Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Informatics

Diploma Thesis

Using MVC Framework implement dynamic change of relational model for development.

Mehul Dakubhai Hirpara

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Mehul Dakubhai Hirpara

Systems Engineering and Informatics
Informatics

Thesis title

Using MVC Framework implement dynamic change of relational model for development

Objectives of thesis

Today's software industry faces rapid growth due to the innovations in the areas of applications and services. The previous approach to development was rigid, making it difficult to maintain and grow online applications such as relational database has an issue like dynamical change of relation model is more difficult and this issue increase extra cost for software development company. Using ORM tool (Hibernate framework) we will try to implement solution which can help to dynamical change in relational model additionally, this scenario will be tested with the Model, View, and Controller design pattern.

This thesis aims is to create a lightweight application that demonstrates how to dynamically update relation models using the ORM tool when the database changes during the development process.

Methodology

In the theoretical component of the thesis, the steps of designing an application will be studied by using scientific literature. Following that, the technical components will be reviewed, with concepts and a suitable process

established. Furthermore, theoretical knowledge will be applied in the development of an application. The MVC architectural best practices for development and maintenance will be used to design an application. The final application will be able to provide an online appointment for the hospital, allowing patients to make appointments without having to call, reducing the strain of telephonic appointments, and doctors to supply their available time through the online application.

The proposed extent of the thesis

50-60 pages

Keywords

JAVA, Spring MVC architecture, database, React, ORM tool(Hibernate)

Recommended information sources

Josef Pavlíček, Cookbook of interaction design for HCI and ID, Online material: https://docs.google.com/presentation/d/1nbLjgEX5mS6kl_cRx6CeKuhd-fzz-kyYn_j03vMLkH4/edit?usp=sharing

Expected date of thesis defence

2022/23 SS – FEM

The Diploma Thesis Supervisor

Ing. Josef Pavlíček, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 8. 3. 2022	Electronic approval: 14. 3. 2022
Ing. Martin Pelikán, Ph.D.	**doc. Ing. Tomáš Šubrt, Ph.D.**
Head of department	Dean

Prague on 28. 11. 2022

Declaration

I declare that I have worked on my diploma thesis titled **" Using MVC Framework implement dynamic change of relational model for development."** by myself, and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that it does not break the copyrights of any person.

In Prague on date of submission 30.11.2022 ____________________________

Acknowledgment

I would like to thank Ing. Josef Pavlicek, Ph.D. for his help in preparing the thesis and invaluable recommendations. He supplied invaluable support from the beginning of the topic choice until the completion of the thesis. I want to thank all professors at the Czech University of Life Sciences for providing knowledge in economics and management.

Using MVC Framework implement dynamic change of relational model for development.

Abstract

Today's software industry faces rapid growth due to innovations in the areas of applications and services. The previous approach to development was rigid, making it difficult to maintain and grow online applications such as relational databases an issue like a dynamical change of relation model is more difficult, and this issue increases the extra cost for the software development company. Using the ORM tool (Hibernate framework) we will try to implement the solution which can help with dynamic change in the relational model additionally, this scenario will be tested with the Model, View, and Controller design pattern. This thesis aims to create a lightweight application that demonstrates how to dynamically update relation models using the ORM tool when the database changes during the development process.

In the theoretical component of the thesis, the steps of designing an application will be studied using professional literature. Following that, the many technical components of the ORM tool will be reviewed, with concepts and a suitable process established. Furthermore, theoretical knowledge will be applied in the development of an application. The MVC architectural best practices for development and maintenance will be used to design an application. The final application will be able to provide an online appointment for the hospital, allowing patients to make appointments without having to call, reducing the strain of telephonic appointments, and doctors to supply their available time through the online application.

Keywords: JAVA, MVC architecture, database, React, ORM tool(Hibernate)

Pomocí MVC Framework implementujte dynamickou změnu relačního modelu pro vývoj.

Abstraktní

Dnešní softwarový průmysl čelí rychlému růstu díky inovacím v oblastech aplikací a služeb. Předchozí přístup k vývoji byl rigidní, takže bylo obtížné udržovat a rozvíjet online aplikace, jako jsou relační databáze, problém jako dynamická změna relačního modelu je obtížnější a tento problém zvyšuje dodatečné náklady pro společnost zabývající se vývojem softwaru. Pomocí nástroje ORM (Hibernate framework) se pokusíme implementovat řešení, které může pomoci s dynamickou změnou v relačním modelu navíc, tento scénář bude testován s návrhovým vzorem Model, View a Controller. Tato práce si klade za cíl vytvořit odlehčenou aplikaci, která demonstruje, jak dynamicky aktualizovat relační modely pomocí nástroje ORM při změně databáze během procesu vývoje.

V teoretické části práce budou nastudovány kroky návrhu aplikace s využitím odborné literatury. Poté bude přezkoumáno mnoho technických součástí nástroje ORM, budou stanoveny koncepce a vhodný proces. Dále budou teoretické znalosti aplikovány při vývoji aplikace. K návrhu aplikace budou použity architektonické osvědčené postupy MVC pro vývoj a údržbu. Konečná aplikace bude schopna poskytnout nemocnici online schůzku, což pacientům umožní domluvit si schůzku, aniž by museli volat, čímž se sníží námaha při telefonických schůzkách a lékaři budou moci poskytovat svůj volný čas prostřednictvím online aplikace.

Klíčová slova: JAVA, architektura MVC, databáze, React, nástroj ORM (Hibernate)

Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction

A Web Application is a distributed application that runs on more than one Computer and communicates through a Network or a Server. Specifically, a Web Application is accessed with a Web Browser as a client and provides the facility to update and manage a program without deploying and installing an Application on Client Computers. Web Applications are used for Email, Online Retail Purchases/Sales, and Online Banking, among others. Moreover, the social world has slowly shifted from interpersonal connections to online connections, the business and organizations are also doing the same. Website development is considered to be a significant tool for businesses. The advantage of Web Applications is that can be accessed and used by millions of people at the same time (Botwe and Davis, 2015). David and we can keep the record of the user in our database which can reduce our burden of physical data storage of the user. Early, most organizations were using static web applications so was increasing the burden of the organization. For example, Company was developing its business web application as static so they cannot do any changes after it is done. When the developer gets some changes in the product at the end of the delivery it was so difficult to hand it in.

In this analysis, we will introduce MVC architecture with a dynamic relational database that helps developer implement code in a modular manner. Also, we will introduce a dynamic relational database where we will store data in a different table with specific relations also, we will introduce dynamic table creation at development time that will help the developer. MVC is a software design pattern that is based on the interconnection of three major component types: Model, View, and Controller, with a significant emphasis on object-oriented programming (OOP) software paradigms. MVC is a framework for creating web applications that follow an MVC design. It is now the most prominent software development architecture. This design manages the code automatically and assists the programmer in developing well-managed Web Applications (Thakur and Pandey, 2019). In the backend, we will use a relational database that will store data in tabular form with specific relations so when we get any changes during the development then the developer can change it without too much time.

In this research, we will implement an online appointment system for hospitals and show, how MVC architecture is used in real-world applications with relational databases. The web- page for the online appointments is made through the help of HTML and data is stored using MySQL. This platform enables patients to book or schedule their time appointments related to health issues. There are various benefits to booking through the web system. The patient can check the time, doctor's name as per their problems, availability of the doctors, their timing, etc. Further, it saves the time of the patients and creates a virtual relationship between the patient and doctors. Through online appointments, patients can also consult with others. Thus a web system is generated for various purposes that are advantageous to both the patient and the doctors. The particular solution can provide notifications that will appear automatically on the visuals. However, on the online portal, the topmost is being suggested which will help the patient to know most of the details of the doctors and also acknowledge the best one for them. The online web system that is being created through HTML has made it easy for the individual to use. The low languages are mainly used for the development of the web page regarding online appointments. This web page can be created with the help of "dynamic pages", but for the web application, information has to be dynamic. On the back end side, we will introduce the MySql database so we can easily store our data in a well-defined format also we can add a new table at development time.

1.2 Objectives

The current study's major goal is to develop a web application that will aid in demonstrating an understanding of dynamic change in relational databases. The following objectives are structured for this purpose:

1. To investigate and have a grasp of prior technologies utilized in database creation.
2. To develop the website using MVC architecture, which aids in understanding the capabilities and qualities of this technology also we will develop a small code where we will create a table based on input information.
3. To study and test the website built with MVC technology and its application in the actual world, demonstrating how relational data changes automatically.

1.3 Research problem

1. How does the relationship of table perform with MVC methodologies when utilized for website development?
2. What software stacks should be used in the development of the website?

1.4 Problem statement

The research will help the readers to understand the importance of designing the websites and their needs in the medical field that we will implement in View. After that, we design models as per patient, doctor, and admin requirements so we can easily manage our data in a relational database base on models after, it we will create another chunk of code in java based on it we will create a dynamic table in the database so front end developer does not require to create a new table in the database. The last and most important part is the controller where we will define business logic and control all the website operations. We will get to know how the design and dynamic changes in form and database have helped out the people in terms of time and stress without background knowledge of the implementation. Through the website, the person can also directly interact with doctors for a solution regarding health issues. With the help of the research, the reader will get familiar with the MVC that is being used in REACT for developing dynamic websites. The Model-View-Controller (MVC) architecture is extremely beneficial for creating interactive and dynamic web applications. It has evolved into the most powerful and influential programming paradigm for creating large-scale and dynamic web applications. Developers can rely on MVC design patterns, which are widely regarded as solutions to recurring problems and are used to create adaptable, reusable, and modular software. Applying the MVC design pattern to Web Applications is thus made more difficult by the fact that current technologies encourage developers to segment the application as early as the design phase.

1.5 Research Significance

The analysis of online appointments has a great significant role in strengthening the relationship between patients and doctors. There is rapid growth in technology and it has made it easy for people to work. The use of software has been used rapidly for the growth of business in every field. With the use of REACT and SQL language, the web page for the online application is being created so that it can become easy to handle and can access through the online portal. Both patients and doctors have found our online appointment system to be time- saving. Different pages with information on doctors, location, forms, etc. Which have been constructed using the REACT programming language (sourcetoad.com, 2020). REACT is used to develop front ends (VIEW) and back ends (MODEL, CONTROLLER), allowing both sides of the interface to operate on various website sides. A cross-platform REACT front end is being developed with the aid of the programming language in order to provide content. The "user interface" is referred to as the "front-end," while the "server, application, and database" are referred to as the "back-end," which will function in the background to send the detailed information. The online appointment is a "smart web application" that will provide doctors and patients with a mechanism to book an online appointment and, login for doctor as well as admin. The website or online portal that was created would provide doctors access to register by offering information that would help patients receive all the necessary and particular information. The information on the page will content the dynamic information about the doctors' hours of availability, their fees for certain parameters, etc (Yadav, 2022). Doctors who register themselves on the website can quickly log in using their registered email address and password. There will be a variety of hacked pages among the various pages that will be categorized. In the dynamic website, we will generate the dynamic page as per the time slot released by doctors if there is not any slot for time then patient will not get the dynamic page. As a result, a web-based system is created for a variety of uses that benefits both patients and medical professionals. The specific solution has the capacity to offer notifications by email that people will receive automatically. However, the greatest doctors are recommended on the top of the page. The research will focus on enveloping the different modules on a platform that will be helpful for the clients to understand (Pop and Altar, 2014). The research will focus on designing the website with help of MVC where we will use dynamic data so patients can get real-time data from the database and it reduce the extra work of the hospital industry.

1.6 Research Framework

The structure of the research which is followed is mentioned below;

1.6.1 Chapter 01

This chapter contributes to the study's summary by outlining its goals and objectives. This also addresses the significance of the study and its setting.

1.6.2 Chapter 02

This chapter examines the available resources and explores a variety of hypotheses and technological advances related to the current research. This covers the summer of modern technology as well as how it benefits

1.6.3 Chapter 03

This chapter provides all of the present study's problem analysis, tools, and tactics. Furthermore, we will include goals, use cases, model specifications. and technologies that we will employ in research, whether it will be implemented or not.

1.6.4 Chapter 04

In this chapter, I will use several usability tests to try to improve the website's design and learn how it works in the real world. Finally, we will send the results of the test.

1.6.5 Chapter 05

A chapter in the implementation section discusses and attaches the application's screen brief and describes each and every page with its function and how it is useful.

1.6.6 Chapter 06

In this chapter, I will use usability tests to try to improve the website's design and learn how it works in the real world. Finally, we will send the results of the test.

1.6.7 Chapter 07

Here, we will write the final conclusion of the researcher and discuss.

1.6.8 Chapter 08

Here, we will provide reference list that we use in this research.

2 Research of existing

2.1 Introduction.

Web applications have aided in the simplification of many of the tasks we conduct on a daily basis, making our lives easier. These programs are extensively utilized to help us solve difficulties at the hospital. Previously, these appointment processes were done manually, which resulted in many instances of overbooking or neglecting to cancel an appointment, which might free up time to schedule another in its place. A web application will be built to simplify the scheduling process and remove human error caused by manually booking appointments. Furthermore, given the hectic lifestyles that many of us lead today, an online appointment management system within a hospital makes ideal sense because it frees up valuable time not only for patients but also for staff. The purpose of this paper is to develop and evaluate an online appointment system for patients, where all processes of appointments are verified. Most aspects of appointment management, such as reservations, confirmations, and cancellations, are controlled automatically by the administration.

2.2 The issue in the existing appointment system.

To eliminate errors caused by human error, it is believed that this new system should be utilized instead of a manual one that requires a person to plan and cancel appointments. The scheduling procedure currently operates as follows.

1. A patient arrives at the clinic to make an appointment.
2. A member of staff records the information on a Scheduling Form. This form is copied and placed in the doctor's mailbox.
3. When the staff has time, the information is then entered into the doctor's calendar. This indicates that even if a patient requests the earliest appointment possible, it may not be placed into the system until the staff member gets around to it. There is a chance that another staff member will fill the slot with a different patient's request, forcing the initial patient's appointment to be rescheduled at an inconvenient time.

4. If the first patient needs to be rescheduled, the staff member must contact the patient, explain the mistake, and try to reschedule. This procedure consumes significant time that could be used to make the patient, staff member, and doctor more productive.
5. Some hospitals are still working on desktop applications, therefore, the patient is disabled to take advantage of the internet era in this case, patients have to call the clinic and schedule their appointment. It is more burden for the staff as well as the patient.

2.3 Adjustment for No-Shows, walk-ins, Urgent patients, Emergencies, and/or second consultations.

Wherever relevant, no-shows, walk-ins, urgent patients, and/or emergencies need to be planned for during the design of an online appointment system. In hospitals where the second consultation occurs frequently, such as in orthopedics, some allowance should be made for the additional demand imposed on doctors (Older, 1996). Even though many administrative mechanisms are found to be effective in reducing the likelihood of patients breaking their appointments (Such as reminders by mail or phone before appointment dates, fees for a failed appointment, etc.). it is not entirely possible to eliminate no-shows (BARRON, 1980). On the other hand, strong links are found between a tendency to attend without an appointment and lower social class and perception of urgency by Taylor (1984) and (Virji, 1990). These findings suggest that a clinic that denies access to walk-ins may further disadvantage these groups. Therefore, in general, a better approach is to anticipate no-shows and walk-ins, and adjust the appointment system in order to reduce their disruptive effects.

2.4 Types of appointment available at a modern hospital

2.4.1 Time slot scheduling.

In this case, your system allocates all available spaces to patients. After your patient selects the best time, the system encourages them to enter their information and then follows up with an email or confirmation. When using time-slot scheduling, your employees must keep an eye out for cancellations and, depending on how late they are, immediately replace it with someone else or risk having an empty block. Patients, on the other hand, feel more important because they know the time slot is all theirs.

2.4.2 Wave scheduling.

Wave scheduling is a patient processing technique that assigns numerous patients—four, five, or six—every half hour. The approach's goal is to have patients waiting in the queue at all times. Patients are not assigned precise timestamps and are handled on a first-come, first-served basis. This scheduling method, however, frequently favors ill/emergencies over regular appointments.

2.4.3 Stream scheduling.

Stream scheduling, also known as time-specific scheduling, involves assigning patients a certain amount of time. The method assures that a patient is available at all times. It provides you with a steady flow of patients and reduces their waiting time, much like a stream of water. The amount of time allotted to each patient is determined by the reason for the visit. You can allot 10 -15 minutes to established patients and 30 - 45 minutes to people with complex demands. The extra time is appropriate for cases that require extra care, such as physical exams or new patients.

2.4.4 Open booking.

Patients are provided a time range in an open booking system, say between 10:00 am and 11:00 pm, where they can honor the appointment using this outpatient appointment scheduling strategy. This method gives specialists enough time to attend to critically ill or emergencies. If the outpatient facility is not overcrowded, an open booking system helps manage clients' erratic flow by reducing average wait time, which reduces complaints.

2.4.5 Clustering scheduling.

The process of clustering patients with comparable health problems or diseases together is known as cluster scheduling. Each category or group is assigned a distinct time block or day to make management easier. Injections, operations, physical examinations, and prenatal clinics are examples of clusters. Clustering success is dependent on the availability of consultants, support personnel, equipment, and a short turnaround time for support activities. Cluster scheduling is an efficient way for busy medical centers to manage a large number of patients who require similar services. It can also cut typical wait times for regular patients—for example, in chronically ill patient clinics.

3 Analysis of the problem

3.1 Product Definitions

In this study, we will develop an online appointment system using a relational database to assist medical patients in reducing the strain of telephone appointments. In this chapter, we will find 'Persons' who will assist us during product testing so that we may build more user-friendly products.

3.2 Personas

Persona design is an ideal way of modeling and summarizing the characteristics of the target audience. Personas are created as virtual personas that help designers better understand users, their needs, experiences, behaviors and goals. Thus, these personas make it possible to simulate how real users will subsequently use the final product. Personas represent individual groups of customers targeted by a given product i.e. a typical user. First, basic information such as name, photo, short story, age, gender, income, job position, motivation and goals must be defined. This information will help identify the user group more easily. There are several types of personas that are interconnected, primary persona - for this persona the user interface is primarily designed, the primary persona will not be satisfied by a design aimed at any other persona, secondary persona - this persona is mostly satisfied with the interface of the primary persona, however has specific additional needs that can be met without compromising the product's ability to satisfy the primary persona, anti-persona - or also negative persona is a user for whom the product is not directly designed.

3.2.1 Primary persona (A)

Name: Adam posla

Age: 25

Sex: Man

Conical: going to hospital, Diabetic problem, Sugar.

A typical day: Adam posla gets up at 7 in the morning and goes to school. he goes to grammar school, but over time he found out that he doesn't enjoy school so much and would rather attend, for example, a clinic high school. When he comes home from school, he surfs the internet and reads all sorts of interesting things about quick appointment at hospital. In the evening, he usually contacts hospital by himself, but because he is responsible and knows that he is going to school again the next day, he cannot waste time at the hospital.

A brief history: Adam posla is currently studying at the grammar, but she is considering transferring to a clinic school, where he could learn more about the medical facilities. Or he could do some extra ordinary work in hospital industry

3.2.2 Secondary persona (B)

Name: Victoria
Age: 21 Sex: Woman
Conical: baking and cooking, on medical leave.

A typical day: Viktorie, as a mother on maternity leave, wakes up with the awakening of her son Vilém. In the morning, he goes for a walk with the stroller so that Vilém can get some fresh air and play a little on the playground. After lunch, she puts it away and has time for herself. She watches a series, calms down and cooks something for dinner for the whole family when her husband comes home from work. Viktorie used to visit bars a lot with her friends from high school, but with Vilém, her life priorities have changed. Once a month, however, she agrees with her husband and goes to a bar with her friends, just like in the old days.

A brief history: Viktorie studied confectionery and baked cakes to order. It's something that fulfills her and sometimes she finds time for a cake even on maternity leave.

3.2.3 Anti-persona (C)

Name: Ctirad Cibulka

Age: 67

Sex: Man

Conical: reading books, doing crosswords, growing vegetables, solving riddles from TV.

A typical day: Ctirad usually gets up, as he has been used to for years, in the morning with the dawn.He stretches a bit, makes breakfast and heads out to the garden to check and water his vegetables. Then he goes for a walk to the library, where he usually spends several hours. Since he doesn't have a smartphone, he tries to find the answer to a question from a television competition in various encyclopedias. He never has a drink in the evening after a meal like his wife, because he has been a long-time teetotaler. So he prefers lemon balm tea to help him fall asleep.

A brief history: Ctirad's father was an alcoholic and sometimes even beat him. That's why Ctirad has such an aversion to alcohol. He studied philosophy, which he later taught. He is now retired and doesn't do much philosophy anymore.

3.3 List of the goals

3.3.1 User goals.

1. On the landing page a selection of popular specialists, a list of top clinic locations, List of outstanding doctors.
2. Doctor and admin login.
3. Select the doctor.
   A. Select the appointment time.
      A. Fill out the form.
      B. Send notification to the user by email.
      C. Clinic address
      D. Doctor information.
4. View standard requirements.
   A. Home Page.
   B. For patients.

C. For doctors.

3.3.2 Doctor goals.

1. On the landing page an appointment schedule list.
   A. Check the detail of the patient.
   B. Send invoice.
2. Plan (release appointment slots).
   A. Add slots in the list so patients get on the home page.
3. Statistical data of the visitor in the month.
4. Logout.

3.3.3 Admin goals

1. Get the overview.
2. Users list (Doctors and counselors)
   A. Doctor's detail. (Add doctor, List of the doctors).
   B. Counselor detail. (Add counselor, List of the counselor).
3. Others.
   A. Clinics. (Add new clinics).
   B. Specialists. (Add new specialties of the doctor).
4. Appointment schedule. (Release the slot for all doctors by admin).
5. Plan. (Check the detail of the plan by date).
6. New patients. (Add detail of the new patient in the database, Confirm the appointment)
7. Posts. (Check the comment and add it to the system).

3.4 Use cases

The following module is focused on the design of screens. This means the placement of the individual on the screen, not the graphical form. Each screen contains its use case scenario and wireframe.

3.4.1 Landing page

When the website launched for the first time, the user expects a lot of things to know about the hospital therefore home page contain some significant information. Also, the home page contains the login page option for the staff of the hospital.

On the home page, the user expects the.

- Login option.
- Doctor details.
- Location information.
- Doctor specialties details.

Scenario

The system shows all fields in the horizontal list so the user can move from left to right and vice versa. Also contain the image of the doctor, specialties, etc. Once the user selects any doctors, locations, or specialties the user will get full details.

The system will display.

-Image of the doctors.
-Image of the specialties.
-Image of the different hospital locations.
-Get the login option from the dashboard option.

The system expects a user action.

- When the "dashboard" is pressed, it redirects the user to the login page.
- When you select to click the image of the specialist the redirect to detail.
- When you select the location form the list then it will redirect to a different location page.
- When you select the doctor then it will redirect to you detail page of the selected doctor.

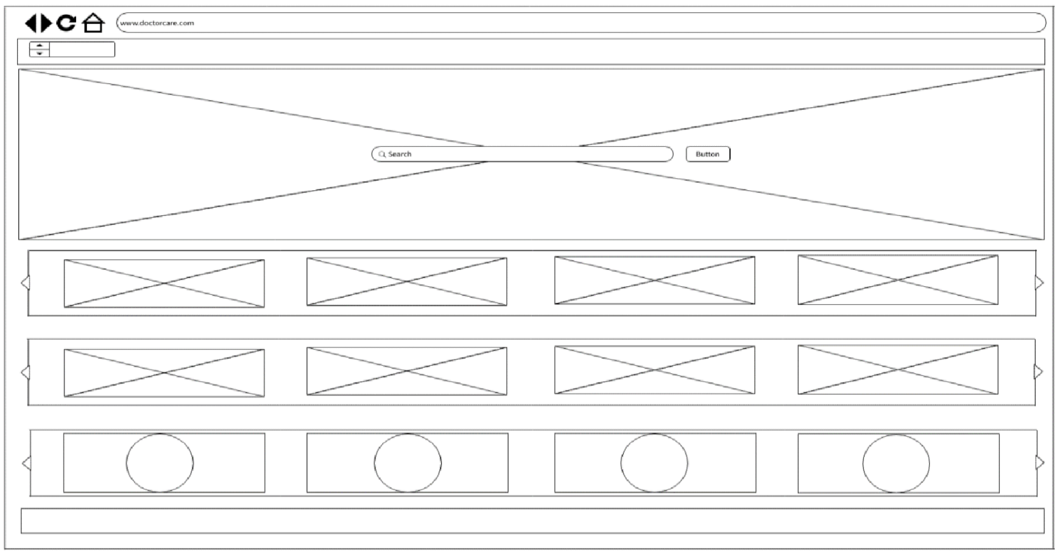


Figure 1: The user landing page.

3.4.2 Appointment option before time release

On the home page, users can select the doctor per their illness. It will redirect to the doctor detail page.

Use Case

The user expects the doctor's detail and time slot. The user expects.

- List of different options on the page.
- Image and detail of the doctor.
- Location of the hospital.
- List the dates then select the date.
- If the time slot is available then it will redirect to the patient detail form else inform the patient that there is no time slot.

Scenario

The system displays a field for a specific date, an image of the doctor with the name, and a link for a different page.

The system will display.

- List of time dates.

- Button for the time slot.
- Doctor detail such as study, article, and fees.

The system expects a user action.

- When selecting the data from the list it will show the different times according to time.
- When the user pressed any option at the navigation bar it will redirect to another page.
- Zoom in and out Use a map for the location.

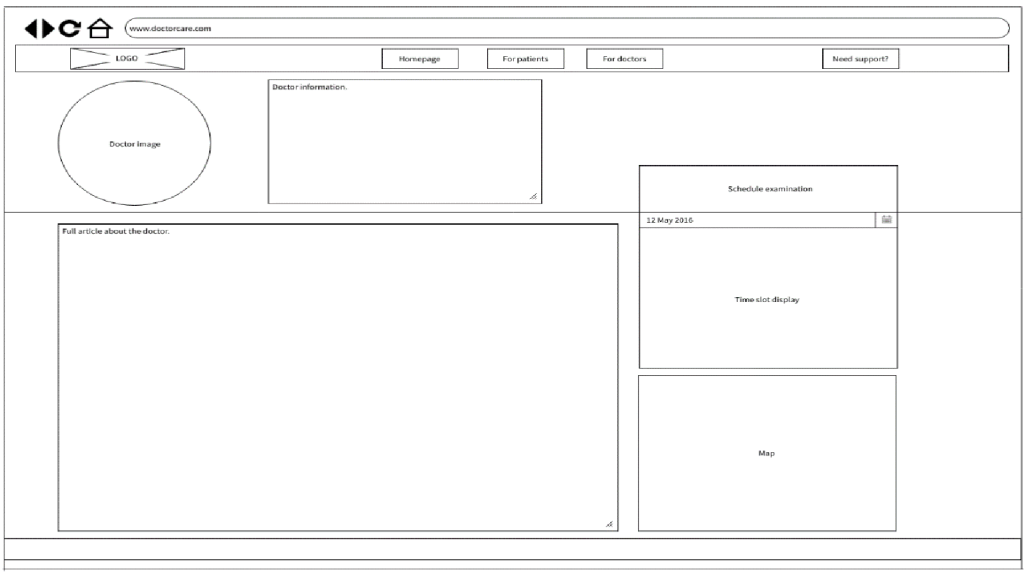


Figure 2: Appointment option before time release.

3.4.3 Login page.

When the staff member selects the login option from the dashboard list it will redirect to the login page. So he expects a screen with login options by entering a user email address and password.

Users expect two options.

- Login.
- Go back.

Scenario.

It will display a login screen, the screen contains an image of the hospital, a label for entering a user email, and a label for entering the password. There is a login button and back to homepage link.

The system will display.

- The “Login” button to login
- The label “your email address” and “your password”
- Text fields for entering user email and password.
- “Forgot password” link to reset a forgotten password.
- “Back to the homepage”.

The system expects a user action.

- When the “Login” button is pressed, it redirects the user to the main screen of the panel.
- When you press the “forgotten password” link, you will be redirected to the page for recovering your forgotten password.
- When you click on the text filed, the keyboard will appear.
- When you press “Back to homepage”, it will redirect you to home page UC= Landing page.

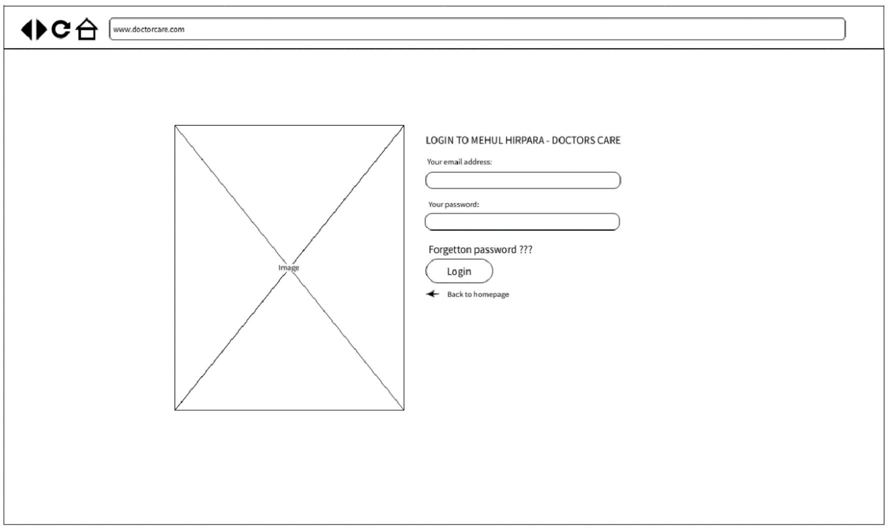


Figure 3. Login page

3.4.4 Doctor panel

On the home page of the doctor panel user get different information about the time slot. Such as the Appointment schedule option, check the plan, and statistical detail.

The user expects to be able to

- Date selection.
- Select the period.
- Display the selected range before the data.
- Check the overall completed visit.
- Check the list of the patient who will be visiting in the upcoming day.

Scenario.

On the screen, the system displays all the detail in the table, flow text with the button, and navigation bar. In the list of the patient, it shows the detail of the patient after clicking the info icon, and also able to edit.

The system will display.

- List of the upcoming patient.
- Button with the “detail”.
- Button with the “Send the information”.
- The navigation bar on the left side with different option
- Profile on option.
- Image logo.
- Small button for hiding and unhiding the navigation bar.

The system expects a use action.

- When pressed
    - Button to check the detail of the user
    - To send the appointment confirmation to the user.
    - Press the icon to check whether did user send the detail to the patient or not?
    - Pressed the profile and system will show logout option.
    - When navigation bar unhide.
- When selected the option from navigation link.

Select the dashboard for home.

Application schedule to display.

Plan where use can provide the appointment time.

Statistical data where you can show data for the month.

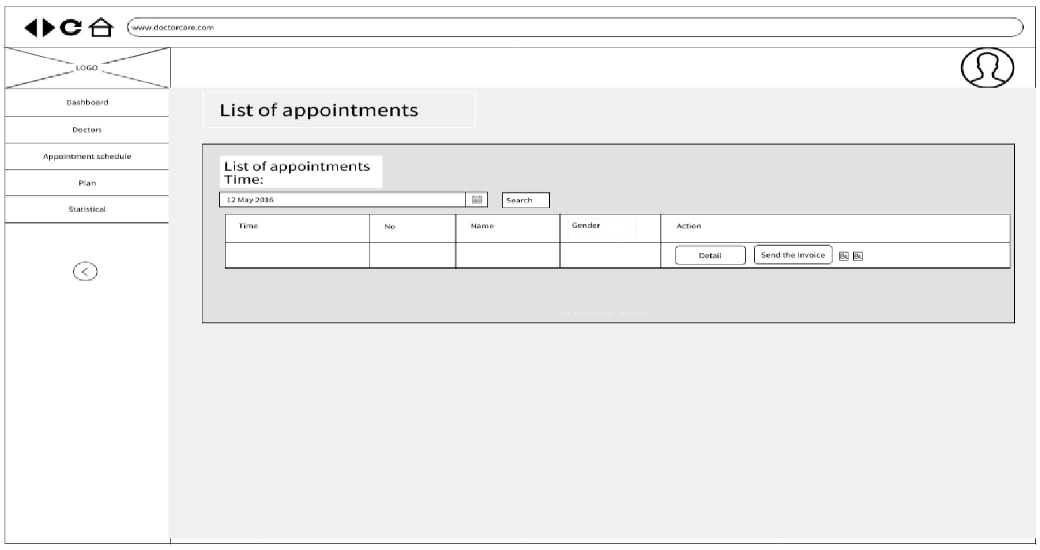


Figure 4. Doctor panel.

3.4.5 Doctor detail page after time slot released by the doctor

Use Case

On the home page, users can select the doctor per their illness. It will redirect to the doctor detail page where you will schedule your time with a prestigious doctor.

The user expects the doctor's detail and time slot.

- Get the time slot option on the screen.
- Image and detail of the doctor.
- Location of the hospital.
- List the dates then select the date.
- If the patient selects the available time slot is available, then it will redirect to the patient detail form else inform the patient that there is no time slot.
- Link for navigation in the web application.

Scenario

The system displays a field for a specific date, the image of the doctor with the name, and a link for the different pages. Once you select the date from the list then it will automatically show upcoming free appointment times for specific doctors. The web application will display a button to go booking form.

The system will display.

- List of time dates.
- Button for the time slot.
- Doctor detail such as study, article, fees, specialties, and so on.
- The navigation bar shows different page redirection options such as the homepage, for patients, for doctors, and so on.

The system expects a user action.

When selected.

- When selecting the data from the list it will show the different times according to time as per a release by doctors.
- Time button to open new reservation form for paction UC= Patient detail form.

When pressed.

- When the user pressed home it will redirect to another page such as the homepage.
- When use pressed for patient will redirect to patient page.

Zoom in and out

- Use map for the location of the hospital.

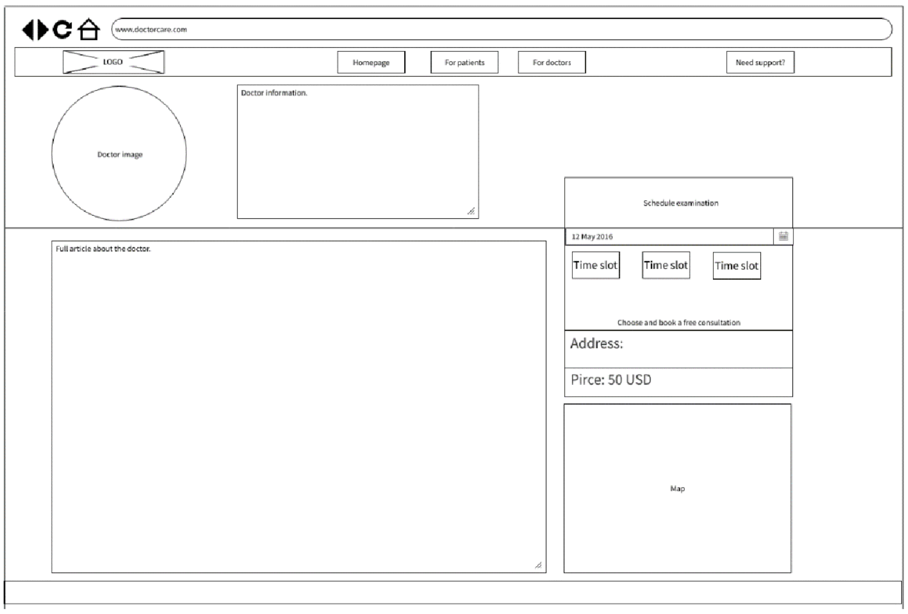


Figure 5. Doctor details page with time slot

3.4.6 Appointment booking form.

Use Case

The user expects the system to request the entry of basic information required for getting the registration, so it expects

- Entering personal data.
- Enter your email.
- Select gender.
- Enter your contact detail.
- Enter your age.
- Enter the address.
- Enter some other info applicable
- Enter additional detail about respiratory history note (If applicable)
- Select the place which you have already visited.
- Upload image document if applicable.
- Additional information if any.

Scenario

The system will display the form screen, where there are a label for filling in data and a button to create an appointment, furthermore, the screen contains a radio button for the selection of the gender and contains a button for upload.

The system will display.

- Labels
  - "Name"
  - "Phone number"
  - "Email address"
  - "Age"
  - "Address"
  - "Respiratory history"
  - "Additional information."

- Radio button

    - Select age

    - Have ever visited a place?

- Button

    - “Choose files” to select the file for uploading.

    - “Confirm medical appointment” button confirms the appointment.

    -“Cancel” button to cancel.

- Process

    - “Processing” hold the state until sending the confirmation email.

- Show the detail of the doctor with the image.

The system expects a user action.

- when you click in the text field then the keyboard will appear.

- When the radio button is clicked the radio button is checked.

- By pressing the “Choose files” button the use will be redirected to the system for file select.

- By pressing the “Confirm medical application” button the user can process of the application, once it done user will be redirect to home page.

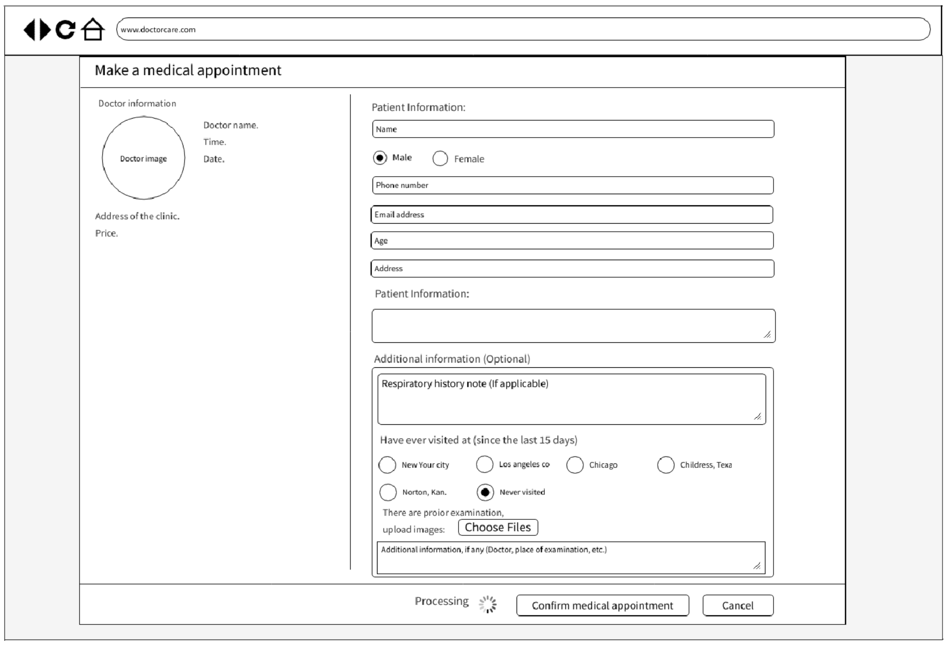


Figure 6. Appointment booking form.

3.4.7 Admin panel

Use Case

On the admin panel, the user expects significant information about the dashboard, such as the total number of patients, Doctors, Posts, etc. Admin also expects the possibility to add new detail in the system.

The user expects to be able to:

- Total number of patient,
- -navigation bar.
- profile.
- Total number doctors.
- -The total most time book
- -The most article supports.

Scenario

On the screen, the system displays a different option for the admin by navigation bar. Label contains the detail of patient, doctors, posts and so on. Admin can use check their profile.

The system will display:

- Label with the final number of the patient,

- Label with how many doctor is available in the hospital.

-Label with doctor most time book and article detail.

- Navigation bar with list of option.

The system expects a user action.

Select the list from the navigation bar

- When click users

  - Admin get the list of different option such as add new doctor from UC=”New doctor form”.

  - Admin get the detail of the supporter.

-When click on other option

  - Click on “Appointment schedule” redirect to the appointment page

  - Click on “Plan” redirect to the time slot page.

  - Click on “New patients” to redirect to the mange patient detail page UC=” New patient confirmation page”.

- When click on the profile the admin will get person detail.

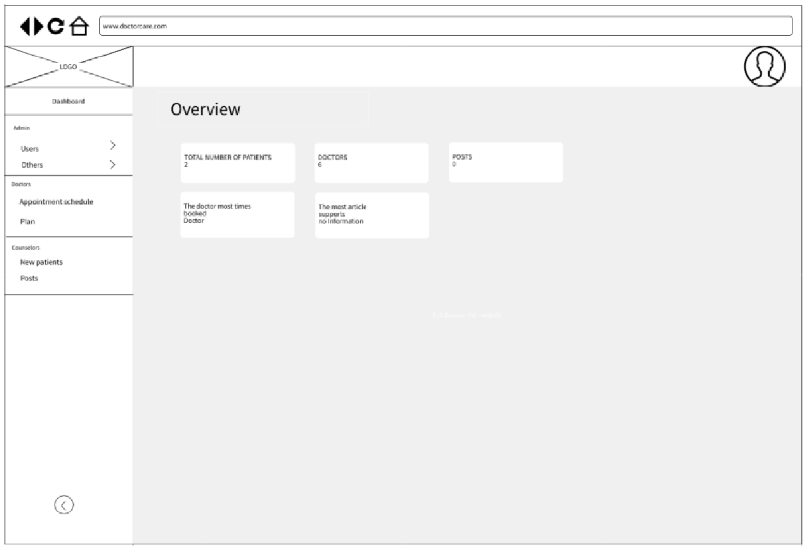


Figure 7. Admin panel

3.4.8 New patient confirmation page

Use Case

On the new patient confirmation page, User expect confirmation button and cancel button. Use navigation bar to move in different tab such as Add new patient, Need confirmation, Confirmed and cancelled.

The system will display:

- Different tab menu.
- Side navigation bar.
- Profile.
- List of the patient detail.
- "Confirm" and "Cancel" button.

Scenario

The system shows a navigation list on the left and side so admin can selection the different options. After selection of the New patients. Its show more detail about the patient.

Allows categories selection.

- There are four different options in the tab view so the admin can move from one tab to another.

The system expects a user action.

- If the user clicks on Add new patient tab, they will get the list of the patient who applied for the appointment.
    - Each appointment has two options such as Receive and Cancel.
        - If the user presses the confirm, the system asks, really wants to add a patient.
            - If yes, the system adds the patient to the confirmation tab.
            - If no, the system adds the patient in the canceled tab.
- If the admin clicks on the Need confirmation tab, they will get the list of a patient who requires an appointment confirmation email from the clinic.
    - Each appointment has two options Confirm and Cancel.
        - If the admin press the confirm, the system asks, really wants to send an email to the patient about the time schedule.
            - If yes, the system sends a confirmation email.
            - If no, they will not anything.

- If the admin clicks the Confirmed tab, will get the list of the confirmed patients.
    - Information highlighted.
- If the admin clicks Cancelled tab, will get the list of the Cancelled patients.
    - Information highlighted.

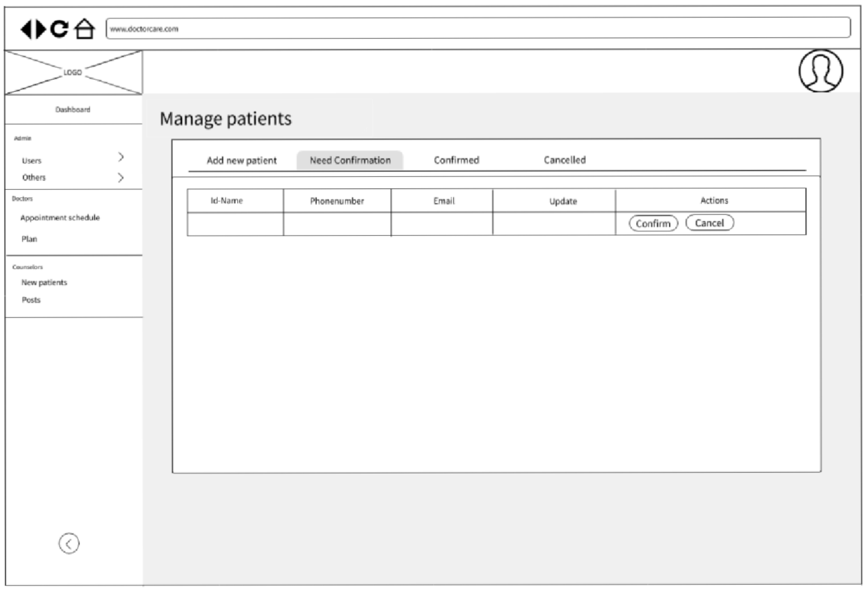


Figure 8. New patient confirmation page.

3.4.9 In the admin panel add new doctor

Use Case

The user expects the system to request entry of basic information about new doctor who will work in this hospital.

- Entering Full name.
- Enter your email.
- Enter password.
- Enter your contact detail.
- Enter the address.
- Select the clinic from the list.
- Select the specialist form the list.
- Additional information if any.

Scenario

The system will display the form screen, where there are label for filling in data and a button to store new doctor detail in the database, furthermore the screen contains list for selection of the clinic and specialist.

The system will display.

- Labels
    - “Full Name”
    - “Phone number”
    - “Email”
    - “Password”
    - “Address”

- List menu.
    - Choose clinic
    - Choose specialist.

- Button
    - “Create” to store the data in database.

- Navigation bar
    - Select form navigation menu.

The system expects a user action.

- when you click in the text filed then keyboard will appear.
- When the clinic list is clicked it will show the list different location.
- When the specialist list is clicked, it will show the list of different filed of the doctor.
- By pressing the “Create” button the user will be redirected to the home page of the admin panel UC=” Admin panel”.

Wireframe.

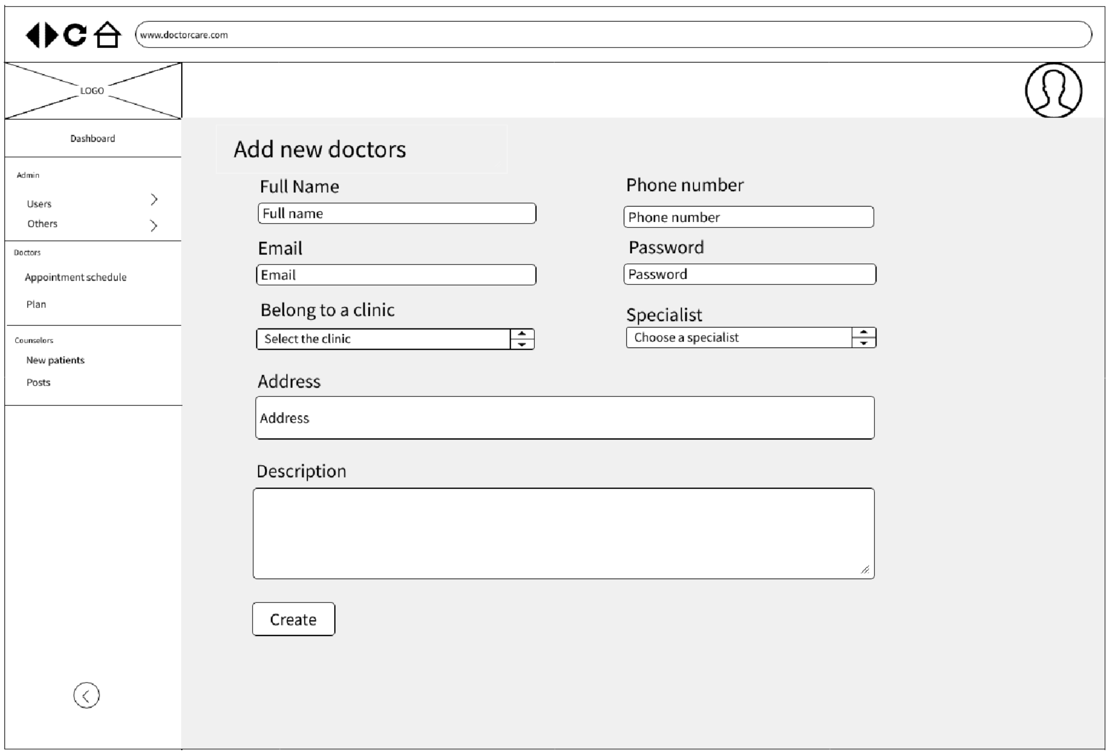


Figure 9. Add New Doctor

3.4.10 In the admin panel add clinic location.

Use case

After the selection of the doctor, patient will move to an appointment page where they get for patient navigation option.

The use expects.

- Explain step by step information, how to reserve your time with doctor for consultation**.**
- Get another options in the navigation bar.

Scenario

The system will show navigation menu on the topside to user can move easily.

The system will display.

- Quick move option on the top side such as homepage, for patients, for doctors need support.
- Processing step information for appointment booking.

The system expects a user action.

When pressed:

- If user press Homepage, then system will redirect to use at landing page UC=”Landing page”.
- If user press For doctors, then it will redirect to user at UC=”For doctor in navigation bar”.

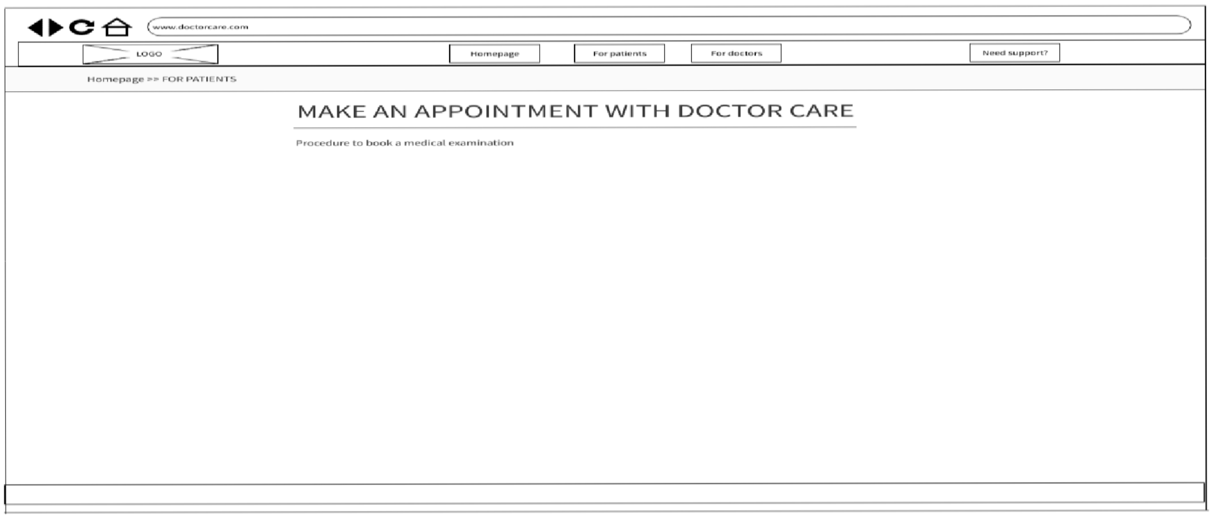


Figure 10. For patient in navigation bar.

3.4.11 For doctor on navigation bar

Use case

After the selection of the doctor, patient will move to an appointment page where they get for doctors navigation option.

The user expects.

- Information about the DoctorCare.
- Possibility to click the button for contact to administration.

Scenario

The system will show navigation menu on the topside to user therefore they can move easily.

The system will display.

- Quick move option on the top side such as homepage, for patients, for doctors need support.
- Information about the DoctorCare system.
- Button for opening new web page.

The system expects a user action.

When pressed:

- If user press Homepage, then system will redirect to use at landing page UC="Landing page".
- If user press For Patient, then it will redirect to user at UC="For patient in navigation bar".
- The button to switch other screen will display the contact form UC= "Contact cooperation form"

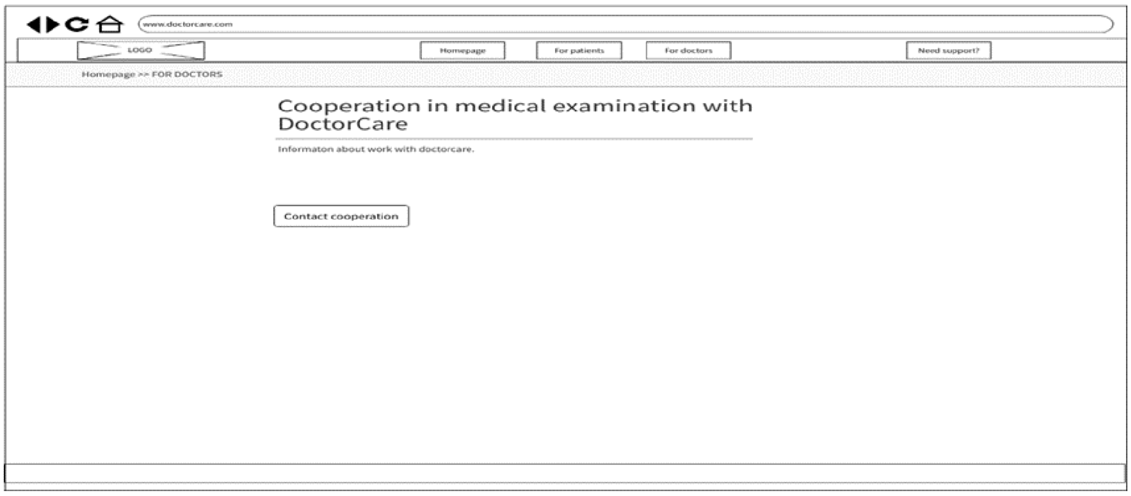


Figure 11. For doctor option.

3.4.12 For Contact cooperation form

Use case

User can share contact detail to the administrator by filling the contact cooperation from.

- Entering Contact.
- Enter your email.
- Enter your Phone number.
- Enter the name of the medical facility.
- Enter your address
- Enter content
- “Send information ” button

Scenario

The system will display the form screen, where there are label for filling in data and a button to send the data to administration for next step.

The system will display.

- Labels
    - “Contact”
    - “Email address”
    - “Phone number”
    - “Name of the medical facility”
    - “Address”
    - “Content”

- Button
    - “Send information” to Management team.

- Navigation bar

- Select form navigation menu.

The system expects a user action.

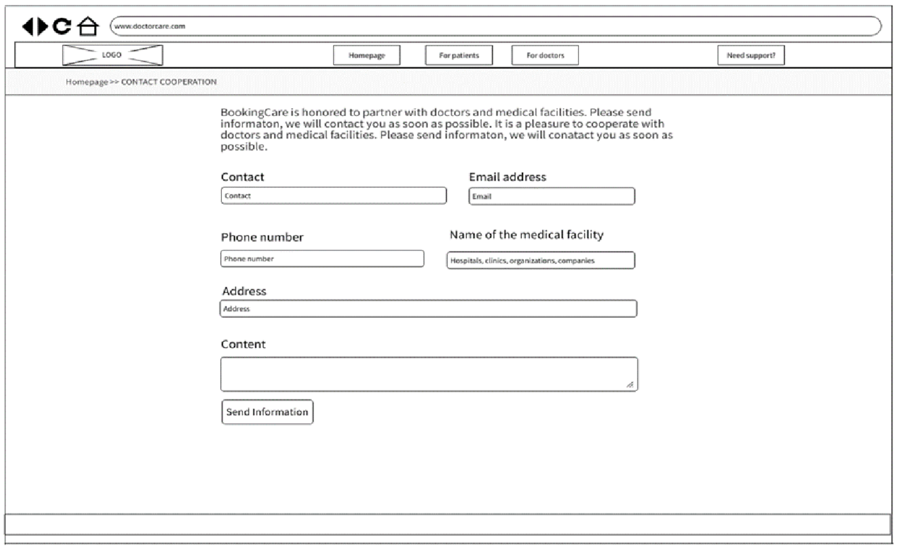
- when you click in the text filed then keyboard will appear.

- If click on Homepage, It will redirect you at home page UC=”Landing page”.

- If user press For Patient, then it will redirect to user at UC=”For patient in navigation bar”.

- If user press For Doctor, then it will redirect to user at UC=”For doctor in navigation bar”.

- Pressed “Send information” after it, System will redirect to user at For doctor page UC=” For doctor in navigation bar”


www.doctorcare.com

LOGO | Homepage | For patients | For doctors | Need support?

Homepage >> CONTACT COOPERATION

BookingCare is honored to partner with doctors and medical facilities. Please send informaton, we will contact you as soon as possible. It is a pleasure to cooperate with doctors and medical facilities. Please send informaton, we will conatact you as soon as possible.

Contact: Contact

Email address: Email

Phone number: Phone number

Name of the medical facility: Hospitals, clinics, organizations, companies

Address: Address

Content

Send Information

Figure 12 Contact Cooperation form.

3.4.13 Dynamic table and form creation

Use Case:

User can select the requirement of the form in this panel so base on it system will create the dynamic table in database and web page in admin which can reduce the burden of the front end developer. When will we get new table requirement then we will create it.

- Entering the label name.
- List show different option about form fields
- Negative sign.
- Positive sign.
- “Send” button

Scenario

System will display the dynamic table create form when admin require to generate the table

The system will display:

Label:

“Label Name”

Negative (-)

Positive (+)

List menu.

- Choose clinic

Button

-Send” information to system for dynamic data creating.

The system expects a user action.

- When you click in the text filed then keyboard will appear.
- When the choose option is clicked it will show the list of different field.
- “Send” button clicked, the crate the new table in the database, and add new form in the admin panel in DoctorCare application.

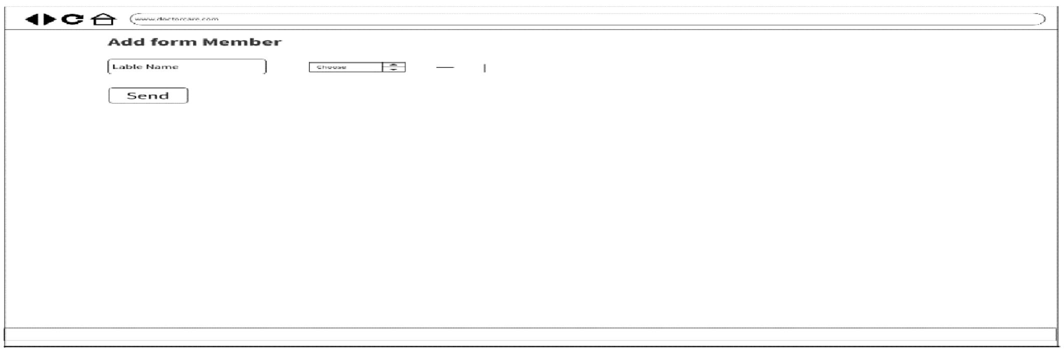


Figure 13 Dynamic table and form creation

3.5 Waterfall Model

These models are utilized in the development of software systems, which aids in the maintenance of new integrated software in the system. This paradigm aids in data integration as well as the implementation and design of software systems for massive data storage analyses. This also helps to preserve the data input and overall system needs (Brown, W. 2004). In this model, the outcome of the on phases acts as an input for the next phase, and it works in a sequential manner therefore first we need to complete one phase then only we go to the next phase. It also aids in the validation of data and the continuous process of analyzing appointment records using various technology instruments.

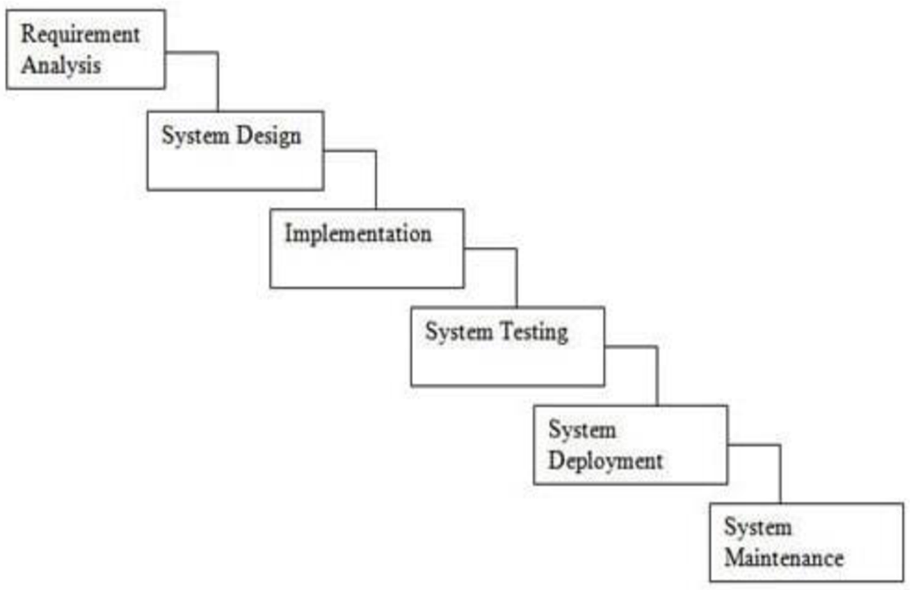


Figure 14. Waterfall Model

The stages of the waterfall model are as follows.

3.5.1 Requirement Analysis

In this stage, requirements gathered and document in a requirement specification document where user gives some specification and ideas to the team.

3.5.2 System Design

System design helps in specifying hardware and the system requirement and also defining overall system architecture. The architecture is a blueprint for structuring the solution to a project that reflects how it will be developed and linked together.

3.5.3 Implementation

Software development begins during this stage. It evolves in small programs known as units, which are short bits of code. Unite is tested for functionality, which is referred as unit testing.

3.5.4 Testing

In this stage, combine all of the units created during the implementation stage. After merging the modules, the overall product is validated to verify if it meets its objectives. Software defects and problems are reported as they are discovered.

3.5.5 Deployment

The product is ready to deliver after validation for both functional and non-functional equirements. It is installed on the user's system.

3.5.6 Maintenance

If any issues arise from the client environment, It will be resolved and the product will be improved during this phase.

3.5.7 Advantages:

- Base model
- Simple and easy
- Best fit for small project

3.5.8 Disadvantages:

- High risk
- No feedback.
- No parallelism.
- No feedback.
- Additional adjustments is more burden.

3.5.9 When to select waterfall model

When software components are fully understood, specified, clarified, and updated. Software design stability. Technology is well known and should not be impulsive. The software requirements are unambiguous. The availability of the requested resources. When the product is small.

3.5.10 Why did we select Waterfall model ?

The react javascript application can be developed using one of the software development life cycle methodologies, such as the Waterfall model. The application is small, all essential resources are available, the software requirements are straightforward and unambiguous, and all functional and non-functional requirements are properly understood and documented. Another argument is that software developers can concentrate on adding features that are required rather than those that are not. Because the Waterfall model describes the software life cycle, which begins with requirements collecting and concludes with development. It is a sequential strategy in which each step has a single goal and guarantees that all requirements are completed before moving on to the next step of the process. The waterfall approach aids in the definition of business goals, the prioritization of challenges, and the clear documentation of these goals as a collection of functional requirements, technical requirements, and acceptance criteria. Furthermore, all of the information provided by the client is simple to grasp and implement. The spiral model is employed when there is a risk parameter in the project development.

4 Technologies will be used

4.1 Object Oriented Programming Principles (OOP)

Object-Oriented Programming is a paradigm that includes several ideas such as inheritance, data binding, polymorphism, and so on. ***Simula*** is the first object-oriented programming language. A genuinely object-oriented programming language is one in which everything is represented as an object. The primary objective of object-oriented programming is to implement real-world concepts like objects, classes, abstraction, inheritance, polymorphism, and so on. The term "object" refers to a physical thing, such as a pen, chair, table, computer, watch, and so forth. Object-Oriented Programming (OOP) is an approach or paradigm for creating programs that use classes and objects. (Schildt, 2006). It simplifies software development and maintenance by introducing the following concepts. As per below, there is three major pillar for OOP principles.

4.1.1 Inheritance:

Inheritance is the process by which one object acquires the properties of another (Schildt, 2006). This is significant because it supports the concept of hierarchical classification. Inheritances expresses the "is-a" and/or "has-a" link between two entities. We can reuse the code from existing super classes in derived classes by using inheritance.

4.1.2 Encapsulation:

The process by which code and the data it manipulates are connected and kept separate from outside interference and unauthorized usage is known as encapsulation (Schildt, 2006). When code and data are protected by an encapsulation, other code that is declared outside the wrapper cannot access the code or data freely. Through a clear interface, access to the code and data inside the wrapper is carefully restricted**.**

4.1.2.1 Abstraction:

Abstraction is a significant component of object-oriented programming. Abstraction is the practice for hiding internal features while displaying functionality. In some cases, it is sufficient for one class to recognize the relationships between the members of another class in order to benefit from it. Using hierarchical classification is an effective technique to control abstraction. This enables us to stack complex systems' semantics and divide them into more manageable chunks.

4.1.2.2 Polymorphism:

The ability to use a single interface for a broad range of operations is a property of polymorphism (from Greek, meaning “many forms”). The actual nature of the problem dictates the specific course of action. There are two types of polymorphism such as static and dynamic. Method overloading is used to establish static polymorphism, whereas method overriding is used to generate dynamic polymorphism.

4.1.3 OOP’s concept in JAVASCRIPTS

Due to the popularity of JavaScript in web development, in this study we will examine some of the Object Oriented mechanisms that JavaScript supports to make the most of it (DHTMLX, 2020). JavaScript is a prototype-based programming language. A prototype-based programming language is a kind of object-oriented programming that uses functions as class constructors. Although JavaScript has the keyword class, it lacks a class declaration. It also makes use of cloning rather than inheritance. JavaScript is an object-oriented programming language that is ideal for creating object-oriented web applications. JavaScript allows you to create your own objects for your own apps. JavaScript can be used to construct independent files, programs, and it can even be embedded in HTML code (Frisbie, 2019). A language becomes object-oriented when specific characteristics or mechanisms, such as objects, classes, encapsulation, and inheritance, are present.

In Object-Oriented Programming, the attributes of an Object are referred to as Properties, and the actions are referred to as methods. An Object is a class instance. Objects are everywhere in JavaScript; practically every element, whether a function, array, or string, is an Object. A method in JavaScript is a property of an object whose value is a function. In JavaScript, objects can be created in several ways.

Object Oriented Programming Language There are no classes in JavaScript; only objects exist. To be more specific, JavaScript is a prototype-based Object Oriented Language, which implies that it does not contain classes; instead, it defines behaviors using a constructor function and then reuses them via the prototype.

The term encapsulation relates to the hiding of data or data sets. Abstraction is the representation of key elements while suppressing background detail. Most OOP languages have access modifiers to limit the scope of a variable, but there are no such access modifiers in JavaScript. However, there are certain ways to limit the scope of variables within the Class/Object.

It is a concept in which some properties and methods of one Object are utilized by another. Unlike most OOP languages, where classes inherit classes, JavaScript Objects inherit Objects, which means that specific aspects (properties and functions) of one object can be utilized by other Objects.

4.2 React

React is a front end component library that is open-source JavaScript library and only in responsible of the application's view layer. Initially developed by Facebook, it was eventually incorporated into services like WhatsApp and Instagram. ReactJS's main goal is to create User Interfaces (UI) that increase the speed of apps (Gackenheimer, 2005). It makes use of virtual DOM (JavaScript object), which enhances the app's performance. ReactJS can easily integrates with different frameworks and can be used on the client and server sides project. It makes use of component and data patterns to make larger apps more readable and maintainable. A ReactJS application is composed of various components, which are all in responsible for generating a discrete chunk of reusable HTML code. The core of every React application is its component. These components are organized within higher level components that specify the application structure. Take a form, for instance, which has numerous components including input fields, labels, and buttons. Each form component may be expressed as a React component, which we then combine to create the form component itself (Xing, Huang and Lai, 2019). The form's structure and the pieces that go inside of it would be specified by the form's components.

Drawback: -

1. It is confusing for human and machine, especially at binding and this keyword.
2. Lifecycle methods, logic spread over different lifecycle methods
3. Hard to test compared to functional components**.**

4.3 Embedded JavaScript:

Embedded JavaScript (EJS) is a simplistic templating language that enables the creation of HTML markup using standard JavaScript. No attachment to any one organization 's strategy. There is no need to recreate iteration and control-flow. It's just plain JavaScript. In this research we use embedded JavaScript so we can take advantages of HTML tags to design our front end view so user can easily interact with the system. It's also help to developer at developing time and debug time. Major future of an Embedded JavaScript is as per below.

Features of EJS.

1. Fast compilation and rendering.
2. Simple template tages<% %>.
3. Customer delimiters[??] instead of <%%>
4. Easy for both server JS and browser.
5. Static caching of templates.

4.4 Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system (By oracle team). A database is mainly controlled by a database management system (DBMS). The term "database system," which is frequently abbreviated to "database," refers to the combination of the data, the DBMS, and the applications that are connected to it. In pesent time, Data is modeled in rows and columns in a series of tables to make processing and data querying efficient after that data can be easily accessed, managed, modified, updated, controlled, and organized. For writing and querying data, most databases adopt structured query language (SQL) which is very popular language for database operation (Codd, 1983). Every organization use data stored in database to make informed business decisions and define the future direction base on available data. Some of the ways organizations use database include the following.

1. Increase business processes.
2. Sore personal data.
3. Keep track records of customers,
4. Secure personal health information.

4.5 Relational database

The database is used to maintain structured data. All organizations depend on the Relational Database Management System (RDBMS) to maintain their day to day working data. Same amount of potential information present in the database also and the database users suffer to retrieve the required information on their own. Structured data are different from unstructured data and it needs additional information about the structure, i.e., data about the data. Structured data contains the collection of related information that is organized in a table format, where column heading represents the field names and the rows below the column heading line contains the actual data that are arranged under the corresponding fields (Adya *et al.*, 2007). The storage of structured data in digital form migrated from flat files to DBMS to RDBMS. DBMS eliminated some problems that were present in flat files. RDBMS includes necessary constructs to make the DBMS robust and reliable

4.5.1 ACID property of RDBMS

The ACID features of the RDBMS are critical to its success. The ACID qualities ensure that RDBMS transactions are executed correctly. A transaction is a single logical data operation. For example, when we transfer money from account A1 to account A2, the transaction must complete completely or not at all, which means that the given amount must be debited from account A1 and credited to account A2. Otherwise, the payment may be deducted from account A1 and not credited to account A2. If the transaction is not completed completely, the database state will become inconsistent. To avoid this issue, the RDBMS provides certain ACID qualities to keep the database consistent.

Atomicity:

The transaction has been completed fully or not at all. There is no third option, which means that transactions do not take place in stages. Each transaction is treated as a separate unit and is either completed or not done at all.

Consistency:

The state of the database remains valid before and after the transaction is executed.

Isolation:

The concurrent execution of transactions yields the same consequences as if the operations were done in isolation. Changes that occur in one transaction will not be visible to any other transaction until the change in that transaction is written to memory or committed. This characteristic assures that running transactions concurrently results in a state that is comparable to running them sequentially in some order.

Durability:

The committed transactions survive any failures. This attribute ensures that once the transaction has completed execution, the database updates and modifications are stored in and written to disk, and that they persist even if the system fails. These modifications are now permanent and are kept in nonvolatile memory. As a result, the transaction's effects are never lost.

4.6 JAVA

Java is a fully object-oriented programming language that focuses on "Write once, run anywhere" (WORA) for application developers, where produced java code can run on any platform via the Java Virtual Machine (JVM) without regard for the underlying computer architecture. Java has grown in popularity, particularly for client-server communications and a wide range of web applications. It was invented in 1995 by James Gosling at Sun Microsystems(Arnold, Gosling and Holmes, 2000). The grammar of Java is inherited by C and C++, along with the object-oriented principles of C++. It offers far more programming enhancements than C and C++. Object-orientation, automated storage management, multithreading capabilities, and exception handling are all built into the Java programming language. It also makes programming easier and less error-prone. These efficiency and safety features are especially appealing for large program development since they allow for modularity and reuse while also reducing various forms of errors.

The major design aim of Java is portability, which means that Java programs must execute similarly on any combination of platforms with tolerable runtime support. Java's magic code, known as byte code, is responsible for making it feasible. Instead of turning the Java code into architecture-specific machine code, the Java compiler turns the source code into a transitory version known as Java byte-code after compilation. Byte-code is similar to machine code, but it can only operate on a unique platform known as the Java virtual machine (JVM). JVMs are intended to be deployed on any platform having a Java Runtime Environment already installed on the end user system in order to execute Java programs. Java provides standard libraries for its programs to use in order to support its features.

4.6.1 Reasons, why did not select JAVA language.

1. **Java is not faster and has poor performance:**

   Java uses a lot of memory and is much slower than native languages like C or C++. It is also slower than other languages such as C and C++ since each code must be interpreted to machine level code.

   This slow performance is caused by the JVM's additional level of compilation and abstraction. Furthermore, because the garbage collector consumes additional CPU time, it can contribute to poor Java performance.

2. **JAVA does not provide backup facilities:**

   Java is primarily concerned with storage and does not prioritize data backup. This is a huge drawback, and it is losing user interest and ratings as a result.

3. **JAVA required more memory spaces:**

When compared to other languages such as C and C++, Java uses a large or sizable amount of memory space. The memory efficiency and system performance may be negatively impacted during garbage collection execution.

4. Verbose and complex codes:

Java code is verbose, which means it contains numerous words and many long and complex sentences that are difficult to read and understand. This may reduce the readability of the code. Java focuses on being more manageable, but it must settle with unnecessarily complex codes and lengthy explanations for each issue.

4.7 Hibernate (ORM)

Object/Relational Mapping (ORM) is a method of changing information because the object-oriented model is linked to the personal database model. Object-Oriented Programming (OOPs) stays established occurring units, although the relational database management system (RDBMS) disreputable ordered families and fields towards stock information. Separated charting sheet, designed for a creator, criticizes the difficulties of the boilerplate cipher. ORM hides the functionality of an old conservative Java Catalogue Connectivity (JDBC) software design that was previously dependent on the saved files. A conservative ORM request promotes a minor object-oriented crossover point known as the Data Access Object (DAO). ORM completed the java perseverance API (JPA) crossover point, which continued the perseverance operation. JPA is a java submission software design crossing point that achieves data exchange between java substances and social files. JPA applications improve the transportability and extension of the cryptography by divorcing the JPA standards from the original API design. The following two parts contrast created planned JPA and JPA operation, which would shatter a creator's interpretation towards sanctifying the method while emerging an API.

4.7.1 Reasons, why did not select Hibernate(ORM).

1. simple and rely on a single database that never changes

2. Data must be entered into database tables; no more SQL queries are required.

3. There are no objects that are assigned to multiple tables.

4. When dealing with complicated data, mapping from objects to tables and vice versa affects performance and increases conversion time.

5. Hibernate does not allow some type of queries which are supported by JDBC. For example, it does not allow to insert multiple objects (persistent data) to same table using single query. Developer has to write separate query to insert each object.

6. Hibernate does not handle complex joins, union and intersection capabilities in databases, hence Jdbc or native SQL queries are required.

4.8 MVC Architecture.

Today's software industry faces rapid growth due to the innovations in the areas of web applications and services. Because of this, businesses or customers established across the globe can transact with each other. In this global scenario, the web application designers have to put forth their efforts in rendering economical, secured, scalable, maintainable and reusable solutions to the customers, worldwide. Even though there exist plenty of web applications that satisfy the parameters discussed above, only a few are developed keeping the future changes in mind. Design Patterns (DPs) (Gamma *et al.*, 1995) are extensions of object object-oriented technologies [8] that make it possible to design reusable architectures. Pattern based web applications are classical approaches that aim at thinking of an application development as a building block game. This approach deals with identifying the pieces that make up the program together with their interrelationships.

Web applications have come across several improvements over a few decades. There is plenty of literature available on effective pattern usage for web applications. While the positive side of pattern usage is explored intensively, the negative sides are not. In other words, it is essential to classify the side effects based on improper applications of patterns together with suitable reengineering models. These models with experimental results will be highly useful in writing pattern-based web applications. One of the structural patterns, the Model/View/Controller (MVC) (Frank Buschmann, 2001) segments an application into the manageable components Model, View and Controller Analysis and Impact of Using

Model/View/Controller (MVC) Pattern for Web Applications with clearer role separations. Due to the problem with partitioning issues pertaining to web applications, the programmers hesitate to use the MVC pattern. In order to help the designers, it is essential to pick an MVC based web application and to make structural changes so that it works better in the case of web applications also.

4.8.1 Introduction to MVC

The MVC architecture pattern divides the UI problem into three parts as shown in below

- **Model**: - Data model to contain the computational parts of the program.
- **View**: - View to render the output.
- **Controller**: - Controller to interact between the user and the View.

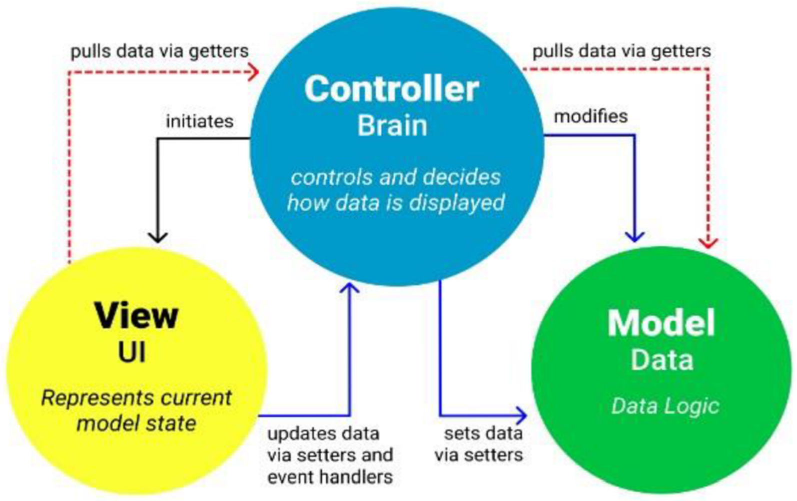
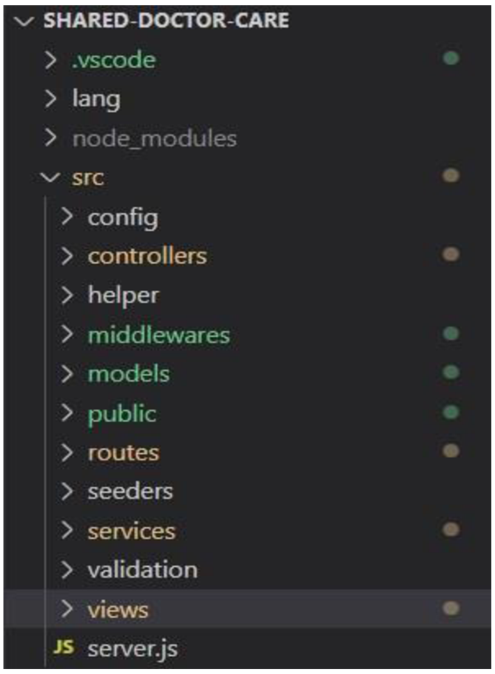


Figure 15. MVC architecture.

MVC structure implementation in our project.



```
∨ SHARED-DOCTOR-CARE
  > .vscode
  > lang
  > node_modules
  ∨ src
    > config
    > controllers
    > helper
    > middlewares
    > models
    > public
    > routes
    > seeders
    > services
    > validation
    > views
    JS server.js
```

Figure 16 - Created from project(Author)

MVC is a fundamental paradigm for interactive applications. This initially appeared in the Smalltalk-80 implementation. MVC is inherited from OO Programming with the integration of GUI and interactive program execution. Simula is a programming language that is designed to support discrete event simulation, has given birth for MVC. The great success of MVC as a popularized OO concept has Events Method Invocations become highly influential on modern programming methodology.

The main theme behind MVC is the logical separation of presentation from behavior and data structure in an interactive multi windowed programming environment. The idea of dividing the system into Model, View and Controller components provides a flexible way to deal with each one of them independently. Because the MVC design paradigm's main purpose is to separate BL from presentation and it is flexible enough to be incorporated into any type of application, it is a prime candidate for adaptation in web-based applications and services. In a programming environment, the View manages the rendering of output; the Controller interprets device input events and triggers appropriate changes in Model and View; the Model manipulates the behavior and data structure. The Model responds to requests for its state information from the View and instructions to change states from the Controller. The View and the Controller pair are associated with each other via an instance variable pointer. Both will have "Model" variables pointing to the Model object. The Model is relatively loose coupled with the other two components and has a fair communication skill. The invocation may happen in any one of the two ways; either the Controller may request the change of the Model and take the responsibility to inform the View of this change or the Model may be updated due to interaction from a third party component that is outside the View-Controller pair.

4.8.2 MVC Vs Other Aps

Layer Vs MVC

The Layer pattern (Buschmann *et al.*, 1996) helps to divide an application into groups of sub-tasks arranging them in a hierarchy from the lowest level of abstraction to the highest level. This increases coupling between the layers. However, the level of independency introduced in the components of MVC will not be available with layers. The major Analysis and Impact of Using Model/View/Controller (MVC) Pattern for Web Applications 37 area of concern for layers will be defining protocols for architectures. (Eg. Layers introduced by the ISO-OSI Reference Model).

Pipeline Vs MVC

A pipeline consists of a chain of process elements arranged so that the output of each element is the input to the next. This increases cohesion between the modules involved. In other words, in order to achieve a task, several processes have to be combined to obtain the end result. On the contrary, MVC emphasizes role separation.

Blackboard Vs MVC

The Blackboard pattern is applied to systems where there are no deterministic strategies available. It involves collaborative efforts of various specialized subsystems in order to achieve the goal. Here, several intermediate goal states are reached by individual subsystems. The collective goal states in a properly arranged sequence may contribute the goal of the system (Chang, Chen and Lai, 1999). MVC is applicable for problems where deterministic solutions are known.

4.9 Spring Boot

A significant open source Java/J2EE application development platform for faster application development is the Spring Framework (Schildt, 2006). With 30% of all Java usage, it is the most widely used Java framework. The Spring Framework's characteristics make it possible to create complex enterprise applications as well as basic Web apps quickly. Main concepts that the Spring framework depends are:

- Dependency Injection (DI)
- Inversion of Control (IoC)
- Aspect-Oriented Programming (AOP).
- Java persistence API (JPA)

IoC refers to a basic idea where control of flow is inverted; rather than the programmer managing a program's flow, external sources (frameworks, services, and other components) take control of it. As described in, DI is a pattern in the form of IoC, which is currently a major idea in Spring and the quickly expanding Google Guide. The foundation of AOP is the idea that code structure and modularity may be improved. JPA is in charge of how to query across the objects and how the object state is mapped to database fields. There are about 20 modules in the Spring framework. Core Container, Data Access/Integration, WEB, AOP, Instrumentation, Messaging, Test are a few of the most crucial components. Web modules are crucial for the creation of Web applications. The Web, WebSockets, Portlet, and Servlet modules are included. The two most popular notions today are defined in the servlet module. Integrations with RestFull WebSerive are the second (REST WS). Spring MVC offers comprehensive treatment on Spring MVC and Spring Web Flow, two potent web frameworks that allow for the creation of multi-layered applications. "Open for expansion, closed for modification" is the status quo. The foundation of Spring MVC is the DispacherServlet, which uses the @RequestMapping annotations to route requests to the relevant @Controllers. A project called Spring Web Flow works in conjunction with Spring MVC to provide reusable web controller modules that contain complex page navigational rules. REST WS is an architecture that enables communication between online computer systems. It is currently one of the methods most frequently used to provide data to different sorts of data consumers.

The Spring framework supports REST WS development. The Java class that will represent REST WS must be annotated with the @RestController annotation. Methods that provide functionality must be annotated with a distinct @RequestMapping annotation. The entire set of HTTP status codes, headers, and body could be specified using the Response Entity class as a method result type. It includes several constructors for carrying the information sent in the HTTP Response.

SPRING BOOT Spring Boot is intended to make Spring application development easier. Among the most significant are:

- Automatic configuration - configure apps as standard Spring applications.
- Starter dependencies - automatically integrate required dependencies.
- The command-line-interpreter - allows application control via console.
- The actuator - displays information about events in the application.

Spring Boot provides a radically faster and more widely accessible getting started experience for all Spring development, as well as a variety of non-functional features that are common to large classes of projects (e.g. embedded servers, security, metrics, health checks, externalized configuration); and with no code generation or XML configur ation requirements. Furthermore, application testing is simplified. It is possible to set up and download the start project with updated dependencies at the URL address: https://start.spring.io/. It is possible to choose between Maven and Gradle projects, and all that remains is to determine which dependencies should be included. WEB module requirement is required for constructing REST WS applications.

5 Implementation

5.1 Introduction

The practical section will go over app conceptualization and deployment. It will also go through some significant sections of code that are required to meet the objectives of this thesis. In this chapter, I will show practical tasks implemented in the programming language and try to show the dynamic form reflected in a relational database.

5.2 Application concept

During the covid-19 circumstance, the thought of implementing this notion came to me. People were afraid to visit the hospital during Covid-19 () since most hospitals had already designed static appointment systems, so people were confused about making an appointment and patients were unsure whether the doctor will be there at the hospital today or not. Furthermore, the old online application doctor could not provide a time frame because the majority of the application relied on a static database. Based on this circumstance, I attempted to create a new online appointment system with a dynamic relational database in which a doctor may provide a dynamic time slot for an appointment so that a patient can schedule an appointment online, and the database changes quickly. When you book a slot, it will be automatically removed from the dynamic web page. Although there are numerous matured applications available in the market for time and task management, I am using the notion here to delve deeper into real-time technology rather than a static concept.

5.3 Requirement, Project creation & Project structure.

It is vital to gain some understanding of the environment by having correct setups for actual coding. The program I intend to implement is cross-platform. It will work in any browser. When the implementation goal is to only create shared code for the front end in the React platform and the backend relational database in MySql. To begin developing for multiple browsers, a Visual Studio application project must be created.

5.4 Screenshot of front end Application

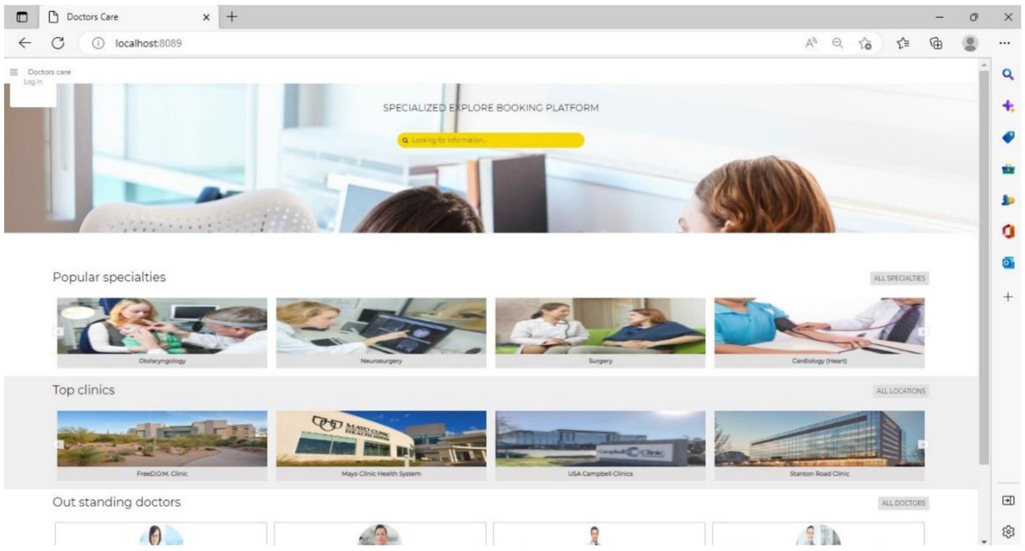
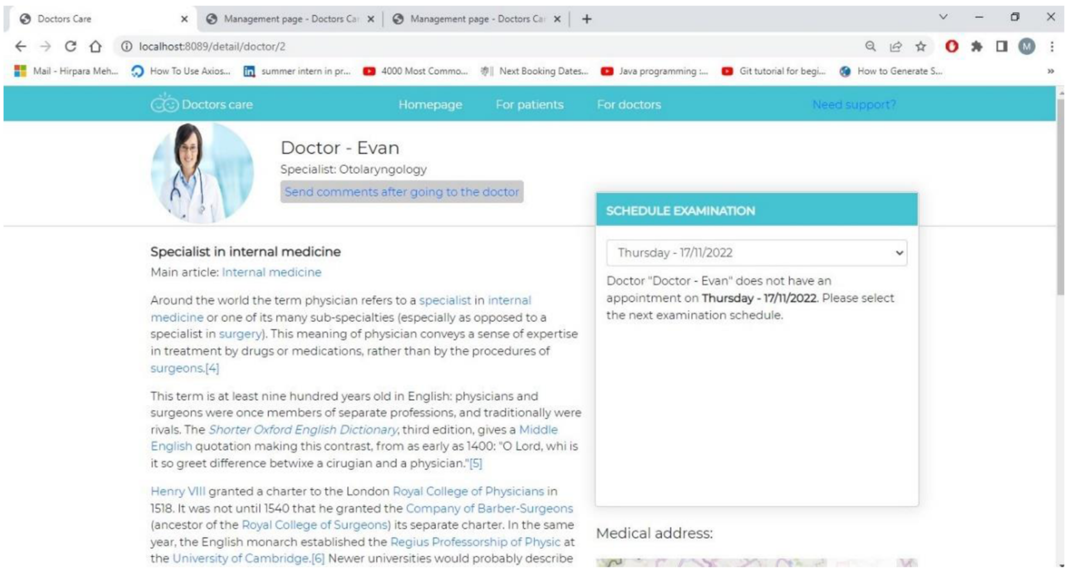


Figure 17. Landing page.

After landing on the online appointment website, there are different option for the patient such popular specialties provide by hospital such as surgery, Neurosurgery and many more so it will help to the patient. In addition, doctorcare website all provide location information which will be help to find nearest location of the hospital. There is also outstanding doctors list with booked appointment therefore base on their busy schedule patient can decide their doctor and get the treatment soon. Once the patient click on the doctor to it will open new web page where patient can get full history of the doctor Figure 18.



Doctor - Evan

Specialist: Otolaryngology

Specialist in internal medicine

Main article: Internal medicine

Around the world the term physician refers to a specialist in internal medicine or one of its many sub-specialties (especially as opposed to a specialist in surgery). This meaning of physician conveys a sense of expertise in treatment by drugs or medications, rather than by the procedures of surgeons.[4]

This term is at least nine hundred years old in English: physicians and surgeons were once members of separate professions, and traditionally were rivals. The *Shorter Oxford English Dictionary*, third edition, gives a Middle English quotation making this contrast, from as early as 1400: "O Lord, whi is it so greet difference betwixe a cirugian and a physician."[5]

Henry VIII granted a charter to the London Royal College of Physicians in 1518. It was not until 1540 that he granted the Company of Barber-Surgeons (ancestor of the Royal College of Surgeons) its separate charter. In the same year, the English monarch established the Regius Professorship of Physic at the University of Cambridge.[6] Newer universities would probably describe

Doctor "Doctor - Evan" does not have an appointment on **Thursday - 17/11/2022**. Please select the next examination schedule.

Medical address:

Figure 18. Doctor detail

In doctor detail web page, where patient will full detail article of the doctor so patient can understand about mastery of the doctor, in addition, there is schedule examination option with the help of the list patient will be able to select upcoming date for examination. If there is not available time, then it will display a message "Done not have an appointment on specific date please select the next examination schedule" (See in Figure 18)

Now doctor come in the picture, on the landing page there is option for login and, this option helps to login for doctor and admin only. Once the doctor clicks on login page then it will redirect to login web page which look as per below (Figure 20. login page)

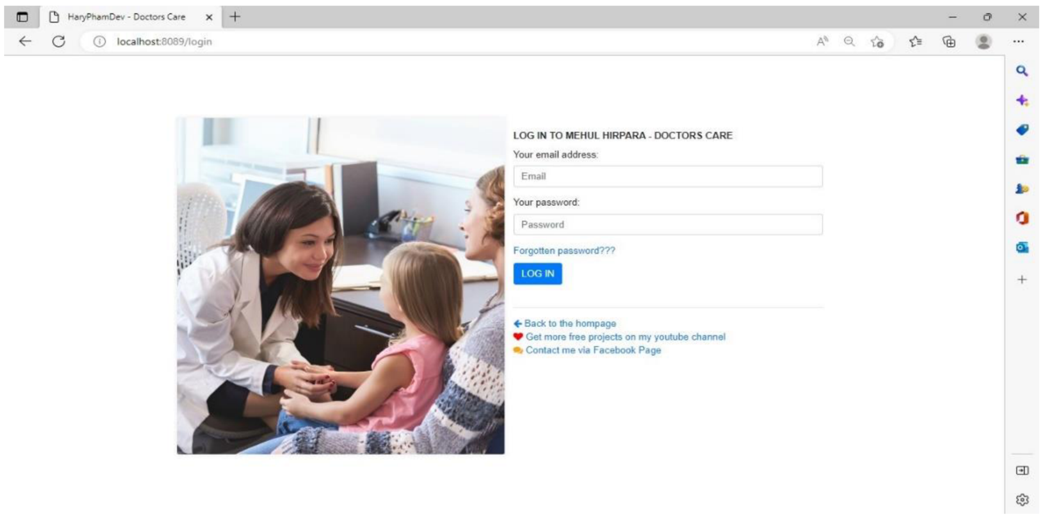

HaryPhamDev - Doctors Care
localhost:8089/login
LOG IN TO MEHUL HIRPARA - DOCTORS CARE
Your email address:
Email
Your password:
Password
Forgotten password???
LOG IN
Back to the hompage
Get more free projects on my youtube channel
Contact me via Facebook Page

Figure 19. login page.

On the login page, doctor gets login with the help of email address there is also other option if doctor forget their email address and password so they can recover by selecting forgotten password. When doctor press log in button then system generate the session (Figure 20 Configuration Session code (Author)) data for which help to connect with the database.

```
let configSession = (app) => {
   app.use(session({
      key: "express.sid",
      secret: "secret",
      store: session store,
      resave: true,
      saveUninitialized: false,
      cookie: { httpOnly: false, secure: false, maxAge : (24 * 60 * 60 * 1000)} // 1day
   }))
```

Figure 20. Configuration Session code (Author)

Based on the previous method doctor will login into the online appointment.

Now doctor land to the doctor panel where doctor get different option such as appointment schedule, plan and statistical. In the plan option, doctor will be creating new time slot list and release. However, he will get the list of the patient, who will be visit in upcoming day in appointment schedule option. (See Figure 22 Doctor panel)

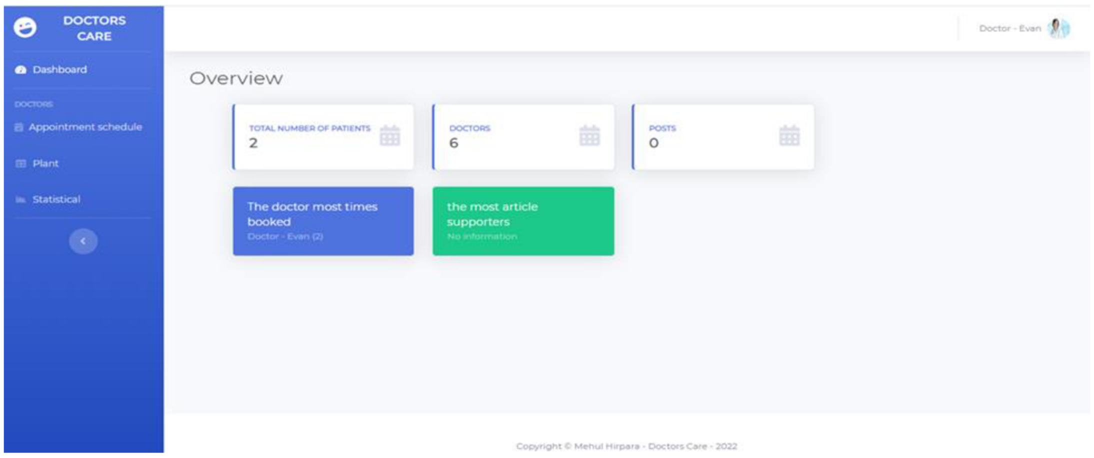
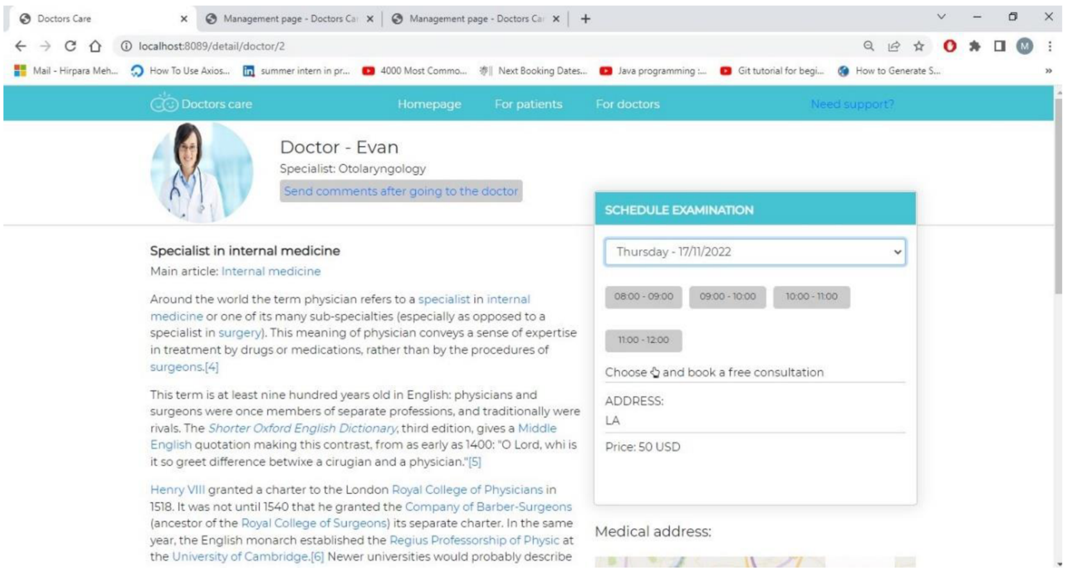


Figure 21. Doctor panel

In the doctor panel doctor will also get the total number of booked examination and number of doctor on the dashboard page.

After release the availability from the doctor end, it will automatically reflect on the website which will generate dynamic content and display whenever it allows from the doctor end. Now doctor detail page (Figure 18 Doctor detail) will look with appointment time slots. (Figure 22 Doctor detail with time slot).



Doctor - Evan

Specialist: Otolaryngology

Specialist in internal medicine

Main article: Internal medicine

Around the world the term physician refers to a specialist in internal medicine or one of its many sub-specialties (especially as opposed to a specialist in surgery). This meaning of physician conveys a sense of expertise in treatment by drugs or medications, rather than by the procedures of surgeons.[4]

This term is at least nine hundred years old in English: physicians and surgeons were once members of separate professions, and traditionally were rivals. The *Shorter Oxford English Dictionary*, third edition, gives a Middle English quotation making this contrast, from as early as 1400: "O Lord, whi is it so greet difference betwixe a cirugian and a physician."[5]

Henry VIII granted a charter to the London Royal College of Physicians in 1518. It was not until 1540 that he granted the Company of Barber-Surgeons (ancestor of the Royal College of Surgeons) its separate charter. In the same year, the English monarch established the Regius Professorship of Physic at the University of Cambridge.[6] Newer universities would probably describe

Choose and book a free consultation

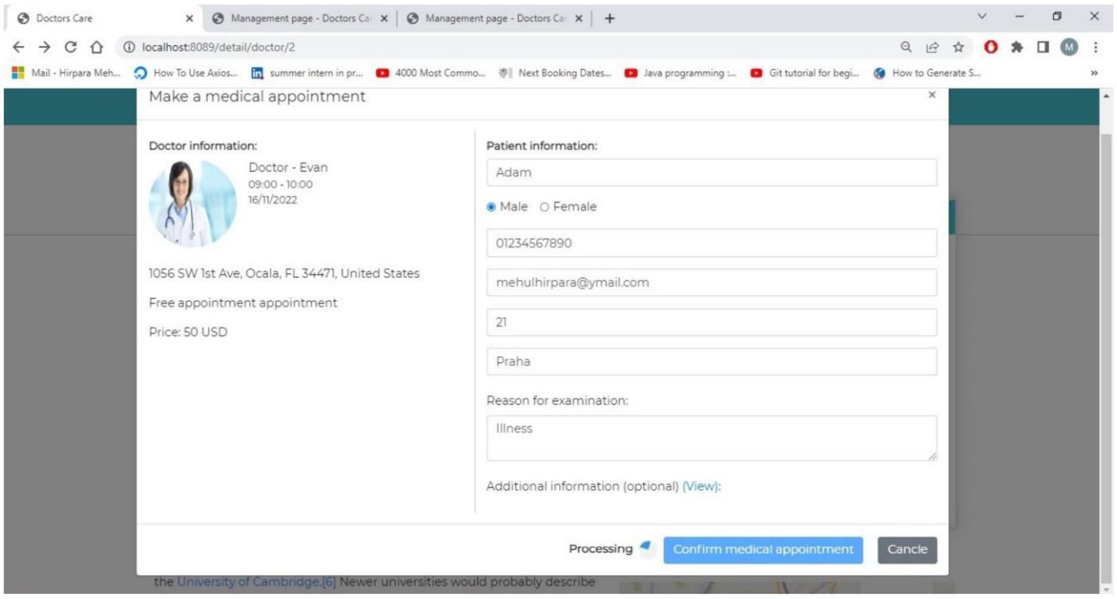
ADDRESS:
LA

Price: 50 USD

Medical address:

Figure 22. Doctor detail with time slot

Now as per selection of the date, user get different time slot (1 hour) which help to hospital to reduce rush of the patient in a same time. Patient can book their time as per there availability so they can save time and money. After click to the time slot system will open new form page where system will ask significant information about the patient.


Make a medical appointment

Doctor information:

Doctor - Evan
09:00 - 10:00
16/11/2022

1056 SW 1st Ave, Ocala, FL 34471, United States

Free appointment appointment

Price: 50 USD

Patient information:

Adam

Male [✓] Female

01234567890

mehulhirpara@ymail.com

21

Praha

Reason for examination:

Illness

Additional information (optional) (View):

Figure 23. Patient information form.

In the patient information form, System will ask about Name, gender, address, contact detail, provide the option for upload extra document, and so on. Once the user completed then they are able to send the detail to the hospital by clicking the “Confirm medical appointment” button. Beside the button, we will be get processing icon while system will work on data. Once the everything complete successfully then user get notification by email where we mention like “Pending for approval”. Now admin role will be entering in the picture for further approval. Admin is able to user login page with the email address and password to enter the admin panel which is look like as per below.

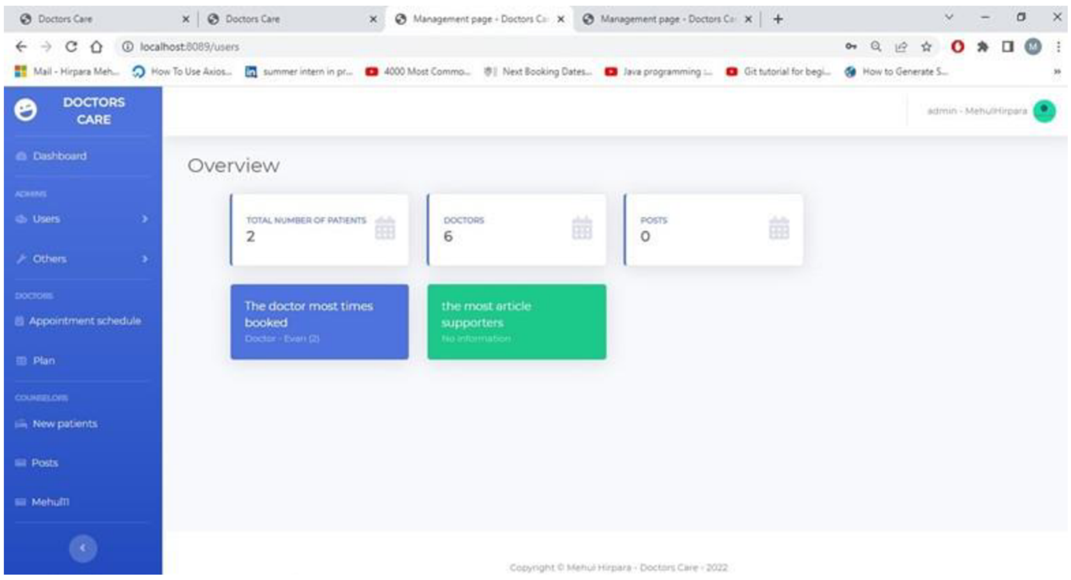
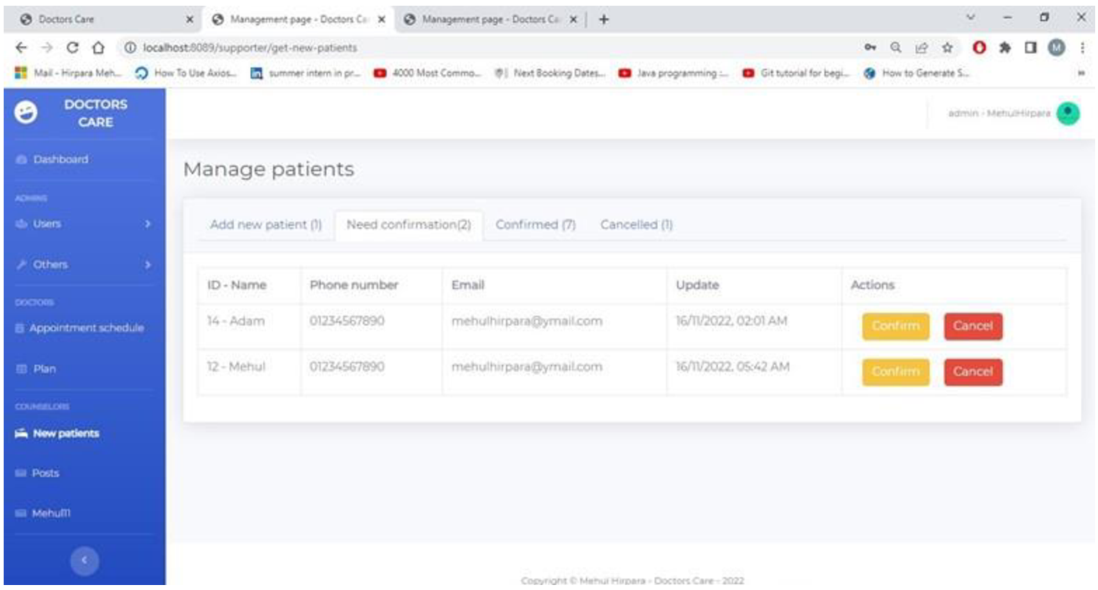


Figure 24. Admin panel

On the admin panel, Admin get different option which will help to manage the hospital work very smoothly. Inside the navigation bar, there are three main option such as “Admins”, “Doctors” and “Counselors”. In the admins, admin can perform different operation on doctors and counselor data such as show, delete, add and modify. In second section, Admin will be able to release the examination schedule slot for the doctor. The last option is Counselors, in this admin will manage the new patient detail as well as posts because using both option admin will share significant information on the website by post and confirmation detail by email to the user.

Now we will dive deeper into “New patients” which is under the counselor’s option. Which the help of it user will take the final decision for approval of the appointment. (Figure 25 Manage patient appointment)



Manage patients

Add new patient (1) | Need confirmation(2) | Confirmed (7) | Cancelled (1)

ID - Name	Phone number	Email	Update	Actions
14 - Adam	01234567890	mehulhirpara@ymail.com	16/11/2022, 02:01 AM	Confirm Cancel
12 - Mehul	01234567890	mehulhirpara@ymail.com	16/11/2022, 05:42 AM	Confirm Cancel

Copyright © Mehul Hirpara - Doctors Care - 2022

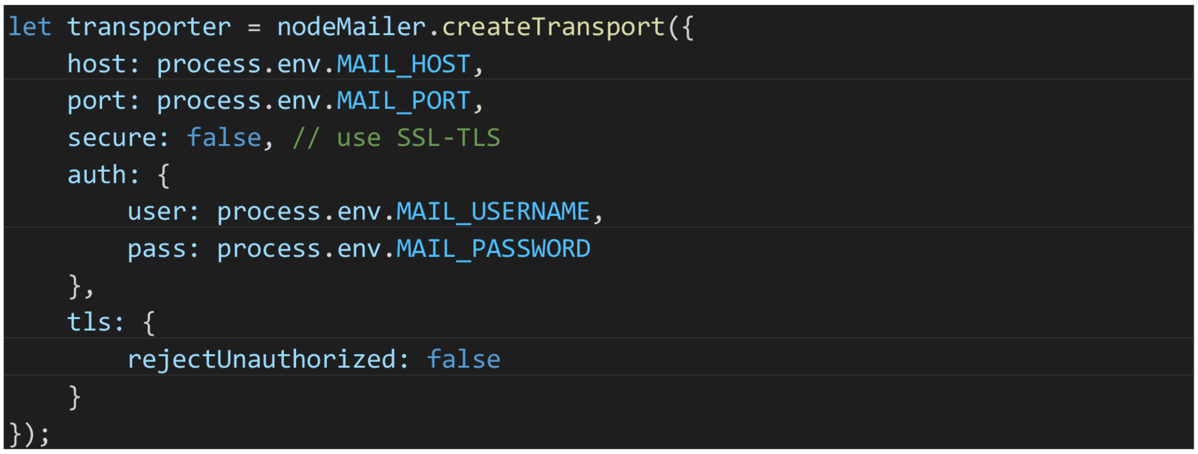
Figure 25. Manage patient appointment

In the mange web page, Admin gets four different options in the navigation bar. The first tab is "Add new patient", In this tab, the user gets the list of the patient who requested for appointment in this hospital so the admin can detail the patient and be able to add it to the database. The second tab is "Need confirmation", In this tab admin is can send appointment confirmation by email so the patient gets final approval for examination. The third tab, show the list of confirmed patient also admin can check the detail of the patient through this tab. The last tab is "Cancelled", if the admin rejects any appointment of the patient then it will appear in the canceled tab where the admin has to mention the reason for cancellation.

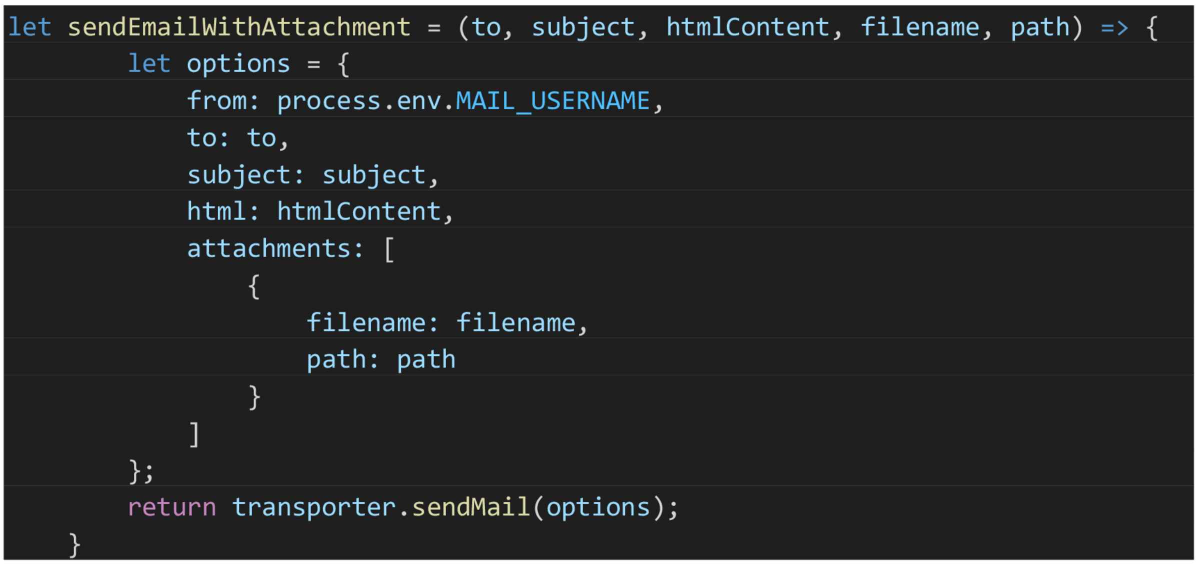
Once the approval and confirmation have been done by the admin then patient detail will automatically display under the "Appointment schedule" option. (Figure 25 Appointment list from the doctor profile.)

5.5 Mailer Module

Here we will discuss different important methods and configuration which is required for web development project. In the below code we use transporter for sending confirmation by email. Furthermore, this function collect the "MAIL_USERNAME" and "MAIL_PASSWORD" value form the .env file where we provide all the configuration setting about it.



```
let transporter = nodeMailer.createTransport({
    host: process.env.MAIL_HOST,
    port: process.env.MAIL_PORT,
    secure: false, // use SSL-TLS
    auth: {
        user: process.env.MAIL_USERNAME,
        pass: process.env.MAIL_PASSWORD
    },
    tls: {
        rejectUnauthorized: false
    }
});
```



```
let sendEmailWithAttachment = (to, subject, htmlContent, filename, path) => {
        let options = {
            from: process.env.MAIL_USERNAME,
            to: to,
            subject: subject,
            html: htmlContent,
            attachments: [
                {
                    filename: filename,
                    path: path
                }
            ]
        };
        return transporter.sendMail(options);
    }
```

Figure 26 Source code :- Mailer module

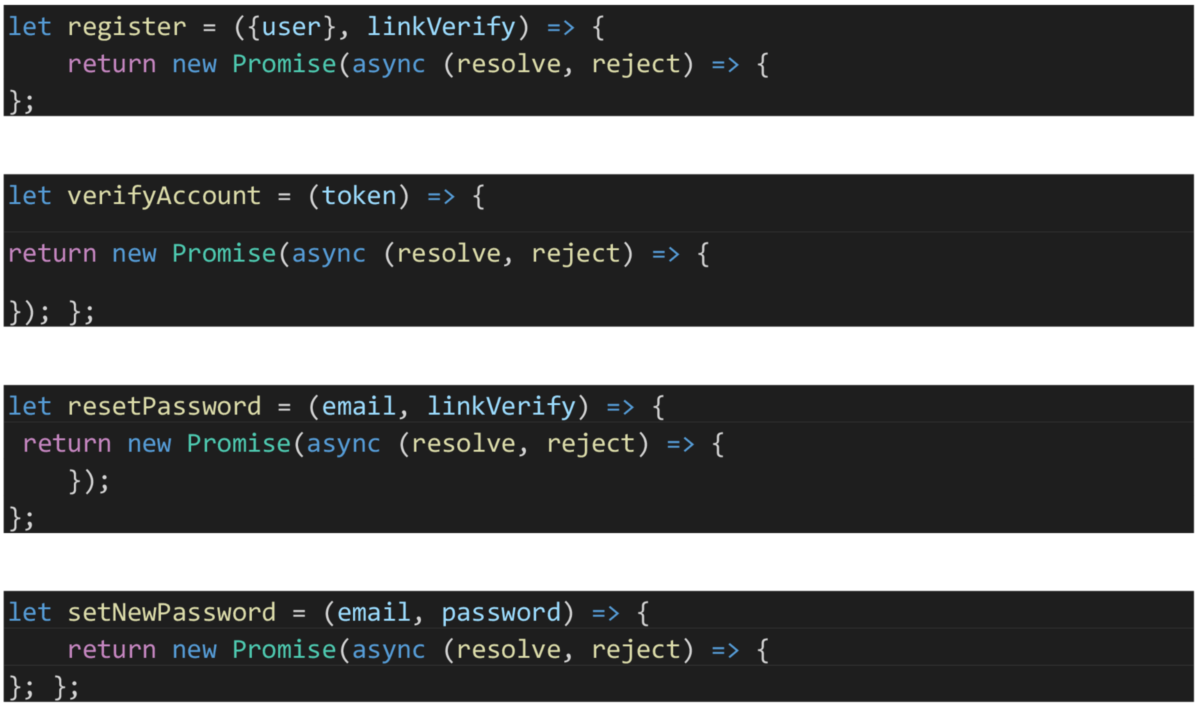
In this application we can also send an email by attachment file. After the treatment doctor is responsible to send all the significant document and invoice by email so user get and use in future references.

Test Mailer

This module we tested in UI specification, every user got the all the email which has been sent by the system such as appointment in process mail, confirmation email, Send invoice by email etc. In addition, also tested the text field of the email input while persona was entering the detail and we got the positive result.

Authenticating services

This JavaScript file is responsible for the check user is valid or not, and we write different method for that such as register, verifyAccount, resetPassword and setNewPassword. All those method will work on token base which will generate base on exiting user detail in database. If token should be match then we will allow user to enter in the system.



```
let register = ({user}, linkVerify) => {
    return new Promise(async (resolve, reject) => {
};
```

```
let verifyAccount = (token) => {

return new Promise(async (resolve, reject) => {

}); };
```

```
let resetPassword = (email, linkVerify) => {
 return new Promise(async (resolve, reject) => {
    });
};
```

```
let setNewPassword = (email, password) => {
    return new Promise(async (resolve, reject) => {
}; };
```

Figure 27 Source code - Authentication check

List of main interfaces that we use in java

Create_Model_File_Int

In this interface we define two different methods with same name such as getModelDetail with different parameters first one is CreateTable_Detail and other is Form_Builder. Base on it system will collect the data form the user and generate the dynamic table and form detail.

```
public interface Create_Model_File_Int

{
	public void getModelDetail(List<CreateTable_Detail>
theCreateTableDetail);

	public void getModelDetails(List<Form_Builder>
theCreateTableDetails);
}
```

Form_Create_Blogic

In this interface we will define the business logic like first we will get the data from the user and base on selected file we will create the dynamic form which will reflect in DoctorCare admin panel.

```
public interface Form_Creater_BLogic
{
	public String getFormCreateDetail(List<Form_Builder>
theFormCreaterBLogic);

	public void addWebHref(String fileName) throws
IOException;
  }
```

Create_Table_DAO

This interface will create the run time table in the data which can help to front end developer. Front end create does not want to know how to table is working and does not require any access for database so this functionality will help and reduce to burden of more coding when user get new table requirement.

```
public interface Form_Creater_BLogic
{
	public String getFormCreateDetail(List<Form_Builder>
theFormCreaterBLogic);

  }
```

5.6 Database Model

A database model is a type of data model that determines the logical arrangement of objects in a particular database. It therefore determines how the data in the database will be stored, organized and what their mutual relations will be. A relational database model stores data in tables that have relationships between them that define their relationships. The model below is defined for a MySQL database.

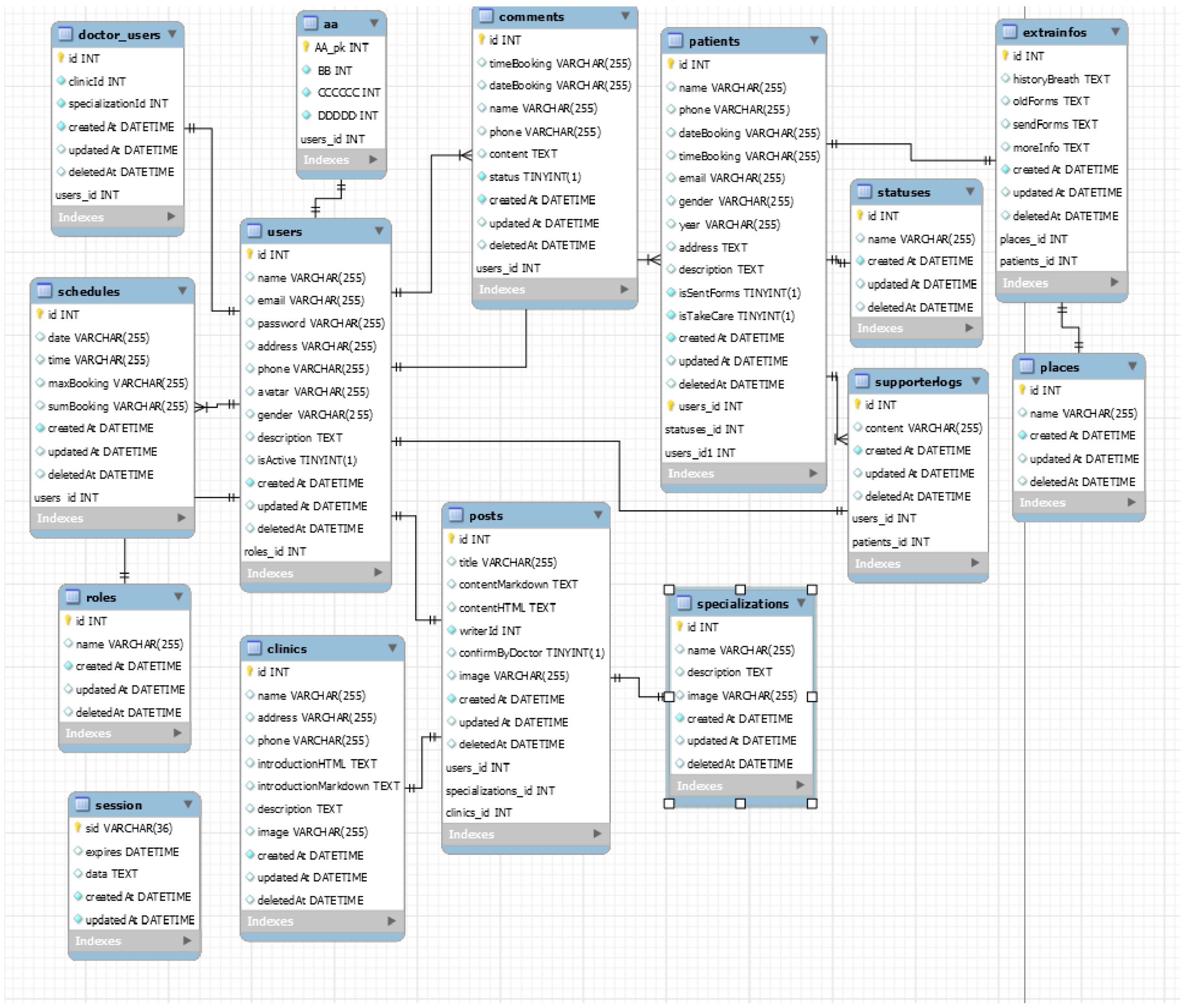


Figure 28 Database model

Users:

In this table, we will record the user's general information as well as private information that can aid in identification, such as email address, password, address, personal contact information, and so on.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each specific product
name	VARCHAR(255)			User name
email	VARCHAR(255)			User email address
password	VARCHAR(255)			User security code
address	VARCHAR(255)			User address detail
phone	VARCHAR(255)			Personal contact detail
avatar	VARCHAR(255)			Information
gender	VARCHAR(255)			Gender
description	TEXT			Shor information
isActive	TINYINT(1)			Check use is active or not
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
roles_id	INT		YES	Reference to roles table

Table 1 Users database table

Doctor Users:

This table is used to store ClinicId and SpecializationId details. We may access other table data based on the previous id. Because this table is related to the user table, it contains the foreign key.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each specific product
clinicId	INT			Hold the clinic id
specializationId	INT			Specialization id
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
User_id	INT		YES	Reference to roles table

Table 2. Doctor_users database table

Schedules:

We will collect information about the date and time in the schedule table so that we can identify some significant information for test maximum booking, and the total number of bookings.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each specific product
date	VARCHAR(255)			Collect the data
time	VARCHAR(255)			Collect the time

maxBooking	VARCHAR(255)			Check max slot per day
sumBooking	VARCHAR(255)			Total number of patient.
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
Users_id	INT		YES	Reference to users table

Table 3. Schedules database table

aa:

This table we design during runtime for testing dynamic data base implementation at run time.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
AA_pk	INT	YES		Unique identifier of each specific product
BB	INT			Name filled
CCCCCC	INT			Surname field
DDDDD	INT			Address field
user_id	INT		YES	Reference to roles table

Table 4. aa database table

Roles:

Because the person's role in the organization is the most significant, we will store the user role in this table and provide access depending on it. It is linked directly to the user table.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each specific
name	VARCHAR(255)			Name of the role
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time

Table 5. Roles database table

Session:

We will save session-related information in the session table, which will be used for password verification when a user enters the system.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
sid	VARCHAR(36)	YES		Unique identifier of each specific
expires	DATETIME			Time expriry
date	TEXT			Date of the use
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time

Table 6. Session database table

Clinics:

In this table we will store clinic related information such as location of the clinic, images of the clinic, history of the clinic and information in html text.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each specific
name	VARCHAR(255)			Clinic name
address	VARCHAR(255)			clinic address
phone	VARCHAR(255)			Contact detail of clinic
introductionHTML	VARCHAR(255)			Bacic information of clinic
intorductionMarkdown	VARCHAR(255)			Information mark down
description	VARCHAR(255)			Description of clinic
image	VARCHAR(255)			Photos of clinic
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
roles_id	INT		YES	Reference to roles table

Table 7. Clinics database table

Comments:

We gave feedback options to end users in this application so that when they finished their visit, they could provide their personal reviews, which we would keep in this table.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**

id	INT	YES		Unique identifier of each commetns
timeBooking	VARCHAR(255)			Time of the appointment book
dateBooking	VARCHAR(255)			Date of the appointment book
name	VARCHAR(255)			Commenters name
phone	VARCHAR(255)			Contact detail
content	TEXT			Sentence of the comments
status	TINYINT(1)			It is active or not.
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
user_id	INT		YES	Reference to user table

Table 8. Comments database table

Posts:

The post table will contain the doctor's details and will be displayed on the website when the user selects the doctor, which will display all of the doctor's articles on the web page.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of each
title	VARCHAR(255)			User name
contentMarkdown	VARCHAR(255)			User email address
contentHTML	VARCHAR(255)			User security code
writerId	VARCHAR(255)			User address detail
confirmByDocotr	VARCHAR(255)			Personal contact detail
Image	VARCHAR(255)			Information
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time

user_id	INT		YES	Reference to user table
Specializations_id	INT		YES	Reference to specializations table
clinic_Id	INT		YES	Reference to clinic table

Table 9. Post database table

Patient:

This table is critical to the project since it contains all of the patient's information, including name, contact information, address, reason for appointment, and a document list if uploaded during the reservation.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of patients
name	VARCHAR(255)			Patient name
phone	VARCHAR(255)			Patient contact detail
dateBooking	VARCHAR(255)			Appointment date
timeBooking	VARCHAR(255)			Time of the appointment in particular date
email	VARCHAR(255)			Email address for further communication
gender	VARCHAR(255)			Gender of the patient
year	VARCHAR(255)			How old is patient
address	TEXT			Permanent residential address
description	TEXT			Information if patient want to share with doctor
isSentForms	TINYINT(1)		YES	Check user for send or not
isTakeCare	TINYINT(1)		YES	Check and show to admin
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time

deletedAt	DATETIME			Transaction deleted date and time
users_id	INT		YES	Reference to user table
statuses_id	INT		YES	Reference to statuses table

Table 10. Patient database table

Specializations:

Specialization tables carry the doctor's major fields such as orthopedics, dentistry, and so on. This table is one-to-one connected to the posts table.

Defined properties

Column Name	Data type	PK	FK	Description
id	INT	YES		Unique identifier of specialization filed
name	VARCHAR(255)			Filed name
description	TEXT			Description of the specialization
image	VARCHAR(255)			Image of specialization field
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time

Table 11 specialization database table

Statuses:

In this table, we will keep the user's status, such as whether or not the user is active, in which case the user can access the system, otherwise not. This table communicates with two other tables, patients and users, using one-to-many and one-to-one relationships.

Defined properties

Column Name	Data type	PK	FK	Description

id	INT	YES		Unique identifier of statuses
name	VARCHAR(255)			Such as success, failed, pending, new, done
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time

Table 12 Statuses database table

Supporterlogs:

We saved various pop-up messages that will display either the patient or management performing some task on the website, such as New appointment has been received, the appointment has been successfully booked, and so on.

Defined properties

Column Name	**Data type**	**PK**	**FK**	**Description**
id	INT	YES		Unique identifier of specialization filed
content	VARCHAR(255)			Filed name
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time
users_id	INT		YES	Reference of users table
patients_id	INT		YES	Reference of patients table

Table 13. Supporterlogs database table

Places:

This table will contain various location details, which will be displayed in the examination form.

Column Name	Data type	PK	FK	Description
id	INT	YES		Unique identifier
name	VARCHAR(255)			Name of the places
createdAt	DATETIME			Transaction created date and time
updatedAt	DATETIME			Transaction updated date and time
deletedAt	DATETIME			Transaction deleted date and time

Table 14 Places database table

5.7 Application Domain Model

Application domain model has been presented as an application domain modeling approach that extends Object-Process Methodology with a classification mechanism with two elements: roles, which are stereotypes-like elements, and multiplicity indicators. We demonstrated the use of application domain by applying it to the domain of Access Control and two applications in this domain.

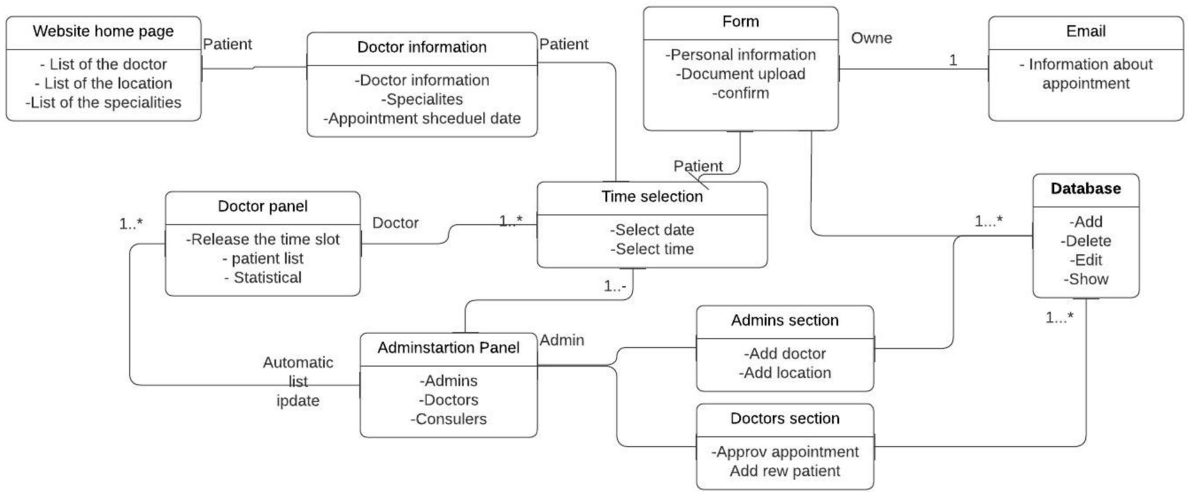


Figure 29 Application domain model

In this online website we have many different components that show in the above. Once patient enter into the website they can select different doctor with specialty after it moves to appointment component where patient can book examination with time. If time is not available on that day means doctor is fully booked. Now we will enter in new component which is doctor panel, using their personal detail they can login into the system and update the time according which can directly reflate on the web page. Once doctor release some time slot the patient can book new appointment that's show in form component where they have to share personal detail, illness any allergic and so on. Now we will enter into the administrative panel, in this admin can add new doctor, add new location. Once the admin receives the new appointment the it will approval and send confirmation to the patient and, now doctor will get new patient in the existing list. This is full life cycle of the one appointment in the system.

5.8 Application Layer Model

The application layer is an important aspect of the creation of any sort of application since it shows how many layers there are and what technologies will be employed in each layer to benefit the end user. There are three main layers in this research, and we divided them into the first, middle, and last layers. The layered structure of the application is depicted in the diagram below.

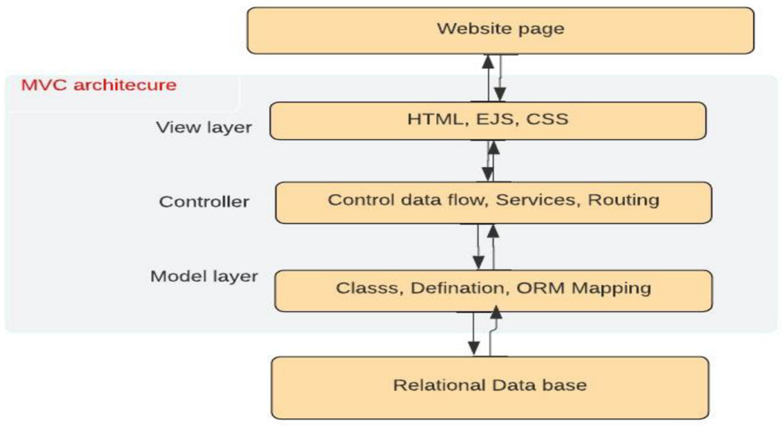


Figure 30 Application layer

In the first layer, the front user will visit our website via various approaches, such as different technology-based web browsers. In addition, the end user can check the website using a mobile browser. The second layer is the application's heart because we used MVC (Model, View, and Controller) architecture on it. Front-end technologies such as HTML, embedded JavaScript

(EJS), and CSS are used in View (cascading style sheet). With their assistance, we created a web page with a better user interface for the end user. We built business logic in the controller, for example, when we receive data from the front end, we will store it in a specified way. As a result, our web application will work smoothly. Furthermore, we specify the application's flow using the route. In the model, we will create multiple classes based on the file so that we can simply map with ORM mapping tools to conduct various operations on supplied data. The data will be stored in a relational database, with MySql as the final layer. This layered is exclusively for the database, so we may do any action on it without mistake and receive a useful result in our daily lives.

6 Usability test

6.1 The Usability test is compound from the below steps

1. UI testing definition:

In this, We have to first define our research area, Based on it we will improve the graphical user interface so the end user can easily understand the application and take more advantage. In our application, we mainly involved more graphics.

2. User group or personas definition.

Personas are great for strongly user-focused applications such as apps for mobile devices. User Groups are ideal for common applications such as internet web pages frequented by a large number of users. Both groups mainly focus on the goal of the product.

3. UI test.

Here, we study the cognitive type of usability study where we use a 5-6 people group which help to improve the usability of the application and got a positive outcome on the test. They can easily carry out the task within a given system.

4. The post-test interview.

In this phase, we got some UI base suggestions from the UI test phase. Where a group of user advice, creates more focus in design where an end user can access all the futures of the application on a single page.

5. The usability issues definition.

Sort out the bug as per the information and updated the system.

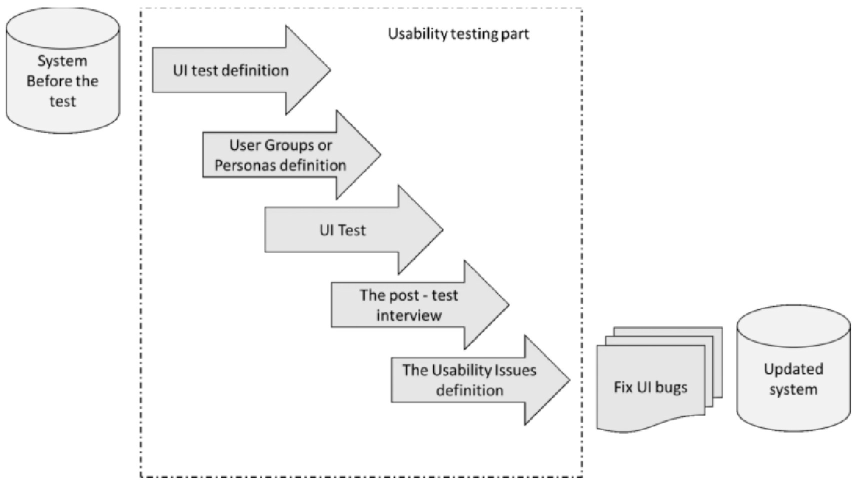


Figure 31. Usability test steps (Pavlickova and Pavlicek, 2019)

6.2 Testing

a functional prototype was subsequently tested on a selected sample of respondents.

6.2.1 Test Scenario

At the beginning of the testing, the participants were introduced to the application. First, they were told what kind of application it is, what it is for, and for whom it is intended, and then they moved on to the actual testing. The scenario was built to test the key features of the application. In the second part of the testing, the participants filled out a simple form to express their opinion about the application and the feelings it evokes in them.

Test Scenario Questions:

1. Open the web application and select the doctor and just click it.
2. Select the date which is suitable for them and also check the detail of the doctor, fees, and location. To check the availability of the Doctor.
3. Open appointment form.
4. Book any 3-time slot from the appointment list.
5. Book 2 time slots with different dates.
6. What type of document file you can upload in the form?
7. Where you can find the location detail of the doctor?

6.2.2 Testing process

Testing of the interactive prototype took place in the laboratories of HUBRU ČZU in Prague - Laboratory for the Study of Human Behavior. The application was tested by qualitative testing using the eye-tracking method. Participants came individually. First, they were introduced to the purpose of creating the application. According to the already mentioned scenario, which was read out loud to the individual participants during the testing, the participant fulfilled the individual tasks and during this process, his vision was scanned over the screen. After completing the scenario, the participants filled out a simple form used to express their opinion. Testing revealed an incorrect position of the button, miss match the color of the button as per standard and therefore a possible correction proposal was subsequently created. The new solution has not yet been tested. From the eye-tracking records, it is clear that users were looking at the right parts of the screen at the given moments. In the end, each participant was asked to fill out a short form with the following questions:

1. You find the application intuitive (it was immediately clear how it is operated)
    a. Possibility to answer numerically 1 (minimum agreement) - 5 (maximum agreement)
2. Can you think of any feature that you would miss in the app?
3. Did something seem incomprehensible or unnecessary?
4. How do you feel when using the application?
    a. Possibility to answer numerically 1 (unpleasant) - 5 (pleasant)
5. What would you change about the application?
6. How likely are you to enjoy this app once it's finished?
    a. Possible answer numerically 1 (low) - 5 (high)

6.3 Formulation of results

None of the participants had a problem with the first two tasks. In the third and fourth tasks, two participants hesitated for a while where to select the appropriate button for action. All the participants had a problem to be used to confirm the appointment. Each of them first tried to look at the first button which was about the cancel then confirm medical appointment button. Furthermore, the color of the cancel button was not as per standard. From these results, it is clear that this form of the application is not intuitive enough and therefore a corrective solution was proposed

6.3.1 Evaluation of the application

Thanks to the expression of the opinions of individual participants using a questionnaire, the following pros and cons were created.

6.3.1.1 Likes

• Users appreciated that the application is clear and easy to use.

• Users liked that both the image and name could be clicked when moving from one web page to another page.

6.3.1.2 Dislikes

• Change the location of the button on the medical appointment form.

• Change font/colors.

• On the home page, the rounded image of the doctor is not like because it looks unclear.

The appointment form is long.

6.3.2 Recommendations for repairs

• It would be good to change the location of the button on the application form as per standard views so the user can directly focus on the confirm medical appointment button.

-The second recommendation is, Change the color of the cancel button.

6.3.3 Fixing a faulty part of the application.

Use Case

The user expects:

- • Full view of the form.
- - Enter personal detail.
- • Allocate button as per standard view.

Scenario

The system will display:

- Different fields where users can write the detail.

- With help of button establishment where you can send the filled form to the management team.

- Buttons for switching to other screens.

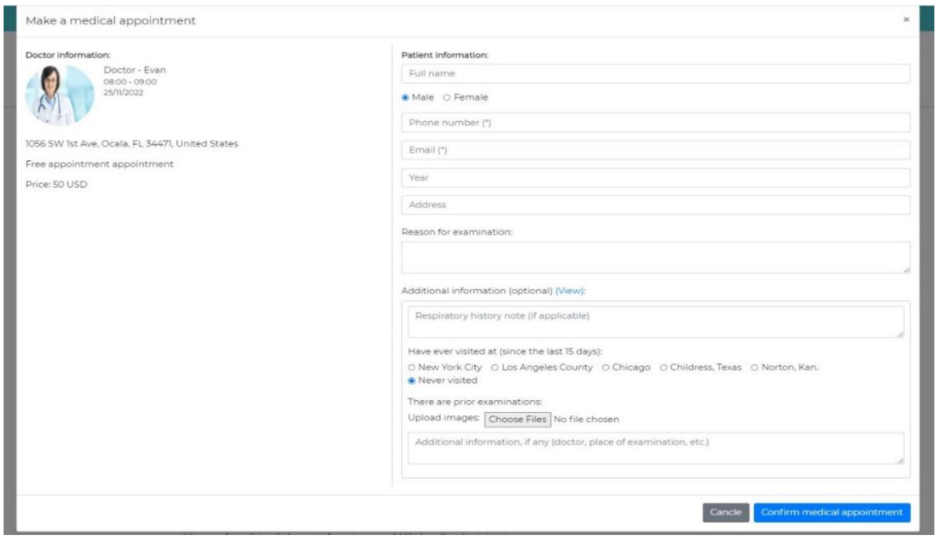
The system expects a user action.

- A keyboard will appear in the text file.
- Button press, store the data in the table, and send confirmation by email.
- Cancel button press, back to home page UC=" Doctor page".
- Press "choose file " to upload the document.

When selection.

- Select Homepage, Which will redirect to UC=" Landing page".
- Select For the patient, Which will redirect to UC=" For patient on navigation bar".
- Select For doctor, Which will redirect to UC=" For Doctor on the navigation bar"

6.3.4 Graphical design



Make a medical appointment

Doctor information:

Doctor - Evan
08:00 - 09:00
25/11/2022

1056 SW 1st Ave, Ocala, FL 34471, United States

Free appointment appointment

Price: 50 USD

Patient information:

Full name

[✓ Male] Female

Phone number (*)

Email (*)

Year

Address

Reason for examination:

Additional information (optional) (View):

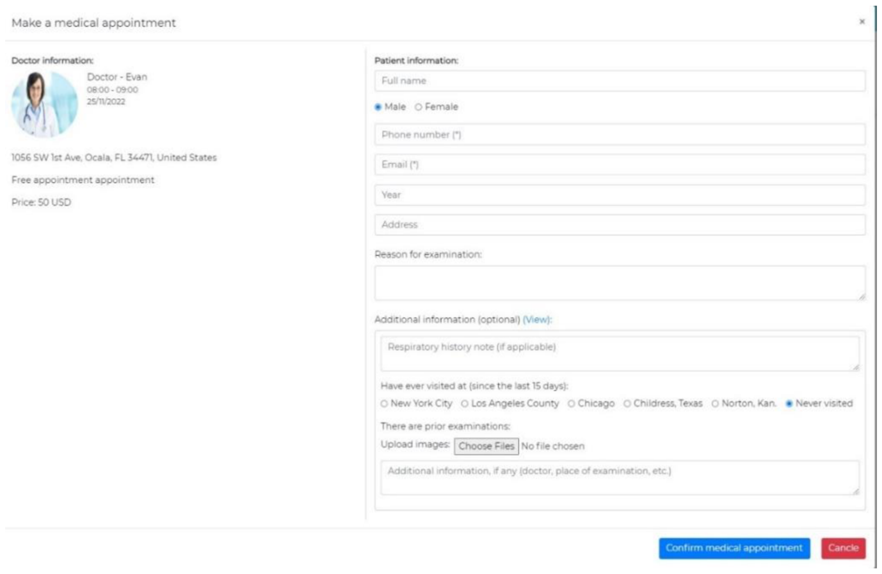
Respiratory history note (if applicable)

Have ever visited at (since the last 15 days):
New York City   Los Angeles County   Chicago   Childress, Texas   Norton, Kan.   [✓ Never visited]

There are prior examinations:
Upload images: Choose Files No file chosen

Additional information, if any (doctor, place of examination, etc.)

Figure 32 Before Test


Make a medical appointment ×

Doctor information:
Doctor - Evan
08:00 - 09:00
25/11/2022

1056 SW 1st Ave, Ocala, FL 34471, United States

Free appointment appointment

Price: 50 USD

Patient information:
Full name
● Male ○ Female
Phone number (*)
Email (*)
Year
Address
Reason for examination:

Additional information (optional) (View):
Respiratory history note (if applicable)
Have ever visited at (since the last 15 days):
○ New York City ○ Los Angeles County ○ Chicago ○ Childress, Texas ○ Norton, Kan. ● Never visited
There are prior examinations:
Upload images: Choose Files No file chosen
Additional information, if any (doctor, place of examination, etc.)

Figure 33. After test

7 Conclusion

The result of the work was supposed to be an improvement of existing online appointment system in the medical field and subsequently a comparison of technology for the development of the online application, and this was achieved. In order for a new application to be designed, it was first necessary to evaluate already existing applications of a similar type. After examining already existing applications, a list of functionalities was created for designing a new application. It was found that DoctorCare application are able to compete with existing web application in term of functionality. For this reason, the theoretical part was first studied, which was devoted to important concepts related to the field of interaction design.

I had little coding knowledge with HTML (actually Domain Object Model), MVC (Model, view, Controller), and relational database at the start of the thesis. My expertise of computer languages aided me in getting started. My purpose was to investigate the impact of implementing dynamic modifications in real time. In addition, learn about the flexibility given by MVC design. During my tenure, I got the opportunity to put my knowledge to use by developing this web application and becoming acquainted with web application development. During my whole study I learned lot of business logic, different mordent concept about the programming language and how to different technology interconnect with each other during run time.

I investigated how a database can handle the dynamic data storage that used to occur as a result of different people perceiving things differently at the application code architecture's fundamental data layer. Also I learn about relation database and how its working because early I was so fascinating to learn like how single query can check all the table as per different condition and how it can be performing different operation in database, went through the process of establishing a Multiplatform project and the challenges that may occur when doing so while writing this thesis. I discovered that there can be numerous dangers that are difficult to understand even for experienced coders such as when we have large application with number of component, it makes more difficult task rather than simple. But, once again, the majority of the answers came from the community which helped to learn how to create a dynamic table in the database which can help during the development phase so it can reduce burden for the developer.

The thesis conducted an in-depth examination of the fundamental ideas underlying each technology before implementing them as part of a fully functional online appointment tool. Based on this research, it is also possible to conclude that webs built with MVC architecture are straightforward to run, have a simple human interface, and it can be run on any devices therefore end user of the website can take more advantage of the modern application. A simple user interface and design makes it easier for patients to approach and make decisions without any hesitation. Making an online appointment system website using the MVC has proven to be really advantageous in terms of hospital business utilization because using this application patient can book quick time slot with phone call on the other hospital staff free from the appointment management burden. Furthermore, these websites have extremely little maintenance that is also very easy to accomplish because we design this application with the help of MVC that is so good and easy to understand, and updating these websites is also very easy because developers can easily reach the region where they want to change and manage the relational database according to business logic.

The researcher encountered numerous challenges in gathering data for web development in order to carry out this study. The researcher also encountered difficulties while constructing the online appointment website due to a lack of data or templates to handle time. As a result, developing the appointment web page has required a significant amount of effort. The researcher also encountered issues with data alignment because learning new technology and implementing it to create an effective website would be a difficult undertaking. . This prototype was subsequently tested on six respondents in the HUBRU laboratories at CZU. An error was discovered during testing, so a corrective solution was proposed.

8 Reference

Adya, A. *et al.* (2007) ‘Anatomy of the ADO.NET Entity Framework’, in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: Association for Computing Machinery (SIGMOD ’07), pp. 877–888. Available at: https://doi.org/10.1145/1247480.1247580.

Arnold, K., Gosling, J. and Holmes, D. (2000) *The Java Programming Language*. 3rd edn. USA: Addison-Wesley Longman Publishing Co., Inc.

BARRON, W.M. (1980) ‘Failed Appointments: Who Misses Them, Why They Are Missed, and What Can Be Done’, *Primary Care: Clinics in Office Practice*, 7(4), pp. 563–574. Available at: https://doi.org/https://doi.org/10.1016/S0095-4543(21)01411-1.

Botwe, D.A. and Davis, J.G. (2015) ‘A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software’, in.

Buschmann, F. *et al.* (1996) *Pattern-Oriented Software Architecture - Volume 1: A System of Patterns*. Wiley Publishing.

Chang, K.-Y., Chen, L.-S. and Lai, C.-K. (1999) *Document-View-Presentation Pattern*.

Codd, E.F. (1983) ‘A Relational Model of Data for Large Shared Data Banks’, *Communications of the ACM*, 26(1), pp. 64–69. Available at: https://doi.org/10.1145/357980.358007.

DHTMLX (2020) *JavaScript Trends in 2020*. Available at: https://codeburst.io/javascript-trends-in-2020-b194bebc5ef8 (Accessed: 28 November 2022).

Frisbie, M. (2019) ‘JavaScript in HTML’, in, pp. 13–24. Available at: https://doi.org/10.1002/9781119366560.ch2.

Gackenheimer, C. (2005) *Introduction to React*.

Gamma, E. *et al.* (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. USA: Addison-Wesley Longman Publishing Co., Inc.

Jeffries, R. *et al.* (1991) ‘User Interface Evaluation in the Real World: A Comparison of Four Techniques’, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery (CHI ’91), pp. 119–124. Available at: https://doi.org/10.1145/108844.108862.

Kenney, E. and Ph, D. (2004) *in Designing a Guide to Medi-Cal Managed Care Quality*.

Maguire, M.C. and Isherwood, P. (2018) ‘A Comparison of User Testing and Heuristic Evaluation Methods for Identifying Website Usability Problems’, in *HCI*.

Martina, L. (2016) *Návrh a tvorba měr pro výpočet kvality procesních modelů,” Czech Technical University in Prague*.

Molich, R. and Nielsen, J. (1990) ‘Improving a Human-Computer Dialogue’, *Commun. ACM*, 33(3), pp. 338–348. Available at: https://doi.org/10.1145/77481.77486.

Neumann Marek (2016) *Míry kvality procesních modelů*.

Older, A.E. (1996) ‘British hospital journal and social service’, *British hospital journal and socal service review* [Preprint].

Pavlickova, P. and Pavlicek, J. (2019) ‘Business Process Models (BPMN and DEMO Notation) - Usability Study’, in, pp. 167–174. Available at: https://doi.org/10.1007/978-3-030-35646-0_13.

Pop, D.-P. and Altar, A. (2014) ‘Designing an MVC Model for Rapid Web Application Development’, *Procedia Engineering*, 69, pp. 1172–1179. Available at: https://doi.org/https://doi.org/10.1016/j.proeng.2014.03.106.

Schildt, H. (2006) *Java: The Complete Reference, Seventh Edition*. 7th edn. USA: McGraw-Hill, Inc.

sourcetoad.com (2020) *The Benefits of Using React*. Available at: https://sourcetoad.com/app-development/the-benefits-of-using-react/https://sourcetoad.com/ (Accessed: 28 November 2022).

Thakur, R.N. and Pandey, U.S. (2019) ‘A Study Focused on Web Application Development using MVC Design Pattern’, *International Research Journal of Engineering and Technology* [Preprint]. Available at: www.irjet.net.

Virji, A. (1990) *A study of patients attending without appointments in an urban general practice*.

Xing, Y., Huang, J. and Lai, Y. (2019) *Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development*. Available at: https://doi.org/10.1145/3313991.3314021.

Yadav, P.K. (2022) *ONLINE HOSPITAL MANAGEMENT*.