

HOSPITAL E-TOKEN MANAGEMENT

Project Report Submitted by

NEETHU JOHN
Reg. No: LAJC18MCA040

In Partial fulfillment for the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS (MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

2019-2021

AMAL JYOTHI COLLEGE OF ENGINEERING

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,

Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS



CERTIFICATE

This is to certify that the project entitled **“HOSPITAL E-TOKEN MANAGEMENT”** is a bonafide record of the work done by **Neethu John LAJC18MCA040** during the academic year **2019-2021** carried out under our supervision. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report. The work report has been approved as it satisfies the academic requirements in respect of the project work prescribed by the university for the Master of Computer Applications Degree. Certified further, that to the best of our knowledge the exact work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this to any other candidate.

Fr. Rubin Thottupuram

Sr.Elsin chakkalackal S.H

Mrs. Ankitha Philip

Head of the Department

Project Coordinator

Project Supervisor

Expert from dept. of Computer Science and Engineering
Amal Jyothi College of Engineering

External Expert appointed by the university

DECLARATION

I hereby declare that the project report “**HOSPITAL E-TOKEN MANAGEMENT**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2019-2021.

Date.....

KANJIRAPPALLY

NEETHU JOHN

Reg. No: LAJC18MCA040

ACKNOWLEDGEMENT

First and foremost, I thank Almighty God for his gracious guidance through the project. I take this opportunity to express my gratitude to all those who have helped me in completing the project successfully

It has been said that gratitude is the memory of the heart. I acknowledge my deep sense of gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** for providing all the infrastructural facilities for us, our Principal **Dr. Z V Lakaparampil** for providing good faculty for guidance.

I take the immense pleasure in expressing my thanks to Head of the Department of Master of Computer Applications, **Fr. Rubin Thottupuram**, for his kind patronages in making this project a successful one. I would like to extend my sincere thanks to our coordinator and my project guide **Mrs. Ankitha Philip** for her guidance and cooperation, without which this would not have been a success.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project. I convey hearty thanks to parents for the moral support, suggestion and encouragement to make this venture a success.

NEETHU JOHN

ABSTRACT

HOSPITAL E-TOKEN MANAGEMENT will automate the management of the hospital making it more efficient and error free for outdoor patient . Hospitals currently use a manual system for visiting Doctor Slip as a token. The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standards, here automate this process. A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized . Through this project we are aiming an website that helps the hospital function and managements, it also help the people who want to visit hospital. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies. The version control System used during the development time was Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

CONTENT

| Sl. No | Topic | Page No |
|----------------|--|--------------|
| | Introduction | 1 |
| 1 | Using Git as a Version Control System | 2 |
| 1.1 | Introduction to GitHub | 3 |
| 1.2 | Working With Git | 4-8 |
| 2 | Project Documentation | 9 |
| 2.1 | Introduction | 10 |
| 2.1.1 | Project Overview | 10 |
| 2.1.2 | Project Specification | 10-11 |
| 2.2 | System Study | 12 |
| 2.2.1 | Introduction | 12 |
| 2.2.2 | Existing system | 13 |
| 2.2.3 | Proposed system | 13-15 |
| 2.3 | Requirement Analysis | 15 |
| 2.3.1 | Feasibility Study | 15 |
| 2.3.1.1 | Economical Feasibility | 15 |
| 2.3.1.2 | Technical Feasibility | 16 |
| 2.3.1.3 | Behavioral Feasibility | 16 |
| 2.4 | Requirement Modeling | 17 |
| 2.4.1 | UML Diagrams | 17-28 |
| 2.5 | System Specification | 28 |
| 2.5.1 | Hardware Specification | 28 |
| 2.5.2 | Software Specification | 28 |
| 2.6 | Software Description | 28 |
| 2.6.1 | PHP | 28 |
| 2.6.2 | MySQL | 29-30 |
| 2.7 | System Design | 31 |
| 2.7.1 | Architectural Design | 31 |
| 2.7.2 | Module Design | 32-35 |
| 2.7.3 | Data Base Design | 36-41 |
| 2.8 | System Testing | 42 |

CONTENT

| | | |
|----------------|---|--------------|
| 2.8.1 | Introduction | 43-44 |
| 2.8.2 | Test Plan | 44 |
| 2.8.2.1 | Unit Testing | 44 |
| 2.8.2.2 | Integration Testing | 44 |
| 2.8.2.3 | Validation Testing | 45 |
| 2.8.2.4 | User Acceptance Testing | 45 |
| 2.8.3 | Test Case | 45-47 |
| 2.9 | Implementation | 47-48 |
| 2.9.1 | Implementation Procedure | 48 |
| 2.9.2 | User Training | 48 |
| 2.9.3 | Operational Document | 49 |
| 2.9.4 | System Maintenance | 49 |
| 2.10 | Conclusion & Future Enhancements | 50 |
| 2.10.1 | Future Enhancement | 53 |
| 2.10.2 | Conclusion | 53 |
| 2.11 | Bibliography | 54 |
| 2.12 | Appendix | 55 |
| 2.12.1 | Sample Code | 55-61 |
| 2.12.2 | Screen Shots | 62-65 |

LIST OF ABBREVIATIONS

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

INTRODUCTION

“HOSPITAL E-TOKEN MANAGEMENT” will automate the management of the hospital making it more efficient and error free for outdoor patient . Hospitals currently use a manual system for visiting Doctor Slip as a token. The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standards, here automate this process. A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized . Through this project we are aiming an application/website that helps the hospital function and managements, it also help the people who want to visit hospital. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

The version control System used during the development time was Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

1.USING GIT AS A VERSION CONTROL SYSTEM

1.1 Introduction to GitHub

GitHub is a web-based version-control and collaboration platform for software developers. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. And it is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

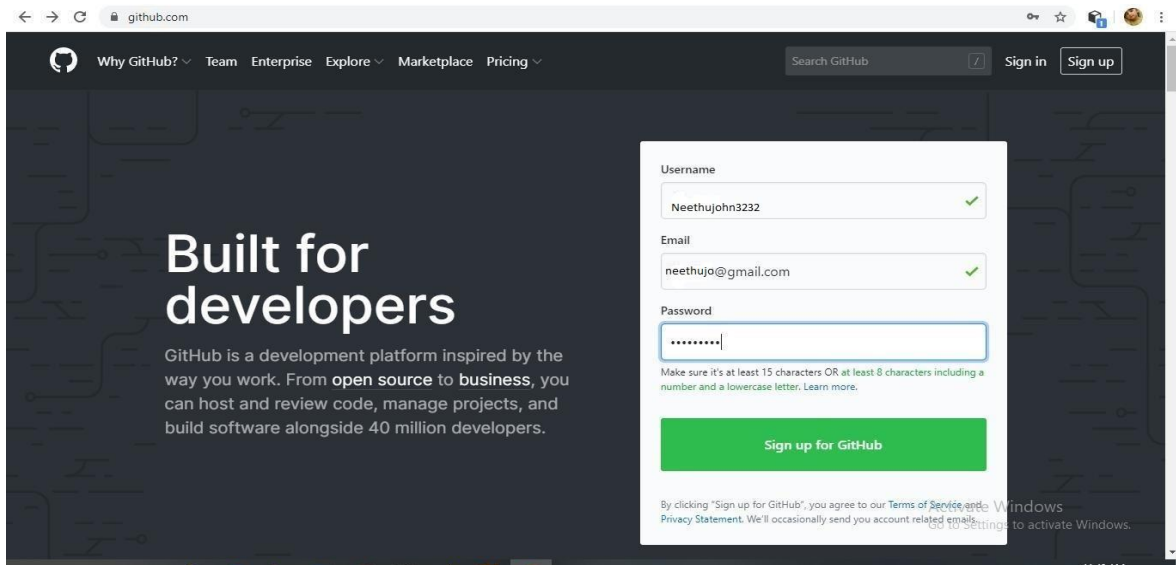
GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

GitHub products and features

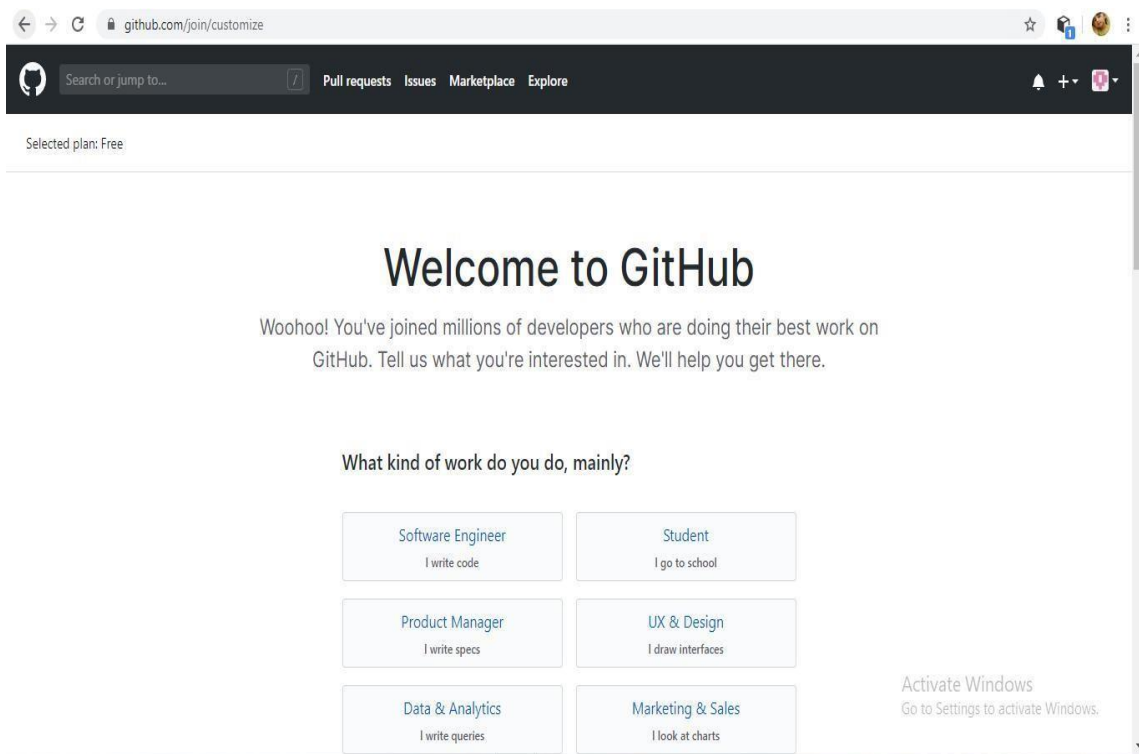
GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tool integration, as well as a litany of third-party apps and services. It offers increased security and auditable than the SaaS version.

1.2 Working with Git

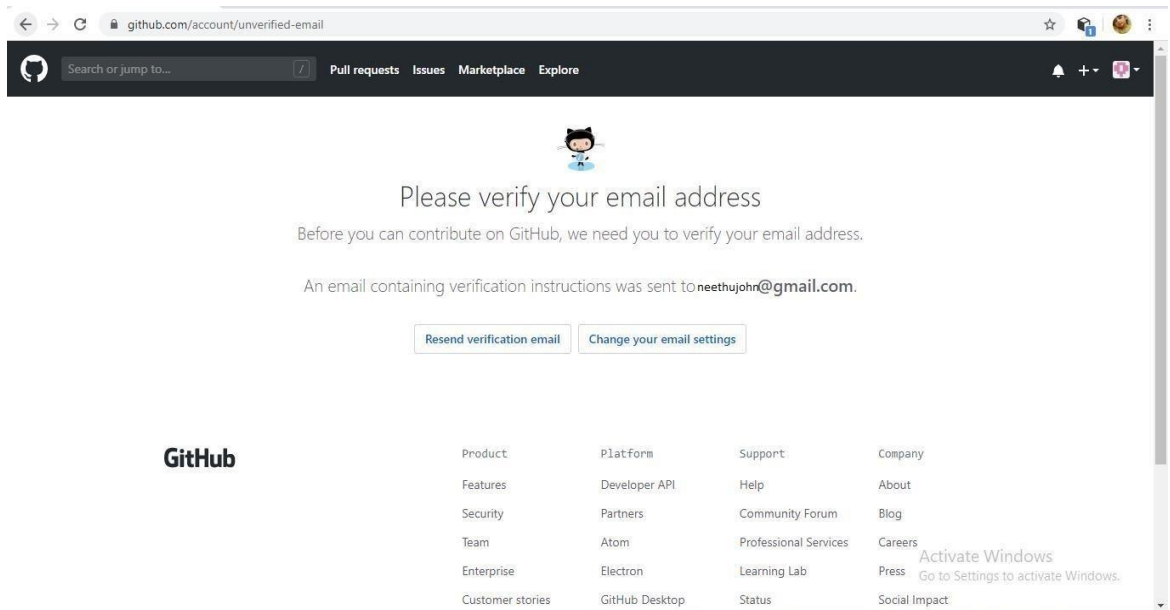
- Click on sign up button and create an account in GitHub.com



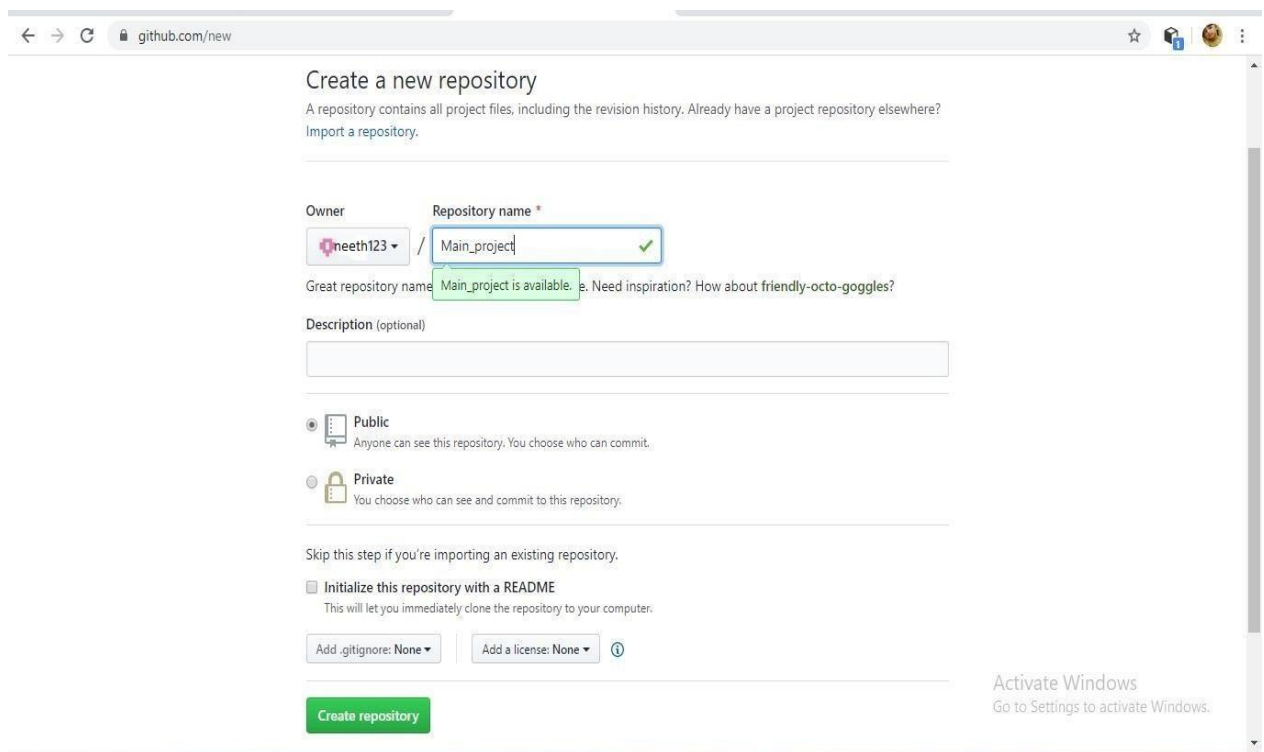
- Once successfully sign in, set up personal account, and choose your plan by selecting the options given below based on our purpose.



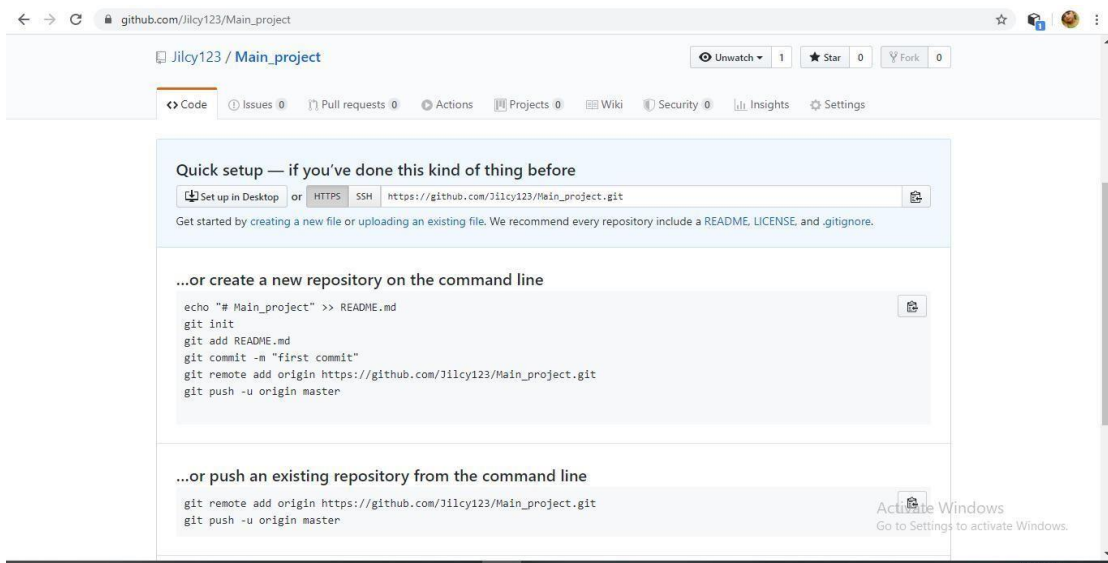
- Once this step is completed then verify the email address and you can access the home page



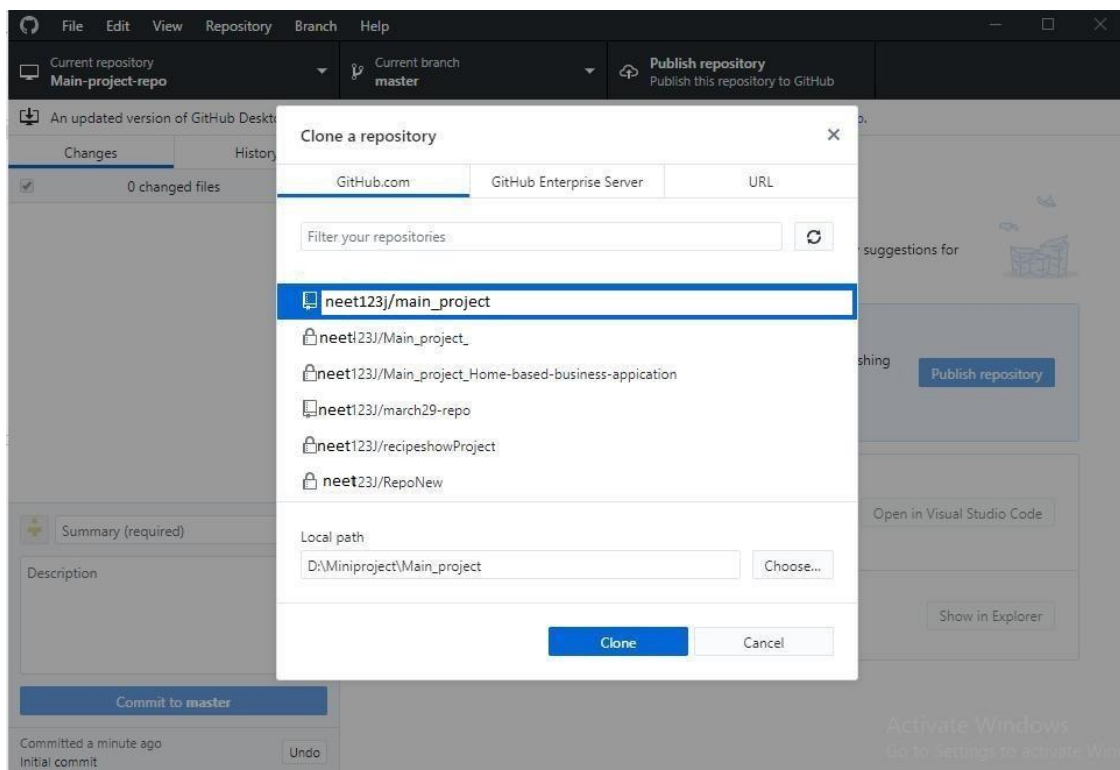
- Create a repository in GitHub



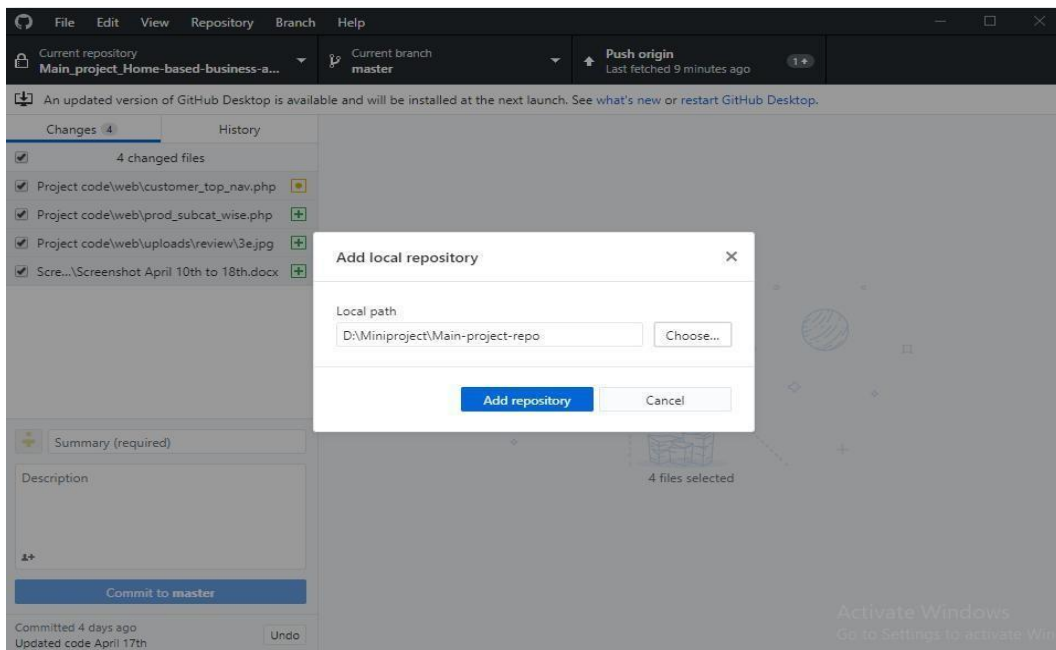
- Once repository is completed, you can setup the repository



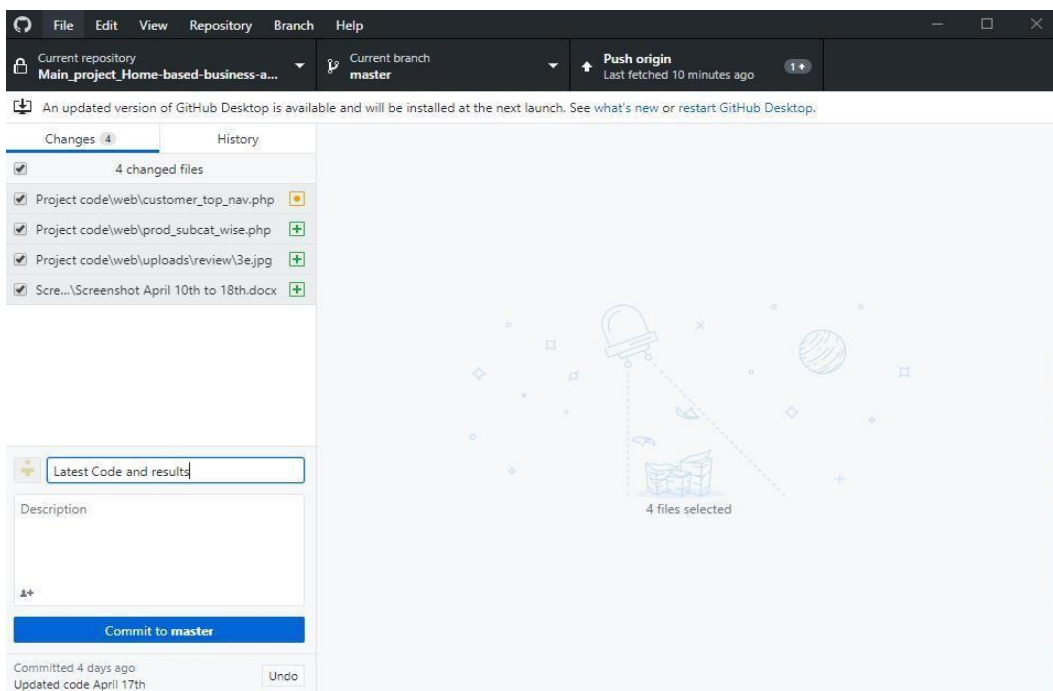
- Download and install GitHub desktop application. Once installed Go to Repository > Clone repository, and select your repository which was created in GitHub.com or Select a local system folder.



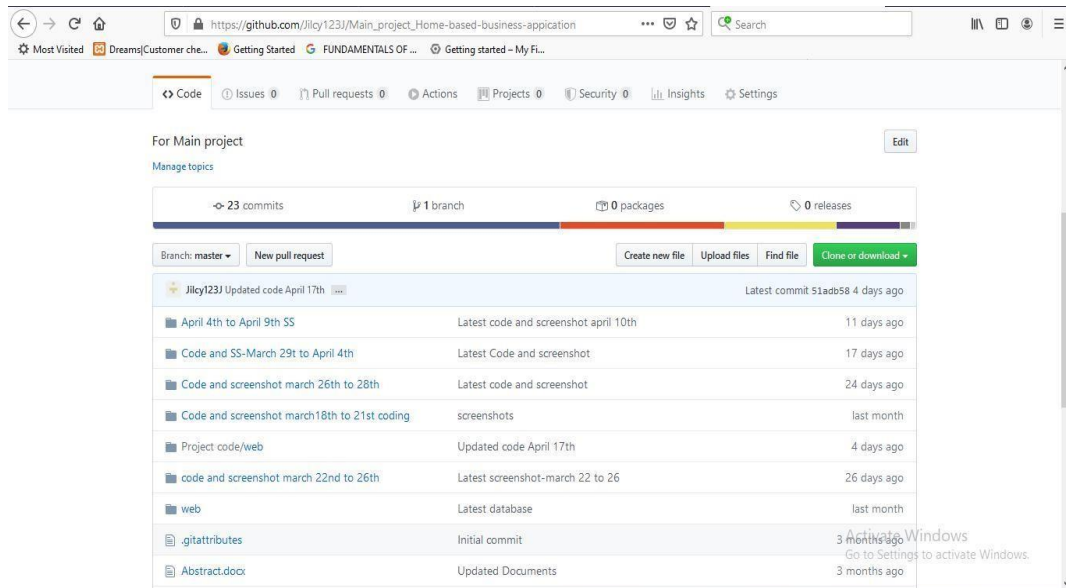
- Add local system folder, then create a local repository with folder name.



- Once repository is created, make changes on the file which is stored inside the local repository and commit to master.



- Push the local repository to origin
- Refresh GitHub.com repository to fetch commits



2. PROJECT DOCUMENTATION

2.1 INTRODUCTION

2.1.1 Project Overview

“**HOSPITAL E-TOKEN MANAGEMENT**” Hospital will automate the management of the hospital making it more efficient and error free for outdoor patient . Hospitals currently use a manual system for visiting Doctor Slip as a token. The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standards, here automate this processes. A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized . Through this project we are aiming an application/website that helps the hospital function and managements, it also help the people who want to visit hospital. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

2.1.2 Project Specification

The proposed system is a website the Hospitals currently use a manual system for visiting Doctor Slip as a token. The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standards, here automate this processes. In e-token management the user can take appointment for doctors online and the user or patient will get a token with a particular time-slot, the hospital can manage the doctors, department and laboratory etc

The system includes 3 modules. They are:

1. Admin Module

The site admin have an overall control on the website. He can perform the functionalities like,

- ☐ Login to the Application
- ☐ Add// Delete hospital
- ☐ View hospitals
- ☐ Update hospitals

2. Hospital Module

Hospital can perform functionalities like,

- ☐ Login and change password
- ☐ View appointed patients, generate reports
- ☐ Add doctors, view doctors update/delete doctors informations
- ☐ Add/View/Update Departments
- ☐ Schedule time and department for doctors
- ☐ Add laboratory and view booked patients and interested users of blood bank.
- ☐ View hospital e-token whole reports.

3. Patient Module

Patients or user can perform the functionalities like,

- ☐ Registration/Login and View/Manage profile/change password
- ☐ Search for the hospital and view hospitals and their facilities
- ☐ view hospital doctors and view hospital departments
- ☐ view token information and can download token details
- ☐ Take an appointment for a particular doctor by selecting a department
- ☐ View and book for laboratory test
- ☐ Send an interest to blood bank
- ☐ Generate sales report based on a given from and to date.
- ☐ Payment with Paytm Gateway

2.2SYSTEM STUDY

2.1.3 Introduction

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.1.4 EXISTING SYSTEM

There is no automated existing system to manage hospitals tokens to manage hospitals processes. The hospital management use a manual system for visiting Doctor Slip as a token. The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information (on forms) is incomplete, or does not follow management standards.

Drawbacks of existing system

- ☐ No publicity for the products
- ☐ One way and face-to-face marketing
- ☐ Human effort is needed.
- ☐ Customers do not get a chance of identify good products or sellers who create these products

2.1.5 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for web growth. On such consideration the system is proposed. In our proposed system there is an admin who can view and manage all the hospitals. It allows customers to make their token appointment and do their transactions by using online payment method .Users of this proposed system are admin, hospitals and cutomers or patients. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Online token have been so drastic that it has evolved to be a part of our life Through this project we are aiming an application/website that helps the hospital function and managements, it also help the people who want to visit hospital. An efficient and cost wise fashion so that an institution's resources may be effectively utilized. Hospital E-Token management will automate the management of the hospital making it more efficient and error free for outdoor.

Advantages of proposed system

The system requires very low system resources and the system will work in almost all configurations. It has got following features:

➤ ***Ensure data accuracy: -***

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ ***Enhanced data security:-***

The experts have reinforced and highlighted the fact that the hospitals, medical centers, and clinics depend on the manual system. It can lead to higher data leakage and theft when compared to automated ones. Installing a complete and automated Hospital E-token indicated that your pieces of the information stay safe and protected from any unauthorized sources and accesses..

➤ ***Better service: -***

We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

➤ ***User friendliness and interactive: -***

The proposed system's interface help the users to perform their operations without any confusions or difficulties. A customer can easily find their interested hospital and can able to take appointments . A patient can download the token with a particular time-slot .

➤ ***Minimum time required:-***

The customers can view products in different ways like category wise, subcategory wise, brand wise or can view by directly searching a product name which helps to purchase a product with minimum time. Similarly sellers can manage their orders and other operations easily using this website.

➤ ***Patient self-service:-***

Patients have their own system accounts where the list of various actions can be performed. They are able to make online appointments or reservation, receive the e-token, receive the consultation of the medical specialists and many more.

2.2 REQUIREMENT ANALYSIS

2.1.6 Feasibility study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

2.3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development. The cost of project, HOSPITAL E-TOKEN was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

2.3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- ✓ Does the existing technology sufficient for the suggested one?
- ✓ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using HTML, CSS in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using HTML, CSS, PHP and MySQL in server back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

2.3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. HOSPITAL E-TOKEN, GUI is user friendly so that users can easily use it without any training.

2.2 Requirement Modeling

2.2.1 UML diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Activity diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1. UML Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

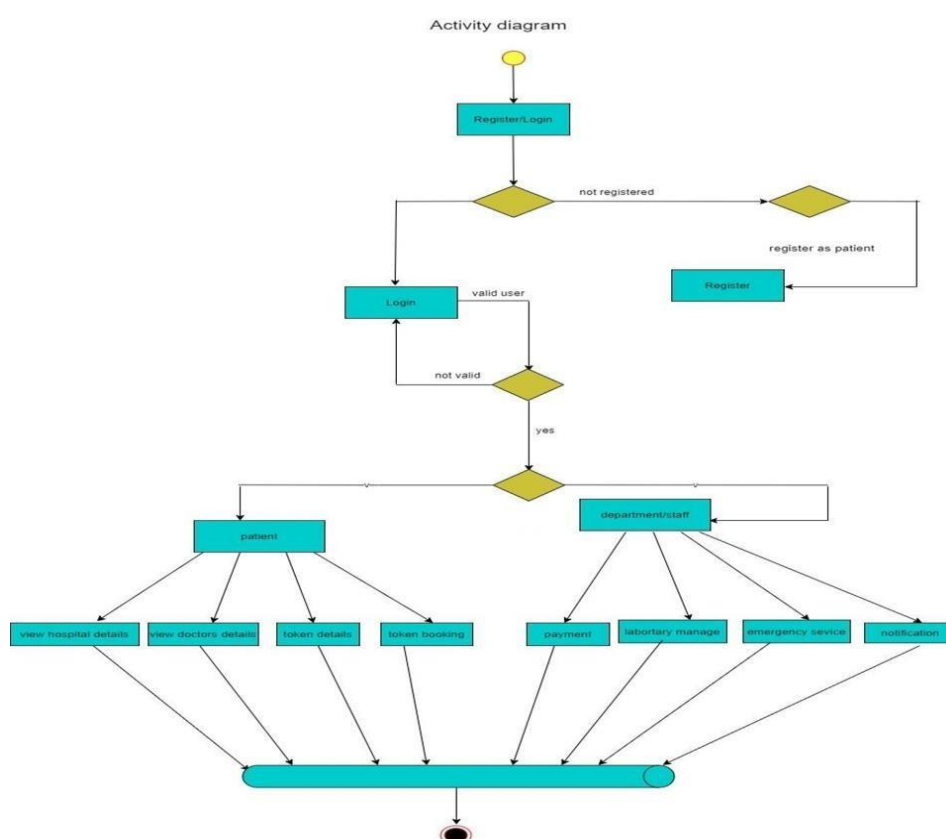


Fig. Activity diagram for hospital e-token

2. UML Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components. The boundary, which defines the system of interest in relation to the world around it. The □ actors, usually individuals involved with the system defined according to their roles. The □ use cases, which the specific roles are played by the actors within and around the system.

□

The relationships between and among the actors and the use cases.

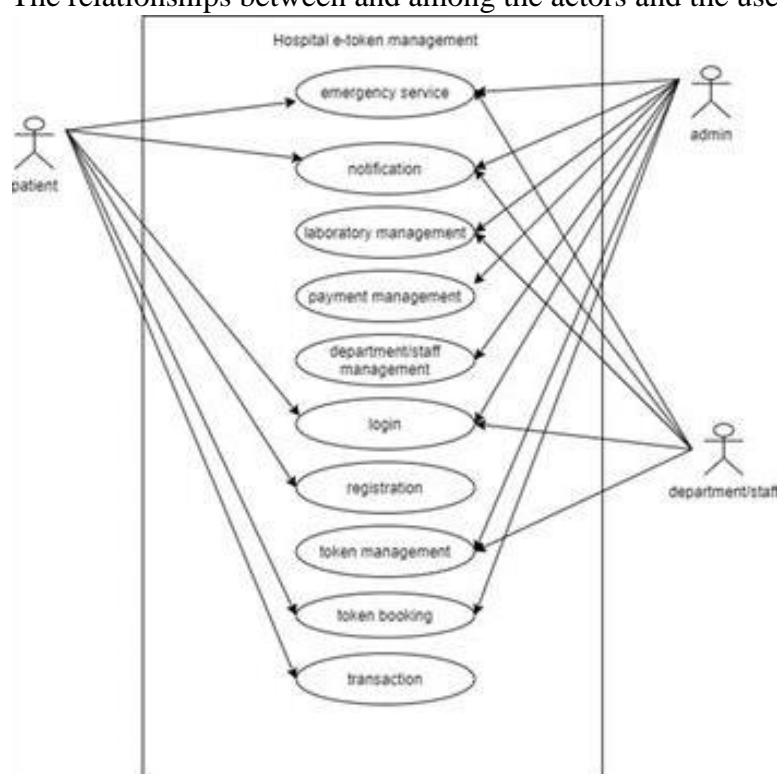


Fig. Use case diagram for Hospital E-token management

3. UML Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Notations of a Collaboration Diagram

Following are the components of a component diagram that are enlisted below:

1. **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
 - The object is represented by specifying their name and class.
 - It is not mandatory for every class to appear.
 - A class may constitute more than one object.
 - In the collaboration diagram, firstly, the object is created, and then its class is specified.
 - To differentiate one object from another object, it is necessary to name them.
2. **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
3. **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.
4. **Messages:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

Collaboration diagram

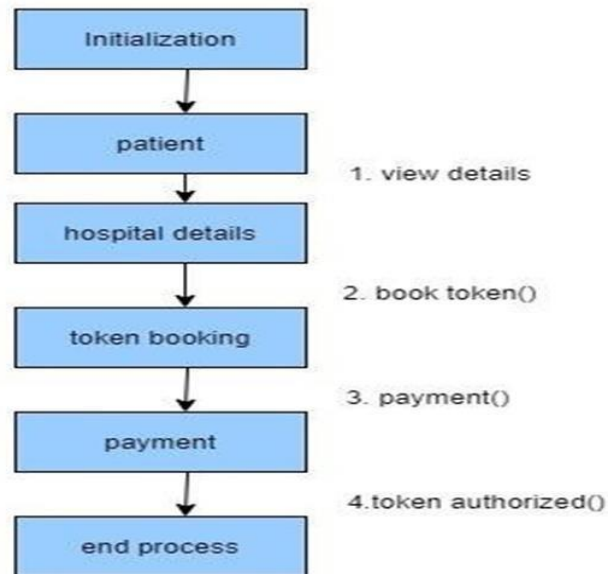


Fig. Collaboration Diagram for hospital e-token management

5. UML State Chart Diagram

It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered.

The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

State chart Diagram

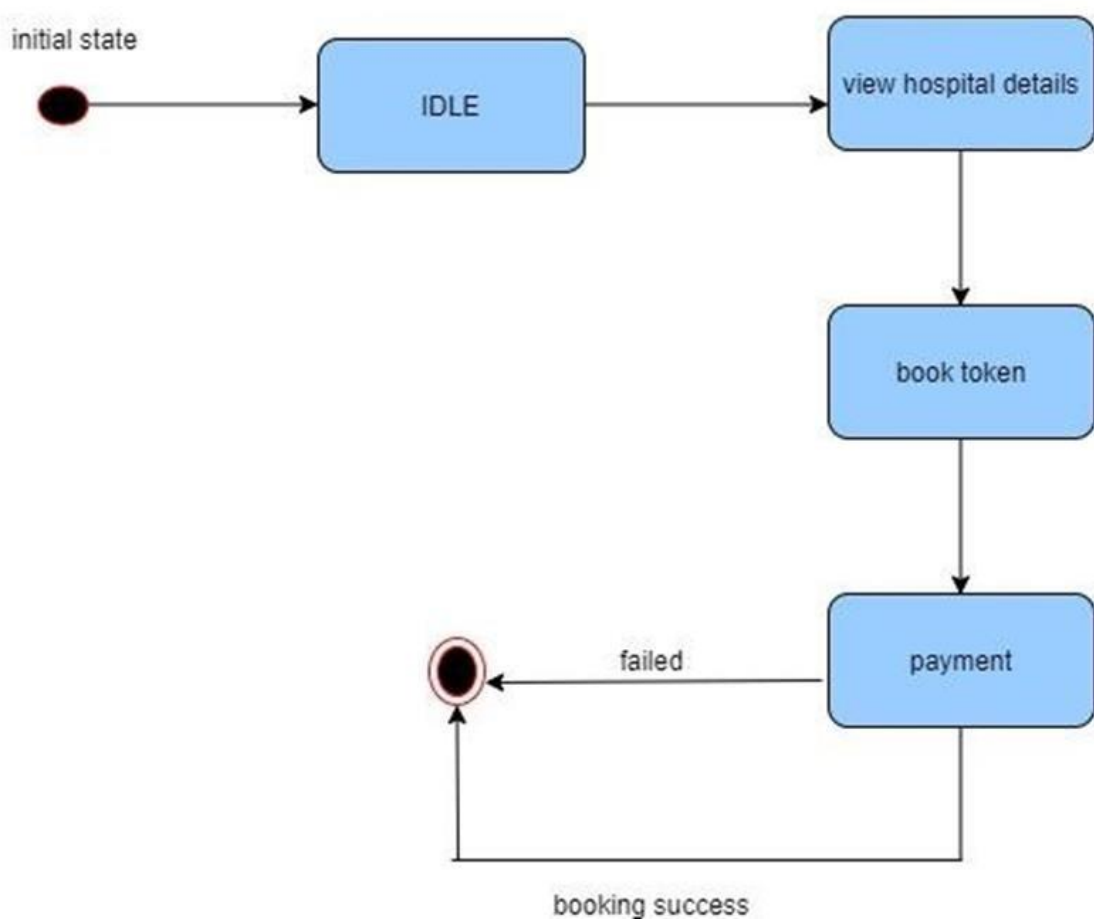


Fig. Statechart Diagram for Hospital e-token management

6 UML Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

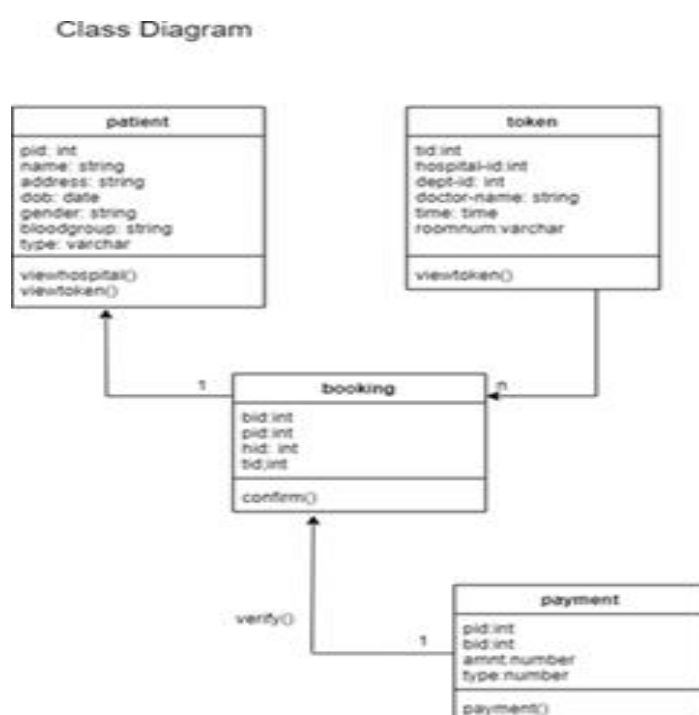


Fig. Class Diagram for hospital e-toen management

7. UML Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams.

Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of the object diagram can be summarized as –

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective.

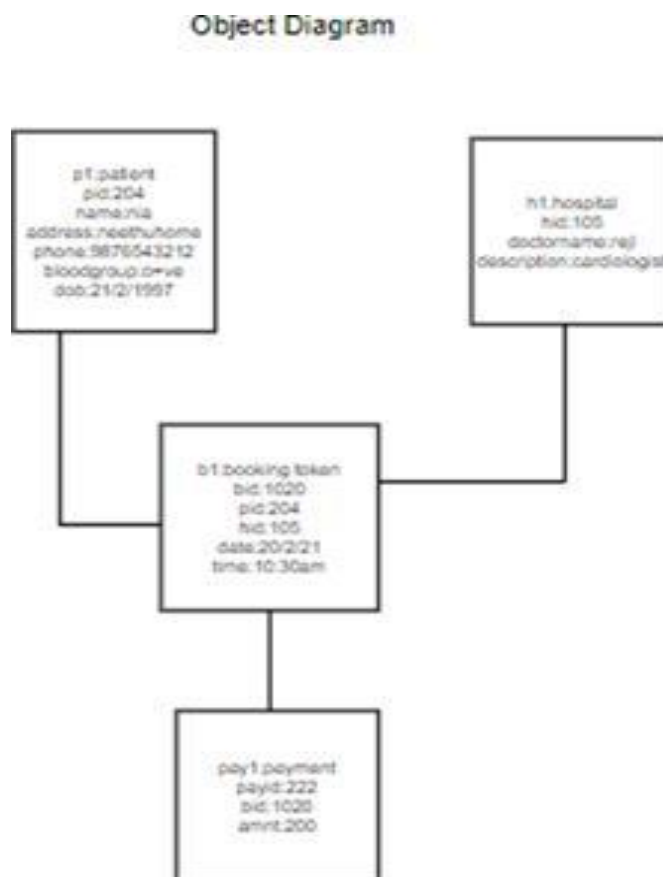


Fig. Object diagram for online shopping site

8 UML Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

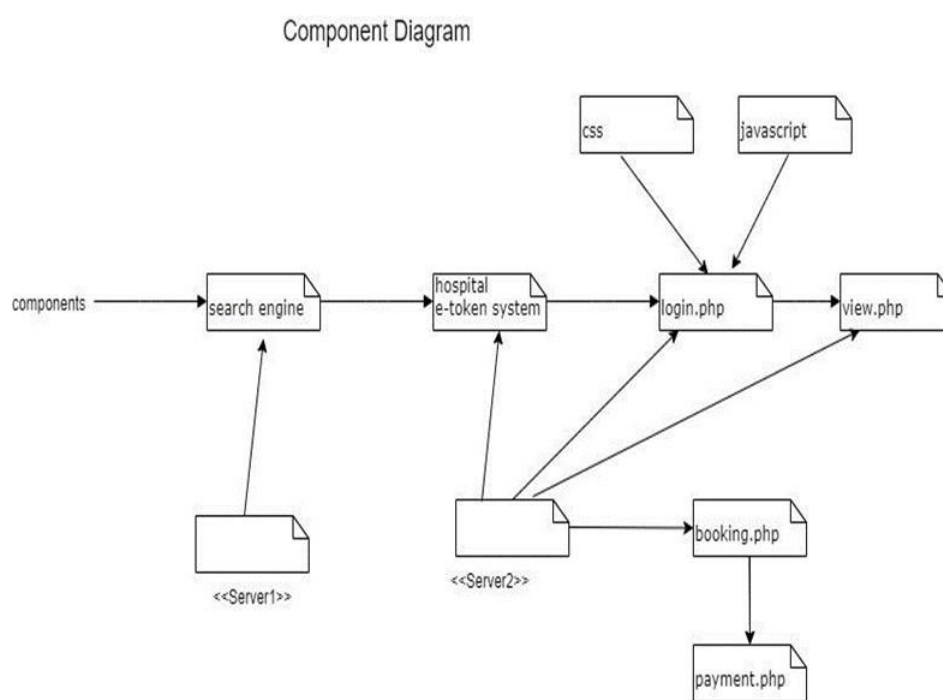


Fig. Component diagram for hospital e-token

9. UML Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

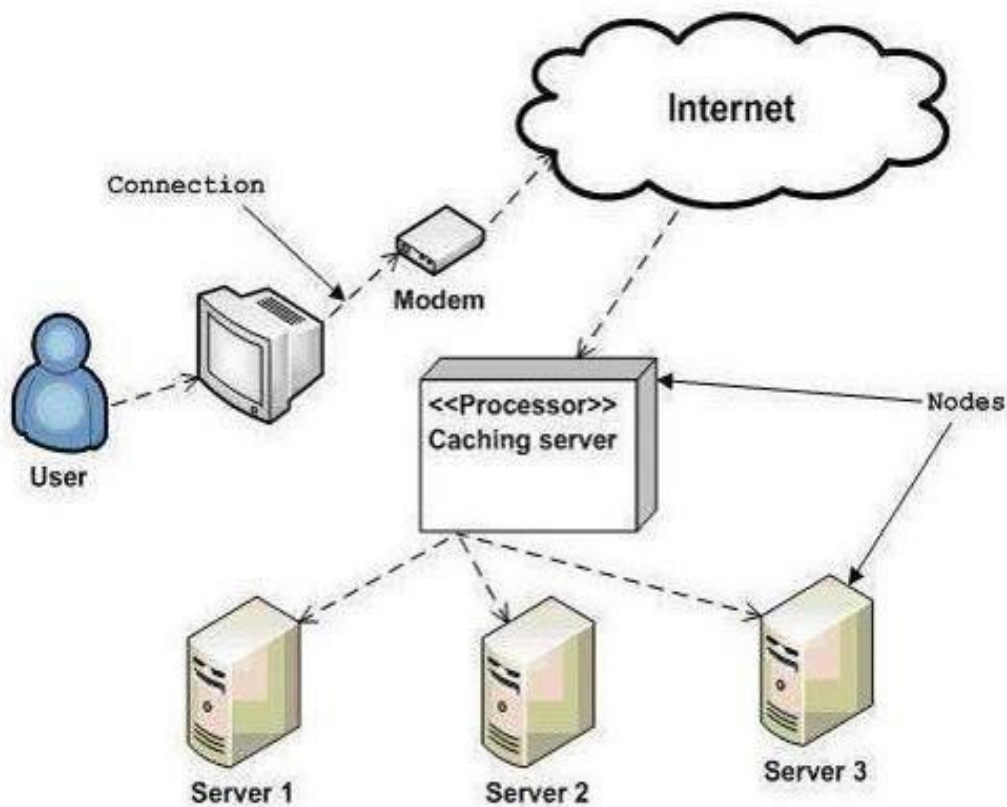


Fig. Deployment diagram

2.3 System specification

2.3.1 Hardware Specification

| | |
|-----------|-----------------|
| Processor | - Intel core i3 |
| RAM | - 4 GB |
| Hard disk | - 1 TB |

2.3.2 Software Specification

| | |
|-------------------|-------------------------------------|
| Front End | - HTML, CSS |
| Backend | - MYSQL |
| Client on PC | - Windows 7 and above. |
| Technologies used | - JS, HTML5, AJAX, JQuery, PHP, CSS |

2.4 Software Description

2.4.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data.

It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

2.4.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

✓ **MySQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

✓ **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well- designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

✓ **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

✓ **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

✓ **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

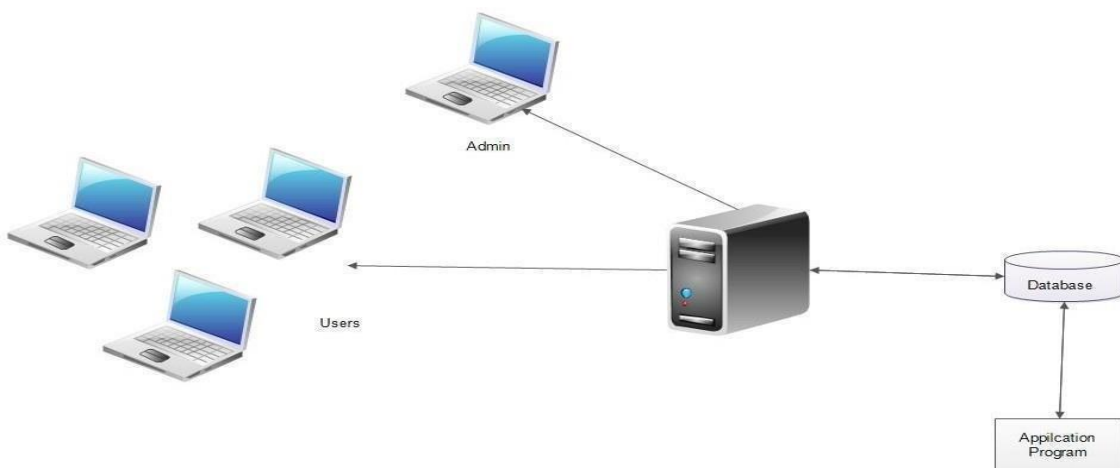
✓ **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

2.5 System Design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

2.5.1 Architectural Design



The registered customer, seller, and admin can access the website through the internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program processes the user's request and provides the required services by taking data from the system database.

2.5.2 Module Design

- **Admin Module**

The administrator of the company is allowed to access all the services in the system. Admin has the overall control of the system. Administrator control and supervise all over the activities of the system. Administrator ensure all hospital activities run on smoothly and efficiently.

- **Hospital Module**

Hospital can perform functionalities like,first ,Login and change password,View appointed patients, generate reports,Add doctors,view doctors update or delete doctors informations,Add,View and Update Departments,Schedule time and department for doctors,Add laboratory and view booked patients and interested users of blood bank. View hospital e-token whole reports.

- **Patients Module**

Patient first want to register and then login and choose one of the best one hospital and take appoiment for doctor based on particular department.After the payment process patient will got a token number with a timeslot. Patients or user can perform the functionalities like, Registration,Login and View,Manage profile and change password,Search for the hospital and view hospitals and their facilities,view hospital doctors and view hospital departments,view token information and can download token details and Take an appointment for a particular doctor by selecting a department,View and book for laboratory test then Send an interest to blood bank ,Generate sales report based on a given from and to date. then do Payment with Paytm Gateway.

2.7.3 Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies.

Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data and Choose the proper name for the data.
- ✓ Choose proper names for the tables and columns, promotes integrity

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

2.7.4 TABLES

1 .Table name: Login tbl

Primary key: login_id

| Field name | Datatype | Size | Description |
|------------|----------|------|----------------------------|
| login_id | int | 10 | Primary key of login table |
| username | Varchar | 50 | Username |
| password | Varchar | 50 | Password for login |
| role | int | 50 | Role of user |

2.Table name: registration tbl

Primary key: reg_id

| Field name | Datatype | Size | Description |
|------------|----------|------|-----------------------------------|
| reg_id | int | 10 | Primery key of registration table |
| Login_id | int | 10 | Foregin key of reg_tbl |
| name | Varchar | 20 | name of patient |
| email | varchar | 20 | email |
| phone | number | 20 | phone |
| dbo | date | 20 | Date of birth |
| Blood_grup | varchar | 20 | Blood group of patient |
| gender | varchar | 10 | Gender |
| address | varchar | 50 | Address of patient |

3. Table name: doctor_tbl

Primary key: doctor_id

| Field name | Datatype | size | Description |
|---------------|----------|------|---------------------------|
| Doctor_id | int | 11 | Primary key of doctor_tbl |
| Hospital_id | int | 11 | Foregin key of doctor_tbl |
| Department_id | int | 11 | Foregin key of doctor_tbl |

4. Table name: dept_tbl

Primary key: Dept_id

Foreign key: Hospital_id

| Field name | Datatype | size | Description |
|-------------|----------|------|-------------------------|
| Dept_id | int | 11 | Primary key of dept_tbl |
| Dept_name | varchar | 20 | Department name |
| Hospital_id | Int | 11 | Foreign key of dept_tbl |

5. Table name: Doctorschedule_tbl

Primary key: schedule_id

Foreignkey: hospital_id, doc_id

| Field name | Datatype | size | Description |
|-------------|----------|------|-----------------------------|
| Schedule_id | int | 11 | Primary_key |
| Doctor_id | Int | 11 | Foreign key of schedule_tbl |
| Hospital_id | Int | 11 | Foreign key of schedule_tbl |
| Start_time | Varchar | 11 | Start_time |
| End_time | Varchar | 20 | End_time |
| status | Varchar | 20 | status |
| count | int | 11 | count |

6. Table name: booking_tbl

Primary key: booking_id

Foreignkey:schedule_id,regid

| Field name | Datatype | size | Description |
|-------------|----------|------|----------------------------|
| booking_id | int | 11 | Primary_key |
| Schedule_id | int | 11 | Foreign key of booking tbl |
| reg_id | Int | 11 | Foreign key of reg tbl |
| Token_no | Varchar | 20 | Number of token |
| Hospital_id | int | 11 | Foregin key |
| time | time | 20 | Time of doctors |
| status | int | 11 | status |

7. Table name: blood_donation_tbl

Primary key: don_id

Foreignkey: reg_id

| Field name | Datatype | Size | Description |
|-------------|----------|------|--------------------------|
| don_id | Int | 11 | Primary key |
| Reg_id | Int | 11 | Foreign key of login tbl |
| Hospital_id | int | 50 | Date |

8. Table name: hospital_tbl

Primary key: hospital_id

| Field name | Datatype | size | Description |
|-------------|----------|------|-----------------------------|
| hospital_id | int | 11 | Primary key of Hospital_tbl |
| Login_id | int | 11 | foreignkey |
| Hosp_name | Varchar | 20 | Name of hospital |
| address | Varchar | 20 | Hospital address |
| Contact_no | number | 20 | Phone number |
| email | varchar | 20 | email |

9. Table name: lab_booking_tbl

Primary key: lab_book_id

| Field name | Datatype | size | Description |
|-------------|----------|------|-----------------------------|
| Lab_book_id | int | 11 | Primary key of Lab_book tbl |
| reg_id | int | 11 | Foreign key |
| Test_id | int | 11 | Foreign key |
| Hospital_id | int | 11 | Foreign key |
| Test_date | date | 10 | Test_date |
| Test_time | time | 10 | Test_date |

10. Table name: lab_test tb

Primary key: test_id

| Field name | Datatype | size | Description |
|-------------|----------|------|-----------------------------|
| Test_id | int | 11 | Primary key of Lab test tbl |
| hospital_id | int | 11 | Foreign key of lab test tbl |
| Test_name | Varchar | 20 | Name of test |
| Test_rate | int | 20 | Rate of test |
| Time_taken | time | 10 | time |
| description | varchar | 20 | |

11. Table name: payment tb

Primary key: pay_id

| Field name | Datatype | size | Description |
|------------|----------|------|----------------------------|
| pay_id | int | 11 | Primary key of Payment tbl |
| reg_id | int | 11 | Foreign key of pay table |
| Test_id | int | 11 | Foreign key payment tbl |
| Test_rate | Varchar | 20 | rate |

2.6 System Testing

2.6.1 Introduction

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult.

2.6.2 Test Plan

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ☐ Unit testing
- ☐ Integration Testing
- ☐ Data validation Testing
- ☐ Output Testing

2.8.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested. Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot.

Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

2.8.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

2.8.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

2.8.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

2.6.3 Test Case

| | | | | | |
|---|---------------------------------------|---|--|--|-------------------|
| Test Case 1 | | | | | |
| Project Name: HOSPITAL E-TOKEN MANAGEMENT | | | | | |
| Login Test Case | | | | | |
| Test Case ID:Fun_1 | | | Test Designed By: NEETHU JOHN | | |
| Test Priority(Low/Medium/High):High | | | Test Designed Date: 20-05-2021 | | |
| Module Name:Login Screen | | | Test Executed By : Mrs. Ankitha Philip | | |
| Test Title : Verify login with valid username and password | | | Test Execution Date: 20-05-2021 | | |
| Description: Test the Login Page | | | | | |
| Pre-Condition :User has valid user name and password | | | | | |
| Step | Test Step | Test Data | Expected Result | Actual Result | Status(Pass/Fail) |
| 1 | Navigation to Login Page | | Login Page should be | Login page displayed | Pass |
| 2 | Provide Valid User name | User Name: admin@gmail.com | User should be able to Login | User Logged in and navigated to Admin Dashboard with records | Pass |
| 3 | Provide Valid Password | Password: Admin123* | | | |
| 4 | Click on Login button | | | | |
| 5 | Provide Invalid User Name or password | User Name: Admin1234 Password: admin | User should not be able to Login | Message for enter valid user or password displayed | Pass |
| 6 | Provide Null User Name or Password | User Name:null Password: null | | | |
| 7 | Click on Login button | | | | |
| Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database | | | | | |

| | | | | | |
|--|---------------------------------|--|--|--|-------------------|
| Test Case 2 | | | | | |
| Project Name:HOSPITAL E-TOKEN | | | | | |
| User Registration Test Case | | | | | |
| Test Case ID:Fun_2 | | | Test Designed By: NEETHU JOHN | | |
| Test Priority(Low/Medium/High):Medium | | | Test Designed Date: 20-05-2021 | | |
| Module Name:User Registration | | | Test Executed By : Mrs. Ankitha Philip | | |
| Test Title : Verify new user registration | | | Test Execution Date: 20-05-2021 | | |
| Description: Test the User registration Page | | | | | |
| Pre-Condition:Usershouldnotbealreadyregistered | | | | | |
| Step | Test Step | Test Data | Expected Result | Actual Result | Status(Pass/Fail) |
| 1 | Navigation to Registration Page | | Registration Page should be displayed | Registrationpage displayed | Pass |
| 2 | Provide null information | Name: null | Validationmessage should be displayed | *Mandatoryfield message displayed | Pass |
| 3 | Provide Valid Details of user | All the mandatory registration details of the user | User should be able to register successfully | Registrationsuccessful and home page displayed | Pass |
| 4 | Click on Register button | | | | |
| Post-Condition: User is validated with database and successfully registered in the application. The Account session details are logged in database | | | | | |

2.7 Implementation

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

2.7.1 Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
 - Their confidence in the software is built up.
 - Proper guidance is imparted to the user so that he is comfortable in using the application.
- Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

2.7.2 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

2.7.3 Operational Document

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

2.7.4 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

2.8 Conclusion and Future Enhancements

2.8.1 Future Enhancement

- ☐ Provide option to take appointments online and will get a token with particular time-slot,so no need for wait for the doctor for long time
- ☐ Include an option for download the token and keep it safe
- ☐ Provide more security.

2.8.2 Conclusion

The current system automates hospital management and makes outdoor patients more efficient and error-free. The hospital currently uses a manual system to help Dr. visit order as a sign. The system requires many paper forms with data storage, which are distributed throughout the hospital's management infrastructure. If the information is incomplete or does not meet management standards, these processes will be automated here. An important part of any hospital work is related to procurement and management

2.9 Bibliography

Books/References:-

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

Websites:-

- www.w3schools.com
- www.jquery.com
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.html

2.12 APPENDIX

2.12.1 SAMPLE CODE

Connection code

```
<?php
```

```
$con=mysqli_connect("localhost","root","","db_hospital_e_token")or die  
("Couldn't connect"); ?>
```

Admin

Add hospitals code:-

```
<?php
```

```
include'connection.php
```

```
'; session_start();
```

```
if(isset($_POST['sub'])
```

```
)
```

```
{
```

```
    $h_name=$_POST['hospital'];
```

```
    $contact_num=$_POST['contact'];
```

```
    $email=$_POST['email'];
```

```
    $addr=$_POST['address'];
```

```
    $user=$_POST['username'];
```

```
    $pass=$_POST['pass_word'];
```

```
    $check_mail=mysqli_query($conn,"SELECT * FROM hospital_tb WHERE email='$email'");
```

```
}
```

```
else{
```

```
mysqli_query($conn,"INSERT INTO login_tb(username,password,role,key_pass)
VALUES('$user','$token','1','$key') ");
$last_login_id=mysqli_insert_id($conn);
```

```
mysqli_query($conn,"INSERT INTO
hospital_tb(login_id,hospital_name,address,contact_no,email)VALUES('$last_login_id','$h_name','$addr','$contact_num','$email')");
```

```
require 'PHPMailer/PHPMailerAutoload.php';

$to_address =$email;

$subject ="Password";
$msg  =$pass;

$mail = new PHPMailer;

$mail->isSMTP();
$mail->Host = 'smtp.gmail.com';
$mail->SMTPAuth = true;
$mail->Username = 'mailertest28@gmail.com';
$mail->Password = 'floodRelief';
$mail->SMTPSecure = 'ssl';
$mail->Port = 465;
$mail->setFrom('mailertest28@gmail.com','hospital');

$mail->addAddress($to_address);
$mail->WordWrap = 50;

$mail->isHTML(true);

$mail->Subject = $subject;
$mail->Body = "<html>
<head>

<title>HOSPITAL E-TOKEN</title>
</head>

<body>

<table width='300'>
```

```
<th scope='row'>Mail From</th>
<td>HOSPITAL E-TOKEN </td>
</tr>
<tr>
<th scope='row'>Subject</th>
<td>$subject</td>
</tr>
<tr>
<th scope='row'>Username</th>
<td>$user</td>
</tr>
<tr>
<th scope='row'>Password</th>
<td>$msg</td>
</tr>
</table>

</body></html>
```

```
$mail->AltBody = 'msg';
```

```
//Read an HTML message body from an external file, convert referenced images to embedded,
```

```
//convert HTML into a basic plain-text alternative body
```

```
//$mail->msgHTML(file_get_contents('contents.html'), dirname(_FILE_));
```

```
if(!$mail->send())
```

```
{
```

```
    echo 'Message could not be sent.';
```

```
    echo 'Mailer Error: ' .
```

```
    $mail->ErrorInfo; exit;
```

```
}
```

```
else
```

```
{
```

```
    echo "<script>alert('Message has been sent');</script>";
```

```
}
```

```
echo "<script> alert(' hospital Registration completed'); </script>";
```

```
}
```

```
}
```

```
?>
```

View hospitals

```
<?php
```

```
include
```

```
'connection.php'
```

```
; session_start();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<!-- employees23:21-->
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
```

```
<link rel="shortcut icon" type="image/x-icon" href="assets/img/favicon.ico">
```

```
<title>VIEW PATIENTS</title>
```

```
<link rel="stylesheet" type="text/css" href="assets/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="assets/css/font-awesome.min.css">
```

```
<link rel="stylesheet" type="text/css" href="assets/css/dataTables.bootstrap4.min.css">
```

```
<link rel="stylesheet" type="text/css" href="assets/css/select2.min.css">
```

```
<link rel="stylesheet" type="text/css" href="assets/css/bootstrap-  
datetimepicker.min.css">
```

```
<link rel="stylesheet" type="text/css" href="assets/css/style.css">
```

```
<!--[if lt IE 9]>
```

```
<script src="assets/js/html5shiv.min.js"></script>
```

```
<script src="assets/js/respond.min.js"></script>
```

```
<![endif]-->
```

```
</head>
```

```
<body>
```

```
<div class="main-wrapper">
```

```
<?php include 'include/header.php'?>
```

```
<?php include 'include/sidebar.php'?>
```

```
<div class="content">
  <div class="row">
    <div class="col-sm-4 col-3">
      <h4 class="page-title">Hospital Details</h4>
    </div>
  </div>
  <div class="row">
    <div class="col-md-12">
      <div class="table-responsive">
        <table class="table table-striped custom-table">
          <thead>
            <tr>
              <th style="min-width:200px;">Hospital Name</th>
              <th>Address</th>
              <th>Contact</th>
              <th class="text-center">Action</th>
            </tr>
          </thead>
          <tbody>
            <?php
              $query=mysqli_query($conn,"SELECT * FROM hospital_tb");
              while ($row=mysqli_fetch_assoc($query)) {
                ?>
                <h2><?php echo $row['hospital_name'];?></h2>
                </td>
                <td><?php echo $row['address'];?></td>
                <td><?php echo $row['contact_no'];?></td>
                <td class="text-center">
                  <!-- <div class="dropdown dropdown-action">
                    <a href="#" class="action-icon dropdown-toggle" data-
toggle="dropdown" aria-expanded="false"><i class="fa fa-ellipsis-v"></i></a>
                    <div class="dropdown-menu dropdown-menu-right">
                      <a class="dropdown-
item" href="edit_patient.php?edit_id=<?php// echo $row['role_id'];?>"><i class="fa fa-pencil m-r- 5"></i>
Edit</a>
```

```

item" href="edit_patient.php?edit_id=?php// echo $row['role_id'];?>"><i class="fa fa-pencil m-r-5"></i>Covid-Test</a>
</div>
</div> -->
<div class="dropdown action-label">
<a class="custom-badge status-purple dropdown-toggle" data- toggle="dropdown" aria-expanded="false">
ACTION
</a>
<div class="dropdown-menu dropdown-menu-right">
<a class="dropdown-
item" href="edit_hospital.php?edit_id=?php echo $row['hospital_id'];?>">Edit</a>
<a class="dropdown-
item" href="delete_hospital.php?delete_id=?php echo $row['hospital_id'];?>">Delete</a>

</div>
</div>
</td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
<div id="delete_employee" class="modal fade delete-modal" role="dialog">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-body text-center">

<h3>Are you sure want to delete this Employee?</h3>
<div class="m-t-20"> <a href="#" class="btn btn-white" data-dismiss="modal">Close</a>
<a type="submit" href="delete.php?delete_id=?php echo $row['collector_id'];?>" class="b tn btn-
danger">Delete</a>
</div>
</div>
</div>
</div>
</div>

```

```
</div>
<div class="sidebar-overlay" data-reff=""></div>
<script src="assets/js/jquery-3.2.1.min.js"></script>
<script src="assets/js/popper.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script src="assets/js/jquery.dataTables.min.js"></script>
<script src="assets/js/dataTables.bootstrap4.min.js"></script>
<script src="assets/js/jquery.slimscroll.js"></script>
<script src="assets/js/select2.min.js"></script>
<script src="assets/js/moment.min.js"></script>
<script src="assets/js/bootstrap-datetimepicker.min.js"></script>
<script src="assets/js/app.js"></script></body>
<!-- employees23:22-->
</html>
```

ADD DEPARTMENTS

```
<?php
include'connection.ph
p'; session_start();
$login=$_SESSION['login_id'];
$query_hospital=mysqli_query($conn,"SELECT * FROM hospital_tb WHERE login_id='$login'");
$row_data=mysqli_fetch_assoc($query_hospital);
$hospital_id=$row_data['hospital_id'];

if(isset($_POST['sub']))
{
    $d_name=$_POST['dep'];

    mysqli_query($conn,"INSERT INTO department_tb(hospital_id,department_name)
VALUES('$hospital_id','$d_name') ");
    echo "<script> alert(' New department added'); </script>";
    echo "<script> window.location.href='add_department.php';</script>";
}
?>
```


HOSPITALS**ADD DOCTORS**

```
<?php
include'connection.ph
p'; session_start();
$login=$_SESSION['login_id'];
$query_hospital=mysqli_query($conn,"SELECT * FROM hospital_tb WHERE login_id='$login'");
$row_data=mysqli_fetch_assoc($query_hospital);
$hospital_id=$row_data['hospital_id'];

if(isset($_POST['sub']))
{
    $d_name=$_POST['doc_name'];
    $quali=$_POST['qualific'];
    $dep_id=$_POST['department'];
    $addr=$_POST['address'];
    mysqli_query($conn,"INSERT INTO doctor_tb
    (hospital_id,department_id,doc_name,qualification,place
) VALUES ('$hospital_id','$dep_id','$d_name','$quali','$addr')");

    echo "<script> alert('Doctor Registration completed');
    </script>"; echo "<script>
    window.location.href='add_doctor.php';</script>";
}

?>

<?php
include'connection.ph
p'; session_start();
$login=$_SESSION['login_id'];
$query_hospital=mysqli_query($conn,"SELECT * FROM hospital_tb WHERE login_id='$login'");
$row_data=mysqli_fetch_assoc($query_hospital);
$hospital_id=$row_data['hospital_id'];
if(isset($_POST['sub']))
    $d_name=$_POST['doc_name'];
    $start_time=$_POST['start'];
    $end_time=$_POST['endtime'];
    $status=$_POST['status'];
```

```
mysql_query($conn,"INSERT INTO
doctor_schedule_tb(doctor_id,hospital_id,start_time,end_time,status) VALUES
('$d_name','$hospital_id','$start_time','$end_time','$status')");

echo "<script> alert('Doctor Schedule Added'); </script>";

echo "<script> window.location.href='schedule.php';</script>";

}

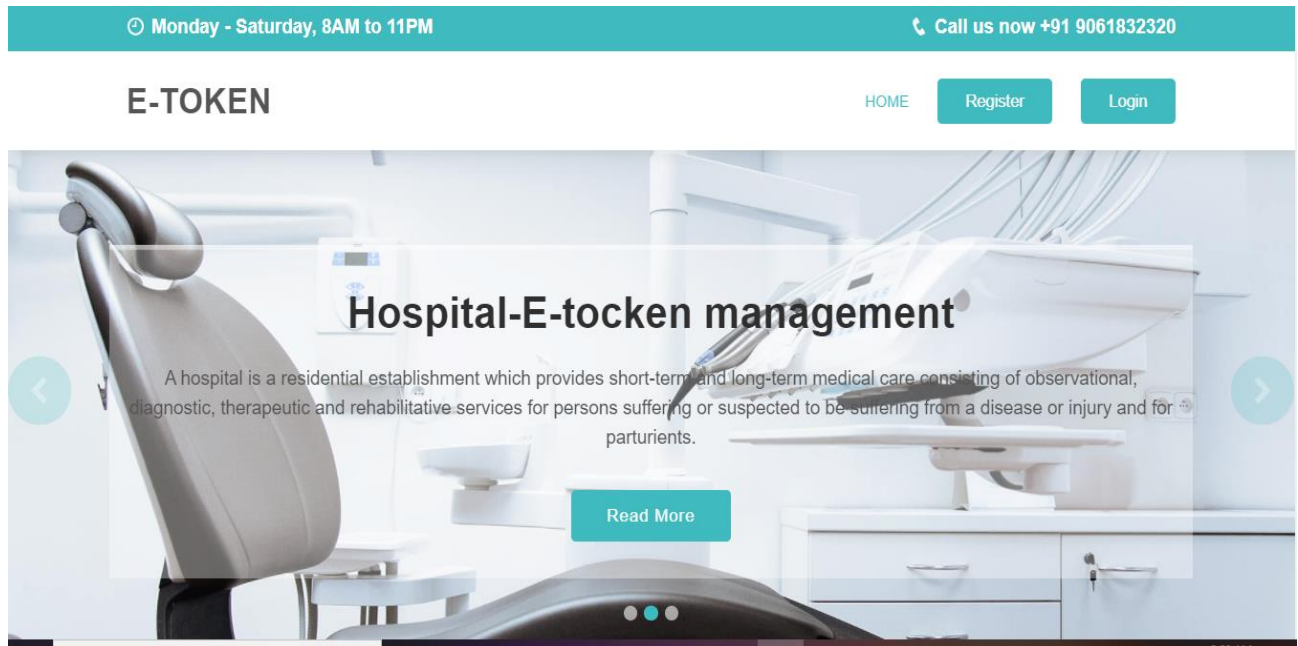
?>
```

Patients

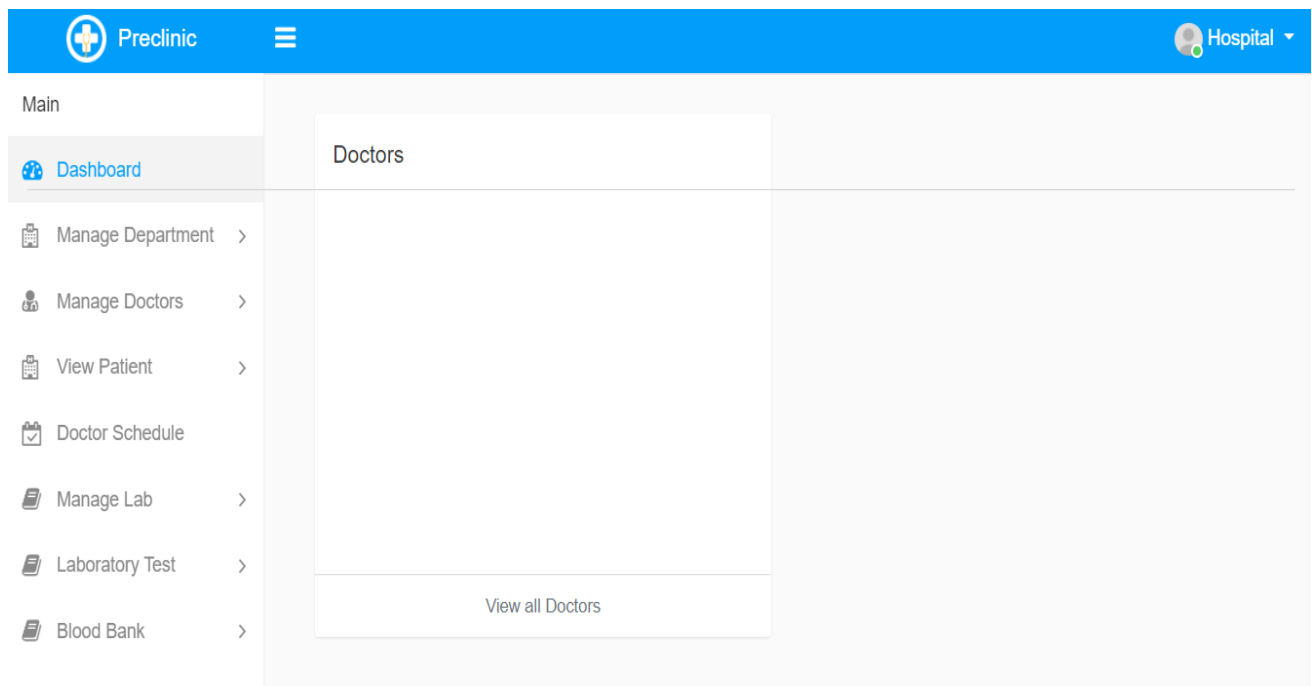
```
<?php
include'loginpanel/connection.
php'; session_start();
$login=$_SESSION['login_id'];
$hospital_id=$_GET['hospital_id'];
$req_query=mysql_query($conn,"SELECT * FROM registration_tb WHERE login_id='$login'");
$reg_data=mysql_fetch_assoc($req_query);
$reg_id=$reg_data['reg_id'];
$check=mysql_query($conn,"SELECT * FROM blood_donation_tb WHERE
reg_id='$reg_id'"); if($row=mysql_num_rows($check))
{
echo "<script>alert('We Remember You!');</script>";
echo "<script>window.location.href='user-index.php?hospital_id=$hospital_id'</script>";}
else{
mysql_query($conn,"INSERT INTO blood_donation_tb (reg_id,hospital_id) VALUES
('$reg_id','$hospital_id') ");
echo "<script>alert('We Will Contact You!');</script>";
echo "<script>window.location.href='user-index.php?hospital_id=$hospital_id'</script>";
```

2.12.2 SCREENSHOTS


○ HOME PAGE





○ HOSPITAL HOME PAGE



○ ADD DOCTORS SCHEDULE PAGE

 Preclinic



 Hospital ▾

Main

Dashboard

Manage Department >

Manage Doctors >

View Patient >

Doctor Schedule

Manage Lab >

Laboratory Test >

Blood Bank >

Add Schedule

Doctor Name

Select Doctor ▾

Start Time


End Time


Schedule Status


☒ Active ☐ Inactive

CREATE SCHEDULE

○ HOSPITAL VIEW DOCTORS PAGE

 Preclinic



 Hospital ▾

Main

Dashboard

Manage Department >

Manage Doctors >

View Patient >

Doctor Schedule

Manage Lab >



Laboratory Test >

Blood Bank >

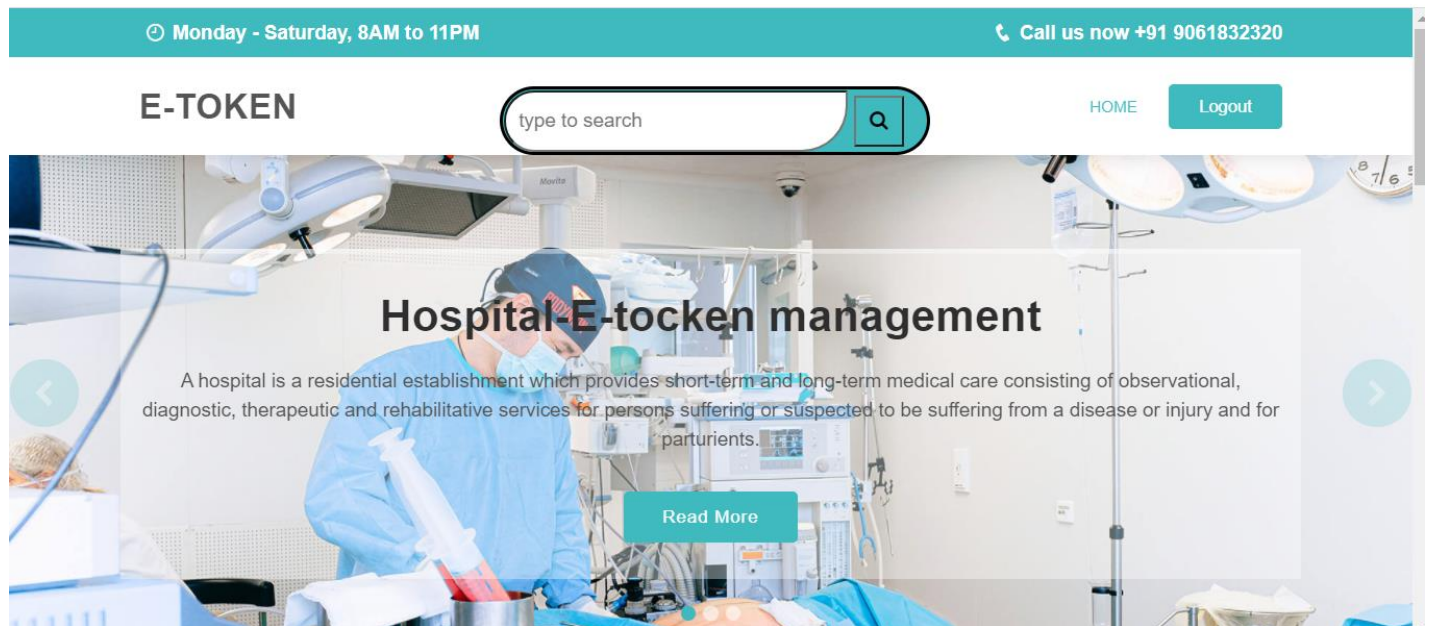
Schedule

+ Add Schedule

+ Close Schedule

| Doctor Name | Available Time | Status | Action |
|---|-------------------|--------|---|
|  NIVYA | 3:57 AM - 3:57 AM | Active |  |

○ PATIENTS HOME PAGE



○ APPOINTMENT PAGE

The screenshot shows the 'APPOINTMENT PAGE' of the 'HOSPITAL E-TOKEN MANAGEMENT SYSTEM'. The 'E-TOKEN' logo is on the left, and a navigation menu with links 'HOME', 'DONATE BLOOD', 'ABOUT', 'DEPARTMENTS', 'DOCTORS', 'LABORATORY', 'BOOKEDAPPOINTMENT', and 'LOGOUT' is on the right. The main heading is 'MAKE AN APPOINTMENT'. Below it is a paragraph of placeholder text: 'Magnum dolores commodi suscipit. Necessitatibus eius consequatur ex aliquid fuga eum quidem. Sit sint consectetur velit. Quisquam quos quisquam cupiditate. Et nemo qui impedit suscipit alias ea. Quia fugiat sit in iste officiis commodi quidem hic quas.' There are two dropdown menus: the first is set to 'psychology' and the second is labeled 'Select Doctor'. A 'Submit' button is located below the dropdowns.

- PAYMENT DETAILS PAGE

E-TOKEN

CARD NUMBER

Valid Card Number

EXPIRY DATE

YYYY

CODE

CV

80 Final Payment

pay