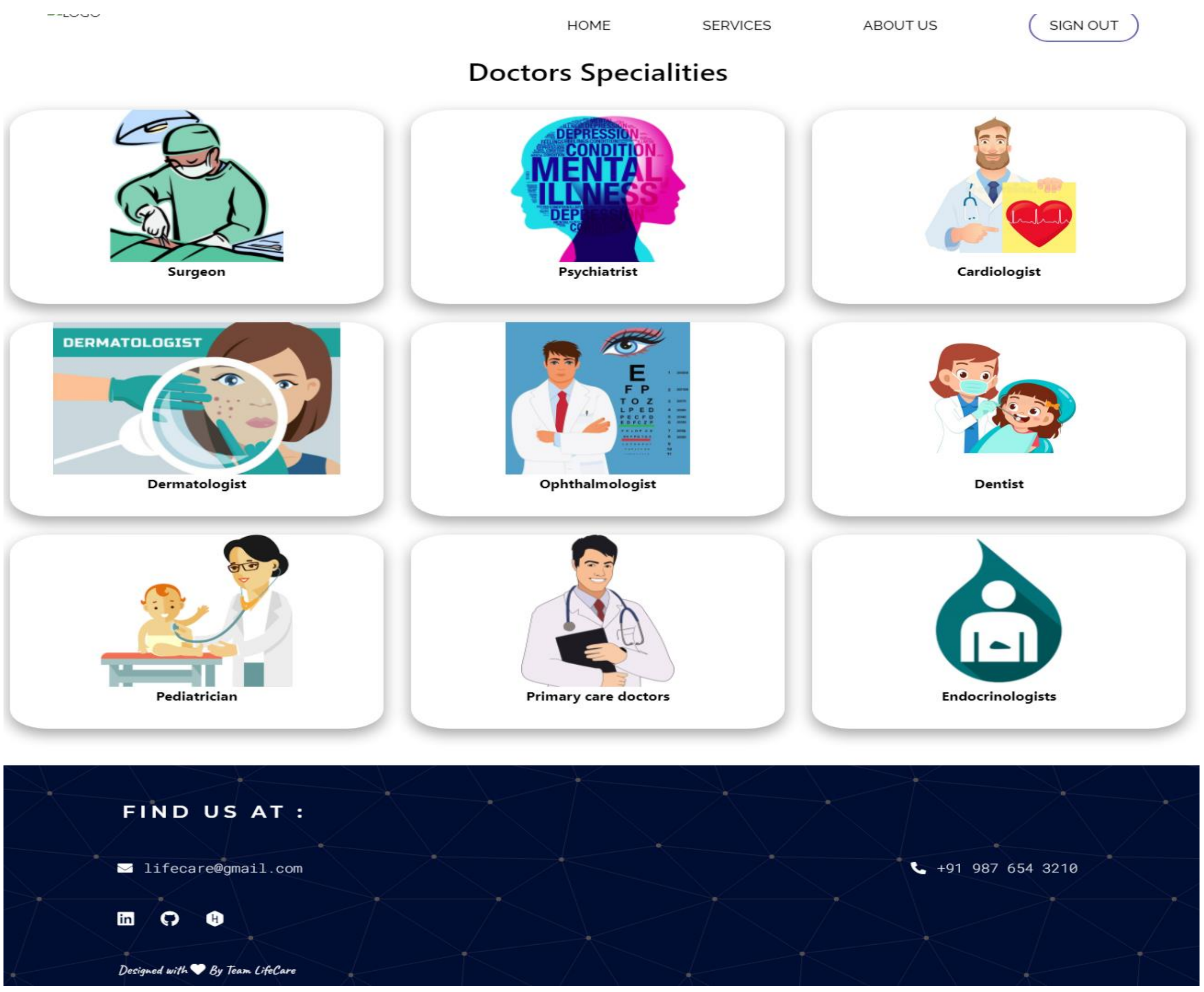


Fig. 5.4: Doctor specialization Page

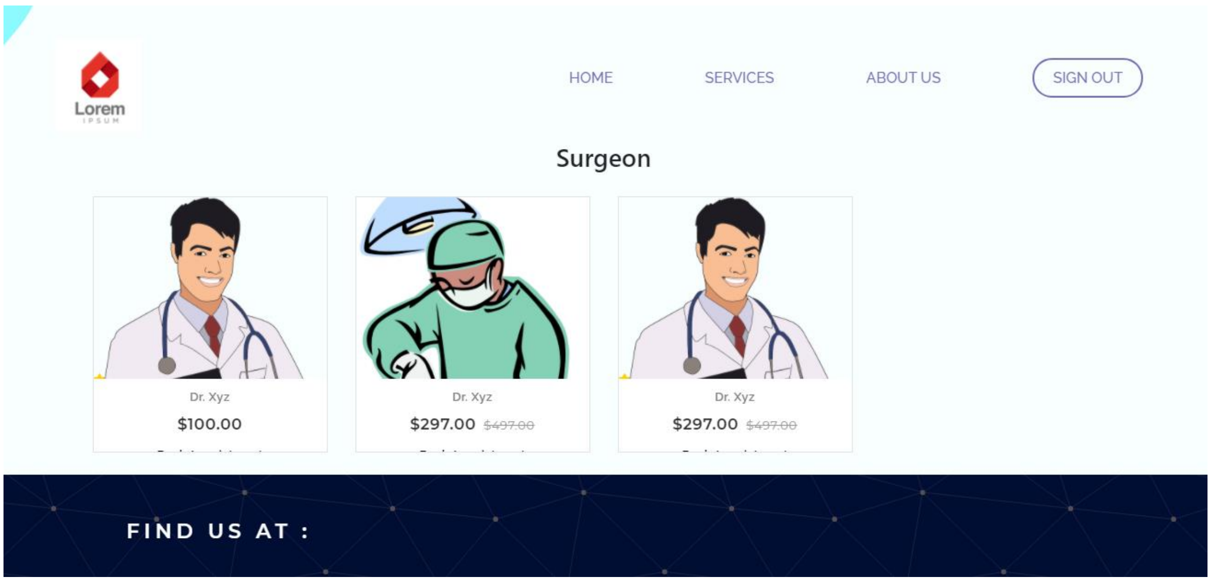


Fig. 5.5: Doctors Page

Services Page

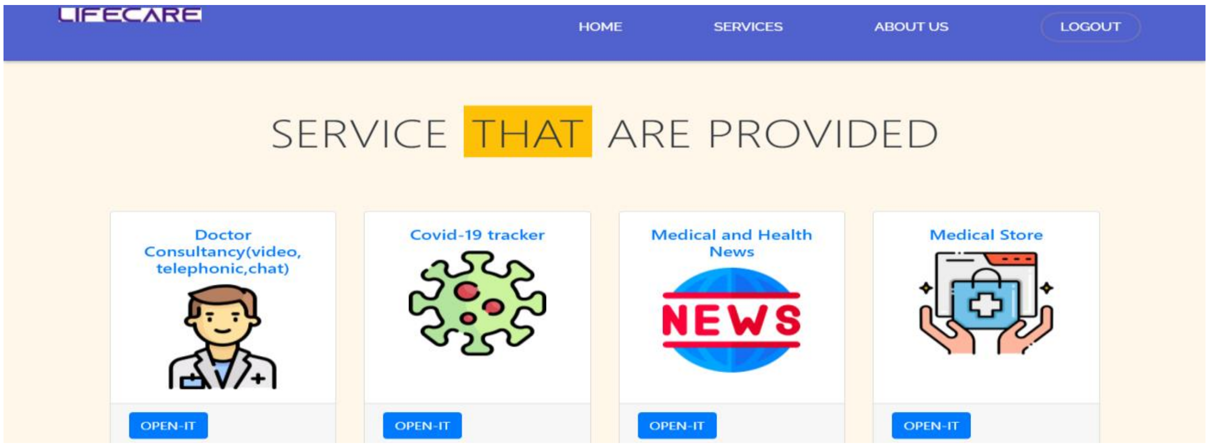


Fig. 5.6: Services Page

COVID Tracking Page

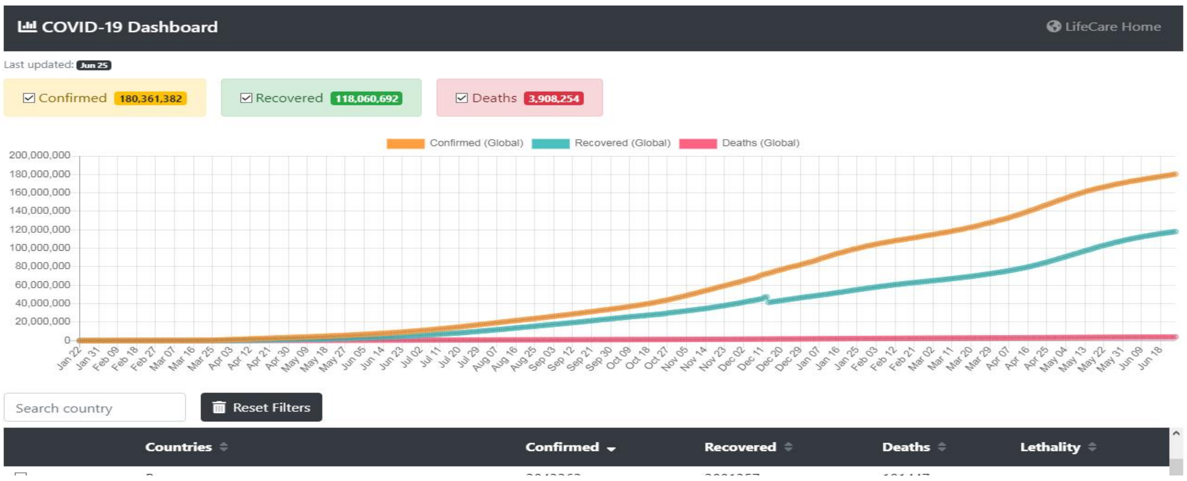


Fig. 5.7: Covid Track Page

Consult form Page

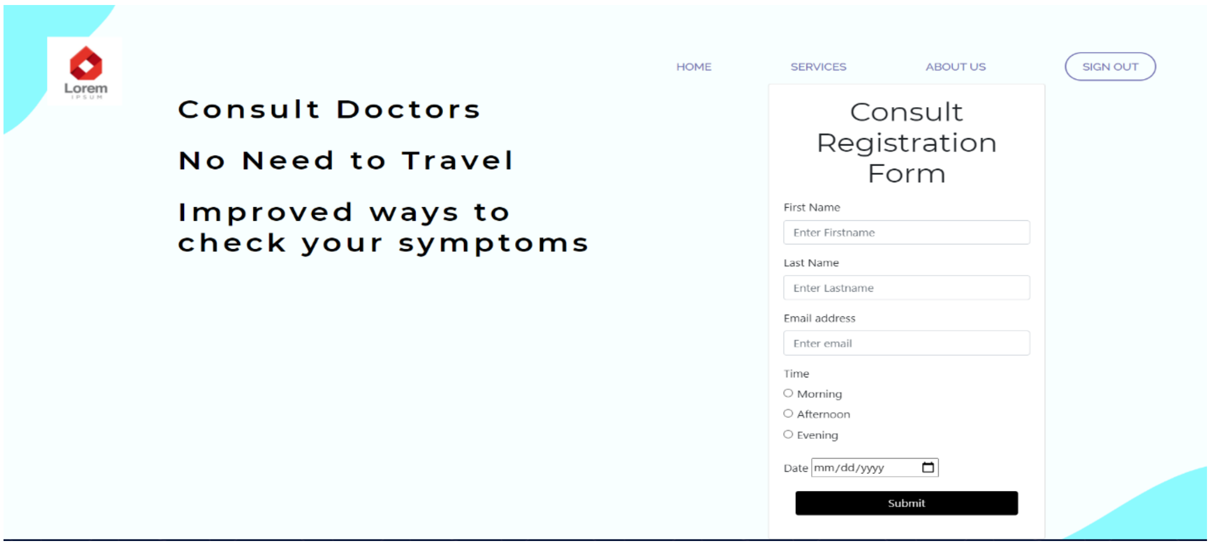
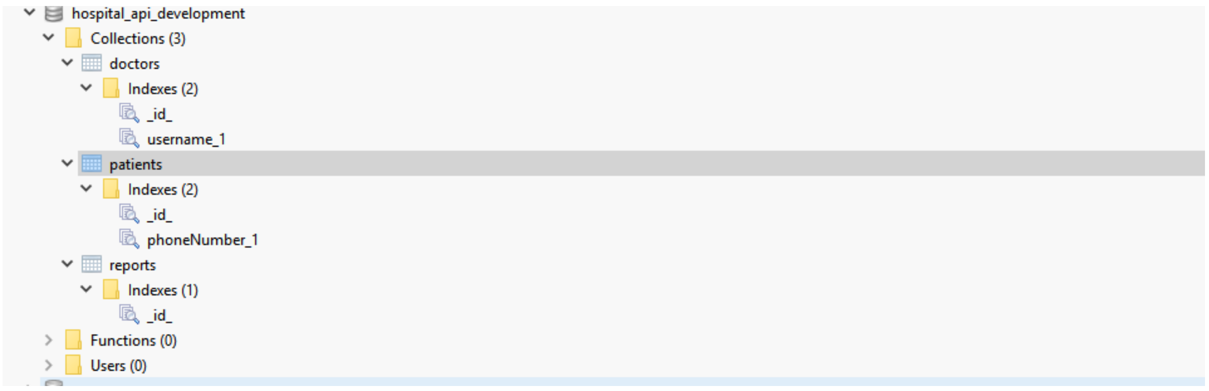


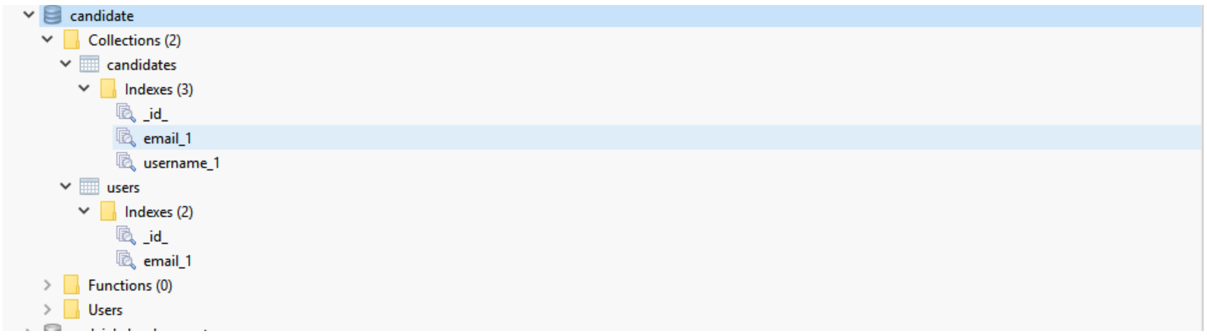
Fig. 5.8: Consult Form Page

4.1.2 Database



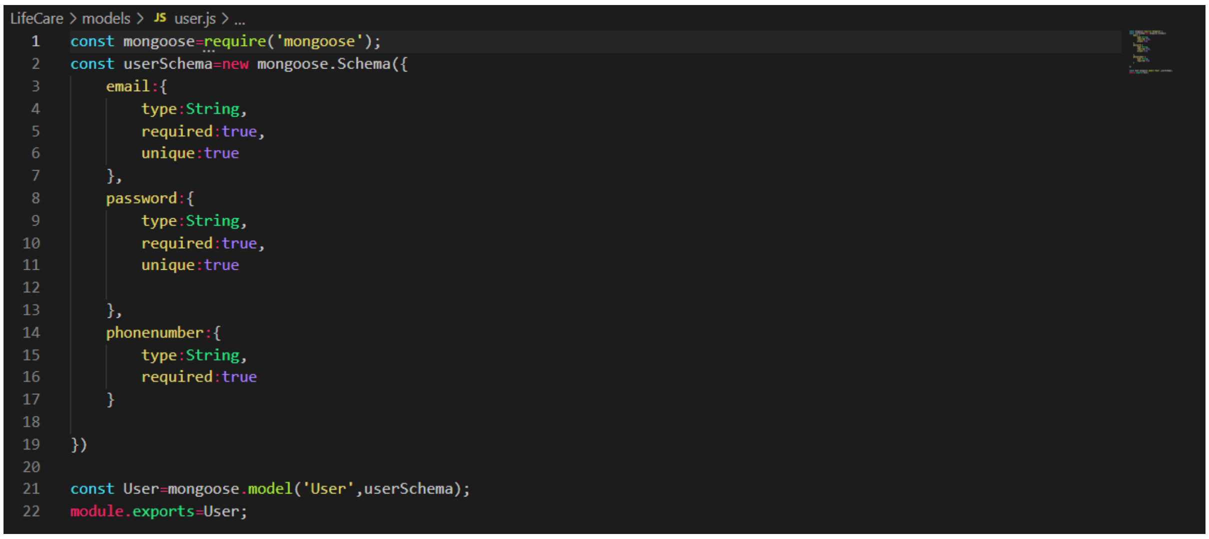
```
hospital_api_development
  Collections (3)
    doctors
      Indexes (2)
        _id_
        username_1
    patients
      Indexes (2)
        _id_
        phoneNumber_1
    reports
      Indexes (1)
        _id_
  Functions (0)
  Users (0)
```

Fig. 6.1: Mongodb Structure



```
candidate
  Collections (2)
    candidates
      Indexes (3)
        _id_
        email_1
        username_1
    users
      Indexes (2)
        _id_
        email_1
  Functions (0)
  Users
```

Fig. 6.2: Db Indexes



LifeCare > models > JS user.js > ...

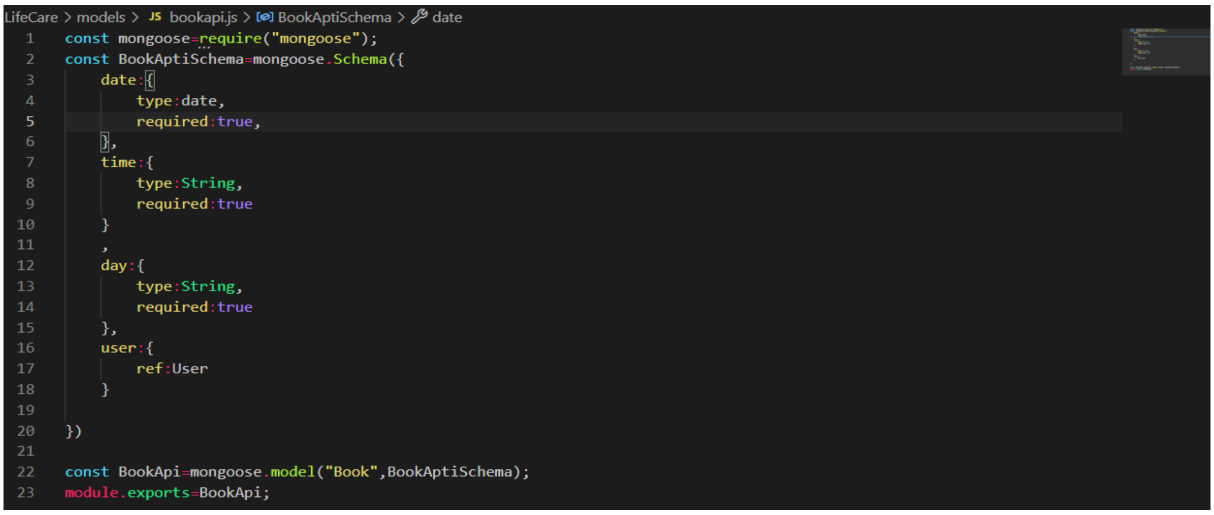
```javascript
const mongoose=require('mongoose');
const userSchema=new mongoose.Schema({
    email:{
        type:String,
        required:true,
        unique:true
    },
    password:{
        type:String,
        required:true,
        unique:true

    },
    phonenumber:{
        type:String,
        required:true
    }

})

const User=mongoose.model('User',userSchema);
module.exports=User;
```

Fig. 6.3: User Schema


LifeCare > models > bookapi.js > BookAptiSchema > date

```
const mongoose=require("mongoose");
const BookAptiSchema=mongoose.Schema({
    date:{
        type:date,
        required:true,
    },
    time:{
        type:String,
        required:true
    }
    ,
    day:{
        type:String,
        required:true
    },
    user:{
        ref:User
    }

})

const BookApi=mongoose.model("Book",BookAptiSchema);
module.exports=BookApi;
```

Fig. 6.4: Booking Schema

4.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

4.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

4.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

4.2.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

Testing Methods:

Testing is a process of executing a program to find out errors. If testing is conducted successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

White Box Testing:

It is a test case design method that uses the control structures of the procedural design to derive test cases. using this testing a software Engineer can derive the following test cases:

Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity.

Black Box Testing:

It is a test case design method used on the functional requirements of the software. It will help a software engineer to derive sets of input conditions that will exercise all the functional requirements of the program. Black Box testing attempts to find errors in the following categories:

Incorrect or missing functions

Interface errors

Errors in data structures

Performance errors

Initialization and termination errors

By Black Box Testing we derive a set of test cases that satisfy the following criteria:

Test cases that reduce by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing.
Test cases that tell us something about the presence or absence of classes of errors rather than errors associated only with a specific test at hand.

Test Approach :

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system.

Top down approach:

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided
stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred.

CHAPTER 5
CONCLUSION

Since we are entering details of the patients electronically in the” LIFECARE ”, data will be secured. Using this application we can retrieve patient’s history with a single click. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed. The need for the Health Centre to computerize the application processing and servicing the Patients request through automated modules is most necessary and now inevitable.

As we have already seen that the need cannot be emphasized for the further development of this system is only timely and helpful to Health Centre, the system defined in the above script is up to date and caters to all kinds of request faced by the Health Centre employees requirements to provide the better service to the patients, being developed in java it is also flexible modularized highly parameterized and hence can be easily deployed by any other application because of its componentized approach.

Thus this project is developed from the beginning with reuse in mind and implicitly uses several design patterns. The architecture of this project is such that it suits the diverse and distributed nature of the Health Centre Applications.

The features provided by the (Lifecare) are in no means comprehensive but by all means full filling all important functionalities of Health Centre services. Inclusion of further functionalities as days go by can be easily done because the project has been developed in a layered architecture. Plug-in modules would easily add new features which change with the times and being performance oriented the project will not face any issues. It is also extensible and scalable as all applications should be thus it can be said that it will meet surges of huge employee and patient requests that may come up in the near future.

REFERENCES

1. IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. OMG. "Unified Modeling Language Specification", Superstructure Version 2.1.1, Febrer 2007
3. [ChC90] E.J.Chikofsky, J.H.Cross. "Reverse Engineering and Design Recovery: A Taxonomy", IEEE Software, Gener-Febrer 1990 Vol.7 Núm. 1 pp.13-17
4. [ChL99] L.L.Constantine, L.A.D.Lockwood. "Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design", Addison-Wesley Professional, 1999
5. [CST03] D.Costal, M.R. Sancho, E.Teniente. "Enginyeria del Software: Especificació. Especificació de sistemes orientats a objectes amb la notació UML", Edicions UPC, 2003
6. [FAT03] E.Faivre, L.Abbal, T.Murail. "EasyPHP", 2003 http://www.easyphp.org/
7. [GNU91] Free Software Foundation, Inc. "GNU General Public License", Version 2, Juny 1991 http://www.gnu.org/licenses/gpl.html
8. [Oli02] A.Olivé. "Modelització conceptual de Sistemes d'Informació. L'estructura", Edicions UPC, 2002
9. [Oli07] A.Olivé. "Conceptual Modeling of Information Systems", Springer, 2007
10. [OMG06] OMG. "Object Constraint Language Specification", Version 2.0, 2006 http://www.omg.org/technology/documents/formal/ocl.htm
11. [OMG07] [USE07] University of Bremen. "A UML-based Specification Environment", 2007 http://www.db.informatik.unibremen.de/projects/USE/

12.N. (n.d.). *About*. Node.Js. Retrieved May 8, 2021, from https://nodejs.org/en/about/

13.*Introduction to HTML*. (n.d.). Html. Retrieved July 3, 2021, from https://www.w3schools.com/html/html_intro.asp

14.*JavaScript Tutorial*. (n.d.). JavaScript. Retrieved July 3, 2021, from https://www.w3schools.com/js/default.asp

15.*The MongoDB 4.4 Manual — MongoDB Manual*. (n.d.). MongoDB. Retrieved May 8, 2021, from https://docs.mongodb.com/manual/

16.*Node.js Get Started*. (n.d.). Node. Retrieved May 10, 2021, from https://www.w3schools.com/nodejs/nodejs_get_started.asp

17.M. A. Akbar et al., "Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model" in IEEE Access, vol. 6, pp. 4811-4823, 2018, doi:10.1109/ACCESS.2017.2787981, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=8241771&amp;isnumber=8274985

18.I. H. Lim et al., "Security Protocols Against Cyber Attacks in the Distribution Automation System" in IEEE Transactions on Power Delivery, vol. 25, no. 1, pp. 448-455, Jan. 2010, doi:10.1109/TPWRD.2009.2021083, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=5299213&amp;isnumber=5361446

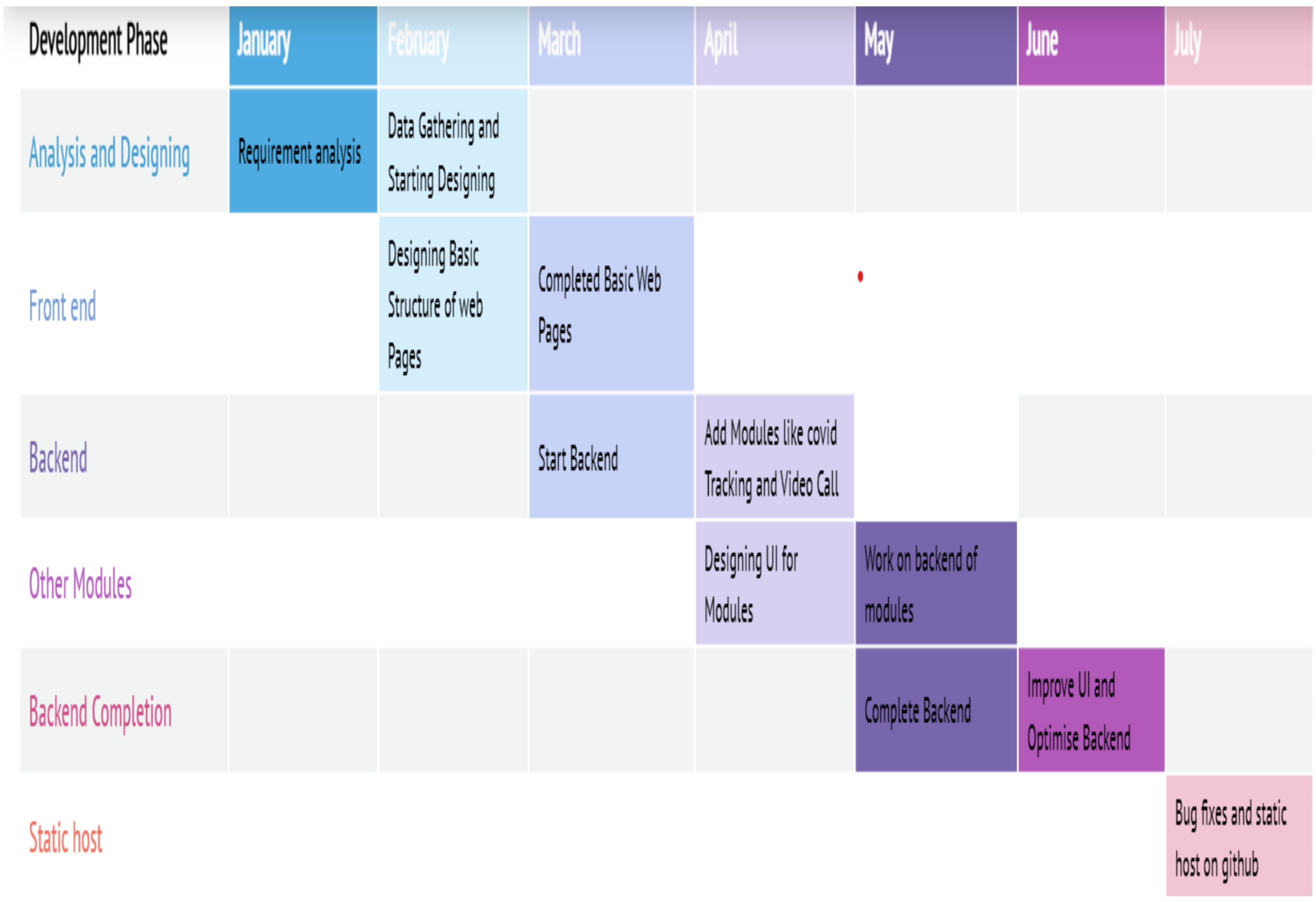
19. Li Chunlin, “;A Java-based method for developing Web application system” Fifth Asia-Pacific Conference on ... and Fourth Optoelectronics and Communications Conference on Communications,1999, pp. 1079-1082 vol.2, doi: 10.1109/APCC.1999.820450, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=820450 &amp;isnumber=17751

20. N. Zhang, Y. Cao and S. Zhang, “Research of web front-end engineering solution in public cultural service project” 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 2017, pp. 623-626, doi: 10.1109/ICIS.2017.7960067, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=7960067 &amp;isnumber=7959951

21. M. Cho and C. Lee, “A low-power real-time operating system for ARC (actual remote control) wearable device” in IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 1602-1609, Aug.2010, doi: 10.1109/TCE.2010.5606303, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=5606303 &amp;isnumber=5606236

APPENDIX A

Planning of Work:



Development Phase	January	February	March	April	May	June	July
Analysis and Designing	Requirement analysis	Data Gathering and Starting Designing					
Front end		Designing Basic Structure of web Pages	Completed Basic Web Pages				
Backend			Start Backend	Add Modules like covid Tracking and Video Call			
Other Modules				Designing UI for Modules	Work on backend of modules		
Backend Completion					Complete Backend	Improve UI and Optimise Backend	
Static host							Bug fixes and static host on github

Fig. 7: Gantt Chart

APPENDIX B

Code Snippets

For creating a new server :

```
const express = require(&#39;express&#39;)
const app = express()
app.get(&#39;/&#39;, function (req, res) {
res.send(&#39;Hello World&#39;)
})
app.listen(3000)
```

Connecting to mongoDb:

```
const mongoose=require(&#39;mongoose&#39;);
mongoose.connect(&#39;mongodb://localhost/LifeCare&#39;, {
useCreateIndex: true });
const db=mongoose.connection;
db.on(&#39;error&#39;,console.error.bind(console,&quot;Err
or connecting to MongoDB&quot;));
db.once(&#39;open&#39;,function(){
console.log(&#39;Connected to Database :: MongoDB&#39;);
});
module.exports = db;
```

User Schema:

```
const mongoose=require(&#39;mongoose&#39;);
const userSchema=new mongoose. Schema({
email:{
```

```
type:String,
required:true,
unique:true
},
password:{
type:String,
required:true,
unique:true
}
})
const User=mongoose.model(&#39;User&#39;,userSchema);
module.exports=User;
```

MongoDb Search by email and password:

```
User.findOne({email: req.body.email}, function(err, user){
if(err){console.log(&#39;error in finding user in signing
up&#39;); return;}
if (!user){
User.create(req.body, function(err, user){
if(err){console.log(&#39;error in creating user while
signing
up&#39;); return;}
return res.redirect(&#39;/paths/SignUp&#39;);
})
}else{
return res.redirect(&#39;/paths/Login&#39;);
}
});
```