MEDICARE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

NIGY ANTONY

Reg.No: SJC17MCA-D030

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for
the award of the Degree

of

MASTER OF COMPUTER APPLICATIONS

Department of Computer Science And Applications

ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY

Bharananganam Pravithanam Road, Kottayam, Palai,
Choondacherry, Kerala 686579

April, 2019

DECLARATION

I undersigned hereby declare that the project report " **MEDICARE MANAGEMENT SYSTEM**" , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Mr. Anish Augustine**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Palai **NIGY ANTONY**

Date: **Reg.No: SJC17MCA-D030**

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
ST.JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY
Palai, Choondacherry Kottayam, 686579
(*Approved by AICTE and affiliated to APJ Abdul Kalam Technological University)*

CERTIFICATE

This is to certify that the report entitled "MEDICARE MANAGEMENT SYSTEM" submitted by "**NIGY ANTONY, Reg.No: SJC17MCA-D030**" to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master in Computer Applications is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mr.Anish Augustine
Internal Guide

Mrs. Soumya George
Project Co-ordinator

Dr.T.D.Jainendrakumar
Head of the Department

Viva-voce held on:..

External Examiner 1: External Examiner 2:

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgment, then let words play the heralding role in expressing my gratitude. To bring something into existence is truly a work of God. I would like to thank God for not letting me down and showing me the silver lining in the dark clouds.

I would like to thank **Dr. J David**, Principal, St.Joseph's College Of Engineering And Technology for his support and encouragement. I convey my heartfelt thanks to **Dr.T.D.Jainendrakumar** (Head of the Department of Master of Computer Applications, St.Joseph's College Of Engineering And Technology ,) for providing an opportunity for the project presentation. It is my pleasure to express my gratitude to the project coordinator **Mrs.Soumya George**, Department of Computer Applications, St.Joseph's College Of Engineering And Technology whose support and constructive criticism has led to the successful completion of the task.

With the biggest contribution to this report, I would like to thank **Mr.Anish Augustine**, Department of Computer Applications who had given me full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the this work.

I would also thank my institution and faculty, my family and friends without whom this project would have been a distant reality.

Nigy Antony

ABSTRACT

This system is designed to coordinate and integrate all the inherent activities involved in the management and running of a medicare in a college. Admin will register the authorized staffs and students. Each student have a unique identification number. Students and faculties can use this site and they can view his case history. The medicare is handled by a nurse. Students can ask and clear doubts through this site. The nurse provide basic treatments for the students and faculties. Doctor can add case history of patients, View questions from patients and answer it. If a patient admitted by seriously injured, using his identification number doctor can find his previous case history like any allergy, previous surgery details if any etc.. And the doctor can refer the patients to the hospitals if it is necessary.

The system consist of four modules:-

1. Admin

2. Doctor

3. Nurse

4. User

The system contains four users, administrator, doctor, nurse and students. Role of admin is to register the authorized staffs and students. The students can upload questions and view the response in the system. Nurse can view the questions that are uploaded by the students and can delete those questions which are seen irrelevant. Doctor can respond to the questions and view the case history of the patients.

CONTENTS

Contents PageNo.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

I.1 PROBLEM DEFINITION

This project is aimed to developing a Medicare System as responsive interface system. This system can be used to store the .The existing Medicare System is the system which under the study for the project that which going through many types of above specified processes like storing medicine details, case history, daily report etc. all these are manually.

This system will help the students to take printout all case histories if necessary. And they can send feedback to the administrator .The system provides the facility to ask questions to the doctor about their treatment. If a patient admitted by seriously injured, using his identification number doctor can find his previous case history like any allergy, previous surgery details if any etc.. And the doctor can refer the patients to the hospitals if it is necessary.

I.2 ABOUT THE ORGANIZATION

The college was founded by a group of well-known academics. They are pioneering educators, having unmatched experience in the field of education with a belief that the continuous search for knowledge is the sole path to success. The Primary focus of the institution is to expose the young minds to be world of technology, instilling in them confidence and fortitude to face new challenges that enables them to excel in their chosen field. The college inculcates the development of all facets of the mind culminating in an intellectual and balanced personality. Our team of dedicated and caring faculty strives to widen the students' horizon of learning thereby achieving excellent results for every student.

St. Joseph's College Of Engineering And Technology, right from inception, has been maintaining high levels of standard in academic and extracurricular realms of activities. We offer BTECH degree courses in 6 engineering disciplines, and Master's Degree courses in Engineering, Computer Application and Business Administration. In the short span of a decade of its existence and among the six batches of students that have graduated, the college bagged several university ranks and has a remarkably high percentage of pass. The students of batch of mca bagged ranks in the university. The college is also the venue of national and state level seminars and symposiums and has emerged as the hub of technical education in the state. The placement scenario is also quite commendable, with several premier industries visiting for St. Joseph’s College Of Engineering And Technology placement and recruitment.

I.3 OBJECTIVE OF THE PROJECT

This system has developed for managing all the activities in a College Medicare, where user can easily search their previous case history as their need. Each user of the system has role and permission according to their role. This system also allow the students to submit their doubts about the treatment and the nurse can approve or reject the doubts. The nurse can respond to the questions from students and can forward the questions to the doctor if it is necessary. The students can take printout of the case histories for any insurance, or for further treatments.

CHAPTER II

INITIAL INVESTIGATION AND FEASIBILITY STUDY

II.1 INITIAL INVESTIGATION

The purpose of this document is to give a clear picture of the module designs of the project Medicare. The website provide an easy way to access case histories and can take their printouts. The registered nurse can approve or reject the questions submitted by the students and the nurse can respond to the questions and nurse can forward the questions to the doctor if it is necessary. This document is developed after a number of consultations with the nurse and doctor considering the complete requirement specifications of the given project. This project also helps to understand various functionalities of the modules in the project as well as it gives a pictorial design of how the website will look like with its functionalities working together various to achieve the requirements.

II.2 EXISTING SYSTEM

The study of the existing system is a pre-requisite for developing any software system. The study of the system reveals many features of the existing system. This gives analyst an insight into the working of the system and helps the developer to design an appropriate system, which will eliminate the many limitations present in the existing system.

Limitation of the Existing System are:-

- It is a manual system
- Process is by means of paperwork
- Difficult to keep all the paper records
- The file manipulating method was not done in a centralized manner.
- Document storing, accessing them takes more time
- Searching process is mainly done manually and it is difficult
- Chances of loss in document containing important details.
- Difficult to find out accurate data in minimal time
- Time consuming and mishandling of reports.

II.3 PROPOSED SYSTEM

The proposed system computerization is developed using SQLite server as back-end and Python django as front-end. The Django framework is managed, type safe environment for application, development and execution. The software is developed as a simulated system and the complex procedures are avoided to make the system easy to use. The proposed system is user friendly and has simplicity and security. In the proposed system the data redundancy can be avoided to certain extend and the data consistency can be maintained. The record keeping and searching process are easy.

II.3.1 Advantages of the Proposed System

- Give solution to the current system problems
- Less time consuming and more efficient
- Result will be very precise and accurate
- Easy to use and fast
- Simple user interface to reduce processing time
- Easy searching and storing documents
- Eliminate chances for errors and reduce effort

II.3.2 Features of the Proposed System

The various features of proposed system are as follows:

- Access to the system and database as per user identification
- The maximum security ensured
- Integrity reliability and integrity of data
- User-friendly and flexible in all aspects
- Data entry updates is quite easy
- Effective table manipulation as facilitated by the rich SQL
- Good validation checking

- Easy maintenance
- Removes chances of leakage of information.
- Provides a better record keeping system

All these form the major aspects and advantages of the proposed system. Provision is made for effective improvements of maintenance are needed at any stage.

II.4 FEASIBILITY STUDY

During system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the organization. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product. While evaluating the existing system, many advantages and disadvantages raised. Analyzing the problem thoroughly forms the vital part of the system study. Problematic areas are identified and information is collected.

The benefits of this site are users can easily interact and get the services without much complexity. It helps to make it possible that more users can interact with the site at a time. Feasibility study is to determine whether the proposed system is technically, economically and behaviourally feasible in all respects.

The main aim of feasibility study is to evaluate alternative site and propose the most feasible and desirable site for development. If there is no loss for the organization then the proposed system is considered financially feasible. A feasibility study is carried out to select the best system that meets performance requirements.

The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behaviour of the system.

In this scenario, problems are identified. Essential data are being gathered for the existing problems. It is necessary that this analysis familiarizes the designer with objectives, activities, and the function of the organization in which the system is to be implemented. The feasibility study was divided into four:- Technical, Economical, Operational and Behavioural. It is summarized below:-

II.4.1 TECHNICAL FEASIBILITY

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs, are identified. While considering the problems of existing system, it is sufficient to implement the new system. The proposed system can be implemented to solve issues in the existing system. It includes the evaluation of and how it meets the proposed system. This system use Python django as front end technology and SQLite Server as back end technology.

II.4.2 ECONOMIC FEASIBILITY

Economic analysis is most frequent used for evaluating of the effectiveness of the candidate system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a candidate system and compare them with the existing system. Except for the initial capital amount and the amount after each financial year, no other huge amount is needed. The expenses can be handles by any participants. So, the system is economically feasible.

This feasibility involves some questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same resources. Here there is no problem. This firm has fully equipped hard ware, and fully fledged software, so no need to spend money on these issues. And as the client and the developer are one, there is no further problem in economic issues.

II.4.3 OPERATIONAL FEASIBILITY

Methods of processing and presentation are all according to the needs of clients since they can meet all user requirements here. The proposed system will not cause any problem under any circumstances and will work according to the specifications mentioned. Hence the proposed system is operationally feasible.

People are inherently resistant to change and computer has been known to facilitate changes. The system operation is the longest phase in the development life cycle of a system. So, Operational Feasibility should be given much importance. This system has a user-friendly interface. Thus it is easy to handle.

II.4.4 BEHAVIORAL FEASIBILITY

In today's world, computer is an inevitable entity. As per the definition of behaviour design, many valid points are recognized in this study. This system behaviour changes according to different environment. In order to ensure proper authentication and authorization and security of sensitive data of the admin or employers, login facilities are provided. These are the main feasibility studies tested in this application.

CHAPTER III
SYSTEM ANALYSIS AND DESIGN

III.1 SOFTWARE REQUIREMENT SPECIFICATION

The primary goal of the system analyst is to improve the efficiency of the existing system.For that study of specification of the requirement is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. An investigation is done whether the up gradation of the system into an application program could solve the problems and eradicate the inefficiency of the existing system. This gives an idea about the system specifications required to develop and install the project ”MEDICARE”.

The System Requirements Specification is based on the System Definition. The requirement specifications are primarily concerned with functional and performance aspect of a software product and emphasis are placed on specifying product characteristics implying how the product will provide those characteristics. One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements. This selection summarizes the application requirement.

III.1.1 HARDWARE REQUIREMENT

- CPU - INTEL(R) PENTIUM(R)
- HARD DISK SPACE -500 GB
- RAM -4 GB
- DISPLAY -19 STANDARD RATIO LCD MONITOR
- KEYBOARD -99-104 KEYS
- CLOCK SPEED -1.99 GHZ

III.1.2 SOFTWARE REQUIREMENT

- OPERATING SYSTEM -WINDOWS 8

- WEB SERVER -IIS 7.5
- FRONT END – PYTHON DJANGO

- BACK END -SQLITE

III.2 SYSTEM DESIGN

Designing the system in an effective way leads to the smooth working of any software's. System design is the process of developing specification for a candidate system that meet the criteria established in the system analysis. Major step in the system design is the preparation f the input forms and output reports in a form applicable to the user. The main objective of the system design is to use the package easily by any computer operator. System design is the creative act of invention, developing new inputs, and database, off-line files, method, procedure and output for processing business to meet an organization objective. System design builds information gathered during the system analysis. This system is designed neatly so that user will never get ambiguity while using the system.

III.2.1 NON-FUNCTIONAL REQUIREMENTS

Performance Requirements

For the efficient performance of the application, network must have high bandwidth so that the task of centralized management does not lead to network jam. Also the hard disk capability must be high so that data can be effectively stored and retrieved.

Security Requirements

Security requirements of this application involves authentication using user name and password so that invalid users are restricted from data access. For the security of data, periodic database backups must be performed so that we can recover data in the case of data loss.

III.3 UNIFIED MODELING LANGUAGE [UML]

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar JAcobson and the RationalSoftware Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the Object Management Group(OMG) as the standard for modelling software development.UML stands for Unified Modeling Language.UML 2.0 helps extend the original UML specification to cover a wider portion of software development efforts including agile practices.Improved integration between structural models like class diagrams and behavior models like activity diagrams.The original UML specified nine diagrams; UML 2.x brings that number up to 13. The four new diagrams are called: communication diagram, composite diagram, interaction overview diagram and timing diagram. It also renamed state chart diagrams to state machine diagrams , also known as state diagrams.

Types of UML diagram

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing and deployment. These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagrams.

Structural UML diagrams

- Class diagram
- Package diagram
- Object diagram

- Component diagram
- Composite structure diagram
- Deployment diagram

Behavioral UML diagrams

- Activity Diagram
- Sequence diagram
- Use case diagram
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

III.3.1 Use case Diagram

To model a system the most important aspect is capture the dynamic behaviour. To modify a bit in details, dynamic behaviour of the system when it is running or operating. So only behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagram consists of actors, use case and their relationships. The diagram is used to model the system of an application. A single use case diagram captures a particular functionality of a system.

Use case Diagram objects:

- Actor
- Use case
- System
- Package Actor

Actor

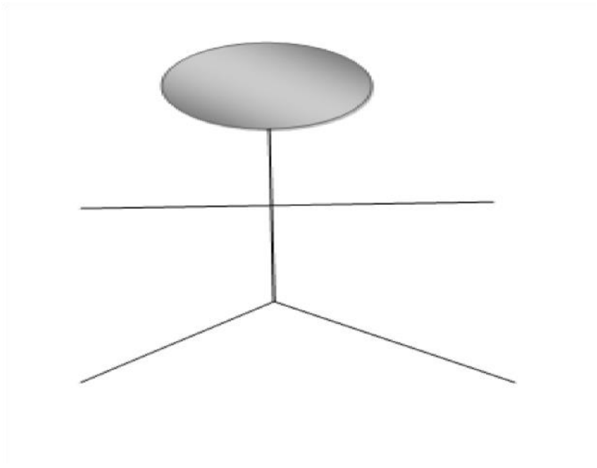


Figure III.1: Actor

Actor is a use case diagram in an entity that performs a role in one given system. This could be a person, organization or an external system usually drawn like skeleton.

Use case

A use case represents a function or an action within the system. Its drawn as an oval and named with the function.

System

Figure III.2: Use Case

System is used to define the scope of the use case and drawn as a rectangle. This is an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

Package

Package is another optional element that is extremely useful in complex diagrams. Similar to use class diagrams, packages are used to group together use cases.

The following is the UML diagram of this system:-

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system.Time is represented in the vertical direction showing the sequence of interaction of the header elements.

Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

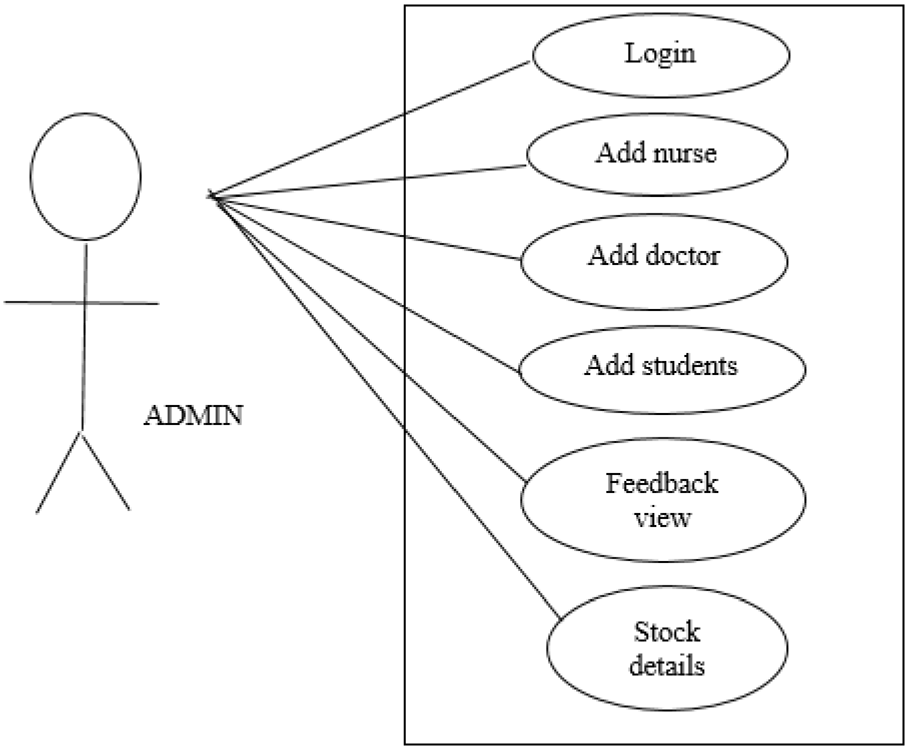


Figure III.3: UML DIAGRAM FOR ADMIN

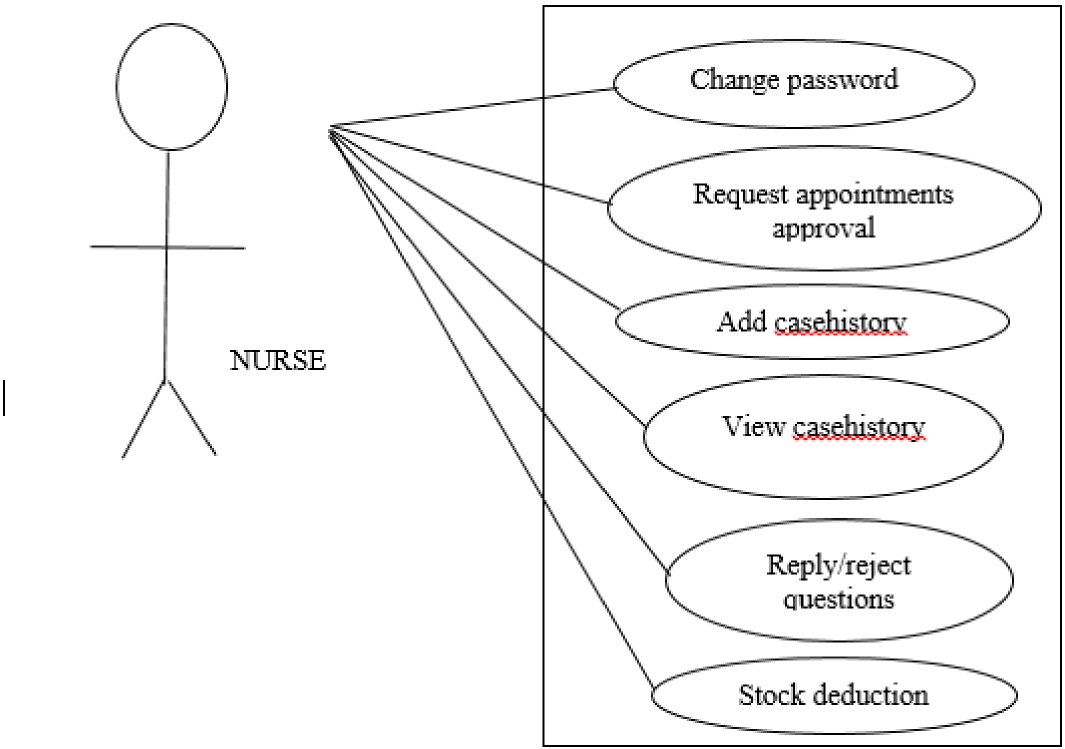
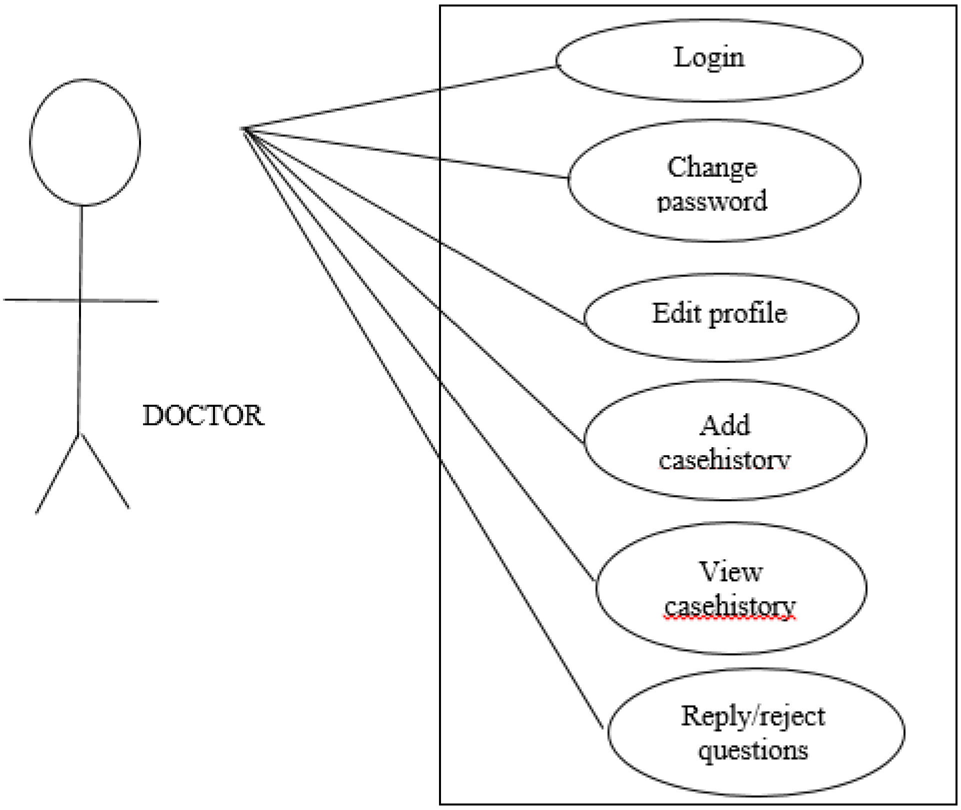
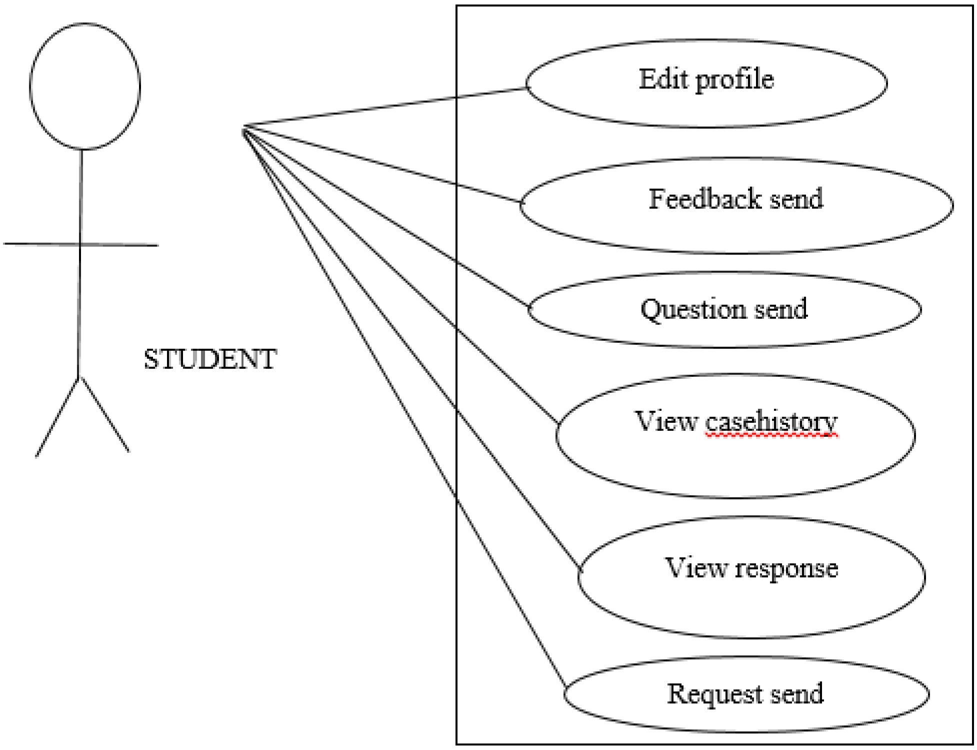


Figure III.4: UML DIAGRAM FOR NURSE



| Figure III.5: UML DIAGRAM FOR DOCTOR



| Figure III.6: UML DIAGRAM FOR STUDENT

Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures in business process, as message sequence charts and call flows for telecoms or wireless system design, and for protocol stack design and analysis.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence Diagrams are typically associated with use case realizations in the logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

III.4 SYSTEM DESIGN

The most creative and challenging phase of the system life cycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specification that will be applied in implementing the candidate system. In system design, we move from the logical to the physical aspects of the life cycle.

The first step is to determine how the output is to be produced and in what format. Then input data and master files have to be designed as the next step and finally the impact of the candidate system on the user and organization are documented and evaluated by the management. After identifying the problem and the limitation of the existing system, a detailed design of the proposed system is conducted.

Free flow personnel interview and reference to previous records prepared manually were the only methods taken to collect necessary information. At present, all organizations are on the path of computerization process.

Design is the phase that indicates the final system. It is the solution, the translation of requirements into ways of meeting them. In this phase the following elements were designed namely, data flow, data stores, processes, procedures was formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

The database tables, input screens, output screens, output reports are designed. After analyzing the various functions involved in the system the database, labels as dictionaries designed. Care is taken for the field name to be in self-explanatory form. Unnecessary fields are avoiding so as not affecting the storage system.

Care must be taken to design the input screen in the most user-friendly way so as to help even the novice users to make entries approximately in the right place. This is being accomplished by the use of giving online help messages, which are brief and cleanly prompts users for appropriate action.

Design is the only way that we can accurately translate a customer's requirements into a finished software product or system. Without design, risk of building an unstable system exist one that will fail when small changes are made, one that will be difficult to test.

All input screens in the system are user friendly and are designed in such a way that even a layman can operate. The sizes of all screens are standardized.

Reports generated in this software give the finer accepts of the required information, which helps in taking vital decision.

The importance of the software design can be stated with a single word quality. Design is a place where quality is fostered in software development. Design is the only way where requirements are actually translated into a finished software product or system.

Mainly this project consists of 4 Modules:

- **Admin**
- **Nurse**
- **Doctor**
- **Student**

Admin Module

Administrator is the main actor in this system. He has the entire control of the system which includes adding all the authorized students. Brief description about the functionalities performed by the admin is given below. After the admin successfully login to this website the admin can perform the functionalities including:

- Admin Login

  By the Username and password admin can login to the system.

- Add Nurse/Delete Nurse

  Admin is responsible to add the nurse of the college and also responsible for delete the nurse.

- Add Doctor/Delete Doctor

  Admin is responsible to add the doctor of the college and also responsible for delete the doctor.

- Add students/ Delete Students

  Admin is responsible to add the authorized students of the college and also responsible for delete the students.

- View feedbacks from students

  Admin is responsible to view the feedbacks from authorized students of the college.

- Manage stock details

  Admin is responsible to fill the necessary medicines.

Nurse Module

Nurse is important actor in the system.Nurse can upload case history of patients.And can manage the stocking details.Nurse can answer to the question from the students and can forward the questions to the doctor if it is necessary.

- Nurse Login

  By using the user name and password, nurse can login to the system.

- Edit profile

  Nurse can update their details and can change password.

- Add/view case history

  Nurse can add the case history and can view their previous histories also.

- Reply/approve questions

  Nurse can send response to the questions and can forward it to the doctor if it is necessary.

- Stock deduction

  Nurse can decrement the number of medicines based on the use.

- Request approval

  Nurse can assign token number to the patients based on the request.

Doctor Module

Doctor is important actor in the system. Doctor can upload case history of patients. And can search the previous case histories. Doctor can answer to the question from the students and can reject the questions to the doctor if it is unnecessary.

- Doctor Login

  By using the user name and password, doctor can login to the system.

- Edit profile

  Doctor can update their details and can change password.

- Add/view case history

  Doctor can add the case history and can view their previous histories also.

- Reject/reply questions

  Doctor can send response to the questions and can reject if it is unnecessary.

Student Module

Student is an important module in the system. Student can see their case histories.And they can take printout the case history.Students can send feedbacks to the administrator.They can ask question to the doctor about their treatment.

- Student login

  By using the user name and password, doctor can login to the system.
- Edit profile

  Students can update their details and can change password.
- Feedback send

  Students can send feedback to the administrator.
- Question send

  Students are able to send questions to the doctors about their treatment.
- View/print case history

  Students can view their case histories and can take printout of the history.
- View response

  Students can see the response from the doctor.
- Request for appointment

  Students can send request for appointment to the nurse.

III.4.1 Input Design

Input design is the process of converting user-oriented input into a computer based format.The goal of the designing input is to make data entry as easy and free from error. In .NET, input to the system is entered through forms. A form is ”any surface on which information is to be entered, the nature of which is determined by what is already on that surface.” If the data going into the system is incorrect, then processing and output will magnify these errors. So designer should ensure that form is accessible and understandable by the user.End-users are people who communicate to the system frequently through the user interface , the design of the input screen should be according to their recommendations.

The data is validated wherever it requires in the project. This ensures only correct data is entered to the system. GUI is the interface used in input design. All the input data are validated in the order and if any data violates any condition the use is warned by a message and asks to re-enter data. If the data satisfies all the conditions then it is transferred to the appropriate tables in the database. This project uses text boxes and drop down to accept user input. If user enters wrong format then it shows a message to the user. User is never lift in confusion as to what is happening. Instead appropriate error messages and acknowledgments are displayed to the user.

III.4.2 Output Design

A quality output is one, which meets the requirement of the end user and presents the information clearly. In any system results of processing are communicated to the user and to the other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship and helps user decision making.

It generally refers to the results and the information that are generated by the system. Effective, descriptive and useful design will improve the relationship with the user and the system because it is the direct source of information to the user. The objective of the output design is to convey the information of all the past activities, required status and to emphasize important events. Outputs from the computers are providing primarily to communicate the results of processing to the user. They also used to provide a permanent copy of these results for later consultation. The major outputs are system flow diagrams and data flow diagram.

III.4.3 DATABASE DESIGN

TABELES

Table III.1: Login table

Field	Type	Size	Constraints	Description
LoginID	Integer	11	Primary Key	Login Id
Username	Varchar	15	NOT NULL	Username
Password	Varchar	100	NOT NULL	Password
Role	Varchar	20	NOT NULL	Role

Table III.2: Register table

Field	Type	Size	Constraints	Description
Id	Integer	20	Primary Key	Id
LoginID	Integer	11	Foreign Key	Login Id
Collegeid	Varchar	50	Unique	College id
Name	Varchar	50	NOT NULL	Name of the student or faculty
Email	Varchar	100	NOT NULL	Email id
Mobile	Integer	20	NOT NULL	Mobile number
Dob	Date	20	NOT NULL	Date of birth
Weight	Varchar	50	NOT NULL	Weight of the student
Height	Varchar	50	NOT NULL	Height of the student
Bloodgroup	Varchar	50	NOT NULL	Bllodgrop of the student

Table III.3: Nurse table

Field	**Type**	**Size**	**Constraints**	**Description**
Id	Integer	20	Primary Key	Id
Nurseid	Integer	15	Unique	Id of the nurse
LoginID	Integer	11	Foreign Key	Login Id
Name	Varchar	100	NOT NULL	Name of the nurse
Dob	Date	50	NOT NULL	Date of birth
Address	Varchar	100	NOT NULL	Address
Email	Varchar	100	NOT NULL	Email id
Mobile	Integer	50	NOT NULL	Mobile number
Qualification	Varchar	20	NOT NULL	Education
Experience	Varchar	20	NOT NULL	Number of years of experience
Image	ImageField		NOT NULL	Image of the nurse

Table III.4: Doctor table

Field	**Type**	**Size**	**Constraints**	**Description**
Id	Integer	20	Primary Key	
Doctorid	Integer	15	Unique	Id of the doctor
LoginID	Integer	11	Foreign Key	Login Id
Name	Varchar	100	NOT NULL	Name of the doctor
Dob	Date	50	NOT NULL	Date of birth
Address	Varchar	100	NOT NULL	Address
Email	Varchar	100	NOT NULL	Email id
Mobile	Integer	50	NOT NULL	Mobile number
Qualification	Varchar	20	NOT NULL	Education
Experience	Varchar	20	NOT NULL	Number of years of experience
Image	ImageField		NOT NULL	Image of the doctor

Table III.5: Medicine table

Field	**Type**	**Size**	**Constraints**	**Description**
Medicineid	Integer	20	Primary Key	Id of the medicine
Medicinename	Varchar	100	NOT NULL	Name of the medicine
Suppliername	Varchar	50	NOT NULL	Name of the supplier
Place	Varchar	50	NOT NULL	Place
Date	Date	20	NOT NULL	Date
Billno	Varchar	50	NOT NULL	Bill number
Quantity	Integer	20	NOT NULL	Number of medicine
Amount	Integer	20	NOT NULL	Price of the medicine

Table III.6: Equipment table

Field	**Type**	**Size**	**Constraints**	**Description**
Equipmentidid	Integer	20	Primary Key	Id of the equipment
Equipmentname	Varchar	100	NOT NULL	Name of the equipment
Suppliername	Varchar	50	NOT NULL	Name of the supplier
Place	Varchar	50	NOT NULL	Place
Date	Date	20	NOT NULL	Date
Billno	Varchar	50	NOT NULL	Bill number
Quantity	Integer	20	NOT NULL	Number of equipment
Amount	Integer	20	NOT NULL	Price of the equipment

Table III.7: Stock table

Field	Type	Size	Constraints	Description
Medicineid	Integer	20	Foreign Key	Id of the medicine
Medicinename	Varchar	100	NOT NULL	Name of the medicine
Amount	Integer	20	NOT NULL	Price of the medicine
Quantity	Integer	20	NOT NULL	Balance number of medicine

Table III.8: student_feedback table

Field	Type	Size	Constraints	Description
student_name	Varchar	100	NOT NULL	Name of the student or faculty
feedback	Varchar	200	NOT NULL	Comments

Table III.9: question table

Field	Type	Size	Constraints	Description
Ids	Integer	20	Foreign Key	Loginid
Collegeid	Varchar	50	Foreign Key	College id
Name	Varchar	100	Foreign Key	Name of the student or faculty
Questions	Varchar	200	NOT NULL	Questions asked
Did	Varchar	100	Foreign Key	Doctor Id

Table III.10: Approved_questions table

Field	**Type**	**Size**	**Constraints**	**Description**
Collegeid	Varchar	50	NOT NULL	College id
Name	Varchar	100	NOT NULL	Name of the student or faculty
Question	Varchar	200	NOT NULL	Questions asked
Doctor_id	Integer	100	NOT NULL	Docto Id
Doctor_name	Varchar	100	NOT NULL	Doctor name

Table III.11: response_from_doctor table

Field	**Type**	**Size**	**Constraints**	**Description**
Question_id	Integer	200	NOT NULL	Questions asked
Response	Varchar	100	NOT NULL	Response to the question

Table III.12: case_history table

Field	**Type**	**Size**	**Constraints**	**Description**
Username	Varchar	50	Foreign Key	College id
Ids	Varchar	100	Foreign Key	Login Id
Name	Varchar	200	Foreign Key	Name of the student
Date	Date	10	NOT NULL	Date of treatment
Diagnosis	Varchar	100	NOT NULL	Diagnosis provided
Clinicalhistory	Varchar	100	NOT NULL	Patient’s current condition
Treatment	Varchar	100	NOT NULL	Treatment details

Table III.13: request_appointms table

Field	**Type**	**Size**	**Constraints**	**Description**
Ids	Integer	10	Foreign Key	Login Id
Reason	Varchar	200	NOT NULL	Reason for the appointment
Appointdate	Date	10	NOT NULL	Appointment date

4.4 DATA FLOW DIAGRAM

Data flow diagrams represent one of the most ingenious tools used for structured analysis.It has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It is the major starting point in the design phase that functionally decomposes the requirements specification down to the lowest level ofdetail. In the normal convention a DFD has four major symbols.

Symbols used in DFD are:

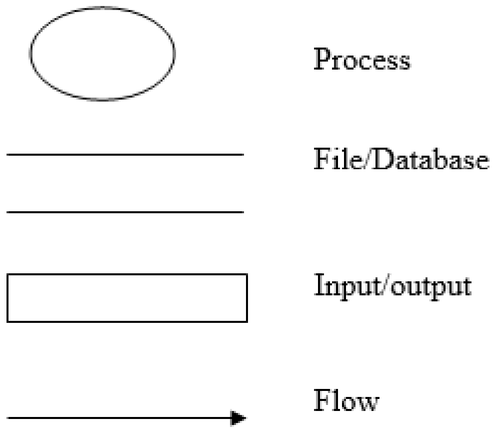
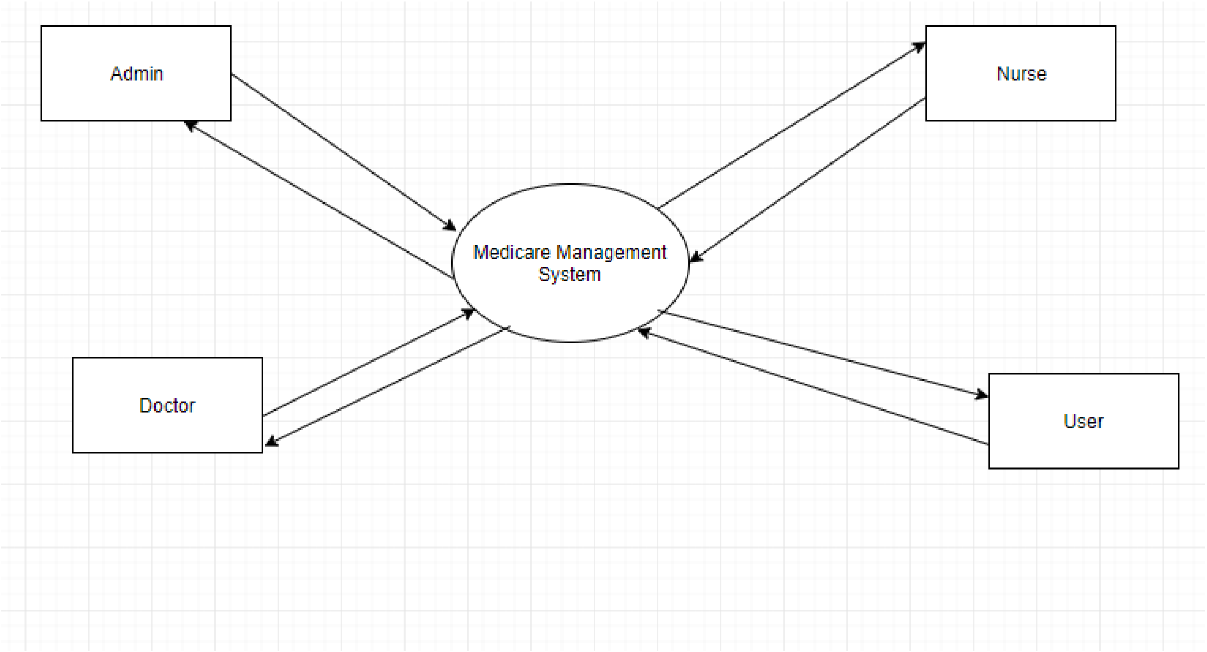
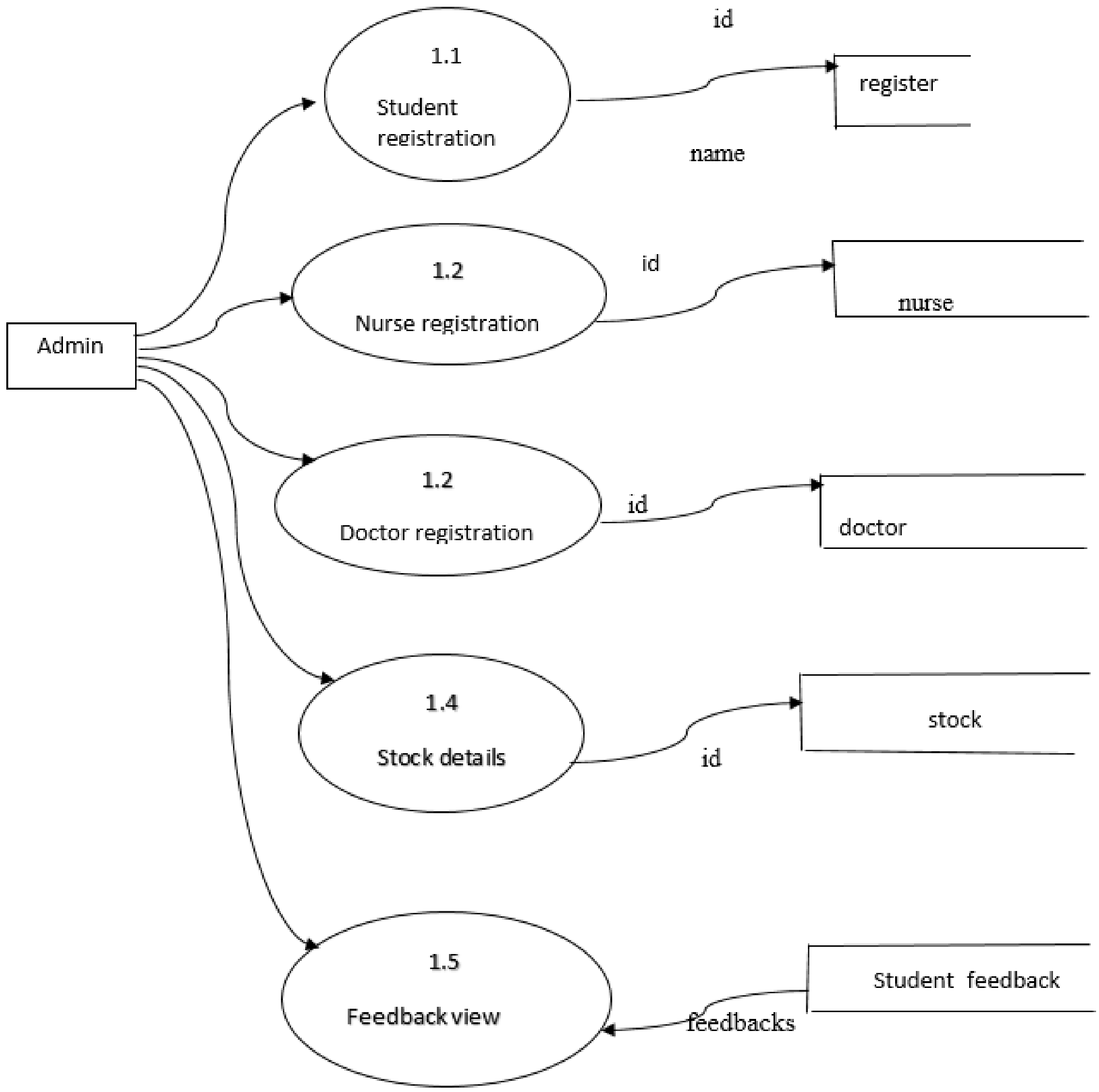


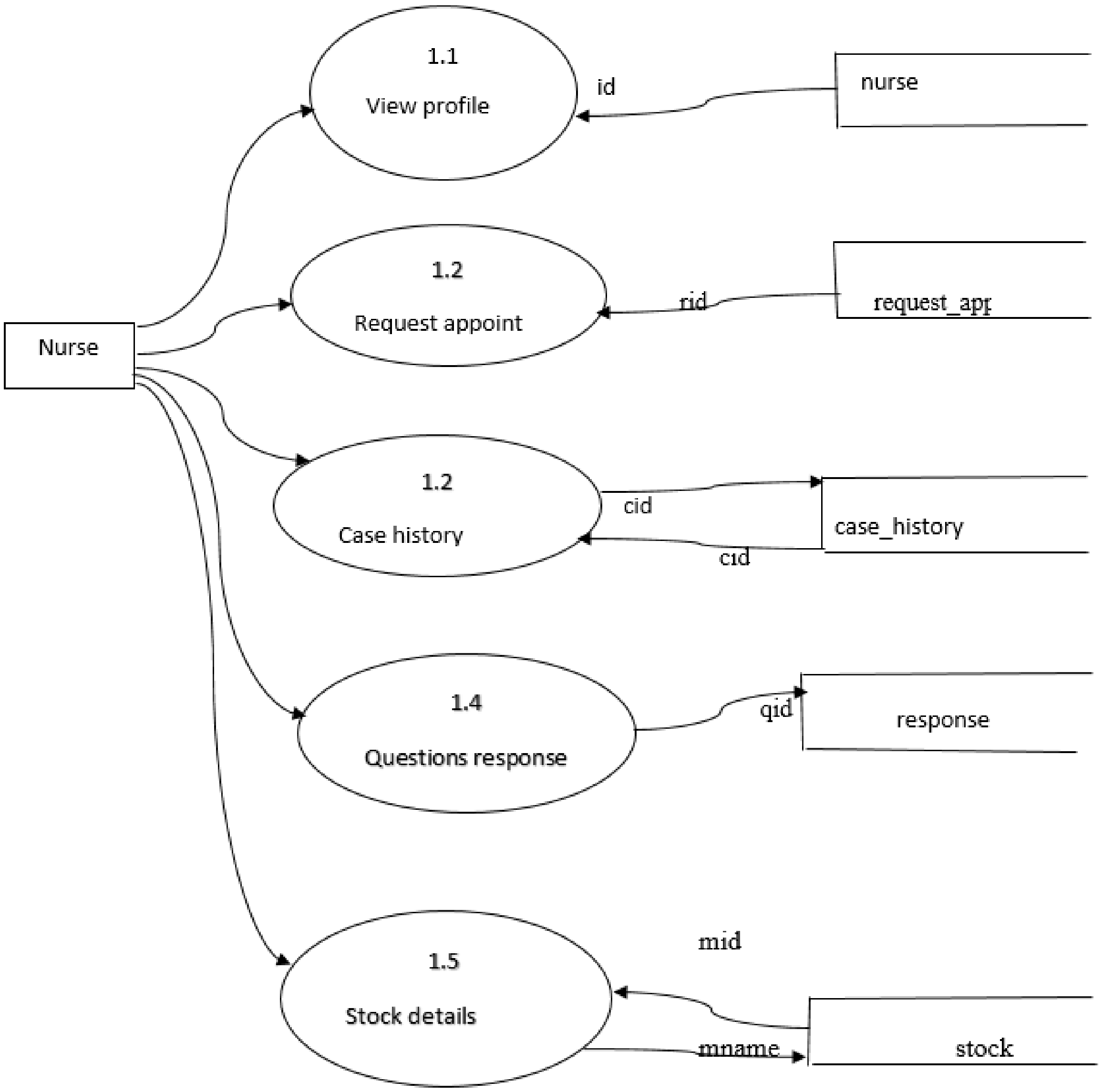
Figure 4.1: DFD Symbols

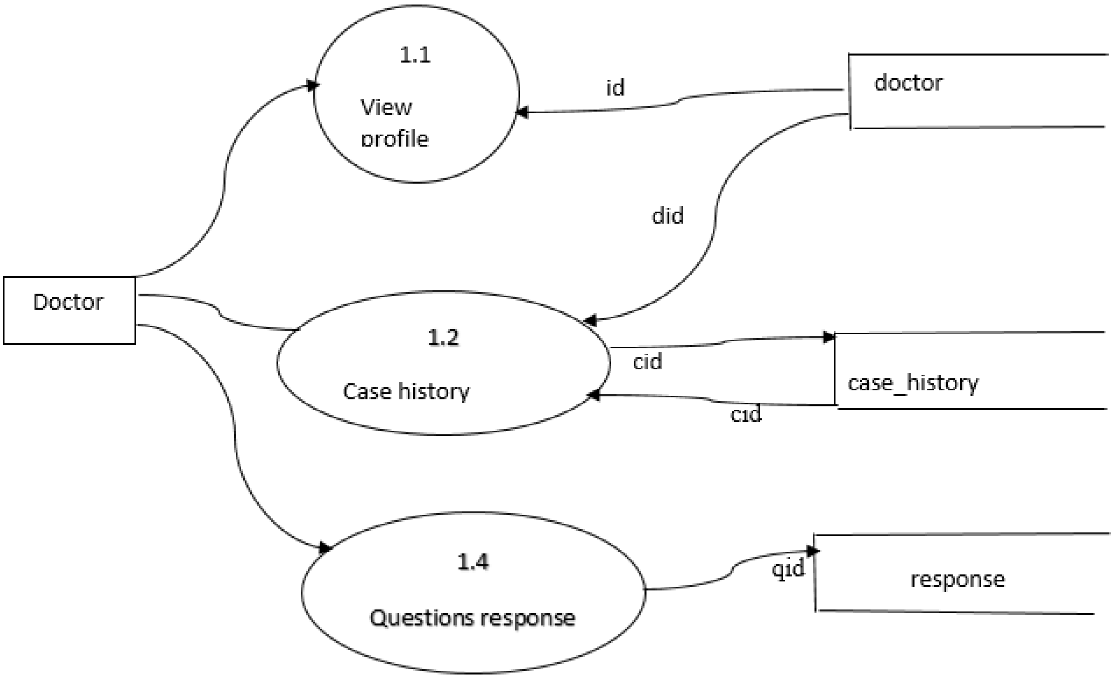
CONTEXT DIAGRAM

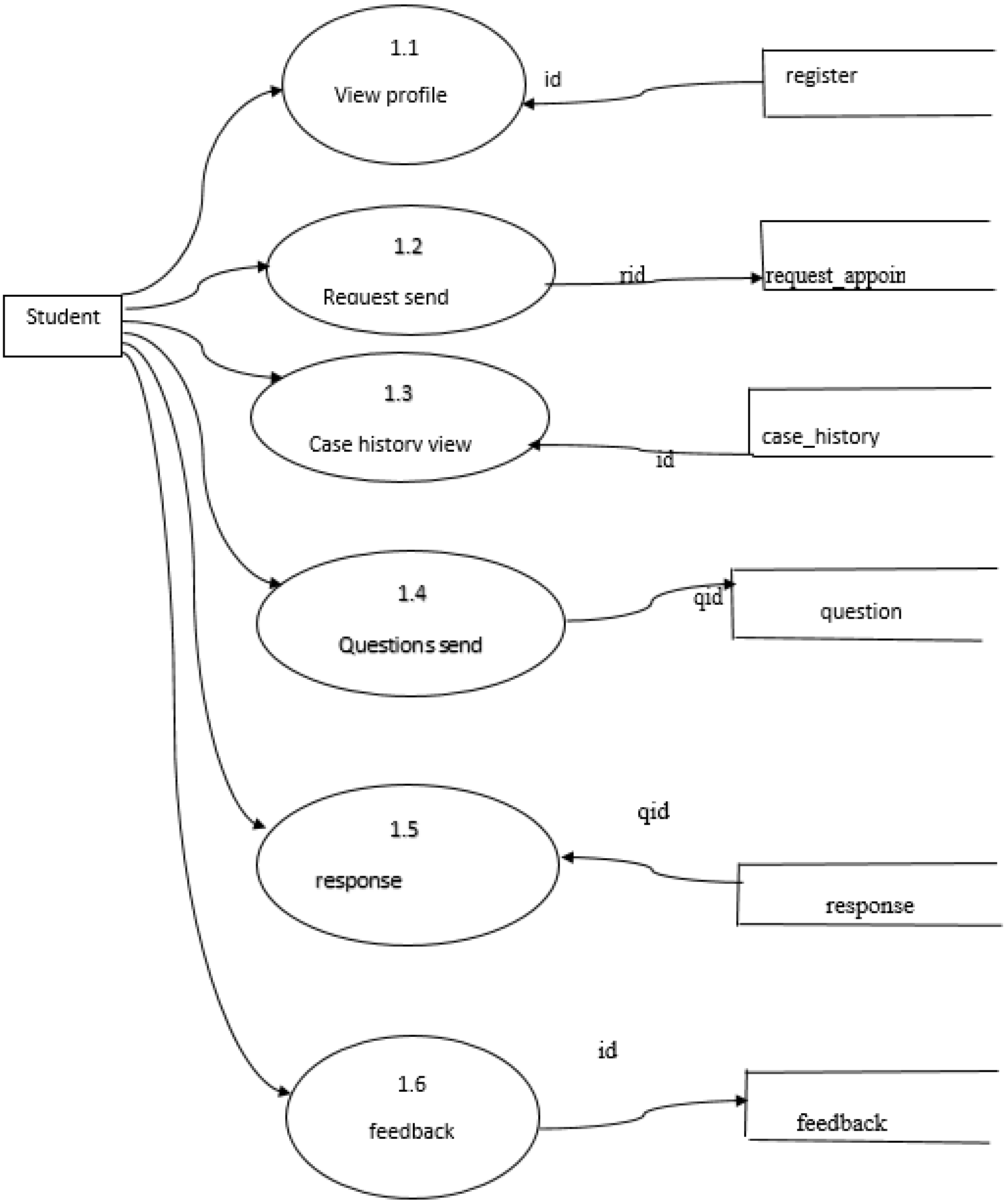


Level 1 DFD


id
1.1
Student
registration
register
name
Admin
1.2
Nurse registration
id
nurse
1.2
Doctor registration
id
doctor
1.4
Stock details
stock
id
1.5
Feedback view
Student feedback
feedbacks


Nurse
1.1
View profile
id
nurse
1.2
Request appoint
rid
request_apr
1.2
Case history
cid
cid
case_history
1.4
Questions response
qid
response
1.5
Stock details
mid
mname
stock


1.1
View profile
id
doctor
did
Doctor
1.2
Case history
cid
case_history
cid
1.4
Questions response
qid
response


Student
1.1
View profile
id
register
1.2
Reauest send
rid
request_appoin
1.3
Case historv view
id
case_history
1.4
Questions send
qid
question
1.5
response
qid
response
1.6
feedback
id
feedback

4.5 ACTIVITY DIAGRAM

Admin

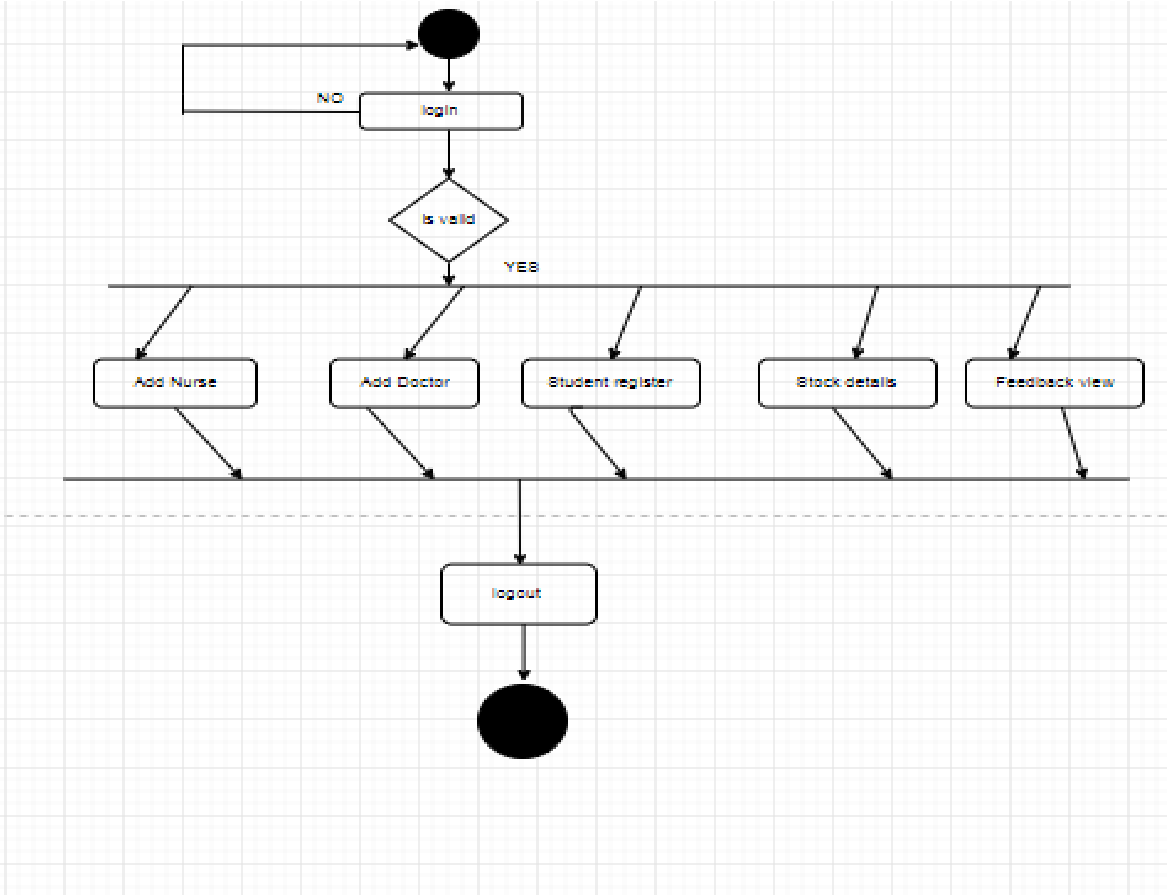


Figure 4.6: admin.jpg

Nurse

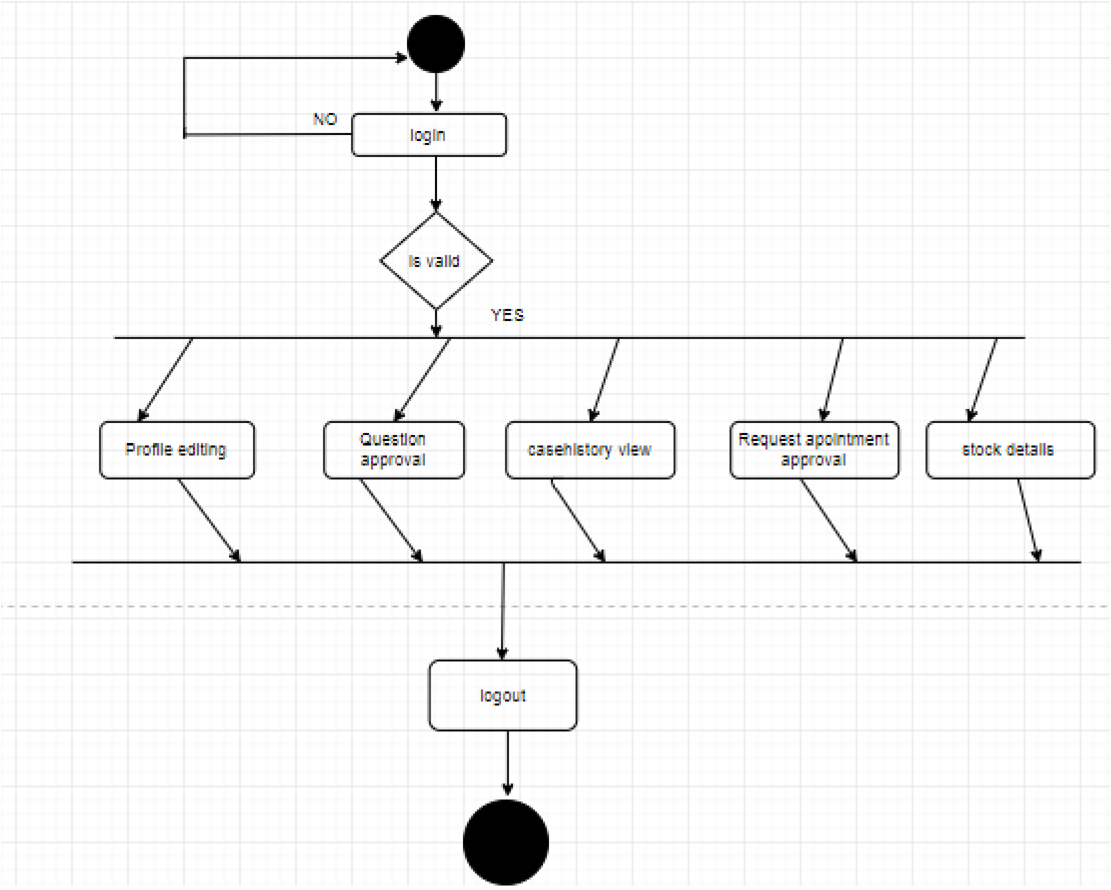


Figure 4.7: nurse.jpg

Doctor

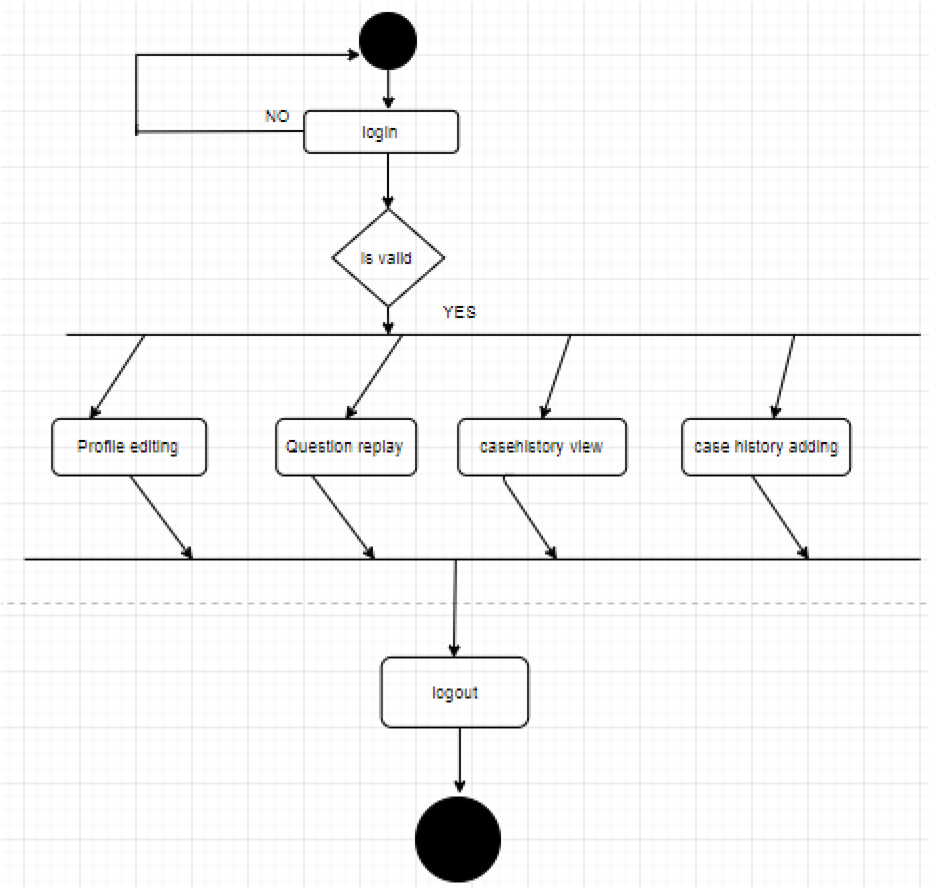


Figure 4.8: doctor.jpg

Student

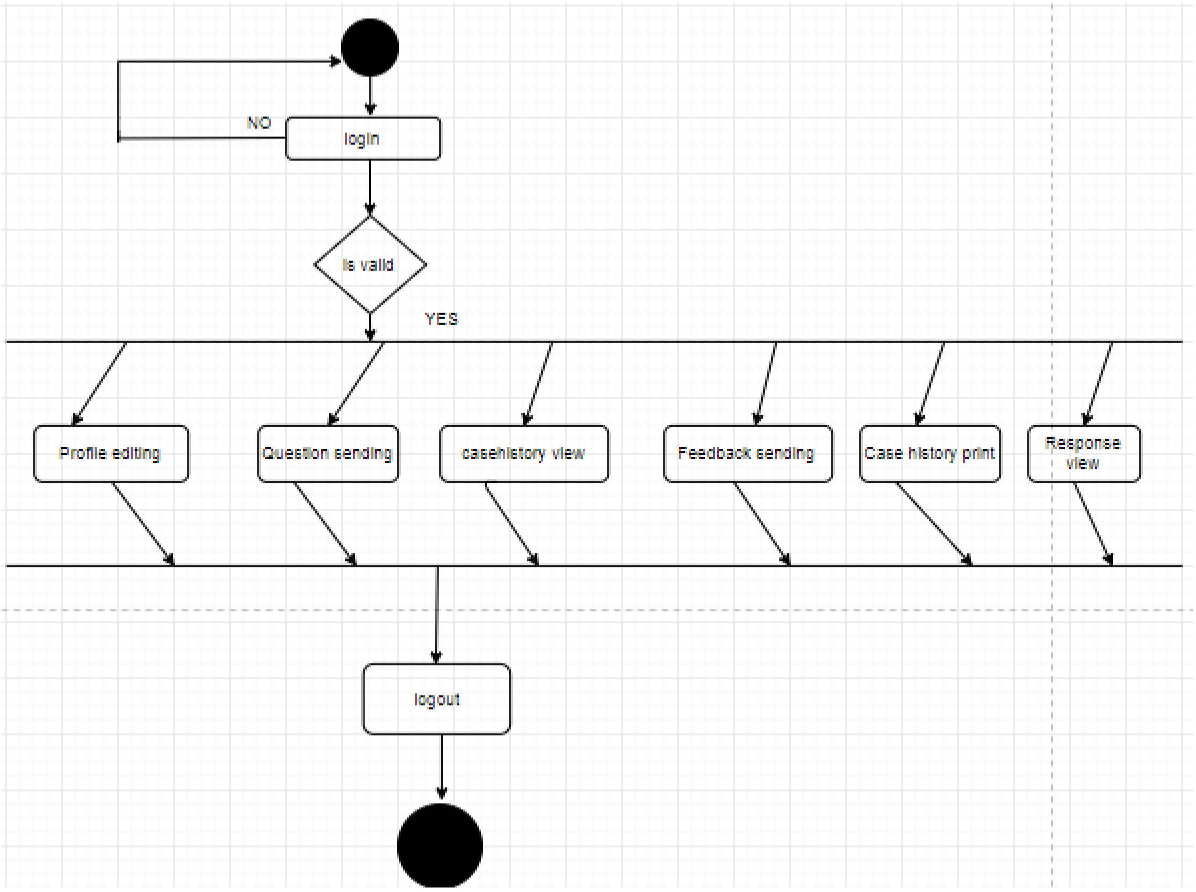


Figure 4.9: user.jpg

4.6 SEQUENCE DIAGRAM

Admin

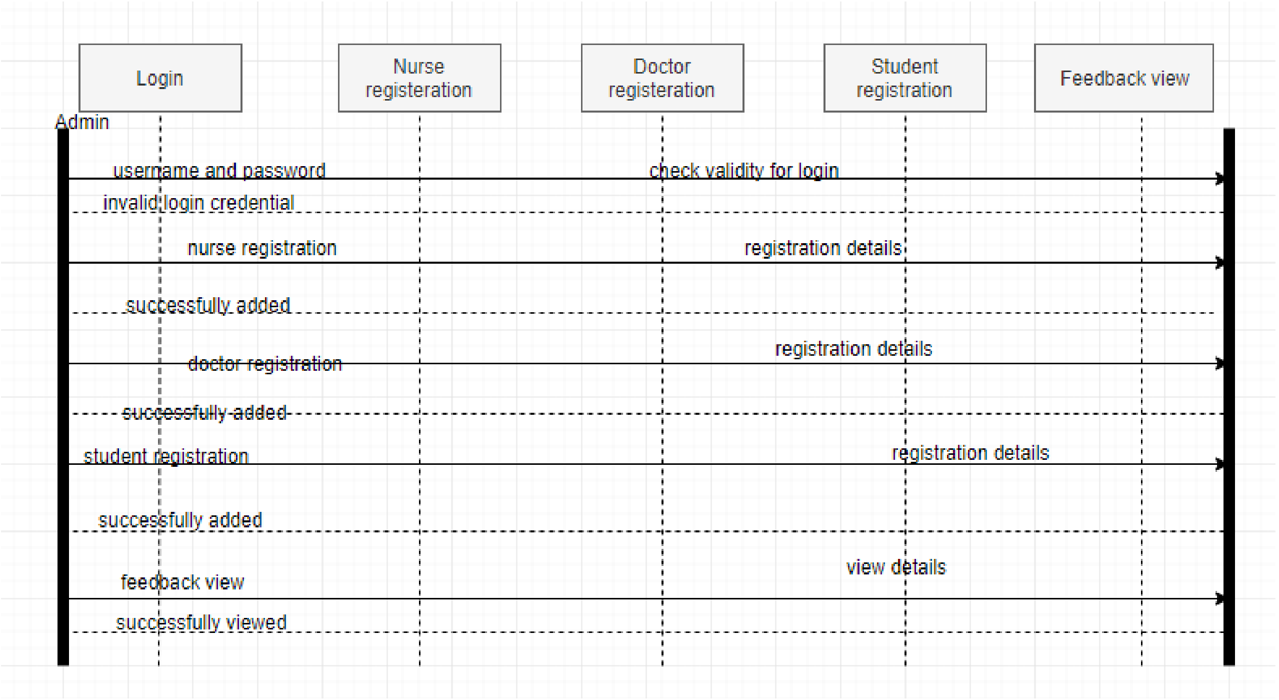


Figure 4.10: adminsequence.png

Nurse

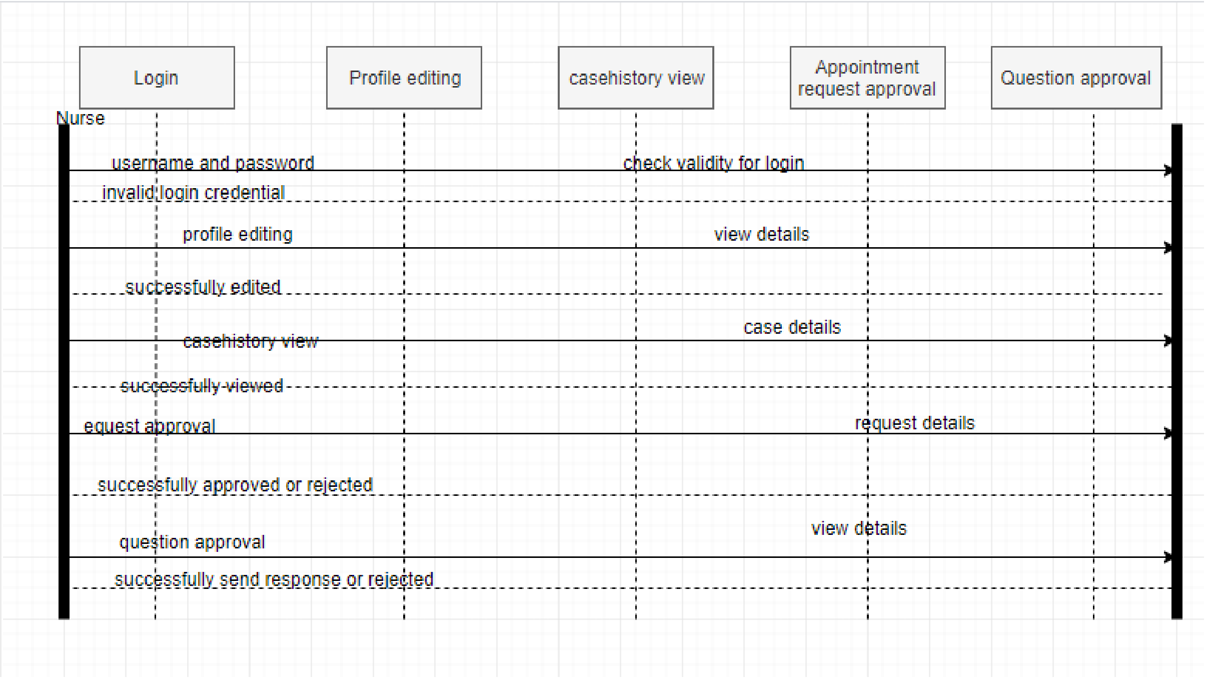


Figure 4.11: nursesequence.png

Doctor

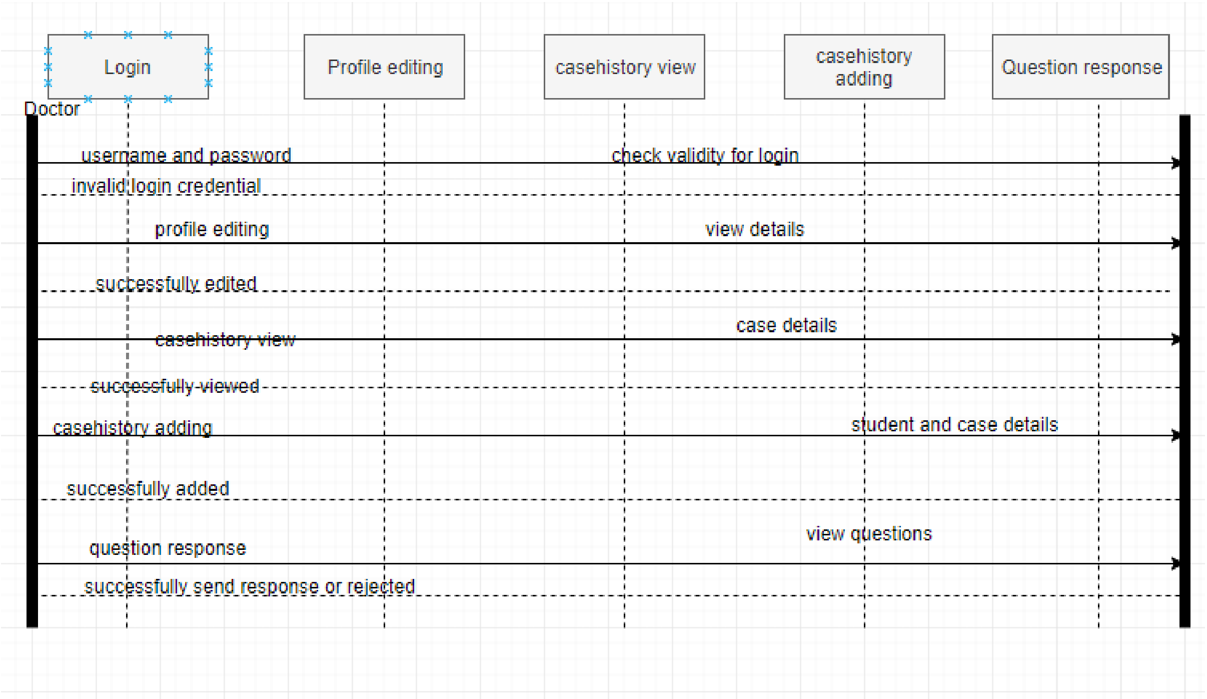


Figure 4.12: doctorsequence.png

Student

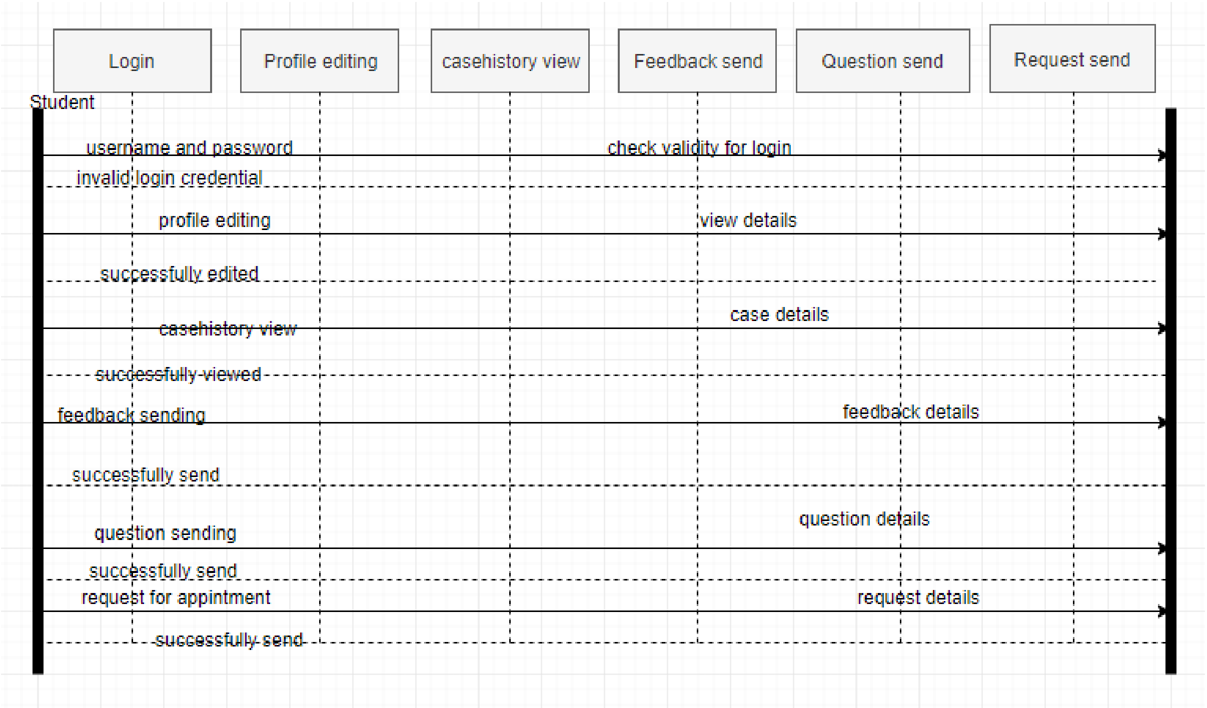


Figure 4.13: usersequence.png

III.5 TOOLS AND PLATFORMS

III.5.1 Introduction to Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- **Python 2 and Python 3**. Both are quite different.

Before we start Python programming, we need to have an interpreter to interpret and run our programs. There are certain online interpreters like http://ideone.com/ or http://codepad.org/ that can be used to start Python without installing an interpreter. Django is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern. It is maintained by the Django Software Foundation, an independent organization established as a 501 non-profit. Django's primary goal is to ease the creation of complex, database-driven websites.

III.5.2 The Python History

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum continued as Python's lead developer until July 12, 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January, 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and van Rossum to a five-member "Steering Council" to lead the project. Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python

2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go transcompiler to improve performance under concurrent workloads.

III.5.3 The Python Platform

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.Some well-known sites that use Django include the Public Broadcasting Service, Instagram,,Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor. It was used on Pinterest, but later the site moved to a framework built over Flask.

III.5.4 Django web framework

Django was created in the fall of 2003, when the web programmers at the Lawrence Journal-World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications. It was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt. In June 2008, it was announced that a newly formed Django Software Foundation (DSF) would maintain Django in the future. Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views", the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper (ORM) that mediates between data models(defined as Python classes) and a relational database ("**M**odel"), a system for processing HTTP requests with a web templating system("**V**iew"), and a regular-expression-based URL dispatcher ("**C**ontroller").

Also included in the core framework are:

- a lightweight and standalone web server for development and testing
- a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database
- a template system that utilizes the concept of inheritance borrowed from object-oriented programming
- a caching framework that can use any of several cache methods
- support for middleware classes that can intervene at various stages of request processing and carry out custom functions
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals
- an internationalization system, including translations of Django's own components into a variety of languages
- a serialization system that can produce and read XML and/or JSON representations of Django model instances
- a system for extending the capabilities of the template engine
- an interface to Python's built-in unit test framework

III.5.5 SQLite

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions, and will store the data as-is if such a conversion is not possible.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

SQLite's code is hosted with Fossil, a distributed version control system that is itself built upon an SQLite database. A standalone command-line program is provided in SQLite's distribution. It can be used to create a database, define tables, insert and change rows, run queries and manage an SQLite database file. It also serves as an example for writing applications that use the SQLite library.

SQLite uses automated regression testing prior to each release. Over 2 million tests are run as part of a release's verification. Starting with the August 10, 2009 release of SQLite 3.6.17, SQLite releases have 100% branch test coverage, one of the components of code coverage. The tests and test harnesses are partially public domain and partially proprietary.

III.5.6 About Github

GitHub (originally known as Logical Awesome LLC) is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers plans for both private repositories and free accounts which are commonly used to host open-source software projects. As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world.

Development of the GitHub platformbegan on October 19, 2007. The site was launched in April 2008 by TomPreston-Werner, ChrisWanstrath, and PJ Hyett after it had beenmade available for a few months prior as a beta release.Projects on GitHub can be accessed and manipulated using the standard Git command-line interface and all of the standard Git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. Multiple desktop clients and Git plugins have also been created by GitHub and other third parties that integrate with the platform. The site provides social networking-like functions such as feeds, followers, wikis (using wiki software called Gollum) and a social network graph to display how developers work on their versions ("forks") of a repository and what fork (and branch within that fork) is newest.A user must create an account in order to contribute content to the site, but public repositories can be browsed and downloaded by anyone. With a registered user account, users are able to discuss, manage repositories, submit contributions to others' repositories, and review changes to code.The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. developers ChrisWanstrath, PJ Hyett, and Tom Preston-Werner.

III.5.7 Normalization

Normalization refers how to implement the relationships and storage of data in the database tables. Keys are used to uniquely define a relationship to another instance or set of information. In first normal form each value in the database table is atomic or represented only once. In second form each instance or raw in the database table must be uniquely identifiable. The table in the third normal form won't have redundant non key information.

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example, it partially provides triggers, and it cannot write to views (however it provides INSTEAD OF triggers that provide this functionality). While it provides complex queries, it still has limited ALTER TABLE function, as it cannot modify or delete columns.

SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is *dynamically typed*. Moreover, it is *weakly typed* in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like `CHECK(typeof(x)='integer')`. Tables normally include a hidden *rowid* index column which gives faster access.[18] If a database includes an Integer Primary Key column SQLite will typically optimize it by treating it as an alias for *rowid*, causing the contents to be stored as a strictly typed 64-bit signed integer and changing its behavior to be somewhat like an auto-incrementing column. Future versions of SQLite may include a command to introspect whether a column has behavior like that of *rowid* to differentiate these columns from weakly-typed, non-autoincrementing Integer Primary Keys. SQLite with full Unicode function is optional.

Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes.

SQLite version 3.7.4 first saw the addition of the FTS4 (full text search) module, which features enhancements over the older FTS3 module. FTS4 allows users to perform full text searches on documents similar to how search engines search webpages. Version 3.8.2 added support for creating tables without rowid,[24] which may provide space and performance improvements.[25] Common table expressions support was added to SQLite in version 3.8.

CHAPTER IV
SYSTEM TESTING

IV.1 TESTING METHODOLOGIES AND STRATEGIES

Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing .There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time consuming process. Furthermore, it requires skilled and experienced testers to develop great quality test cases.

Even with the wide availability of automation tools for testing, the degree of automation mostly remains at the automated test script level and generally significant amount of human intervention is required in testing. In addition data collected, as testing is conducted provides a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process. Testing begins at the module level and work towards the integration of entire computer based system. No testing is completed without verification and validation part.

The goal of verification and validation activities are to access and improve the quality of work products generated during the development and modification of the software. Testing plays a vital role in determining the reliability and efficiency of the software and hence is very important stage in software development. Tests are to be conducted on the software to evaluate its performance under a number of conditions. Ideally, it should do so at the level of each module and also when all of them are integrated to form the completed system.

In the project ”MEDICARE” the testing has been successfully handled with the modules. The test data was given to each and every module in all respect and got the desired output. Each module that has been tested is found working properly.

IV.1.1 Unit Testing

Here we test each module individually and integrated the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as ”module testing”. The modules of the ”MEDICARE” are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of ” MEDICARE” independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These error are verified and corrected and so that the unit perfectly fits to the project.

IV.1.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub-functions, when combined they may not perform the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance. The Modules of this project are connected and tested.

After splitting the programs into units, the units were tested together to see the defects between each module and function. It is testing to one or more modules or functions together with the intent of finding interface defects between the modules or functions. Testing completed at as part of unit or functional testing, integration testing can involve putting together of groups of modules and functions with the goal of completing and verifying meets the system requirements.

IV.1.3 system Testing

System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System

Testing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/System is tested in an environment that closely resembles the production environment where the application will be finally deployed.

The perquisites for System Testing are:-

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.

- Testing should be completed in an environment closely resembling the production environment. When necessary iterations of System Testing are done in multiple environments.

IV.1.4 User Acceptance Testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system

TEST CASE

S.NO	PAGE	TEST STEP	EXPECTED RESULT	ACTUAL RESULT	STATUS
		Enter valid username and password	Should validate the user and provide access to user accounts	Got the entry to user accounts	pass

1.	Login page	Enter invalid username or password	Should flash an error message	Error message shown	Pass
		Enter invalid category	Should flash an error message	Error message shown	Pass
2.	Add students	Enter all the fields with valid entries	should flash a successful message	successful message shown	Pass
		Enter invalid entries	Should flash an error message	Error message shown	Pass
3.	Add stocks	Enter all the fields with valid entries	should flash a successful message	successful message shown	Pass
		Enter invalid entries	Should flash an error message	Error message shown	Pass

4.	change password Page	Enter valid username and click submit button	Provide access to set new password	Got access to set new password	Pass
		Enter invalid username and click find account	Should flash an error message	Error message shown	Pass

CHAPTER V

SYSTEM IMPLEMENTATION

The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested software system into actual work environment. Implementation is concerned with translating design specification with source code. The primary goal of implementation is to write the source code to its specification that can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system design into operational one. The three types of implementation are:-implementation of a computerized system to replace a manual system, implementation of a new system to replace existing one and implementation of a modified system to replace an existing one.

The implementation is the final stage and it is an important phase. It involves the individual programming ; system testing, user training , and the operational running of developed proposed system that constitute the application subsystem. The implementation phase of the software development is concerned with translating design specification in the source code. The user tests the developed system and the changes are according to the needs. Before implementation, Several tests have been conducted to ensure no errors encountered during the operation. The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs ,installs and operated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

Chapter VI

SCOPE OF FUTURE ENHANCEMENTS

Making enhancements is all about perfective maintenance. It means adding, modifying or redeveloping the code to support changes in the specifications. It is necessary to keep up with changing user needs and the operational environment. More money and time is spend on perfective maintenance than on corrective or adaptive maintenance together. In the proposed system Medicare, future enhancements are possible in the following areas:

- We can adjust the stock details using a threshold value.
- In future there will be a facility to send questions via email.
- Upgradations are possible.

CHAPTERVII

CONCLUSION

The project titled ”MEDICARE,this system has developed for automating all the activities in the college medicare , where user can easily search their case histories as their need.The project is developed as a Web Application by using PYTHON DJANGO as the front end and SQLite as the back end.Each user of the system has role and permission according to their role.The students can take printout of the case histories for further treatment.Students can ask their doubts about their treatment to doctor. If a patient admitted by seriously injured, using his identification number doctor can find his previous case history like any allergy, previous surgery details if any etc.. And the doctor can refer the patients to the hospitals if it is necessary.Thus this system is very beneficial for the students and staff.

REFERENCES

1. Anthony Brun "Python Programming: A Step By Step Guide From Beginner To Expert" (Beginner, Intermediate & Advanced) Kindle Edition

2. Samuel Dauzon , Aidas Bendoraitis, Arun Ravindran "Django: Web Development with Python Paperback" – Import, 28 Dec 2016

3. Nischay kumar Hegde "Python Programming Fundamentals"- A Beginner's Handbook Paperback – 2018

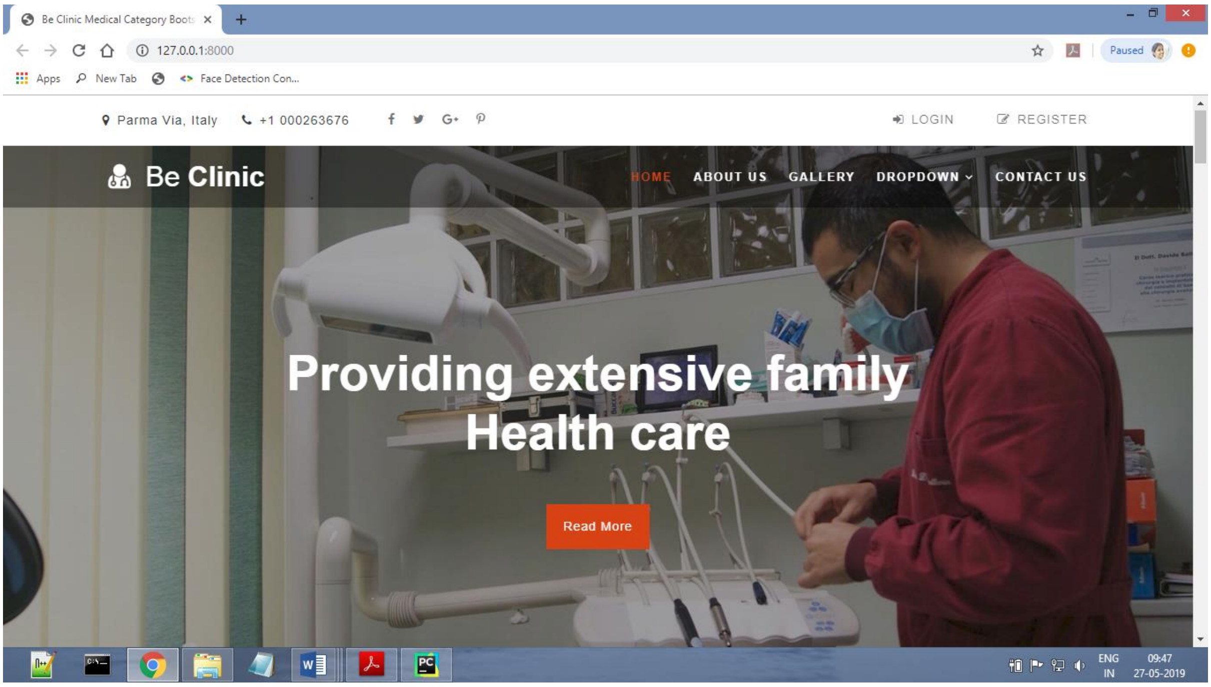
4. https://www.w3schools.com/python/python_intro.asp

5. https://realpython.com/tutorials/django/

APPENDICES

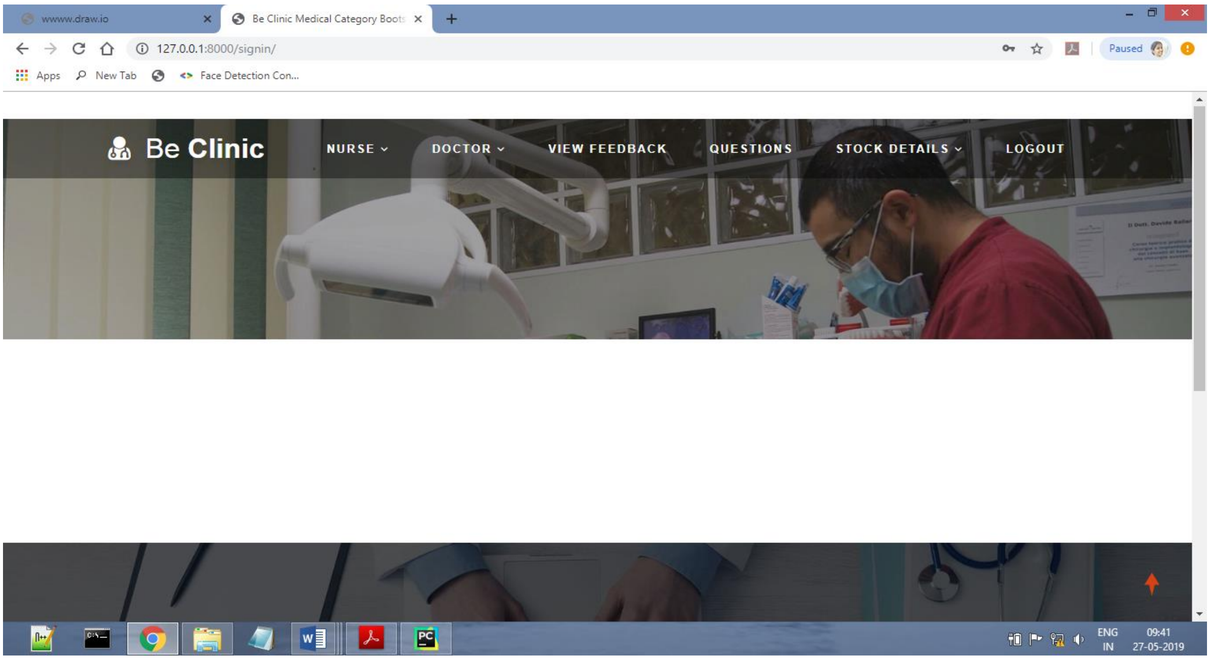
A.1 SCREEN SHOTS INPUT FORM,OUTPUT FORMS

HOMEPAGE

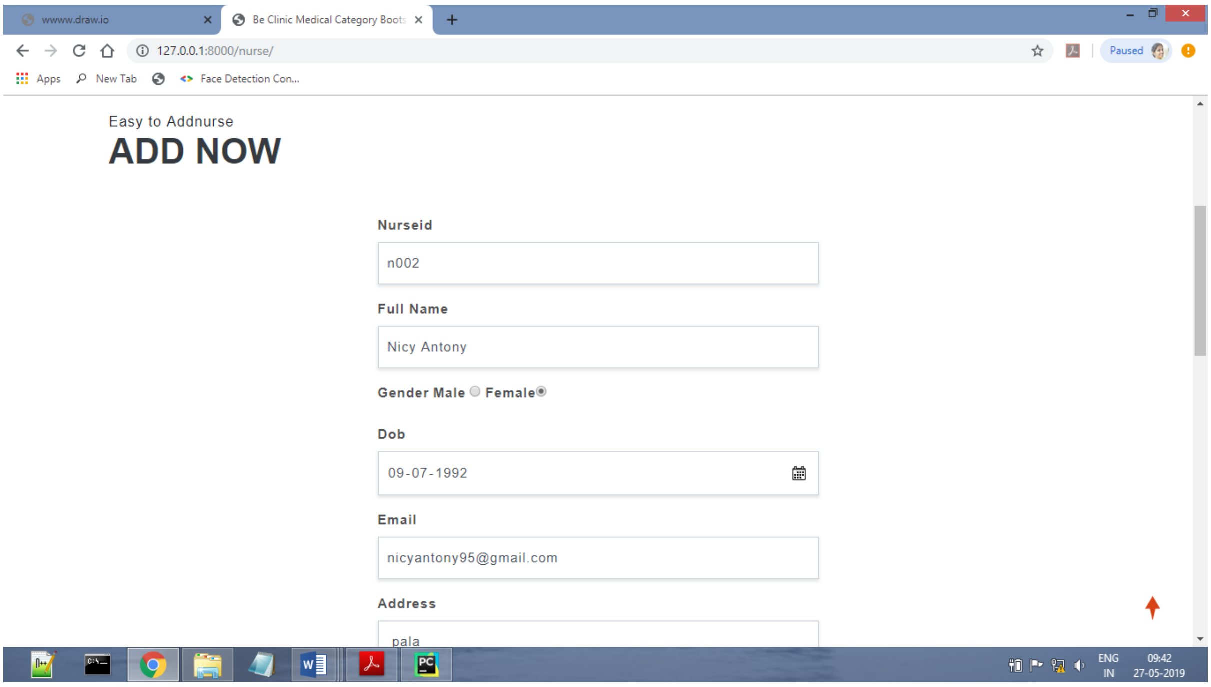


ADMIN

Homepage



Nurse registration



Easy to Addnurse

ADD NOW

Nurseid

n002

Full Name

Nicy Antony

Gender Male ○ Female●

Dob

09-07-1992

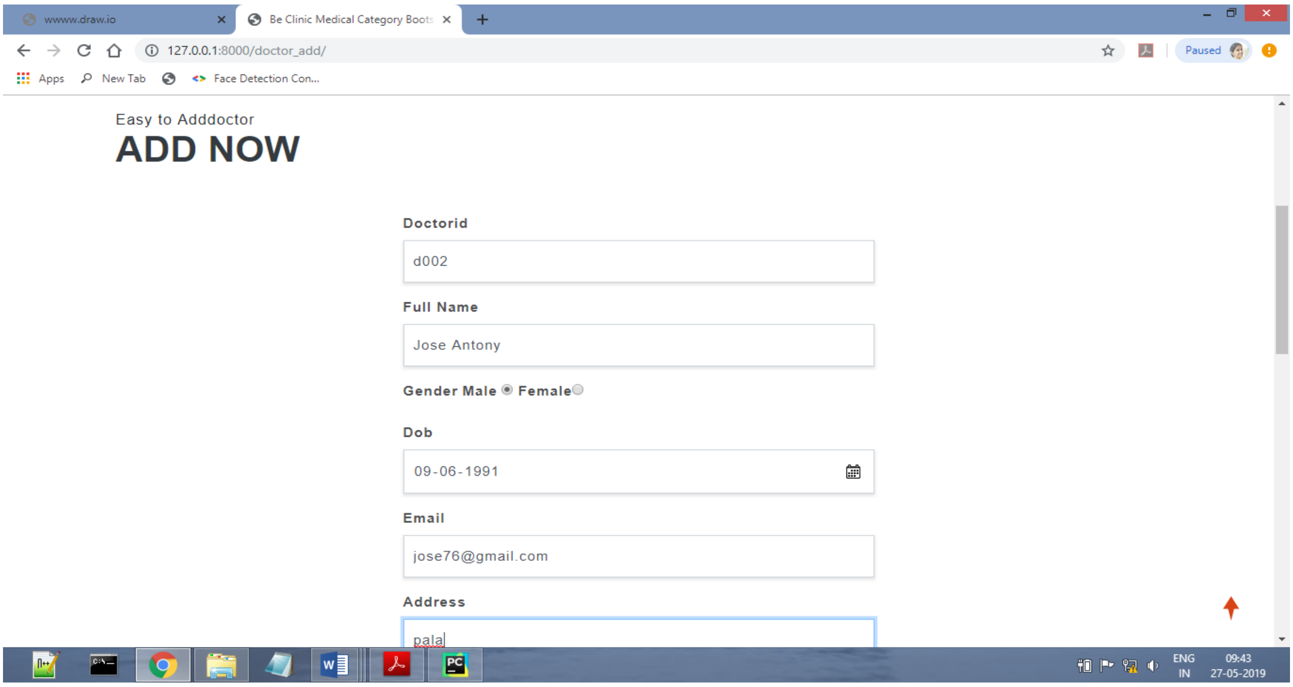
Email

nicyantony95@gmail.com

Address

pala

Doctor registration



Easy to Adddoctor

ADD NOW

Doctorid

d002

Full Name

Jose Antony

Gender Male ● Female○

Dob

09-06-1991

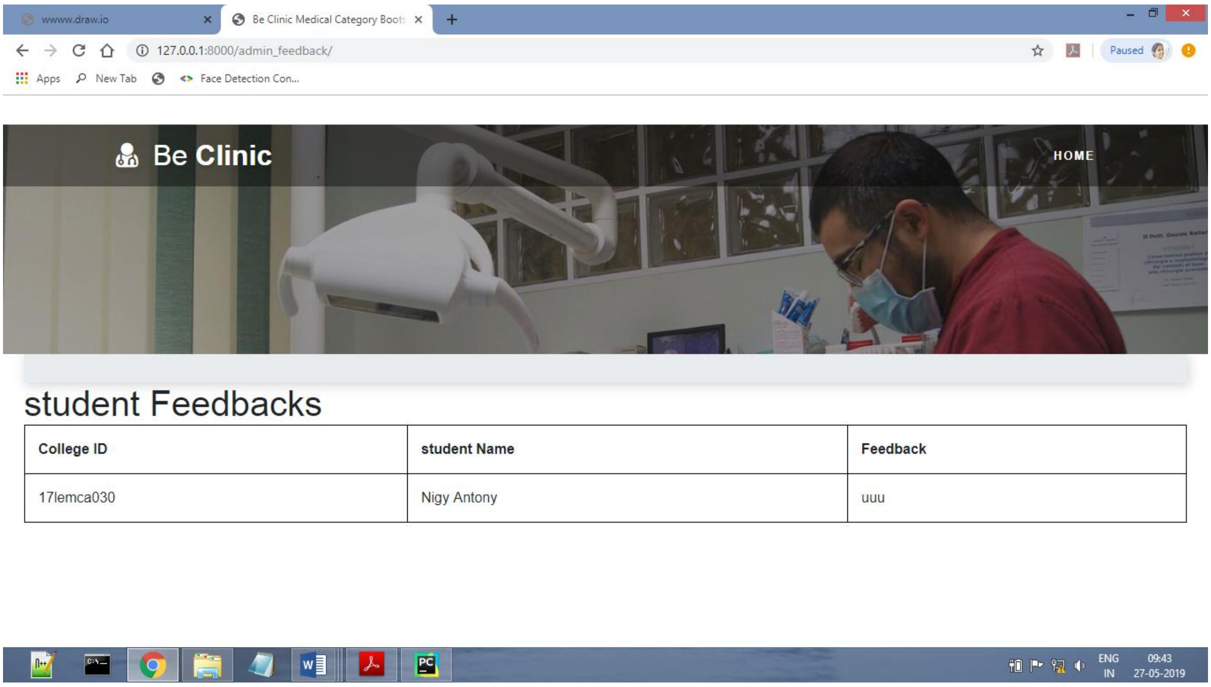
Email

jose76@gmail.com

Address

pala

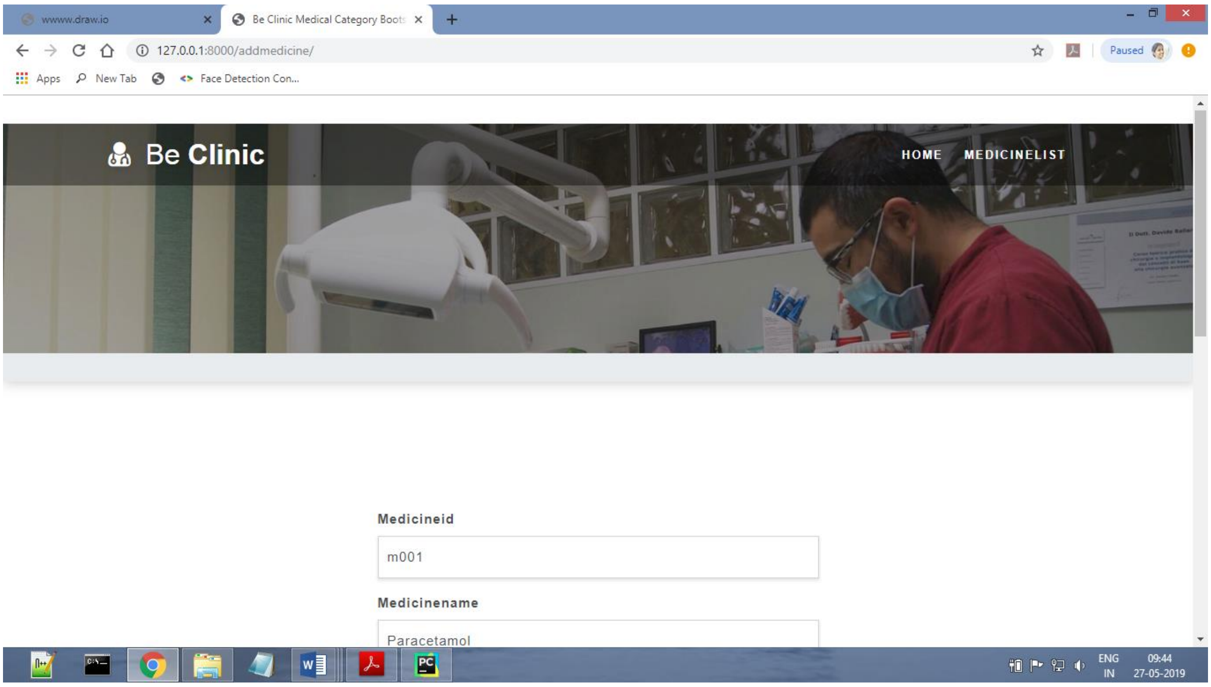
Feedback form



student Feedbacks

College ID	student Name	Feedback
17lemca030	Nigy Antony	uuu

Medicine entry



Medicineid

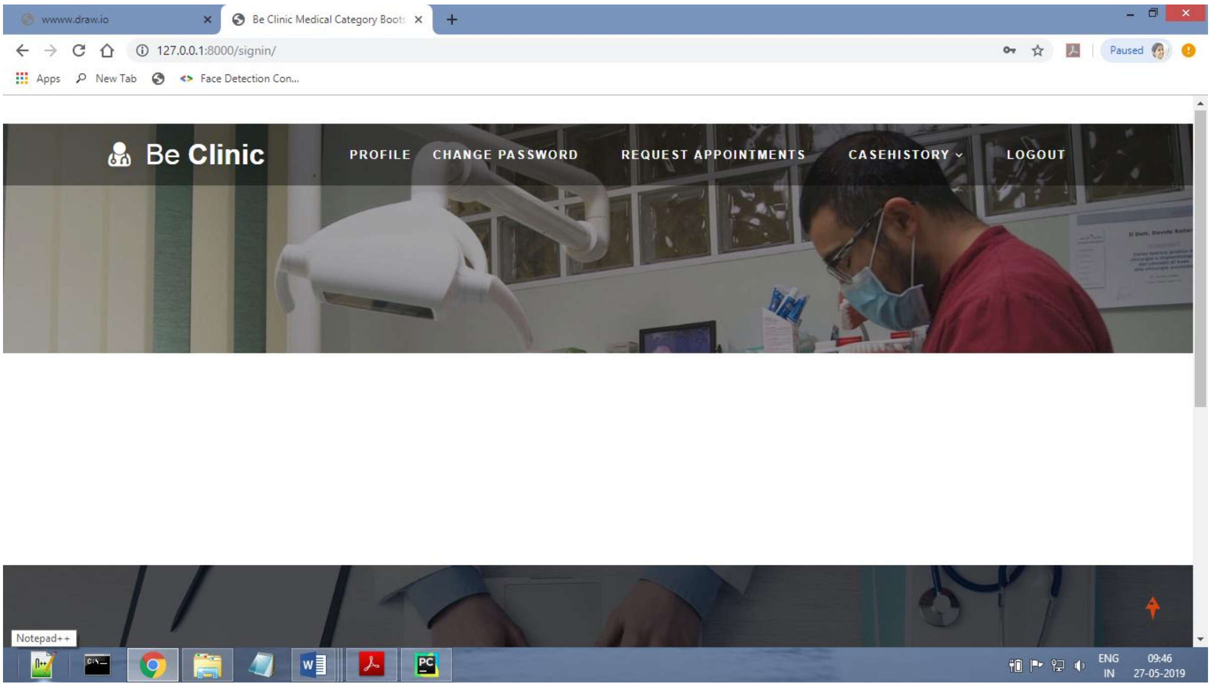
m001

Medicinename

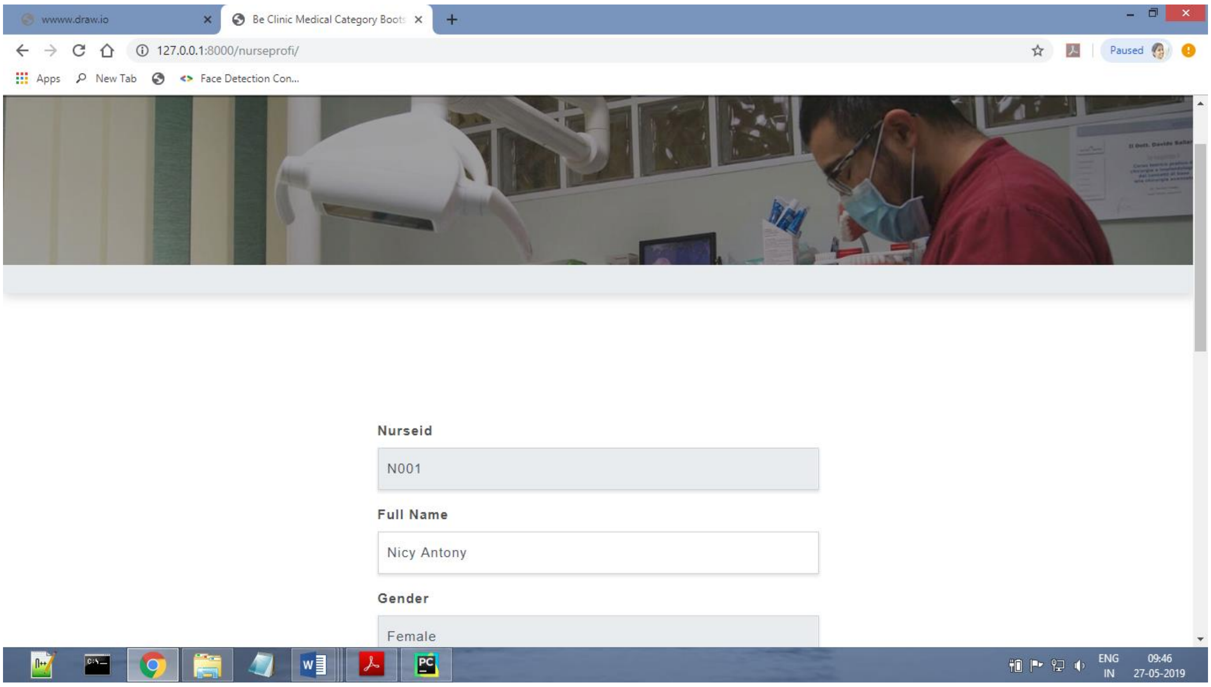
Paracetamol

NURSE

Nurse homepage



Profile editing



Nurseid

N001

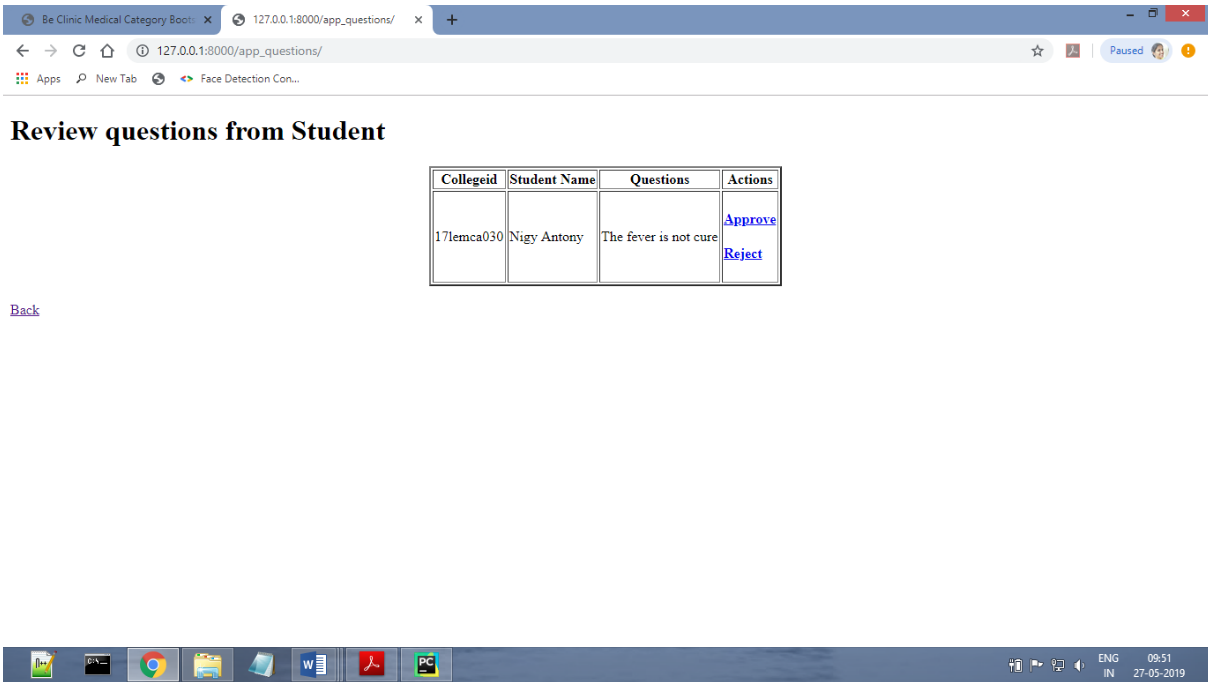
Full Name

Nicy Antony

Gender

Female

Questions form



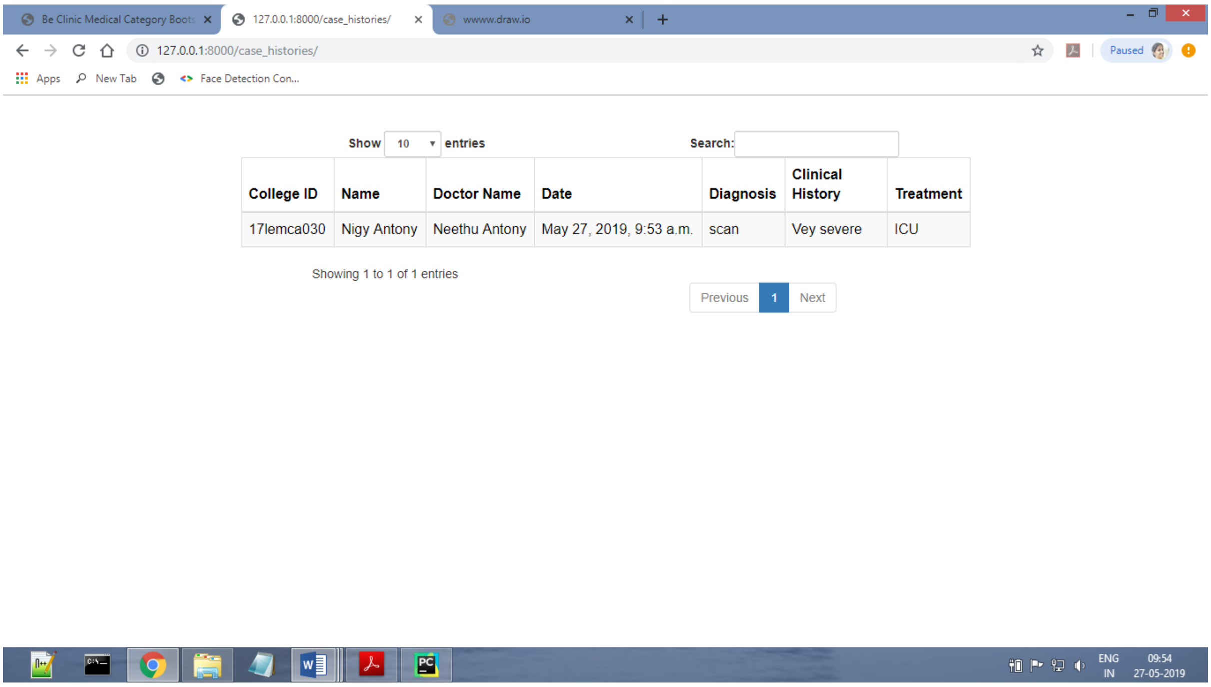
Be Clinic Medical Category Boots
127.0.0.1:8000/app_questions/
127.0.0.1:8000/app_questions/
Paused
Apps
New Tab
Face Detection Con...
ENG
IN
09:51
27-05-2019

Review questions from Student

Collegeid	Student Name	Questions	Actions
17lemca030	Nigy Antony	The fever is not cure	Approve Reject

Back

Casehistory list

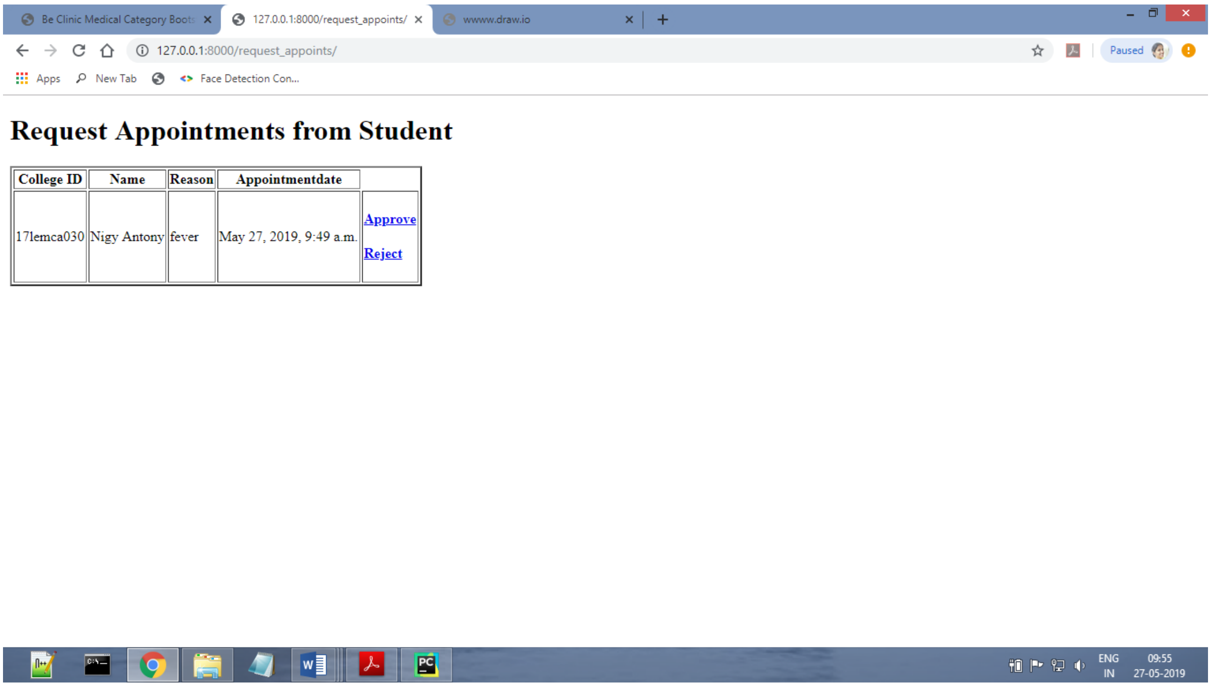


Be Clinic Medical Category Boots
127.0.0.1:8000/case_histories/
www.draw.io
127.0.0.1:8000/case_histories/
Paused
Apps
New Tab
Face Detection Con...
Show 10 entries
Search:

College ID	Name	Doctor Name	Date	Diagnosis	Clinical History	Treatment
17lemca030	Nigy Antony	Neethu Antony	May 27, 2019, 9:53 a.m.	scan	Vey severe	ICU

Showing 1 to 1 of 1 entries

Request appointments



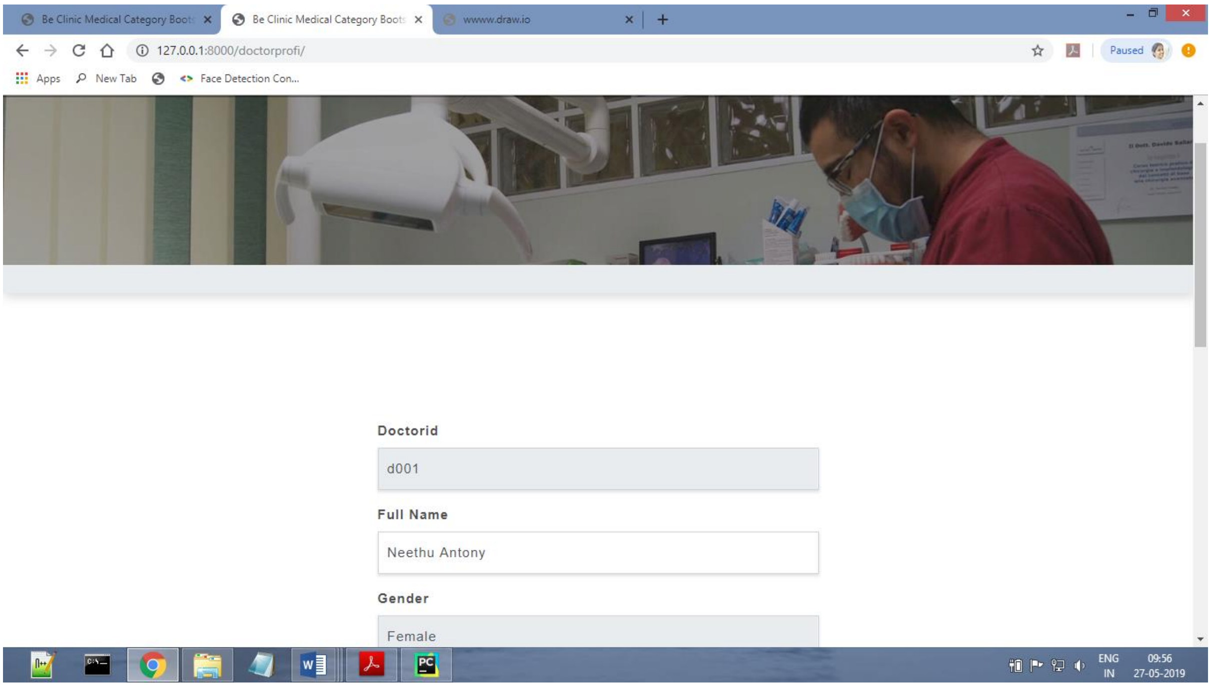
Be Clinic Medical Category Boots
127.0.0.1:8000/request_appoints/
www.draw.io
127.0.0.1:8000/request_appoints/
Paused
Apps
New Tab
Face Detection Con...

Request Appointments from Student

College ID	Name	Reason	Appointmentdate	
17lemca030	Nigy Antony	fever	May 27, 2019, 9:49 a.m.	Approve Reject

DOCTOR

Profile editing



Be Clinic Medical Category Boots
Be Clinic Medical Category Boots
www.draw.io
127.0.0.1:8000/doctorprofi/
Apps
New Tab
Face Detection Con...
Paused
ENG
IN
09:56
27-05-2019

Doctorid

d001

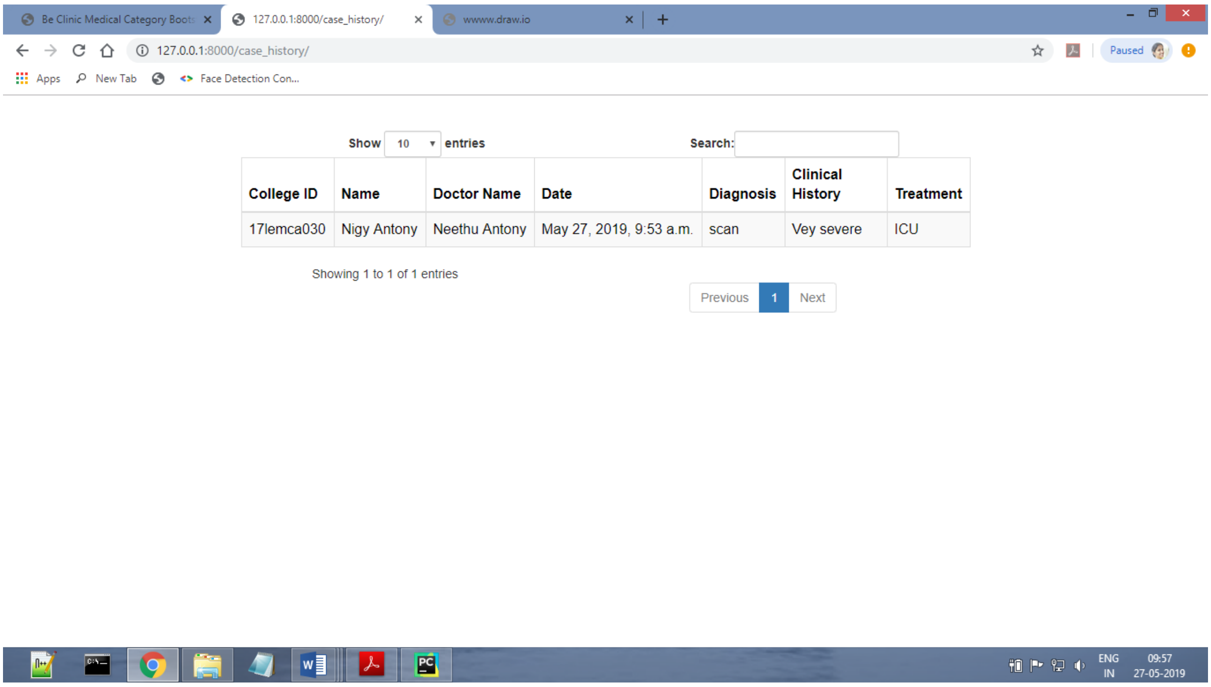
Full Name

Neethu Antony

Gender

Female

Casehistory view



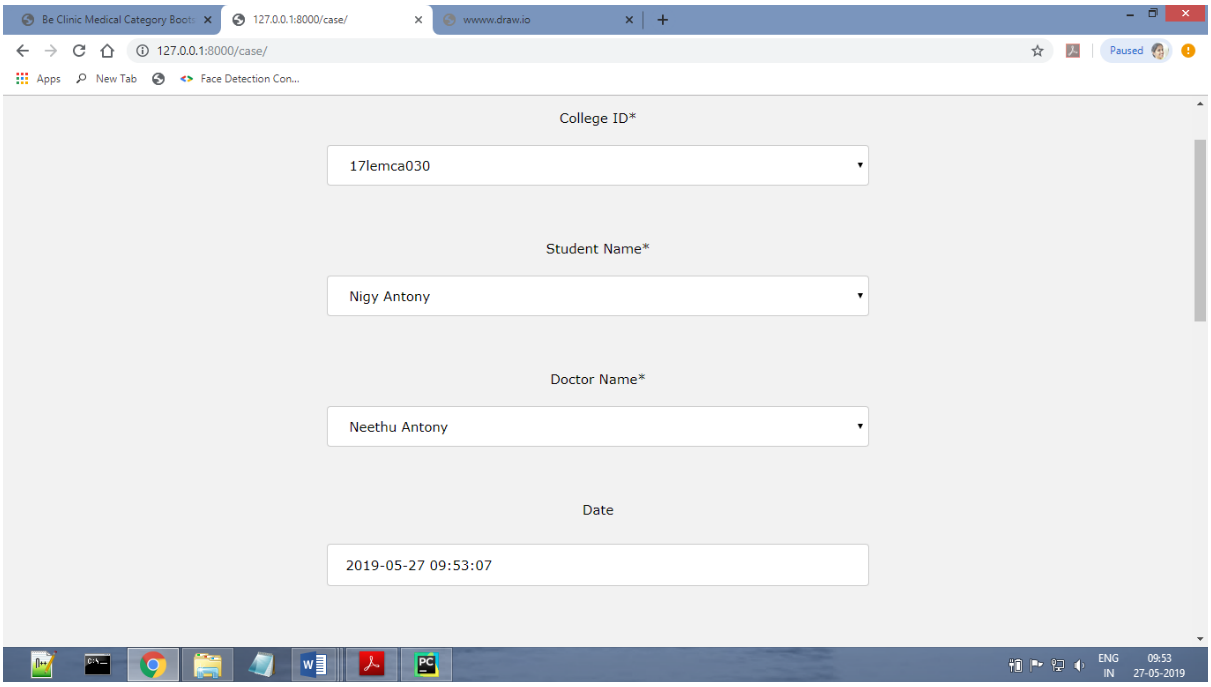
Be Clinic Medical Category Boot
127.0.0.1:8000/case_history/
www.draw.io
127.0.0.1:8000/case_history/
Paused
Apps
New Tab
Face Detection Con...
ENG
IN
09:57
27-05-2019

Show 10 entries Search:

College ID	Name	Doctor Name	Date	Diagnosis	Clinical History	Treatment
17lemca030	Nigy Antony	Neethu Antony	May 27, 2019, 9:53 a.m.	scan	Vey severe	ICU

Showing 1 to 1 of 1 entries

Case history adding



College ID*

17lemca030

Student Name*

Nigy Antony

Doctor Name*

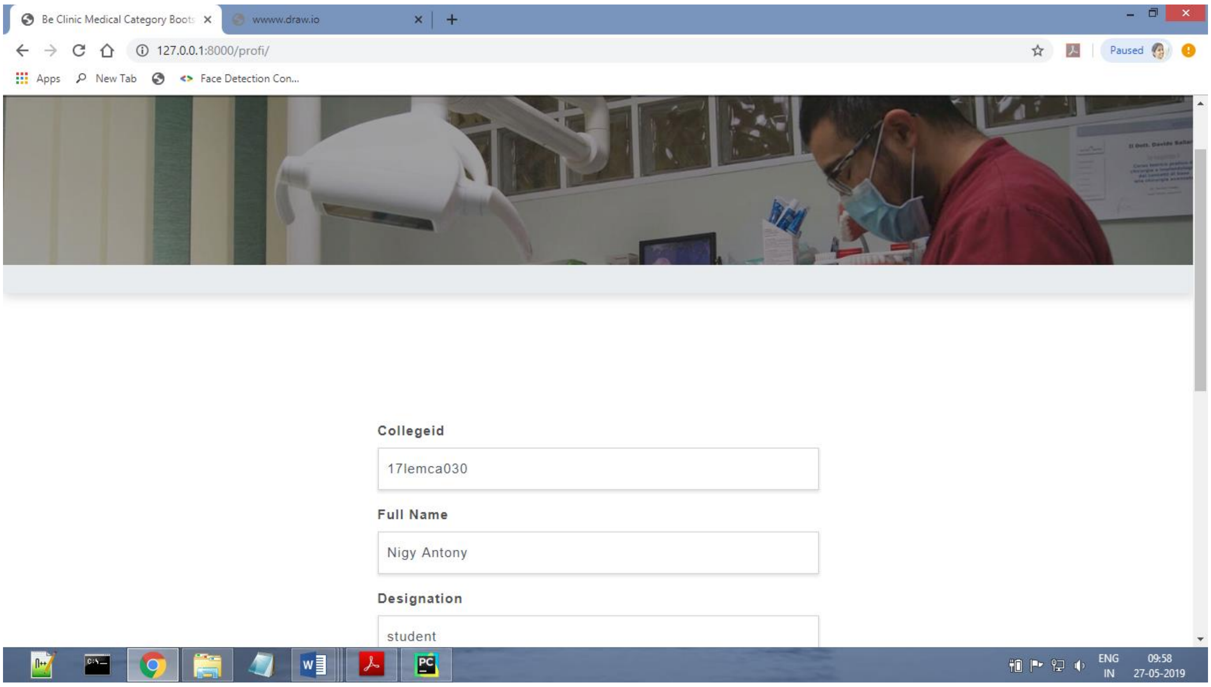
Neethu Antony

Date

2019-05-27 09:53:07

STUDENT

Profile editing



Collegeid

17lemca030

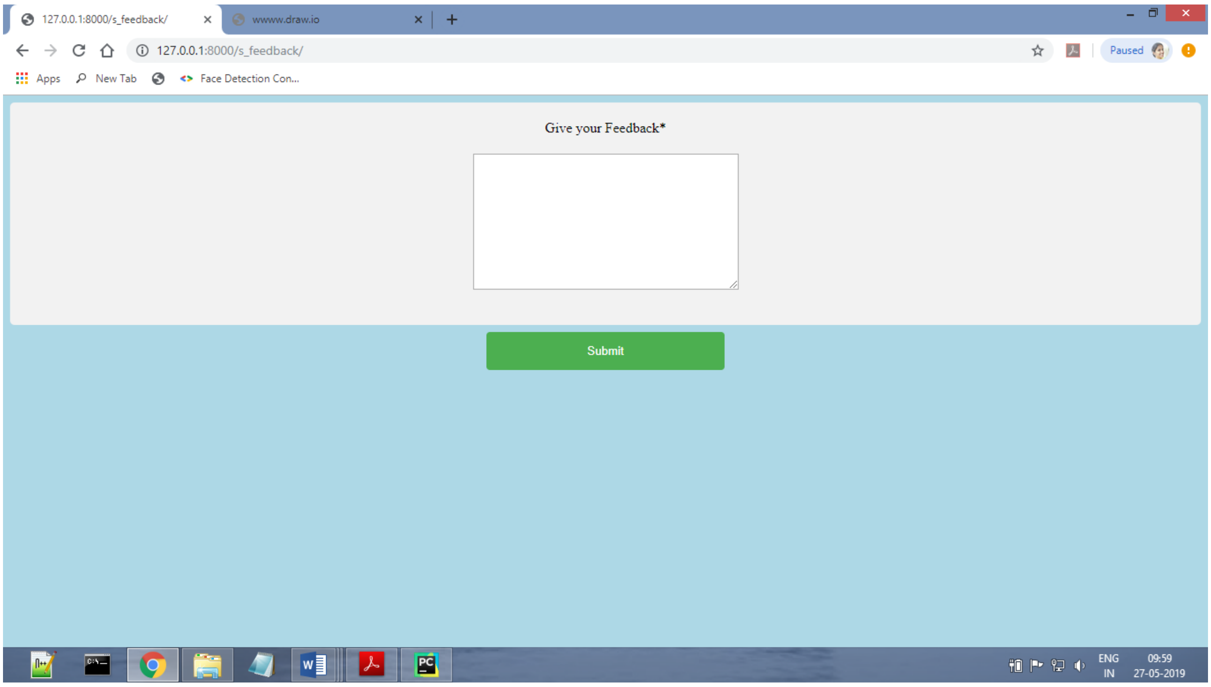
Full Name

Nigy Antony

Designation

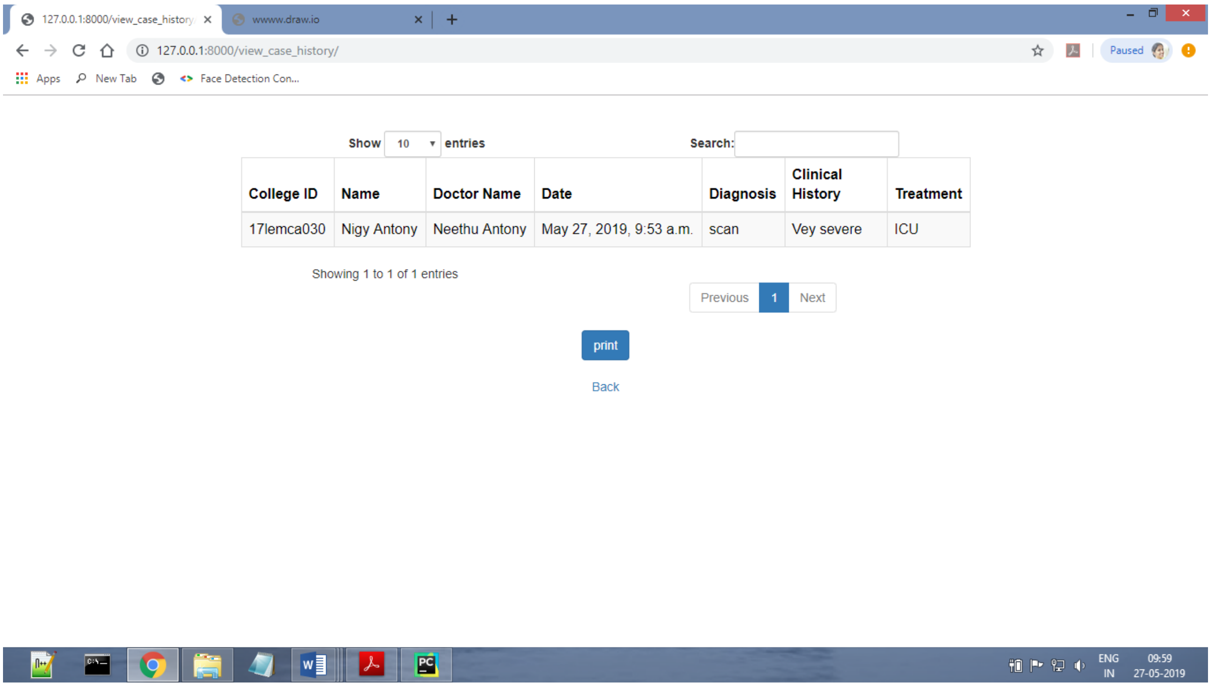
student

Feedback sending form



Give your Feedback*

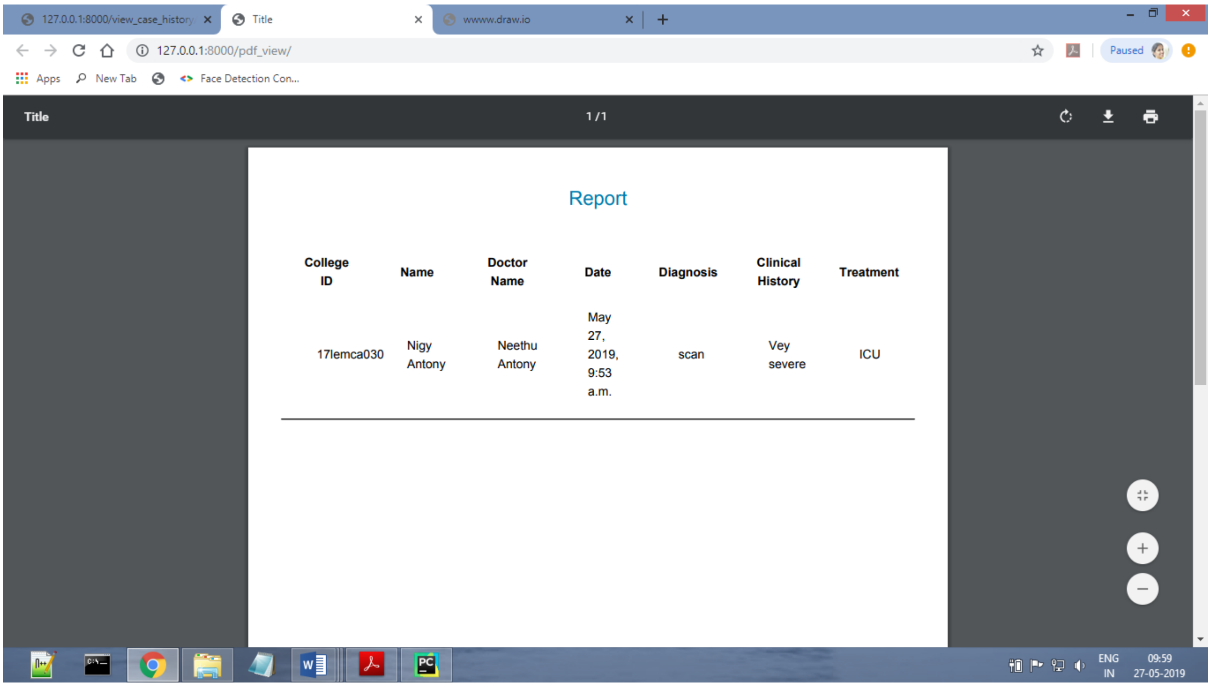
Viewing Casehistory



College ID	Name	Doctor Name	Date	Diagnosis	Clinical History	Treatment
17lemca030	Nigy Antony	Neethu Antony	May 27, 2019, 9:53 a.m.	scan	Vey severe	ICU

Showing 1 to 1 of 1 entries

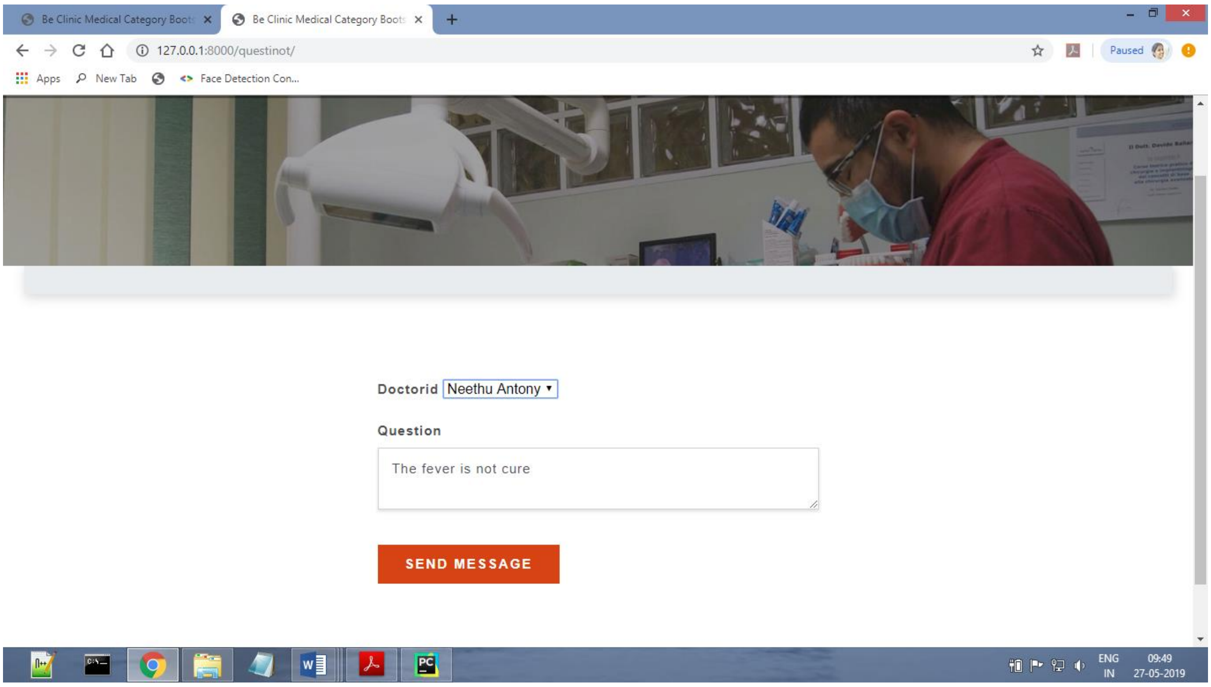
Printout of casehistory



Report

College ID	Name	Doctor Name	Date	Diagnosis	Clinical History	Treatment
17lemca030	Nigy Antony	Neethu Antony	May 27, 2019, 9:53 a.m.	scan	Vey severe	ICU

Question sending form



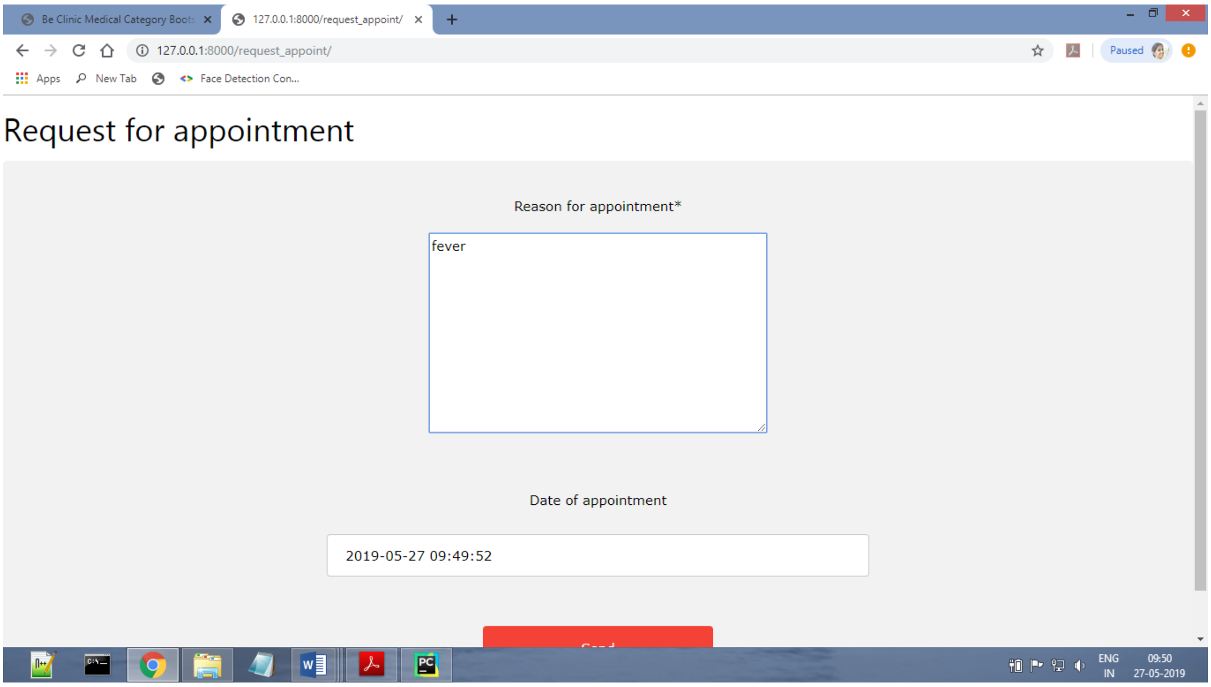
Doctorid Neethu Antony

Question

The fever is not cure

Be Clinic Medical Category Boots
Be Clinic Medical Category Boots
127.0.0.1:8000/questinot/
Apps
New Tab
Face Detection Con...
Paused
SEND MESSAGE
ENG
IN
09:49
27-05-2019

Request for appointment



Request for appointment

Reason for appointment*

fever

Date of appointment

2019-05-27 09:49:52

Be Clinic Medical Category Boot
127.0.0.1:8000/request_appoint/
127.0.0.1:8000/request_appoint/
Paused
Apps
New Tab
Face Detection Con...
ENG
IN
09:50
27-05-2019

Sample Code

```
from django import forms
from django.forms import DateInput
from testapp.models import *

class CaseForm(forms.ModelForm):
    class Meta:
        model = case_history
        fields = ['username','Name','ids','date','Diagnosis','clinicalhistory','treatment']

        labels ={
            "username": "College ID",
            "Name": "Student Name",
            "ids":  "Doctor Name",
            "date": "Date",
            "Diagnosis": "Diagnosis",
            "clinicalhistory": "Clinical history",
            "treatment":"Treatment"

        }




from django.db import models


class case_history(models.Model):
    username = models.ForeignKey(logins,on_delete=models.CASCADE)
    ids = models.ForeignKey(doctors,on_delete=models.CASCADE)
    Name = models.ForeignKey(Register, on_delete=models.CASCADE)
    date = models.DateTimeField(default=datetime.now, blank=True)
    Diagnosis = models.TextField()
    clinicalhistory = models.TextField()
    treatment = models.TextField()


<!DOCTYPE html>
<html lang="en">
<head>
    <style>
input[type=text], select {
  width: 50%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 20%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>
    <meta charset="UTF-8">
    <title></title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    {%load crispy_forms_tags %}
</head>
<body>
<h1> Add Case History</h1>
<div class="container" id="formWrapper">
 <form method="post" autocomplete="off" align="center">
     {%csrf_token%}
     {{formcase|crispy}}
     <input type="submit" class="w3-button w3-red" value="ADD">
 </form>
    </div>
</body>
</html>


<html>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
<link rel="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link rel="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap.min.css">
<script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.10.19/js/dataTables.bootstrap.min.js"></script>
</head>
<style>
form {
  display: table;
}
label {
  display: table-row;
}
input {
  display: table-cell;
}
p1 {
  display: table-cell;
}
</style>
<div id="header_order_form" align="center">
    <br>
  <form id="order_header" method="post" enctype="multipart/form-data">
       {% csrf_token%}
      <br>
                  <table id="example" class="table table-striped table-bordered"
style="width:200%">
    <thead>
        <tr>
            <th style="color:black;">College ID </th>
            <th style="color:black;">Name</th>
            <th style="color:black;">Doctor Name</th>
            <th style="color:black;">Date</th>
            <th style="color:black;">Diagnosis </th>
            <th style="color:black;">Clinical History </th>
            <th style="color:black;">Treatment</th>

        </tr>
    </thead>
    <tbody>
        {% for d in dict %}
        <tr>
            <td style="color:black;">{{d.Collegeid}}</a></td>
            <td style="color:black;">{{d.Name}} </td>
            <td style="color:black;">{{d.name}} </td>
            <td style="color:black;">{{d.date}} </td>
            <td style="color:black;">{{d.Diagnosis}} </td>
            <td style="color:black;">{{d.clinicalhistory}} </td>
            <td style="color:black;">{{d.treatment}} </td>
        </tr>
        {%endfor%}

    </tbody>

</table>
  </form>
</div>
</body>
<script>
$( document ).ready(function() {
$('#example').DataTable();
});
</script>
</html>
```