MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

HEALTH CARE APPLIC ATION

Submitted By:

Saurabh Rajput

0901CS1911109

Manisha Pal

0901CS203D04

Faculty Mentor:

Ms . Khushboo Agrawal,

Assistant Professor , Computer science and engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY - JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

HEALTH CARE APPLICATION

A project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Saurabh Rajput

0901CS191109

Manisha Pal

0901CS203D04

Faculty Mentor:

Ms . Khushboo Agrawal ,

Assistant Professor , Computer science and engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957
MAY - JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that Manisha Pal (0901CS203D04) has submitted the project report title healthcare application under the mentorship of Prof . Khushboo Agrawal , in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

09/05/2013

Khushboo Agrawal
Faculty Mentor
Assistant Professor
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that Manisha Pal (0901CS203D04) has submitted the project report title healthcare application under the mentorship of Prof . Khushboo Agrawal , in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

09/05/2013

Khushboo Agrawal
Faculty Mentor
Assistant Professor
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship Ms. Khushboo Agrawal , Assistant Professor, Computer Science and Engineering.

We declare that We have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Saurabh Rajput
0901CS191109
III Year
Computer Science and Engineering

Manisha Pal
0901CS203D04
III Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, Madhav Institute of Technology and Science to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, Dr. R. K. Pandit and Dean Academics, Dr. Manjaree Pandit for this.

I would sincerely like to thank my department, Department of Computer Science and Engineering, for allowing me to explore this project. I humbly thank Dr. Manish Dixit, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of Ms. Khushboo Agrawal , Assistant Professor, Computer Science and Engineering. for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Saurabh Rajput
0901CS191109
III Year
Computer Science and Engineering

Manisha Pal
0901CS203D04
III Year
Computer Science and Engineering

ABSTRACT

The purpose of the project entitled as “HELATHCARE system” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

सार

"हेलैथकेयर सिस्टम" नामक परियोजना का उद्देश्य अस्पताल के फ्रंट ऑफिस प्रबंधन को कम्प्यूटरीकृत करना है ताकि सॉफ्टवेयर विकसित किया जा सके जो उपयोगकर्ता के अनुकूल सरल, तेज और लागत प्रभावी हो। यह रोगी की जानकारी, निदान विवरण आदि के संग्रह से संबंधित है। परंपरागत रूप से, यह मैन्युअल रूप से किया जाता था। सिस्टम का मुख्य कार्य रोगी के विवरण और डॉक्टर के विवरण को पंजीकृत करना और स्टोर करना है और इन विवरणों को आवश्यकता पड़ने पर पुनःप्राप्त करना है, और इन विवरणों को सार्थक रूप से हेरफेर करना भी है सिस्टम इनपुट में रोगी विवरण, निदान विवरण शामिल हैं, जबकि सिस्टम आउटपुट इन विवरणों को प्राप्त करना है। पर्दा डालना। उपयोगकर्ता नाम और पासवर्ड का उपयोग करके अस्पताल प्रबंधन प्रणाली में प्रवेश किया जा सकता है। यह या तो एक व्यवस्थापक या रिसेप्शनिस्ट द्वारा पहुँचा जा सकता है। केवल वे ही डेटाबेस में डेटा जोड़ सकते हैं। डेटा आसानी से पुनर्प्राप्त किया जा सकता है। डेटा व्यक्तिगत उपयोग के लिए अच्छी तरह से सुरक्षित है और डेटा प्रोसेसिंग को बहुत तेज़ बनाता है।

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: PROJECT OVERVIEW

1.1 Introduction

Digital health, or digital healthcare, is a broad, multidisciplinary concept that includes concepts from an intersection between technology and healthcare.

Digital Health Applies Digital Transformation To The Healthcare Field,Incorporating

Software, Hardware And Services.

With Digital Healthcare Systems, Along With Other Technologies, patient information now easily collected, collated, verified, and utilized for research and development of new drugs and treatments to further improve human health and well-being.

1.2 Scope

Digital tools are giving providers a more holistic view of patient health through access to data and giving patients more control over their health. Digital health offers real opportunities to improve medical outcomes and enhance efficiency

These technologies can empower consumers to make better-informed decisions about their own health and provide new options for facilitating prevention, early diagnosis of life-threatening diseases, and management of chronic conditions outside of traditional health care settings. Providers and other stakeholders are using digital health technologies in their efforts to:

- Reduce inefficiencies,
- Improve access,
- Reduce costs,
- Increase quality, and.

1.3 Project Features

- Easier access to healthcare data. ...
- Focus on quality care. ...
- Improved patient satisfaction. ...
- Cost savings. ...
- Better patient care. ...
- Increased efficiency. ...

- No prescription errors. ...
- Streamlined prescriptions and more time for patient care.

1.4 Feasibility study

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

1.4.1 Operational feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

1.4.2 Technical feasibility

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

1.4.3 Economical feasibility

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Hardware Requirements

Number	Description
1	PC with 250 GB or more Hard disk.
2	PC with 4 GB RAM.
3	PC with Pentium 1 and Above.

2.2 Software Requirements

Number	Description	Type
1	Operating System	Windows XP / Windows 7/8/10
2	Language	Html,css,javascript,
3	Database	mongodb
4	IDE	Visual Code
5	Browser	Google Chrome

CHAPTER 3 : DESIGN & PLANNING

3.1 Software Development Life Cycle Model

3.1.1 WATERFALL MODEL

The waterfall model was selected as the SDLC model due to the following reasons:

Requirements were very well documented, clear and fixed.

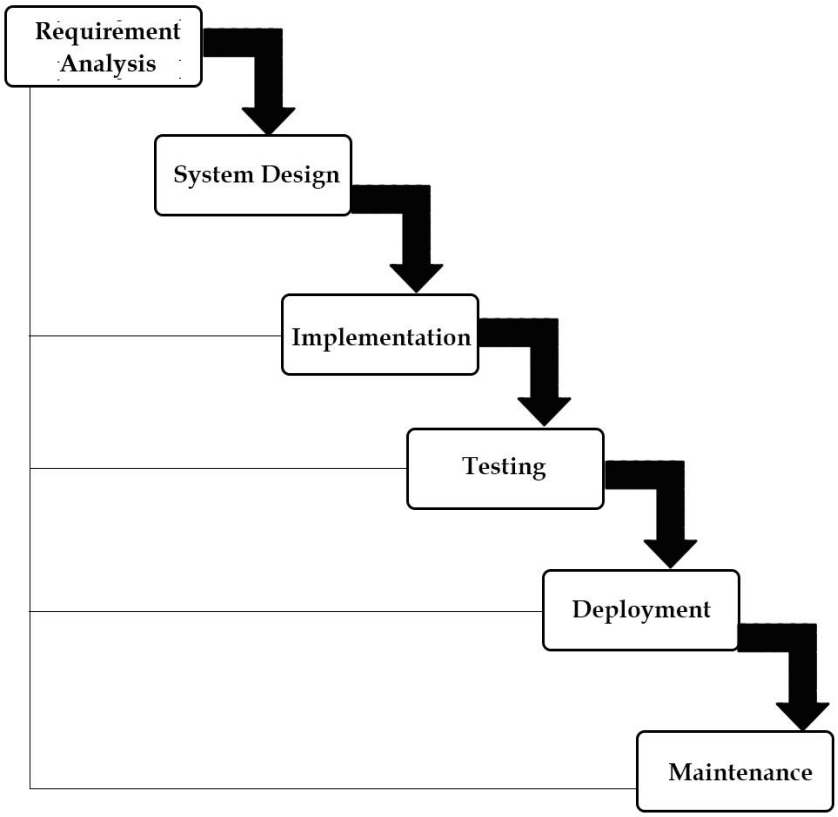
Technology was adequately understood.

Simple and easy to understand and use.

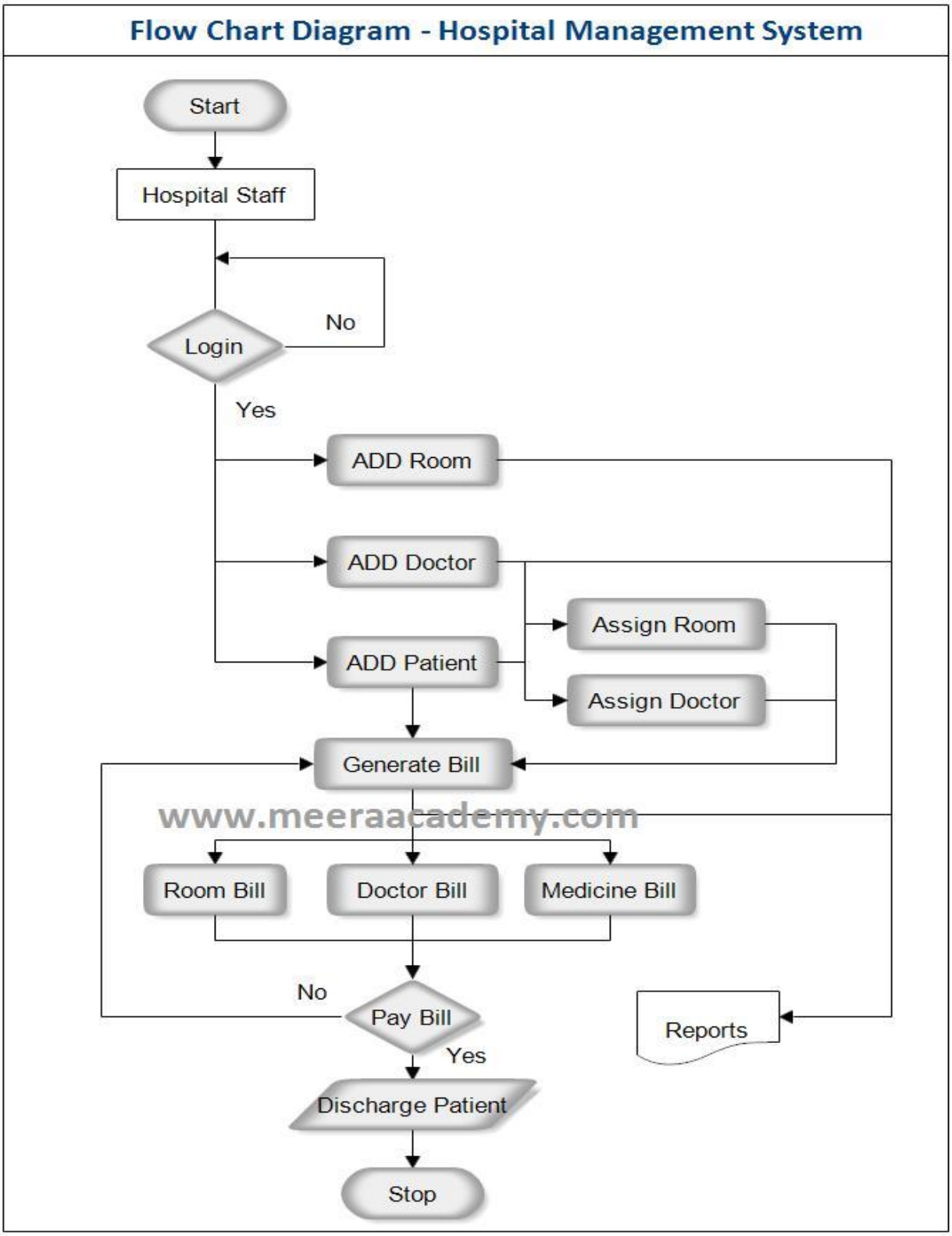
There were no ambiguous requirements.

Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

Clearly defined stages.

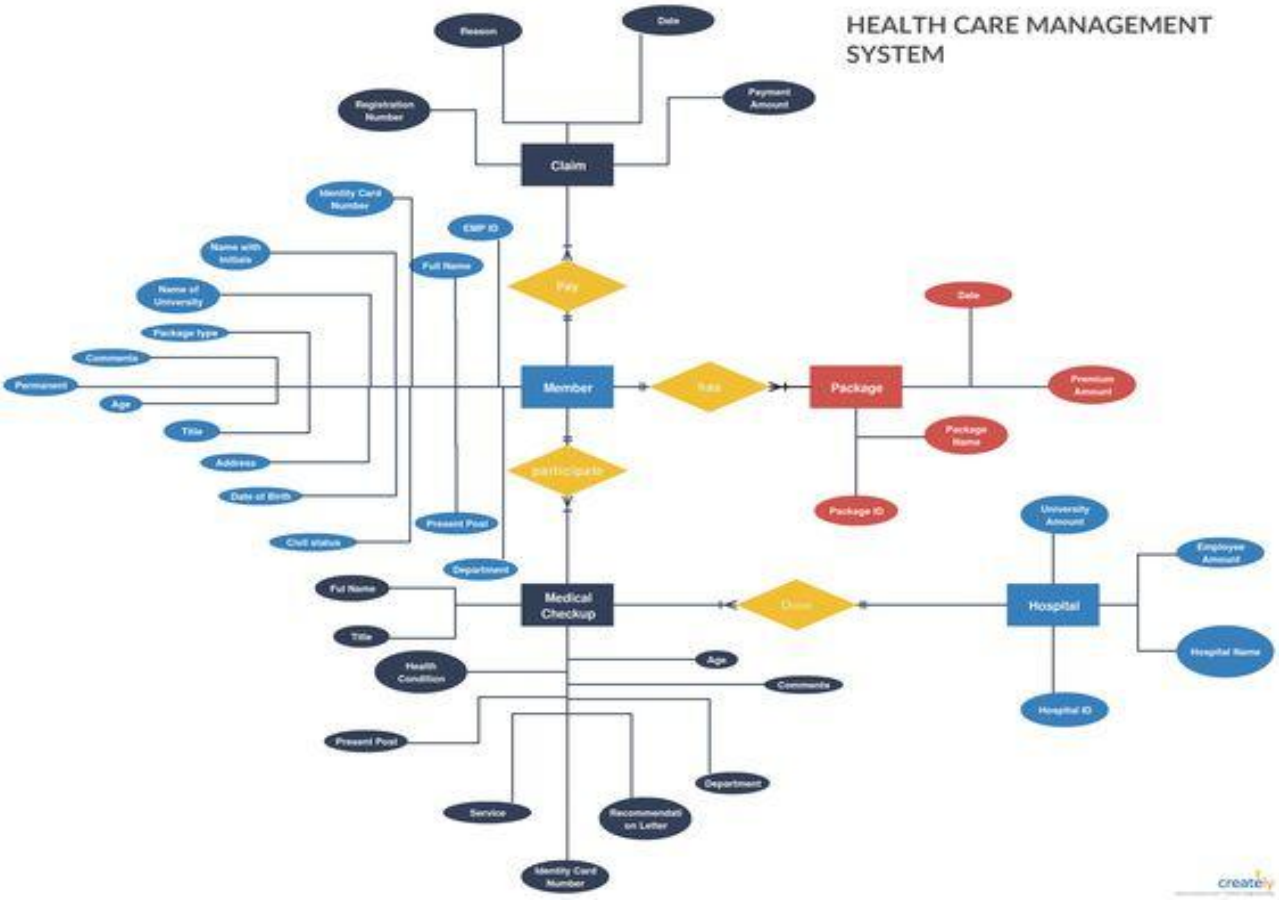


3.2 Flow Chart

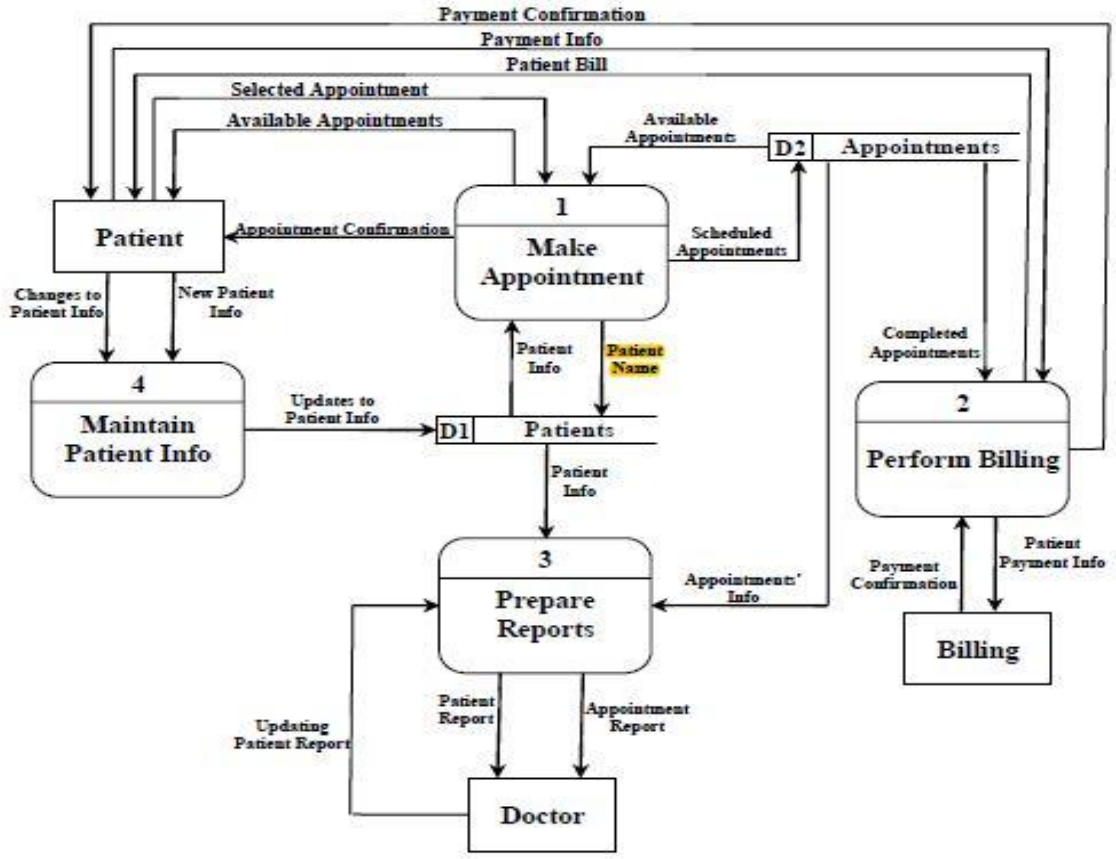


Health Care Flow Chart ()

3.3 ER Diagram:-



3.4 DFD Diagram:-



CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

4.1 : Front End

4.1.1 Html

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

4.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

4.1.3 Javascript

JavaScript is commonly used for creating web pages. It allows us to add dynamic behavior

to the webpage and add special effects to the webpage. On websites, it is mainly used

for validation purposes. JavaScript helps us to execute complex actions and also enables

the interaction of websites with visitors.

4.1.4 Visual studio code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.

Features include support for debugging, syntax highlighting, intelligent code completion, snippets,

code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences,

and install extensions that add additional functionality.

4.1.5 Bootstrap

Fewer Cross browser bugs. A consistent framework that supports major of al

l browsers and CSS compatibility fixes. Lightweight and customizable.

Responsive structures and styles. That's why I was used bootstrap.it makes my webpage responsive and dynamic.

4.2 Backend

4.2.1 Express.js

It is used for designing and building web applications quickly and easily.

Web applications are web apps that you can run on a web browser.

Since Express. js only requires javascript, it becomes easier for programmers

And developers to build web applications and api without any effort.

The express framework is built on top of the node. Js framework and helps in fast-tracking development of server-based applications. Routes are used to divert users to different parts of the web applications based on the request made

Express.js is a Node js web application server framework, which is specifically designed for building single-page, multi-page, and hybrid web applications.

It has become the standard server framework for node.js. Express is the backend part of something known as the MEAN stack.

The MEAN is a free and open-source JavaScript software stack for building dynamic web sites and web applications which has the following components;

1) MongoDB – The standard NoSQL database

2) Express.js – The default web applications framework

4.2.2 Mongo db

Node.js is popularly being used in web applications because it lets the application run while it is fetching data from the backend server. It is asynchronous, event-driven and helps to build scalable web applications. Even though Node.js works well with MySQL database, the perfect combination is a NoSQL like MongoDB

wherein the schema need not be well-structured. MongoDB represents the data as a collection of documents rather than tables related by foreign keys. This makes it possible for the varied types of data dealt over the internet to be stored decently and accessed in the web applications using Node.js. Another option is using MongoDB that also stores the data as JSON/BSON environment

4.2.3 Mongoose

Mongoose is a Node.js-based Object Data Modeling (ODM) library for MongoDB. It is akin to an Object Relational Mapper (ORM) such as SQLAlchemy for traditional SQL databases. The problem that Mongoose aims to solve is allowing developers to enforce a specific schema at the application layer. In addition to enforcing a schema, Mongoose also offers a variety of hooks, model validation, and other features aimed at making it easier to work with MongoDB.

CHAPTER 5 : TESTING AND IMPLEMENTATION

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

5.1 : Unit testing

5.1.1 Introduction

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

5.1.2 Benifits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

5.2 : Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

5.2.1 Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to

support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

5.3 : Software Verification & Validation

5.3.1 Introduction

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements.This is done through dynamic testing and other forms of review.Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

Validation : Are we building the right product?

Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

5.3.2 Classification of methods

In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission-critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

5.3.3 Test cases

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

5.4 : System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

CHAPTER 6: ADVANTAGES

Gives patients faster access to providers and care. ...

Improves medication adherence. ...

Makes remote patient monitoring possible and easy. ...

Increases medication reconciliation accuracy, which improves patient safety. ...

Improves provider communication and coordination.

CHAPTER 7 : CONCLUSION

Taking into account all the mentioned details, we can make the conclusion that the helathcare system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users. Developing the hospital system software is a great opportunity to create the distinct, efficient and fast delivering healthcare model. Implementation of hospital management system project helps to store all the kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services. This beneficial decision covers the needs of the patients, staff and hospital authorities and simplifies their interactions. It has become the usual approach to manage the hospital. Many clinics have already experienced its advantages and continue developing new hospital management system project modules.