

|                                  |  |
|----------------------------------|--|
| <b>Project Title</b>             | <b>E-Health Care Management System</b>   |
| <b>Project Definition / Goal</b> | E-Health Care Management System is a website That designed For the appointment booking process, allowing patients to Book schedule appointments based on the available sessions. |
| <b>Project Guide</b>             | <b>Mr. Kevin Parekh</b>  |
| <b>Operating System</b>          | Windows 7 and Upgrade  |
| <b>Web Server</b>                | Apache HTTP XAMPP  |
| <b>Frontend Used</b>             | HTML,CSS,JAVASCRIPT,BOOTSTRAP  |
| <b>Backend Uses</b>              | MySQL  |
| <b>Connectivity</b>              | PHP  |
| <b>Project Duration</b>          | 100 Days   |
| <b>Team Size</b>                 | 2 Person   |
| <b>Submitted By</b>              | <b>Mr. Kartik Patel [25260110]</b><br><b>Mr. Rudra Patel [25260229]</b>  |
| <b>Submitted To</b>              | Shree Swaminarayan College of Computer Science. Gurukul campus, Sardarnagar, Bhavnagar. Gujrat-364001.   |

# **PREFACE**

It has been stated, each long journey starts with a little step now a day. The world becomes complex and going towards direction of computerization. So,

**M.K. Bhavnagar University** has included project and industrial study as a part of **Bachelor of Computer Application (BCA)** course.

In today's dynamic and rapidly evolving world, the role of technology is more crucial than ever before. It serves as the backbone of modern enterprises, enabling them to adapt, thrive, and stay ahead of the curve.

**E-Health Care Management System** emerges as a testament to this belief, embodying the spirit of progress and transformation.

As we embark on this journey, it's essential to acknowledge the collaborative spirit that underpins this project. From the initial conceptualization to the final implementation, numerous individuals have contributed their expertise, insights, and passion to bring this vision to life.

This preface sets the stage for what lies ahead in this document. It provides an overview of the project's objectives, scope, methodologies, and anticipated outcomes.

Additionally, it offers insights into the broader context within which **E-Health Care Management System** operates, highlighting the challenges, opportunities, and significance of leveraging technology in Healthcare industry.

Together, let's embark on this exciting journey and unlock the boundless potential that **E-E-Health Care Management System** holds for the future of **Healthcare industry**.

Thank you

## **ACKNOWLEDGEMENT**

On the successful completion of my work I would like to express my sincere thanks to all those who guided, advised, inspired and supported me during my project work at **Shree Swaminarayan College of Computer Science, Bhavnagar**.

It gives me immense pleasure to acknowledge the contribution and assistance of individuals to this project. I would like to express my gratitude to those who have made major contributions to this project.

First and foremost, we extend our heartfelt appreciation to **Mr. Kevin Parekh (Asst. Professor, Shree Swaminarayan College of Computer Science)**, for their unwavering support and belief in the vision of **E-Health Care Management System**. Their guidance, encouragement, and commitment have been integral to steering this project towards fruition.

I would like to acknowledge the **Dr. Kalpesh U. Gundigara (Academic Head, Shree Swaminarayan College of Computer Science)** has been a true visionary, who has constantly inspired us for this report.

Lastly, we express our gratitude to all individuals who, directly or indirectly, have contributed to the realization of **E-Health Care Management System**. Your collective efforts have left an indelible mark on this journey, and for that, we are profoundly thankful.

Together, your support, dedication, and collaboration have transformed the vision of this system into a tangible reality. As we look towards the future, we remain inspired by the spirit of innovation and partnership that has defined this project.

## **ABSTRACT**

**E-Health Care Management System** is a comprehensive software solution designed to streamline and optimize the process of scheduling and managing appointments between patients and healthcare providers. In today's digital age, efficient appointment management is crucial for enhancing the patient experience, optimizing resource utilization, and improving overall healthcare delivery.

Furthermore **E-Health Care Management System** integrates robust security measures to safeguard patient confidentiality and comply with healthcare regulations, ensuring the privacy and integrity of sensitive medical information. By digitizing and automating the appointment management process, it also reduces administrative burdens, minimizes errors, and enhances the overall efficiency of healthcare practices.

In conclusion, **E-Health Care Management System** offers a comprehensive solution to the challenges associated with appointment scheduling and management in healthcare settings. By leveraging technology to streamline processes and improve communication between patients and healthcare providers, it aims to enhance the quality of patient care, optimize resource utilization, and ultimately contribute to a more efficient and patient-centered healthcare delivery system.

## LIST OF TABLES

| SR.NO | Table Name  |
|-------|-------------|
|       |             |
| 1     | admin       |
| 2     | appointment |
| 3     | article     |
| 4     | comment     |
| 5     | doctor      |
| 6     | feedback    |
| 7     | patient     |
| 8     | pending     |
| 9     | rating      |
| 10    | schedule    |
| 11    | specialist  |
| 12    | transaction |
| 13    | user        |
|       |             |

## **INDEX**

| SR.NO | Description                     |                                    |                       | Page No                    |
|-------|---------------------------------|------------------------------------|-----------------------|----------------------------|
|       |                                 |                                    |                       |                            |
| 1     | Introduction                    |                                    |                       | 1-3                        |
|       | 1.1                             | Background                         |                       | 2                          |
|       | 1.2                             | Objective                          |                       | 2                          |
|       | 1.3                             | Scope                              |                       | 3                          |
| 2     | Software Development Life Cycle |                                    |                       | 4-9                        |
|       | 2.1                             | What Is SDLC?                      |                       | 5                          |
|       | 2.2                             | Advantage Of SDLC?                 |                       | 6                          |
|       | 2.3                             | Phases Of SDLC?                    |                       | 8                          |
| 3     | Requirement And Analysis        |                                    |                       | 10-28                      |
|       | 3.1                             | Problem Definition                 |                       | 11                         |
|       | 3.2                             | Requirement Specification          |                       | 11                         |
|       | 3.3                             | Software and Hardware Requirements |                       | 12                         |
|       |                                 | 3.3.1                              | Software Requirements | 12                         |
|       |                                 | 3.3.2                              | Hardware Requirements | 12                         |
|       | 3.4                             | Planning and Scheduling            |                       | 13                         |
|       | 3.5                             | Preliminary Production Description |                       | 14                         |
|       | 3.6                             | System Analysis                    |                       | 16-28                      |
|       |                                 | 3.6.1                              | Overview of System    | 16                         |
|       |                                 | 3.6.2                              | Feasibility Study     | 21                         |
|       |                                 | 3.6.3                              | Tools                 | 23-28                      |
|       |                                 |                                    | 3.6.3.1               | Introduction to <b>PHP</b> |

|          |                      |                   |                                   |                                   |              |
|----------|----------------------|-------------------|-----------------------------------|-----------------------------------|--------------|
|          |                      |                   | 3.6.3.2                           | Introduction to <b>JAVASCRIPT</b> | 24           |
|          |                      |                   | 3.6.3.3                           | Introduction to <b>MYSQL</b>      | 25           |
|          |                      |                   | 3.6.3.4                           | Introduction to <b>HTML</b>       | 26           |
|          |                      |                   | 3.6.3.5                           | Introduction to <b>CSS</b>        | 27           |
|          |                      |                   | 3.6.3.6                           | Introduction to <b>BOOTSTRAP</b>  | 28           |
| <b>4</b> | <b>System Design</b> |                   |                                   |                                   | <b>29-93</b> |
|          | 4.1                  | Database Design   |                                   |                                   | 29-46        |
|          |                      | 4.1.1             | Introduction                      |                                   | 30           |
|          |                      | 4.1.2             | Types of Data Dictionary          |                                   | 31           |
|          |                      | 4.1.3             | Why data dictionary is important? |                                   | 32           |
|          |                      | 4.1.4             | Data Dictionary                   |                                   | 33           |
|          | 4.2                  | Data Flow Diagram |                                   |                                   | 47-64        |
|          |                      | 4.2.1             | What is Data Flow Diagram         |                                   | 47           |
|          |                      | 4.2.2             | Component of Data Flow Diagram    |                                   | 48           |
|          |                      | 4.2.3             | Rules of Data Flow Diagram        |                                   | 51           |
|          |                      | 4.2.4             | Advantage of Data Flow Diagram    |                                   | 51           |
|          |                      | 4.2.5             | Data Flow Diagram                 |                                   | 54           |
|          | 4.3                  | E-R Diagram       |                                   |                                   | 65-69        |
|          |                      | 4.3.1             | Introduction                      |                                   | 65           |
|          |                      | 4.3.2             | Why use ER diagram?               |                                   | 65           |
|          |                      | 4.3.3             | Component of ER diagram           |                                   | 66           |
|          |                      | 4.3.4             | ER Diagram                        |                                   | 69           |
|          | 4.4                  | UML Diagrams      |                                   |                                   | 70-86        |
|          |                      | 4.4.1             | Introduction                      |                                   | 70           |
|          |                      | 4.4.2             | Characteristic                    |                                   | 71           |
|          |                      | 4.4.3             | Types of UML                      |                                   | 72-84        |

|   |                          |                         |         |                  |         |
|---|--------------------------|-------------------------|---------|------------------|---------|
|   |                          |                         | 4.4.3.1 | Use Case Diagram | 73      |
|   |                          |                         | 4.4.3.2 | Activity Diagram | 80      |
|   | 4.5                      | Input /Output Design    |         |                  | 88-94   |
| 5 | Testing & Implementation |                         |         |                  | 95-119  |
|   | 5.1                      | Introduction of Testing |         |                  | 96      |
|   | 5.2                      | Importance of Testing   |         |                  | 97      |
|   | 5.3                      | Benefits of Testing     |         |                  | 97      |
|   | 5.4                      | Testing Plan            |         |                  | 98      |
|   | 5.5                      | Testing Strategies      |         |                  | 99      |
|   | 5.6                      | Testing Methods         |         |                  | 103     |
|   | 5.7                      | Test Cases              |         |                  | 109     |
|   | 5.8                      | Implementation Approach |         |                  | 120     |
| 6 | Conclusion               |                         |         |                  | 121-124 |
|   | 6.1                      | Conclusion              |         |                  | 122     |
|   | 6.2                      | Limitation of System    |         |                  | 122     |
|   | 6.3                      | Future Scope of System  |         |                  | 123     |
|   | 6.4                      | Bibliography            |         |                  | 124     |
|   |                          |                         |         |                  |         |



## **LIST OF FIGURES**

| <b>SR.NO</b> | <b>Figure</b> | <b>Title</b>                           | <b>Page No.</b> |
|--------------|---------------|--|-----------------|
| 1            | 2.1           | Software Development Life Cycle (SDLC) | 5               |
| 2            | 3.4           | Planning and Scheduling                | 13              |
| 3            | 3.6.2         | Feasibility study                      | 21              |
| 4            | 3.6.3.1       | PHP                                    | 23              |
| 5            | 3.6.3.2       | JAVASCRIPT                             | 24              |
| 6            | 3.6.3.3       | MYSQL                                  | 25              |
| 7            | 3.6.3.4       | HTML                                   | 26              |
| 8            | 3.6.3.5       | CSS                                    | 27              |
| 9            | 3.6.3.6       | BOOTSTRAP                              | 28              |
| 10           | 4.2.5.1       | Level – 0 DFD                          | 54              |
| 11           | 4.2.5.2       | Admin 1 <sup>st</sup> Level DFD        | 55              |
| 12           | 4.2.5.3       | Admin 2 <sup>nd</sup> Level DFD        | 56              |
| 13           | 4.2.5.4       | Doctor 1 <sup>st</sup> Level DFD       | 59              |
| 14           | 4.2.5.5       | Doctor 2 <sup>nd</sup> Level DFD       | 60              |
| 15           | 4.2.5.6       | Patient 1 <sup>st</sup> Level DFD      | 62              |
| 16           | 4.2.5.7       | Patient 2 <sup>nd</sup> Level DFD      | 63              |
| 17           | 4.3.3         | Component of ER Diagram                | 66              |
| 18           | 4.3.4         | ER Diagram                             | 69              |
| 19           | 4.4.3         | Types of UML                           | 72              |
| 20           | 4.4.3.1.1     | Admin Use-case Diagram                 | 77              |
| 21           | 4.4.3.1.2     | Doctor Use-case Diagram                | 78              |
| 22           | 4.4.3.1.3     | Patient Use-case Diagram               | 79              |
| 23           | 4.4.3.2.1     | Admin Activity Diagram                 | 85              |
| 24           | 4.4.3.2.2     | Doctor Activity Diagram                | 86              |
| 25           | 4.4.3.2.3     | Patient Activity Diagram               | 87              |
| 26           | 5.5           | Testing Strategies                     | 99              |
| 27           | 5.6.1         | Black-Box Testing                      | 103             |

| SR.NO | Figure | Title             | Page No. |
|-------|--------|-------------------|----------|
|       |        |                   |          |
| 28    | 5.6.2  | White-Box Testing | 105      |
| 29    | 5.6.3  | Grey-Box Testing  | 107      |
|       |        |                   |          |

# **INTRODUCTION**

Healthcare, an essential pillar of societal well-being, stands at the intersection of human welfare and technological advancements. The integration of technology into healthcare practices has become imperative to address the evolving needs of patients, enhance the efficiency of healthcare providers, and ensure the secure management of critical health information. In response to these dynamic challenges, we propose the development and implementation of an E-Health Care Management System

### **1.1 Background:**

The healthcare industry is undergoing a transformative shift towards digital solutions to enhance patient care, improve efficiency, and streamline administrative processes. In line with this evolution, this proposal introduces an E-Health Care Management System that leverages technology to revolutionize healthcare delivery.

### **1.2 Objectives:**

- The primary objectives of the proposed E-Health Care Management System are:
- To digitize and centralize healthcare records for efficient and secure management.
- To facilitate seamless communication and collaboration among healthcare professionals.
- To enhance patient experience by providing easy access to healthcare services and information.
- To automate administrative tasks, reducing paperwork and improving operational efficiency.

To ensure compliance with healthcare regulations and prioritize data security and privacy.

### **1.3 Purpose and Scope:**

- The purpose of this project is to create an integrated E-Health Care Management System that caters to the needs of healthcare providers, staff, and patients. The scope of the system includes:
- Patient Information Management
- Electronic Health Records (EHR) Integration
- Appointment Scheduling and Management
- Prescription and Medication Tracking
- Billing and Insurance Claims Processing
- Communication and Collaboration Tools for Healthcare Professionals
- Patient Portal for Access to Health Information

The system aims to optimize healthcare processes, improve patient care, and provide a user-friendly platform for healthcare stakeholders.

## **SOFTWARE DEVELOPMENT LIFE CYCLE**

### **[SDLC]**

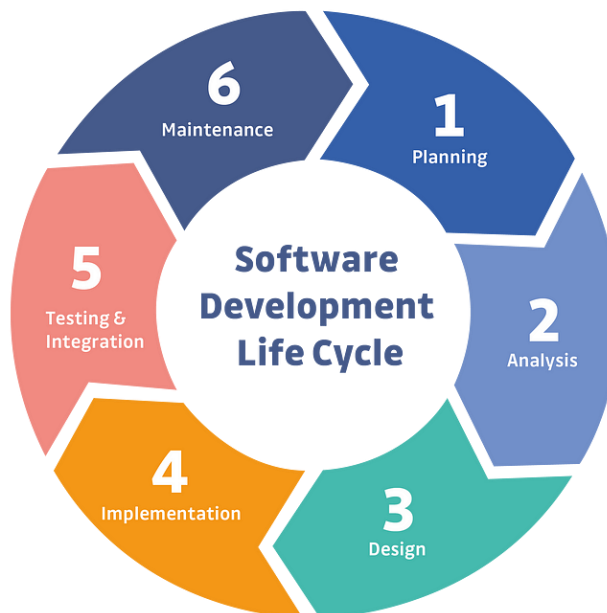
## 2.1 What is SDLC ?

“The **System Development Life Cycle (SDLC)** method is classically thought of as the set of activities that **analyst, designers, and users** carry out to develop and implement an information system.”

The approach for the development of the project is called the **Software Development Life Cycle (SDLC)**. The concepts and working of SDLC is a **standard, guaranteed** way to achieve deadlines and completion of milestones for **project preparation and completion**.

The software development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

SDLC encapsulates the sequential steps that give software solutions, to achieve the twin goals of effectiveness and high-quality output promptly.



[Figure-2.1 Software Development Life Cycle (SDLC)]

## **2.2 Advantages Of SDLC :**

- **Structured Approach:**

SDLC offers a structured approach to software development, efficient planning, and organization of tasks for developers. This structured methodology not only minimizes errors but also enhances productivity, ensuring the timely delivery of high-quality software.

- **Risk Management:**

A great aspect of SDLC is its ability to identify and effectively manage risks inherent in the software development process. By pinpointing potential risks early on, developers can proactively address and mitigate them, ultimately diminishing the overall risk associated with software development

- **Consistency:**

SDLC establishes a foundation for consistency in software development through a standardized framework and methodology. This consistency is instrumental in elevating the quality of the software, guaranteeing that the final product aligns seamlessly with client expectations.

- **Collaboration:**

SDLC fosters a collaborative environment among team members by providing a common application framework and language for communication.



# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

This collaborative synergy not only enhances the overall quality of the software but also ensures that the end product precisely fulfills the client's requirements.

- **Cost-Effective:**

SDLC proves to be a cost-effective approach by identifying potential issues early in the development process through the use of prototyping tools like Figma and others. Early issue detection allows developers to take proactive measures, significantly reducing overall development costs. This cost-effective attribute positions SDLC as a strategic choice in the realm of software development.

## 2.3 Phases of SDLC :

### 1. Planning

Planning is the phase where the team clarifies its cost and resource requirements. It's also the right time to define the processes that will be useful in meeting the objectives. Feedback from stakeholders, among other things, is taken into consideration during planning. It's useful to get feedback from customers, programmers or salespeople. Their opinions can significantly ease software development.

### 2. Analysis

The analysis part is the most important stage of the SDLC. It's the responsibility of the most senior members of the team and is done with whatever data they have. During the analysis, the strengths and weaknesses of the project are debated. Such activities are designed to assess the overall feasibility of the project.

### 3. Design

At this stage, the description of the expected features and operation of the system takes place. A project's design phase is the time for developing any components. Also, communication with services from third parties is improved. This is a good moment to take care of front-end representations. Design is usually in the Design Specification Document (DSD).

### **4. Implementation**

The fourth element included in the SDLC is development. This is the stage that determines the actual creation of the software. Development should be supported by the use of access control or source code management applications. Both solutions allow developers to keep track of changes to the code. At the same time they ensure consistency between projects and guarantee meeting the set goals. Software Development Life Cycle in development has the ability to predict any delays resulting from waiting for test results or code compilation. It's worth emphasizing that software development requires appropriate information flow. Thanks to proper instructions and comments, it is possible to solve any technical problems and streamline work.

### **5. Testing & Deployment**

Any software should be properly tested before being made available to users. The testing process should answer the most important questions about the functionality of the final solution.

That is why security or performance tests are indispensable for eliminating bugs that the user might encounter. By eliminating such cases, project teams increase the level of satisfaction from using the software.

### **6. Maintenance**

The delivery of the project does not mean the end of the work at all. Equally important is maintenance, which involves making any regulations in the final phase. Such adjustments made by the maintenance team can be done on an ongoing basis or appear with each new version of the software.

## **REQUIREMENT & ANALYSIS**

### **3.1 Problem Definition :**

The existing healthcare management systems face challenges such as manual processes, limited accessibility, communication gaps, and data security concerns. These challenges hinder the efficiency and quality of healthcare services.

### **3.2 Requirement Specification :**

The requirements for the E-Health Care Management System include:

- User-friendly interfaces for administrators, patients, and healthcare professionals.
- Robust security measures to ensure confidentiality and integrity of patient information.
- Integration with existing healthcare databases and systems.
- Real-time communication tools for healthcare professionals.
- Automated appointment scheduling and reminders.
- Prescription and medication tracking features.
- Billing and insurance claims processing modules.

### 3.3 Software and Hardware Requirements :

#### 3.3.1 Software Requirements:

The software requirements include a secure database system, development frameworks, communication tools, and encryption protocols.

|                  | Requirement                   |
|------------------|-------------------------------|
| Front-end Tools  | HTML,CSS,JAVASCRIPT,BOOTSTARP |
| Back-end Tools   | PHP,MYSQL(DATABASE)           |
| Editor           | VS CODE,NOTEPAD++             |
| Browser          | GOOGLE CHROME,MICROSOFT EDGE  |
| Operating System | WINDOWS 10                    |

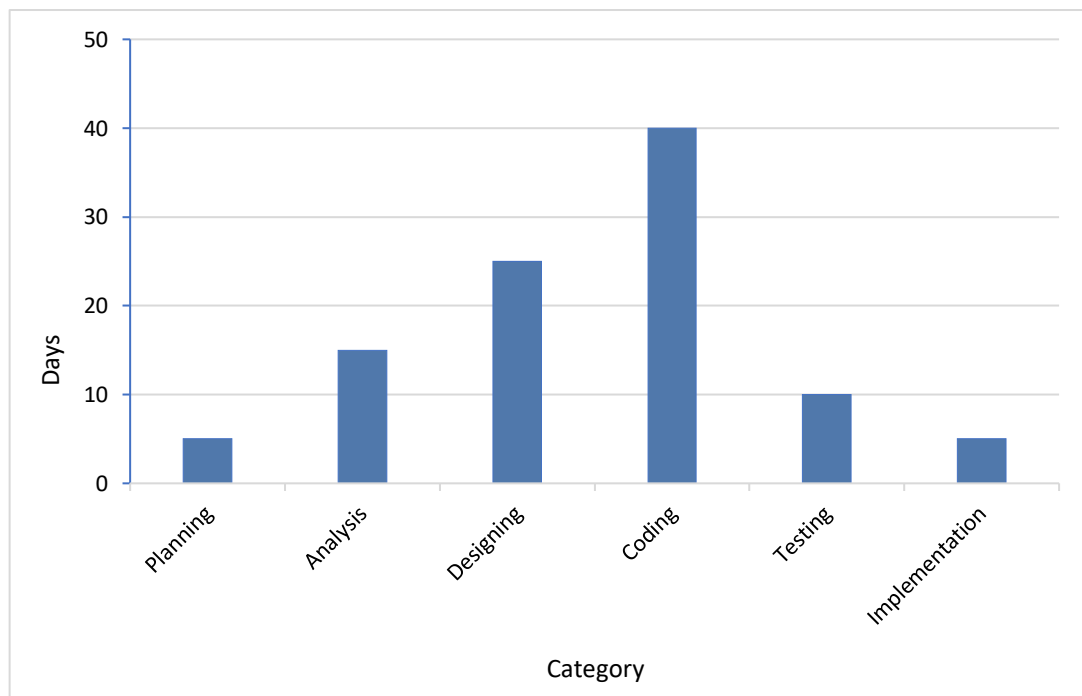
#### 3.3.1 Hardware Requirements:

The hardware requirements include servers for data storage, networking equipment, and devices for end-users.

| Component | Requirement           |
|-----------|-----------------------|
| Processor | Intel core I3,1.70GHz |
| RAM       | 512 MB                |
| Hard Disk | 20 GB                 |

### **3.4 Planning and Scheduling :**

The process of planning primarily deals with selecting the appropriate policies and procedures in order to achieve the objectives of the project. Scheduling converts the project action plans for scope, time cost and quality into an operating timetable.



**[Figure-3.4 Planning and Scheduling]**

### **3.5 Preliminary Production Description :**

#### **3.5.1 Admin**

The admin interface will provide tools for managing user accounts, overseeing system functionality, and generating reports.

#### **3.5.2 Patient**

Patients can access their health records, schedule appointments, receive prescriptions, and communicate with healthcare professionals through the patient portal.

#### **3.5.3 Doctor**

Doctors will have access to patient records, communication tools, appointment schedules, and prescription management features.

##### **Features:**

The E-Health Care Management System will include the following key features:

- Centralized Electronic Health Records (EHR): Digitized patient records for easy access and comprehensive management.
- Communication Tools: Real-time messaging and video conferencing for effective collaboration among healthcare professionals.
- Appointment Scheduling: Automated scheduling with reminders to improve the efficiency of healthcare services.
- Prescription and Medication Tracking: Tools for doctors to prescribe and track medications, enhancing patient safety.



- Billing and Insurance Claims: Streamlined processes for billing and insurance claims to reduce administrative burden.
- Patient Portal: A user-friendly portal for patients to access their health information, schedule appointments, and communicate with healthcare providers.

This E-Health Care Management System aims to bring about a positive change in healthcare delivery, ensuring efficiency, accessibility, and improved patient outcomes.

### **3.6 SYSTEM ANALYSIS :**

#### **3.6.1 OVERVIEW OF SYSTEM:**

- **PURPOSE:**

The purpose of this project is to create an integrated E-Health Care Management System that caters to the needs of healthcare providers, staff, and patients. The scope of the system includes:

- Patient Information Management
- Electronic Health Records (EHR) Integration
- Appointment Scheduling and Management
- Prescription and Medication Tracking
- Billing and Insurance Claims Processing
- Communication and Collaboration Tools for Healthcare Professionals
- Patient Portal for Access to Health Information

The system aims to optimize healthcare processes, improve patient care, and provide a user-friendly platform for healthcare stakeholders

- **Privacy & Security:**

- E-Health care Management System is designed with privacy and security features to protect patient's doctor's and admin's data and ensure compliance.

- **Availability:**

- E-Health care Management System is available for free to users with a registration, including Doctor and Patient.

- **FEATURE:**

- **User Login:**

The "User Login" feature in healthcare management systems allows authorized users to securely access their accounts using unique credentials. It ensures data security by verifying user identities before granting access to sensitive information. Upon login, users are directed to personalized dashboards or profiles based on their roles. Additional features may include password recovery and activity logs for audit purposes, ensuring proper access control and protecting patient data.

- **User Registration:**

The "User Registration" feature in healthcare management systems allows individuals to create new accounts by providing necessary information and creating unique login credentials. Validation checks ensure data accuracy, and additional verification steps may be included for security. Once registered, users gain access to tailored functionalities based on their roles. This feature facilitates user onboarding, access control, and customization within the system, enhancing usability and effectiveness.

- **Dash-Board:**

The "Dashboard" feature in healthcare management systems offers a central hub for users to access critical information and functionalities. It presents visually organized metrics such as patient appointments, medical records, and operational performance indicators. Customizable and interactive, it enhances efficiency and decision support by providing a comprehensive overview and integration with other system components.

- **View Doctor:**

The "View Doctor" feature allows users to access detailed profiles of medical professionals, including credentials, specialties, and patient reviews. With search filters based on location, specialty, and availability, it enhances transparency and trust in healthcare provider selection. Patients can book appointments directly and may access additional features like video introductions for a more personalized experience. Ultimately, this feature empowers patients to make informed decisions and fosters efficient healthcare experiences.

- **View Schedule:**

The "View Schedule" feature in healthcare management systems allows providers to easily access and manage their appointment schedules. It offers a centralized platform with customizable views for filtering appointments and updating statuses. Integrated with functionalities like patient reminders and billing systems, it streamlines workflow, enhances efficiency, and ensures a smooth healthcare experience for providers and patients alike.

- **Book Schedule:**

The "Book Schedule" feature enables patients to conveniently schedule appointments online by viewing available slots, selecting a suitable time, and inputting relevant information. With options to filter appointments and receive confirmation details and reminders, it empowers patients to take control of their healthcare. Integrated with real-time provider calendars, it ensures accurate availability and streamlines appointment scheduling for patients from anywhere, anytime.

- **Edit Schedule:**

The "Edit Schedule" feature in healthcare management systems allows providers to swiftly modify appointments, such as rescheduling or updating details, with seamless changes saved. Offering search options for efficient edits and integration with patient records and billing, it ensures coherence in care. Patients are promptly notified of changes, enhancing flexibility and efficiency for providers while maintaining care quality.

- **Delete Schedule:**

The "Delete Schedule" feature in healthcare management systems allows providers to effortlessly remove appointments from their schedules with a few clicks. It ensures accurate record-keeping. Providers can add notes before deletion, and patients are promptly notified of changes, streamlining schedule management while maintaining organization. Overall, this feature enhances efficiency in maintaining up-to-date schedules with minimal effort.

- **Feedback:**

The "Feedback" feature enables patients to share opinions and ratings on their healthcare experiences, including care quality and communication. Options for rating experiences and written comments facilitate detailed feedback. Providers analyze feedback to identify areas for improvement, fostering patient satisfaction and engagement. Integration with patient records maintains a comprehensive view, and responses to feedback promote transparency. Overall, this feature enhances patient-centered care and quality improvement in healthcare management systems.

- **View Article:**

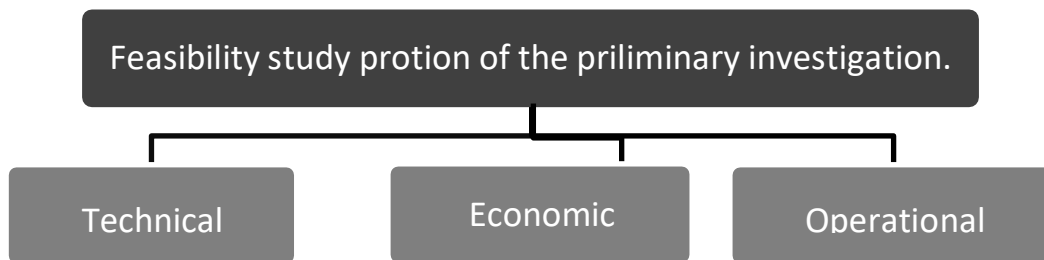
The "View Articles from Doctors" feature enables patients to access educational resources authored by healthcare providers, covering various medical topics and treatments. Patients gain reliable information directly from trusted professionals, enhancing health literacy and patient empowerment. Integration with patient profiles allows personalized recommendations based on individual histories, fostering communication and informed decision-making. Overall, this feature supports patient-provider communication and promotes informed decision-making in healthcare management systems.

### **3.6.2 FEASIBILITY STUDY:**

Feasibility Analysis is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report.

The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to established standards.

There are three aspects in feasibility study



**[Figure-3.6.2 Feasibility study]**

#### **1) Technical Feasibility:**

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility of proposed project refers to the software and hardware requirements.

The project is developing using HTML, CSS, PHP and other front end tool and MySQL is used for DBMS. The proposed project can be implementing on maximum browser support.

**2) Economic Feasibility:**

This feasibility determines whether there are sufficient benefits in creating to make the cost acceptable, or is the cost of the system too high. The software using to develop the proposed system is cost efficient. HTML, CSS and PHP tools are available for free and open source.

**3) Operational Feasibility:**

Operational feasibility assesses the range in which the required software performs a series of levels to solve business problems and customer requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed.

This system provides various functions, it is important to measure the feasibility of each function for measuring overall feasibility of this system. Status check, request form, adds mechanics etc. are easily operated using proposed project.



### **3.6.3 TOOLS :**

#### **3.6.3.1 Introduction to PHP (HYPERTEXT PRE-PROCESSOR):**

PHP is a scripting language originally designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications.

While PHP was originally created by Rasmus Lerdorf in 1995, the main implementation of PHP is now produced by the PHP Group and serves as the defector standard for PHP as there is no Formal specifications.



**[Figure-3.6.3.1 PHP]**

PHP is free software released under the PHP license; however it is incompatible with the GNU general public license (GPL), due to restrictions on the uses of the term PHP.

PHP is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP has grown from simple beginnings to a full-fledged object oriented language that can run both windows and UNIX/Linux platforms. The main advantages of using PHP over other platforms, such as Java or .net, are that it is smaller, much simpler to install, and more lightweight, needing only a fraction of the memory of the Java runtime of the .Net CLR. Since it is an open source language and we do not have to purchase it, it seemed best choice for the development of our project as we had to maintain a very low budget throughout our project.

#### **FEATURE OF PHP:**

- Simplicity
- Portability
- Speed
- Open source
- Extensible

### **3.6.3.2 Introduction to JAVASCRIPT:**

Java script is a scripting language used to enable programmatic access to objects within other applications. It was developed by Brendan Eich.

It is primarily used in the form of client-side JavaScript for the development of dynamic website. Java script is a dialect of the ECMA script standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions.



**[Figure-3.6.3.2 JAVASCRIPT]**

Java script was influenced by many languages and was designed to look like java, but be easier for non-programmers to work with.

Java script, despite the name, is essentially unrelated to the java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and java script copies many java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape building sun's Java runtime with their then-dominant browser. The key design principles within Java Script are inherited from the self and scheme programming languages.

### **FEATURES OF JAVASCRIPT:**

- Light Weight Scripting language
- Dynamic Typing
- Object-oriented programming support
- Functional Style
- Platform Independent
- Prototype-based
- Interpreted Language

### **3.6.3.3 Introduction to MYSQL:**

MySQL is a high-performance, multiuser relational database management system for database driven software applications. Designed around three fundamental principles-speed, stability and ease of use, and freely available under the GNU (General Public License). MySQL has been dubbed “the world’s most popular open-source database” by its parent company, MySQL AB.



**[Figure-3.6.3.3 MYSQL]**

Today, MySQL is available for a wide variety of platforms, including Linux, Mac OS and Windows.

### **FEATURE OF MYSQL:**

The following features of MySQL are

- Speed
- Reliability
- Security
- Scalability and portability
- Ease of use
- Compliance with existing standards
- Wide application support

#### **3.6.3.4 Introduction to HTML(HYPER TEXT MARKUP LANGUAGE):**

HTML stands for Hyper Text Markup Language. HTML is not only way to present information on the web, but it's the glue that holds everything to gather. In addition to being a markup language for displaying text, images and multimedia, HTML provides instructions to web browsers in order to control how documents are viewed and how they relate to each other. For all its simplicity, HTML is a very powerful language.



**[Figure-3.6.3.4 HTML]**

- HTML stands for Hyper Text Markup Language.
- An HTML file is a text file containing small markup tags
- The markup tags tell the web browser how to display to the page.
- An HTML file must have an HTM or HTML extension.
- An html file can be created using a simple text editor.

### **3.6.3.5 Introduction to CSS(CASCADING STYLE SHEET):**

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation (that is, the look and formatting) of a document written in a Markup language. It's most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document, including SVG and XUL.

CSS is designed preliminary to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the colours, fonts and layout.



**[Figure-3.6.3.5 CSS]**

This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on braille-based, tactile devices.

### **3.6.3.6 Introduction to BOOTSTRAP:**

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. It solves many problems which we had once, one of which is the cross-browser compatibility issue.



**[Figure-3.6.3.2 BOOTSTRAP]**

Nowadays, the websites are perfect for all the browsers (IE, Firefox and Chrome) and for all sizes of screens (Desktop, Tablets, and Phones).

All Thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

#### **Why Bootstrap?**

- Faster and Easier Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It designed to be responsive to mobile devices too.

## **SYSTEM DESIGN**

## **4.1 Database Design :**

### **4.1.1 Introduction of Data Dictionary**

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation. A Data Dictionary also provides metadata about data elements. The metadata included in a Data Dictionary can assist in defining the scope and characteristics of data elements, as well the rules for their usage and application.

### **What Is Data Dictionary?**

- A data dictionary contains metadata i.e data about the database.
- The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.
- The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc.

The data dictionary generally contains the information about the following:

- Names of all the database tables and their schemas.



- Details about all the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the table such as where they are stored and how.
- Table constraints such as primary key attributes and foreign key information etc.
- Information about the database views that are visible.

### 4.1.2 Types of Data Dictionary:

- ❖ The different type of data dictionary are as follows:

#### 1. Active Data Dictionary:

- If the structure of the database or its specifications change at any point of time, it should be reflected in the data dictionary.
- This is the responsibility of the database management system in which the data dictionary resides.
- So, the data dictionary is automatically updated by the database management system when any changes are made in the database.
- This is known as an active data dictionary as it is self-updating.

#### 2. Passive Data Dictionary:

- This is not as useful or easy to handle as an active data dictionary.

- A passive data dictionary is maintained separately to the database whose contents are stored in the dictionary.
- That means that if the database is modified the database dictionary is not automatically updated as in the case of Active Data Dictionary.

### 4.1.3 Why data dictionary is important?

- The main reason companies use data dictionaries is to document and share data structures and other information for all involved with a project or database.
- Using a shared dictionary ensures the same quality, meaning, and relevance for all data elements for all team members.
- The data dictionary will define conventions for the project and consistency throughout the dataset.
- Without a data dictionary, there's a higher risk of losing crucial information in translation and transition. Using a data dictionary also helps teams analyse the data easier later on.

### 4.1.4 Data Dictionary

Database Name : ehealthcare

#### 1. Table Name : admin

This Table Is Used To Store Information Of Admin.This Table Has Multiple Field Like admin\_id,admin\_email,admin\_name and admin\_password this field are used to login in system for administration.

| Sr.no | Field Name     | Data Type | Size | Constraint                     | Description                                |
|-------|----------------|-----------|------|--------------------------------|--|
| 1     | admin_id       | Int       | 3    | Primary Key,<br>Auto Increment | Use to Store Admin Id                      |
| 2     | admin_email    | Varchar   | 25   | Unique,<br>Not Null            | Use to Store Email Id                      |
| 3     | admin_name     | Varchar   | 15   | Not Null                       | Use to Store Admin Name                    |
| 4     | admin_password | Varchar   | 50   | Not Null                       | Use to Store Password In<br>Encrypted Form |
| 5     | admin_phoneno  | Varchar   | 12   | Not Null,<br>Unique            | Use To Store Admin Phone<br>No             |

### 2. Table Name : appointment

This Table Is Use to store the appointments Detail. This Table information Is Visible for Admin and particular doctor according to his schedule and also visible to patients for his/her appointments.

| Sr.no | Field Name  | Data Type | Size | Constraint                  | Description   |
|-------|-------------|-----------|------|-----------------------------|---|
| 1     | appo_id     | Int       | 3    | Primary Key, Auto Increment | Use To Store Unique Id Of Appointments                        |
| 2     | patient_id  | Int       | 3    | References (patient table)  | Use To Store Patient Id That Reference From patient Table     |
| 3     | appo_no     | Int       | 3    | Not Null                    | Use To Store Which No Of appointment is Book for This patient |
| 4     | sche_id     | Int       | 3    | References (schedule table) | Use To Store schedule Id That Reference From schedule Table   |
| 5     | appo_date   | Date      |      | Not Null                    | Use To Store Date Of Appointment                              |
| 6     | appo_status | Int       | 1    | Not Null                    | Use to Store Status Of Appointment                            |

### 3. Table Name : article

This Table Is Use When Any Doctor Want's TO Post His/her Article On any Health Topic Then This Table Is Used TO Store This Information . also It Contain Image Field With That Doctor Give Visual media and Additionally View Counter That Count Total View On Article .

| Sr.no | Field Name          | Data Type | Size | Constraint                  | Description  |
|-------|---------------------|-----------|------|-----------------------------|--|
| 1     | article_id          | Int       | 3    | Primary Key, Auto Increment | Use to Store Article Unique Id                           |
| 2     | doc_id              | Int       | 3    | Reference (doctor Table)    | Use to Store doctor Id That Reference From Doctor Table. |
| 3     | article_date        | Date      |      | Not Null                    | Use To Store Date of Article.                            |
| 4     | article_title       | Varchar   | 25   | Not Null                    | Use To Store Title Of Any Article.                       |
| 5     | article_description | Varchar   | 5000 | Not Null                    | Use To Store Detail of Article.                          |
| 6     | article_image       | Varchar   | 25   | Not Null                    | Use To Store Path Of Image That Use In Article.          |
| 7     | article_view        | Int       | 3    | Not Null                    | Use TO Store Total View Of Article                       |

#### 4. Table Name : comment

This Table Is Used TO Store Information of Article Comment That commented By Any Patient on system And All User Are Able TO See This Comment Publicly and share His/her Opinion.

| Sr.no | Field Name | Data Type | Size | Constraint                  | Description   |
|-------|------------|-----------|------|-----------------------------|---|
| 1     | cmt_id     | Int       | 3    | Primary Key, Auto Increment | Use To Store Unique Id Of Comment                         |
| 2     | article_id | Int       | 3    | References (article table)  | Use To Store article Id That reference From article Table |
| 3     | cmt_detail | Varchar   | 200  | Not Null                    | Use To Store Feedback Of Patient                          |
| 4     | patient_id | Int       | 3    | References (article table)  | Use To Store Patient Id That reference From patient Table |
| 5     | cmt_date   | Date      |      | Not Null                    | Use To Store Date Of Comment                              |
| 6     | cmt_time   | Time      |      | Not Null                    | Use To Store Time Of Comment                              |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

### 5. Table Name : doctor

This Table Is Used to Store Information Of Doctor.This Table Is Use When Any Patient Or doctor Want To See The Information Related TO Doctor On System Then This Table Is Use.

| Sr.no | Field Name   | Data Type | Size | Constraint                    | Description                                 |
|-------|--------------|-----------|------|-------------------------------|---|
| 1     | doc_id       | Int       | 3    | Primary Key, Auto Increment   | Use To Store Doctor Id                      |
| 2     | doc_email    | Varchar   | 25   | Unique , Not Null             | Use to store doctor email                   |
| 3     | doc_name     | Varchar   | 20   | Not Null                      | Use To Store Doctor's Name                  |
| 4     | doc_password | Varchar   | 50   | Not Null                      | Use to Store Password In Encrypted Formate  |
| 5     | doc_address  | Varchar   | 50   | Not Null                      | Use to store Address Of Hospital Or Clinic  |
| 6     | doc_gender   | Varchar   | 6    | Not Null                      | Use to Store Gender Of Doctor.              |
| 7     | doc_phoneno  | Varchar   | 12   | Unique , Not Null             | Use To Store Doctor's Phone Number          |
| 8     | spec_id      | Int       | 2    | References (specialist Table) | Use to Store Doctor's Specialist            |
| 9     | doc_img      | Varchar   | 30   | Not Null , unique key         | Use To Store Path Of Image                  |
| 10    | doc_charge   | Int       | 5    | Not Null                      | Use To Store Charge Of Doctor's Appointment |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

|    |                |         |     |          |  |
|----|----------------|---------|-----|----------|--|
| 11 | doc_dob        | Date    |     | Not Null | Use to Store Date of Birth Of Doctor               |
| 12 | doc_experience | Int     | 2   | Not Null | Use To Store How Many Years Experience Doctor Have |
| 13 | doc_about      | Varchar | 100 | Not Null | Use To Store Some Detail Of Doctor About Doctor    |



### 6. Table Name : feedback

This Table Is used To Store Information Of Feedback That Give  
By Patient for Doctor .

| Sr.no | Field Name           | Data Type | Size | Constraint                           | Description  |
|-------|----------------------|-----------|------|--------------------------------------|--|
| 1     | feedback_id          | Int       | 3    | Primary Key,<br>Auto Increment       | Use To Store Unique<br>Id Of Feedback                                      |
| 2     | doc_id               | Int       | 3    | References<br>(doctor table)         | Use To Store doctor<br>Id That Reference<br>From doctor Table              |
| 3     | sche_id              | Int       | 3    | References<br>(schedule table)       | Use To Store<br>schedule Id That<br>Reference From<br>schedule Table       |
| 4     | feedback_description | Varchar   | 200  | Not Null                             | Use To Store The<br>Feedback That Given<br>By Patient                      |
| 5     | patient_id           | Int       | 3    | References<br>(patient table)        | Use To Store Patient<br>Id That Reference<br>From patient Table            |
| 6     | rating               | Int       | 1    | Not Null                             | Use to Store Star<br>Given By Patient Out<br>Of 5                          |
| 7     | appo_id              | Int       | 3    | References<br>(appointment<br>table) | Use To Store<br>appointment Id That<br>Reference From<br>appointment Table |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

### 7. Table Name : patient

This Table Is Used to Store Information of Patient In System.

This table Is Use When Any Patient Want to Book His/her Appointment.

| Sr.no | Field Name       | Data Type | Size | Constraint                     | Description  |
|-------|------------------|-----------|------|--------------------------------|--|
| 1     | patient_id       | Int       | 3    | Primary Key,<br>Auto Increment | Use To Store Unique Id Of Each Patient             |
| 2     | patient_email    | Varchar   | 25   | Not Null,<br>Unique Key        | Use To Store Email Address Of Patient              |
| 3     | patient_name     | Varchar   | 15   | Not Null                       | Use To Store Name Of Patient                       |
| 4     | patient_password | Varchar   | 50   | Not Null                       | Use To Store Password Of Patient in encrypted Form |
| 5     | patient_gender   | Varchar   | 6    | Not Null                       | Use to Store Gender Of Patient                     |
| 6     | patient_address  | Varchar   | 50   | Not Null                       | Use To Store Address Of Patient                    |
| 7     | patient_dob      | Date      |      | Not Null                       | Use To Store Date Of Birth Of Each Patient         |
| 8     | patient_phoneno  | Varchar   | 12   | Not Null,<br>Unique key        | Use To Store Mobile Number Of Patient              |
| 9     | patient_img      | Varchar   | 30   | Not Null                       | Use To Store Image Of Patient In System            |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

### 8. Table Name : pending

This Table Is Used to Store Information Of Doctor At Register Time Unit Admin Approve Request. This Table information Is Only Visible For Admin To Inquiry Doctor Is Legit Or Not

| Sr.no | Field Name   | Data Type | Size | Constraint                    | Description                                 |
|-------|--------------|-----------|------|-------------------------------|---|
| 1     | doc_id       | Int       | 3    | Primary Key, Auto Increment   | Use To Store Doctor Id                      |
| 2     | doc_email    | Varchar   | 25   | Unique , Not Null             | Use to store doctor email                   |
| 3     | doc_name     | Varchar   | 20   | Not Null                      | Use To Store Doctor's Name                  |
| 4     | doc_password | Varchar   | 50   | Not Null                      | Use to Store Password In Encrypted Formate  |
| 5     | doc_address  | Varchar   | 50   | Not Null                      | Use to store Address Of Hospital Or Clinic  |
| 6     | doc_gender   | Varchar   | 6    | Not Null                      | Use to Store Gender Of Doctor.              |
| 7     | doc_phoneno  | Varchar   | 12   | Unique , Not Null             | Use To Store Doctor's Phone Number          |
| 8     | spec_id      | Int       | 2    | References (specialist Table) | Use to Store Doctor's Specialist            |
| 9     | doc_img      | Varchar   | 30   | Not Null , unique key         | Use To Store Path Of Image                  |
| 10    | doc_charge   | Int       | 5    | Not Null                      | Use To Store Charge Of Doctor's Appointment |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

|    |                |         |     |          |  |
|----|----------------|---------|-----|----------|--|
| 11 | Doc_dob        | Date    |     | Not Null | Use to Store Date of Birth Of Doctor   |
| 12 | Doc_experience | Int     | 2   | Not Null | Use To Store How Many Years Experience Doctor Have                             |
| 13 | Doc_about      | Varchar | 100 | Not Null | Use To Store Some Detail Of Doctor About Doctor                                |
| 14 | Status         | Int     | 1   | Not Null | Use To Set Status Of Doctor Request<br>[1-Pending<br>2-Accepted<br>3-Rejected] |

### 9. Table Name : rating

This Table Is Used To Store Information On Rating Related That How many patient rate the Doctor Or How many Points .

| Sr.no | Field Name        | Data Type | Size | Constraint                  | Description                 |
|-------|-------------------|-----------|------|-----------------------------|-----------------------------|
| 1     | rate_id           | Int       | 3    | Primary Key, Auto Increment | Use To Store Schedule Id    |
| 2     | doc_id            | Int       | 3    | References (doctor table)   | Use To Store Doctor Id      |
| 3     | rate_total_rating | Int       | 4    | Not Null                    | Use To Store Schedule Title |
| 4     | rate_total_review | Int       | 4    | Not Null                    | Use To Store Schedule Date  |

### 10. Table Name : schedule

This Table Is Use To Store Information Of Doctor's Session Or schedule. This Schedule Are Create by Doctor It Self and Also Admin Can manage It.

| Sr.no | Field Name  | Data Type | Size | Constraint                  | Description                                   |
|-------|-------------|-----------|------|-----------------------------|---|
| 1     | sche_id     | Int       | 3    | Primary Key, Auto Increment | Use To Store Schedule Id                      |
| 2     | doc_id      | Int       | 3    | References (doctor Table)   | Use To Store Doctor Id                        |
| 3     | sche_title  | Varchar   | 20   | Not Null , unique key       | Use To Store Schedule Title                   |
| 4     | sche_date   | date      |      | Not Null                    | Use To Store Schedule Date                    |
| 5     | sche_start  | time      |      | Not Null                    | Use To Store Schedule Starting Time           |
| 6     | sche_end    | time      |      | Not Null                    | Use To Store Schedule Ending Time             |
| 7     | sche_noappo | Int       | 3    | Not Null                    | Use TO Store Maximum Appointment For Schedule |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

### 11. Table Name : specialist

This Table Is Use to Store Information Of All Specialist. This Table Is Used In Doctor table For Get Doctor's Specialist.

| Sr.no | Field Name | Data Type | Size | Constraint                  | Description                     |
|-------|------------|-----------|------|-----------------------------|---------------------------------|
| 1     | spec_id    | Int       | 2    | Primary Key, Auto Increment | Use to Store Specialist Type Id |
| 2     | spec_type  | Varchar   | 25   | Not Null                    | Use to Store Specialist Type    |

### 12. Table Name : transaction

This Table Is Used TO Store Information of Transaction That Perform By Any Patient On Booking Appointment .This Table Have Multiple Reference Table For get Multiple Data Easy

| Sr.no | Field Name | Data Type | Size | Constraint                        | Description   |
|-------|------------|-----------|------|-----------------------------------|---|
| 1     | tra_id     | Int       | 3    | Primary Key,<br>Auto Increment    | Use To Store Unique Id Of Transaction                             |
| 2     | doc_id     | Int       | 3    | References<br>(doctor table)      | Use To Store Doctor Id That reference From doctor Table           |
| 3     | sche_id    | Int       | 3    | References<br>(schedule table)    | Use To Store schedule id That Reference From schedule Table       |
| 4     | patient_id | Int       | 3    | References<br>(patient table)     | Use To Store patient id That Reference From schedule Table        |
| 5     | appo_id    | Int       | 3    | References<br>(appointment table) | Use To Store Appointment Id That Reference From appointment Table |
| 6     | charge     | Int       | 5    | Not Null                          | Use To Store Charge Of Schedule / Session .                       |
| 7     | tra_status | Int       | 1    | Not Null                          | Use To Store Status Of Transaction                                |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

### 13. Table Name : user

This Table Is used When Any User Login In system It check Tyoe Of User And Redirect the reference to According Table On System For login .

| Sr.no | Field Name | Data Type | Size | Constraint                     | Description  |
|-------|------------|-----------|------|--------------------------------|--|
| 1     | user_id    | Int       | 3    | Primary Key,<br>Auto Increment | Use to Store user id   |
| 2     | user_email | varchar   | 25   | Not Null                       | Use to Store email id of all system user                       |
| 3     | user_name  | Varchar   | 15   | Not Null                       | Use to store name  |
| 4     | user_type  | char      | 1    | Not Null                       | Use to Store User Type<br>[A-Admin,<br>P-Patient,<br>D-Doctor] |



## **4.2 Data Flow Diagram :**

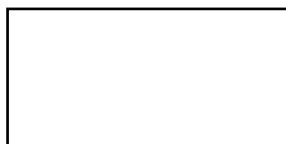
### **4.2.1 What is Data Flow Diagram?**

- A Data Flow Diagram(DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically.
- It shows how information enters and leaves the system, what changes the information and where information is stored.
- The purpose of DFD is to show the scope and boundaries of a system as a whole.
- It may be used as a communications tool between a system analyst and any person who plays a part in the system that acts as the starting point for redesigning a system
- It usually begins with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system.
- To elaborate further from that, we drill down to a level 1 diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required.
- Progression to levels 3, 4 and so on is possible but anything beyond level 3 is not very common.

### **4.2.2 Component of Data Flow Diagram**

- The two main types of notation used for data flow diagrams are Yourdon-Coad and Gane-Sarson, both named after their creators, all experts who helped develop DFD methodology: Ed Yourdon, Peter Coad, Chris Gane and Trish Sarson.
- There are some differences in style between the notation types. For example, Yourdon and Coad notation uses circles to represent processes, whereas Gane and Sarson notation use rectangles with rounded corners.
- Because DFD symbols vary, it's important to be consistent with whatever notation you choose in order to avoid confusion.
- Here we will be using the Yourdon-Coad notations
- All data flow diagrams include four main elements: entity, process, data store and data flow.

- **Entity:**

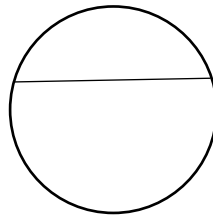


**[Entity]**

- Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed.

- Since they are external to the system being analysed, these entities are typically placed at the boundaries of the diagram.

- **Process :**



**[Process]**

- An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all process must have inputs and outputs on a DFD.
- Processes are typically oriented from top to bottom and left to right on a data flow diagram.

- **Data Source :**



**[Data Source]**

- A data store does not generate any operations but simply holds data for later access

- Data stores could consist of files held long term or a batch of documents stored briefly while they wait to be processed.
- Input flows to a data store include information or operations that change the stored data. Output flows would be data retrieved from the store.

- **Data Flow :**



**[Data Flow]**

- Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow.
- Input and output data flows are labelled based on the type of data or its associated process or data store, and this name is written alongside the arrow.

### **4.2.3 Rules of Data Flow Diagram**

- All Data Flows must flow to or from a process.
- A process must have at least one input flow and one output flow.
- The input to a process must be sufficient to produce output flows.
- Process must transform data.
- Data cannot flow between two entities.
- Data cannot flow directly from an entity to data store.

### **4.2.4 Advantage of Data Flow Diagram**

Using a data flow diagram offers several benefits, especially during system analysis, design, and documentation stages. Here are some of the critical advantages of employing DFDs:

#### **1. Visual Representation**

DFDs provide a clear graphical representation of a system's processes, data flows, data stores, and external entities. This visual element helps technical and non-technical stakeholders grasp system components and their interrelationships more easily.

#### **2. System Overview**

The context diagram (level 0 DFD) offers a bird's-eye view of the entire system, facilitating a high-level understanding of system boundaries, major processes, and external interactions.

#### **3. Modular Decomposition**

DFDs allow for a top-down modular decomposition of a system. As one moves from higher-level DFDs to more detailed ones, one can delve deeper into specific system aspects without getting overwhelmed by the system's entirety.

#### **4.Communication Tool**

DFDs are an excellent communication tool between analysts, designers, developers, and other stakeholders. They ensure everyone consistently understands the system's structure and functionality.

#### **5.Identification of Redundancies**

DFDs can help identify redundant or unnecessary data processes by mapping out data flows, leading to streamlined system design.

#### **6.Enhanced Error Detection**

DFDs can aid in pinpointing inconsistencies, missing elements, or potential bottlenecks within the system, which can then be addressed during the design phase.

#### **7. Documentation**

DFDs contribute to system documentation, providing future developers, analysts, and managers with valuable insights into system operations and data flow.

#### **8. Facilitates System Improvements**

Over time, as the system needs to evolve or be upgraded, DFDs can assist in pinpointing areas of improvement, integration, or modification.

### **9. Boundary Clarification**

DFDs help clarify a system's boundaries by distinguishing between external entities and internal processes. This distinction is crucial for defining the scope of system development projects.

### **10. Validation**

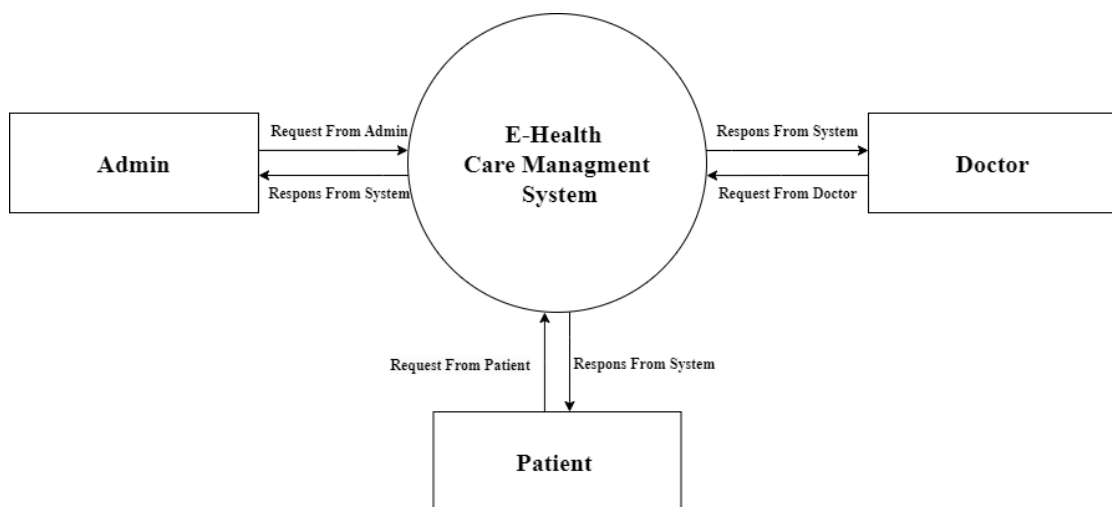
DFDs can validate the proposed design with end-users or stakeholders, ensuring that the design aligns with the system's goals and requirements.

DFDs act as a roadmap for system development, offering clarity, facilitating communication, and ensuring the system is designed efficiently and effectively.

#### 4.2.5 Data Flow Diagram

##### SYSTEM : E-Health Care Management System

❖ Level-0 DFD :



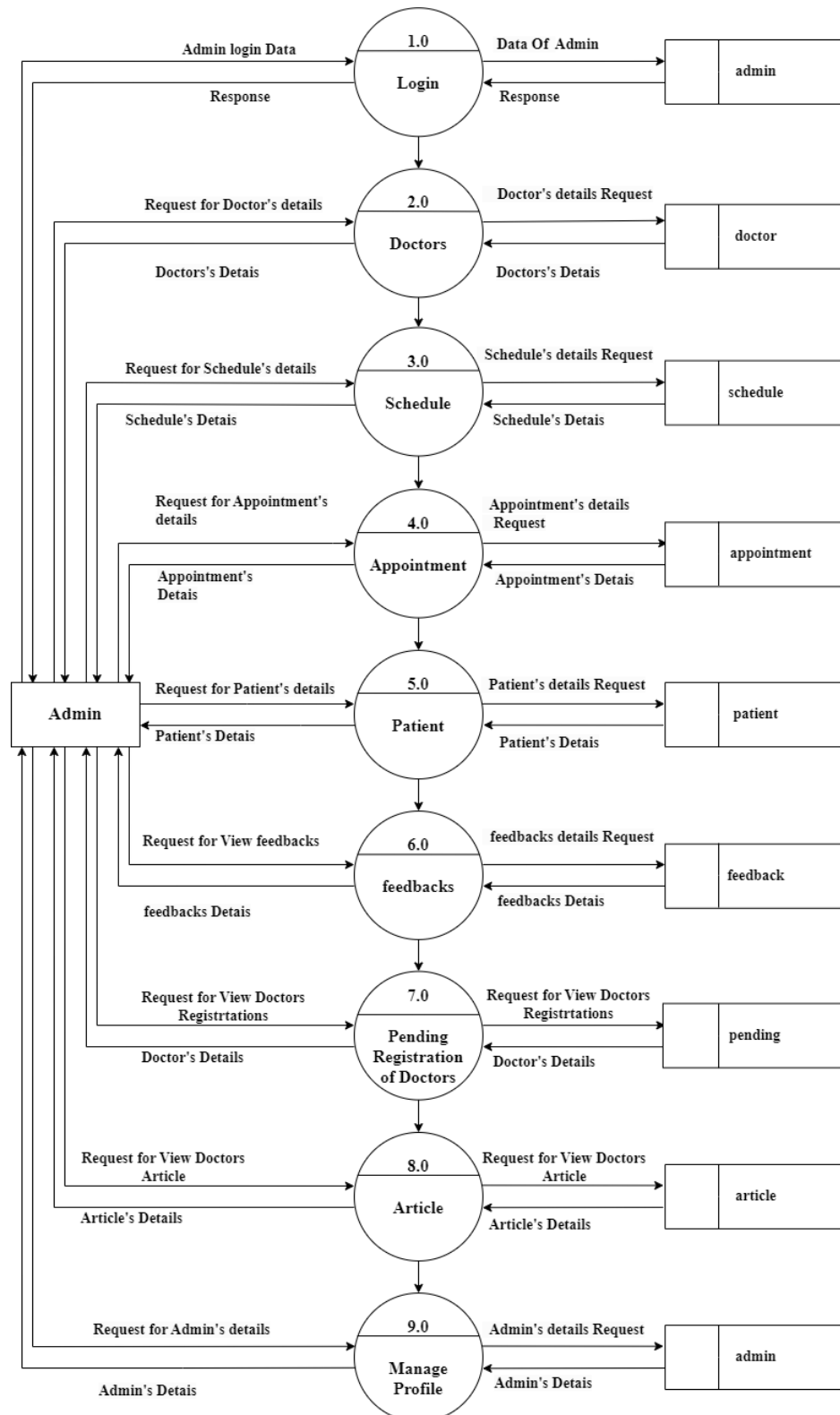
[Figure- 4.2.5.1 Level-0 DFD]



# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

### ❖ Admin Level-1 DFD:

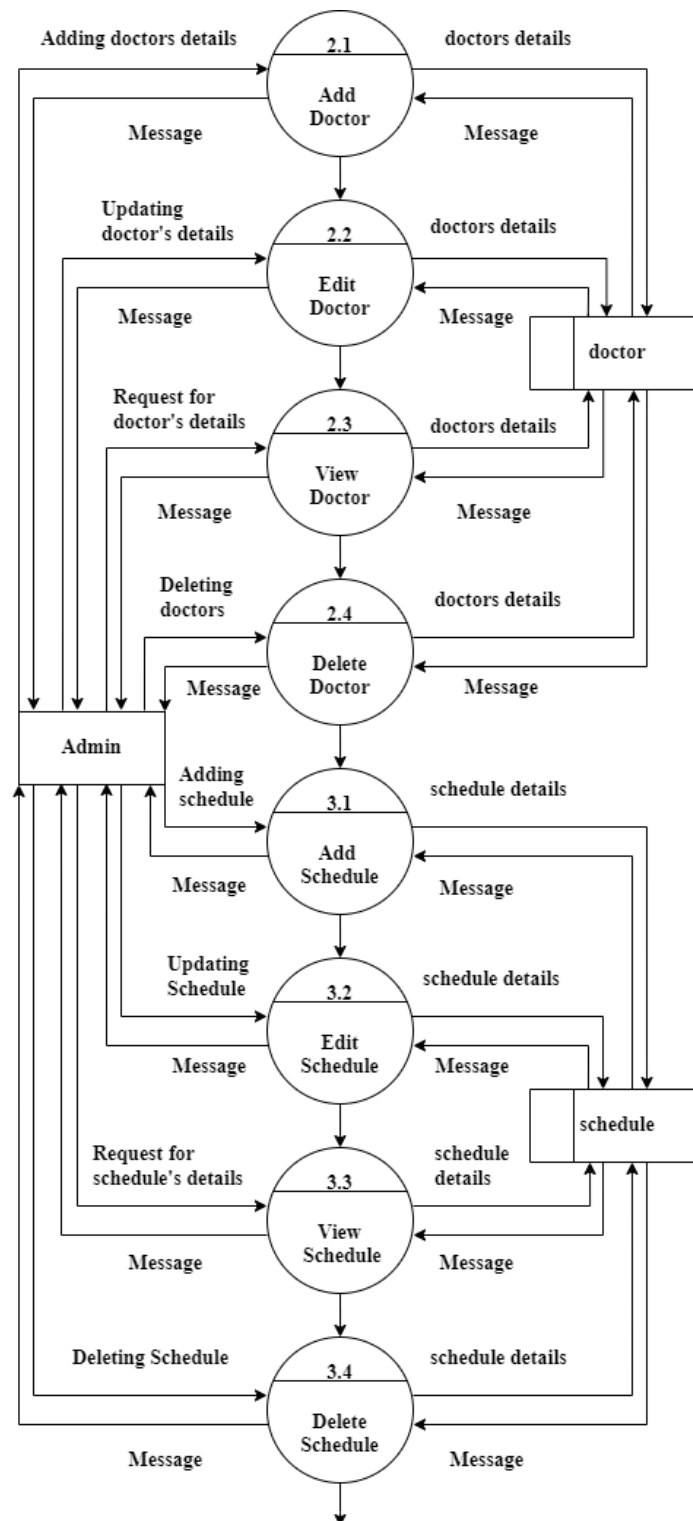


[Figure-4.2.5.2 Admin Level-1 DFD]

**Shree Swaminarayan Collage Of Computer Science**  
**Project Title : E-Health Care Management System**

---

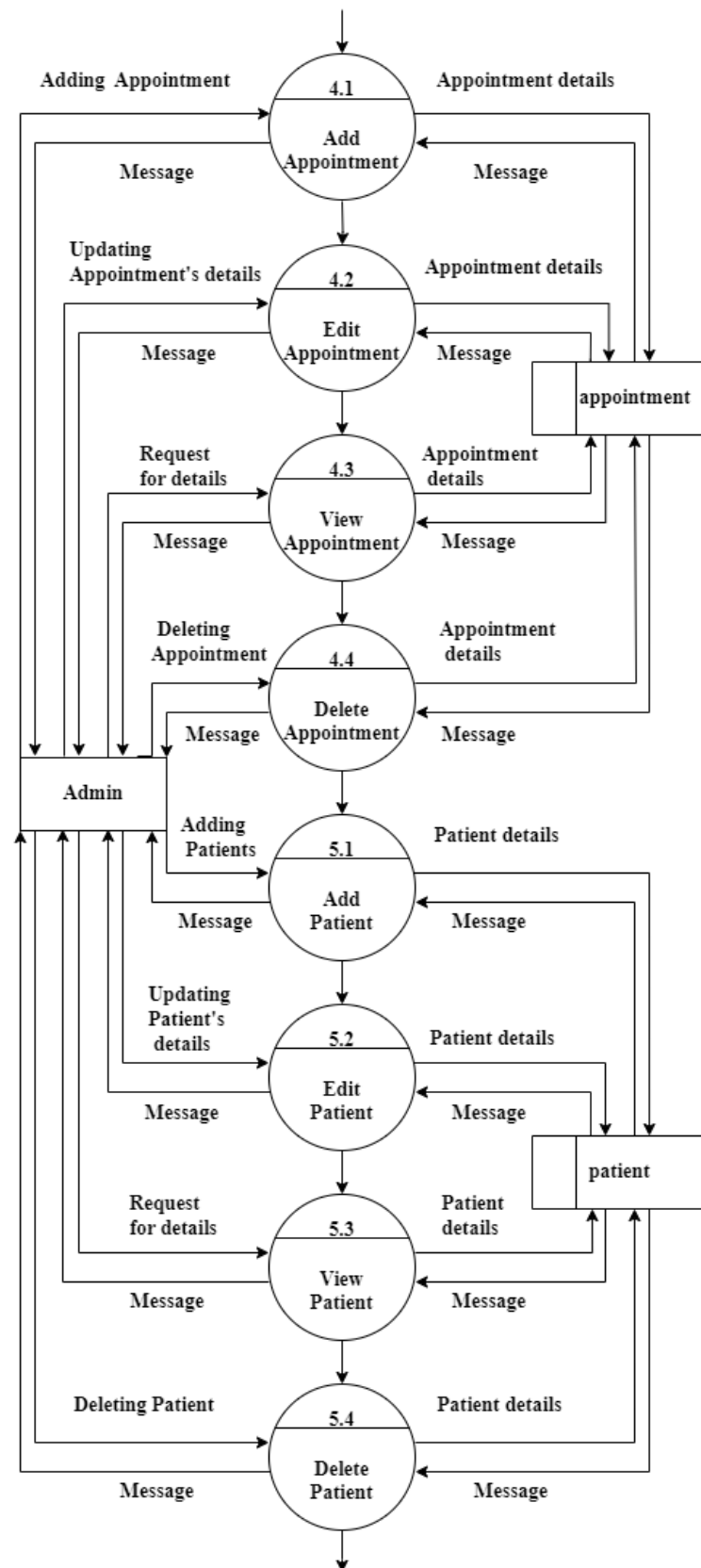
❖ **Admin Level-2 DFD :**



**To Be Continue...**

**Shree Swaminarayan Collage Of Computer Science**  
**Project Title : E-Health Care Management System**

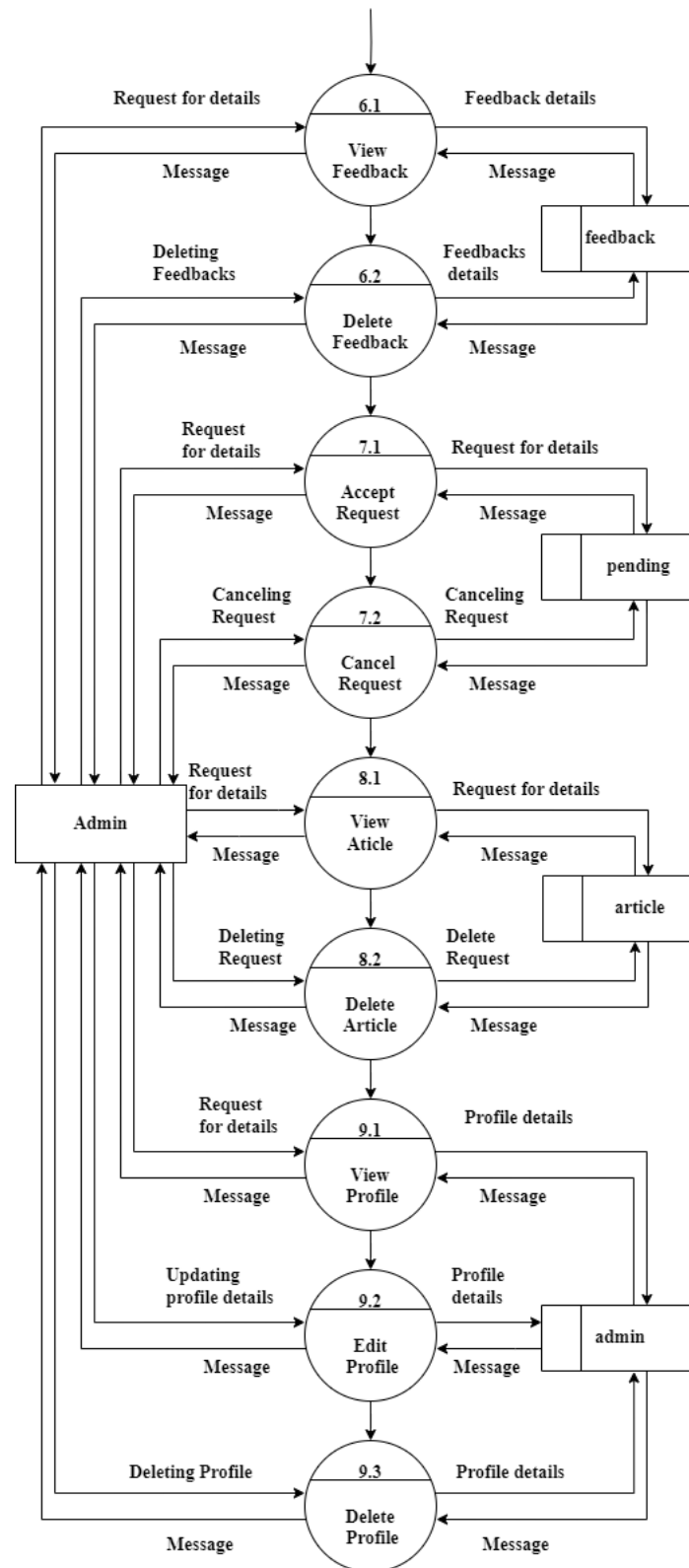
---



**To Be Continue...**

**Shree Swaminarayan Collage Of Computer Science**  
**Project Title : E-Health Care Management System**

---

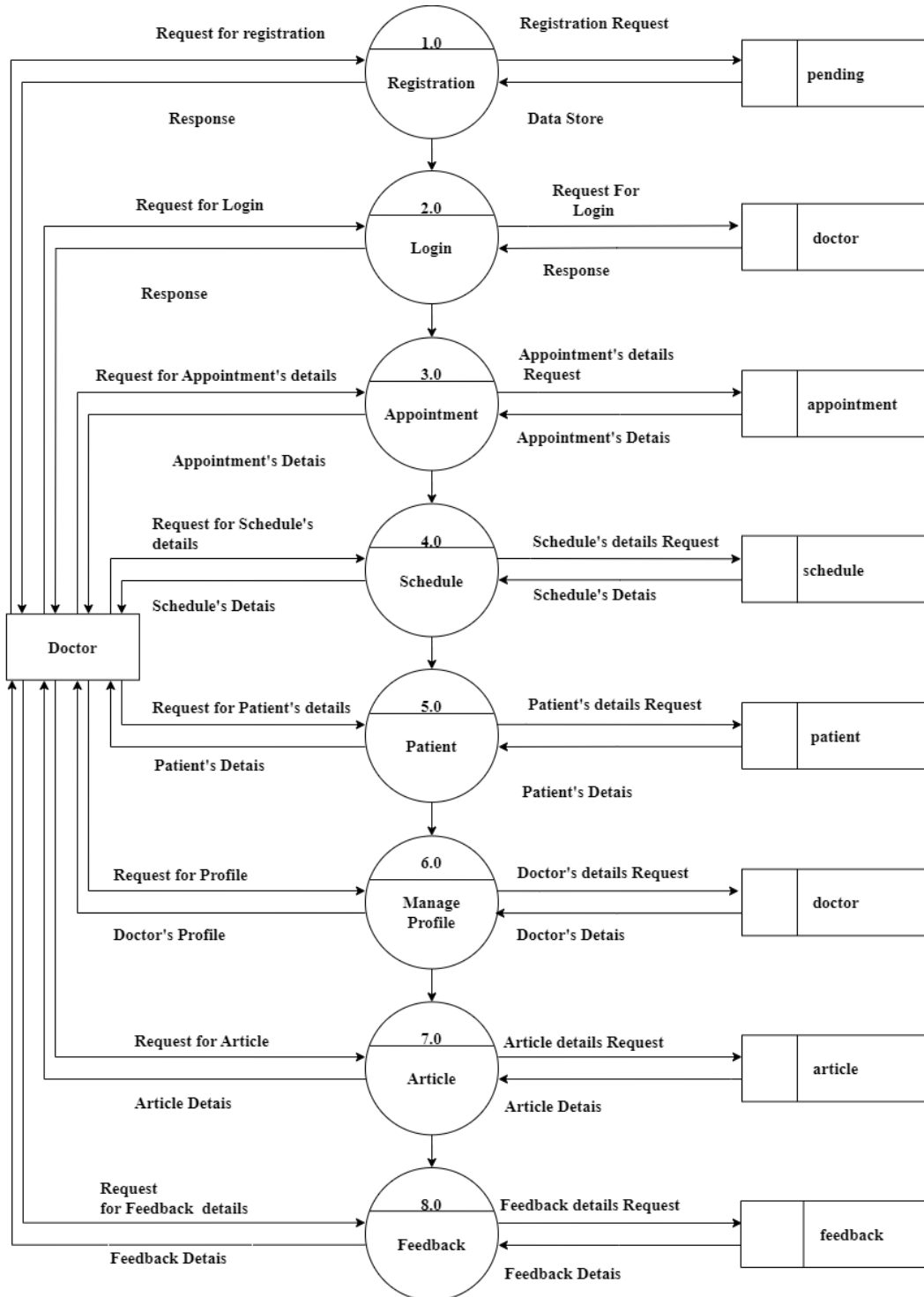


**[Figure-4.2.5.3 Admin Level-2 DFD]**

# Shree Swaminarayan Collage Of Computer Science

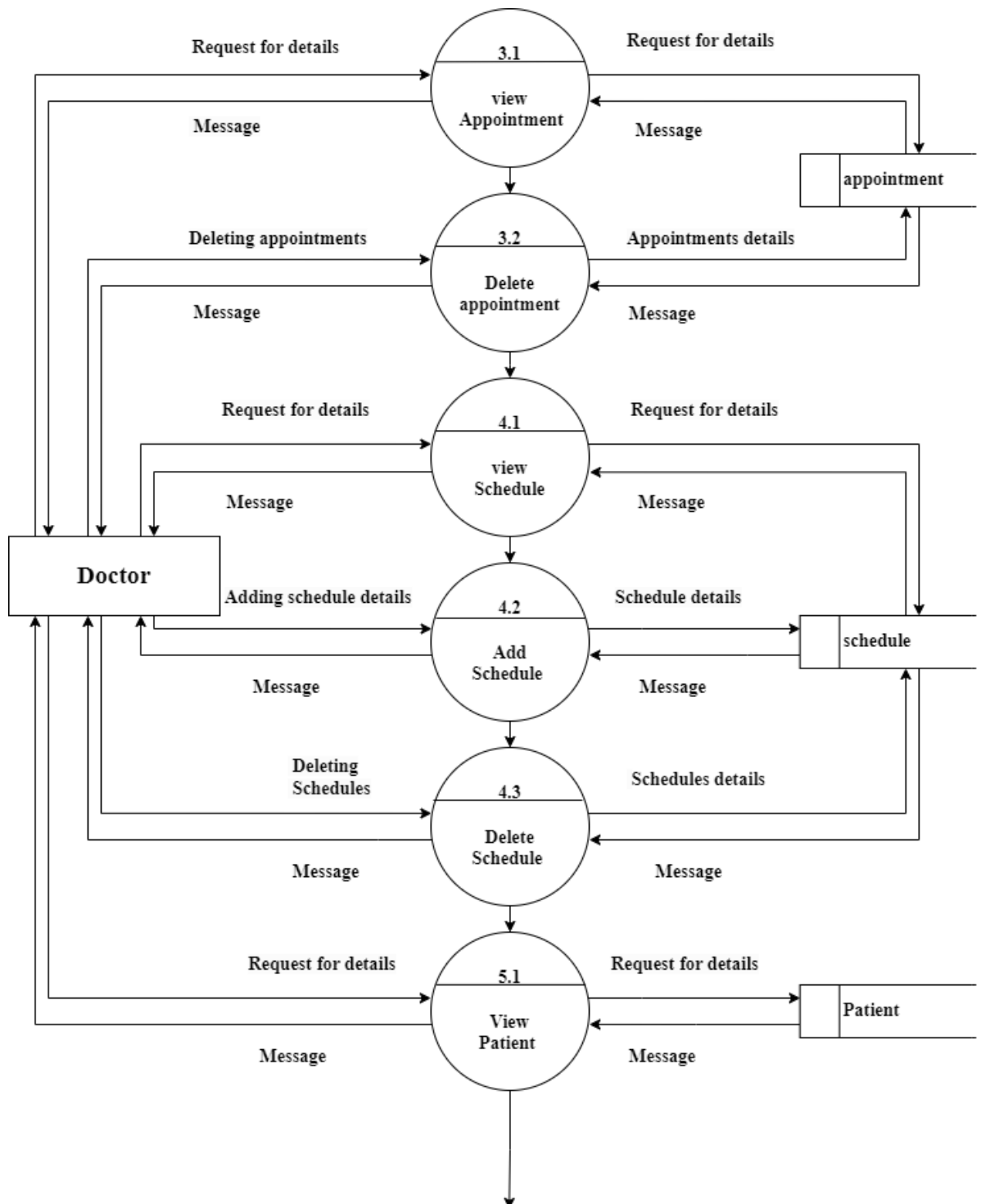
## Project Title : E-Health Care Management System

### ❖ Doctor Level-1 DFD:



[Figure-4.2.5.4 Doctor Level-1 DFD]

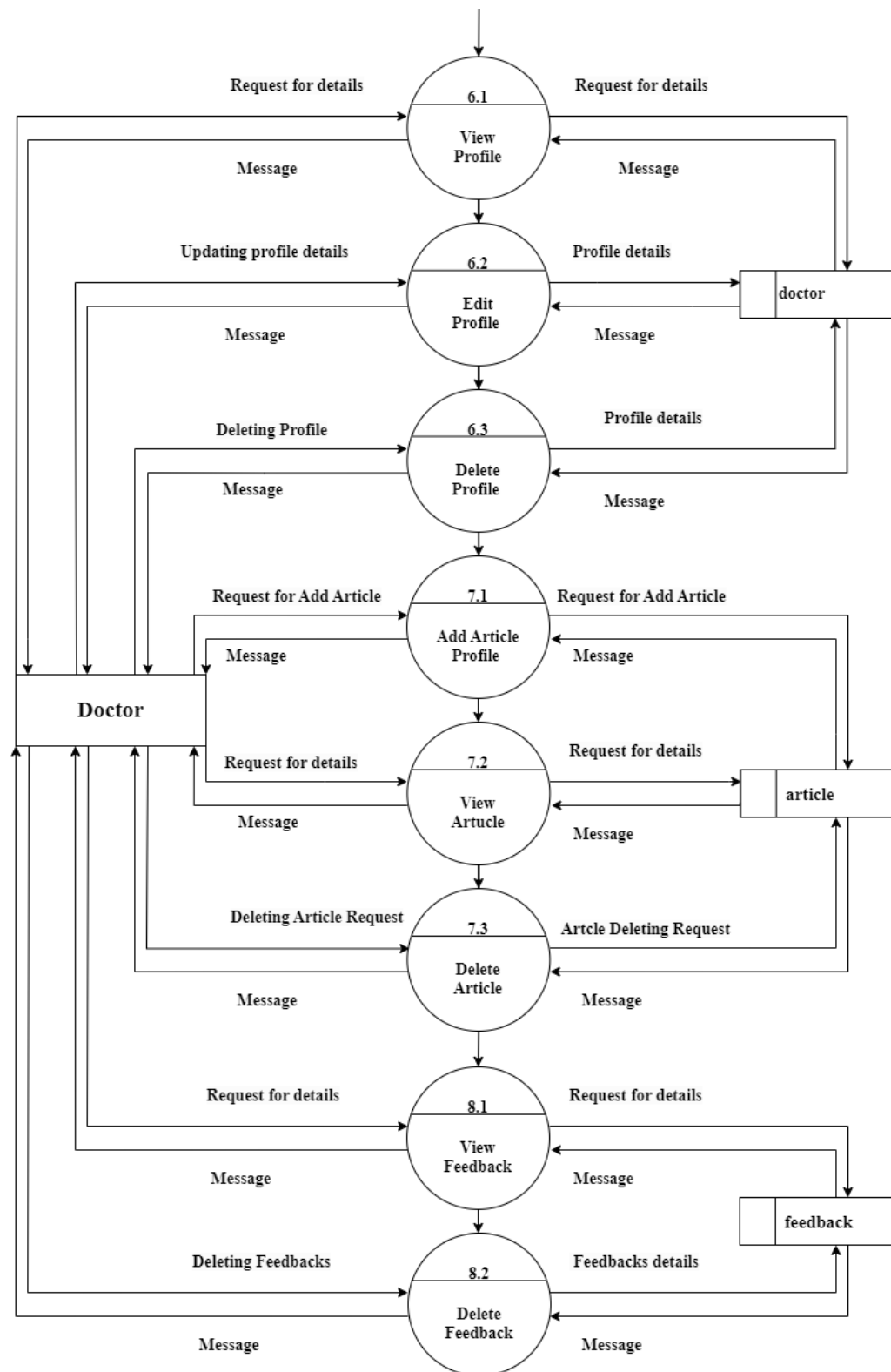
❖ Doctor Level-2 DFD:



To Be Continue...

**Shree Swaminarayan Collage Of Computer Science**  
**Project Title : E-Health Care Management System**

---

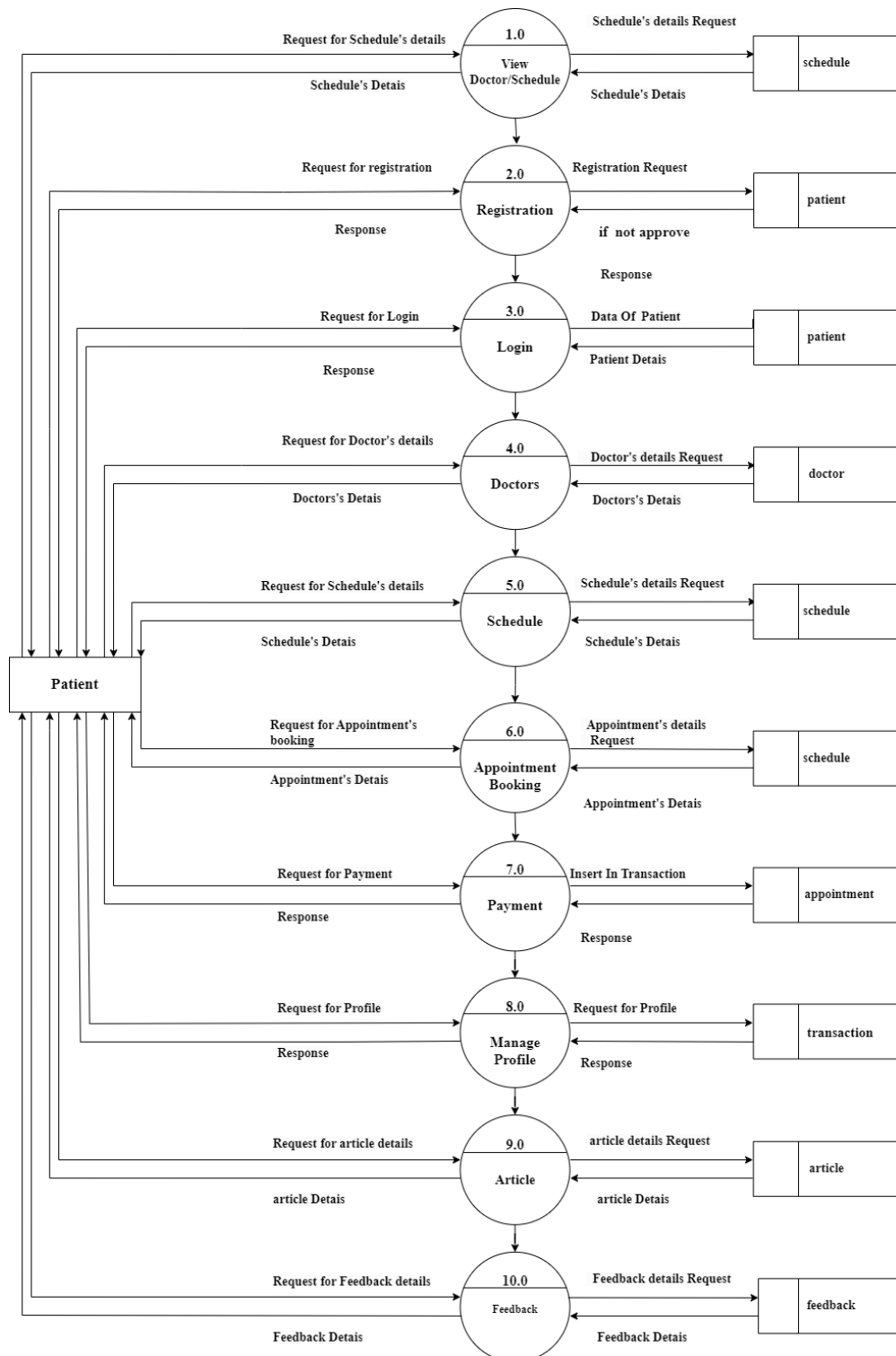


**[Figure-4.2.5.5 Doctor Level-2 DFD]**

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

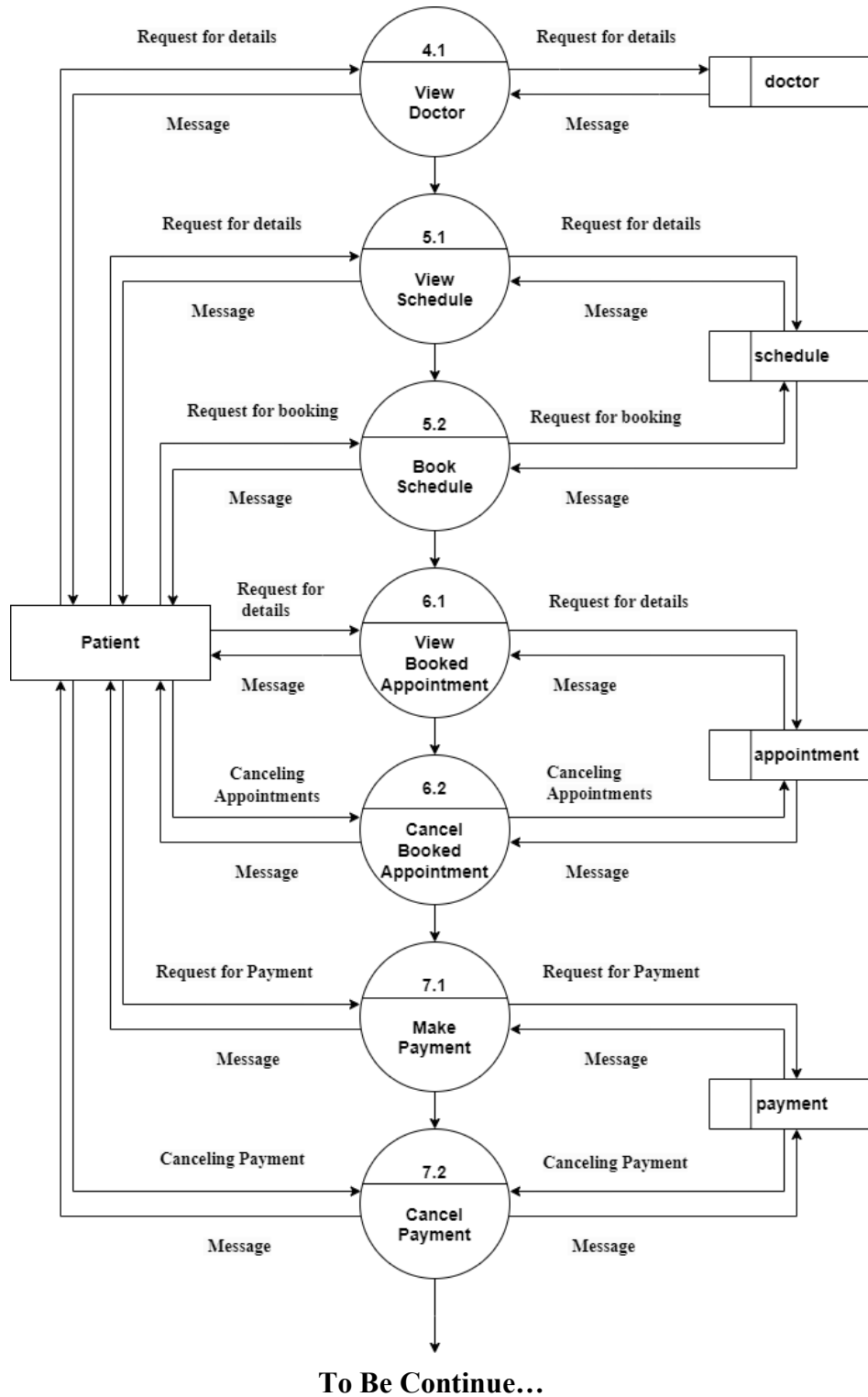
### ❖ Patient Level – 1 DFD:



[Figure-4.2.5.6 Patient Level – 1 DFD]



❖ **Patient Level – 2 DFD:**





## **4.3 E-R Diagram :**

### **4.3.1 Introduction**

An Entity Relation(ER) Diagram is a specialized graphics that illustrates the interrelationship between entities in a database. ER diagrams often use symbols to represent 3 different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

An Entity Relationship Model (ERM), in software engineering is an abstract and conceptual representation of data. Entity Relationship modelling is a relational schema database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relation database, and its requirements in a top-down fashion.

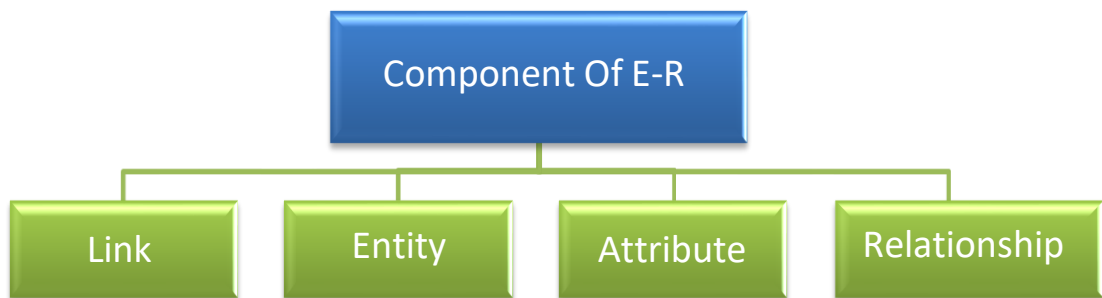
### **4.3.2 Why use ER diagram?**

Here, are prime reasons for using the ER Diagram.

- Helps you to define terms related to entity relationship modeling. It Provide a preview of how all your tables should connect, what fields are going to be on each table.
- Helps to describe entities, attributes, relationships. It ER diagrams are translatable into relational tables which allows you to build databases quickly.
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications.

- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram.
- ERD Diagram allows you to communicate with the logical structure of the database to users.

#### **4.3.3 Component of ER diagram**



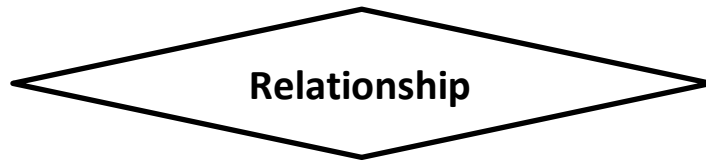
**[Figure-4.3.3 Component of ER Diagram]**

##### **1. ENTITY:**



Entity is the thing which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object defined within the information system about which you want to store information. Entities are distinct things in the enterprise.

## 2. RELATIONSHIP:



A relationship describes how entities interact. For example, the entity “carpenter” may be related to the “table” entity by the relationship “builds” or “makes”. Relationship are represented by diamond shapes and are labelled using verbs.

There are four type of relationship

| onship                    | Symbol              |
|---------------------------|---------------------|
| One to one relationship   | <u>1</u> — <u>1</u> |
| One to many relationship  | <u>1</u> — N        |
| Many to one relationship  | N — <u>1</u>        |
| Many to many relationship | <u>N</u> — <u>N</u> |

## 3. ATTRIBUTE:



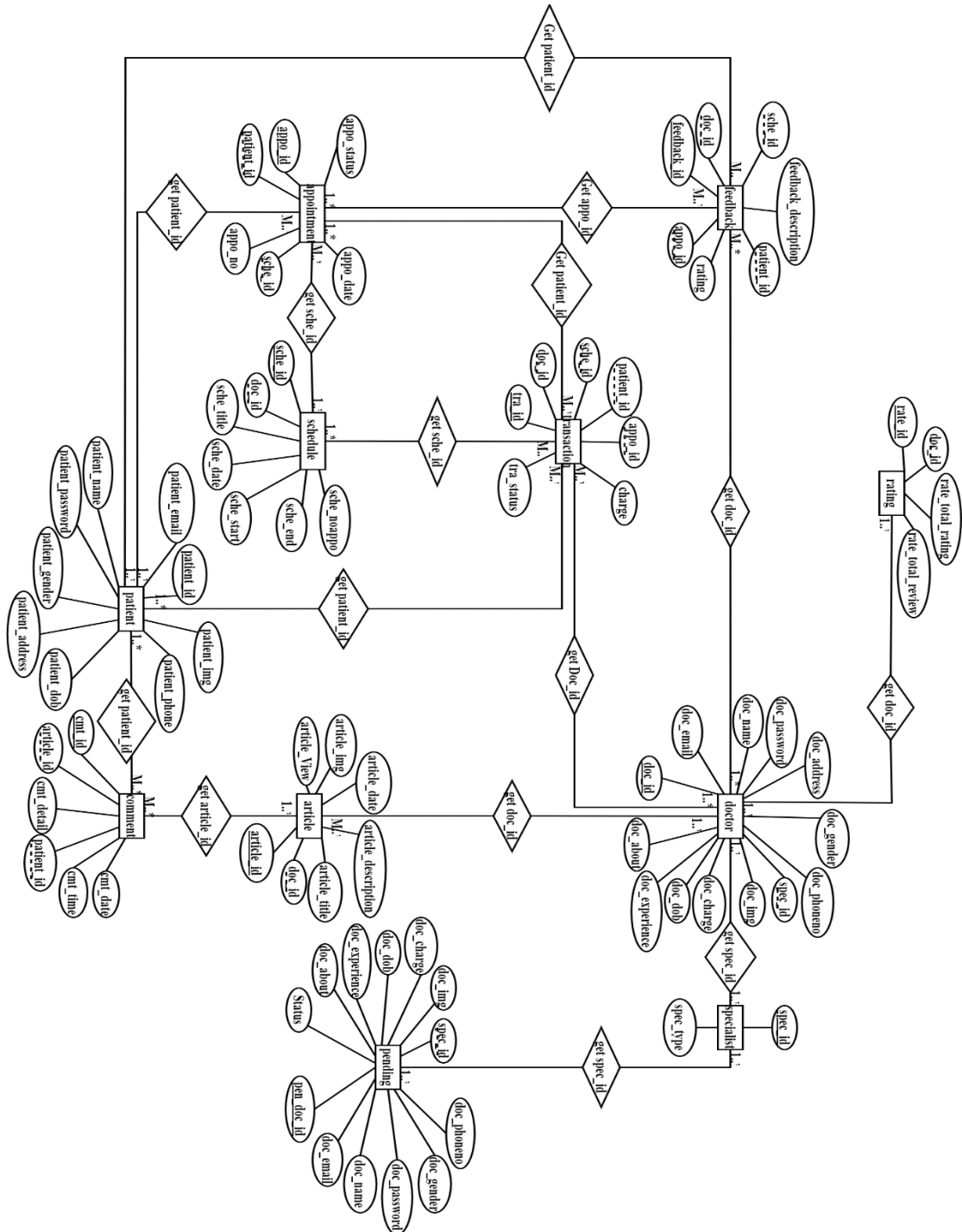
Attribute are the properties of the entities and relationship, descriptor of the entity. Attribute are elementary pieces of information attached to an entity.

**4. LINK:**

\_\_\_\_\_

Link is connected entity to attribute or attribute to entity.

#### 4.3.4 ER Diagram



[Figure-4.3.4 ER Diagram]

## 4.4 UML Diagrams :

### 4.4.1 Introduction

UML (Unified Modeling Language) is a general-purpose, graphical modeling language in the field of Software Engineering. UML is used to specify, visualize, construct, and document the artifacts (major elements) of the software system.

The UML was developed in 1994-95 by Grady Booch, Ivar Jacobson, and James Rumbaugh at the Rational Software. In 1997, it got adopted as a standard by the Object Management Group (OMG).

It is a pictorial representation of classes, objects, and relationships between them. UML diagram is a model that describes a part of a system. It is used to define the functionality or a design of a system.

There are two broad categories of diagrams and they are again divided into subcategories.

#### ❖ Structural Diagrams

Structure diagrams show the things in the modelled system. In a more technical term, they show different objects in a system.

#### ❖ Behavioural Diagrams

Behavioural diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system.



#### **4.4.2 Characteristic of UML:**

It is a generalized modeling language.

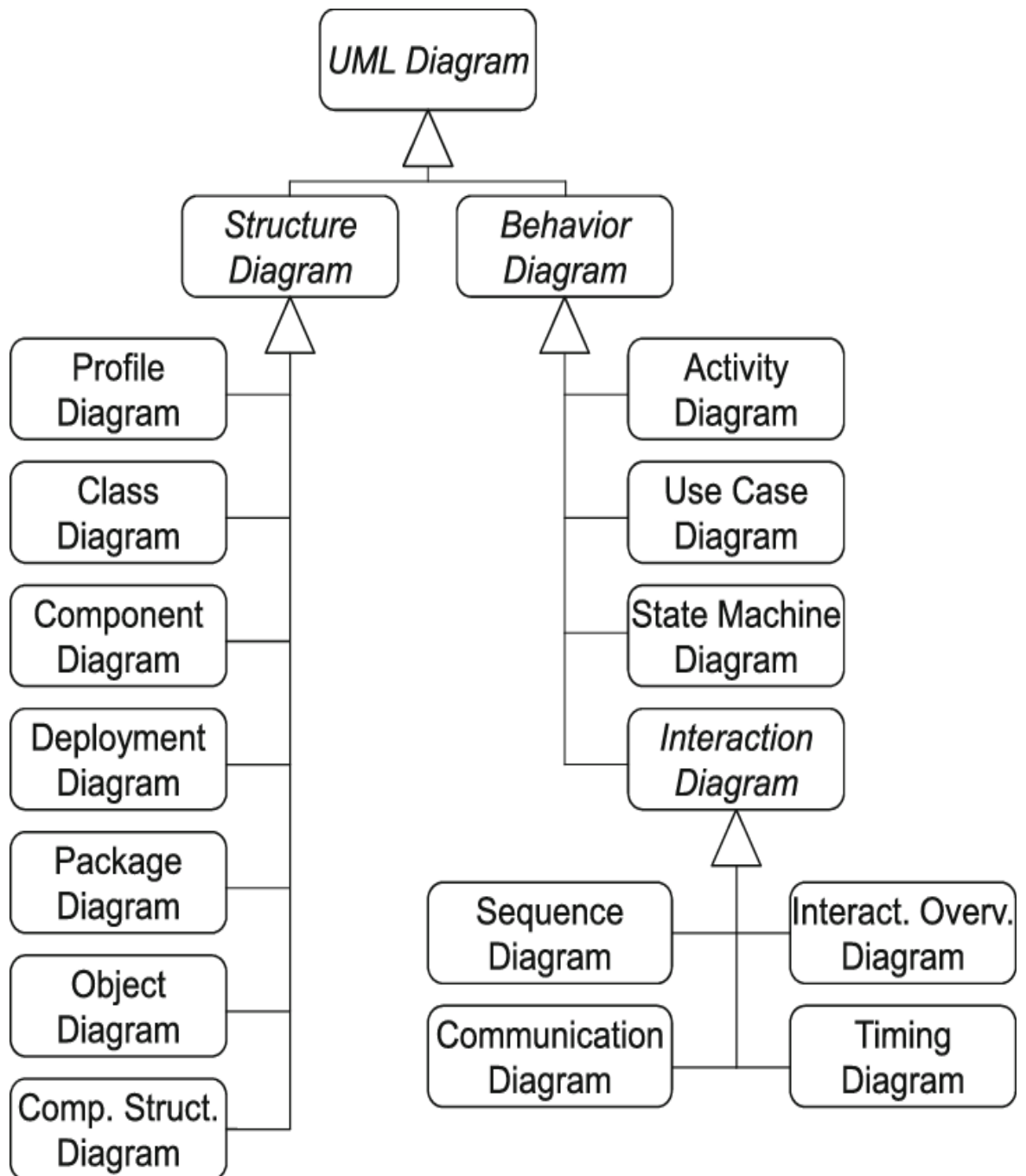
It is different from software programming languages such as Python, C, C++, etc.

It is a pictorial language which can be used to generate powerful modeling elements.

It is related to object-oriented designs and analysis.

It has unlimited applications even outside the software industry. It can be used to visualize the workflow of a factory.

#### 4.4.3 Types of UML:



[Figure-4.4.3 Types of UML]

### 4.4.3.1 USE CASE :

#### ❖ What Is Use Case Diagram:

- A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped.
- Use cases specify the expected behaviour (what), and not the exact method of making it happen.
- Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram).
- A key concept of use case modeling is that it helps us design a system from the end user's perspective.
- It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour.

#### ❖ Origin Of Use Case Diagram:

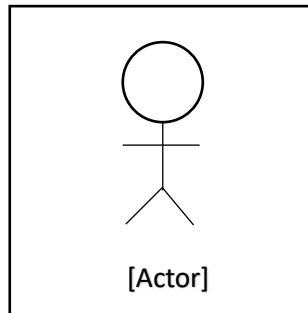
These day use case modelling is often associated with UML, although it has been introduced before UML existed. Its brief history is as follow:-

In 1986, Ivar Jacobson first formulated **textual** and **visual modelling** techniques for specifying use cases.

In 1992 his co-authored book Object-Oriented Software Engineering - A Use Case Driven Approach helped to popularize the technique for capturing functional requirements, especially in software development.

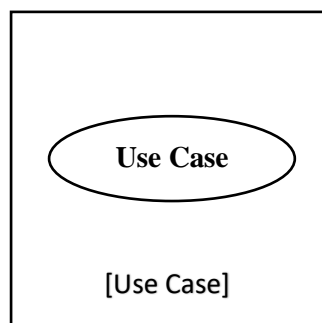
**❖ Symbols / Notations Of Use Case Diagram :**

**i. ACTOR :**



- The actor is an entity that interacts with the system. An actor is an entity that initiates the use case from outside the scope of a use case.
- It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system. A user is the best example of an actor.

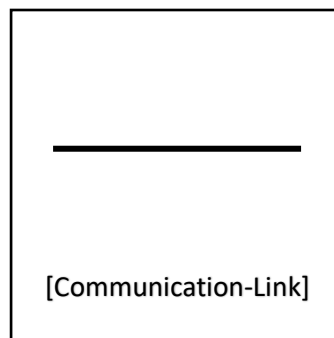
**ii. USE CASE:**



- A use case describes a sequence of actions that provide something to an actor.

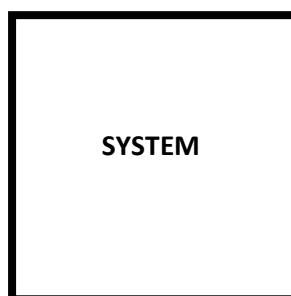
- Use cases are used to represent high-level functionalities and how the user will handle the system

### iii. COMMUNICATION LINK:



- The participation of an actor in a use case is shown by connecting an actor to a use case by a solid link.
- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

### iv. BOUNDARY OF SYSTEM:



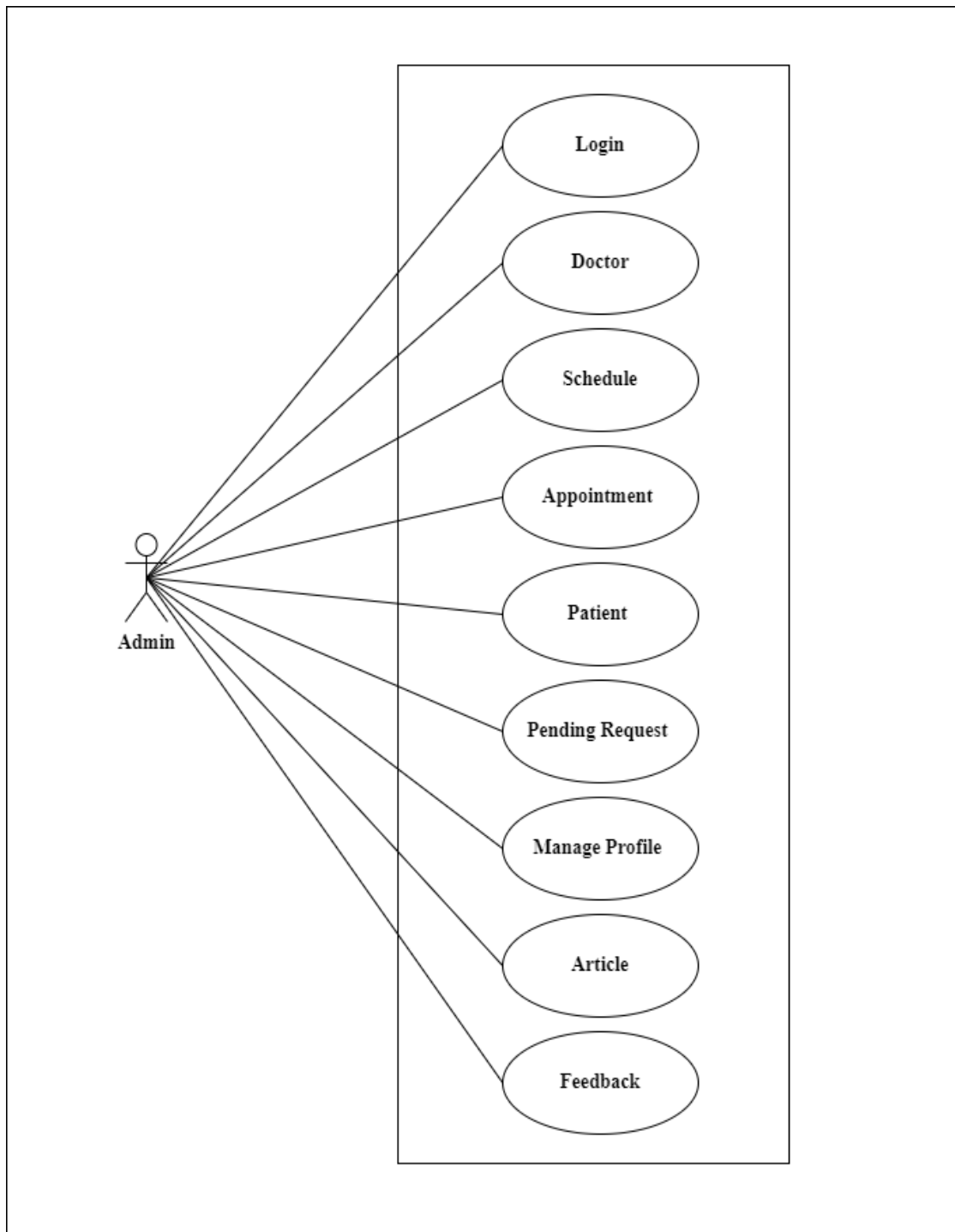
# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

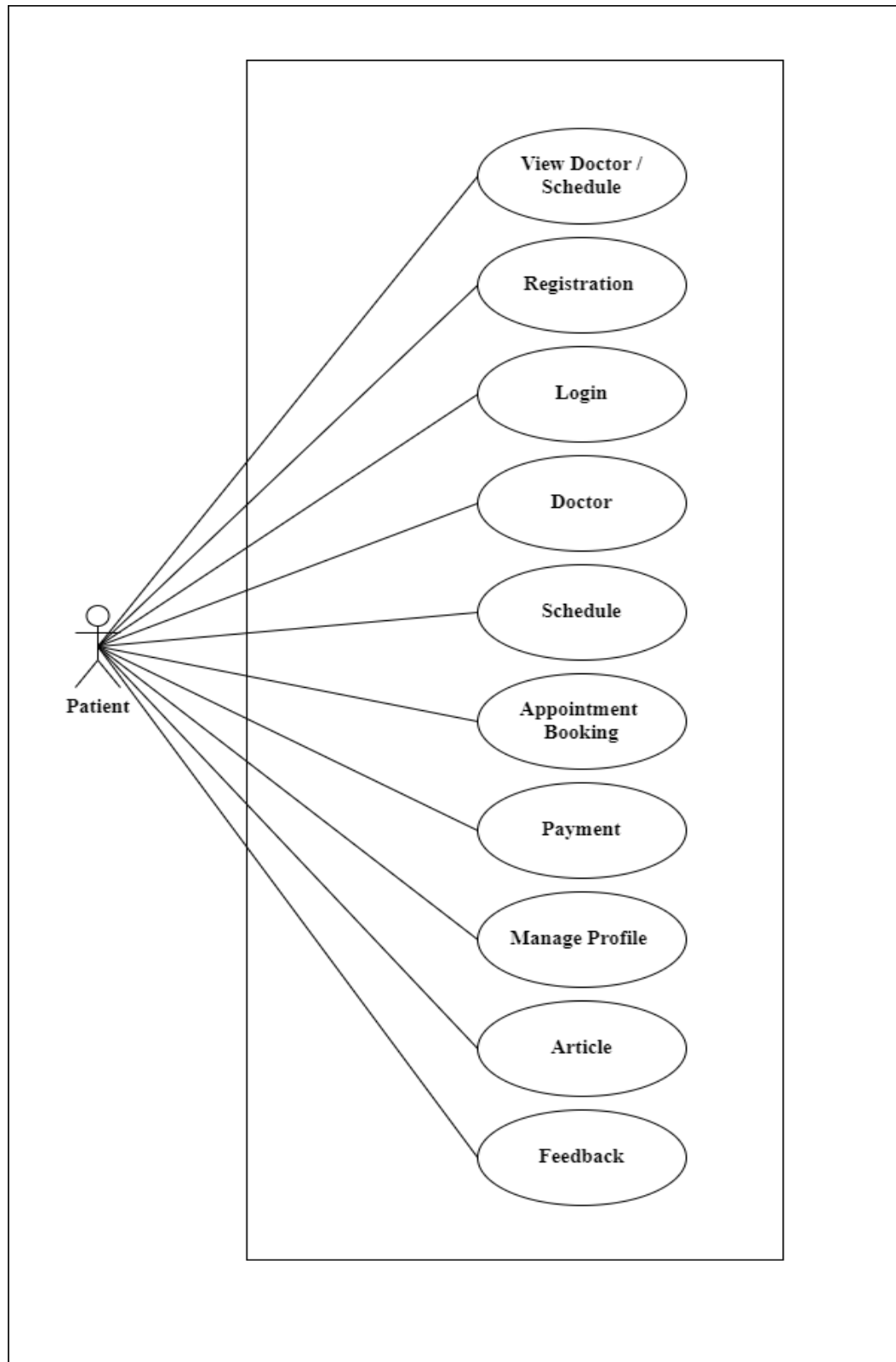
- The system boundary is the entire system as defined in the requirements document. It indicates the scope of your system.
- Anything within the box represents functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used.

❖ **Use-case Diagram [Admin]:**



**[Figure-4.4.3.1.1 Use-case Diagram-Admin]**

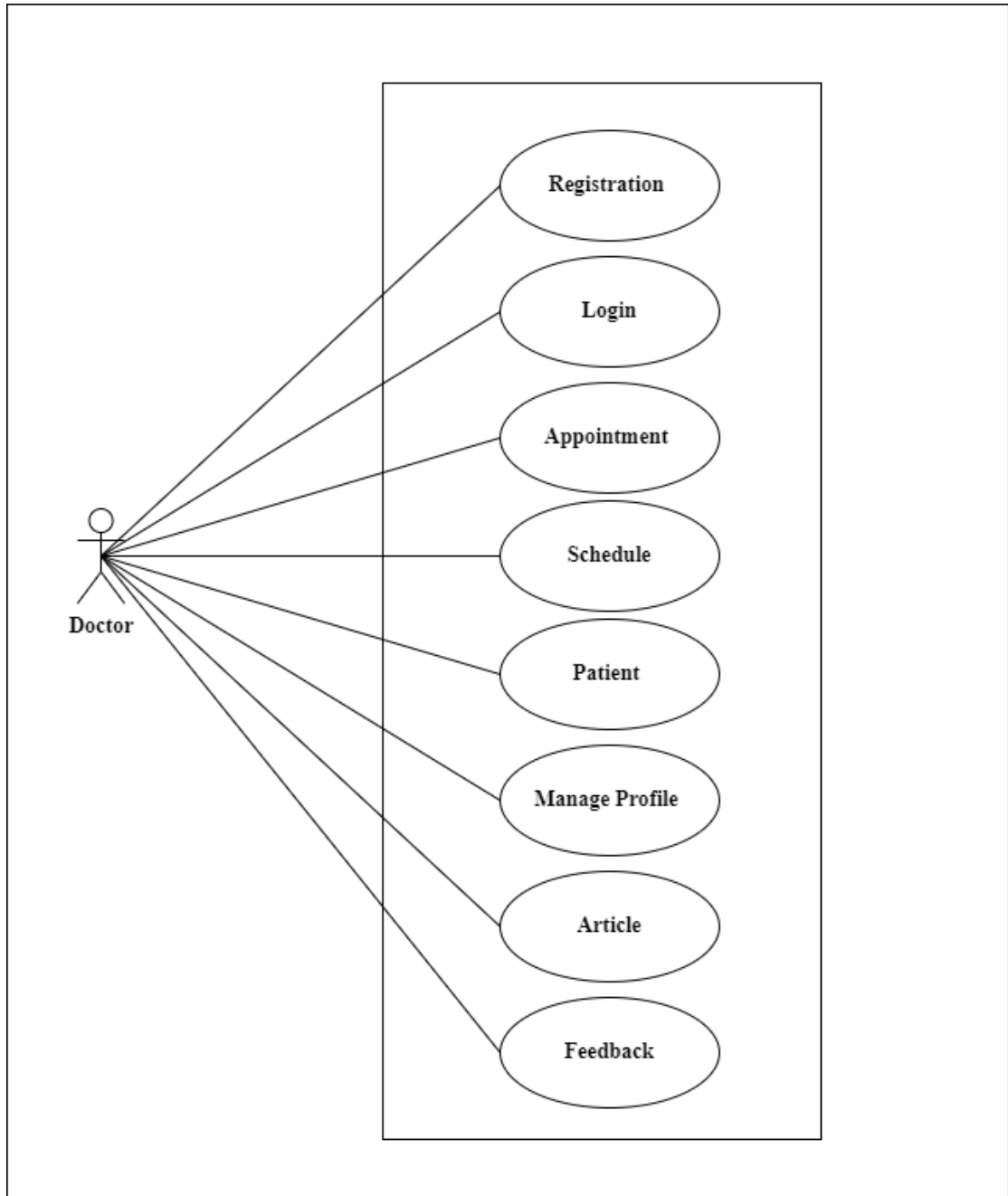
❖ **Use-case Diagram [Patient]:**



**[Figure-4.4.3.1.2 Use-case Diagram -Patient]**



**❖ Use-case Diagram [Doctor]:**



**[Figure-4.4.3.1.3 Use-case Diagram-Doctor]**

#### **4.4.3.2 ACTIVITY DIAGRAM :**

##### **❖ What is Activity Diagram?**

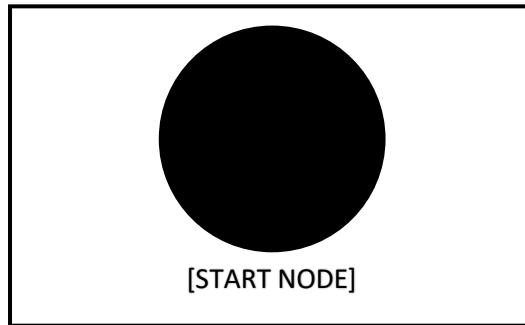
- Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity.

##### **❖ When to use Activity Diagram?**

- Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction.
- Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.
- It is also suitable for modelling how a collection of use cases co-ordinate to represent business workflows.
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction

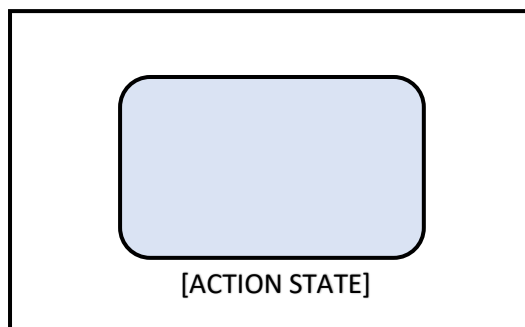
**❖ SYMBOL AND NOTATION OF ACTIVITY DIAGRAM :**

**• START NODE :**



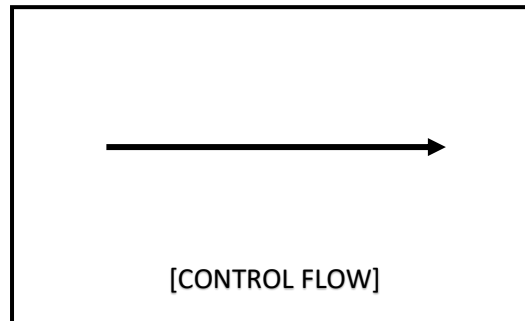
- A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.

**• ACTION STATE :**



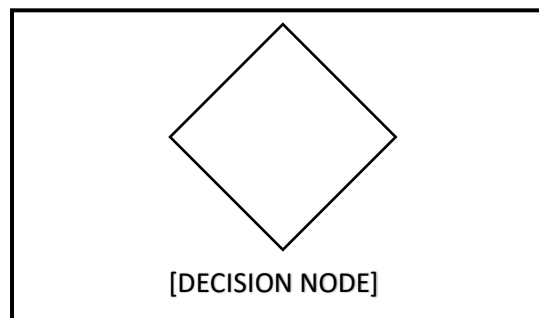
- An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.

- **CONTROL FLOW:**



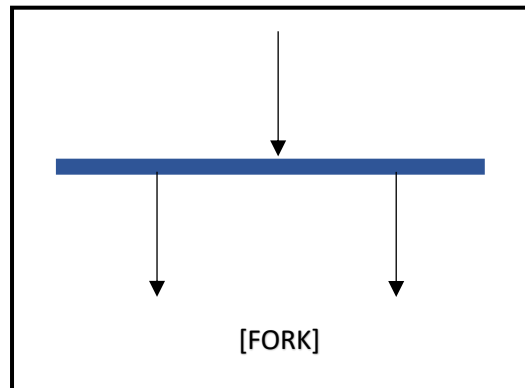
- Control flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

- **DECISION NODE :**



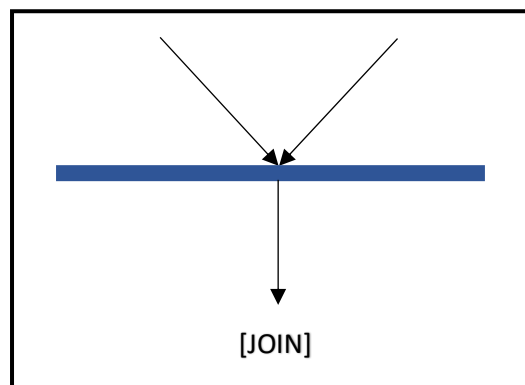
- A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

➤ **FORK:**



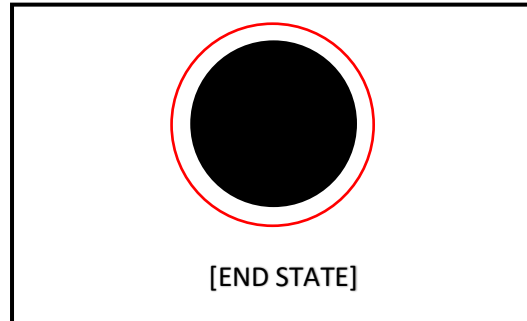
- Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join.

• **JOIN :**



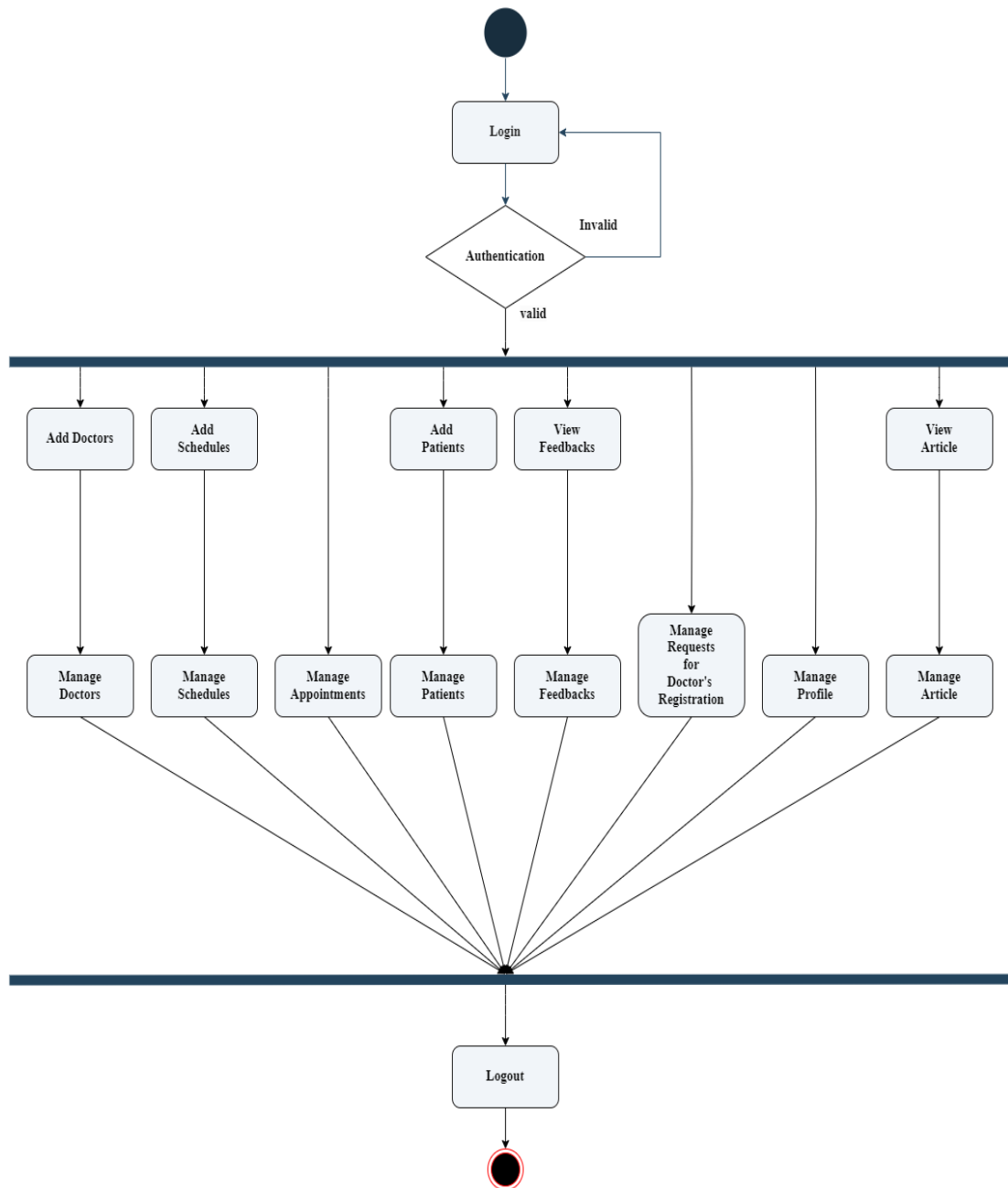
- Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.

- **END STATE:**



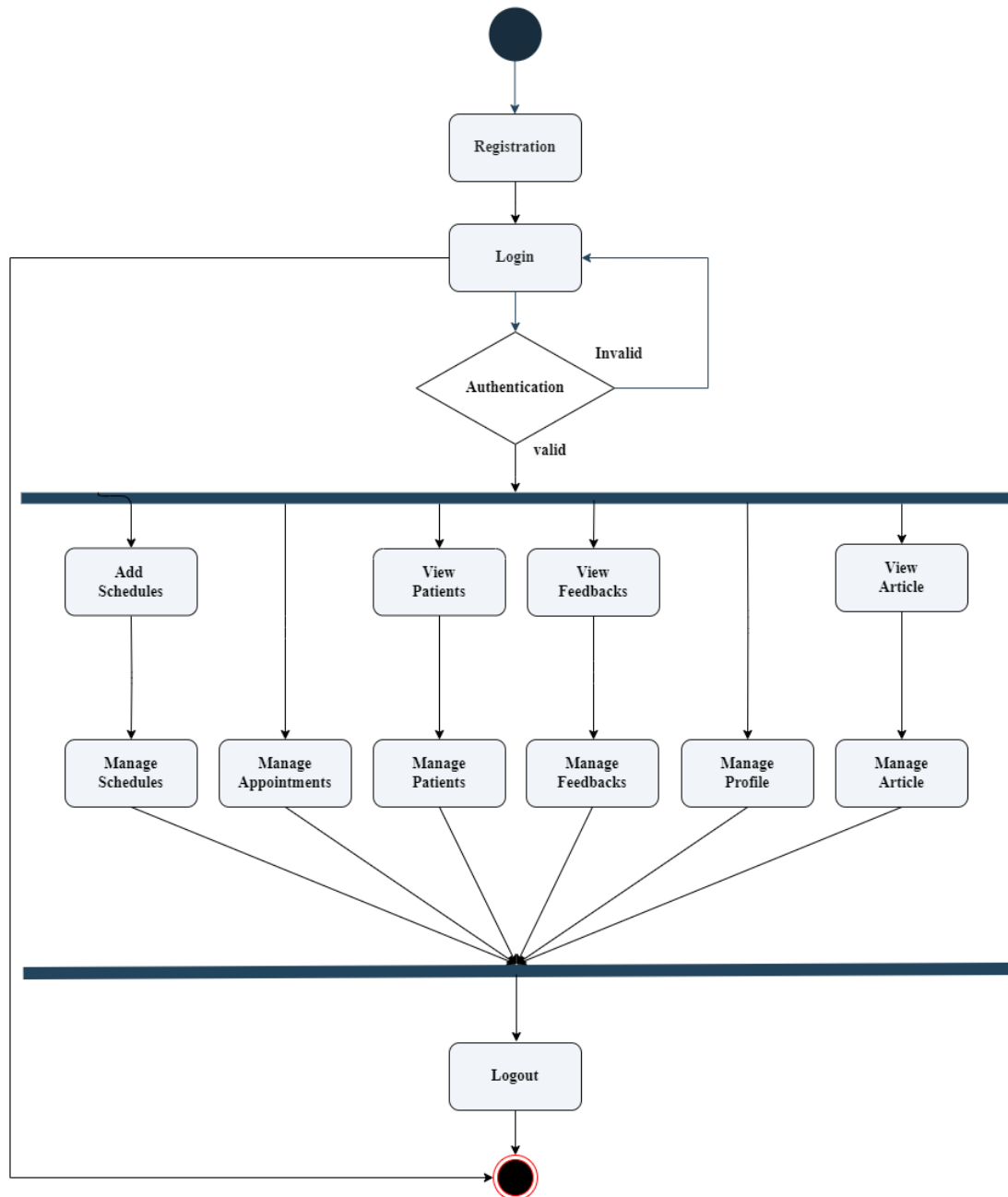
Marks the end state of an activity and represents the completion of all flows of a process.

❖ Activity Diagram – Admin



[Figure-4.4.3.2.1 Activity Diagram – Admin]

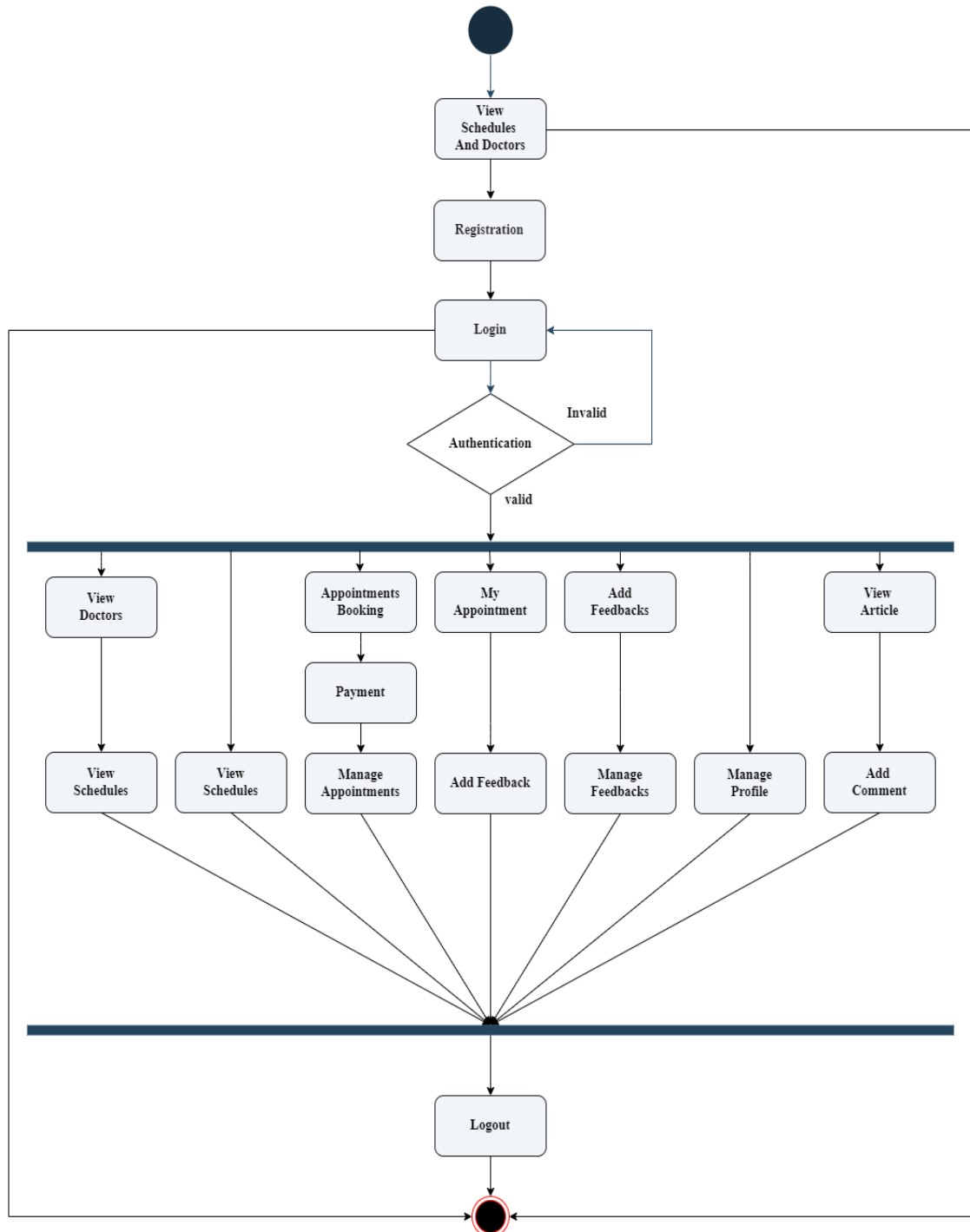
**❖ Activity Diagram – Doctor**



**[Figure-4.4.3.2 Activity Diagram – Doctor]**



**❖ Activity Diagram – Patient**



**[Figure-4.4.3.2.3 Activity Diagram – Patient]**


## **4.5 Input /Output Design:**

### **4.5.1 Patient Registration Page**

←

PatientDoctor

### Patient Registration



Email

Password

Re-Enter Password

Name

Phone NO

Gender  

Male▼

Date Of Birth  

mm/dd/yyyy📅

Address

Register

Reset


This is Patient registration page of E-Health Care Management System. This Page Is Use TO Register New Patient At Run Time When You Successfully Create Your Account You Immediately Get Access Of All Functionality That Provide To Other Patient.

### 4.5.2 Doctor Registration Page

←

Patient

Doctor



### Doctor Registration

Email

Name

Password

Re-Enter Password

Phone No

Charge

Gender

Specialist

Date of birth

Experience

About

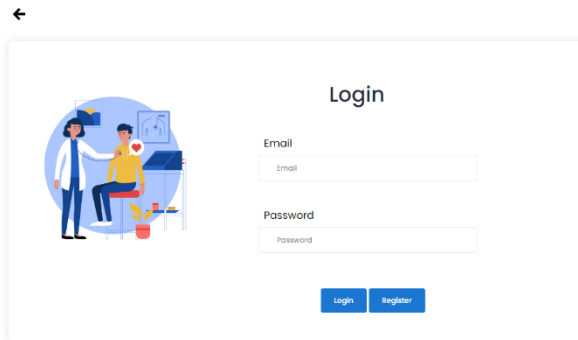
Address

This is Doctor registration page of E-Health Care Management System.

# Shree Swaminarayan Collage Of Computer Science

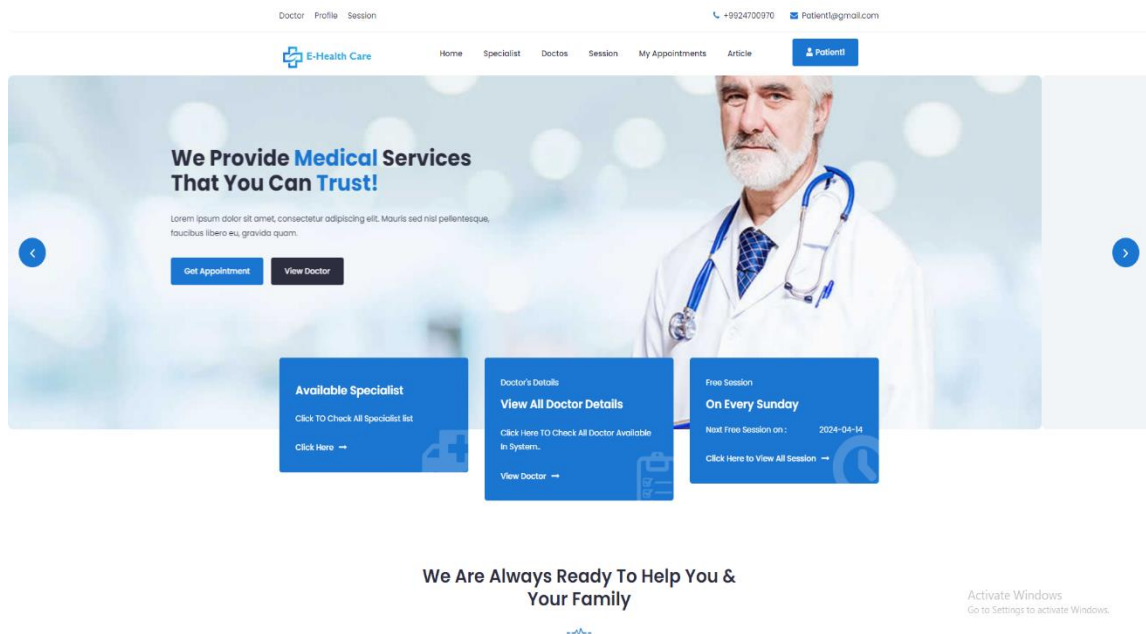
## Project Title : E-Health Care Management System

### 4.5.3 Login Page



This is login page of E-Health Care Management System.

### 4.5.4 Patient Home Page

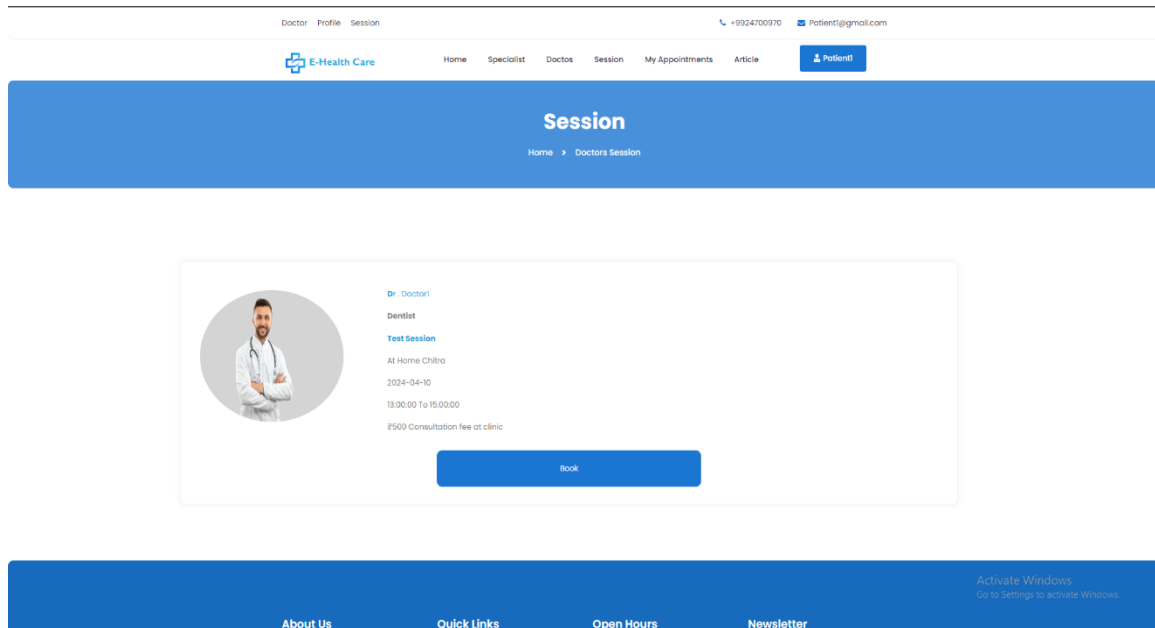


This is Home page of E-Health Care Management System on Patient side.

# Shree Swaminarayan Collage Of Computer Science

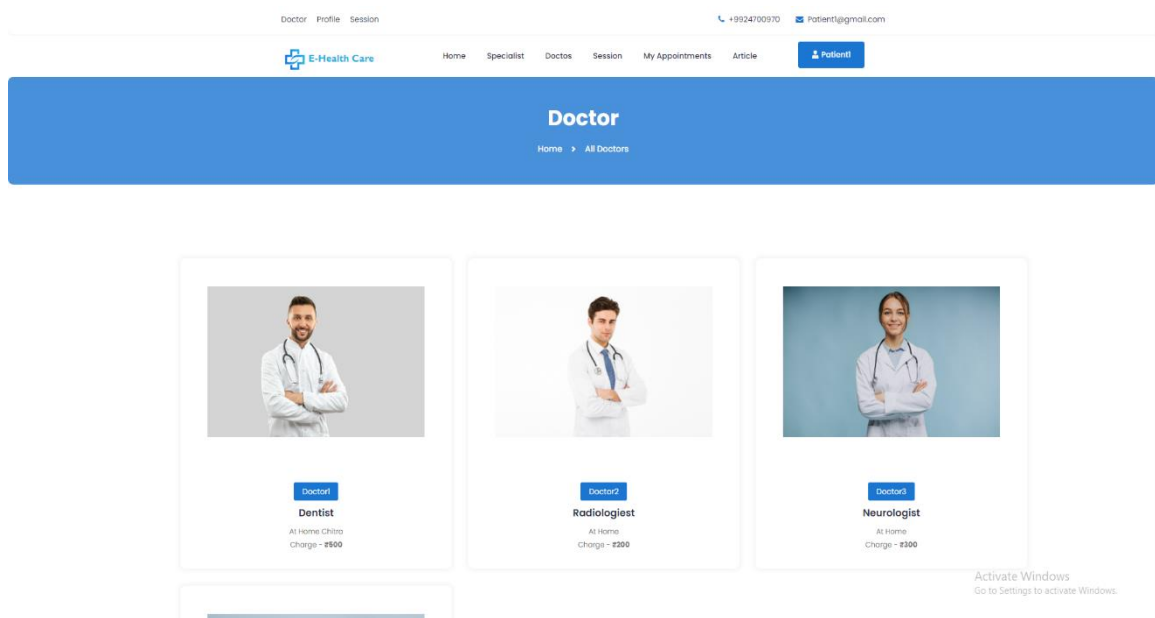
## Project Title : E-Health Care Management System

### 4.5.5 Available Session Page



This is Session page of E-Health Care Management System on Patient side.

### 4.5.6 Doctors Page

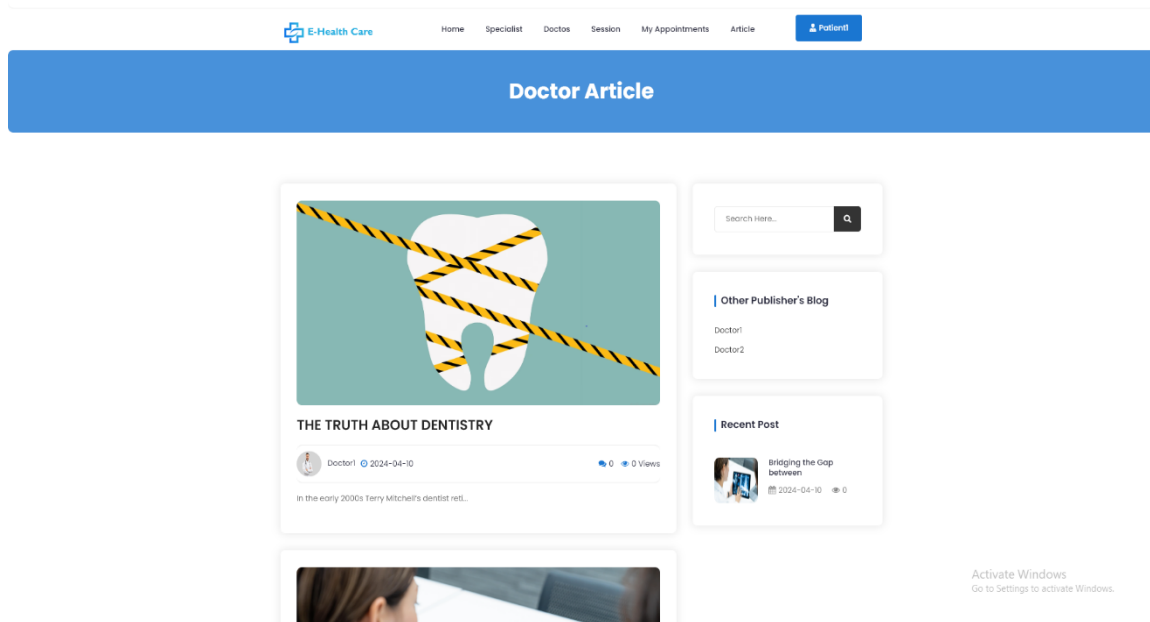


This is Doctors page of E-Health Care Management System on Patient side.

# Shree Swaminarayan Collage Of Computer Science

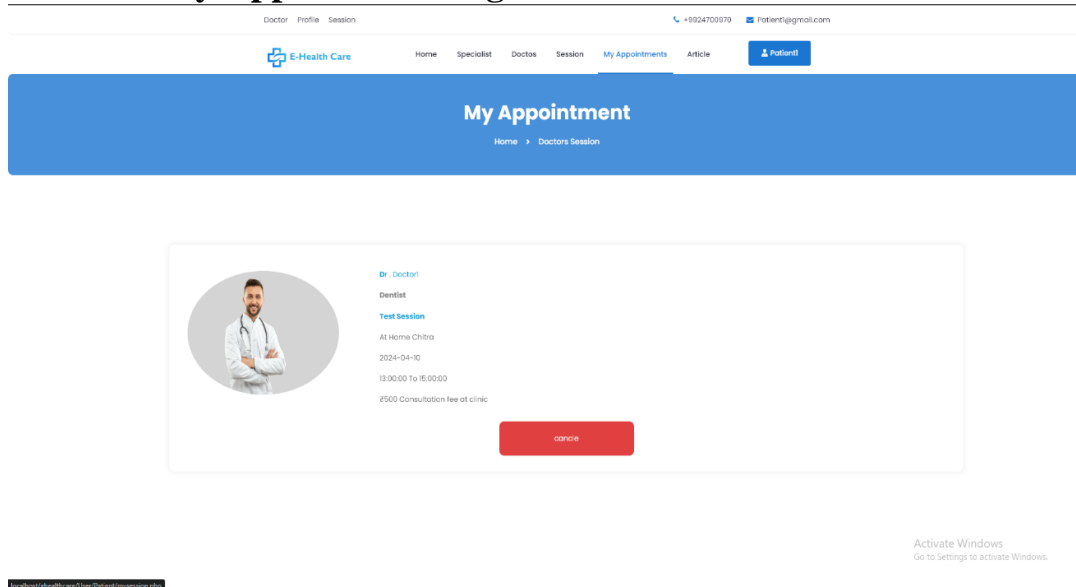
## Project Title : E-Health Care Management System

### 4.5.7 Article Page



This is Article page of E-Health Care Management System on Patient side.

### 4.5.8 My Appointment Page

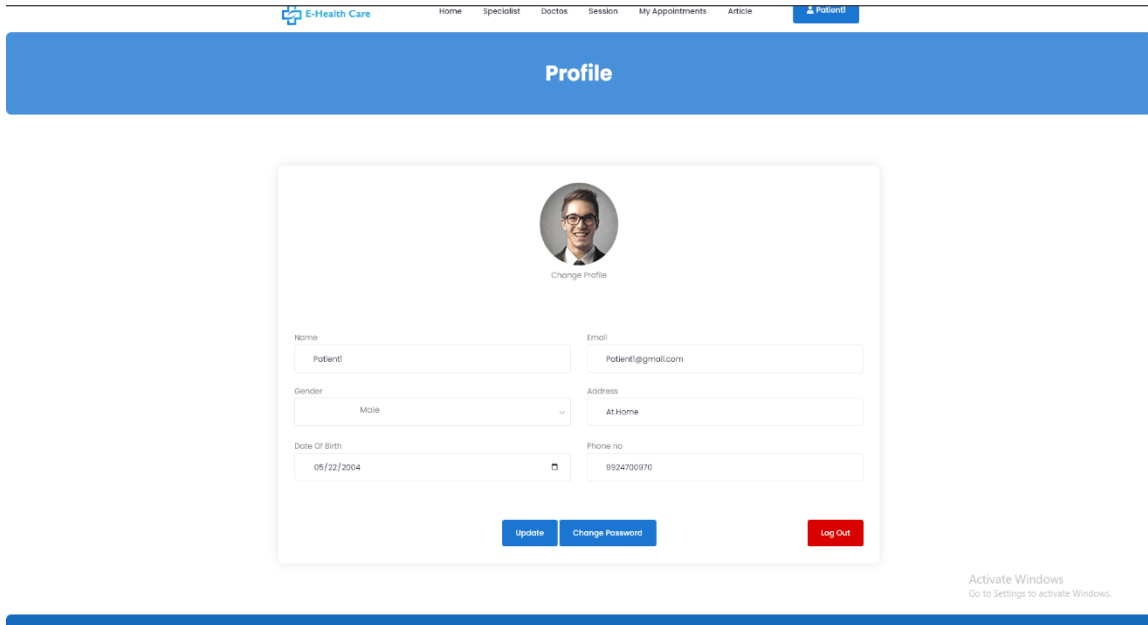


This is My Appointment page of E-Health Care Management System on Patient side.

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

### 4.5.9 Profile Page



The screenshot shows the 'Profile' page of the E-Health Care Management System. The page has a blue header with the system name and navigation links. The main content area is white and contains a profile card. The card features a circular profile picture of a man with glasses, a 'Change Profile' link, and a form with fields for Name, Email, Gender, Address, Date Of Birth, and Phone no. Below the form are 'Update', 'Change Password', and 'Log Out' buttons. A footer bar contains 'About Us', 'Quick Links', 'Open Hours', 'Newsletter', and an 'Activate Windows' watermark.

E-Health Care Home Specialist Doctors Session My Appointments Article Patient1

### Profile

Change Profile

Name Patient1 Email Patient1@gmail.com

Gender Male Address At Home

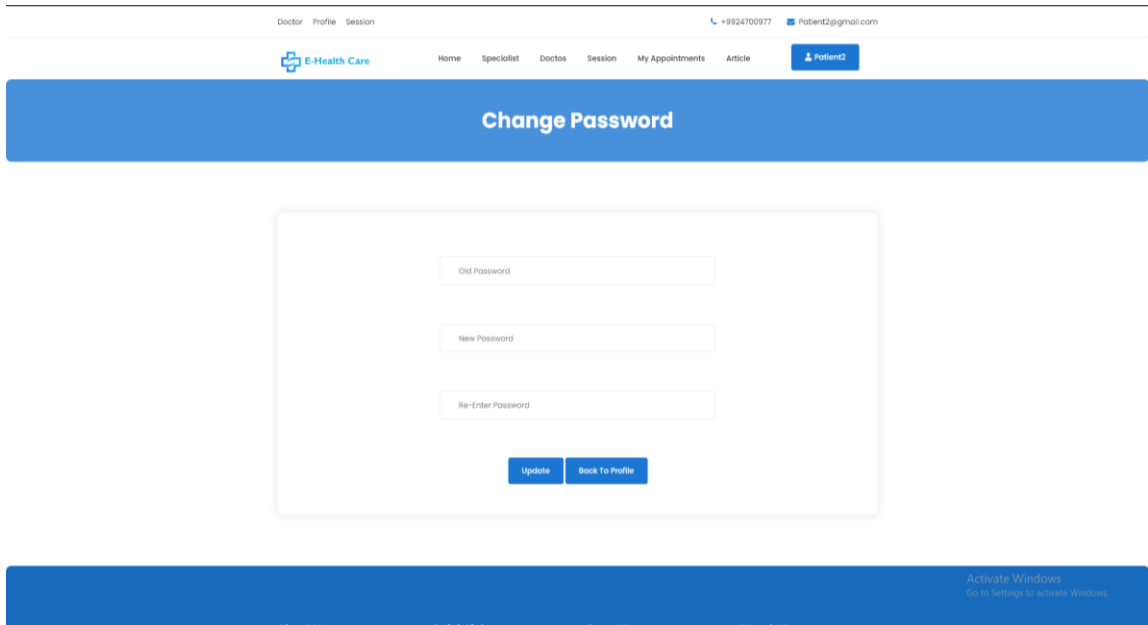
Date Of Birth 05/22/2004 Phone no 9924700976

Update Change Password Log Out

Activate Windows  
Go to Settings to activate Windows.

This is Profile page of E-Health Care Management System on Patient side

### 4.5.10 Change Password Page



The screenshot shows the 'Change Password' page of the E-Health Care Management System. The page has a blue header with the system name and navigation links. The main content area is white and contains a form with fields for Old Password, New Password, and Re-Enter Password. Below the form are 'Update' and 'Back To Profile' buttons. A footer bar contains 'About Us', 'Quick Links', 'Open Hours', 'Newsletter', and an 'Activate Windows' watermark.

Doctor Profile Session +9924700977 Patient2@gmail.com

E-Health Care Home Specialist Doctors Session My Appointments Article Patient2

### Change Password

Old Password

New Password

Re-Enter Password

Update Back To Profile

Activate Windows  
Go to Settings to activate Windows.

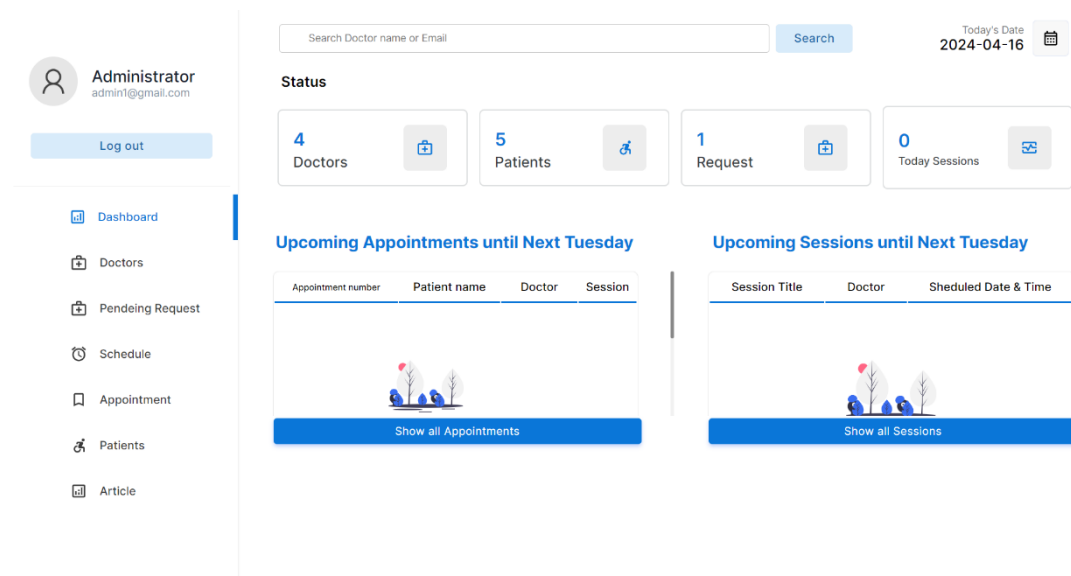
About Us Quick Links Open Hours Newsletter

This is Change Password page of E-Health Care Management System on Patient side

# Shree Swaminarayan Collage Of Computer Science

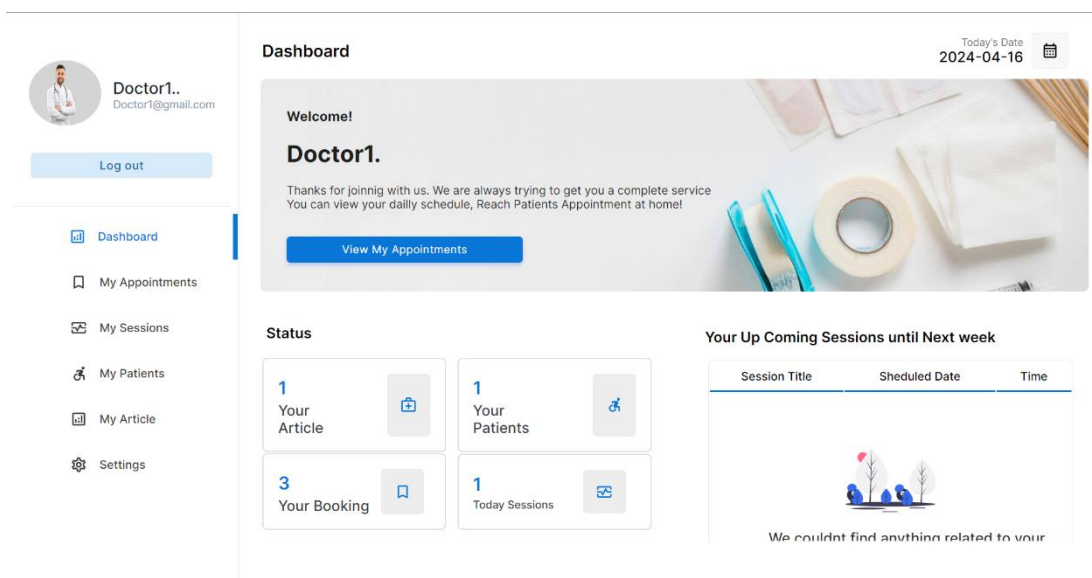
## Project Title : E-Health Care Management System

### 4.5.11 Admin Panel



This Is Admin Panel Dashboard Of E-Health Care Management System.

### 4.5.12 Doctor Panel



This Is Doctor Panel Dashboard Of E-Health Care Management System.



## **TESTING & IMPLEMENTATION**

## **5.1 Introduction of Testing:**

- Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.
- It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.
- The purpose of software testing is to identify errors, gaps or missing equirements in contrast to actual requirements.

### **Following are the characteristic that process the testing templates:**

- The developer should conduct the successful technical reviews to perform the testing successful.
- Testing starts with the component level and work from outside toward the integration of the whole computer-based system.
- Different testing techniques are suitable at different point in time.
- Testing is organized by the developer of the software and by an independent test group.
- Debugging and testing are different activities, then also the debugging should be accommodated in any strategy of testing.

## 5.2 Importance of Testing:

- Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product.
- Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.
- Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss.

## 5.3 Benefits of Testing:

- **Cost-Effective:** It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
- **Security:** It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
- **Product quality:** It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
- **Customer Satisfaction:** The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience

## 5.4 Testing Plan:

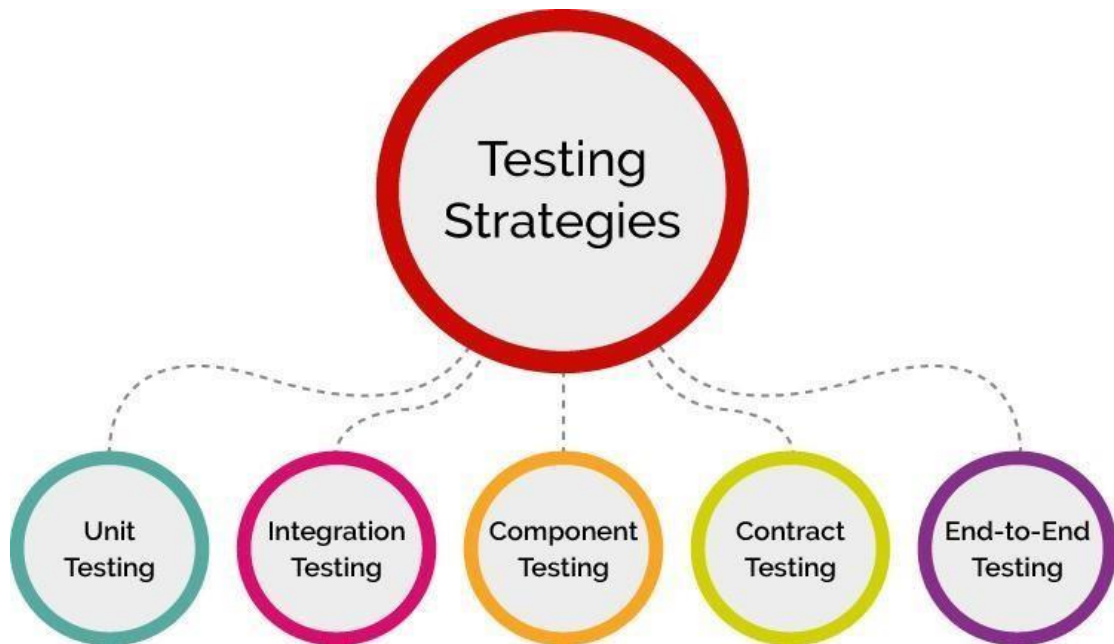
The main Objective of doing testing is to identify all defects existing in software. Basically, the testing of software consists of providing the program with a set of test inputs (test case) and observing that whether the software behaves as expected.

Testing is the process of executing a program with the explicit intention of finding errors, which makes the program fail. The tester is actually trying to make the program fail. A successful test is the one that finds errors.

Regardless of which strategies the analysts follow, there are preferred practices to ensure that the testing is useful. The levels of tests and practices to ensure that testing is useful. The levels of tests and types of test data, combined with testing libraries are important aspects of test process. Among the various testing practices or strategies that are followed by analysts, the two important ones are unit testing and system testing.

Software testing is crucial step in determining whether a software application is viable, ready for market and free of bugs. No software will be completely free of glitches but through software testing can and will make sure that it is as error free as humanly possible.

### 5.5 Testing Strategies:



**[Figure-5.5 Testing Strategies]**

- **UNIT TESTING :**

- Tests a single class or a set of closely coupled classes.
- These unit tests can either be run using the actual objects that the unit interacts with or by employing the use of test doubles or mocks.
- In Unit testing, the smallest piece of testable software is tested in the application to determine whether it behaves as expected or not.
- Tests are typically run at the class level or around a small group of related classes. In unit testing, an important distinction is seen based on whether or not the unit under test is isolated from its collaborators.

- Unit tests are usually written by the programmers using their regular tools. The only difference being the use of the same sort of unit testing framework.

### **There are further two types of testing in Unit Testing:**

#### **i. Sociable Unit Testing:**

It focuses on testing the behaviour of modules by observing changes in their state. This treats the unit under test as a black box tested entirely through its interface.

#### **ii. Solitary Unit Testing:**

It looks at the interactions and collaborations between an object and its dependencies, which are replaced by test doubles.

### **• INTEGRATION TESTING:**

- Integration tests are used to test communication between services. These tests are designed to test basic success and error paths over a network boundary.
- Different components interact with each other for their functional dependency, while communicating with each other integration test verifies the communication paths between the components and detect the interface defects.
- Here, all test modules are integrated together and tested as a subsystem. It checks that the communication paths between the subsystem work correctly while interacting with its peers. In micro service architecture, they are typically used to verify interactions between layers of integration code and the external components to which they are integrating.

- When the automated tests are written for the modules which are interacting with an external component, the basic goal is to verify the modules are interacting sufficiently with the external component.
  - It is very difficult to trigger abnormal behaviour such as a timeout or slow responses from the external component. Special tests are written to ensure that test respond as expected in the unexpected circumstances
  - **Persistence integration tests** provide assurances that the schema assumed by the code matches that is available in the data store.
  - With unit testing and integration testing, we can have confidence in the correctness of the logic contained in the individual modules that make up the micro service, but we cannot be sure that the micro services work together as a whole to satisfy business requirements.
- 
- **COMPONENT TESTING:**
    - Tests the full function of a single micro service. During this type of testing, any calls to external services are mocked in some way.
    - A component is any well-encapsulated, coherent and independently replaceable part of a larger system. In a micro service architecture, the components are the services themselves.
    - A component's complex behaviour is avoided by isolating it from its peers, also isolation help in controlling the test environment for the component.
  - **CONTRACT TESING:**
    - Test the agreed contract for APIs and other resources that are provided by the micro service.

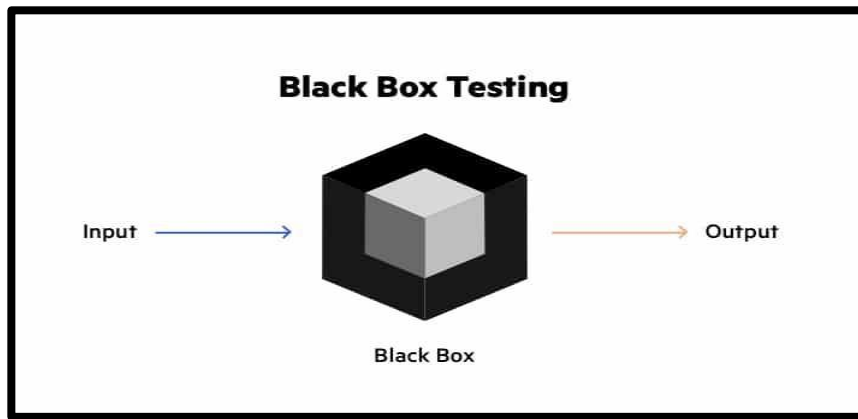
- At the boundary of the external service, an Integrated contract test is done to verify the contract that is expected by the consuming service. This test verifies that the component meets a contract.
  - A test suite is written to verify only those aspects of the producing service that is in use. The behaviour of the service is not deeply tested, response latency and throughput should be within acceptable limits when input and output of the service call contain required attributes. This test is written by each test consuming team and then packaged. The main aim of this test is to know the impact of the changes made by the maintainers on the consumers.
- **END TO END TESTING:**
    - End-to-End Testing, Tests a complete flow through the application or micro service. Usually used to test a golden path or to verify that the application meets external requirements.
    - End-to-End testing tests the whole system from end to end. It verifies that the entire system meets the external requirements and eventually achieve its goal. Without bothering about the internal architecture of the application business goal should be achieved by the End-to-End testing.
    - The system is fully deployed and is treated as a black box and the test is exercised. With Public interference through GUIs and API, the system is manipulated. End to End Tests is more business facing.
    - This test verifies that the firewall, proxies, and load balancers are correctly configured.

In micro service architecture, for one behaviour, there are many micro services which interact to respond to that behaviour, an end-to-end testing provides value by adding coverage of gaps between the system.



## **5.6 Testing Methods:**

### **I. BLACK-BOX TESTING:**



**[Figure-5.6.1 BLACK-BOX TESTING]**

- The technique of testing without having any knowledge of the interior workings of the application is called black-box testing.
- The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

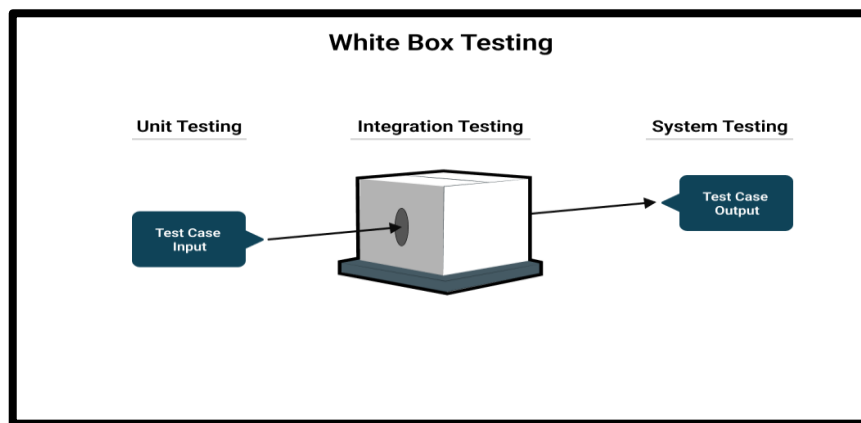
---

❖ The following table lists the advantages and disadvantages of black-box testing.

| Advantages  | Disadvantages   |
|---|---|
| Well suited and efficient for large code segments.  | Limited coverage, since only a selected number of test scenarios is actually performed.               |
| Code access is not required.  | Inefficient testing, due to the fact that the tester only has limited knowledge about an application. |
| Clearly separates user's perspective from the developer's perspective through visibly defined roles.  | Blind coverage, since the tester cannot target specific code segments or error-prone areas.           |
| Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems. | The test cases are difficult to design.   |

## II. WHITE-BOX TESTING:

- White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.



[Figure-5.6.2 WHITE-BOX TESTING]

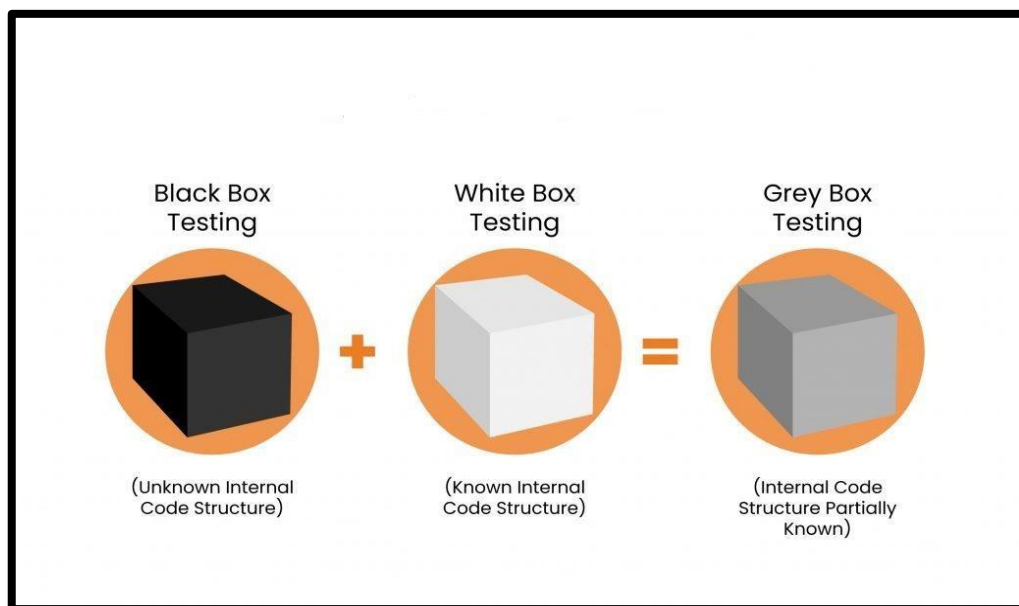
- The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

❖ The following table lists the advantage and disadvantage of white-box testing.

| Advantages   | Disadvantages  |
|--|--|
| As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively. | Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased.   |
| It helps in optimizing the code.   | Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested.<br><br>An application. |
| Extra lines of code can be removed which can bring in hidden defects.  | It is difficult to maintain white-box testing, as it requires specialized tools like code analysers and debugging tools.   |

### **III. GREY-BOX TESTING:**

- Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application.
- In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.
- Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge.
- Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database.
- Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.



**[Figure-5.6.3 GREY-BOX TESTING]**

❖ The following table lists the advantage and disadvantage of grey-box testing.

| Advantages  | Disadvantages  |
|---|--|
| Offers combined benefits of black-box and white-box testing wherever possible.  | Since the access to source code is not available, the ability to go over the code and test coverage is limited.  |
| Grey box testers don't rely on the source code; instead, they rely on interface definition and functional specifications.   | The tests can be redundant if the software designer has already run a test case.   |
| Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling. | Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested. |

## 5.7 Test Case :

- 1<sup>st</sup> Test Case :

| TEST CASE ID    | 1-A  |
|-----------------|--|
| Name            | Login Validation   |
| Test Scenario   | It Will Check Login Data Of User If Username And Password Both Are True      |
| Test Step       | 1.Open Web-site<br>2.Click On “Log In”<br>3.Enter Detail<br>4.Click On Login |
| Test Data       | Email: admin1@gmail.com<br>Password: admin123                                |
| Expected Result | User Login As Admin Of System  |
| Actual Result   | As Expected  |
| Pass / Fail     | Pass   |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **2<sup>nd</sup> Test Case:**

| TEST CASE ID    | 1-B   |
|-----------------|---|
| Name            | Login Validation  |
| Test Scenario   | It Will Check Login Data Of User If Username And Password Both Or Either One Of Them Is False |
| Test Step       | 1.Open Web-site<br>2.Click On “Log In”<br>3.Enter Detail<br>4.Click On Login                  |
| Test Data       | Email: admin1@gmail.com<br>Password: admin  |
| Expected Result | Message: Wrong username Or Password !!  |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |



- **3<sup>rd</sup> Test Case:**

| TEST CASE ID    | 1-C  |
|-----------------|--|
| Name            | Login Validation   |
| Test Scenario   | It Will Check Login Data Of User If Username And Password Both or One Of Them Is Empty |
| Test Step       | 1.Open Web-site<br>2.Click On “Log In”<br>3.Enter Detail<br>4.Click On Login           |
| Test Data       | Email: ...<br>Password: ...  |
| Expected Result | Message: Enter Username Or Password First!!  |
| Actual Result   | As Expected  |
| Pass / Fail     | Pass   |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **4<sup>th</sup> Test Case:**

| TEST CASE ID    | 2-A   |
|-----------------|---|
| Name            | Patient Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Patient Account  |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Patient ”<br>3.Enter Detail<br>4.Click On Register   |
| Test Data       | Email : Patient1@gmail.com<br>Password : 123<br>Repeat Password : 123<br>Name : Patient1<br>Mobile No : 8460888834<br>Gender : Male(Radio Button)<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Address : Chitra Bhavnagar |
| Expected Result | Message : Account Created Successfully !!   |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **5<sup>th</sup> Test Case:**

| TEST CASE ID    | 2-B   |
|-----------------|---|
| Name            | Patient Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Patient Account And Check For Already Exist Email.   |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Patient ”<br>3.Enter Detail<br>4.Click On Register   |
| Test Data       | Email : Patient1@gmail.com<br>Password : 123<br>Repeat Password : 123<br>Name : Patient2<br>Mobile No : 8460888835<br>Gender : Male(Radio Button)<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Address : Chitra Bhavnagar |
| Expected Result | Message : Email Already Exist !!  |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |

- **6<sup>th</sup> Test Case:**

| TEST CASE ID    | 2-C   |
|-----------------|---|
| Name            | Patient Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Patient Account And Check For Already Exist Mobile No.   |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Patient ”<br>3.Enter Detail<br>4.Click On Register   |
| Test Data       | Email : Patient2@gmail.com<br>Password : 123<br>Repeat Password : 123<br>Name : Patient2<br>Mobile No : 8460888834<br>Gender : Male(Radio Button)<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Address : Chitra Bhavnagar |
| Expected Result | Message : Mobile No Already Exist !!  |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |

- **7<sup>th</sup> Test Case:**

| TEST CASE ID    | 2-D   |
|-----------------|---|
| Name            | Patient Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Patient Account And Check All Filed Must Have Data                       |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Patient ”<br>3.Enter Detail<br>4.Click On Register   |
| Test Data       | Email : ...<br>Password : ...<br>Repeat Password : ...<br>Name : ...<br>Mobile No : ...<br>Gender : ...<br>DOB : ...<br>Address : ... |
| Expected Result | Message : All Detail Must Be Filed !!   |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **8<sup>th</sup> Test Case:**

| TEST CASE ID    | 3-A  |
|-----------------|--|
| Name            | Doctor Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Doctor Account  |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Doctor”<br>3.Enter Detail<br>4.Click On Register  |
| Test Data       | Email : Doctor1@gmail.com<br>Name : Doctor1<br>Password : Doctor123<br>Repeat Password : Docto123<br>Mobile No : 9924700970<br>Charge : 500<br>Gender : Male(Radio Button)<br>Specialist : Cardiologists<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Experience : 10[Years]<br>About :I’m Doctor 1<br>Address : At Home |
| Expected Result | Message : Doctor Details Add For Inquiry !!  |
| Actual Result   | As Expected  |
| Pass / Fail     | Pass   |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **9<sup>th</sup> Test Case:**

| TEST CASE ID    | 3-B  |
|-----------------|--|
| Name            | Doctor Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Doctor Account And Check For Already Exist Email.   |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Doctor”<br>3.Enter Detail<br>4.Click On Register  |
| Test Data       | Email : Doctor1@gmail.com<br>Name : Doctor1<br>Password : Doctor123<br>Repeat Password : Docto123<br>Mobile No : 9924700977<br>Charge : 500<br>Gender : Male(Radio Button)<br>Specialist : Cardiologists<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Experience : 10[Years]<br>About :I’m Doctor 1<br>Address : At Home |
| Expected Result | Message : User With This Email Is Already Exist !!   |
| Actual Result   | As Expected  |
| Pass / Fail     | Pass   |

# Shree Swaminarayan Collage Of Computer Science

## Project Title : E-Health Care Management System

---

- **10<sup>th</sup> Test Case:**

| TEST CASE ID    | 3-C  |
|-----------------|--|
| Name            | Doctor Registration Validation   |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Doctor Account And Check For Already Exist Mobile No.   |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Doctor”<br>3.Enter Detail<br>4.Click On Register  |
| Test Data       | Email : Doctor2@gmail.com<br>Name : Doctor3<br>Password : Doctor123<br>Repeat Password : Docto123<br>Mobile No : 9924700970<br>Charge : 500<br>Gender : Male(Radio Button)<br>Specialist : Cardiologists<br>DOB : 05/22/2004 [ MM/DD/YYYY]<br>Experience : 10[Years]<br>About :I’m Doctor 1<br>Address : At Home |
| Expected Result | Message : User With This Mobile Number Already Exist !!  |
| Actual Result   | As Expected  |
| Pass / Fail     | Pass   |



- **11<sup>th</sup> Test Case:**

| TEST CASE ID    | 3-D   |
|-----------------|---|
| Name            | Doctor Registration Validation  |
| Test Scenario   | It Will Check All Data Of Filed That Enter By user To Create Doctor Account And Check All Filed Must Have Data  |
| Test Step       | 1.Open Web-site<br>2.Click On “Register As Doctor”<br>3.Enter Detail<br>4.Click On Register   |
| Test Data       | Email : ...<br>Name : ...<br>Password : ...<br>Repeat Password : ...<br>Mobile No : ...<br>Charge : ...<br>Gender : ...<br>Specialist : -Specialist-<br>DOB : ...<br>Experience : ...<br>About : ...<br>Address : ... |
| Expected Result | Message : All Detail Must Be Filed !!   |
| Actual Result   | As Expected   |
| Pass / Fail     | Pass  |

## 5.8 Implementation Approach :

During this phase, the newly developed E-Healthcare Management System is installed and put into operation, transitioning data and components from the old system to the new one through a direct cut-over. Both system analysts and end-users witness the realization of implemented changes.

### ACTIVITIES :

- Installation of the Software: The E-Healthcare Management System software is installed on the designated servers and workstations.
- Software Updates: Necessary updates and patches are applied to ensure the system operates efficiently and securely.
- User Training: End-users receive comprehensive training to familiarize themselves with the functionalities and workflows of the new system.

Upon completion of the implementation phase, the system's performance is evaluated based on the requirements outlined during the definition phase and compared against the design specifications. The phase concludes when all requirements are successfully met, and the system's performance aligns with the intended design.

This phase marks a crucial milestone in the development and deployment of the E-Healthcare Management System, ensuring that it is ready for full-scale operation and effectively meets the needs of healthcare providers and patients alike.

## **CONCLUSION**

## 6.1 Conclusion :

In conclusion, the implementation of an **E- Health Care Management System** represents a significant advancement in the healthcare industry. It not only streamlines the management of patient data and health records but also enhances the accessibility and quality of care provided to patients. By leveraging technology, such systems can offer real-time updates, reduce errors, and facilitate better communication among healthcare professionals

Ultimately, an **E-Health Care Management System** is a step towards a more efficient, integrated, and patient-centered healthcare ecosystem, promising improved outcomes for patients and a more manageable workload for healthcare providers.

As technology continues to evolve, it is imperative that these systems adapt and incorporate new innovations to meet the ever-changing needs of the healthcare sector.

So, this knowledge that we gain from this project is like this. The work that we have done will remain in our minds forever. The knowledge that we have gained is invaluable.

The project was started as our learning experience which really helped us as to how deal with real world solution. This was new but wonderful experience of professional environment and many of the tools used.

## 6.2 Limitation of System :

### 6.2.1 User training and adoption:

Healthcare providers and staff need to be trained on how to use the e-healthcare management system properly. Resistance to change and lack of training can affect the system's adoption and utilization.

### 6.2.2 Limited access for certain populations:

Some patients may not have access to the technology needed to use e-healthcare management systems, leading to disparities in healthcare access and quality.

### **6.3 Future Scope of System:**

#### **1. Scaling up the Platform:**

Our platform's expansion strategy aims to extend service from select cities to entire states and counties. This scale-up will involve infrastructure upgrades, resource optimization, and strategic market analysis to meet regional demands.

We'll focus on enhancing server and network capacities, localizing content, and partnering with key local entities. This comprehensive approach will position our platform as an accessible solution for a broader audience, amplifying its reach and impact.

#### **2. Implementation of Virtual Doctor Consultations:**

In response to telemedicine's rising demand, we're set to launch remote doctor-patient consultations. Key integrations include video conferencing, secure channels, and EHR systems for uninterrupted interactions.

Doctors will get a streamlined interface for managing appointments and patient data, while patients will enjoy an easy-to-use platform for scheduling and secure medical sharing. Adhering to healthcare regulations and privacy standards is crucial.

This telemedicine initiative promises to improve healthcare accessibility and patient outcomes, marking a shift towards a more patient-centric healthcare model.

## **6.4 Bibliography:**

We took references from below describe listing websites to build  
successful web Application

**Youtube** : <https://www.youtube.com/>

**Google** : <https://www.google.com/>

**ChatGTP** : <https://chat.openai.com/>

**Fortis Healthcare** : <https://www.fortishealthcare.com/>

### **BOOKS:**

1 . Programming using PHP

2 . MySQL/PHP database application