

clini-C-are – A CLINICAL AUTOMATION SYSTEM

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

Tanu Dhoot

(EN16CS301277)

Under the Guidance of
Prof. Sachin Solanki
Mr. Manjunath Awathi



**Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331**

May, 2020

clini-C-are – A CLINICAL AUTOMATION SYSTEM

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

Tanu Dhoot

(EN16CS301277)

Under the Guidance of
Prof. Sachin Solanki
Mr. Manjunath Awathi



**Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331**

May, 2020

Report Approval

The project work “**clini-C-care – A Clinical Automation System**” is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation:

Affiliation:

External Examiner

Name:

Designation:

Affiliation:

Declaration

I hereby declare that the project entitled “**clini-C-care – A Clinical Automation System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science and Engineering’ completed under the supervision of **Prof. Sachin Solanki, Assistant Professor, Department of Computer Science and Engineering**, Faculty of Engineering, Medi-Caps University, Indore, and **Mr. Manjunath Awathi, Corporate Trainer and Software Consultant, IBM** is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Date: _____

Tanu Dhoot (EN16CS301277)

Certificate

We, **Prof. Sachin Solanki** and **Mr. Manjunath Awathi** certify that the project entitled “**clini-C-are – A Clinical Automation System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Tanu Dhoot** is the record carried out by her under our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Prof. Sachin Solanki
Computer Science & Engineering Department
Medi-Caps University, Indore

Mr. Manjunath Awathi
Corporate Trainer and Software Consultant
IBM India Pvt. Ltd.

Dr. Suresh Jain
Head of the Department
Computer Science & Engineering
Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somanı**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda**, Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Prof. (Dr.) Suresh Jain** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my External Guide, **Mr. Manjunath Awathi**, Corporate Trainer and Software Consultant, IBM India Pvt. Ltd. as well as to my Internal Guide, **Prof. Sachin Solanki**, Assistant Professor, Department of Computer Science and Engineering, Medi-Caps University, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Tanu Dhoot (EN16CS301277)

B.Tech. IV Year

Department of Computer Science & Engineering
Faculty of Engineering
Medi-Caps University, Indore

Abstract

‘clini-C-are’ – A Clinical Automation System is a project developed as a part of training with International Business Machines Corporation (IBM). This training was a process included in the Pre-Joining Program of the organization.

Medical care is an important aspect of the global world. With the increase in the number of diseases, and almost a discovery of an entirely new disease every year, the world has seen a huge increase in the number of patients. The data for each and every patient must be recorded and the manual application of the same proves to be quite challenging. Thus, it is important for the medical care industry to incline towards and rely fairly on digital devices and applications.

Automation refers to the state of being automatically operated and is the step beyond mechanization where it requires physical labour to perform the functions manually. The automation system replaces the conventional hard-wired system by automating the process functions with the use of various software tools with minimal human intervention.

With the evident increase in the number of patients and business, it becomes fairly difficult for all the providers of healthcare facilities to deliver excellent services with manual work. The services mentioned can range from the information on the drugs prescribed, the consultations with doctors, to the provision and delivery along with the ordering of drugs.

‘clini-C-are’ – A Clinical Automation System is a web based application that provides an important customer interface to the clinic operations. It provides an easy interaction between the patients and the clinic as well as all the areas of any clinic. It employs the .NET Framework to provide an interactive application, easy on the users, but tough on the workings. The working of the medicine sector can thus be greatly improved.

Keywords: Medical Care, Clinics, Patient Surge, Business Expansion, Digitalization, Automation System.

Table of Contents

		Page No.
	Report Approval	ii
	Declaration	iii
	Certificate	iv
	Acknowledgement	v
	Abstract	vi
	Table of Contents	vii
	List of figures	ix
	List of tables	xi
	Abbreviations	xii
Chapter 1	Introduction	
	1.1 Introduction	1
	1.2 Literature Review	3
	1.2.1 Impact of CAS on Healthcare Industry	3
	1.2.2 Clinical Benefits	4
	1.2.3 Workflow Benefits	4
	1.2.4 Administration Benefits	5
	1.2.5 Revenue Enhancement	6
	1.3 Objectives	6
	1.4 Significance	8
	1.5 Problem in existing system	9
	1.6 Organization	10
	1.6.1 Introduction to IBM	10
	1.6.2 About Training	11
Chapter 2	System Requirement Analysis	
	2.1 Information Gathering	12
	2.1.1 Functional Requirements	12
	2.1.2 Non-Functional Requirements	13
	2.2 System Feasibility	14
	2.2.1 Economical	15
	2.2.2 Technical	15
	2.2.3 Behavioral	16
	2.3 Platform Specification (Development & Deployment)	16
	2.3.1 Hardware	16
	2.3.2 Software Implementation Language/ Technology	16
Chapter 3	System Analysis	
	3.1 Information Flow Representation	18
	3.1.1 Entity-Relationship Diagram	18
	3.1.2 Data Dictionary	20
	3.1.3 Data Flow Diagram	21
	3.1.4 Class Diagram	21

Chapter 4	Design	
	4.1 Architectural Design	23
	4.1.1 Architectural Context Diagram	24
	4.1.2 Architectural Behavioral Diagram	24
	4.1.3 Description of Architectural Diagram	25
	4.2 Procedural/Modular Approach	26
	4.2.1 Modules Used	26
	4.2.2 Internal Data Structures	30
	4.3 Data Design	31
	4.3.1 Data objects and Resultant Data Structures	36
	4.4 Interface Design	37
	4.4.1 Human-Machine Interface Design Specification	37
	4.4.2 I/O Forms	40
	4.5 Reports	49
Chapter 5	Testing	
	5.1 Testing Objective	52
	5.2 Testing Scope	53
	5.3 Testing Principles	53
	5.3.1 Seven Principles in Testing	54
	5.4 Testing Methods Used	55
	5.4.1 Unit Testing	55
	5.4.2 Validation Testing	55
	5.5 Test Cases	56
	5.6 Sample Test Data & Results	62
Chapter 6	Limitations	64
Chapter 7	Future Scope	65
Chapter 8	Learning After Training	
	8.1 Learning During Training	66
	8.2 Learning After Training	66
Chapter 9	Conclusion	68
Chapter 10	Bibliography and References	69

List of Figures

Figure No.	Figure Name	Page No.
3.1	E-R Diagram of CAS	19
3.2	Data Dictionary of CAS	20
3.3	Data Flow Diagram of CAS	21
3.4	Class Diagram of CAS	22
4.1	Use Case Diagram of CAS	25
4.2	Models in CAS	29
4.3	Controllers in CAS	29
4.4	Views in CAS	30
4.5	Home Page of clini-C-are	38
4.6	User Dashboard of Admin in clini-C-are	38
4.7	User Dashboard of Patient in clini-C-are	38
4.8	User Dashboard of Physician in clini-C-are	39
4.9	User Dashboard of Salesperson in clini-C-are	39
4.10	User Dashboard of Supplier in clini-C-are	39
4.11	Login Page in clini-C-are	40
4.12	Request Admin Page in clini-C-are	41
4.13	Registration Page in Admin Module in clini-C-are	41
4.14	View Physician Page in Admin Module in clini-C-are	42
4.15	View Drugs Page in Admin Module in clini-C-are	42
4.16	Request Appointment Page in Patient Module in clini-C-are	43
4.17	View Inbox Page in Patient Module in clini-C-are	43
4.18	Order Drugs Page in Patient Module in clini-C-are	43
4.19	View Patients' Appointments in Physician Module in clini-C-are	44
4.20	Update Patients' History in Physician Module in clini-C-are	44
4.21	View Salesperson's Messages in Physician Module in clini-C-are	44
4.22	Add Drugs Page in Salesperson Module in clini-C-are	45
4.23	View Drugs Page in Salesperson Module in clini-C-are	45
4.24	Place Orders Page in Salesperson Module in clini-C-are	46
4.25	View Patients' Orders Page in Salesperson Module in clini-C-are	46
4.26	View Orders Page in Supplier Module in clini-C-are	47
4.27	View Messages Page in Supplier Module in clini-C-are	47
4.28	Edit Personal Details Page of Physician in clini-C-are	48
4.29	Change Password Page in clini-C-are	48
4.30	Home Controller of clini-C-are	50
4.31	Admin Controller of clini-C-are	50
4.32	Patient Controller of clini-C-are	50
4.33	Physician Controller of clini-C-are	51
4.34	Salesperson Controller of clini-C-are	51
4.35	Supplier Controller of clini-C-are	51

5.1	Login Failed	62
5.2	Verification in Registration Page	62
5.3	User Successfully Registered	63
5.4	Date Check Constraint on Manufacturing and Expiry Date of Drugs	63
5.5	Message Successfully Sent	63

List of Tables

Table No.	Table Name	Page No.
4.1	User Table in CAS Database	31
4.2	Role Details Table in CAS Database	31
4.3	Specialization Data Table in CAS Database	32
4.4	Admin Table in CAS Database	32
4.5	Patient Table in CAS Database	32
4.6	Physician Table in CAS Database	33
4.7	Supplier Table in CAS Database	33
4.8	Salesperson Table in CAS Database	33
4.9	Appointment Table in CAS Database	34
4.10	Drug Table in CAS Database	34
4.11	Patient Order Details Table in CAS Database	35
4.12	Salesperson Order Details Table in CAS Database	35
4.13	Drug Delivery Table in CAS Database	35
4.14	Inbox Table in CAS Database	36
4.15	Request Admin Table in CAS Database	36
5.1	Testing in Home Controller	56
5.2	Testing in Admin Controller	56
5.3	Testing in Patient Controller	58
5.4	Testing in Physician Controller	59
5.5	Testing in Salesperson Controller	60
5.6	Testing in Supplier Controller	61
8.1	Weekly Schedule	66

Abbreviations

Abbreviation	Description
CAS	Clinical Automation System
GDP	Gross Domestic Product
IT	Information Technology
EMR	Electronic Medical Record
SQL	Structured Query Language
MVC	Model-View-Controller
LINQ	Language Integrated Query
WPF	Windows Presentation Foundation
WCF	Windows Communication Foundation
GUI	Graphical User Interface
IFD	Information Flow Diagram
UML	Unified Modelling Language
AJAX	Asynchronous JavaScript and XML
XML	Extensible Markup Language
API	Application Programming Interface
DOM	Document Object Model

Chapter - 1

Introduction

1.1 Introduction

The dictionary defines automation as “the technique of making an apparatus, a process, or a system operate automatically.” In scientific terms, automation is defined as the technology by which a process or procedure is performed with minimal human assistance. The Automation System side lines the conventional manual labour work by automating the business functionalities using software tools.

The Medicinal Area has seen a large growth due to the rapid increase in the number of regular diseases, as well as the discovery of unknown diseases which are quite lethal. While this is a bad news for the global world, it is also a bad news for the medical business as many of them still dwell on the traditional ways of papers and books. Currently, there are a huge number of private and public hospitals that store the records manually using physical case studies. It's a very formal way of storing data and records but the demerit of storing data in this manner is that poor handling of record books can lead to loss of data, and there is no provision of backup.

In these Medical Institutions, paper is mostly used for workflows. The patient registration is done using enquiry form, records are made using files, diagnosis is recorded on simple clinical papers, bill statements are opened using a book, and appointments are recorded in a register. For any clinical authority to find any information, they are required to go through piles of files to get to any particular information which may or may not be quite small. This utilizes a lot of time and energy, thus making the job inefficient; and also provides a lot of trouble for the patient if they are direly ill.

This can be solved using a computer based system to manage the Business Process. Michael Hammer and James Champy define a Business Process as a collection of activities that take one or more kinds of input and create an output that is a value to the customer. Once a system can manage the business process, a lot of work can be automated and no manual operation is needed.

Clinical Automation System (CAS) can be easily defined as a business process. The use of CAS can enhance the services and the work flow of all the activities that happen in the clinics; where it helps in reducing the workload of medical staff, the number of man power needed and making it more manageable and easier to control.

The Clinical System is an important sector in the global healthcare. Any person with the most common to the rarest disease visits a medical facility to get checked on their health. It is an obligation as well as a necessity to take care of patients in every way possible. The health of its citizens defines the health and growth of a nation thus making it an important contributing factor to the country's GDP and global economy.

The private and public Medical Institutions across India exercise the traditional way of storing data on clinical signed papers. Every patient has to maintain heaps of files as a record for their medicinal history, along with the various bills provided by the institution. They also might have to wait in long queues for an appointment as many of them require that appointments are to be taken in person or over the phone calls. While all the institutions have to maintain the patient records on case files, some renowned institutions use a digital way of storing bills.

In this technological world which is moving each day towards more and more sophisticated technology, esp. achieving its recognition as an "IT world", introducing IT in Medical Industry can be very fruitful. Our country is a developing country moving towards the era of modernization and technology but has not achieved much.

The project "clini-C-are" helps in tying up the medical institutions and the patients together. Prior to this system a lot of processes we handled manually and consumed a lot of time and resources. Based on the requirements this system has been designed to enable users to have easy access to the common information about the clinic. Using this, users will now be able to search for more information on the prescribed drugs. Patients will be able to interact with their physicians directly via an inbuilt messaging system. The system also has the provision for physicians and salespersons to order for more drugs directly with the suppliers. Suppliers in turn can view the orders placed in the system and approve it as well.^[1]

In short, it supports a part of the information infrastructure and workflow management to support the day to day activities of a clinic. Based on the various surveys conducted prior to using this system, it will now allow a majority of information to be available for patients online thus reducing the intake of telephonic queries at the clinic. It also forms a common platform for Physicians, Patients, Suppliers and Salesperson to perform their tasks efficiently and faster.

The project “clini-C-are” could help in improving the medical sector of India. And along with that, this project as a part of the training helped us to use knowledge we have learned practically and learn new techniques and trends to develop an application and help us in the academic side too. So, “clini-C-are” basically aims in fulfilling both academic and social responsibility and help both patients and medical authorities.

1.2 Literature Review

A wide-ranging literature review of CAS implementation over the past decade reveals that clinical, workflow, administrative, and revenue enhancement benefits of the CAS outweigh barriers and challenges — but only if healthcare organizations redesign certain work processes. Among other key efforts, organizations must train and motivate users to navigate CAS systems, as well as develop a common structured language. Clinicians who used CASS found that electronic access to clinical information saves time and provides a thorough and efficient way to manage patient information.

1.2.1 Impact of CAS on Healthcare Industry

In 1991, the IOM published a report recommending the implementation of the CAS by 2001 to improve the care of patients and to reduce waste. The Computer-based Patient Record Institute (CPRI) stated, if providers continue with their current paper systems, they will lack the tools needed to manage the quality and costs of healthcare, the scientific basis for healthcare will continue to be undermined, and healthcare reform will be impeded.

Few organizations have published studies describing the actual costs and benefits attained from implementing CASSs. The benefits associated with CASSs are organized into four categories: clinical, workflow, administrative, and revenue enhancement. Renner states that measuring all the benefits associated with CPRs is virtually impossible, and that it is probably safe to select those that can make the greatest financial difference, and incorporate them into a financial model.

1.2.2 Clinical Benefits

Clinical benefits seen after implementing a CAS include: better access to the chart, improved clinical decision making and disease management, enhanced documentation, simplified patient education, and increased free time to spend with patients, accompanied by improved perception of care and quality of work life. These benefits ultimately result in better delivery of patient care.

Khoury and Tierney et al suggested that the EMR might be the only practical way to apply practice guidelines while clinicians are delivering care to patients. Khoury cited three instances where computerized reminders resulted in improvements in quality of care.^[4]

Kaiser Permanente of Ohio saw these best practice guidelines compliance improvements after implementing a medical automated record system and adding reminders at the point of care:

- Aspirin use in patients with coronary artery disease increased from 56 percent to 82 percent in 27 months, while lipid-lowering agents increased from 10 percent to 20 percent in 7 months.
- ACE inhibitor use in patients with congestive heart failure increased from 54 percent to 66 percent.
- Percentage of hypertensive patients taking non-recommended medications decreased from 16 percent to 12 percent in 12 months.
- Percentage of patients older than 64 years of age who were offered an influenza vaccination during a primary care visit increased from 56 percent to 69 percent in 36 months.

1.2.3 Workflow Benefits

Workflow solutions save several people from doing the same work those others have already done, or save them from postponing their own tasks while others finish theirs. With a CAS, many people can make changes to the record from different locations in parallel. The data gathered at the initial encounter is carried over to each subsequent visit within the integrated delivery system, eliminating the need to re-gather information.

Workflow benefits resulting from CAS implementation include improved data intake, reduced transcription costs, reduced labour costs, and improved communication and better management of referrals, lab results, prescriptions, and drug recalls.

The first workflow benefit was improved data intake. After implementing the CAS, patients were no longer frustrated by having to record the same information again and again on paper forms in different office settings. The CAS had all of that information readily available for providers. Dassenko and Slowinski reported a reduction in nurse intake time from 35 minutes to 20 minutes for initial office visits and from 35 minutes to 15 minutes for return visits at University of Wisconsin Hospital and Clinics.

Second, when clinicians completed their documentation at the point of care, there was no need for transcription at the end of the day. Mildon and Cohen valued this as an estimated savings of \$300 to \$1,000 per month, per physician.

Third, CASs reduces the need for clerical staff by reducing staff workload and improving work processes. Each user's workstation becomes a chart room, which gives staff access to the charts electronically instead of pulling and re-filing them manually. Basch reported adding several thousand more patients per year without additional staffing, eliminating overtime hours for medical assistants, and eliminating one full-time employee in medical records.

Fourth, CASs have a built-in e-mail system, which results in improved communication by allowing staff the ability to message each other from any workstation. The CAS coordinated workflow communication in the business office and patient care departments.^[4]

1.2.4 Administrative Benefits

Administrative benefits after CAS implementation include objective monitoring of physician practices, outcomes research, and disease management. CAS implementation also resulted in easier means to create report cards for managed care plans, simplified and improved claims processing, and better customer service. The structured data in the CAS enabled administrators to generate forms and graphs that show the data in an organized format.

Another claims processing advantage cited by Dassenko and Slowinski and MedicaLogic was improved cash flow as a result of more complete billing. With the CAS, account representatives were able to view an organized chart and search the entire chart by a key word.

Finally, Dassenko and Slowinski found that customer service was enhanced because patient accounting representatives were more available for questions and because questions were answered more completely and easily with online access to patient files. Most patient calls are very basic, and do not require the expertise of a patient accounting representative. If the information is organized and available online, other staff can handle these questions while patient accounting representatives can handle the more complicated problems.^[4]

1.2.5 Revenue Enhancement

Mildon and Cohen stated, “There is more to money than saving it. CASs also can enhance revenue.” CASs increase revenue through effective management of information at the point of care and in the billing office. Revenue enhancement depends on factors such as more effective health maintenance programs, better accuracy of coding and improved administrative and workflow functions.^[13]

1.3 Objectives

“clini-C-are” – A Clinical Automation System follows the objective of fully automating the daily operations of a clinic by leading the clinic to operate in effective and efficient way throughout the help of an automation system so that they could eliminate the paper work that originally happened in a clinic.

Specifically, the aim is to fulfil the following:

- Shrink the space and resources needed by the clinic**

Originally clinics record their patient information in a paper and for frequently visiting patients, more papers will be added in to the patient records so that the physicians have enough space to write down the diagnosis. The CAS system provides the ability to store the patient information into a database so that the data is stored in a more organized manner and

no space and resources are wasted in the clinic. Furthermore, the database supports the clinic to store up to millions of patient records in an easy managerial way.^[2]

- **Make clinical functions easy**

The patients as well the physicians have to spend up quite a lot of time for performing basic functions repeatedly. The CAS system connects all the end users together and provides all the basic operations of a clinic such as requesting appointments, ordering drugs, checking on the current appointments, searching for patient and physician information, and messaging the end users. This saves up a lot of time and work on the clinical functions.^[2]

- **Save the communication time**

As all the staff will share the same database among the clinic, the nurses do not have to spend time and energy running over to get the patient records back and forth. All the staff inside the clinic can retrieve the updated information from the shared database. The physicians can also be updated on the position of drugs available in the clinic so that the clinic won't be out of them in the time of emergencies.^[2]

- **Increase the privacy level**

Patient privacy is an important aspect and the medical history of any patient must be recorded properly and confidentially. The CAS system provides a secure way to the patient's medical records as it restricts the access rights to the authority user only. Only the physician who has treated the patient can access their medical history and make changes to the same. Not even the administrator has the access to view the information.^[2]

The CAS system is a web based system which offers an easy to use interface for patients, physicians and suppliers. "clini-C-are" is a Microsoft .NET based web application and it provides a common platform for the different users responsible for the smooth operations of the clinic. All users will be able to view details about the different drugs and medicines available in the clinic. Patients and Physicians will now be able to interact with one another online thus saving precious time and delays that are common to all clinics. Physicians and clinic salesmen can also reach out to suppliers and place orders for drugs and medicines based on the current available stock at the clinic. Suppliers will use this system to verify whether there are any orders from the clinic and approve the same based on their stock. Overall this system supports four different types of users: Physicians, Patients, Suppliers and Salesmen.

Apart from these users, the Admin user will be able to administer the site in general and will be able to create and modify any of the information which is used in the system. An important feature of this system is to provide an easy to use two way messaging system between the different users of the system, providing a private inbox for all users of the system.

1.4 Significance

The prime significance of “clini-C-are” is to provide an efficient and effortless way for the medical sector with the help of high efficiency management tools, computerized and systematic patient records, drug inventory records and appointment records. It will implement all the daily activities of a clinic but in a paperless and time-efficient manner.

The medical institutions can benefit greatly by implementing a Clinical Automation System as it smoothens the functioning of the clinic. The CAS system supports five different types of users: Admin, Physicians, Patients, Suppliers and Salesman. The system consists of information about all these users and provides authorized access to individuals regarding the records of other users. Additionally, the CAS system with its detailed clinical functionalities can make the tiresome approach of visiting a clinic less time-consuming and effectively less cumbersome, thus providing significant relief to already suffering patients and improving the quality of medical care by the efficient medical staff. As mentioned, the needs of IT and computer applications in field of Medicine and lack of the same in existing trend, this project has a lot of significance to cater the need and provide the effectiveness to existing trend of the Medical Industry.

Such a system with features like information about the users, services like editing user profile, requesting and accepting/rejecting appointments, drug inventory, ordering drugs, messaging connections among the various end users can help the patients and the medical staff. With this system, the clinics can be made more approachable and the end users can be put at significant ease.

Moreover, they can be cost-effective on the business as a significant amount of time and energy is saved thus giving access to the physicians and the medical staff to take care of more patients by taking care of the repetitive tasks. The CAS system can efficiently cut down the on

time the medical staff spends on handling mundane issues- thus increasing the productivity and creativity.

1.5 Problem in Existing System and Justification

Currently, the clinics and hospitals follow the traditional system of maintaining the patient records, history and billing summary on paper. Such a system is neither user friendly, nor secure or reliable and has some major limitations.

A clinic can engage the simple paper keeping way for a small clinic that sits just one or two physicians and has a fairly low number of patients. But as the business starts growing steadily and employs more physicians, registered nurses, receptionist, and officers, the rise in the number of patients is also significant. With the growing business, quite a few problems are out: the receptionist is overwhelmed by phone calls. Many of these calls are from patients who want to get information about the prescribed drugs. Some patients want to consult with the doctors. It takes long time for receptionist to check if these patients are existing patients and are already registered as the clinic may offer free consultation for registered users who are within a time limit of their previous appointment, and they have to go through a number of records for the same. Also, the clinic wants to learn that the drugs they want to prescribe are (nearly) out of stock, and what the prices of drugs in current market are. After purchase order is made, when the drugs are arrived or the orders are not placed in a timely manner, this make sale department officer take a lot of time to contact suppliers.

A significant number of calls from patients relate to the drugs information. Since such information rarely, if ever, changes, it is a waste of time for doctors to give the same information to different patients. Second, to facilitate communication between patients and doctors, they should be able to contact on the web. The clinic can use the E-mail addresses for this purpose, but it is a concern for the E-mail privacy of the physicians as some patients can possibly misuse the physician's E-mail address. Third, although it is relatively rare that drug orders are lost or are not placed on time, this problem is troublesome. Part of the success of the clinic is its ability to offer low prices on drugs to its patients. When some patients need the drugs immediately due to the severity of their symptoms, but no such drugs are available, patients have to purchase the drugs from a pharmacy at higher prices.

Justification

As there is a limit to human capacity, there is only a certain amount of work that one can accomplish within a day. There is also a limit to the resources that can be utilized and the budget that can be allotted to the employment of clinical staff.

1.6 Organization

1.6.1 Introduction to IBM

The International Business Machines Corporation (IBM) is an American multinational technology company headquartered in Armonk, New York, with operations in over 170 countries. The company began in 1911, founded in Endicott, New York. IBM is incorporated in New York.

IBM produces and sells computer hardware, middleware and software, and provides hosting and consulting services in areas ranging from mainframe computers to nanotechnology. Nicknamed as Big Blue, the company has the largest number of employees, known as “IBMers” in India. IBM India Private Limited is the Indian subsidiary of IBM, which is headquartered at Bengaluru, Karnataka.

IBM has a large and diverse portfolio of products and services. As of 2016, these offerings fall into the categories of cloud computing, Artificial intelligence, commerce, data and analytics, Internet of Things, IT infrastructure, mobile, Digital workplace and security.

IBM Cloud includes infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS) offered through public, private and hybrid cloud delivery models

IBM provides a training prior to joining of its employees. The main objective of the training is to provide a hands-on experience to the trainers, so that they can get updated on the latest and complete aspects of any technology before taking on their roles for the organization, which is Associative Developer for Cloud Computing Services in our case.

1.6.2 About Training

For achieving this objective, we were trained on all the platforms that will be required to deploy a project for IBM clients, such as Microsoft SQL Server, HTML, CSS, JavaScript, and the various libraries of C#, which include the subjects of classes and inheritance, abstract classes, generic classes, delegates, extension methods, exception handling, event handling, collection framework, ADO.NET, LINQ, WPF, WCF and ASP.NET, MVC Framework as well as AJAX.

Chapter - 2

System Requirement Analysis

2.1 Information Gathering

A requirement is a vital feature of a new system which may include processing or capturing the data, controlling the activities of business, producing information and supporting the management.

Information gathering is a key part of the feasibility analysis process. Information gathering is both art and a science. It is a science because it requires a proper methodology and tools in order to be effective. It is an art too, because it requires a sort of mental dexterity to achieve the best results.

2.1.1 Functional Requirements

2.1.1.1 Edit Profile

All the end users would be provided with the login credentials to access this system. All the users of this system shall be able to edit his or her profile except for non-editable information like the Patient ID. Using this screen all the users will be able to modify their personal information including the password.^[3]

2.1.1.2 Browse Drug Information

This functionality will be available to any user of the system except the Supplier. Users without valid credentials will be considered as guest users. The visitor shall be able to browse the information on drugs and physicians.^[3]

2.1.1.3 Order Drugs

This functionality will be available to Patient, Salesperson and Physician users. These users shall be able to raise an order for any drugs prescribed in the clinic. The order management module will be primarily used by the suppliers and salesmen to expedite the ordering process that is currently being used at the clinic. By providing online access to suppliers and salesmen this can be achieved faster and efficiently.^[3]

2.1.1.4 Mail Box

This unique functionality of the “clini-C-are” is the ability to provide a personalized messaging system between the users of the clinic system. Patients can use this facility to

directly interact with their Physicians to clarify queries or resolve any issues via messages instead of visiting the clinic for every enquiry. This not only reduces the time Physicians need to spend with Patients for minor clarifications but also enables them to utilize their time by assisting Patients with a more immediate needs at the clinic and respond to Patients online for minor queries. Patients in turn do not have to wait in queues to have their queries clarified. Physicians can also utilize this facility to interact directly with Salesperson in order to ensure that the drugs available at the clinic are ordered on time. Overall this messaging system will be used as an important communication system between the users of the system, thus improving the efficiency of the clinic to a great extent.^[3]

2.1.1.5 View Drug Order

The suppliers are the only third party vendors of the clinic who will support the need for medicines and drugs at the clinic. By providing online access to Suppliers, they can now monitor the clinic for any possible orders for drugs. This can be done using the View Drug Orders functionality. Once the Supplier views the pending orders in the system, he can take appropriate action on the order by approving the order or in case of any clarifications on the order items, he can directly reach out to the Salesperson via the messaging system. This functionality speeds up the order processing activities in the clinic and enables the Suppliers to be constantly aware of the needs of the clinic and prepare to supply the medicines accordingly.^[3]

2.1.2 Non-Functional Requirements

2.1.2.1 Usability

For maximum accessibility, the system interface shall not use HTML frames but it shall be accessible by any forms-capable browser that supports the use of the <table> tag. The interface should be simple and yet can easily be navigated even by inexperienced Web users.

2.1.2.2 Performance

The following requirements are specified for the Web Server performance.

- The Web Server shall be able to handle at least 30 simultaneous user sessions.
- The system shall require no more than 5 seconds to retrieve and to respond to a client's request for a static Web page.
- The system shall require no more than 10 seconds to respond to a dynamic Web page.

However, in case of operations that may take longer than the specified period, users shall be notified in advance accordingly. [3]

2.1.2.3 Robustness

For robustness, the data entered by the user shall be checked at the client side. In case of data entry errors, helpful messages shall be displayed so that the user understands the cause of the errors. [3]

2.1.2.4 Reliability

The maximum acceptable downtime for maintenance and upgrade shall not exceed more than two hours a week. [3]

2.1.2.5 Security

The system must ensure that only authorized people can view or update records strictly based on the functional requirements. Only an existing patient of the clinic shall be able to edit his or her profile and leave messages to and retrieve messages from a physician and order drugs and check order status from the clinic. Only doctors are authorized to edit their profile, update patients' medical history, order drugs, check order status and leave messages to and retrieve messages from patients. Only a Salesperson is authorized to take orders and update their status from the patients and the physicians, place orders to the supplier, check the order status as well as communicate with the doctors and the supplier. Only a supplier is authorized to take orders online and may leave a message regarding the order to the doctor who has placed the order. Guests shall be able to view only drug and physician information only. [3]

2.2 System Feasibility

Feasibility is a process that identifies, describes and evaluates proposed system and selects the best system for the job. During the study, the problem definition is solved and all aspects of problem to be included in the system are determined. Size of project, cost and benefits are also estimated with greater accuracy.

2.2.1 Economical Feasibility

“Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the cost then it is not worth going ahead.” Economic analysis the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. The cost involved will be of development and maintenance of the software. But the benefits of the project will certainly cut down the costs involved in the traditional management of the project which was done manually. Since, hardware and software requirements of the system are comparatively moderate, hence it is economically feasible.

2.2.2 Technical Feasibility

The technical feasibility of the project deals with the availability and its actual implementation ability with the existing tools and techniques available in the software market world. The following are the notable points about the technical feasibility of the project:

- Used web development technologies like HTML5, CSS, and Bootstrap.
- Used ASP.NET MVC 5 for building the application which is a high-end web framework based on Model-View-Controller architecture combined with up-to-date techniques from Agile Development and the best parts of the existing ASP.NET platform.
- Used ADO.NET to provide a bridge between the front end controls and the back end database, which was SQL Server. This was integrated with LINQ.
- AJAX was used for sending data to the server in the background and updating the web pages without reloading i.e. asynchronously.

With all these perspectives taken into consideration, the project is technically feasible to implement.

2.2.3 Behavioral Feasibility

After the analysis of the requirement from the proposed system and specifications of the proposed system a feasibility study of the projected system is conducted. The feasibility study is done to find whether the system is beneficial to the users and organization or not. The feasibility study is carried out to select the best system that meets performance requirements.

The feasibility study includes the investigation of the information needs of the end user and objectives, constraints, basic resource requirements and cost benefits. The main and prime objective of feasibility study is not to solve the problem, but to acquire a sense of its scope.

2.3 Platform Specification (Development and Deployment)

The software is being developed on Windows 10 64-bit operating system. It is a web application that will be available for everyone once deployed over a domain.

2.3.1 Hardware

The project will require a System with interactive user-interface, a system that supports latest web-browsers and technologies in order for smooth accessing of the project.

2.3.2 Software Implementation/Technology

Developer End

- Visual Studio 2019 – IDE from Microsoft for Front-end and Back-end Development
- .NET Framework – Developer platform made up of tools, programming languages and libraries for building applications
- ASP.NET Framework – Platform with tools and libraries for building web applications
- Entity Framework – Object-Database mapper for .NET
- Microsoft SQL Server Management Studio – software application used for configuring, managing, and administering components within Microsoft SQL Server
- Front-end languages like HTML, CSS, JavaScript and Bootstrap along with AJAX
- Back-end Language – C# and its libraries

Customer End

- Any Web Browser – To access web application of the platform

Chapter – 3

System Analysis

The Merriam-Webster dictionary defines system analysis as “the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way”. Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose. It is also "an explicit formal inquiry carried out to help a decision maker identify a better course of action and make a better decision than she might otherwise have made."

3.1 Information Flow Representation

An information flow diagram (IFD) is a diagram that shows how information is communicated (or "flows") from a source to a receiver or target (e.g. A→C), through some medium. The medium acts as a bridge, a means of transmitting the information. The diagrammed system may also include feedback, a reply or response to the signal that was given out. The return paths can be two-way or bi-directional: information can flow back and forth.

An IFD can be used to model the information flow throughout an organisation. An IFD shows the relationship between internal information flows within an organisation and external information flows between organisations. It also shows the relationship between the internal departments and sub-systems.

An IFD usually uses "blobs" to decompose the system and sub-systems into elemental parts. Lines then indicate how the information travels from one system to another. IFDs are used in businesses, government agencies, television and cinematic processes.

3.1.1 Entity-Relationship Diagram

Entity Relationship Diagram, also known as ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: the major entities within the system scope, and the inter-relationships among these entities. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). [6]

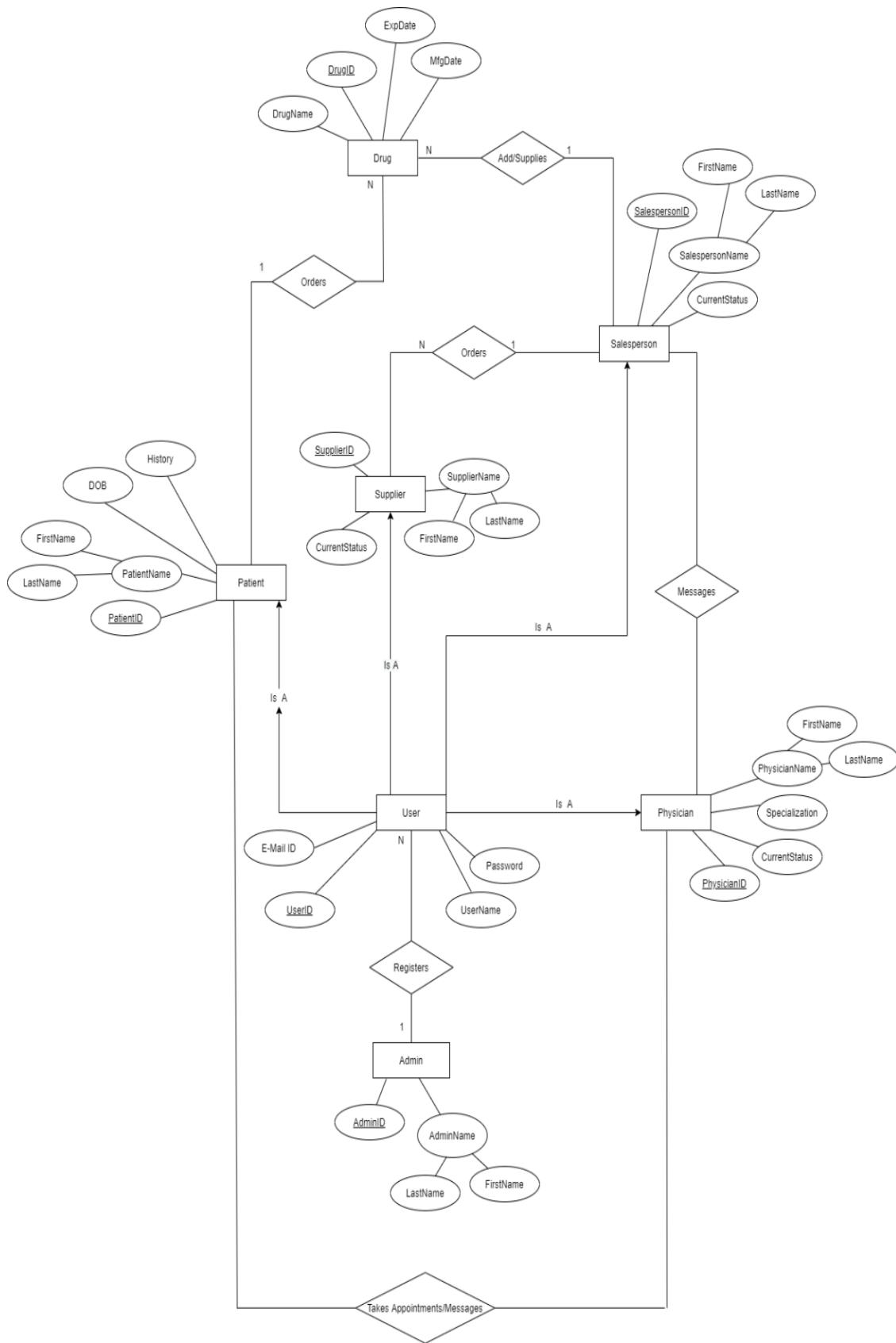


Fig 3.1 E-R Diagram of Clinical Automation System

3.1.2 Data Dictionary

A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata. Also called a Data Definition Matrix, a data dictionary provides detailed information about the business data, such as standard definitions of data elements, their meanings, and allowable values.^[6]

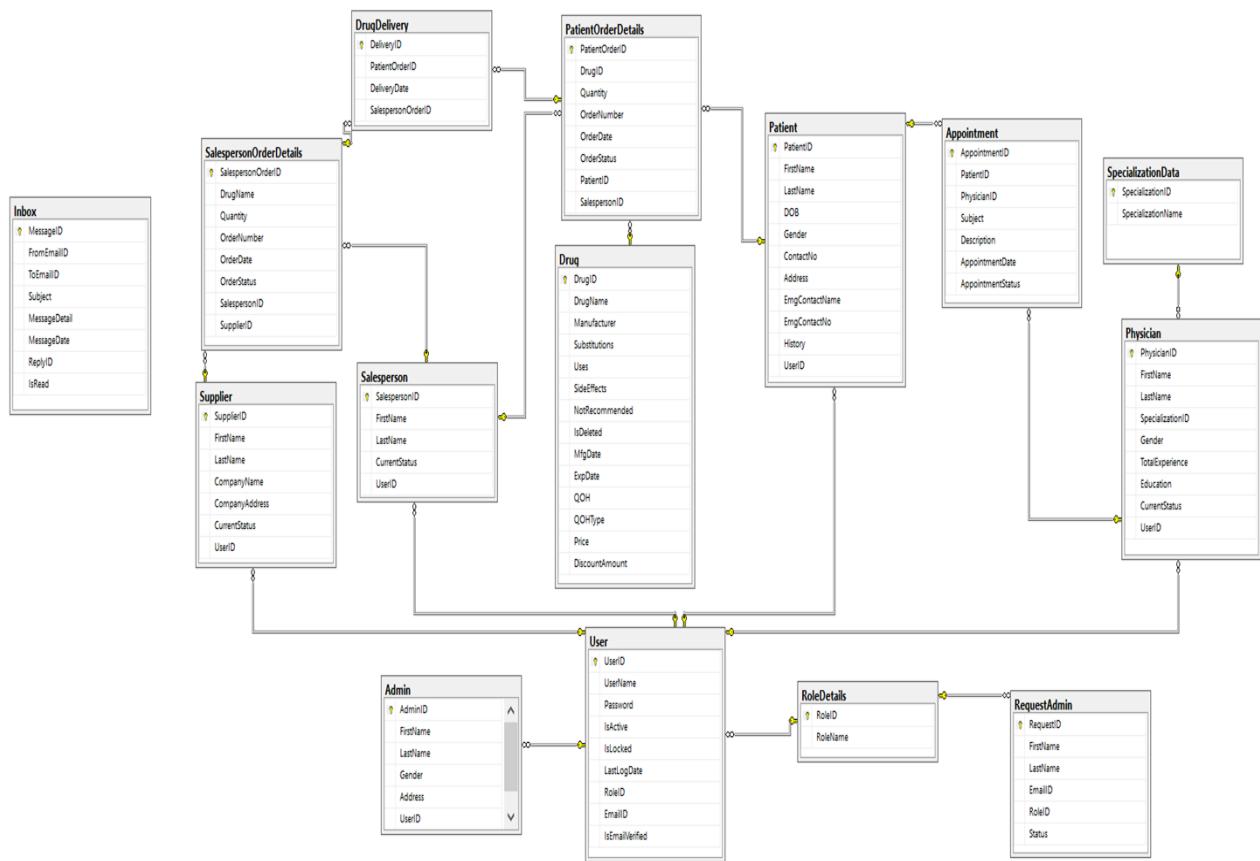


Fig 3.2 Data Dictionary of Clinical Automation System

3.1.3 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, and there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer.

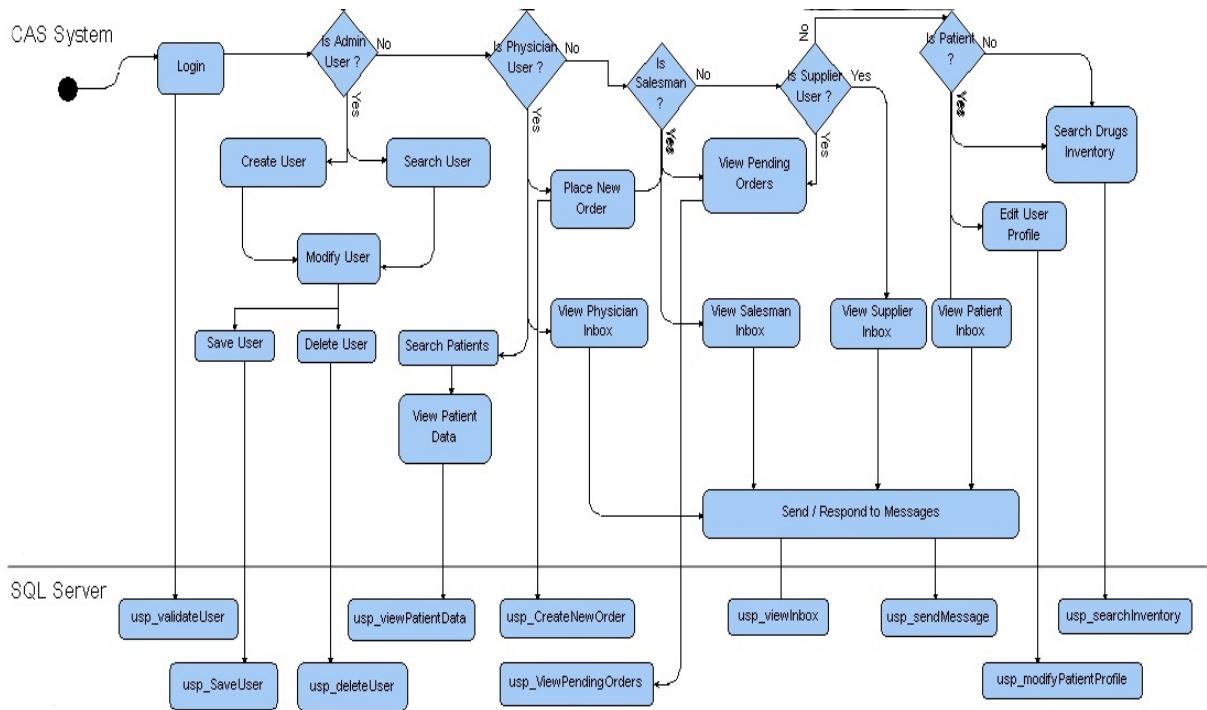


Fig 3.3 Data Flow Diagram of Clinical Automation System

3.1.4 Class Diagram

A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. [6]

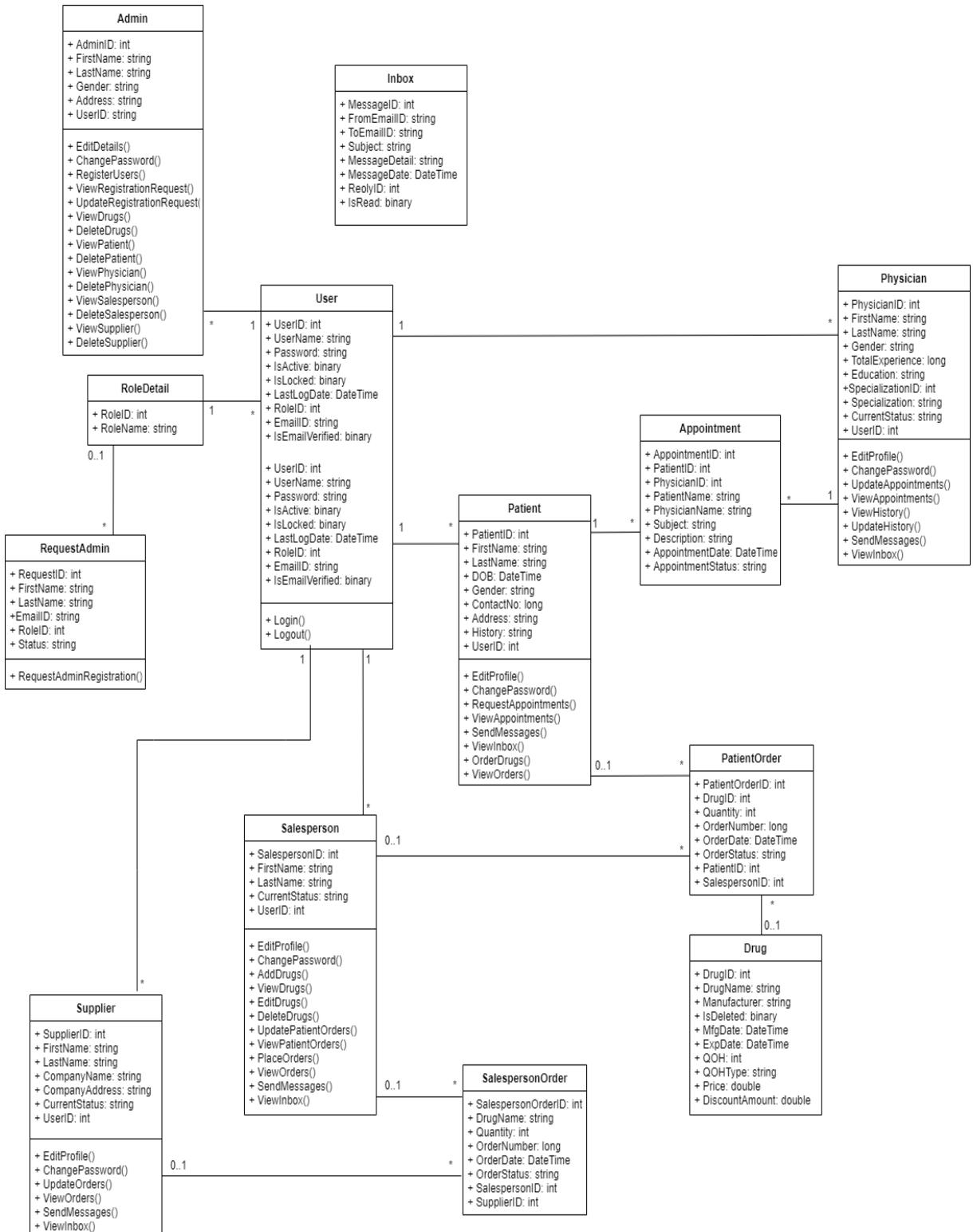


Fig 3.5 Class Diagram of Clinical Automation System

Chapter - 4

Design

Software architecture involves the high-level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability, and availability. A software architecture must describe its group of components, their connections, interactions among them and deployment configuration of all components. Software architecture can be defined in many ways –

- **UML** – UML is one of object-oriented solutions used in software modeling and design.
- **Architecture View Model (4+1 view model)** – Architecture view model represents the functional and non-functional requirements of software application.
- **ADL (Architecture Description Language)** – ADL defines the software architecture formally and semantically

4.1 Architectural Design

The software needs the architectural design to represent the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectural styles. Each style will describe a system category that consists of:

- A set of components (e.g.: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

4.1.1 Architectural Context Diagram

The context diagram is used to establish the context and boundaries of the system to be modelled: which things are inside and outside of the system being modelled, and what is the relationship of the system with these external entities.

A context diagram, sometimes called a level 0 data-flow diagram, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.

In order to produce the context diagram and agree on system scope, the following must be identified:

- external entities
- data-flows

4.1.2 Architectural Behavioral Diagram

Behavioral Diagrams depict the elements of a system that are dependent on time and that convey the dynamic concepts of the system and how they relate to each other. The elements in these diagrams resemble the verbs in a natural language and the relationships that connect them typically convey the passage of time.

Use Case Diagrams:

Use Case diagrams capture Use Cases and relationships among Actors and the system; they describe the functional requirements of the system, the manner in which external operators interact at the system boundary, and the response of the system.

Actor:

An actor represents the roles that the users of the use cases play. An actor may be a person (e.g. student, customer), a device (e.g. workstation), or another system (e.g. bank, institution).

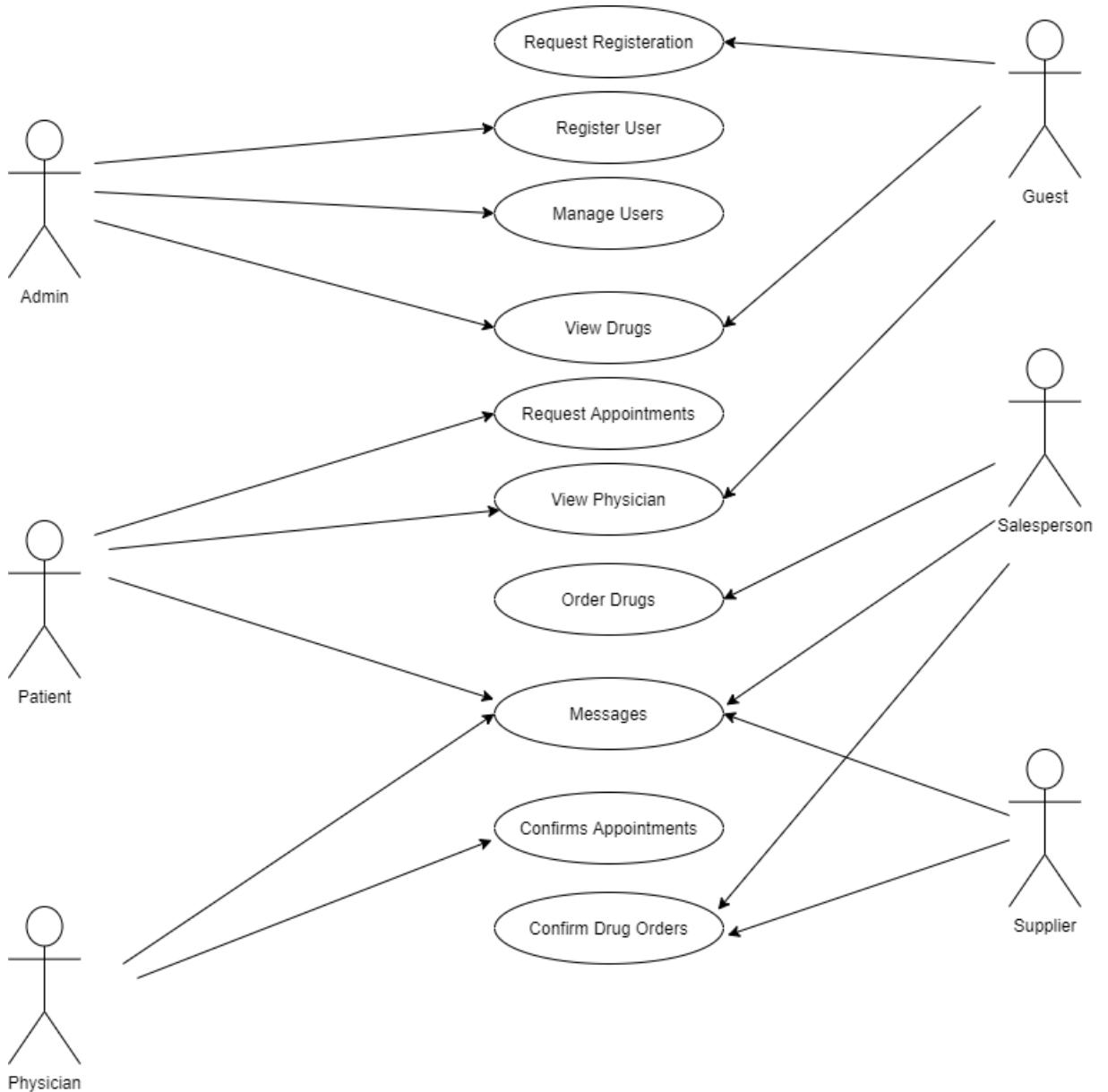


Fig 4.1 Use Case Diagram of Clinical Automation System

4.1.3 Description of Architecture diagrams

Behavioral diagrams are used to depict how the software behaves and interacts with the different conditions of the environment with different actors. There are various kind of actors that can act on a software.

In our project there are two actors, one is a guest user, and then there are Admin Users, Patient Users, Physician Users, Supplier User, and Salesperson User, that can interact with the system. There are many use cases that defines the working of the project. Admin User can

Register New Users, Manage Existing Users, and can Manage Drugs. Patient User can manage their profile, Request Appointments, View Physicians, Order Drugs, and Message Physician. Physician User can manage their profile, Confirm Appointments, and Message Salesperson and Reply to Patients. Salesperson User can manage their profile, Manage Drugs Inventory, Confirm Patient Drug Orders, and Order Drugs from Supplier, Message Supplier and Reply to Physician. Supplier User can manage the profile, Confirm Salesperson Drug Orders and Reply to Salesperson. While the guest user can view drug inventory and physician information already available and request the admin to register to the account.

The various actors in the system interact in various capacities with the system. The current version is a prototype version and hence the actors have limited interaction with the software. However, as the system grows in utility the functionalities will keep on adding up.

However, the kind of actors is unlikely to change. As the functionalities add up the use cases will grow and also relationships between various use cases will also develop.

4.2 Modular and Procedural Approach

.NET Framework provides various inbuilt libraries for web development purposes. In this project we have used various .NET libraries for various purposes. Since this is a Web Development project the libraries that we used are ASP.NET and worked on MVC framework. Secondly for database connection purposes we use Entity Framework with LINQ and ADO.NET. For the development of GUI, we used front-end technologies.

4.2.1 Modules Used

4.2.1.1 System

The System namespace contains fundamental classes and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions. Other classes provide services supporting data type conversion, method parameter manipulation, mathematics, remote and local program invocation, application environment management, and supervision of managed and unmanaged applications.^[9]

The System namespace is the root of all namespaces in the .NET Framework, containing all other namespaces as subordinates. It also contains the types that we felt to be the most fundamental and frequently used.

Major highlights of this namespace are as follows:

- Includes the primitive data types
- Includes the Object data type
- Contains the abstract base class Array

4.2.1.2 System.Web.Mvc

ASP.NET MVC namespaces and classes are located in the System.Web.Mvc assembly. System.Web.Mvc namespace contains classes and interfaces that support the MVC pattern for ASP.NET Web applications. This namespace includes classes that represent controllers, controller factories, action results, views, partial views, and model binders. System.Web.Mvc.Ajax namespace contains classes that support Ajax scripts in an ASP.NET MVC application. The namespace includes support for Ajax scripts and Ajax option settings. System.Web.Mvc.Async namespace contains classes and interfaces that support asynchronous actions in an ASP.NET MVC application. System.Web.Mvc.Html namespace contains classes that help render HTML controls in an MVC application. The namespace includes classes that support forms, input controls, links, partial views, and validation.^[11]

Some important features of System.Web.Mvc namespace are listed as follows:

- **Controller Class**

Provides methods that respond to HTTP requests that are made to an ASP.NET MVC Web site.

- **ActionResult Class**

Represents the result of an action method. All the public methods of a Controller class are called Action methods.

- **HttpGetAttribute Class**

Represents an attribute that is used to restrict an action method so that the method handles only HTTP GET requests.

- **HttpPostAttribute Class**

Represents an attribute that is used to restrict an action method so that the method handles only HTTP GET requests.

- **JsonResult Class**

Represents a class that is used to send JSON-formatted content to the response.

4.2.1.3 System.Linq

The System.Linq namespace provides classes and interfaces that support queries that use Language-Integrated Query (LINQ). LINQ API uses two main static classes: Enumerable and Queryable. The static Enumerable class includes extension methods for classes that implements the `IEnumerable<T>` interface. `IEnumerable<T>` type of collections are in-memory collection like List, Dictionary, SortedList, Queue, HashSet, LinkedList. The static Queryable class includes extension methods for classes that implements the `IQueryable<T>` interface.^[10]

4.2.1.4 System.Net

The System.Net namespace provides a simple programming interface for many of the protocols used on networks today. The `WebRequest` and `WebResponse` classes form the basis of what are called pluggable protocols, an implementation of network services that enables you to develop applications that use Internet resources without worrying about the specific details of the individual protocols.^[12]

4.2.1.5 MVC Framework

MVC stands for Model, View, and Controller. MVC separates an application into three components – Model, View, and Controller.

- **Model**

Model represents domain specific data and business logic in MVC architecture. It maintains the data of the application. Model objects retrieve and store model state in the persistence store like a database. Model class holds data in public properties. All the Model classes reside in the Model folder in MVC folder structure.^[5]

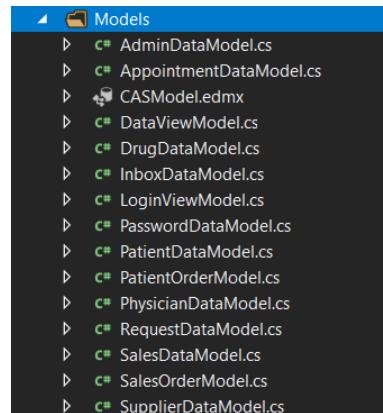


Fig 4.2 Models in Clinical Automation System

- **Controller**

The Controller in MVC architecture handles any incoming URL request. Controller is a class, derived from the base class `System.Web.Mvc.Controller`. Controller class contains public methods called Action methods. Controller and its action method handles incoming browser requests, retrieves necessary model data and returns appropriate responses. In ASP.NET MVC, every controller class must end with a word “Controller”.^[5]

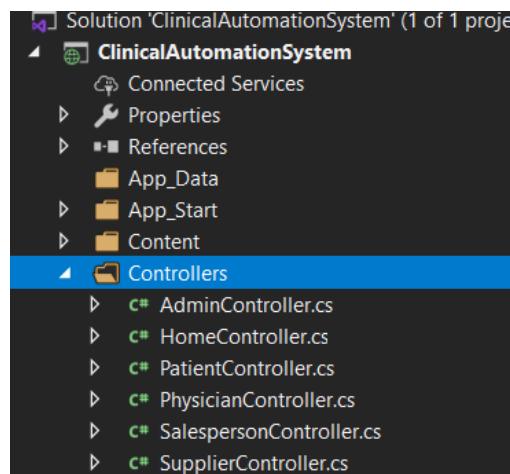


Fig 4.3 Controllers in Clinical Automation System

- **View**

View is a user interface. View enables data from the model to the user and also enables them to modify the data. ASP.NET MVC views are stored in Views folder. Different action methods of a single controller can render different views, so the Views folder contains a separate folder for each controller with the same name as controller, in order to accommodate multiple views.^[5]

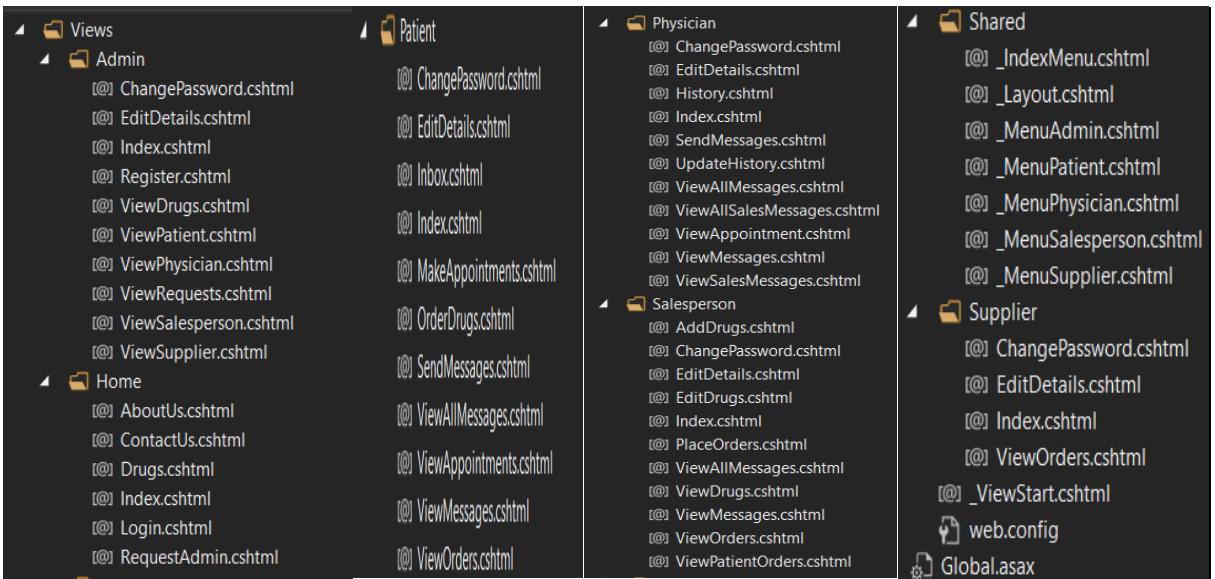


Fig 4.4 Views in Clinical Automation System

4.2.1.6 AJAX

AJAX stands for Asynchronous JavaScript and XML. AJAX is not a programming language and uses a combination of: A browser built-in XMLHttpRequest object (to request data from a web server) and JavaScript and HTML DOM (to display or use the data).

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

4.2.2 Internal Data Structures

While dealing with machine learning projects it becomes inevitable to deal with large chunks of data. To deal with large amounts of data it requires proper organization of data with the help of various data structures. Various data structures are used to store data in various formats. For example, an array stores homogenous data while a list can store heterogeneous data too.

To deal with the large amount of string and character data in this project we have used a distinct data structures in the Collection library of .NET, which is list for its distinct features.

List:

A list is a linear data structure that can store any data type in a linear form in a single row. It is a collection which is ordered and changeable.

Since there is currently no requirement of high end and extensive searching, there is also no requirement of data structures like data frames, trees or hash tables. The simple and elegant use of the simple data structure makes the program light on the computer's memory but also makes the processing faster and easier.

4.3 Data Design

Data Design is defined as the design process aiming at the definition of the application's data. We have an extensive database for this project as information needs to securely held for quite a long period of time.^[7]

Field Name	Data Type	Length	Key	Default Value
UserID	int	-	Primary Key	Not Null
UserName	varchar	15		Not Null
Password	varchar	20		Not Null
IsActive	bit	-		Not Null
IsLocked	bit	-		Not Null
LastLogDate	date	-		Null
RoleID	int	-	Foreign Key	Not Null
EmailID	varchar	50		Null
IsEmailVerified	bit	-		Null

Table 4.1 User Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
RoleID	int	-	Primary Key	Not Null
RoleName	varchar	50		Not Null

Table 4.2 Role Details Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
SpecializationID	int	-	Primary Key	Not Null
RoleName	varchar	50		Not Null

Table 4.3 Specialization Data Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
AdminID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
Gender	varchar	10		Null
Address	varchar	MAX		Null
UserID	int	-	Foreign Key	Not Null

Table 4.4 Admin Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
PatientID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
DOB	date	-		Null
Gender	varchar	10		Null
ContactNo	nchar	10		Null
Address	varchar	MAX		Null
EmgContactName	varchar	50		Null
EmgContactNo	nchar	10		Null
History	nvarchar	MAX		Null
UserID	int	-	Foreign Key	Not Null

Table 4.5 Patient Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
PhysicianID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
Gender	varchar	10		Null
SpecializationID	int	-	Foreign Key	Null
TotalExperience	int	-		Null
Education	varchar	100		Null
CurrentStatus	varchar	20		Null
UserID	int	-	Foreign Key	Not Null

Table 4.6 Physician Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
SupplierID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
CompanyName	varchar	50		Null
CompanyAddress	varchar	MAX		Null
CurrentStatus	varchar	20		Null
UserID	int	-	Foreign Key	Not Null

Table 4.7 Supplier Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
SalespersonID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
CurrentStatus	varchar	20		Null
UserID	int	-	Foreign Key	Not Null

Table 4.8 Salesperson Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
AppointmentID	int	-	Primary Key	Not Null
PatientID	int	-	Foreign Key	Not Null
PhysicianID	int	-	Foreign Key	Not Null
Subject	varchar	100		Null
Description	varchar	MAX		Not Null
AppointmentDate	datetime2	7		Not Null
AppointmentStatus	varchar	20		Not Null

Table 4.9 Appointment Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
DrugID	int	-	Primary Key	Not Null
DrugName	varchar	200		Not Null
Manufacturer	varchar	50		Not Null
Substitutions	varchar	200		Null
Uses	varchar	250		Null
SideEffects	varchar	250		Null
NotRecommended	varchar	250		Null
IsDeleted	bit	-		Null
MfgDate	date	-		Not Null
ExpDate	date	-		Not Null
QOH	int	-		Not Null
QOHType	varchar	20		Not Null
Price	money	-		Not Null
DiscountAmount	money	-		Null

Table 4.10 Drug Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
PatientOrderID	int	-	Primary Key	Not Null
DrugID	int	-	Foreign Key	Null
Quantity	int	-		Null
OrderNumber	int	-		Null
OrderDate	date	-		Null
OrderStatus	varchar	20		Null
PatientID	int	-	Foreign Key	Null
SalespersonID	int	-	Foreign Key	Null

Table 4.11 Patient Order Details Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
SalespersonOrderID	int	-	Primary Key	Not Null
DrugName	varchar	50		Null
Quantity	int	-		Null
OrderNumber	int	-		Null
OrderDate	date	-		Null
OrderStatus	varchar	20		Null
SalespersonID	int	-	Foreign Key	Null
SupplierID	int	-	Foreign Key	Null

Table 4.12 Salesperson Order Details Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
DeliveryID	int	-	Primary Key	Not Null
DeliveryDate	date	-		Null
PatientOrderID	int	-	Foreign Key	Null
SalespersonOrderID	int	-	Foreign Key	Null

Table 4.13 Drug Delivery Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
MessageID	int	-	Primary Key	Not Null
FromEmailID	varchar	50		Not Null
ToEmailID	varchar	50		Not Null
Subject	varchar	100		Null
MessageDetail	varchar	MAX		Not Null
MessageDate	date	-		Not Null
ReplyID	int	-		Not Null
IsRead	bit	-		Not Null

Table 4.14 Inbox Table in CAS Database

Field Name	Data Type	Length	Key	Default Value
RequestID	int	-	Primary Key	Not Null
FirstName	varchar	50		Null
LastName	varchar	50		Null
EmailID	varchar	50		Null
RoleID	int	-	Foreign Key	Null
Status	varchar	20		Null

Table 4.15 Request Admin Table in CAS Database

4.3.1 Data objects and Resultant Data Structures

A data object is a region of storage that contains a value or group of values. Each value can be accessed using its identifier or a more complex expression that refers to the object. In addition, each object has a unique data type. The data type of an object determines the storage allocation for that object and the interpretation of the values during subsequent access. The data objects used in the system are: Strings, Integers and Floating-point numbers. The resultant Data structures are in the form of lists that are used for dataset operations. All the operations and functions are applied to these models to work with data and obtain results.

4.4 Interface Design

The interface of the project is quite realistic and lively but also quite simple as the GUI for “clini-C-are” website is quite interactive, quick responsive and functional, that gives us a great platform to interact with our users, even those who have no technical experience.

The functioning of the website starts with our Home Page which allows all the users to navigate to the Login page along with the View Drugs page and the View Physicians page while the Guest User can even go to the Request Admin page to request the admin to register them for the page.

After the Login, the users are redirected to their respective User Dashboards and then they can navigate to the various functionalities that are improvised and designed for their respective roles.

4.4.1 Human Machine Interface Design Specification

The human machine interface design of the project gives the look and feel of a website that is easy to use and handle by people of any age group with basic computer knowledge. The interface provided is simple with a focus on giving a comforting feel to navigate while not overwhelming the users who are either on their job or are seriously sick.

The interface is designed to give a feel of comfort and simplicity, which can be further accommodated to the needs of that particular clinic. Thus, it can be said that this project can be a base project for the website development of not one but quite a few different clinics.

The users after login are directed to their respective Home Pages, where they can follow the links provided to accomplish various tasks according to their respective jobs and roles.



Fig: 4.5 Home Page of clini-C-are

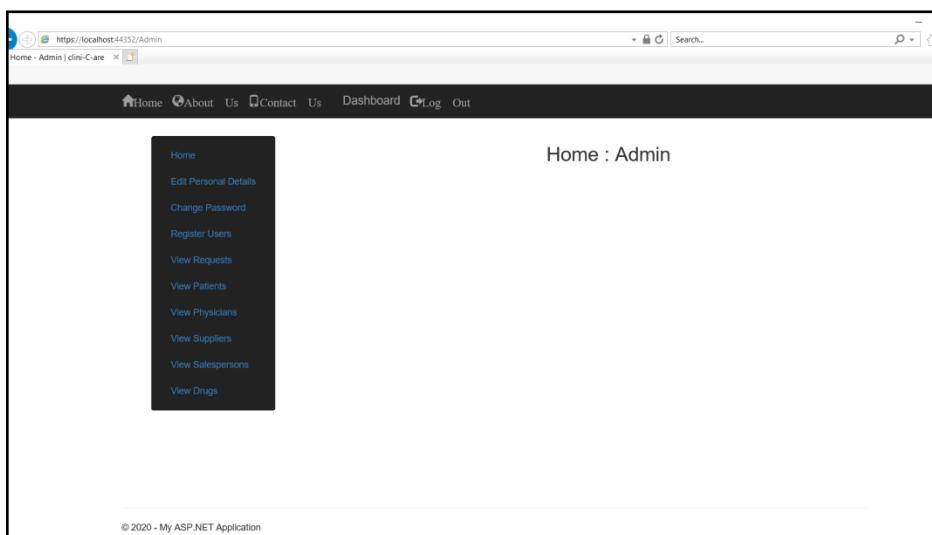


Fig: 4.6 User Dashboard of Admin in clini-C-are

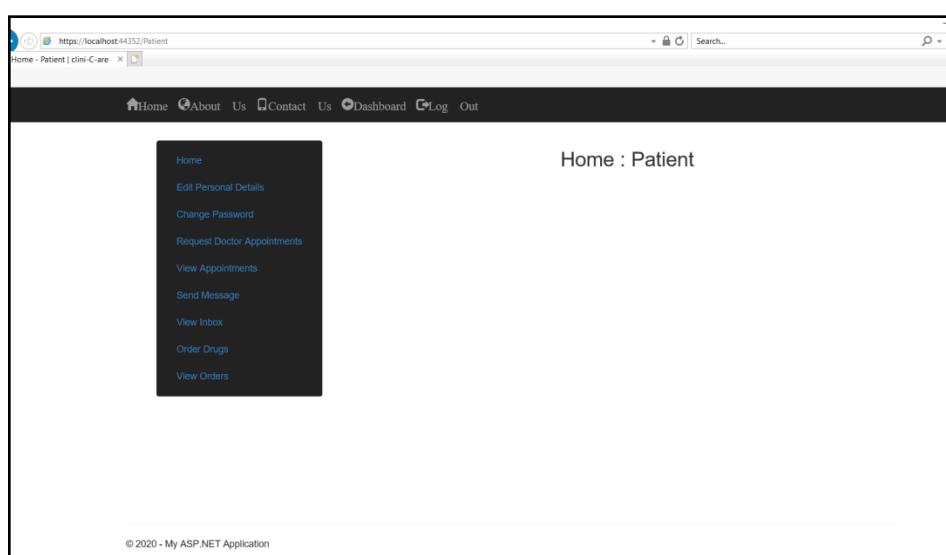


Fig: 4.7 User Dashboard of Patient in clini-C-are

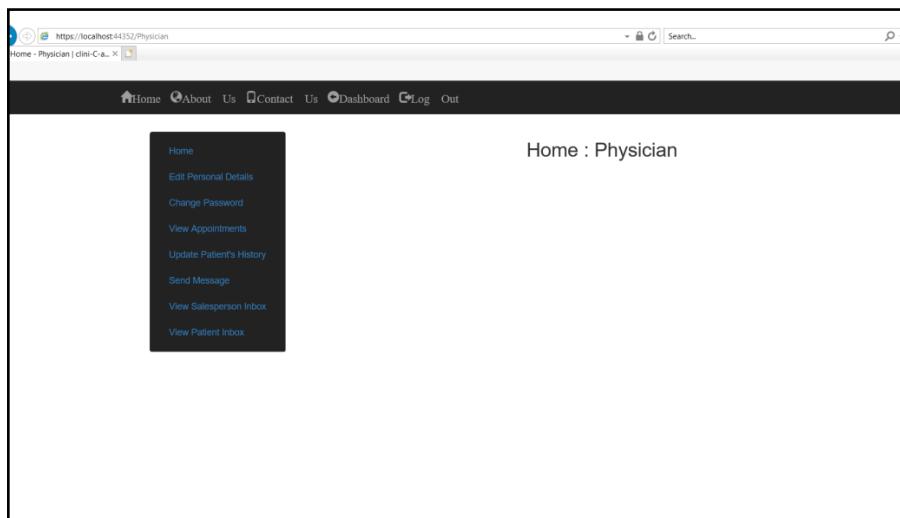


Fig: 4.8 User Dashboard of Physician in clini-C-are

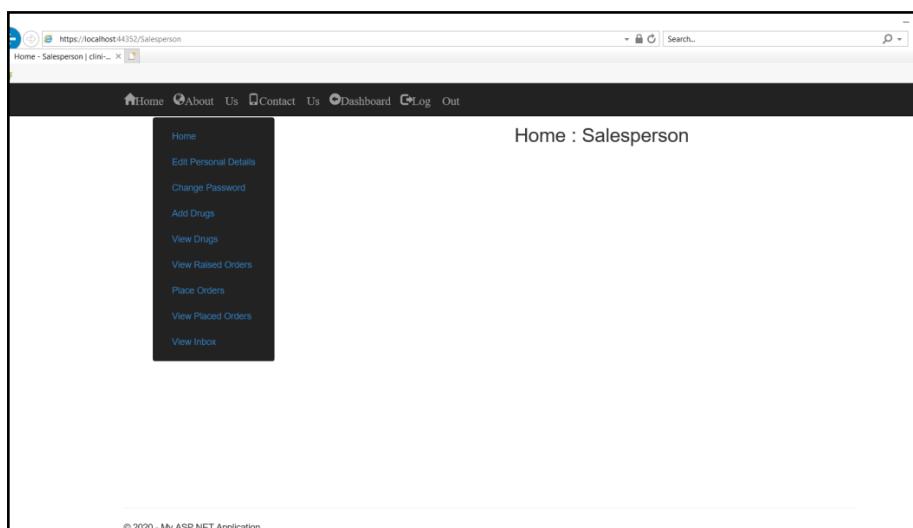


Fig: 4.9 User Dashboard of Salesperson in clini-C-are

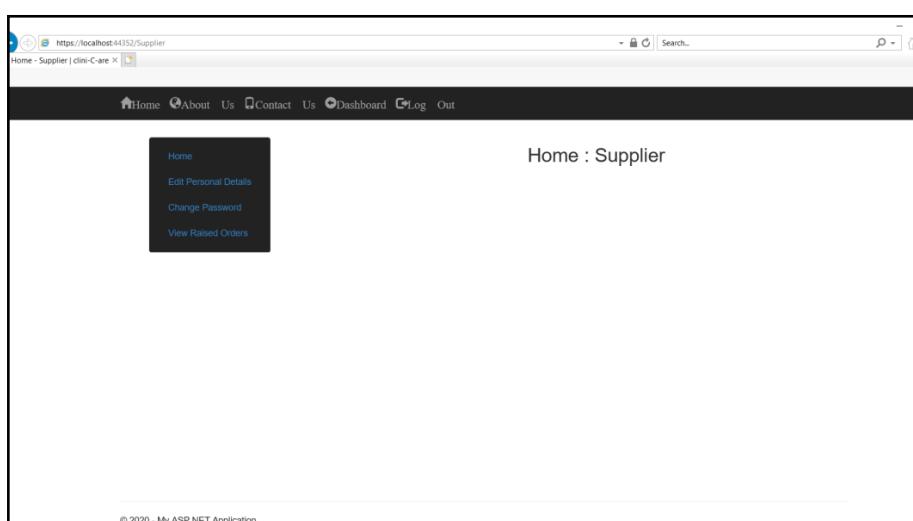


Fig: 4.10 User Dashboard of Supplier in clini-C-are

4.4.2 I/O Forms

The input of the user is taken using the website by using the frontend technologies like HTML, CSS, Bootstrap and AJAX which when integrated inside the ASP.NET MVC Framework makes the working of the project extraordinary.

As it is mentioned earlier and also indicated in the User Dashboards of various users above, every user has a different role to play in the medical sector and hence, in this application. All the inputs that are taken are integrated with the working of MVC Framework, Entity Framework and C# libraries to ensure proper and efficient functioning of the application.

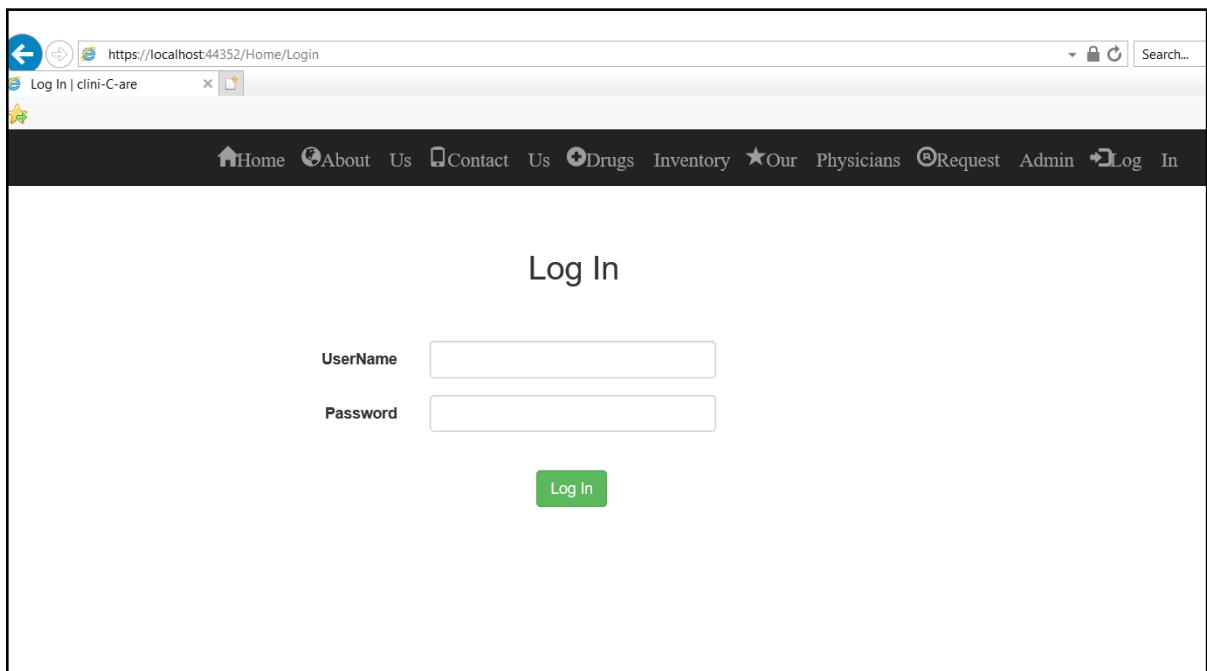


Fig: 4.11. Login Page in clini-C-are

The workings of the various Users/Roles are:

- **Guest**

The Guest User is only authenticated to view the list of drugs available on the clinic and the physicians that are currently available on the current working hours of the clinic. They can further request the Admin to add them as a user.

Request Admin

First Name:

Last Name:

E-mail ID:

Role:

- Select Role
- Patient
- Physician
- Supplier

Request

Fig: 4.12. Request Admin Page in clini-C-are

• Admin

The Admin User is the only user authenticated to register new users. They are the only one with the access to the account information of all the users and can delete those users if necessary. Further they have the control to delete the drugs from the inventory.

Registration

UserName:

E-mail ID:

Role:

Submit

Home

[Edit Personal Details](#)

[Change Password](#)

[Register Users](#)

[View Requests](#)

[View Patients](#)

[View Physicians](#)

[View Suppliers](#)

[View Salespersons](#)

[View Drugs](#)

Fig: 4.13. Registration Page in Admin Module in clini-C-are

Physician Name	Gender	Total Experience	Education	Current Status	Specialization	E-mail ID	
Astha Mishra	Female	18	MBBS, MD-Medicine, DM-Neurology	Active	Neurologist	asthamishra315@gmail.com	<button>Delete</button>
Shubham Jain	Male	20	MBBS, DCH	Active	Pediatrician	25.shubhamjain.1998@gmail.com	<button>Delete</button>
Sanjay Gujrati	Male	25	MBBS, MD	Active	Cardiologist	gujratisanjay@yahoo.com	<button>Delete</button>

Fig: 4.14. View Physicians Page in Admin Module in clini-C-are

Drug Name	Manufacturer Name	Manufacturing Date	Expiry Date	Price	Discount Amount	Quantity Available	Is Deleted	
Halovate F Cream	Glenmark	15-06-2019	14-06-2022	187	158.95	50 tubes	False	<button>Delete</button>
Vomikind-MD 4 DT	Mankind Pharma Ltd	24-03-2020	23-03-2023	39.7	33.7	50 strips	False	<button>Delete</button>
Crocin Advance	GlaxoSmithKline	12-11-2019	11-11-2021	14.95	13.95	35 strips	False	<button>Delete</button>
Levocet M	Hetero Drugs Ltd	13-12-2019	12-12-2021	84	71.4	50 strips	False	<button>Delete</button>

Fig: 4.15. View Drugs Page in Admin Module in clini-C-are

• Patient

The Patient User has the facility to request for doctor appointments, view the status of those appointments, send message queries to their physicians, view their inboxes, order drugs and check the status of those drugs.

host:44352/Patient/MakeAppointments?Length=7

Request Doctor Appointment

Specialization: Select Specialization

Physician: Select Physician

Appointment Date: 17-05-2020 19:43:05

Subject:

Description:

Request Appointment

Fig: 4.16. Request Appointment Page in Patient Module in clini-C-are

44352/Patient/ViewAllMessages/0

View All Messages

Message From	Messages	Date
guratsanjay@yahoo.com	Yes, keep taking them. Will update you tomorrow.	15-05-2020
tanuthoot16@gmail.com	Okay, Sir.	15-05-2020
guratsanjay@yahoo.com	Send me the reports now.	15-05-2020

Send

Fig: 4.17. View Inbox Page in Patient Module in clini-C-are

host:44352/Patient/OrderDrugs

Order Drugs

Drug: Order Advance

Total Quantity Available: 45 strips

Order Quantity:

Order Drug

Fig: 4.18. Order Drugs Page in Patient Module in clini-C-are

- **Physician**

The Physician User has the authorization to Update Patients' Appointments, View and Update Patients' History, along with the facility to communicate with the Patients and the Salesperson.

Patient Name	Gender	Subject	Description	Appointment Date	Appointment Status
Tanu Dhoot	Female	Regarding ECG	Have to show the ECG reports.	16-05-2020 10:10:00	Accepted
Dhruv Dhoot	Male	Regarding Bypass Surgery	Bypass Surgery Followup Session	18-05-2020 12:45:00	Accept Reject

Fig: 4.19. View Appointments Page in Physician Module in clini-C-are

Got a minor problem in the ECG Report. Took some medications. The

Update History

Fig: 4.20. Update Patient's History Page in Physician Module in clini-C-are

Message From	Messages	Date
guptashubh254@gmail.com	Yes, they are. How much quantity do you need?	15-05-2020
guratsanjay@yahoo.com	50 strips.	15-05-2020

Send

Fig: 4.21. View Salesperson Messages Page in Physician Module in clini-C-are

- **Salesperson**

The Salesperson User has been provided the facilities to Add Drugs to the drug inventory, Edit and Delete those drugs, Update Patients' Orders, and Place Orders to the Supplier for the clinic and View the Placed Orders. They are also provided the messaging facility to contact with the Physician and Supplier.

44352/Salesperson/AddDrugs?Length=11

Home About Us Contact Us Dashboard Log Out

Add Drugs

Drug Name	<input type="text"/>
Manufacturer Name	<input type="text"/>
Substitutions	<input type="text"/>
Uses	<input type="text"/>
Side Effects	<input type="text"/>
Not Recommended To	<input type="text"/>
Manufacturing Date	<input type="text"/>
Expiry Date	<input type="text"/>
Quantity On Hand	<input type="text"/>

Fig: 4.22. Add Drugs Page in Salesperson Module in clini-C-are

44352/Salesperson/ViewDrugs?Length=11

Home About Us Contact Us Dashboard Log Out

View Drugs

Drug Name	Manufacturer Name	Manufacturing Date	Expiry Date	Price	Discount Amount	Quantity Available	Is Deleted	Edit	Delete
Halovate F Cream	Glenmark	15-06-2019	14-06-2022	187	158.95	50 tubes	False	<button>Edit</button>	<button>Delete</button>
Vomikind-MD 4 DT	Mankind Pharma Ltd	24-03-2020	23-03-2023	39.7	33.7	50 strips	False	<button>Edit</button>	<button>Delete</button>
Crocin Advance	GlaxoSmithKline	12-11-2019	11-11-2021	14.95	13.95	45 strips	False	<button>Edit</button>	<button>Delete</button>
Levocabet M	Hetero Drugs Ltd	13-12-2019	12-12-2021	84	71.4	50 strips	False	<button>Edit</button>	<button>Delete</button>

Fig: 4.23. View Drugs Page in Salesperson Module in clini-C-are

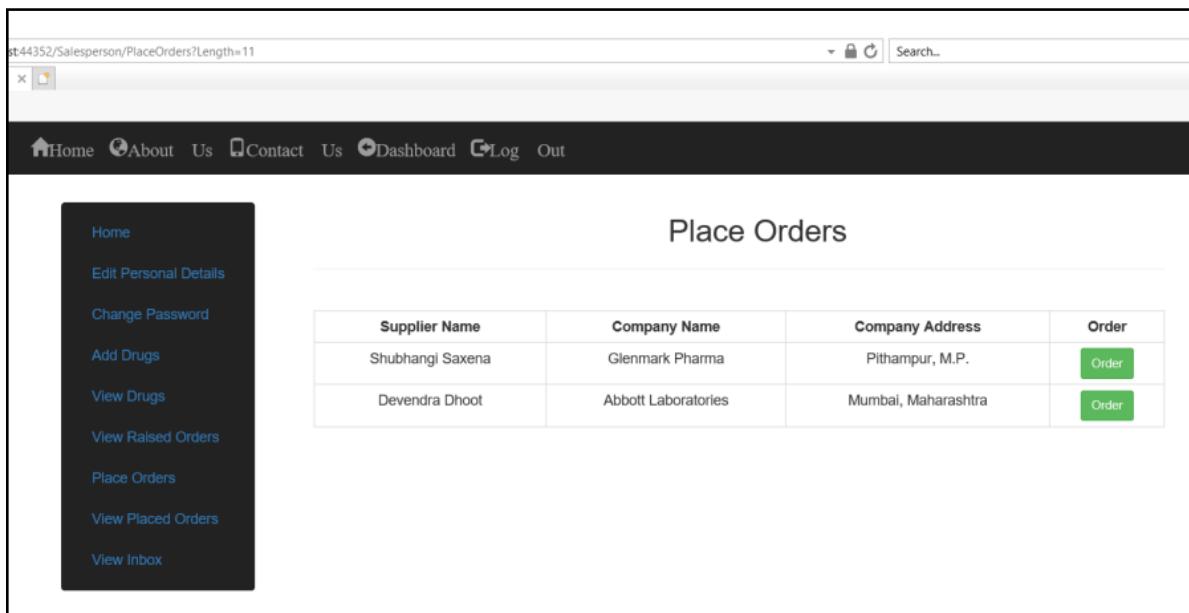


Fig: 4.24. Place Orders Page in Salesperson Module in clini-C-are

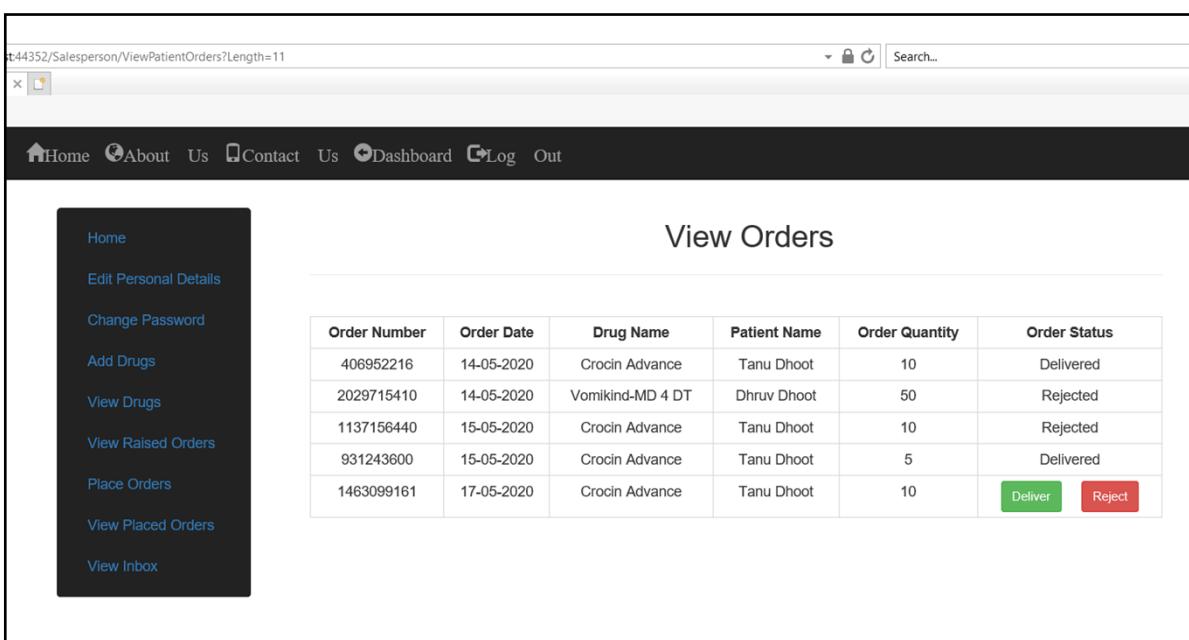


Fig: 4.25. View Patients' Orders Page in Salesperson Module in clini-C-are

• Supplier

The Supplier User has just the authority to Update the Salesperson's Orders and to Reply to his messages.

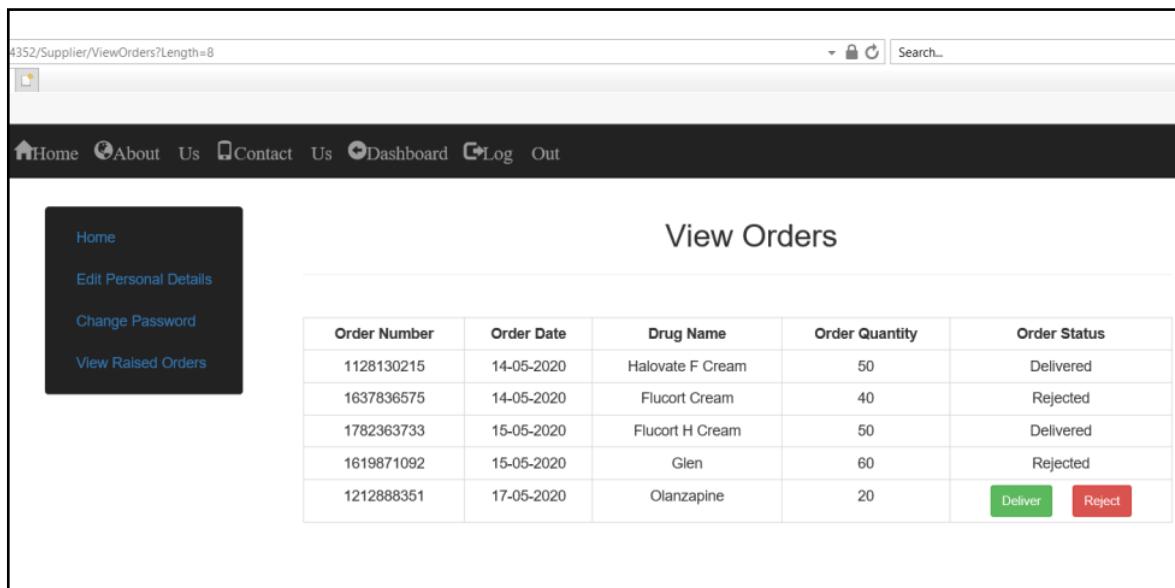


Fig: 4.26. View Orders Page in Supplier Module in clini-C-are

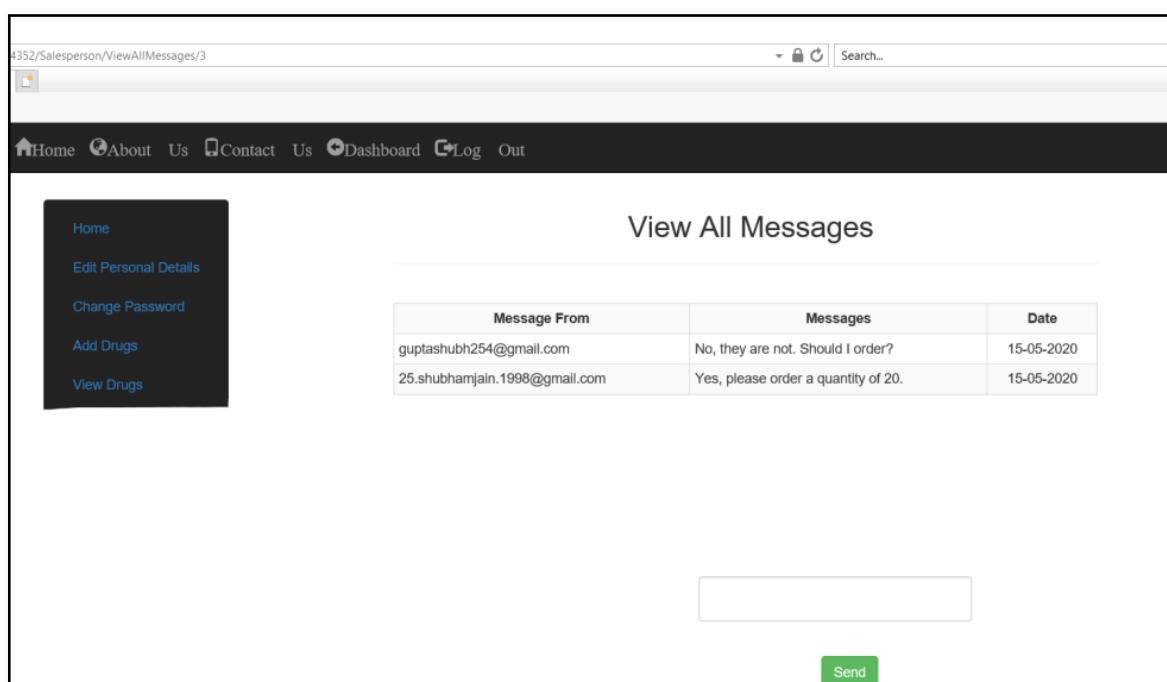


Fig: 4.27. View Messages Page in Supplier Module in clini-C-are

Along with these two pages are common for all the registered users:

- The Edit Personal Details Page is specifically designed differently for each registered user to accommodate their needs.

The screenshot shows a web browser window with the URL <http://127.0.0.1:44352/Physician/EditDetails?Length=9>. The page title is "Edit Personal Details". On the left, there is a sidebar menu with links: Home, Edit Personal Details (which is active and highlighted in blue), Change Password, View Appointments, Update Patient's History, Send Message, View Salesperson Inbox, and View Patient Inbox. The main content area has the title "Edit Personal Details". It contains the following form fields:

First Name	Sanjay
Last Name	Gujrati
Gender	Male <input checked="" type="radio"/> Female <input type="radio"/>
Total Experience	25
Education Details	MBBS, MD
Current Status	Active
Role	Cardiologist

Update

Fig: 4.28. Edit Personal Details of Physician in clini-Care

- The Change Password Page is a facility provided to every registered user to change their password.

The screenshot shows a web browser window with the URL <http://127.0.0.1:44352/Salesperson/ChangePassword?Length=11>. The page title is "Change Password". On the left, there is a sidebar menu with links: Home, Edit Personal Details (which is active and highlighted in blue), Change Password (which is also active and highlighted in blue), Add Drugs, View Drugs, View Raised Orders, Place Orders, View Placed Orders, and View Inbox. The main content area has the title "Change Password". It contains three input fields:

Enter Old Password	
Enter New Password	
Confirm New Password	

Change Password

Fig: 4.29. Change Password in clini-Care

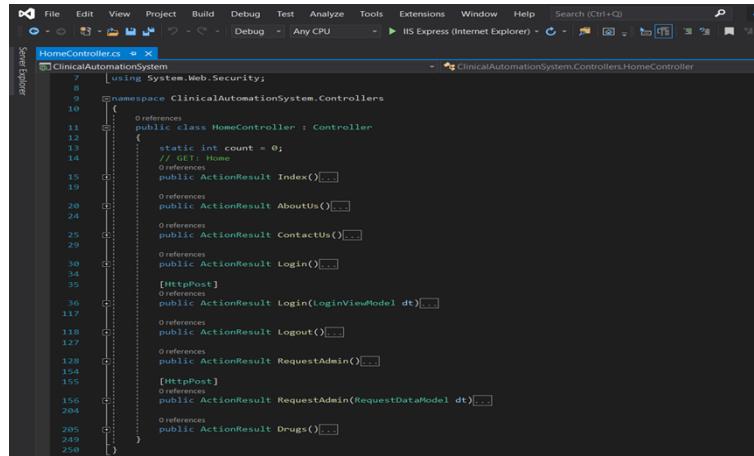
4.5 Reports

This project was assigned to us as a part of our pre-joining training with IBM. It was aimed to polish our skills on the platform and get us in the groove of the common working. The whole process went in a reverse chronological order where, first, our needs from the project were identified and then we went on creating the website.

The initial part of the project was to identify the basic needs and the workflow to build our project on. Thus, the project was initiated by constructing a SRS Specification Document. This was further margined by getting a view of what the end product must look like.

A significant and crucial part of this project was a database design that while holding all of the information, must be secure, and is good for both information storage as well as retrieval. Thus, a database was designed while taking the utmost care that it is able to store all the essential information needed for a smooth functioning of a clinic. This was then integrated with the project using the Entity Framework.

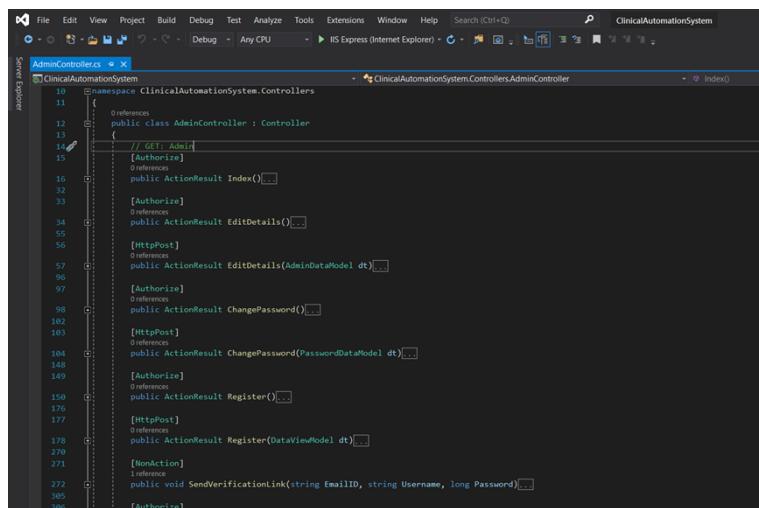
For the designing of the website, the development of GUI and the back-end functionalities went hand in hand as that is the beauty of the MVC Framework. The designing proceeded in individual steps and modules, by moving from the home page of the website, to the simple login and registration facilities, further proceeding to the designing of the home pages of each individual user, implementing the functionalities of the appointment module, the messaging module and the drug module. This helped our website to be recognized as a whole working project ready for deployment.



The screenshot shows the HomeController.cs file in Visual Studio. The code defines a HomeController class that inherits from Controller. It contains several action methods: Index, AboutUs, ContactUs, Login, Logout, RequestAdmin, and Drugs. The Login method is annotated with [HttpPost]. The Logout and RequestAdmin methods are also annotated with [HttpPost]. The RequestAdmin method takes a RequestDataModel parameter.

```
1  using System.Web.Security;
2
3  namespace ClinicalAutomationSystem.Controllers
4  {
5      public class HomeController : Controller
6      {
7          static int count = 0;
8          // GET: Home
9          public ActionResult Index()
10         {
11             count++;
12             return View();
13         }
14
15         public ActionResult AboutUs()
16         {
17             return View();
18         }
19
20         public ActionResult ContactUs()
21         {
22             return View();
23         }
24
25         [HttpPost]
26         public ActionResult Login(LoginViewModel dt)
27         {
28             if (ModelState.IsValid)
29             {
30                 // Process login logic here
31                 return RedirectToAction("Index");
32             }
33             else
34             {
35                 ModelState.AddModelError("", "Invalid login credentials");
36                 return View(dt);
37             }
38         }
39
40         [HttpPost]
41         public ActionResult Logout()
42         {
43             Session.Abandon();
44             return RedirectToAction("Index");
45         }
46
47         [HttpPost]
48         public ActionResult RequestAdmin(RequestDataModel dt)
49         {
50             // Process request admin logic here
51             return RedirectToAction("Index");
52         }
53
54         public ActionResult Drugs()
55         {
56             return View();
57         }
58     }
59 }
```

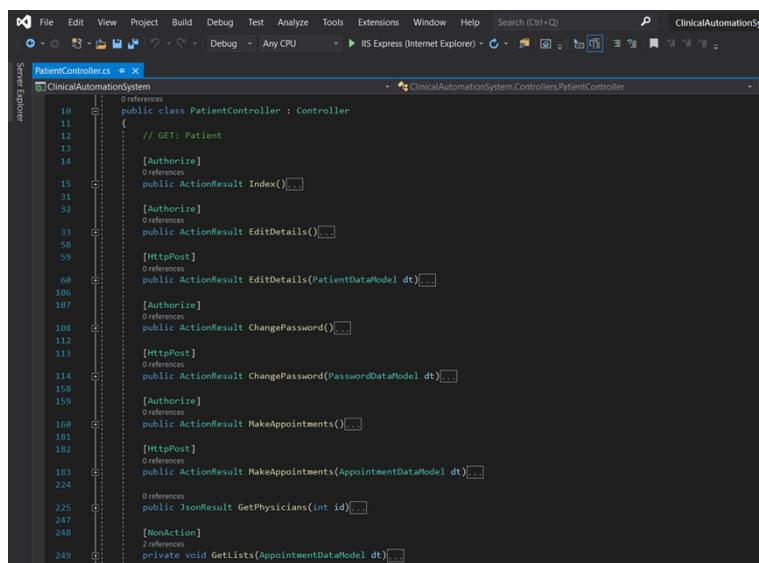
Fig. 4.30 Home Controller of clini-C-are



The screenshot shows the AdminController.cs file in Visual Studio. The code defines an AdminController class that inherits from Controller. It contains several action methods: Index, EditDetails, ChangePassword, Register, and SendVerificationLink. The Index method is annotated with [Authorize]. The EditDetails, ChangePassword, and Register methods are annotated with [HttpPost]. The Register method takes a DataViewModel parameter. The SendVerificationLink method is annotated with [NonAction].

```
1  using System.Web.Security;
2
3  namespace ClinicalAutomationSystem.Controllers
4  {
5      public class AdminController : Controller
6      {
7          // GET: Admin
8          [Authorize]
9          public ActionResult Index()
10         {
11             return View();
12         }
13
14         [Authorize]
15         public ActionResult EditDetails()
16         {
17             return View();
18         }
19
20         [HttpPost]
21         public ActionResult EditDetails(DataViewModel dt)
22         {
23             // Process edit details logic here
24             return RedirectToAction("Index");
25         }
26
27         [Authorize]
28         public ActionResult ChangePassword()
29         {
30             return View();
31         }
32
33         [HttpPost]
34         public ActionResult ChangePassword(PasswordDataModel dt)
35         {
36             // Process change password logic here
37             return RedirectToAction("Index");
38         }
39
40         [Authorize]
41         public ActionResult Register()
42         {
43             return View();
44         }
45
46         [HttpPost]
47         public ActionResult Register(DataViewModel dt)
48         {
49             // Process register logic here
50             return RedirectToAction("Index");
51         }
52
53         [NonAction]
54         public void SendVerificationLink(string EmailID, string Username, long Password)
55         {
56             // Send verification link logic here
57         }
58     }
59 }
```

Fig. 4.31 Admin Controller of clini-C-are



The screenshot shows the PatientController.cs file in Visual Studio. The code defines a PatientController class that inherits from Controller. It contains several action methods: Index, EditDetails, ChangePassword, MakeAppointments, GetPhysicians, and GetLists. The Index method is annotated with [Authorize]. The EditDetails, ChangePassword, and MakeAppointments methods are annotated with [HttpPost]. The MakeAppointments method takes an AppointmentDataModel parameter. The GetPhysicians method returns a JsonResult. The GetLists method is annotated with [NonAction] and takes an AppointmentDataModel parameter.

```
1  using System.Web.Security;
2
3  namespace ClinicalAutomationSystem.Controllers
4  {
5      public class PatientController : Controller
6      {
7          // GET: Patient
8          [Authorize]
9          public ActionResult Index()
10         {
11             return View();
12         }
13
14         [Authorize]
15         public ActionResult EditDetails()
16         {
17             return View();
18         }
19
20         [HttpPost]
21         public ActionResult EditDetails(PatientDataModel dt)
22         {
23             // Process edit details logic here
24             return RedirectToAction("Index");
25         }
26
27         [Authorize]
28         public ActionResult ChangePassword()
29         {
30             return View();
31         }
32
33         [HttpPost]
34         public ActionResult ChangePassword(PasswordDataModel dt)
35         {
36             // Process change password logic here
37             return RedirectToAction("Index");
38         }
39
40         [Authorize]
41         public ActionResult MakeAppointments()
42         {
43             return View();
44         }
45
46         [HttpPost]
47         public ActionResult MakeAppointments(AppointmentDataModel dt)
48         {
49             // Process make appointments logic here
50             return RedirectToAction("Index");
51         }
52
53         [NonAction]
54         public JsonResult GetPhysicians(int id)
55         {
56             // Get physicians logic here
57             return Json(new { id });
58         }
59
60         [NonAction]
61         private void GetLists(AppointmentDataModel dt)
62         {
63             // Get lists logic here
64         }
65     }
66 }
```

Fig. 4.32 Patient Controller of clini-C-are

A screenshot of the Visual Studio IDE showing the code for the PhysicianController.cs file. The code defines a controller for managing physicians. It includes methods for listing physicians, editing physician details, changing passwords, viewing appointments, updating appointment details, and updating physician history. The code uses [Authorize] attributes and [HttpPost] annotations.

```
19  public class PhysicianController : Controller
20  {
21      // GET: Physician
22      [Authorize]
23      public ActionResult Index()...
24
25      [Authorize]
26      public ActionResult EditDetails()...
27
28      [HttpPost]
29      public ActionResult EditDetails(PhysicianDataModel dt)...
30
31      [Authorize]
32      public ActionResult ChangePassword()...
33
34      [HttpPost]
35      public ActionResult ChangePassword>PasswordDataModel dt)>...
36
37      [Authorize]
38      public ActionResult ViewAppointment()...
39
40      [HttpPost]
41      public ActionResult UpdateAppointment(int? ID, string str)...
42
43      [Authorize]
44      public ActionResult History()...
45
46      [Authorize]
47      public ActionResult UpdateHistory(int? ID)...
48  }
```

Fig. 4.33 Physician Controller of clini-C-are

A screenshot of the Visual Studio IDE showing the code for the SalespersonController.cs file. The code defines a controller for managing salespeople. It includes methods for listing salespeople, editing salesperson details, changing passwords, adding drugs, and viewing drugs. The code uses [Authorize] attributes and [HttpPost] annotations.

```
10  public class SalespersonController : Controller
11  {
12      // GET: Salesperson
13      [Authorize]
14      public ActionResult Index()...
15
16      [Authorize]
17      public ActionResult EditDetails()...
18
19      [HttpPost]
20      public ActionResult EditDetails(SalesDataModel dt)...
21
22      [Authorize]
23      public ActionResult ChangePassword()...
24
25      [HttpPost]
26      public ActionResult ChangePassword>PasswordDataModel dt)>...
27
28      [Authorize]
29      public ActionResult AddDrugs()...
30
31      [HttpPost]
32      public ActionResult AddDrugs(DrugDataModel dt)...
33
34      [Authorize]
35      public ActionResult ViewDrugs()...
36
37      [Authorize]
38  }
```

Fig. 4.34 Salesperson Controller of clini-C-are

A screenshot of the Visual Studio IDE showing the code for the SupplierController.cs file. The code defines a controller for managing suppliers. It includes methods for listing suppliers, editing supplier details, changing passwords, viewing orders, and updating orders. The code uses [Authorize] attributes and [HttpPost] annotations.

```
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8
9  namespace ClinicalAutomationSystem.Controllers
10 {
11     public class SupplierController : Controller
12     {
13         // GET: Supplier
14         [Authorize]
15         public ActionResult Index()...
16
17         [Authorize]
18         public ActionResult EditDetails()...
19
20         [HttpPost]
21         public ActionResult EditDetails(SupplierDataModel dt)...
22
23         [Authorize]
24         public ActionResult ChangePassword()...
25
26         [HttpPost]
27         public ActionResult ChangePassword>PasswordDataModel dt)>...
28
29         [Authorize]
30         public ActionResult ViewOrders()...
31
32         [HttpPost]
33         public ActionResult UpdateOrder(int? ID, string str)...
34     }
35 }
```

Fig. 4.35 Supplier Controller of clini-C-are

Chapter – 5

Testing

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

5.1 Testing Objectives

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.^[6]

Testing can either be done manually or using an automated testing tool:

- **Manual** – This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager. Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.
- **Automated** - This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

5.2 Testing Scope

Scope of testing includes the process to determine all those features or functionality as one may say that will be considered for Testing during a particular level of testing in a particular release. The scope is determined during the Test planning phase where in the test plan we mention the scope of testing we will consider.

The project has two parts one is the user interface, i.e. the front-end and the back-end functionality and the other one is the database connectivity. The MVC Framework has view-model dependency and thus it was very important that the front-end was specifically designed to achieve proper functioning.

The back-end was tested with the help of Visual Studio Debugger to ensure that the front-end and back-end had proper connectivity. The functionalities of the back-end were checked every time a new module was implemented and thus, the proper functioning of the application was ensured.

5.3 Testing Principles

Testing is a process of executing a program with the aim of finding the error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

Testing systems that don't always return the same answers require new approaches. This is especially true when testing systems whose responses adapt to what they have learned from previous transactions. Software testing, in theory, is a fairly straightforward activity. For every input, there should be a defined and known output.

We enter values, make selections, or navigate an application and compare the actual result with the expected one. If they match, we nod and move on. If they don't, we possibly have a bug. The point is we already know what the output is supposed to be. Granted, sometimes an output is not well defined, there is some ambiguity, and you get disagreements on whether or not a particular result represents a bug.

5.3.1 Seven Principles in Testing

There are seven principles in testing:

- **Exhaustive testing is not possible:** It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case. It can test only some test cases and assume that software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc. and which is impractical.
- **Early Testing:** To find the defect in the software, early test activity shall be started. The defect detected in early phases of SDLC will be very less expensive. For better performance of software, software testing will start at initial phase i.e. testing will perform at the requirement analysis phase.
- **Defect Clustering:** In a project, a small number of the module can contain most of the defects. Pareto Principle to software testing states that 80% of software defects comes from 20% of modules.
- **Pesticide Paradox:** Repeating the same test cases again and again will not find new bugs. So, it is necessary to review the test cases and add or update test cases to find new bugs.
- **Testing is Context Dependent:** Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the Android application.
- **Absence of Errors Fallacy:** If built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfill all the customer requirements.
- **Testing Shows Presence of Defects:** The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks

about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it cannot prove that software is defects free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not removes all defects.

5.4 Testing Methods used

5.4.1 Unit Testing

Unit Testing is to make sure that the module that has been developed can run smoothly and is executable without any bugs or system error. Unit testing will basically test on a particular part or unit, such as functions, procedures, or modules. It is to ensure that the particular function that has been developed can fulfill the requirement of the user and that too without any error. Unit testing is normally performed by the programmer who writes the code. Every time after they finish writing the code, they will run the debugger and test whether the function of source code can run smoothly or not. If there are bugs or system errors, the developer will try to solve them and run the debugger again. If the function can run smoothly without error and can meet the required function, the developer will test the whole module all over again.^[6]

5.4.2 Validation Testing

Validation Testing is one kind of testing which will test on whether the system is able to handle the wrong data entered by the user or not. It is to ensure that the data entered by the user is relevant and in the correct format or correct data types. For this testing, we will test whether the system can handle the data that is entered by the user. This testing is very important because the wrong data entered may cause the system error or accidentally insert the wrong data into the database and mislead the system.^[6]

5.5 Test Cases

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Check User's Login Credentials	Check whether the login credentials for the user are valid.	Prompt out error message that credentials are invalid.	Prompt out the error message.	Pass
Login User	Redirecting user to their respective dashboard after login.	Redirect user to their respective dashboard.	Redirects User.	Pass
Request Admin	Request Admin to register them as a user.	Successfully add the data into the RequestAdmin table.	Successfully added the data into the RequestAdmin table.	Pass
View Drugs	Check whether the system can list out all the drug record.	Display all the drug information retrieved from the database.	Display all the drug information.	Pass

Table 5.1 Testing in Home Controller

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Edit Profile	Check whether the profile can be successfully added and edited.	Successfully add and update the data in the Admin table.	Successfully added and updated the data in the Admin table.	Pass
Change Password	Check whether the login password can be successfully edited.	Successfully change the password.	Successfully updated the password column in the User table.	Pass
Register User	Check whether the user is registered	Successfully add the data into the User table.	Successfully added the data into the User table.	Pass
Send E-mail	Check if E-mail is sent to registered user	Successfully send E-mail.	E-mail successfully sent.	Pass
View Requests	Check whether the system can list out all the requests record.	Display all the requests retrieved from the database.	Displays all the requests.	Pass
View Users	Check whether the system can list out all the users according to their respective roles.	Display all the users according to their respective roles.	Displays all the users according to their respective roles.	Pass
Delete Users	Check whether the user record can be deleted.	Successfully delete the user record.	Successfully removed the user record from the database.	Pass
View Drugs	Check whether the system can list out all the drug record.	Display all the drug information retrieved from the database.	Displays all the drug information.	Pass
Delete Drugs	Check whether the drug record can be deleted.	Successfully change the IsDeleted column to true in the drug record.	Successfully changed the IsDeleted column in the database.	Pass

Table 5.2 Testing in Admin Controller

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Edit Profile	Check whether the profile can be successfully added and edited.	Successfully add and update the data in the Patient table.	Successfully added and updated the data in the Patient table.	Pass
Change Password	Check whether the login password can be successfully edited.	Successfully change the password.	Successfully updated the password column in the User table.	Pass
Request Appointment	Check whether the new appointment can be added into database.	Successfully add the data into the Appointment table.	Successfully added the data into the Appointment table.	Pass
View Appointment	Check whether the system can list out the patient's appointments record.	Display the patient's appointments retrieved from the database.	Displays the patient's appointments.	Pass
Send Messages	Check whether a new message can be added in the database.	Successfully add the data into the Inbox table.	Successfully added the data into the Inbox table.	Pass
View Messages	Check whether the system can list out the patient's messages.	Display all the patient's messages retrieved from the database.	Displays all the patient's messages.	Pass
Order Drugs	Check whether a new order can be added in the database.	Successfully add the data into the PatientOrderDetails table.	Successfully added the data into the PatientOrderDetails table.	Pass
View Orders	Check whether the system can list out all the patient's orders.	Display all the patient's orders retrieved from the database.	Displays all the patient's orders retrieved from the database.	Pass

Table 5.3 Testing in Patient Controller

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Edit Profile	Check whether the profile can be successfully added and edited.	Successfully add and update the data in the Physician table.	Successfully added and updated the data in the Physician table.	Pass
Change Password	Check whether the login password can be successfully edited.	Successfully change the password.	Successfully updated the password column in the User table.	Pass
View Appointment	Check whether the system can list out the physician's appointments record.	Display the physician's appointments retrieved from the database.	Displays the physician's appointments.	Pass
Update Patient's History	Check whether the History of the patient can be updated.	Successfully update the data in the History column of the Patient's table in the database.	Successfully updated the data in the History column of the Patient's table in the database.	Pass
Send Messages	Check whether a new message can be added in the database.	Successfully add the data into the Inbox table.	Successfully added the data into the Inbox table.	Pass
View Messages	Check whether the system can list out the patient's messages.	Display all the patient's messages retrieved from the database.	Displays all the patient's messages.	Pass

Table 5.4 Testing in Physician Controller

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Edit Profile	Check whether the profile can be successfully added and edited.	Successfully add and update the data in the Salesperson table.	Successfully added and updated the data in the Salesperson table.	Pass
Change Password	Check whether the login password can be successfully edited.	Successfully change the password.	Successfully updated the password column in the User table.	Pass
Add Drugs	Check whether the drug information can be added into database.	Successfully add the data into the Drug table.	Successfully added the data into the Drug table.	Pass
View Drugs	Check whether the system can list out the drug records.	Display the drug records retrieved from the database.	Displays the drug records.	Pass
Edit Drugs	Check whether the drug record can be successfully edited.	Successfully update the data in the Drug table.	Successfully updated the data in the Drug table.	Pass
Delete Drugs	Check whether the drug record can be deleted.	Successfully change the IsDeleted column to true in the drug record.	Successfully changed the IsDeleted column to true.	Pass
Send Messages	Check whether a new message can be added in the database.	Successfully add the data into the Inbox table.	Successfully added the data into the Inbox table.	Pass
View Messages	Check whether the system can list out the patient's messages.	Display all the patient's messages retrieved from the database.	Displays all the patient's messages.	Pass
Order Drugs	Check whether a new order can be added in the database.	Successfully add the data into the PatientOrderDetails table.	Successfully added the data into the PatientOrderDetails table.	Pass

View Raised Orders	Check whether the system can list out all the patient's orders.	Display all the patient's orders retrieved from the database.	Displays all the patient's orders retrieved from the database.	Pass
View Orders	Check whether the system can list out all the patient's orders.	Display all the patient's orders retrieved from the database.	Displays all the patient's orders retrieved from the database.	Pass

Table 5.5 Testing in Salesperson Controller

Test Name	Test Objective	Expected Result	Actual Result	Remark
Validation	To check whether the validation will be executed if the required fields are blank or wrong data type.	Prompt out error message to inform the user that the blank fields or incorrect data types are entered.	Prompt out the error message.	Pass
Edit Profile	Check whether the profile can be successfully added and edited.	Successfully add and update the data in the Salesperson table.	Successfully added and updated the data in the Salesperson table.	Pass
Change Password	Check whether the login password can be successfully edited.	Successfully change the password.	Successfully updated the password column in the User table.	Pass
Send Messages	Check whether a new message can be added in the database.	Successfully add the data into the Inbox table.	Successfully added the data into the Inbox table.	Pass
View Messages	Check whether the system can list out the patient's messages.	Display all the patient's messages retrieved from the database.	Displays all the patient's messages.	Pass
View Raised Orders	Check whether the system can list out all the patient's orders.	Display all the patient's orders retrieved from the database.	Displays all the patient's orders retrieved from the database.	Pass

Table 5.6 Testing in Supplier Controller

5.6 Sample Test Data and Results

There are many test cases for the various functioning of this application. Some test cases are listed below:

- Login fails if the Login credentials for any user are invalid.

The screenshot shows a web browser window with the following details:
URL: http://127.0.0.1:44352/Home/Login
Page Title: Log In
Form Fields:

- UserName: admin
- Password: *****

A green "Log In" button is present. Below the button, a red error message is displayed: "Login Failed. Invalid E-mail ID or Password. Two More Attempts Left."

Fig. 5.1 Login Failed

- Validation check is done for all the required fields for every input-output page.

The screenshot shows a web browser window with the following details:
URL: http://127.0.0.1:44352/Admin/Register
Page Title: Registration
Left Sidebar:

- Home
- Edit Personal Details
- Change Password
- Register Users
- View Requests
- View Patients
- View Physicians
- View Suppliers
- View Salespersons
- View Drugs

Main Content:

Registration

Required Fields (with validation messages):

- User Name: Please Enter Your User Name
- E-mail ID: Please Enter Your E-mail ID
- Role: Please Enter the Role

Submit Button

Message: Registration Unsuccessful.

Fig. 5.2 Verification in Registration Page

Fig. 5.3 User Successfully Registered

- Various Validations have been provided with the back-end to check various constraints.

Fig. 5.4 Date Check Constraint on the Manufacturing and Expiry Date of Drugs

Fig. 5.5 Message Successfully Sent

Chapter - 6

Limitations

The project has been developed on ASP.NET and therefore can be deployed only on a Microsoft Windows system. So for a user to be able to deploy this project it becomes mandatory to have a licenced Windows system. This limitation also raises scalability issues in the system as when more functionalities are added to the system and the system data grows in volume more licensing of software and infrastructure will be required. ASP.NET cannot be installed in the older versions of windows and requires the very latest version.

The ASP.NET compiler is quite unstable and sometimes gives false exceptions and errors for the same code that was compiled successfully. This creates a lot of problems in maintenance and the system has to continuously monitored and updated to check for any new errors that can arise.

The current project does not have a payment API integrated to it and any payment will have to be done in cash. An online transaction API can be added to it in the future.

In the current system, the filter based search system is such that certain combinations of filters produce blank results. This can be a limitation as this can portray the clinic in bad light as to having lesser than required manpower.

The front-end of the project is quite simple now. It can be improved to give a more comforting and warming feel to both the patients and the clinic authorities.

Chapter - 7

Future Scope

There are a lot of functionalities that can be developed in this project in the future. This project does not currently have a payment interface linked to it and a payment API can be added to avoid the use of cash.

Due to the ongoing COVID-19 situation in the world there is a widespread fear among the people to go out in public places and specially clinics and hospitals where there are high chances of infection. This application can be integrated in the future with video call application like “Twilio” to provide online consultation to patients via video call for a fixed time basis.

This app can be upgraded to double to serve as a tele-medicine service where the user can order home delivery of drugs by for both prescription and generic drugs.

There is also a going demand among pet owners for an online consultation services for their pets. This app can be upgraded to include services for veterinary doctors so that pet owners can upgrade their accounts from regular accounts to premium accounts and get consultations for their pets also.

Chapter – 8

Learning After Training

8.1 Learning During Training

WEEK	TASK
Week 1	Learnt the basics of programming, Algorithms and Data Structures, RDBMS Concepts, SQL Queries
Week 2	Learnt HTML, CSS, JavaScript .NET Framework 4.5 C# Introduction
Week 3	Core C# Programming: Object Oriented Concepts, Exception Handling, Garbage Collection, Delegates, Interfaces, Type Specification, etc.
Week 4	C# Continued: ADO.NET Usage, LINQ Expressions, CRUD Functionality Implementation
Week 5	Logging, Exception Handling, .NET Entity Framework Introduction
Week 6	Learnt ASP.NET Framework, Developed Web Application for Employee Registration using ASP.NET
Week 7	Learnt ASP.NET MVC, Developed Web Application on ASP.NET MVC Architecture, Software Development Methodology
Week 8	Case Study (Project on Clinical Automation System)

Table 8.1 Weekly Schedules

8.2 Learning After Training

IBM is a premier technology giant working in the field of software development. This training was the pre-joining training of IBM and the project was developed as a part of that training.

Working on this project gave me deep insight about how things move in a professional software firm. While working in an IT company is fun but it equally challenging too.

While working on the development of this project I got plenty of opportunities to hone my coding skills and use my knowledge to solve software engineering problems as well as learn new things that are very useful in any software development project.

This project gave me deep insights about working on the MVC model and how breaking up the project in modules helps in fast development of the project which utilizes the talent of each member of the team to the best possible extent and develop a project that meets the demand of the project in the best possible manner.

On the technical front this project taught me a great deal about managing various profiles in a single web application, messaging using server programming in Web-based Applications and also the benefits of developing the project in modules.

Developing the project in modules not only helps in simplifying the development process but also error identification and error correction becomes easier. Also error in one module of the software does not hinder the development of the other modules.

While the technical learning greatly enhanced my skills, I also got to experience all the phases of the software development life cycle. I got the hands on experience of documenting the requirements as well as of design and development. Working under a strict deadline set by IBM and college also taught me how to use various SDLC models to my benefit. In this case the waterfall model came to be very useful as the requirements were documented early on and were never changed and also there was no time available to send the project back to development so the waterfall model proved to be very useful in making sure that I created everything correctly in the very first attempt.

Chapter - 9

Conclusion

This training was carried out in two parts. The first part was a formal training given to me by IBM and the second part consisted of developing a project based on what we have learnt.

I was trained in Microsoft SQL Server, HTML, CSS, JavaScript, and C# programming fundamentals, ADO.NET, LINQ, ASP.NET, MVC Framework using Visual Studio 2019 by IBM in the duration of the training provided to us by them and I was then assigned to develop this project on my own from the very beginning of the SDLC phases. This project was developed entirely by me from documenting the requirements to testing the final product.

This opportunity has given me the chance to work on SDLC on my own and develop a better understanding of how a project is developed.

clini-C-are is a Clinic Automation System that allows digitalizing the daily operations of a regular clinic through computer intervention. This provides a common platform for the patient to connect directly with the physician and take appointments and visit the clinic at a specified time. This reduces the waiting time of the patient and also reduces the load on the clinic infrastructure.

During the time of a global pandemic where the world is dealing with a highly infectious disease, this application becomes even more important as it allows the users to get an access to physicians and specialists from their homes. This adds to the societal impact this application will have in the upcoming days and in addition to the business impact.

Overall, the development of this project was very satisfactory to me both as an individual and a professional.

Chapter - 10

Bibliography and References

10.1 Bibliography

- [1] Simons, E., n.d. Clinic Automation System - High Level Design (Microsoft) V1.0. 1st ed. IBM, pp.1-8.
- [2] Simons, E., n.d. Clinic Automation System - Low Level Design (Microsoft) V1.0. 1st ed. IBM, pp.1-13.
- [3] Simons, E., n.d. SRS Clinic Automation_BRD. 1st ed. IBM, pp.1-6.
- [4] Orfali, R., Harkey, D. and Edwards, J., 1999. Client/Server Survival Guide. 3rd ed. John Wiley & Sons, Inc.
- [5] MacDonald, M., 2012. Beginning ASP.NET 4.5 In C#. Apress.
- [6] Tutorialspoint.com. 2020. Tutorails Point Online Web Tutorials. [online] Available at: <<https://www.tutorialspoint.com/index.htm>> [Accessed 18 May 2020].
- [7] Ben-Gan, I., 2008. Microsoft SQL Server 2008 T-SQL Fundamentals. 2nd ed. Microsoft Press.
- [8] W3schools.com. 2020. W3schools Online Web Tutorials. [online] Available at: <<https://www.w3schools.com/>> [Accessed 8 May 2020].
- [9] Docs.microsoft.com. 2020. System Namespace. [online] Available at: <<https://docs.microsoft.com/en-us/dotnet/api/system?view=dotnet-plat-ext-3.1>> [Accessed 8 March 2020].
- [10] Docs.microsoft.com. 2020. System.Linq Namespace. [online] Available at: <<https://docs.microsoft.com/en-us/dotnet/api/system.linq?view=netcore-3.1>> [Accessed 20 April 2020].
- [11] Docs.microsoft.com. 2020. System.Web.Mvc Namespace. [online] Available at: <<https://docs.microsoft.com/en-us/dotnet/api/system.web.mvc?view=aspnet-mvc-5.2>> [Accessed 25 April 2020].
- [12] Docs.microsoft.com. 2020. System.Net Namespace. [online] Available at: <<https://docs.microsoft.com/en-us/dotnet/api/system.net?view=dotnet-plat-ext-3.1>> [Accessed 10 May 2020].
- [13] Berg, M., 1999. Accumulation and coordinating: Occasions for information technology in medical work. 1st ed., pp. 373-401

