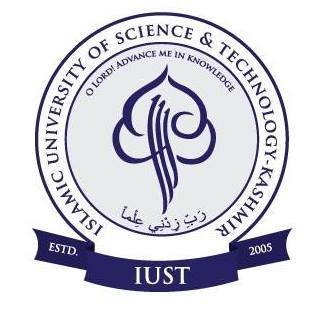
**MASTERS OF COMPUTER APPLICATIONS**

**Project title**

**Online veterinary care system**

**Submitted to**

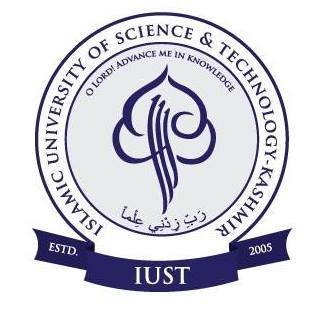


Department of Computer Science, School Of Engineering and Technology (SOET).

Islamic University of Science and Technology,

Awantipora, Pulwama, Kashmir 192122.

**ONLINE VETERINARY CARE SYSTEM**



**Project submitted to the Department of Computer Science, School of Engineering and Technology, Islamic University of Science and Technology in fulfillment of the requirement for the award of the degree of**

**MASTERS OF COMPUTER APPLICATIONS**

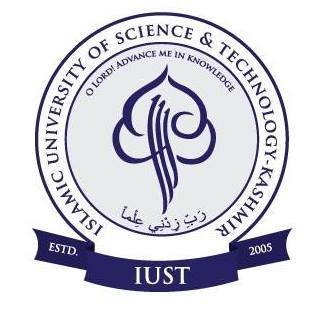
**Submitted by**

1. **Mohammad Waseem Dar (MCA-17-24)**
2. **Zubair Ahmad Lone (MCA-17-46)**
3. **Humaira Parray (MCA-17-17)**
4. **Jasiyah Jabbar (MCA-17-06)**

**Department of Computer Science, School of Engineering and Technology(SOET).**

**Islamic University of Science and Technology,**

**Awantipora, Pulwama, Kashmir, 192122.**



CERTIFICATE

This is to certify that the Project entitled

“**Online veterinary care system**”

Is the original work carried out by

1. **Mohammad Waseem Dar (MCA-17-24)**
2. **Zubair Ahmad Lone (MCA-17-46)**
3. **Humaira Parray (MCA-17-17)**
4. **Jasiyah Jabbar (MCA-17-06)**

**Supervisor Head of the Department**

**Mrs. Shabana Nargis Dr.Javaid Iqbal**

Assistant Professor Department of computer science

DOCS, IUST, Awantipora, Pulwama, 192122 IUST, Awantipora, 192122

**Student Declaration/Certificate**

We the students “***Mohammad Waseem Dar(MCA-17-24), Zubair Ahmad Lone (MCA-17-46), Humaira Parray(MCA-17-17), Jasiyah Jabbar(MCA-17-06)***” hereby declare that the work, which is being presented in the project entitled “**Online veterinary care system”** in the fulfillment of the requirement for the award of **Masters of Computer Application(MCA)** degree in the session 2020, is an authentic record of our own work, carried out under the supervision of **Mrs**. **Shabana Nargis**(Assistant Prof, Department of Computer Science, Islamic University of Science & Technology, Awantipora).

The matter embodied in this Project has not been submitted for the award of any other degree.

1. **Mohammad Waseem Dar (MCA-17-24)**

-------------------------------------------------------------------

1. **Zubair Ahmad Lone (MCA-17-46**

---------------------------------------------------------------------

1. **Humaira Parray (MCA-17-17)**

-----------------------------------------------------

1. **Jasiyah Jabbar (MCA-17-06)**

---------------------------------------------------------------------

This is to certify that the above statements made by the candidates are correct the best of my knowledge.

**Supervisor Head of the Department**

**Mrs. Shabana Nargis Dr.Javaid Iqbal**

Assistant Professor Department of computer science

DOCS, IUST, Awantipora, Pulwama, 192122 IUST, Awantipora, 192122

**Acknowledgement**

All the praises for Almighty ALLAH, our head bows in humble towards Him. He blessed us with the strength and courage to accomplish this task.

We fall short of words to pen down anything for the co-operation and encouragement extended by our Parents throughout the completion of this project. A mere thanks is little for their love, encouragement and support. Their prowess and sacrifice culminated in to the men, we are.

We are highly thankful to our project guide **Mrs. Shabana Nargis**(Assistant Professor, Department of Computer Science, IUST, Awantipora,Pulwama),without her support this project was very much difficult for us.

We are also very thankful to our co-guide. He helped us very dedicatedly and sincerely.

We are also thankful to our teachers and other staff members of our department. They have been very helpful and kind to us throughout our MCA course.

We would also like to extend our thanks to all those who gave us support and encouragement in one way or other and helped us in the execution of this project.

**INDEX**

1. INTRODUCTION …………………………………………………………….1

1.1) PROJECT OVERVIEW…………………………..……………………….1

1.2) PROJECT OBJECTIVES…………………………………………………..1

2. PROPOSED SYSTEM........................................................................................2

2.1) FEASIBILITY STUDY……………………………………………………..2

2.2) PROJECT PLAN………………………………………………...………....5

2.3 SOFTWARE ENGINEERING PARADIGM USED……………………….5

3. SYSTEM ANALYSIS…………………………………………………………..6

3.1) REQUIREMENTS ENGINEERING………………………………………6

3.2) ANALYSIS MODEL…....…………………………………………………7

3.3) SOFTWARE REQUIREMENT SPECIFICATION……………………….9

4. SYSTEM DESIGN ……………………………………….………………….…14

4.1) MODULE DESIGN ……………………………………………………......19

4.2) DATA FLOW DAIGRAMS (DFD)………………………………….....….19

4.4)UML DIAGRAMS.………………………………………………..… ……23

4.5) ERD…………………………………………………………….. ..……… 24

4.6) INPUT /OUTPUT DESIGN……………………………………..………….24

5. CODING…………………………………………………………………………28

6. TESTING ………………………………………………………………………..45

7. IMPLEMENTATION OF PROJECT …………………………………………...53

8. SNAPSHOTS….…………………………………………..……………………..54

9. CONCLUSION………………………………………………….…..…………...67

10. FUTURE WORK/RECOMMENDATIONS……………………………………68

11. APPENDICES…………………………………………………………………..69

12. BIBLIOGRAPHY……………………………………………………………...70

**Chapter 1. INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Online applications are playing an important role in our day to day life. From online shopping to online doctor booking, these applications are saving time and helping ineffective management of resources. As far as veterinary care system is concerned, many applications for online veterinary systems like “vet coach live” are available, however these applications are limited in providing the information about vaccination schedule, no prompt response in case of emergency and less number of appointments. These applications charge fee when users want detailed information about vaccination, disease, treatment etc. Our application will overcome these limitations by introducing instant messaging where doctor can chat with user through instant messaging. Our application will also provide more information on vaccination schedule free of cost.

**1.2) PROJECT OBJECTIVES:**

Our goal is to develop a robust application which will help users to cure their pet. Following are the objectives of our application:

1. To help users to book appointments with the veterinary doctors for the treatment of their pets.
2. To help users to chat with doctors instantly in case of emergency.
3. To help users to get enough information regarding vaccination schedule, camps to be organized etc.

**Chapter 2. PROPOSED SYSTEM**

**Existing system:**

At present, there are many websites like https://www.petcoach.co/, http://vetcoachlive.com/ that provide information related to pets and provide answers to users’ questions related to the diseases of pets.

**Disadvantages:**

* Existing websites do not provide much information and brief answers to users’ questions and require money when more information is needed.
* In case of emergency, users do not get prompt response for treating their pets.

**Proposed system:**

In proposed system website is developed by which users can book appointments with the doctors available for their pets. They can also use instant messaging service provided by the application in case of emergency. Enough information is also available to the users which can be beneficial to them regarding the diseases of their pets and the vaccines available.

**Advantages:**

* Users can book appointments by selecting a specific doctor and the time slot available.
* Users can instantly chat with the doctors in case of emergency and doctor can recommend them the immediate treatment.
* A lot of information will be available to the users regarding the pets’ diseases and the vaccines.
* Users can give feedback about doctors and the feedback will be evaluated by the Admin and if negative feedback is found, the doctor may be rejected.

**2.1 Feasibility Study:**

A feasibility study is undertaken to determine the possibility of either improving the existing system or developing a completely new system. This helps to obtain an overview of the problem and to get rough assessment of whether feasible solutions exist. The purpose of feasibility study is to determine whether the requested project is successfully realizable. There are three aspects of feasibility study namely:

**Types of Feasibility Study: -**

**2.1.1) Technical feasibility.**

**2.1.2) Economic feasibility.**

**2.1.3) Operational feasibility.**

**Technical feasibility:**

Technical feasibility is concerned with specifying equipment and software that will successfully support the required task. It centres on the required /existing computer system(hardware/software) and to what extent it can support the proposed application. For example, if the current computer is operating at 80% capacity, then running another application could overload the system or require additional hardware. This requires financial consideration to accommodate technical enhancement. This should answer the following questions:

* Whether the project can be carried out with the existing equipment?
* Whether the existing software is enough?
* If a new technology is required, how best can it be implemented?
* In case of our application, the following software that is required for developing the application is:

Microsoft Visual Studio 2017 or higher.

Microsoft SQL Server 2014 or higher.

For running the application, the following software is required:

Internet Information Services.

The following Operating System is required to develop and run the application:

Microsoft Windows 8 or higher with Internet Information Services installed. OR

Microsoft Windows Server 2016 or higher with Internet Information Services installed.

And the following hardware is required for

developing and running the application:

Intel Core i3 or higher.

8GB of RAM or higher.

100GB of Hard Disk Space or higher in case there is too much data stored in database.

**Economic feasibility:**

Economic feasibility study is the most frequently used method for evaluating the effectiveness of a new system. Cost-benefit analysis is performed to determine the benefits and savings that are expected from the new system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the new system. By conducting this study, the analyst can ascertain the following:

* Whether the project is economically feasible?
* If enough funds are not available, then what are the sources of funds?
* Whether there are sufficient benefits when compared to the costs incurred?

In case of our application, the main resources are the developers developing the application. There will be one time cost of developing the application. Also there will be the cost of domain and hosting when the application is launched to the general public. But nowadays there are many hosting and domain providers which provide domain and hosting services at very low costs. When the application is complete and launched to the general public, then we will earn from appointments which are booked by the users and we can also display ads on the website which will be a source of revenue.

**Operational Feasibility:**

Operational feasibility is concerned with human, organizational and political aspects. Operational feasibility covers two aspects. One is a technical performance aspect and the other is acceptance within the organization. Technical performance includes issues such as determining whether the system can provide the right information for the organization’s personnel, and whether the system can be organized so that it always delivers this information at the right place at the right time.

Our application will be technically sound because we will use RESTful (Representational State Transfer) API’s which are light weight and perform faster than simple server side based web applications. And it will also be accepted by public because it offers beneficial material to the users and also benefits them in making appointments and hence saving their time and serving their pets in case of emergency. And also this application does not harm any person’s social, political or economic interests.

**2.2. Project Planning:**

Software Project Management Begins with a set of activities that are collectively called Project Planning. Before the project begins, the manager and the software team must estimate the work to be done, the resources that will be required, and the time that will elapse from start to finish.

**Project Estimation:**

Project Estimation is an attempt to determine how much money, effort, resources, and time it will take to build a specific software-based system or product. Estimation begins with a description of the scope of the product. The problem is then decomposed into a set of smaller problems, and each of these is estimated using historical data and experience as guides. Problem complexity and risk are considered before a final estimate.

Our project will require the following amount of resources, money and time:

* 120 work hours x 100/hour = 12000
* Core i3 Laptop/Desktop or higher configuration
* 4GB of RAM or higher (Recommended 8GB)
* 10GB of Disk Space or higher
* Windows 8 or higher operating system

To test on multiple systems, Internet Information Services is required and network sharing should be on in all the systems.

**Project Planning Objectives:**

**Software Scope:** The first activity in Software Project Planning is the determination of software scope. Software Scope describes the data and control to be processed, function, performance, constraints, interfaces and reliability. In case of our project, the data are the details of the users like name, email, phone no etc., the details of doctors, the details of booking slots, the details of appointments, messages to be shared between users and doctors and the vaccination schedule and other information available on the website. Our project will use RESTful api’s which are light weight and are very much responsive.

**Feasibility:** Once the scope has been identified, it is reasonable to ask: “Can we build software to meet this scope? Is the project feasible?”. Our project will meet the goals and objectives discussed above.

**Resources:** Another software planning task is the estimation of the resources required to accomplish the software development effort. We have estimated that our project will need almost 120 work hours of a developer.

**Decomposition:** Software Project estimation is a form of problem solving, and in most cases, the problem to be solved is too complex to be considered in one piece. For this we decompose the problem, re-characterizing it as a set of smaller problems. In our project, we have decomposed the whole software into the modules discussed above.

**2.3. Software Engineering Paradigm Used**

We have used Object Oriented Programming Paradigm in developing this application.

**Chapter 3. SYSTEM ANALYSIS**

**Introduction**

System analysis is the process of examining the situation with the intent of improving it through better procedures and methods. System design is the process of planning a new system to either replace or complement an existing system. But before any planning is done, the old system must be through understood and the requirements determined. System analysis is therefore, the process of gathering and interpreting facts, diagnosing problems and using the information to re-comment improvements in the system. Or in other words, system analysis means a detailed explanation or description. Before computerizing a system under consideration, it has to be analysed. We need to study how it functions currently, what are the problems, and what are the requirements that the proposed system should meet.

System analysis is conducted with the following objectives in mind:

1.Identify the customer’s need.

2.Evaluate the system concept for flexibility.

3.Allocate functions to hardware, software, people, database and other system elements.

4.Establish cost and schedule constraints.

5.Create a system definition that forms foundation for all the subsequent engineering work.

**3.1 Requirement Engineering:**

A requirement is defined as a condition or capability that must be met or fulfilled by a system to specify a contract, standard, specification. The requirements defined for a system should be correct, consistent, verifiable and traceable. Requirement Engineering is the process of eliciting, understanding, specifying and validating customer’s requirements. Requirements Engineering is the first technical step in the software process. Analysis must focus on information functional and behavioural domains of the problem. Analysis is about understanding situations, not solving problems. This is the first and important phase of software development because what will be developed on analysis. Only when a complete analysis is made, recommendations for design can be made.

Requirement Engineering provides the appropriate mechanism for understanding what the customer wants, analysing needs, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously validating the specification, and managing the requirements as they are transformed into an operational system.

The iterative process of requirement engineering consists of following steps: -

**Inception**

At this stage different sources of information including doctors, pet owners and shepherds were interviewed. The intent was to build a basic understanding of the problem, the people who want the nature solution that is desired and the effectiveness of preliminary communication and collaboration.

**Elicitation**

At this stage we asked the users (pet owners) what the objectives for the system are, what is to be accomplished, how the system fits into the needs and finally how the system is to be used on a day to day basis.

**Elaboration**

We expanded and refined the information obtained during inception and elaboration. We focused developing a refined technical model of software functions, features and constraints.

**Negotiation**

In this activity we reconciled the conflicting requirements proposed by the various pet owners, doctors and other health workers.

**Specification**

Here we produced a written document specifying all the captured requirements in a consistent and therefore more understandable manner. This document served as the foundation subsequent software engineering activities.

**Validation**

The product produced as a consequence of requirements engineering was assessed for quality during this step, in order to ensure that all software requirements have been stated unambiguously; that inconsistencies, omissions and errors have been detected and corrected; and that the work product conforms to the standard established for the process the project and the product.

In case of our software, we do not have to get requirements from any person or group of persons. Instead, we have to analyze existing system and see the limitations of these systems and try to overcome those limitations in the existing systems. For this purpose, we have analyzed some websites like https://www.petcoach.co/ and vetcoachlive.com. After analyzing these websites, we have noted some limitations and will overcome those limitations in our software.

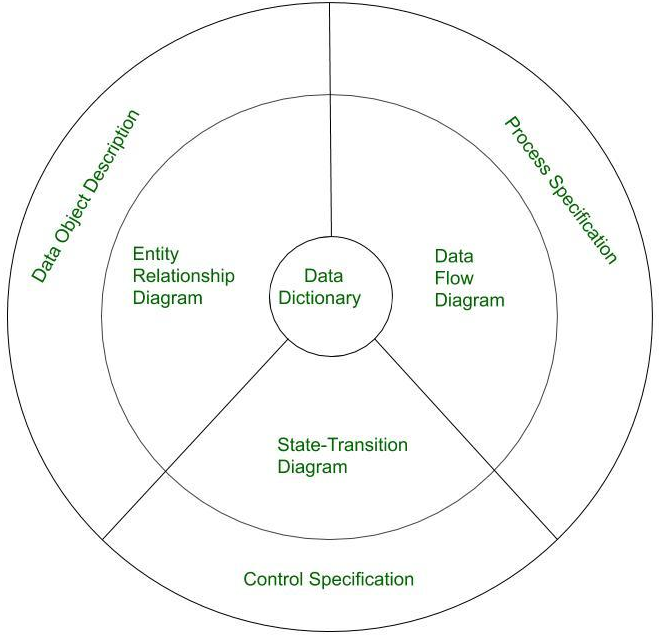
**3.2. Analysis Model**

**Analysis Model** is a technical representation of the system. It acts as a link between system description and design model. In Analysis Modelling, information, behaviour and functions of the system is defined and translated into the architecture, component and interface level design in the design modelling.

**Objectives of Analysis Modelling:**

1. It must establish a way of creation of software design.
2. It must describe requirements of customer.
3. It must define set of requirements which can be validated, once the software is built.

**Elements of Analysis Model:**



**Figure 1. Analysis Model**

**Data Dictionary:**

It is a repository that consists of description of all data objects used or produced by software. It stores the collection of data present in the software. It is a very crucial element of the analysis model. It acts as a centralized repository and also helps in modelling of data objects defined during software requirements.

**Entity Relationship Diagram (ERD) :**

It depicts relationship between data objects and used in conducting of data modelling activity. The attributes of each object in the Entity Relationship Diagram can be described using Data object description. It provides the basis for activity related to data design.

**Data flow Diagram(DFD) :**

It depicts the functions that transform data flow and it also shows how data is transformed when moving from input to output. It provides the additional information which is used during the analysis of information domain and serves as a basis for the modelling of function. It also enables the engineer to develop models of functional and information domain at the same time.

**State Transition Diagram (STD) :**

It shows various modes of behaviour (states) of the system and also shows the transitions from one state to other state in the system. It also provides the details of how system behaves due to the consequences of external events. It represents the behaviour of a system by presenting its states and the events that cause the system to change state. It also describes what actions are taken due to the occurrence of a particular event.

**Process Specification:**

It stores the description of each functions present in the data flow diagram. It describes the input to a function, the algorithm that is applied for transformation of input, and the output that is produced. It also shows regulations and barriers imposed on the performance characteristics that are applicable to the process, and layout constraints that could influence the way in which the process will be implemented.

**Control Specification:-**

It stores the additional information about the control aspects of the software. It is used to indicate how the software behaves when an event occurs and which processes are invoked due to the occurrence of the event. It also provides the details of the processes which are executed to manage events, .

**Data object description :**

It stores and provides the complete knowledge about a data object present and used in the software. It also gives us the details of attributes of the data object present in Entity Relationship Diagram. Hence, it incorporates all the data objects and its attributes.

**3.3. Software Requirement Specification**

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

Qualities of SRS:

* Correct
* Unambiguous
* Complete
* Consistent
* Ranked for importance and/or stability
* Verifiable
* Modifiable
* Traceable

Types of Requirements:

The below diagram depicts the various types of requirements that are captured during SRS.



**Figure 2. Types of requirements captured during SRS**

We have gathered all the requirements for our application and found out that our application should be able to perform the following activities.

* Enable a user/doctor to register with the system.
* Enable the admin to approve/reject a doctor, activate/deactivate a user.
* Enable the doctor to add booking slots available.
* Enable the user to book appointment against a selected date and booking slot.

Online veterinary care system is an online appointment booking platform which provides services of booking appointments, curing of pets, online consultation, vaccination schedule, information about production of hygienic milk and fertilization management.

**Document Conventions**

When writing the SRS for online veterinary care the following conventions are:

* To make the document more effective and readable.
* We used the normal font style, 11 font size and headings are underlined.

**Intended Audience and Reading Suggestions**

This Software Requirements document is intended for:

The intended audience of this document is all major stakeholders which include the developers, testers, the project user and anyone evaluating the project.

− Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes moreorganised for modifications.

-End users of this application who wish to read about what this project can do.

**Product Scope**

Online veterinary care is an online web based platform that will provide facilities to the users to care their pets in a better way. Our software will provide free consultation and would provide information about vaccination schedule.

The scope of our project is to design a safe and easy to use website for the services provided to users.

**Overall Description**

**Product Perspective**

Online veterinary care system is a new and innovative which has additional feature of instant messaging. The product aimed at the person who do not want to visit the doctor physically.

**Product Functions**

• Un- registered users may register to the system.

. Un-registered users may search to the system.

• Registered person may login to the system.

• Administrator may add/update/delete useful tips.

• Administrator may add/update/delete vaccination schedule.

• Administrator may get reports from the system.

User Classes and Characteristics

**Registered Users:** Any user that is registered with the system,. Registered users can book appointment and can get consultation. They can share vaccination schedule and can get their pets cured.

**Un-registered users**: Un-registered users cannot get the information.

2.4 Operating Environment

Online veterinary care system will operate on the latest versions of Google Chrome (Chrome 2018), Mozilla Firefox (version 59.0.1), and Internet Explorer (version 11). Users will be able to use the software using desktops, laptops or mobile devices.

2.5 Design and Implementation Constraints

4GB RAM (at least).

10GB of disk space.

Windows 8 operating system.

**User Documentation**

User should be familiar with the terms like registration and login . User can get the information through the web application.

**2.7 Assumptions and Dependencies**

• All Users need email id or phone number for registration.

• All Users must have internet connection and internet browser.

• There will be no server latencies.

• All governmental regulations will be considered by the system.

**External Interface Requirements**

**3.1 User Interfaces**

• Inputs will be entered via standard web controls such as combo box, check box, text

box, calendar, etc.

• Navigation and acceptance will be handled with buttons.

**3.2 Hardware Interfaces**

Web or mobile browsers will be used to access the software.

**3.3 Software Interfaces**

Frameworks will be used for user interfaces. Profile information will be gathered using web forms. ASP.NET will be used to access the database, to validate the input, and to retrieve/display the results.

**Communications Interfaces**

The system will use TCPIP protocol for communication, SMTP protocol for email and HTTP

protocol for website. User form data will be transferred using HTTP-POST method and search

data will be transferred using HTTP-GET method. Password data will be encrypted.

**Registration**

**Description and Priority**

Non-registered users are required to register to perform the “login” function. This feature is of high priority.

**Stimulus/Response Sequences**

“Register” button/link click: Register data form will be displayed.

“Submit” button click: Register data will be validated, error messages will be displayed as labels or dialog boxes, successful registration will forward the user to user main page.

**Functional Requirements**

Un-registered user has to register with e-mail id or phone number and has to set a strong password through which he/she can then login into the system.

Login

Description and Priority

Registered Users are required to login in order to perform functions like “create a profile”. This feature is of high priority for registered Users and low priority for un-registered users.

Stimulus/Response Sequences

“Login” button/link click: Form will be displayed.

**Functional Requirements**

The various functional modules that can be implemented by the system will be

**Registration:-** The user must be registered.

**Login: -** Customer login to the system by entering valid user id and password.

**Chapter 4. SYSTEM DESIGN**

**Overview:**

Design is a meaningful engineering representation of something that is to be built. It can be traced to a customer’s requirements and at the same time assessed for quality against a set of predefined criteria for “good” design. In the software engineering context, design focuses on four major areas of concern data, architecture, interfaces, and components.

**System Design Steps:**

Design begins with the requirements model. We work to transform this model into four levels of design detail: The Data Structure, the System Architecture, the Interface Representation, and the Component Level Detail. During each design activity, we apply basic concepts and principles that lead to high quality. The outcome of System Design is the Design Specification.

**Software Design and Software Engineering:**

Software Design sits at the technical kernel of software engineering and is applied regardless of the software process model that is used. Beginning once software requirements have been analysed and specified, software design is the first of three technical activities- design, code generation, and test- that are required to build and verify the software. Each activity transforms information in a manner that ultimately results in validated computer software.

Each of the elements of the analysis model provides information that is necessary to create the four design models required for a complete specification of design.

**Data Design:** The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide the basis for the data design activity. Part of data design may occur in conjunction with the design of software architecture. More detailed data design occurs as each software component is designed.

We have used Microsoft SQL Server to store the data in the database. The corresponding model classes were created in the project e.g. for storing doctor, admin and user the User model was created with some properties like Name, Email, Mobile etc.

**Architecture Design:** The architectural design defines the relationship between major structural elements of the software, the “design patterns” that can be used to achieve the requirements that have been defined for the system and the constraints that affect the way in which architectural design patterns can be applied.

We have used MVC design pattern to design the application. In this model in ASP.NET, the request goes to the controller when it arrives e.g., when we load the Login Page the request goes to the Login Controller in the Controllers folder and then to the Index action method. That action method returns the Index.cshtml view to the browser which is located in View/Login folder of the application. This view contains actual HTML, CSS and Javascript required to run this page. In the same way, other pages are rendered.

**Interface Design:** The interface design describes how the software communicated within itself, with systems that interoperate with it, and with humans who use it. An interface implies a flow of information (e.g., data and / or control) and a specific type of behaviour. Therefore, data and control flow diagrams provide much of the information required for interface design.

We have used HTML5, CSS3 and Bootstrap for the User Interface Design. We have used HTML5 for designing the interface of the pages and styling it with CSS3 and Bootstrap. CSS3 and Bootstrap gives rich layout to the pages. Bootstrap framework is used for creating responsive pages i.e., the pages which can adjust automatically on any type of device like desktop, mobile, tab etc. according to the width of the device.

**Component Level Design:** The component level design transforms structural elements of the software architecture into a procedural description of software components. Information obtained from the PSPEC, CSPEC, and STD serve as the basis for component design.

We have used three components. viz., Admin, Doctor and User. Admin component can monitor the registered doctors and users. The admin can approve or reject any doctor or user. Doctor can add booking slots, view appointments booked against the booking slots by users. User can book an appointment. Users can chat with doctors or admin. Doctors can chat with admin or users. Admin can chat with users and doctors.

**Modules:**

**Admin:**

When a doctor registers with the system, he will not be immediately available on the website. Instead, first he will be reviewed by Admin and if found genuine, then the Admin will approve the doctor. Admin can also add information like Vaccination Schedule, Fertility Management, Production of Hygienic Milk and various types of Information Dissemination into the system.

**User:**

A user who wants treatment for his pet(s) regularly or occasionally can register with the system. After registering, he can book appointment, chat with a doctor in case of emergency. A user can also see the information like vaccination schedule, camps to be organized on the website and for this he doesn’t need to register on the website.

**Doctor:**

A doctor can register with the system by entering his/her details and then he/she can be approved by admin and after approval, he/she can enter his/her booking slots available. He/she can also chat with the users and provide them necessary recommendations for treatment if they contact him/her in case of emergency.

**Data Flow Diagrams:**

**Figure 4: - Context level diagram (level 0).**

**Figure 5:- Level 1 DFD for vaccination.**

User

Admin

Login (Username & Password)

Acknowledge Login (Username & Password)

Acknowledge

(Regd./Unregd.)

Doctor

Login (Username & Password) Acknowledge

(Regd./Unregd.)

**Figure 4. Context level diagram (level 0).**

Add Vaccination

Status

(Added/Not added)

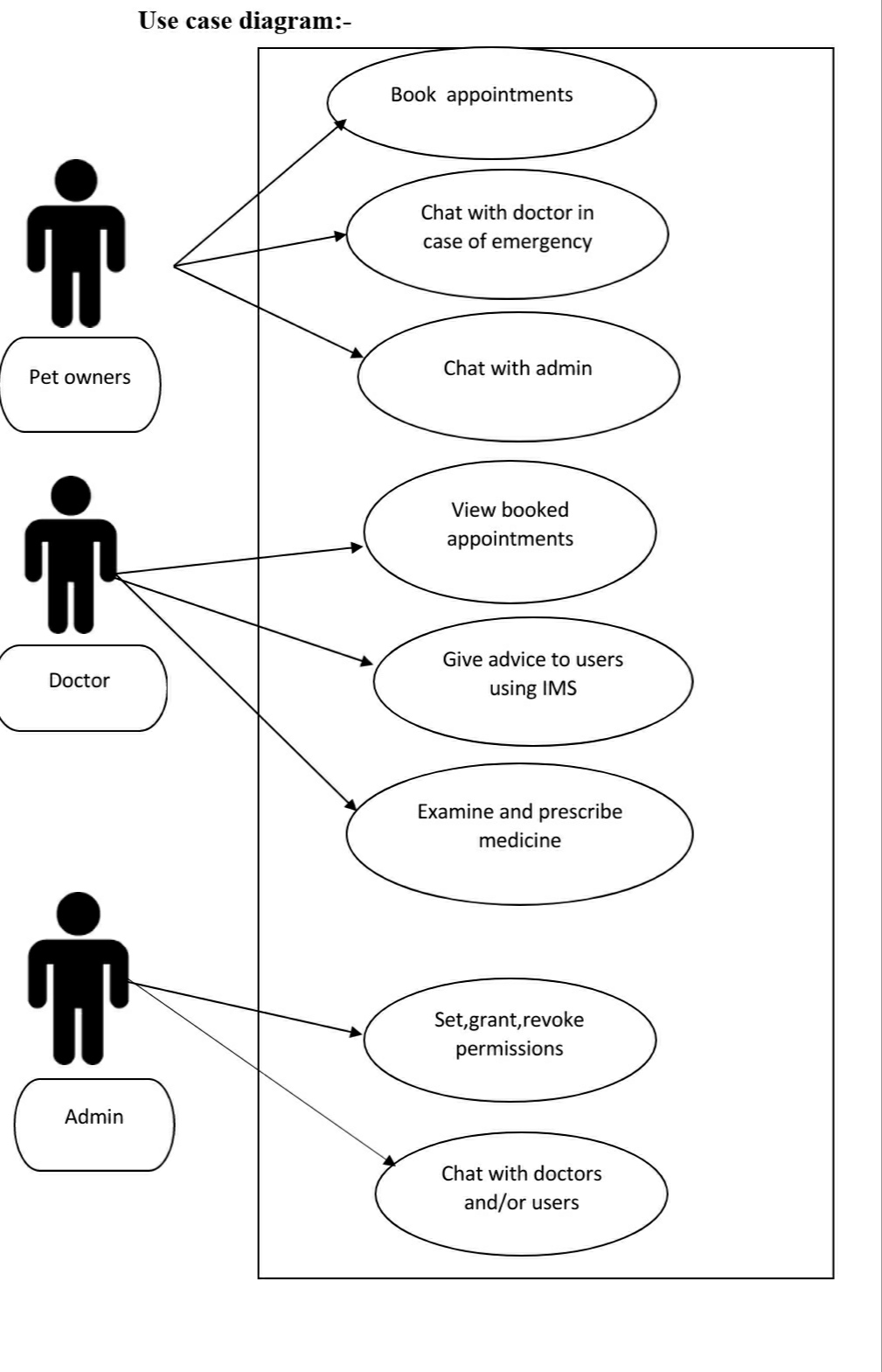
Admin

Edit Vaccination

Status

(Edited/Unedited)

**Figure 5. Level 1 DFD for vaccination**

****

**Figure 6. Use Case Diagram**

**Entity Relationship Diagram:**

**Figure 6. Use Case Diagram**

User

Doctor

Register

Admin

**Figure 7. Entity Relationship Diagram**

**Input /output design:**

In our project, the elements like textboxes and dropdowns have been used for input and tables and pages have been used for output.

**Chapter 5 :- Coding**

function doLogin() {

if ($('#username').val() == '') {

alert('Please enter username.');

$('#username').focus();

return;

}

if ($('#password').val() == '') {

alert('Please enter password.');

$('#password').focus();

return;

}

$.ajax({

url: "/api/token",

type: "POST",

data: $.param({ grant\_type: 'password', username: $("#username").val(), password: $("#password").val() }),

headers: { 'Content-Type': 'application/x-www-form-urlencoded' },

beforeSend: function () {

$('#loading').show();

},

complete: function () {

$('#loading').hide();

},

success: function (res) {

console.log(res);

localStorage.setItem('token', JSON.stringify(res));

if (res.role == 'Admin') {

window.location = '/admin';

} else if (res.role == 'Doctor') {

window.location = '/doctor';

} else {

window.location = '/user';

}

},

error: function () {

console.log('Error occured.');

alert('Invalid Username/ Password');

}

});

}

function doRegister() {

if (!/^[a-zA-Z ]+$/.test($('#name').val())) {

alert('Only letters and spaces are allowed in name.');

$('#name').focus();

return;

}

var reg = /^([A-Za-z0-9\_\-\.])+\@@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

if (!reg.test($('#email').val())) {

alert('Please enter a valid email.');

$('#email').focus();

return;

}

if (!/[0-9]+/.test($('#mobile').val())) {

alert('Only digits allowed in mobile no.');

$('#mobile').focus();

return;

}

if ($('#mobile').val().length > 10 || $('#mobile').val().length < 10) {

alert('Exactly 10 digits allowed in Mobile Number.');

$('#mobile').focus();

return;

}

if ($('#passwordRegister').val() === '') {

alert('Password is required');

$('#passwordRegister').focus();

return;

}

if ($('#passwordRegister').val() != $('#confirmPassword').val()) {

alert('Passwords do not match.');

return;

}

$.ajax({

url: '/Login/GetCaptchaString',

method: 'GET',

beforeSend: function () {

$('#loading').show();

},

complete: function () {

$('#loading').hide();

},

success: function (res) {

if (res !== $('#CaptchaText').val()) {

alert('Captcha not matched. Please try again...');

$('#CaptchaImage').removeAttr('src').attr('src', '/Login/ShowCaptchaImage?' + new Date().getTime());

return;

}

var userObj = {};

userObj.name = $('#name').val();

userObj.email = $('#email').val();

userObj.mobile = $('#mobile').val();

userObj.password = $('#passwordRegister').val();

userObj.isActive = true;

userObj.role = "User";

$.ajax({

url: '/api/User/Post',

data: JSON.stringify(userObj),

method: 'POST',

dataType: 'json',

contentType: "application/json; charset=utf-8",

beforeSend: function () {

$('#loading').show();

},

complete: function () {

$('#loading').hide();

},

success: function (res) {

if (res.added) {

alert('Registration done successfully. Please login now to access your account.');

} else {

alert(res.message);

}

},

error: function (err) {

console.log('Error: ', err);

alert(err);

}

});

},

error: function (err) {

console.log('Error: ', err);

alert(err.statusText);

}

});

}

**Login/Register Page Frontend code**

[HttpPost]

public IHttpActionResult Post(User user)

{

user.ID = Guid.NewGuid();

user.Password = HashPassword(user.Password);

user.AddedOn = DateTime.Now;

return Ok(\_bal.RegisterUser(user));

}

UserBAL \_bal = new UserBAL();

[NonAction]

public string HashPassword(string password)

{

// Use input string to calculate MD5 hash

using (System.Security.Cryptography.MD5 md5 = System.Security.Cryptography.MD5.Create())

{

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(password);

byte[] hashBytes = md5.ComputeHash(inputBytes);

// Convert the byte array to hexadecimal string

StringBuilder sb = new StringBuilder();

for (int i = 0; i < hashBytes.Length; i++)

{

sb.Append(hashBytes[i].ToString("X2"));

}

return sb.ToString();

}

}

**Register Backend coding**

function doRegister() {

if (!/^[a-zA-Z ]+$/.test($('#name').val())) {

alert('Only letters and spaces are allowed in name.');

$('#name').focus();

return;

}

var reg = /^([A-Za-z0-9\_\-\.])+\@@([A-Za-z0-9\_\-\.])+\.([A-Za-z]{2,4})$/;

if (!reg.test($('#email').val())) {

alert('Please enter a valid email.');

$('#email').focus();

return;

}

if (!/[0-9]+/.test($('#mobile').val())) {

alert('Only digits allowed in mobile no.');

$('#mobile').focus();

return;

}

if ($('#mobile').val().length > 10 || $('#mobile').val().length < 10) {

alert('Exactly 10 digits allowed in Mobile Number.');

$('#mobile').focus();

return;

}

if ($('#address').val() === '') {

alert('Address is required.');

$('#address').focus();

return;

}

if ($('#password').val() === '') {

alert('Password is required');

$('#password').focus();

return;

}

if ($('#passwordRegister').val() != $('#confirmPassword').val()) {

alert('Passwords do not match.');

return;

}

$.ajax({

url: '/Login/GetCaptchaString',

method: 'GET',

beforeSend: function () {

$('#loading').show();

},

complete: function () {

$('#loading').hide();

},

success: function (res) {

if (res !== $('#CaptchaText').val()) {

alert('Captcha not matched. Please try again...');

$('#CaptchaImage').removeAttr('src').attr('src', '/Login/ShowCaptchaImage?' + new Date().getTime());

return;

}

var userObj = {};

userObj.name = $('#name').val();

userObj.email = $('#email').val();

userObj.mobile = $('#mobile').val();

userObj.address = $('#address').val();

userObj.password = $('#passwordRegister').val();

userObj.isActive = false;

userObj.role = "Doctor";

$.ajax({

url: '/api/User/Post',

data: JSON.stringify(userObj),

method: 'POST',

dataType: 'json',

contentType: "application/json; charset=utf-8",

beforeSend: function () {

$('#loading').show();

},

complete: function () {

$('#loading').hide();

},

success: function (res) {

if (res.added) {

alert('Registration done successfully. Please login now to access your account.');

} else {

alert(res.message);

}

},

error: function (err) {

console.log('Error: ', err);

alert(err.statusText);

}

});

},

error: function (err) {

console.log('Error: ', err);

alert(err.statusText);

}

});

}

**Doctor Registration frontend code**

var username = JSON.parse(localStorage.getItem('token')).username;

var identityCardImagePath = '';

var aadharCardImagePath = '';

var \_URL = window.URL || window.webkitURL;

$("#fupImageIdentityCard").on('change', function () {

var file, img;

if ((file = this.files[0])) {

img = new Image();

img.onload = function () {

sendFileIdentityCard(file);

};

img.onerror = function () {

alert("Not a valid file:" + file.type);

};

img.src = \_URL.createObjectURL(file);

}

});

**Chapter 6:- Testing**

**Introduction**

Software testing is a critical element of the ultimate review of specification design and coding. Testing of software leads to the uncovering of errors in the software functional and performance requirements are met. Testing also provides a good indication of software reliability and software quality as a whole. The result of different phases of testing are evaluated and then compared with the expected results. If the errors are uncovered they are debugged and corrected. A strategy approach to software testing has the generic characteristics:

Testing begins at the module level and works “outwards” towards the integration of the entire computer based system.

Different testing techniques are appropriate at different points of time.

Testing and debugging are different activities, but debugging must be accommodating in the testing strategy.

**Goals & Objectives:**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Our objective is to design test processes that systematically uncover different classes of errors and do so with minimum amount of time and effort.

**Statement of Scope:**

A description of the scope of the software testing is developed. All the features to be tested are noted as follows. The basic principles that guides software testing are:

All test cases should be traceable top customer requirements. The most severe defects from the customer’s point of view are those that cause the program to fail to meet its requirements.

Test case should be planned long before testing begins. Testing plan can begin as soon as the requirement model is complete. Detailed definition of the test cases can begin as soon as the design is solidified. Therefore, the entire test can be planned before any code has been generated.

Testing should begin “in the small” and progress towards “in the large”. The first test planned and executed generally focus on the individual modules. As testing progresses testing shifts focus in an attempt to find errors in integrating clusters of modules and ultimately in the entire system.

**Testing Principles:**

The basic principles that guide software testing are:

All the cases should be traceable top customer requirements. The most severe defects from the customer’s point of view are those that cause the program to fail to meet its requirements.

Test case should be planned long before testing begins. Testing plan can begin as soon as the requirement model is complete. Detailed definition of the test cases can begin as soon as the design is solidified. Therefore, all the test can be planned before code has been generated.

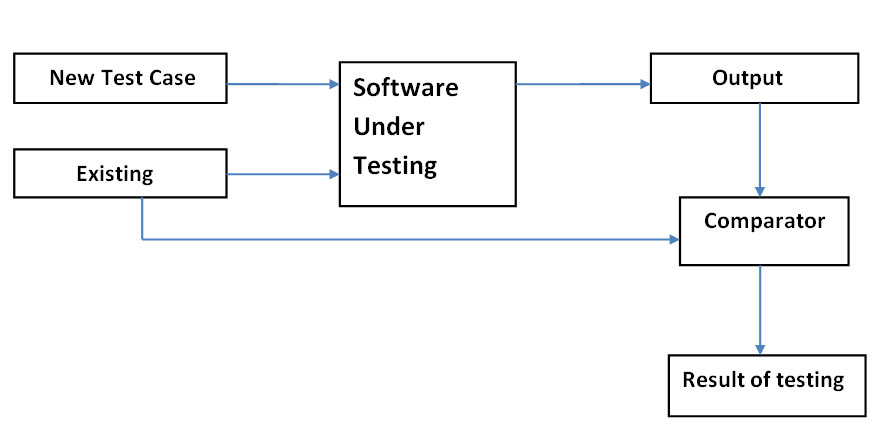
The Pareto Principle applies to software testing. Stated simply the Pareto Principle implies that 80% of all errors uncovered during testing will likely to be traceable to 20% of all program modules. The program of course is to isolate these suspect modules and to thoroughly test them.

Testing should begin “in the small” and “in the large”. The first test planned

and executed generally focus on the individual modules. As testing progresses testing shifts focus in an attempt to find errors in integration clusters of modules and ultimately in the entire system.

Exhaustive testing is not possible. The number of paths permutations for even a moderately sized program is exceptionally large. For this reason, it is impossible to execute every combination of path during testing. It is possible however to ensure that all conditions in the procedural design have been exercised. To be most effective an independent third party should conduct testing. The third party has the highest probability of finding the errors.

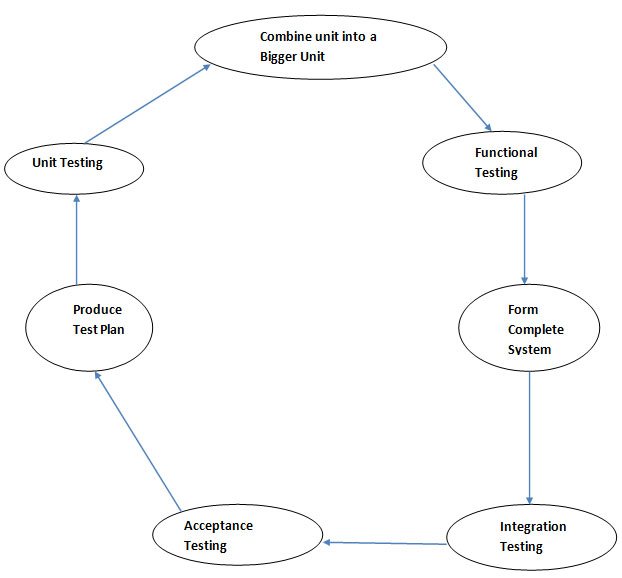
Test case: Before the project is released, it has to pass through a test cases suit, so that the required functionality is met and previous functionality of the system is also not broken to do this, there is an existing test cases which checks the previous functionality. New test cases are prepared and added to this existing test suit to check for the added functionality.

 A pictorial representation of this can be shown as follows:

**Figure 8. Test Case Suit**

**Testing Process:**

The testing process can be shown as:

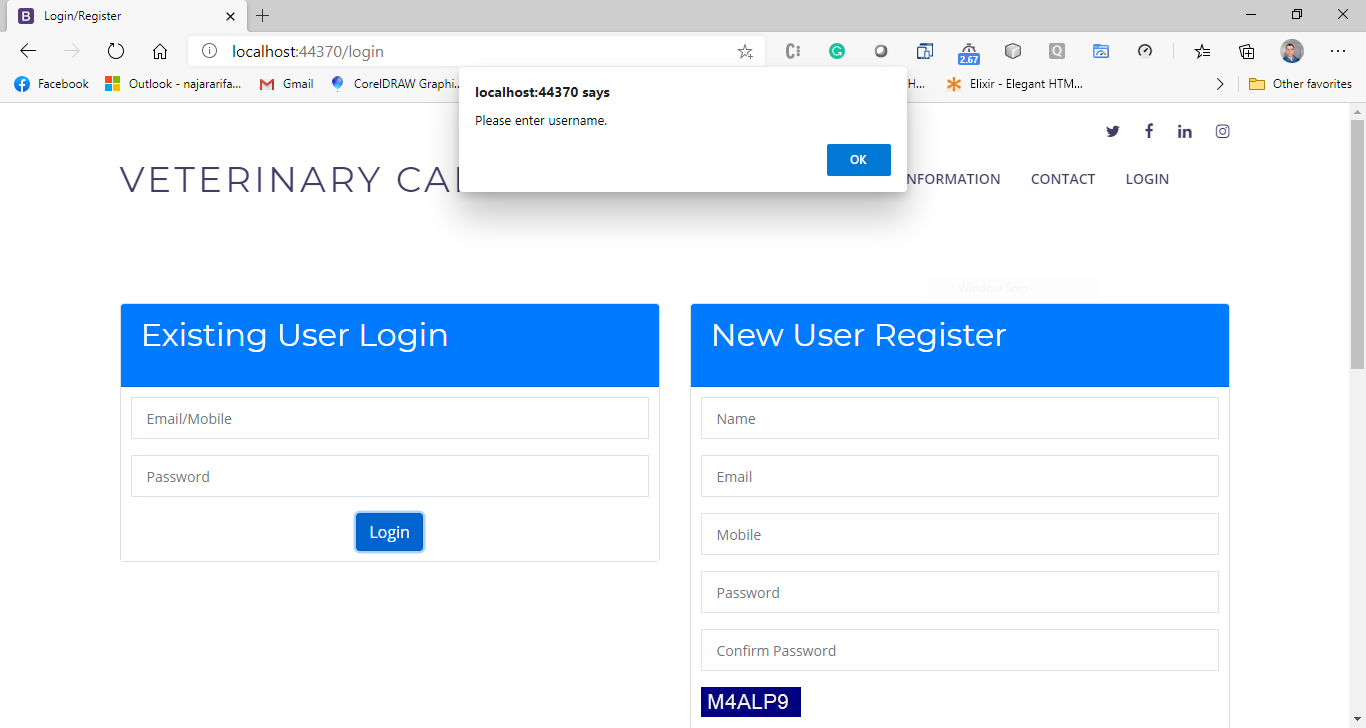


**Figure 9. Testing Process**

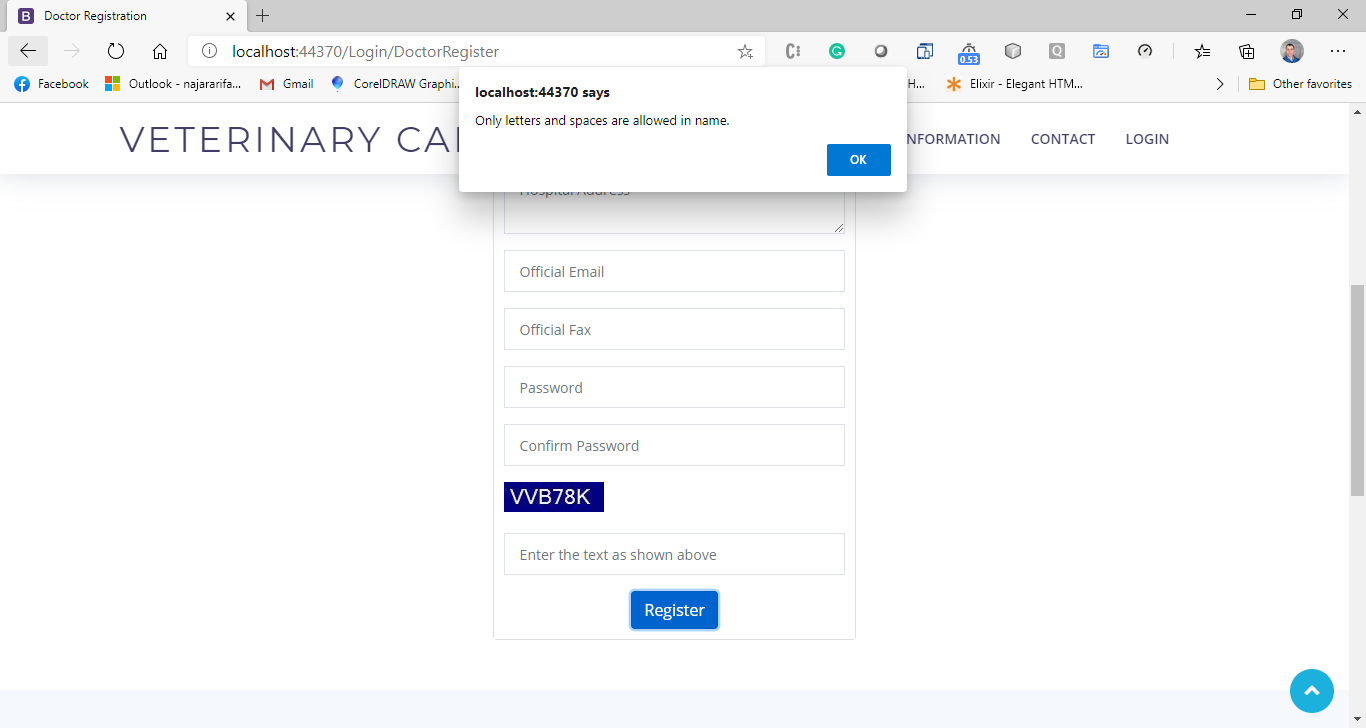
**6.1 Testing Approaches Used in Project:**

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The unit testing is normally considered as an adjunct step to coding step. Because modules are not a standalone program, drivers and/or stubs software must be developed for each unit. A driver is nothing more than a “main program” that accepts test cases data and passes it to the module. A stub serves to replace the modules that subordinate to the modules to be tested. A stub may do minimal data manipulation, prints verification of entry and returns.

**Black Box Testing:** Black Box Testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This type of testing was performed on our application to see if all the parts of the application were working as expected and giving the expected results. Upon performing this testing, our application gave the expected results in all the functional units.



**Figure 10. Login Validation**



**Figure 11. Doctor Registration Validation**

When we kept the username and password boxes empty and clicked on Login button, it gave the prompt as above.

**Functional Test:** Each part of the code was tested individually and the pages were tested individually on all platforms to see if they are working properly.

**Performance Test:** These determines the amount of execution time spent on various parts of units and the resulting throughput, response time given by the module. Our project takes less time in executing any action. It also works on slow networks like 2g.

**Stress Test:** A lot of test files were made to work at the same time in order to check how much workloads can the unit bear.

**Structure Test:** These tests were made to check the internal logic of the program and traversing particular execution paths. The application behaved correctly and gave the correct output in all aspects.

**Validation Test:** This test was done on Login/Register pages, Book Appointment page, Doctor Registration page etc. The first validation test was to ensure that the user does not left the text box blank. Another test was for proper email. If the user entered incorrect email, it gave the error message. Similarly, the mobile no validation test was also done.

**Modular Integration Testing:** Modular integration testing is done to ensure that the module is working independently. The inputs as required by the module are given as required and the output is tested as per the specifications. All the three modules viz., Admin, Doctor and User were tested individually and they worked correctly.

**6.2. Test Cases**

A **TEST CASE** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, post condition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer. The following table depicts the test cases which we have executed on our application.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S. No | Test Case Name | Description | Test Case Date | Input | Output | Results | Debugging |
| 1 | Login | Login using mobile/email and password | 10-10-2020 | Email/Mobile, Password | If email/mobile and/or password incorrect, displays incorrect username/password | If successful, redirects to appropriate module. |  |
| 2 | Register | Register as a user | 10-10-2020 | Name, Email, Mobile, Password, Confirm Password | If registration successful, displays the message “Registration done successfully” | If registration unsuccessful, displays message “Error occurred. Please try again.” |  |
| 3 | Doctor Register | Register as a doctor | 11-10-2020 | Name, Email, Mobile, Address, Official Email, Hospital Address | If registration successful, displays message “Registration done successfully” | If error occurs, displays message “Error occurred. Please try again…” |  |
| 4 | Approve/Reject Doctor | Admin approve/rejects a doctor | 11-10-2020 | List of doctors, button against each doctor | If action was successful, displays appropriate message | If approved, doctor can login and add booking slots. |  |

**Table 1. Test Cases used in our application**

**Chapter 7. IMPLEMENTATION OF PROJECT**

**Introduction to ASP.NET:**

ASP.NET is a framework for developing Web Applications using either C# or VISUAL Basic as the backend programming language .it was developed by Microsoft when they developed the .NET framework in 2003 .it uses the concept if web forms which means that the web pages are designed in a way that the developer who has already developed Windows application with windows form can easily develop Web Application with Web Forms provided by ASP.NET.

**Introduction to C#:**

C# is a high level programming language which was developed by Microsoft in early 1990’s .it uses OOPS concept like abstraction, encapsulation, overriding, overloading etc. this application is largely used in web development for the application written in ASP.NET .it is also used to develop Windows Application using either the windows forms or the Windows Presentation Technology .it can also be used to create services, API’S etc.

**Introduction to Visual Studio:**

It is an integrated Development Environment which can be used to develop Windows Applications, Web Applications, and Mobile Applications.

**Software Requirements**

**Operating System:** Windows 7/8/10.

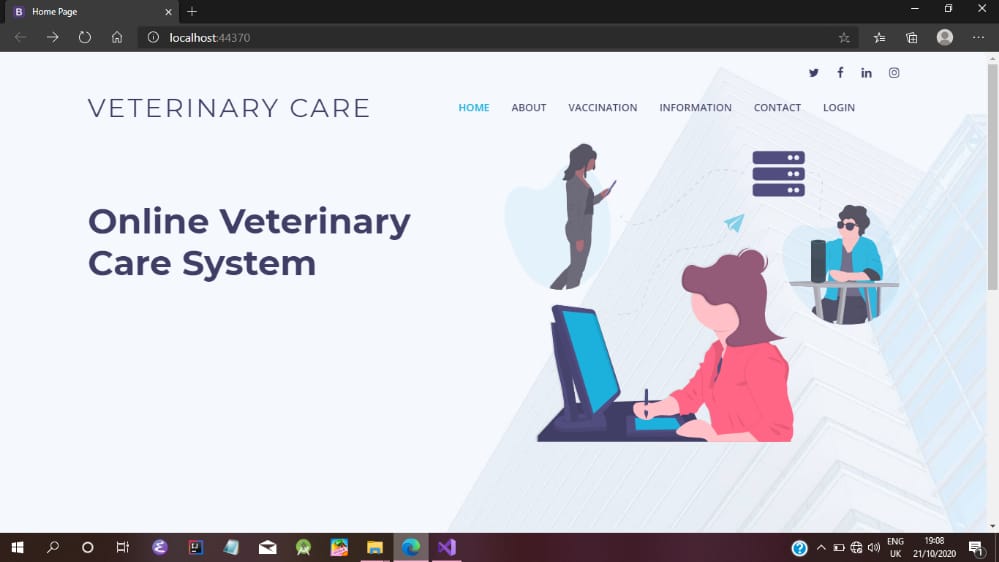
**Tools:** SQL Server 2012 or Later. Microsoft Visual Studio 2019

**Hardware Requirements:** Intel Core i3 or higher.

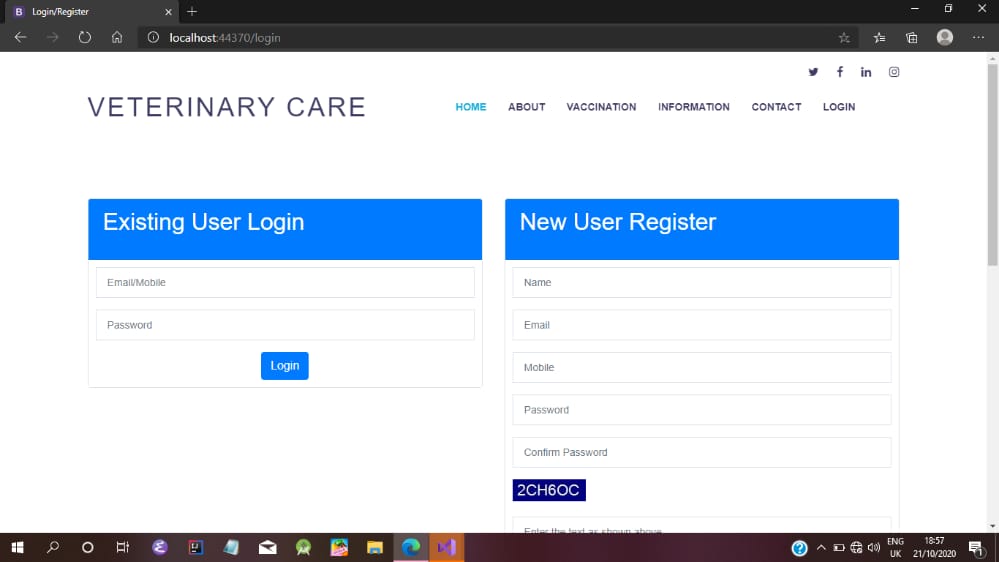
8GB of RAM or higher.

100GB of Hard Disk Space or higher in case there is too much data stored in database.

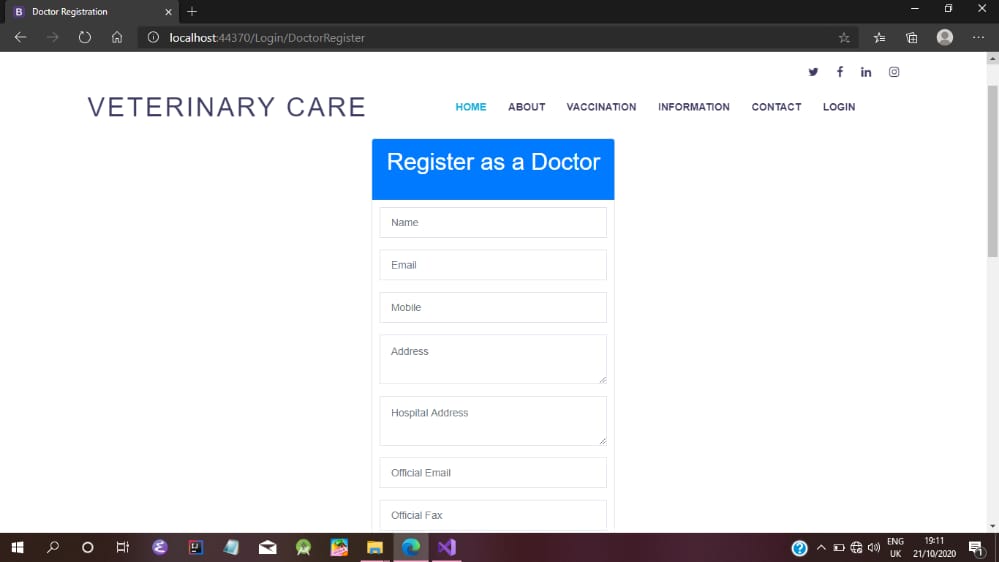
**Chapter 8. SNAPSHOTS**



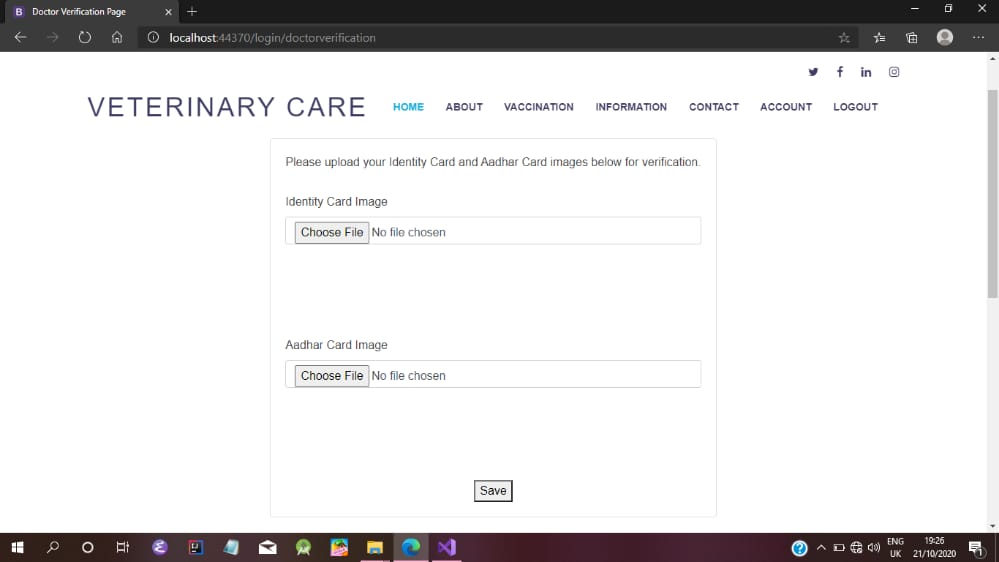
**Figure 12. Home Page Screenshot**



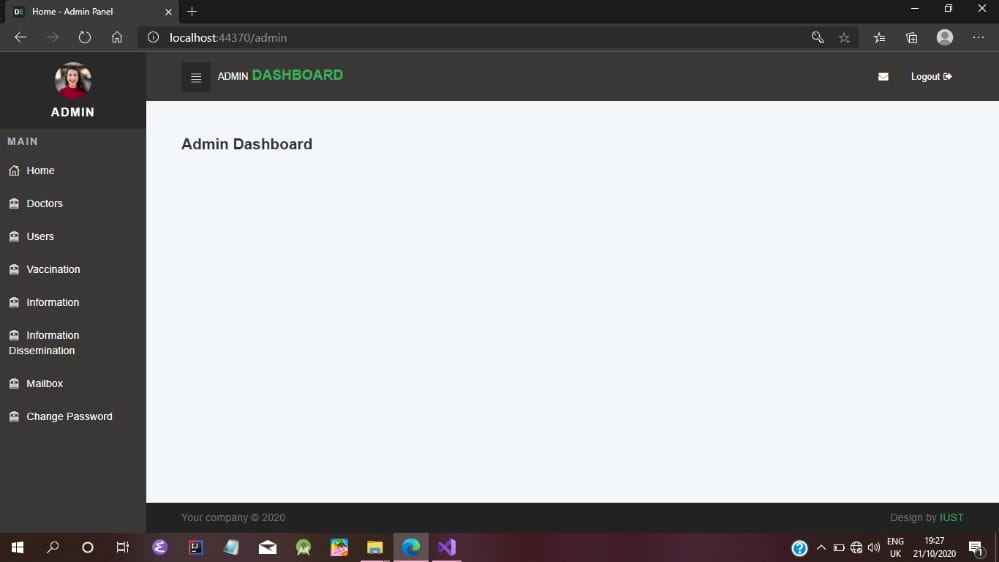
**Figure 13. Login/ Register Page Screenshot**



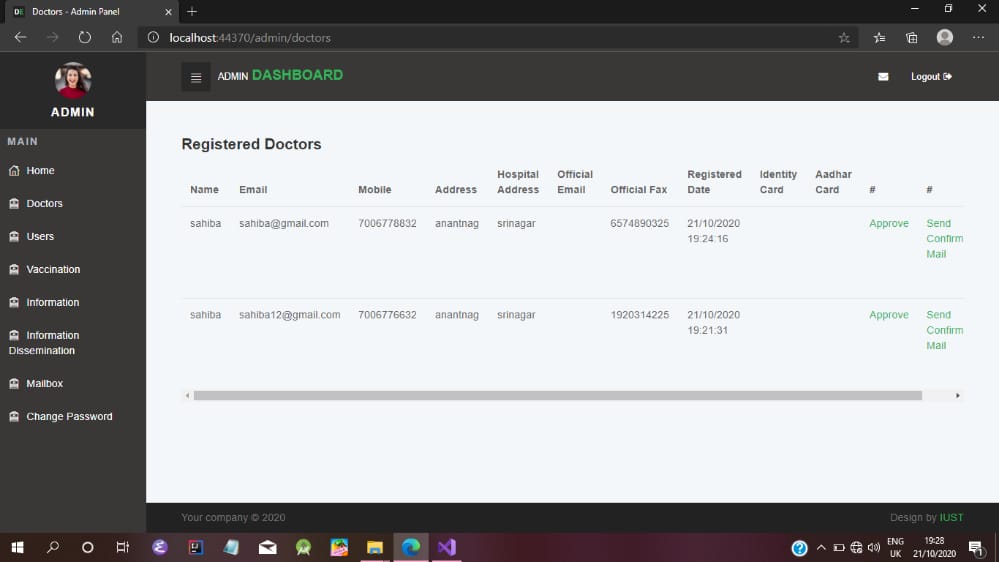
**Figure 14. Doctor Registration Page**



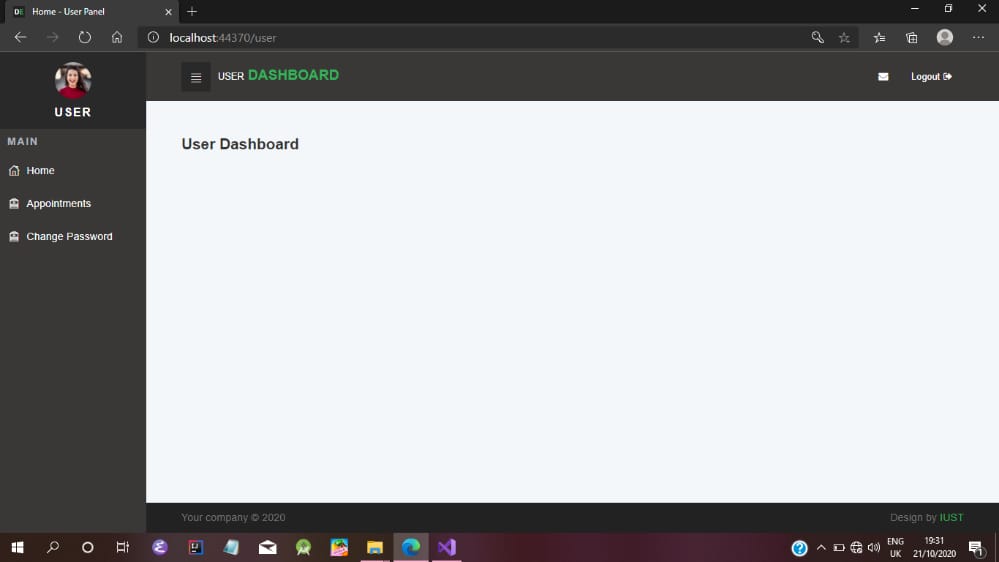
**Figure 15. Doctor Verification Page**



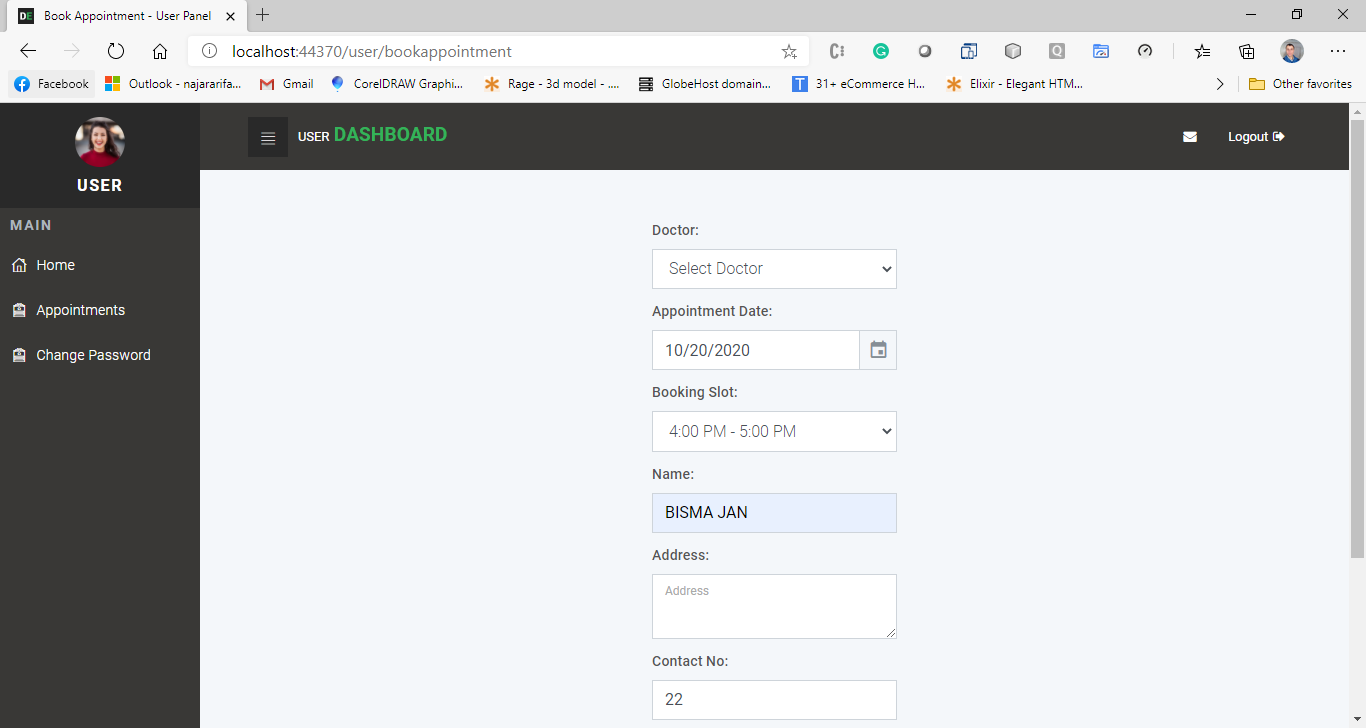
**Figure 16. Admin Dashboard**



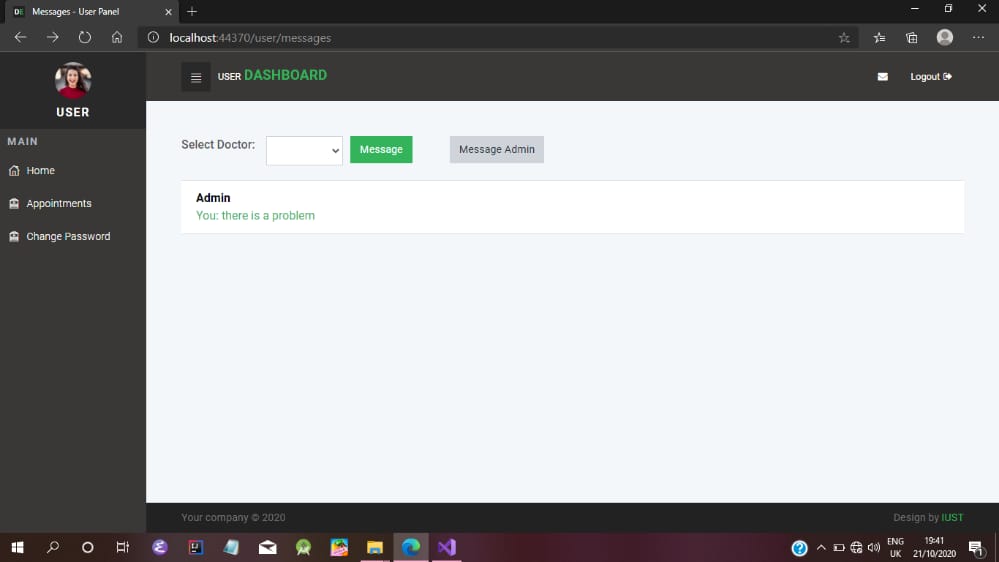
**Figure 17. Registered Doctors as seen by Admin**

****

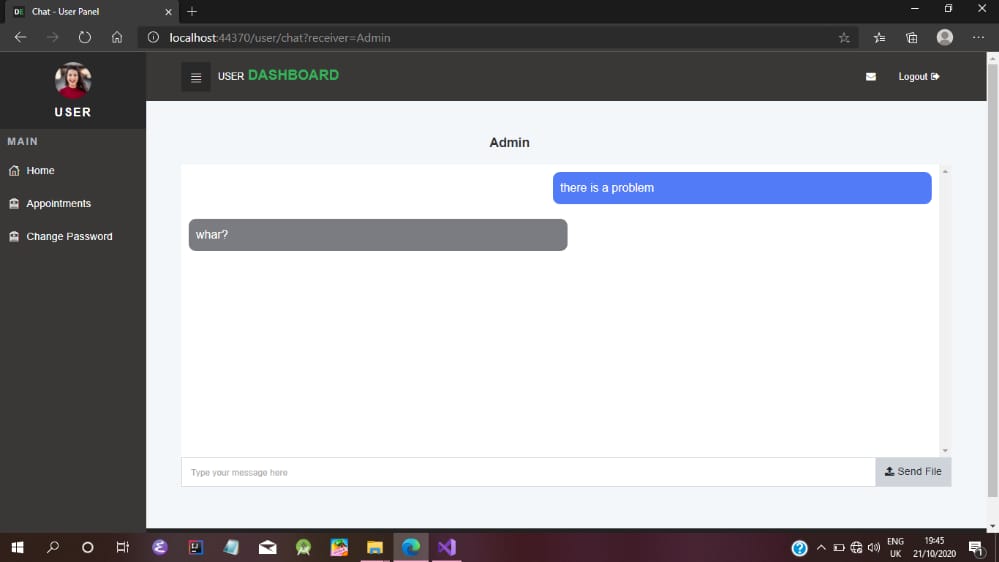
**Figure 18. User dashboard**



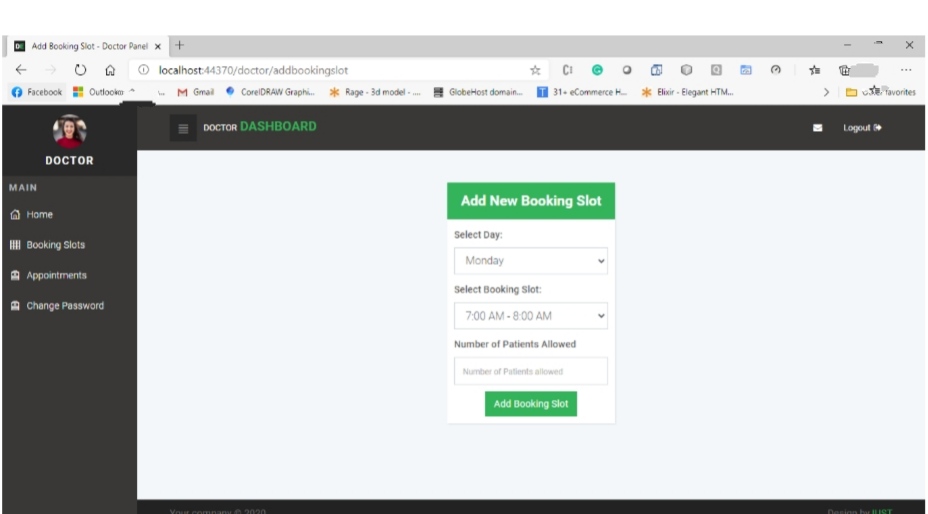
**Figure 19. Book Appointment Page**



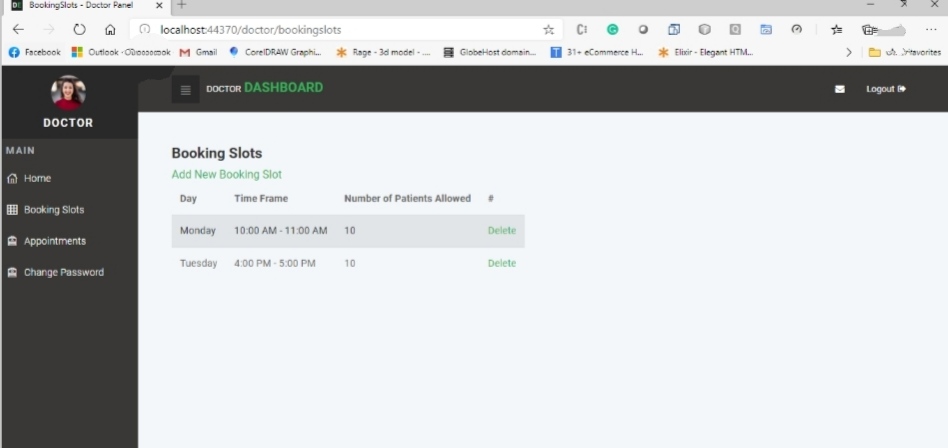
**Figure 20.** **Messages Page**



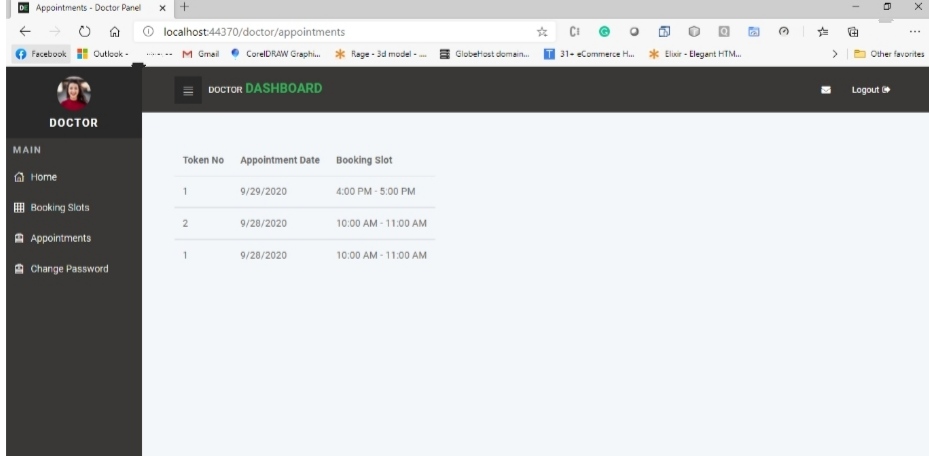
**Figure 21. Chat Page**



**Figure 22. Add Booking Slot Page**



**Figure 23. View Booking Slots Page**



**Figure 24. Doctor View Appointments Page**

**Chapter 9. CONCLUSION**

The main purpose of this project is:

* To enable the user to book the appointments with the veterinary doctors to cure their pets.
* To help users to chat with doctors instantly in case of emergency.
* To help users to get enough info regarding vaccination schedule, camps to be organised etc.

There are many websites like petcare.com, vetcoach.live available for the treatment of pets but these websites have problem that users do not get instant feedback for the treatment of their pets in case of emergency. They also have very less number of appointments available. Also in our province, none of these can be used because they are not available here. So we decided to design an application for our own province so that users can book appointments online using our application and can also chat with doctors/ admin to get the feedback instantly.

**Chapter 10. FUTURE WORK/RECOMMENDATIONS**

In future we wish to add some more features like enabling the doctor to make his/her booking slots inactive if the doctor is not available. User will also be able to cancel or reschedule appointments. Chat system will also be made good.

**Chapter 11. Appendices**

**Appendix A: Goals of the system**

The main goals of this system are to provide users with an easy to use system to contact doctors to cure their pets either through booking appointments or chatting with the doctors. In case there is some serious problem, the user can contact with the admin also. It will allow only authentic doctors and users to enter the system. Users will get very much information about curing their pets rather than searching for a doctor offline.

**Appendix B: Technologies used in the system**

This system uses ASP.NET MVC and RESTful Web Api’s provided by ASP.NET MVC architecture to achieve the goals described above. It uses C# as the server side language and Microsoft SQL Server as the DBMS for database.

**Appendix C: Limitations of the system**

This application has the limitation that the user cannot rebook or cancel an appointment which will be addressed in the future releases. Any person can also access the register api with the help of POSTMAN or any other REST api's tool to register the user without the verification. This issue can be resolved by using an OTP provided in the SMS to the user’s mobile and that will also be addressed in future releases.

**Chapter 12. Bibliography**

* <https://smartbear.com/learn/automated-testing/software-testing-methodologies/>
* <https://www.guru99.com/test-case.html>
* <https://www.tutorialspoint.com/software_testing_dictionary/software_requirement_specification.htm>
* <https://www.geeksforgeeks.org/analysis-modelling-in-software-engineering/#:~:text=Analysis%20Model%20is%20a%20technical,design%20in%20the%20design%20modeling>.

**Books Used:**

C# in Depth

Wrox C#

Beginning ASP.NET 3.5: Learn ASP.NET Step by Step