

## 1. INTRODUCTION

### 1.1.ABOUT THE PROJECT

In the present era, the importance of patient health tracker sites are need day by day. Patients needs a simple interface .Patient Health Tracker is a website that provides distant communication accessible to doctors and patients.It deals with the follow-up of the patients by their respective doctors.Every organization whether big or small has challenges to overcome and manage the information of patient , medicines , doctors , treatments , reports. Every patient has different needs,therefore book doctors for their own needs.For a doctor,it acts as an analytical tool covering the complete details of various diseases that have come across,their respective symptoms ,medicines used for curing,level of progress ,and side effects if any.The need of healthcare services is growing with the increase in population and number of patients.In this system,the main objective is to consult the patients and manages the detail of patient. Here, shows the information and description of the patient,tests.Increase efficiency of managing the patient and manage the information of patient.The main control over the system is Admin.Admin has the only power to approve doctors in this system.There are several doctors who are specialized in various categories.Details of the doctor needs at the time of registration.Add,Edit and View the doctors specialization.Patients can register with their details and at the first time of registering there must be a registration fee for patient.if a patient searching for multiple doctors with different in specialization there must be additional registration fees.Booking is confirmed only when payment is completed.When payment is completed,provides tokens.Tokens are provided by the doctors.There is a limit in number of tokens a day,which means there is limit of patients in a day.Payment mode preferred through Debit/Credit card.

Front End: HTML,CSS,JAVASCRIPT,PHP  
Back End: SQL

## **2. REQUIREMENT ANALYSIS AND SPECIFICATION**

### **2.1. SYSTEM STUDY**

The Patient Health Tracker System manages the Operations of a Hospital. We can store the details of every operations that occurs in the system.

System study refers to the process of examining a situation with the intent of improving it through better process and methods. System study is, therefore the process of gathering and interpreting facts, diagnosing problem and using the information to recommend changes in the system, in other words it means a detailed explanation of description. Before computerizing a system under consideration, it must be analyzed. We need to study how it functions currently, what are problems and what are the requirements that the proposed software should meet.

The main components of making software are:

- System and software requirements analysis
- Design and implementation of software
- Ensuring, verifying and maintaining software integrity

The Patient health tracker provides user friendly interface which can be operated by anyone with little knowledge about the computer system. It stores the information needed in a database which can be accessed by the admin, doctor and patient. It should maintain a well organized database for storing the information. This helps to eliminate the storage of invalid data.

### **2.1.1. Existing System**

Existing system is based on manual work and all the process are done manually, So they maintain registers and files for recording all details of the system. They maintain several registers for recording the entry of daily transactions. They keep the bill book or cash book to maintain the record or transaction. In current system billing is manual and time consuming. The existing system is not very convenient as the patient has to spend lot of time and resources to go to hospital. It is tedious and time consuming process. Existing system was a manual system. There are also frequent chances of data redundancy and data is easy to lose.

#### **Disadvantages of Existing system**

- Time Consuming
- Editing of data becomes a tedious job
- Proper Generation of Report

### **2.1.2. Proposed System**

The proposed system is interactive, highly user friendly and designed exclusively for the Patient Health Tracker System. The system covers almost all the functional areas. The Patient Health Tracker System is a database system used to store the information regarding patient details, doctor details etc.

All the operations and activities related to The Patient Health Tracker System can be carried out efficiently. The objectives of the proposed system are to overcome the major limitation of existing system enabling effective management of the patient details thereby improving the performance of administration. The patient health tracker is a suitable one. Comparison of doctors can be done more easily in patient health tracker. The Patient can log into the website and find their doctors with their needs. For the ease of patients, the website can be accessed 24\*7. It is more efficient and reliable than the existing system. This system understands the needs of the patient and everything is available at the doorstep.

#### **Advantages of Proposed System**

- High processing speed
- Easy to retrieve old records by using search feature
- Minimal errors
- Greater portability

### **2.1.3. Feasibility Study**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, time and effort spent on it. Feasibility study lets the developer foresee the future of the project and its usefulness. Finding out whether a new system is required or not. The study is carried out to the best system that meet performance requirement. This entails identification, description and evaluation of candidate system and selection of the best system for the best system for the job. It simply identifies whether the proposed system is feasible to the organization or not.

There are three aspects in the feasibility study portion of the preliminary investigation

- Technical feasibility
- Economic feasibility
- Operational feasibility

#### **2.1.3.1. Technical Feasibility**

The system must be evaluated from technical viewpoint first. The assessment of this feasibility must be based on outline design of the system requirement in the terms of input, output, programs and procedure having identified an outline system, the investigation must go on to suggest the type of equipment, required method of developing the system, method of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed with latest technology. There are only minimal constraints involved in this project.

**2.1.3.2. Economic Feasibility**

Here an evaluation of development cost weighted against the ultimate income or benefit derived from the developed system. The cost for the development of the project has been evaluated and we want to check that the cost does not exceed beneficial cost of the system. The economic and financial analysis is used for evaluating the effectiveness of the candidate system. This project also undergone economic feasibility study and found that it is feasible. So, the cost for development does not exceed its beneficial cost. This brought to as the conclusion that the system is economically feasible in the context.

**2.1.3.3. Operational Feasibility**

In operational feasibility the entire application is checked whether the system will be used if it is developed and implemented. Also, it is checked whether there will be resistance from user that may undermine the possible application benefits. There is no barrier for implementing the system. The system also helps to access the information immediately as need arises. Thus, the system is found to be operational feasible.

## **2.2.USER CHARACTERISTICS**

The Patient Health Tracker System provides the user to perform their task in an easy and much less complex way to avoid redundancy. This system ensures that the users assessing the system can ensure maximum efficiency and they can depend on the system for desired results.

There are three user characteristics in this system:

- Admin
- Doctor
- Patient

### **2.2.1. Admin**

Administrator or Admin is the super user and main controller of this system. Administrator controls all the activities of the Patient Health Tracker. Admin can add and view the doctors and patients. He/ She is the primary user in the system. He/ She can add, edit, view and delete the details of patients and doctors. Admin can view and can also generate several reports.

### **2.2.2. Doctor**

Doctors are those people who trained to help the people who are affected by any diseases. A doctor holds a very important place in our society. There are different fields of doctors to treat various illness of people. Doctors not only treat diseases, but also advice patients about diet, health. There are many specialist doctors in a specific field.

### **2.2.3. Patient**

Patient can perform various activities like registering into the account, editing his/her details, viewing the doctors, searching the doctors and can make payment through online. Patients are those people who need medical supports from the doctor. A person under health care or the person may be waiting for the care called a patient. Actually a person who is receiving medical care, or who is cared by a particular doctor called a patient.

## **2.3. SYSTEM SPECIFICATION**

### **2.3.1. HARDWARE SPECIFICATION**

The selection of hardware is very important in the existence and proper working of any software. When selecting the hardware, the size and capacity requirements are also important. Below is some of the hardware that is required by the system.

|           |                                       |
|-----------|---------------------------------------|
| Processor | Intel Core i3-3220 (3.3 GHz) or above |
| RAM       | 4 GB or above                         |
| Storage   | 512 GB or above                       |
| Other     | Keyboard and Mouse                    |



### 2.3.2. SOFTWARE SPECIFICATION

|                  |                   |
|------------------|-------------------|
| Operating system | Windows7/8/8.1/10 |
| Front end        | PHP               |
| Back end         | SQL               |

### 2.3.3. ABOUT THE SOFTWARE TOOLS AND PLATFORMS

#### FRONT END TOOL

PHP: PHP is a server side scripting language. PHP is Hypertext Preprocessor. It is an open source scripting language. It is widely used all over the world. PHP is a scripting language generally used to make websites. PHP is used to connect Web Page with a database. It is a free and cross-platform language. In addition to its ability to access many types of databases, it is also essential to highlight its ability to create dynamic pages, as well as the possibility of separating the design of the content of a website.

#### Features:

- Simple and Easy
- Security
- Platform independent
- Flexibility
- Familiarity
- Error Reporting

HTML: The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.

**CSS:** Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

**JAVASCRIPT:** Javascript is often abbreviated as JS, is a programming language. Javascript is high level, often in-time compiled and multi paradigm.

#### **BACK END TOOL**

**SQL:** Is a free open source database that facilitates the effective management of the database by connecting them to the software. It is stable, reliable and the powerful solution with advanced features and advantages. Server management studio is a software application first launched with Microsoft SQL server 2005 that is used for configuring, managing and administering all components with Microsoft SQL server. The tool includes both script editors and graphical tools which work with object and features of the server.

#### **Key Capabilities of SQL**

- High Availability
- Performance and Scalability
- Security
- Manageability
- Developer Productivity
- Business Intelligence

### **3. SYSTEM MODELING**

The most creative and challenging phase of the system development is system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The first step is to determine how the output is to be produced and in what format. Second input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. The point is to choose such an environment in which we will be able to operate within a convenient and easy way. The most creative and challenging phase of the system development is system design. It provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The analyst should understand the requirements of the user and develop the system according. Design goes through the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The application program as an interface between the users and the database should be an accurate reflection of the database on the screen; hence a well analyzed and defined structure is needed. The user interface should be easy to understand and operate on for the users. The first step is to determine how the output is to be produced and in what format it has to be produced. Second, input data along with the master files have to be designed to meet the requirements of the proposed output.

The analyst must ensure that the interaction between the user and the interface is simple to understand. To ensure that everything works properly and as it has been expected, test performances have to be done upon the system functionality. Testing plays an important role in identifying any minor errors after system design and it will be corrected.

### **3.1. MODULES AND DESCRIPTIONS**

The Patient Health Tracker System helps the Patients to consult various Doctors through online and Administrator to manage the entire system easily. There are several Modules in this system. They are:

- 1.Doctor Specialization
- 2.Doctor Registration
- 3.Patient Registration
- 4.Booking Management
- 5.Prescription Management

#### **1. DOCTOR SPECIALIZATION**

Some doctors only specializes in certain diseases or injuries or may only specialized on one part of the human body. These types of doctors are called Specialists. In this system, Doctors are grouped corresponding to their specialists.

For eg: Some doctors specialize in diseases of the stomach or intestines. The Admin can Add, Edit and View the Specialization details.

#### **2. DOCTOR REGISTRATION**

This module manages the details of Doctor. Doctor can register to the website by giving his name, address, phone number, Certificates, Any identities etc... Admin only approves the Doctors who are registered. Register, Edit and View Doctor details.

### **3. PATIENT REGISTRATION**

This module performs the actions of managing the details of Patient. The admin can perform the following tasks such as add Patient details, update, view the Patient details. These details are helpful for administrators to manage the Patient. Register, Edit and View Patient details.

### **4. BOOKING MANAGEMENT**

In Booking Management, the patient books a doctor as per their needs. Booking is only confirmed when the payment process is completed. When booking is completed, patients receive a token, whether there is no token means that booking is not completed or processing. Add, Edit and View Booking details.

### **5. PRESCRIPTION MANAGEMENT**

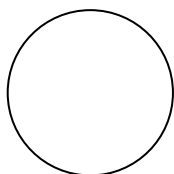
This module performs the actions of managing the prescription details. After the consultation, doctors prescribe medicines to patients. The Admin can Add, Edit and View Prescription details.

### 3.2.DATA FLOW DIAGRAM(DFD)

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

A DFD is also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

In the DFD, there are four symbols:



Process that transforms data flow



Source or Destination of data



Data store



Data flow

### **Rules for drawing data flow diagrams**

Rule 1: Establish the context of the data flow diagram by identifying all of the net input and output data flows.

Rule 2: Select a starting point for drawing the DFD.

Rule 3: Give meaningful labels to all data flow lines.

Rule 4: Label all processes with action verbs that relate input and output data flows.

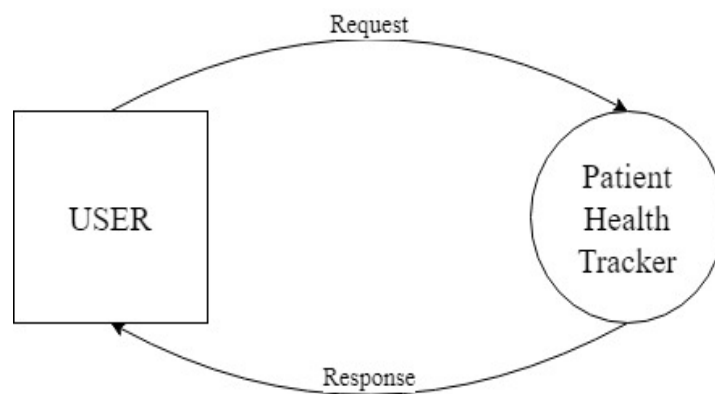
Rule 5: Omit insignificant functions routinely handled in the programming process.

Rule 6: Do not include control or flow of control information.

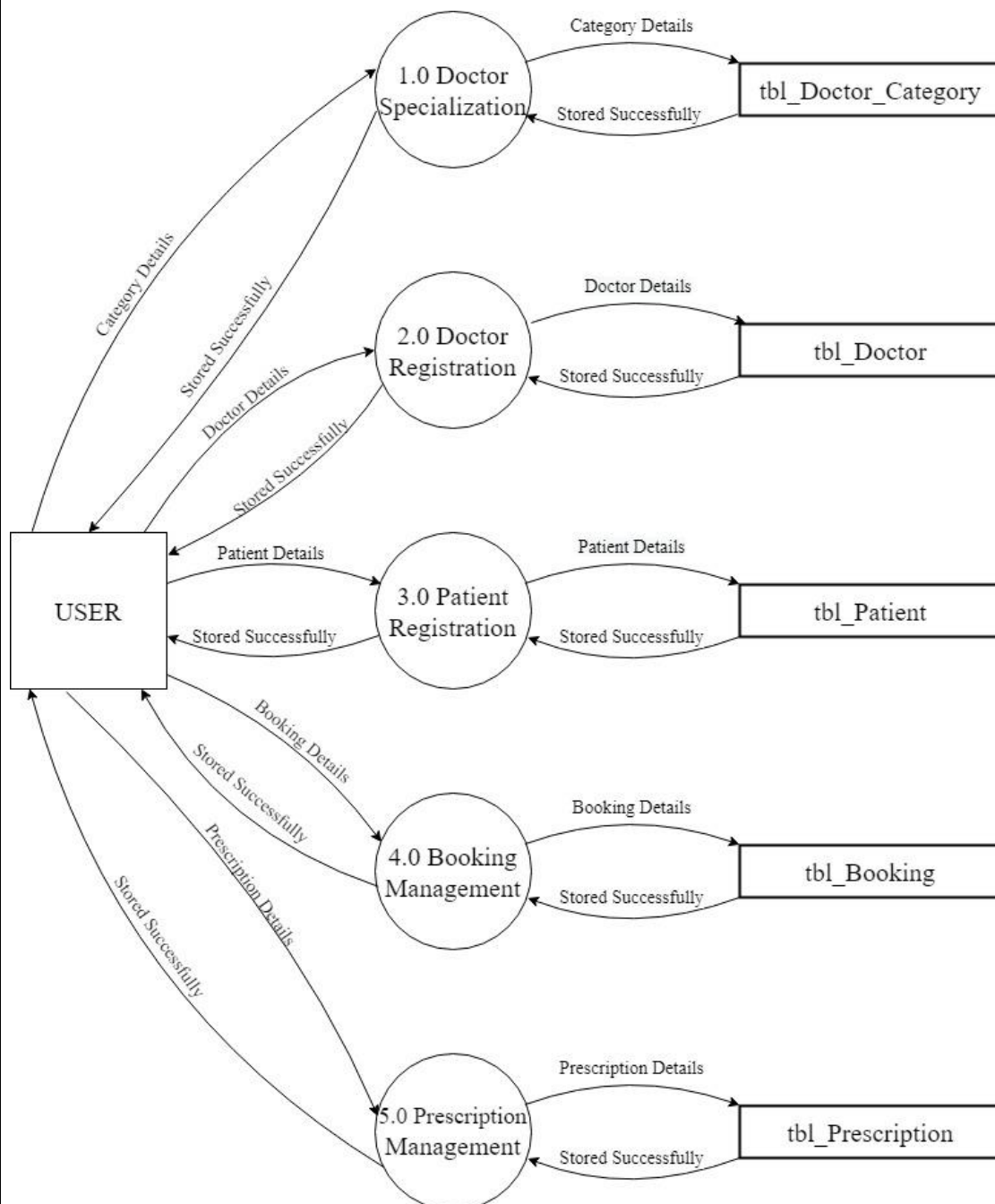
Rule 7: Do not try to put too much information in one DFD.

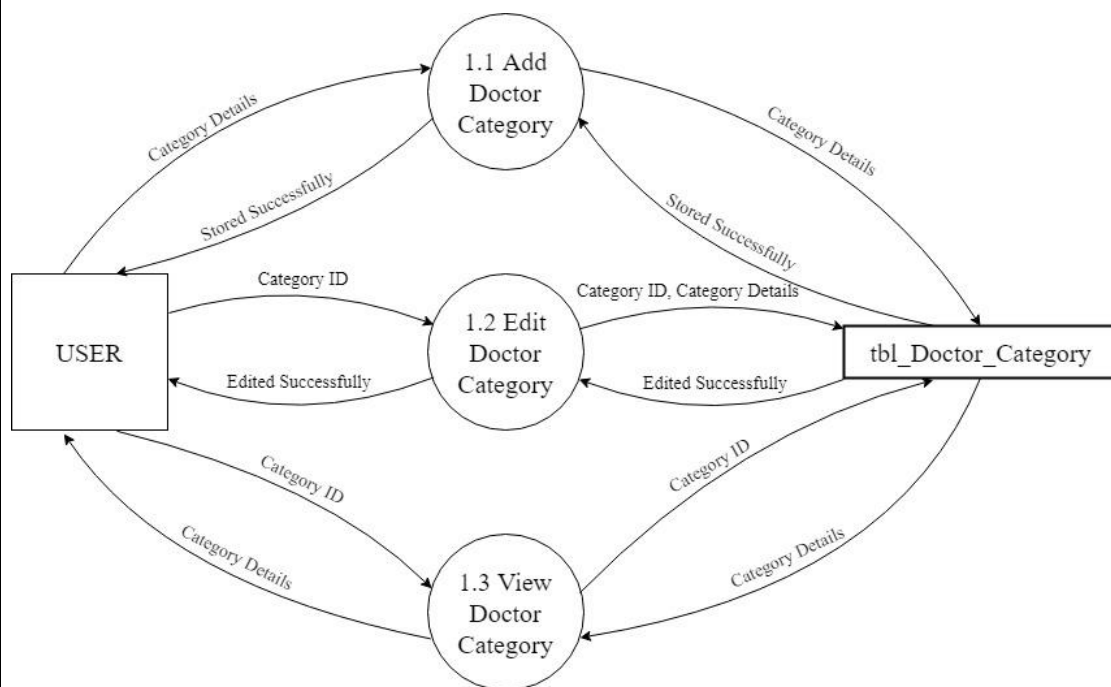
Rule 8: Be prepared to start over.

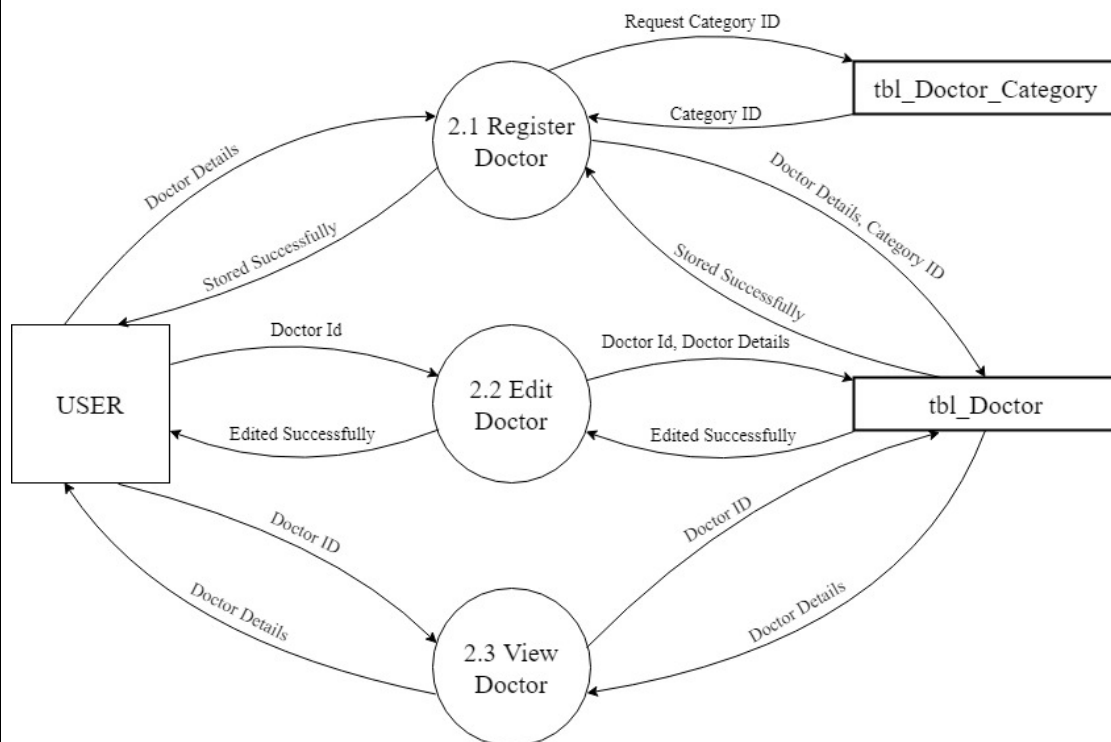
**Level 0 DFD Showing Patient Health Tracker**

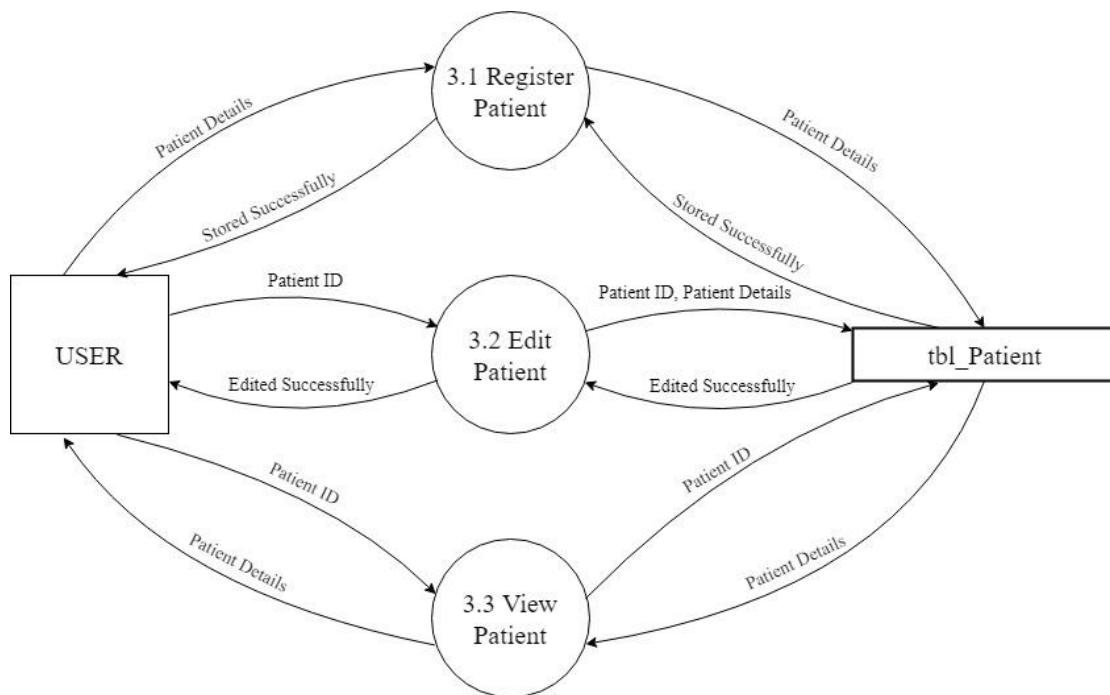




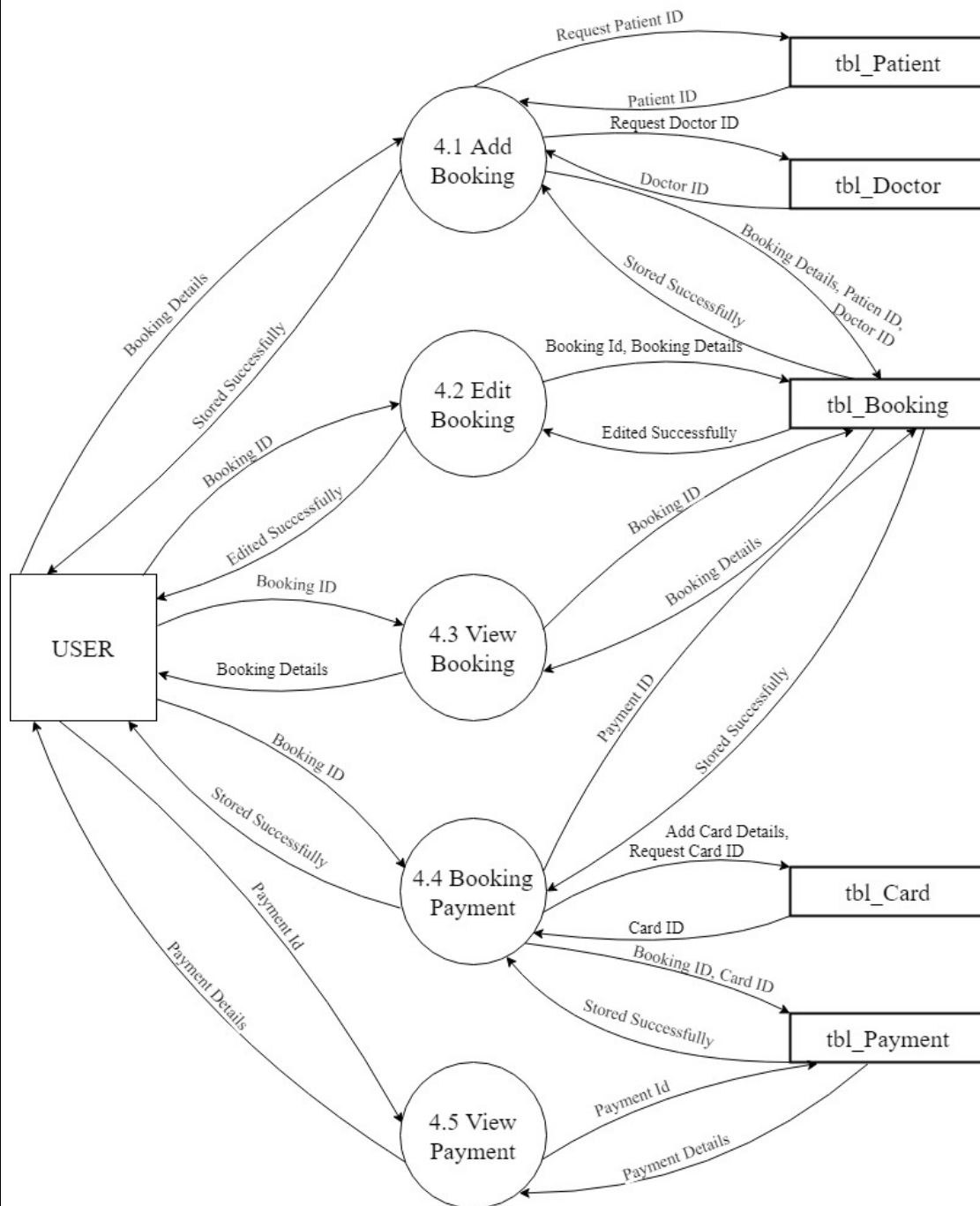
**Level 1 DFD Showing Patient Health Tracker**

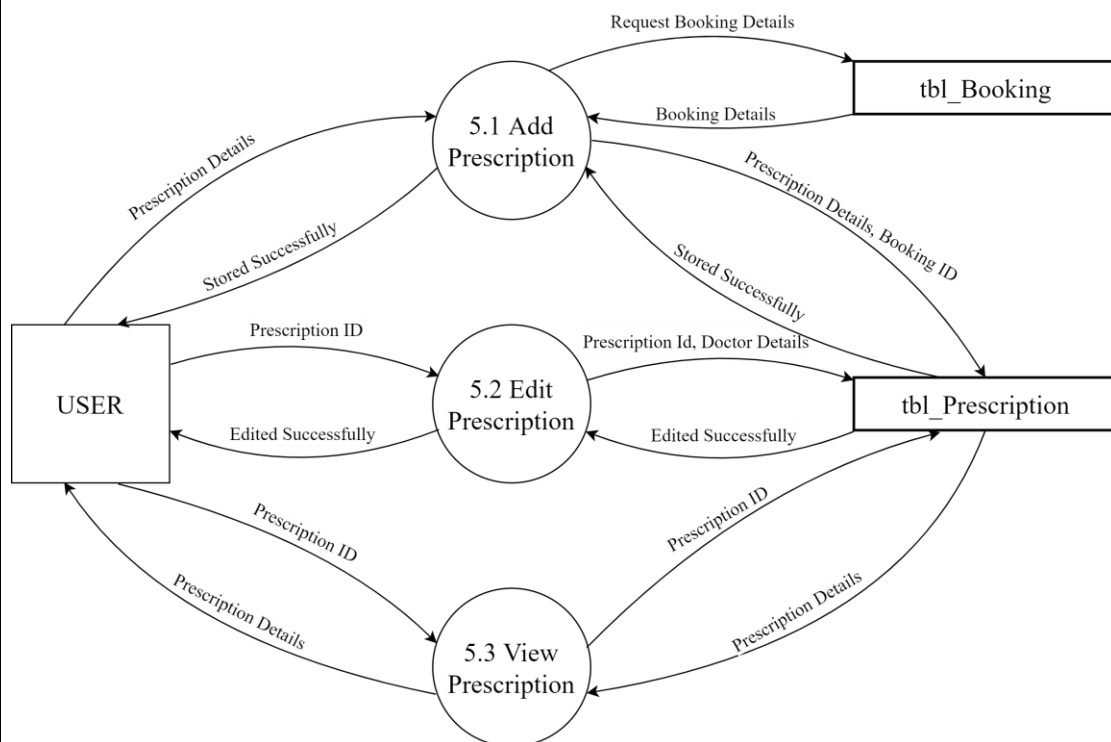
**Level 2 DFD Showing Doctor Specialization**

**Level 2 DFD Showing Doctor Registration**

**Level 2 DFD Showing Patient Registration**

### Level 2 DFD Showing Booking Management



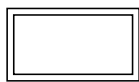
**Level 2 DFD Showing Prescription Management**

### 3.3. ENTITY RELATIONSHIP DIAGRAM

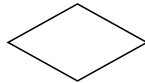
The ER model is a conceptual data model that views the real world as a construct of entities and associations or relationships between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. The ER modeling technique is frequently used for the conceptual design of database applications and many database applications and many database design tools employ its concepts.



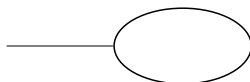
Entity Type



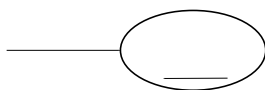
Weak Entity Type



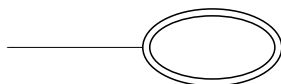
Relationship Type



Attribute

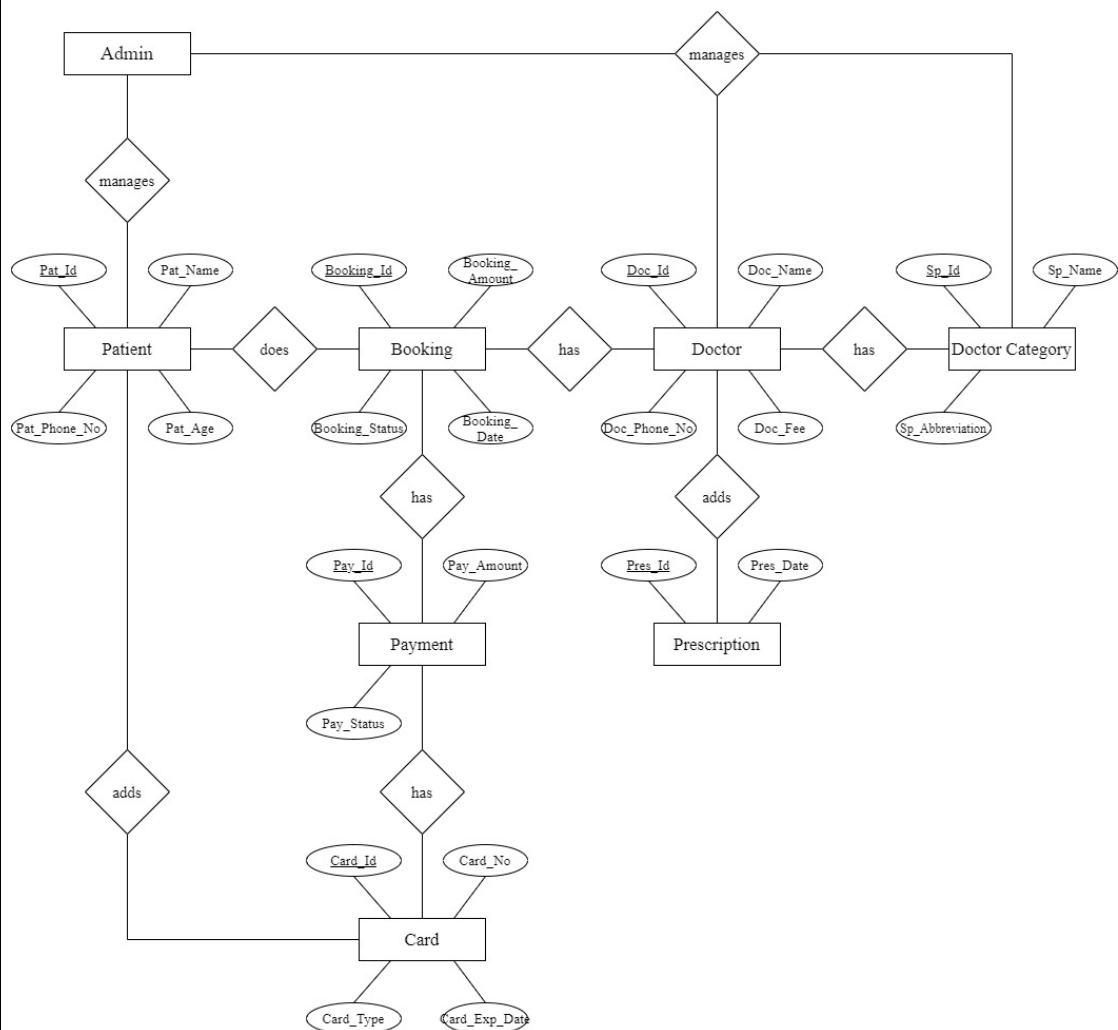


Key attribute



Multivalued Attribute

### ER Diagram Showing Patient Health Tracker



## 4. SYSTEM DESIGN



#### **4.1. INPUT DESIGN**

Input design is the process of converting a user-oriented description of the inputs to a computer-based system into a programmer-oriented specification. The quality of system input determines the quality of system output. Input specification describes the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of errors. The input design also determines whether the user can interact efficiently with the system.

Input design requires consideration of the needs of the data entry operator. Three data entry considerations are:

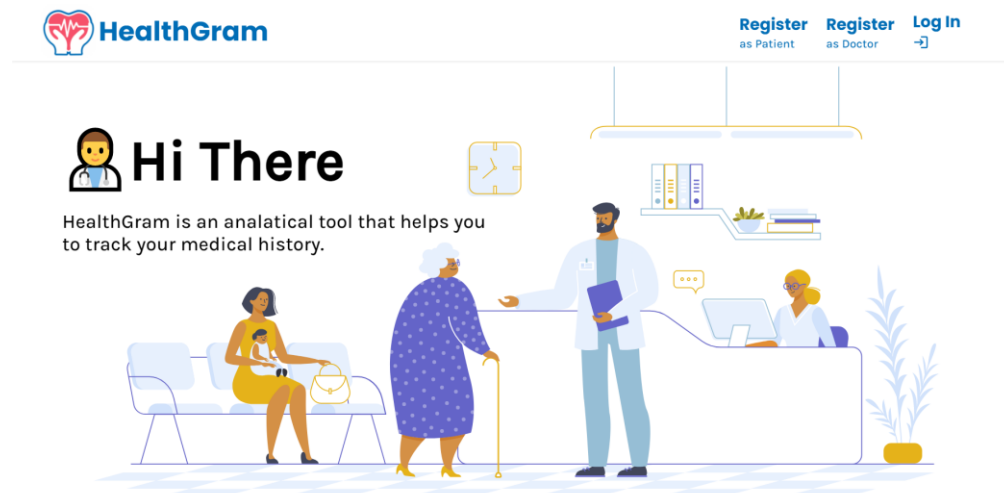
- The field length must be documented
- The sequence of fields must match the sequence of the fields on the source document.
- The data format must be identified to the data entry operator.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry can be controlled by input design. Input design is the process of converting user-oriented inputs to a computer-based format. There are three major approaches for entering data into the computer. They are menus, formatted forms and prompts. A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user chooses from the list of option. A formatted form is a preprinted form or a template that request the user to enter data in appropriate location. It is a fill-in-the-blank type form. The form is flashed on the screen as a unit. In prompt the system displays one enquiry at a time, asking the user for a response.

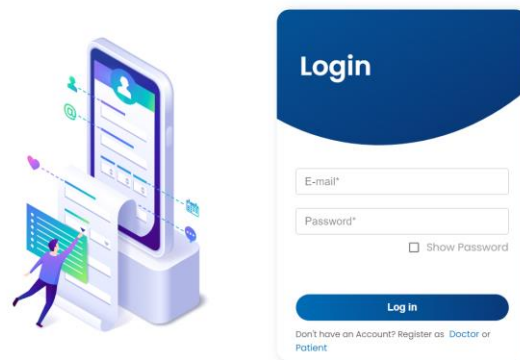
## Home Page

**Description:** This is the home page for all unauthenticated users



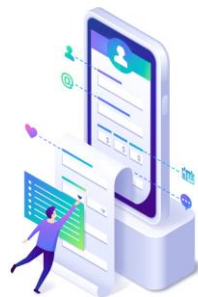
## Login Page

**Description:** This is the Login page for all the users



## Doctor Registration Page

**Description:** This is the registration form for doctors



Welcome  
DOCTOR

Please Fill Your Credentials To Continue !

Page 5 out of 5

UPLOAD DOCUMENTS

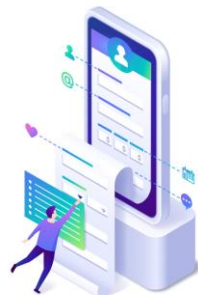
Upload your Certificate Of Degree Completion\*  
 No file chosen

Upload your Driving Licence\*  
 No file chosen

Profile Picture  
 No file chosen

Register

Not a Doctor? [Register as Patient](#)  
Already Have An Account? [Log In](#)



Welcome  
DOCTOR

Please Fill Your Credentials To Continue !

Page 1 out of 5

PERSONAL INFO

Next

Not a Doctor? [Register as Patient](#)  
Already Have An Account? [Log In](#)



Welcome  
DOCTOR

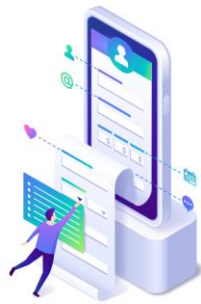
Please Fill Your Credentials To Continue !

Page 2 out of 5

ADDRESS DETAILS

Next

Not a Doctor? [Register as Patient](#)  
Already Have An Account? [Log In](#)



## Welcome DOCTOR

Please Fill Your Credentials To Continue !

Page 3 out of 5

### JOB DETAILS

Specialization

Fee\*

Self Description

Next

[Not a Doctor? Register as Patient](#)  
[Already Have An Account? Log In](#)



## Welcome DOCTOR

Please Fill Your Credentials To Continue !

Page 4 out of 5

### ACCOUNT DETAILS

E-mail\*

Password\*

☐ Show Password

Confirm Password\*

Next

[Not a Doctor? Register as Patient](#)  
[Already Have An Account? Log In](#)

## Patient Registration Page

**Description:** This is the registration form for patients



## Hi There

Please Fill Your Credentials To Continue !

Page 1 out of 4

### PERSONAL INFO

Full Name\*

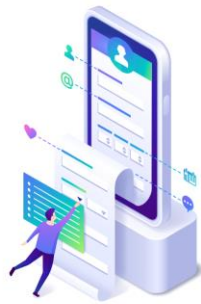
Phone Number\*

Gender\*

DOB\*

Next

[Not a Patient? Register as Doctor](#)  
[Already Have An Account? Log In](#)



## Hi There

Please Fill Your Credentials To Continue !

Page 2 out of 4

### ADDRESS DETAILS

House Number\*

Street \*

District\*

Pincode\*

Next

Not a Patient? [Register as Doctor](#)  
Already Have An Account? [Log In](#)



## Hi There

Please Fill Your Credentials To Continue !

Page 3 out of 4

### ACCOUNT DETAILS

E-mail\*

Password\*

☐ Show Password

Confirm Password\*

Next

Not a Patient? [Register as Doctor](#)  
Already Have An Account? [Log In](#)



## Hi There

Please Fill Your Credentials To Continue !

Page 4 out of 4

### UPLOAD DOCUMENTS

Upload your Driving Licence OR Any Proof For Your Age\*

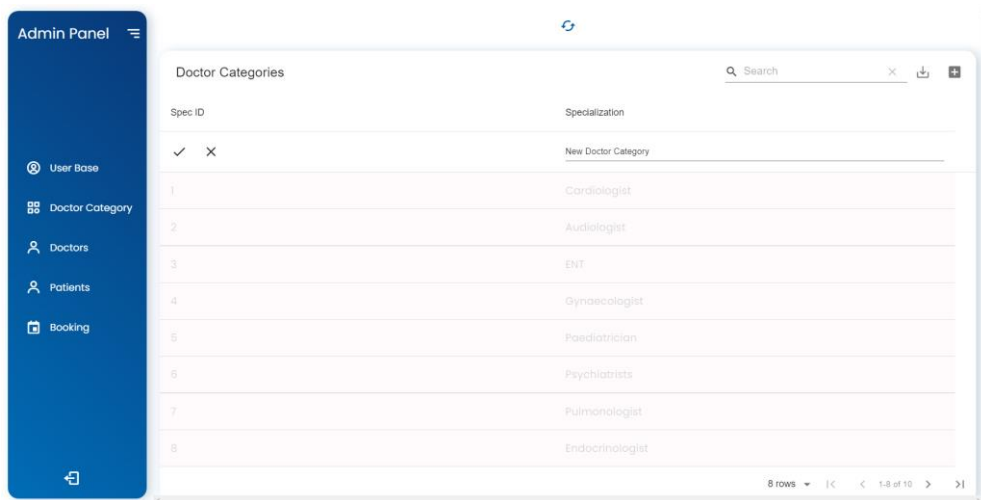
No file chosen

Register

Not a Patient? [Register as Doctor](#)  
Already Have An Account? [Log In](#)

## Adding Doctor Category

**Description:** Admin can add new doctor category



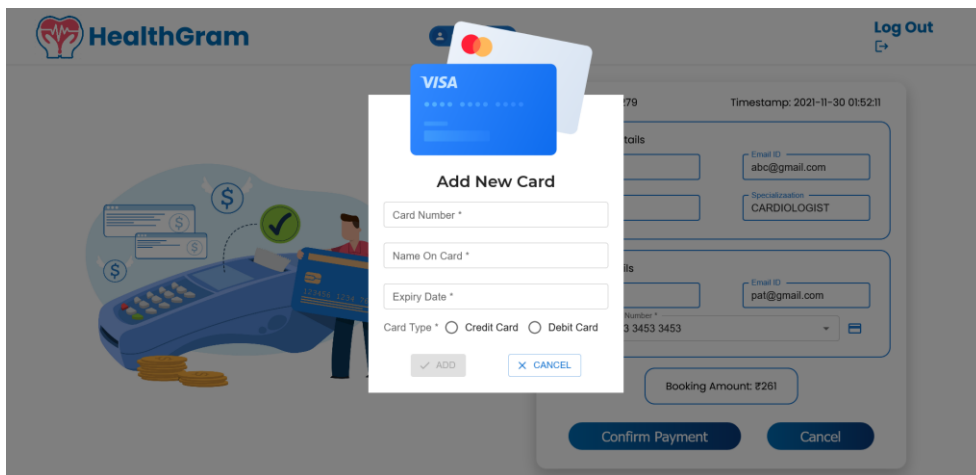
The screenshot shows the Admin Panel interface. On the left is a sidebar with navigation links: User Base, Doctor Category, Doctors, Patients, and Booking. The main content area is titled 'Doctor Categories' and contains a table with columns 'Spec ID', 'Specialization', and 'New Doctor Category'. The table lists 8 rows of doctor specializations.

| Spec ID | Specialization  | New Doctor Category |
|---------|-----------------|---------------------|
| 1       | Cardiologist    |                     |
| 2       | Audiologist     |                     |
| 3       | ENT             |                     |
| 4       | Gynaecologist   |                     |
| 5       | Paediatrician   |                     |
| 6       | Psychiatrists   |                     |
| 7       | Pulmonologist   |                     |
| 8       | Endocrinologist |                     |

At the bottom right of the table, it indicates '8 rows' and '1-8 of 10'.

## Adding Card

**Description:** This form adds new card



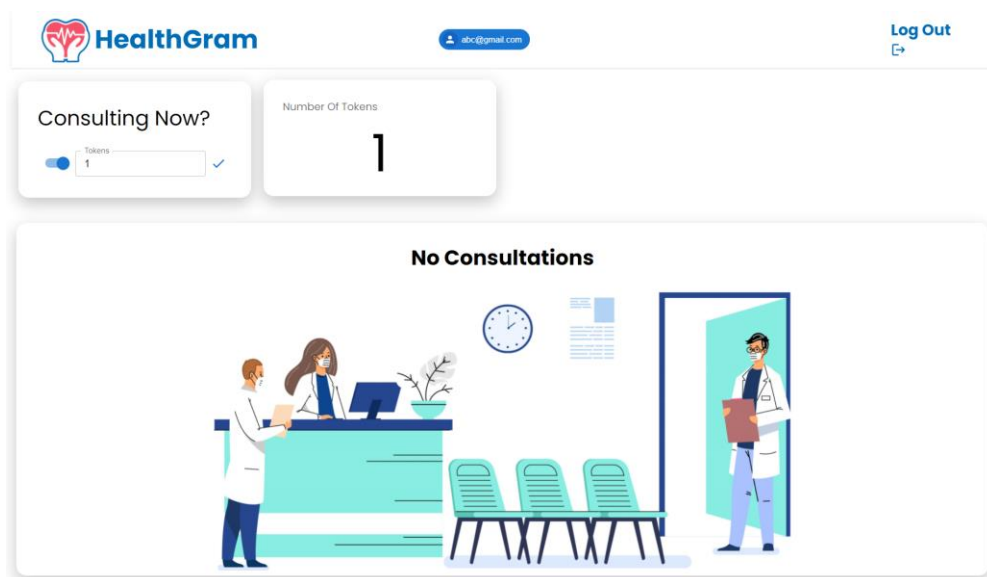
The screenshot shows the HealthGram app interface. A 'Log Out' button is in the top right. A 'VISA' card is displayed at the top. Below it is the 'Add New Card' form with the following fields:

- Card Number \*
- Name On Card \*
- Expiry Date \*
- Card Type \* ☐ Credit Card ☐ Debit Card

At the bottom of the form are 'ADD' and 'CANCEL' buttons. In the background, a payment confirmation screen is visible, showing a 'Booking Amount: ₹261' and 'Confirm Payment' and 'Cancel' buttons.

## Add Tokens

**Description:** Doctor can set the amount of tokens



The screenshot shows a web interface for 'HealthGram'. At the top, there is a logo on the left, a user profile 'abc@gmail.com' in the center, and a 'Log Out' button on the right. Below the header, there are two main sections. The first section, titled 'Consulting Now?', contains a toggle switch and a text input field labeled 'Tokens' with the value '1'. The second section, titled 'Number Of Tokens', displays a large number '1'. Below these sections is a large illustration of a doctor's office. The illustration shows a reception desk with two staff members, a clock on the wall, and a doctor standing near a door. The text 'No Consultations' is written above the illustration.

## 4.2. OUTPUT DESIGN

One of the important features of an information system for users is the output it produces. Output is the information delivered to users through the information system. Without quality output, the entire system appears to be unnecessary that users will avoid using it. Users generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user through an interactive process, until the result is considered to be satisfactory.

Output design has been an ongoing activity almost from the beginning of the project. In the study phase, outputs were identified and described general in the project directive. A tentative output medium was then selected and sketches made for each output. In the feasibility analysis, a “best” new system was selected; its description identified the input and output media. In the design phase the system has included an evaluation and selection of specific equipment for the system.

Outputs from computer systems are required primarily to communicate the results of processing to the user. They are also used to provide a permanent copy of these results for later consultation.

## Booking Page

**Description:** This page is where we book a doctor



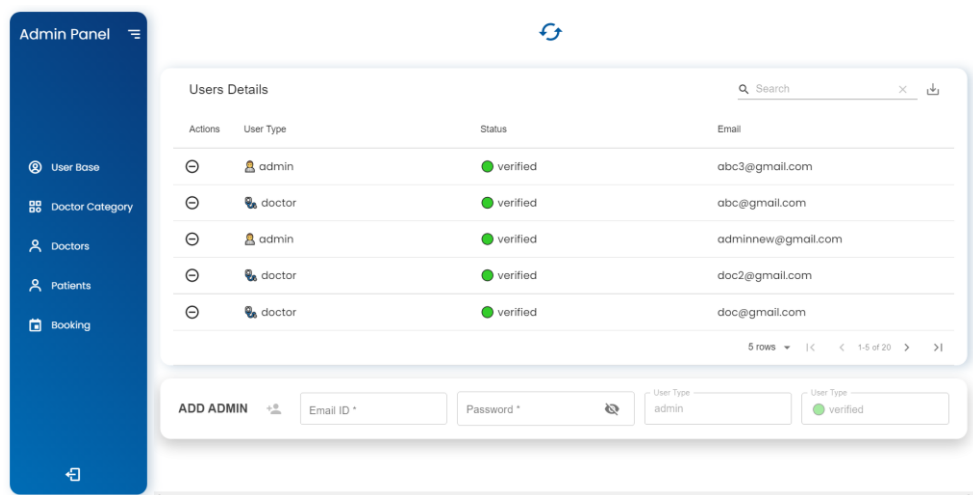
The mockup shows a web interface for 'HealthGram'. At the top, there's a blue header with the logo, a user email 'pat@gmail.com', and a 'Log Out' link. The main heading is 'Book Your Doctor Now'. Below it is a progress bar with three steps: 1. Find Doctor, 2. Doctor Confirmation, and 3. Payment. The 'Find Doctor' step is active. Below the progress bar are filters for 'Doctor Gender' (All, Male, Female), a 'Fee Range' slider (Min ₹ 0 to Max ₹ 9,999), and a search bar 'Search For Doctor Name/Specialization'. There are four doctor cards displayed in a 2x2 grid. Each card includes a profile picture, name, specialization, age, gender, a brief description, a fee, and a 'BOOK NOW' button.

| Doctor Name   | Specialization | Age          | Gender | Fee (₹) |
|---------------|----------------|--------------|--------|---------|
| Dr. ABC       | Cardiologist   | 25 Years old | MALE   | 253     |
| Dr. DOCUSER   | ENT            | 21 Years old | MALE   | 234     |
| Dr. DONA MISS | Paediatrician  | 31 Years old | FEMALE | 800     |
| Dr. FTJF      | Pulmonologist  | 25 Years old | FEMALE | 746     |



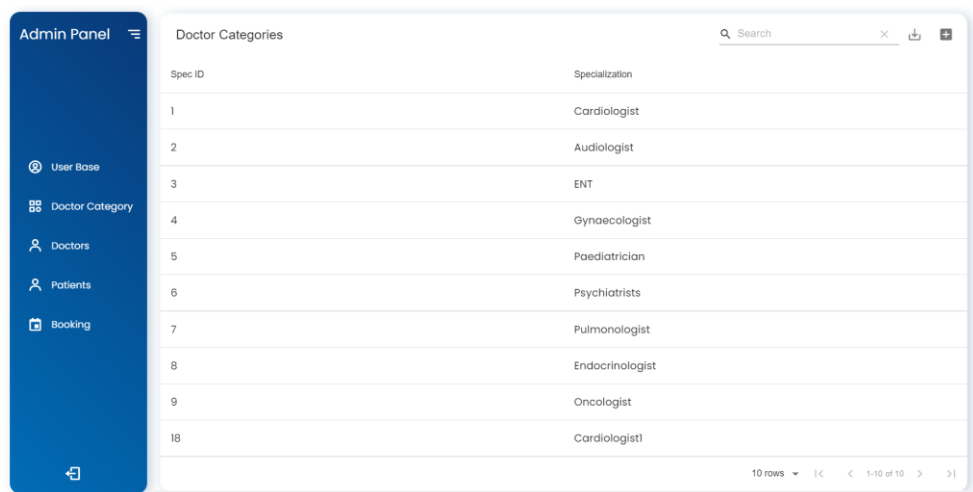
## User Details

**Description:** This page shows details of every user



## Doctor category

**Description:** This page shows every doctor specialization



## Doctor Details

**Description:** This page shows details of every doctor , verify them, make them active or inactive

| Doctors to be verified |        |                   |               |      |               |            |                     |
|------------------------|--------|-------------------|---------------|------|---------------|------------|---------------------|
| Actions                | Doc ID | Specialization ID | Email         | Name | Phone Number  | DOB        | Registered on       |
| >                      | 36     | 36                | tom@gmail.com | tom  | +919895167005 | 1990-12-12 | 2021-10-14 04:00:03 |
| >                      | 37     | 37                | job@gmail.com | job  | +917564634576 | 2000-10-10 | 2021-10-12 10:03:31 |

| Doctors Details |        |          |                   |                    |           |               |            |                     |        |     |
|-----------------|--------|----------|-------------------|--------------------|-----------|---------------|------------|---------------------|--------|-----|
| Actions         | Doc ID | Status   | Specialization ID | Email              | Name      | Phone Number  | DOB        | Registered on       | Tokens | Fee |
| >               | 35     | verified | 35                | testdoc@gmail.com  | testdoc   | +917034179071 | 1990-11-11 | 2021-10-12 08:54:36 | 10     | 800 |
| >               | 38     | verified | 38                | donamiss@gmail.com | dona miss | +915738462189 | 1990-11-26 | 2021-11-29 19:52:42 | 1      | 800 |

## Patient Details

**Description:** This page shows details of every patient, verify them, make them active or inactive

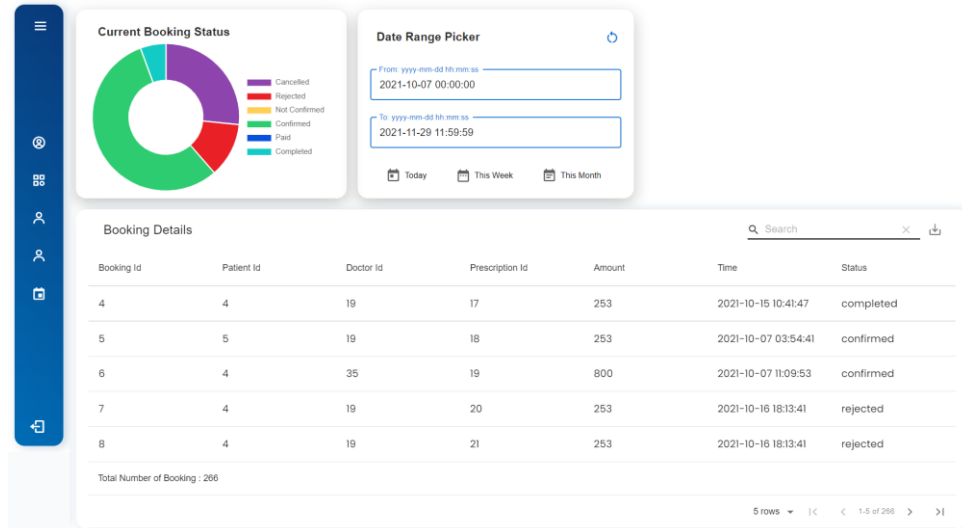
| Patients to be verified |        |                 |       |               |            |                     |
|-------------------------|--------|-----------------|-------|---------------|------------|---------------------|
| Actions                 | Pat ID | Email           | Name  | Phone Number  | DOB        | Registered on       |
| >                       | 6      | jacob@gmail.com | jacob | +915836163846 | 2000-11-11 | 2021-10-12 10:05:00 |

| Patients Details |            |              |                 |       |               |            |                     |        |              |        |          |          |
|------------------|------------|--------------|-----------------|-------|---------------|------------|---------------------|--------|--------------|--------|----------|----------|
| Actions          | Patient ID | Status       | Email           | Name  | Phone Number  | DOB        | Registered on       | Gender | House Number | Street | District | Pin Code |
| >                | 6          | not verified | jacob@gmail.com | jacob | +915836163846 | 2000-11-11 | 2021-10-12 10:05:00 | male   | fggf         | fgh    | fghf     | 34       |
| >                | 3          | verified     | new@gmail.com   | name  | +919895167005 | 2000-11-11 | 2021-09-17 11:14:53 | male   | 132          | st     | dist     | 12       |

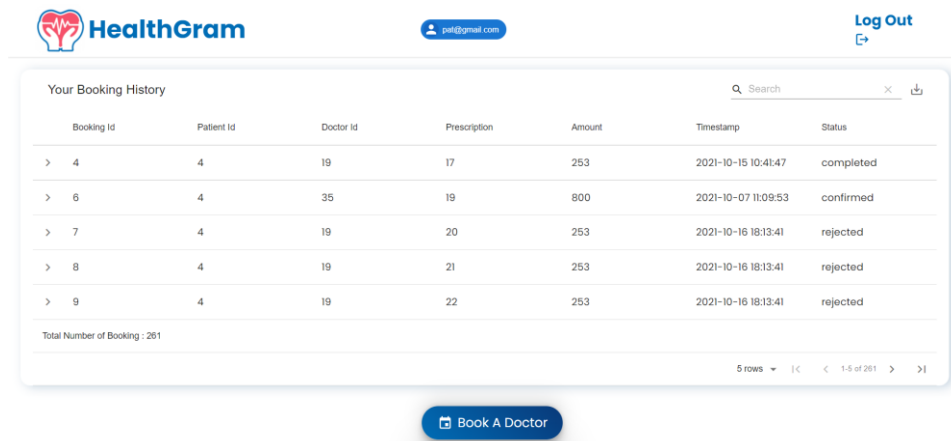
## Booking Details

**Description:** This page shows every booking details



## Booking History

**Description:** This page shows booking history of a patient



Payment Summary

**Description:** This page shows summary of payment during booking

✓


Find Doctor

✓

Doctor Confirmation

3

Payment



Booking ID: 279Timestamp: 2021-11-30 01:52:11

Doctor Details

Name

abc

Email ID

abc@gmail.com

Gender

MALE

Specialization

CARDIOLOGIST

Your Details

Name

iPatient

Email ID

pat@gmail.com

Select Card Number \*

4345 3453 3453 3453

Booking Amount: ₹261

Confirm Payment

Cancel

### **4.3. DATABASE DESIGN**

#### **4.3.1. Normalization**

Designing a database is a complex task and the normalization theory is a useful aid in this design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form.

A bad database design may lead to certain undesirable situations such as:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, normalization may be used. If the database is in a normalized form, the data can be restructured and can maintain it easily. This is important that the databases using that we are using may free from data redundancy and inconsistency. For this need we maintain the tables in a normalized manner.

#### **First Normal Form**

A relation is in first Normal Form (1NF), if and only if all its attributes are based on single domain. The objective of normalizing a table is in to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

#### **Second Normal Form**

A table is said to be in second Normal Form (2NF), when it is in 1 NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

#### **Third Normal Form**

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

**TABLE DESIGN****Table Name:** tbl\_Userbase**Table Description:** Stores User Details

| FIELD       | DATA TYPE   | CONSTRAINT  | DESCRIPTION   |
|-------------|-------------|-------------|---|
| Username    | Varchar(35) | Primary Key | Login Email   |
| Password    | Varchar(35) | Not Null    | Password  |
| User_Type   | Varchar(25) | Not Null    | Type of user(admin, doctor,patient)                     |
| User_Status | Varchar(25) | Not Null    | Current Status of User(not verified,verified, inactive) |

**Table Name:** tbl\_Login**Table Description:**Stores Login Details

| FIELD       | DATA TYPE   | CONSTRAINT  | DESCRIPTION             |
|-------------|-------------|-------------|-------------------------|
| Login_Id    | Int         | Primary Key | Login Id                |
| Username    | Varchar(25) | Foreign Key | Login Email             |
| Login_Time  | DateTime    | Not Null    | Date and Time of Login  |
| Logout_Time | DateTime    | Null        | Date and Time of Logout |

**Table Name:** tbl\_Doctor\_Category

**Table Description:** Doctor Specialization Details

| FIELD   | DATATYPE    | CONSTRAINT  | DESCRIPTION                 |
|---------|-------------|-------------|-----------------------------|
| Sp_Id   | Int         | Primary Key | Doctor<br>Specialization ID |
| Sp_Name | Varchar(35) | Unique      | Name of Category            |

**Table Name:** tbl\_Doctor

**Table Description:** Doctor Details

| FIELD               | DATATYPE     | CONSTRAINT  | DESCRIPTION                                   |
|---------------------|--------------|-------------|---|
| Doc_Id              | Int          | Primary Key | Doctor ID                                     |
| Sp_Id               | Int          | Foreign Key | Doctor<br>Specialization ID                   |
| Username            | Varchar(35)  | Foreign Key | Doctor Login<br>Email                         |
| Doc_Name            | Varchar(35)  | Not Null    | Doctor's Name                                 |
| Doc_phone_No        | Numeric(10)  | Unique      | Doctor Phone<br>Number                        |
| Doc_Dob             | Date         | Not Null    | Age of Patient                                |
| Doc_Gender          | Varchar(19)  | Not Null    | Gender of Doctor                              |
| Doc_House_No        | Varchar(10)  | Not Null    | Doctor House<br>Number                        |
| Doc_Street          | Varchar(15)  | Not Null    | Doctor Street                                 |
| Doc_Dist            | Varchar(15)  | Not Null    | Doctor District                               |
| Doc_Pin             | Numeric(6)   | Not Null    | Doctor Pin Code                               |
| Doc_Date_Registered | DateTime     | Not Null    | Date And Time<br>when Doctor<br>Registered    |
| Doc_No_of_Tokens    | Numeric(3)   | Not Null    | Number of<br>Patients the<br>Doctor Can Treat |
| Doc_Fee             | Decimal(5,2) | Not Null    | Fee of Doctor                                 |
| Doc_Pic             | Longtext     | Null        | Profile Picture of<br>Doctor                  |
| Doc_Proof           | Longtext     | Unique      | Doctor's ID Proof                             |
| Doc_Age_Proof       | Longtext     | Unique      | Doctor's ID Proof                             |
| Doc_Description     | Text         | Not Null    | About the Doctor                              |

**Table Name:** tbl\_Patient**Table Description:** Patient details

| FIELD               | DATATYPE    | CONSTRAINT  | DESCRIPTION                           |
|---------------------|-------------|-------------|---------------------------------------|
| Pat_Id              | Varchar(8)  | Primary Key | Patient ID                            |
| Username            | Varchar(35) | Foreign Key | Patient Login Email                   |
| Pat_Name            | Varchar(35) | Not Null    | Name of Patient                       |
| Pat_Phone_No        | Varchar(13) | Unique      | Patient Phone Number                  |
| Pat_Dob             | Varchar(10) | Not Null    | Age of Patient                        |
| Pat_Gender          | Varchar(19) | Not Null    | Gender of Patient                     |
| Pat_House_No        | Varchar(35) | Not Null    | Patient House Number                  |
| Pat_Street          | Varchar(35) | Not Null    | Patient Street                        |
| Pat_Dist            | Varchar(35) | Not Null    | Patient District                      |
| Pat_Pin             | Numeric(6)  | Not Null    | Patient Pin Code                      |
| Pat_Date_Registered | DateTime    | Not Null    | Date And Time when Patient Registered |
| Pat_Id_Proof        | Longtext    | Unique      | Patient's ID Proof                    |

**Table Name:** tbl\_Card**Table Description:** Card details

| FIELD         | DATATYPE    | CONSTRAINT  | DESCRIPTION         |
|---------------|-------------|-------------|---------------------|
| Card_Id       | Varchar(8)  | Primary Key | Card ID             |
| Pat_Id        | Varchar(8)  | Foreign Key | Patient ID          |
| Card_No       | Varchar(25) | Unique      | Card Number         |
| Card_Exp_Date | Date        | Not Null    | Expiry Date of Card |
| Card_Type     | Date        | Not Null    | Card Type           |

**Table Name:** tbl\_Payment



**Description:** Payment details

| FIELD      | DATATYPE     | CONSTRAINT  | DESCRIPTION       |
|------------|--------------|-------------|-------------------|
| Pay_Id     | Varchar(8)   | Primary Key | Payment ID        |
| Card_Id    | Varchar(8)   | Foreign Key | Card ID           |
| Pay_Amount | Decimal(8,2) | Not Null    | Amount to be Paid |
| Pay_Status | Varchar(10)  | Not Null    | Booking Status    |

**Table Name:** tbl\_Prescription

**Description:** Prescription details

| FIELD        | DATATYPE       | CONSTRAINT  | DESCRIPTION               |
|--------------|----------------|-------------|---------------------------|
| Pres_Id      | Varchar(8)     | Primary Key | Prescription ID           |
| Doc_Id       | Varchar(8)     | Foreign Key | Doctor ID                 |
| Pres_Date    | DateTime       | Not Null    | Prescribing Date and Time |
| Prescription | VarBinary(MAX) | Not Null    | Prescription Details      |

**Table Name:** tbl\_Booking

**Description:** Booking details

| FIELD          | DATATYPE     | CONSTRAINT  | DESCRIPTION       |
|----------------|--------------|-------------|-------------------|
| Booking_Id     | Varchar(8)   | Primary Key | Booking ID        |
| Pat_Id         | Varchar(8)   | Foreign Key | Patient ID        |
| Doc_Id         | Varchar(8)   | Foreign Key | Doctor ID         |
| Pres_Id        | Varchar(8)   | Foreign Key | Prescription ID   |
| Pay_Id         | Varchar(8)   | Foreign Key | Payment ID        |
| Booking_Amount | Decimal(8,2) | Not Null    | Amount to be Paid |
| Booking_Date   | DateTime     | Not Null    | Date of Booking   |
| Booking_Status | Varchar(10)  | Not Null    | Booking Status    |

## 5.1. INTRODUCTION

Software testing can be looked upon among the many process in organization that provides the last opportunity to correct any plane in the development system. System testing includes selecting tests and test data that have more problem of finding errors. System testing is vital for the success of any software system. The system makes a logical assumption that all part of the system works efficiently and goal is achieved. The system is tested for online response, ability to store and stress recovery from failure and usability. System testing requires a test plan that consists of several key activities and steps for programming and user acceptance testing.

Another benefit of system testing is its utility as a user-oriented system before implementation.

## **LEVELS OF TESTING**

Some of the methods of the system testing are given below.

### **Unit testing**

In this test each module is tested individually before integration it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing. In this test each module is tested whether it is producing the desired output and if any error occurs it can be corrected easily.

### **Integration testing**

It is the systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. Thus the relationship between different modules is checked in this testing for overall performance of testing. Thus, in integration testing step, all errors uncovered are corrected for next testing steps. The objective of the test is to take all the modules such as administrator, user and modules are integrated in this testing step and then the entire program is tested.

### **Validation testing**

It is where requirements established as a part of software requirements analysis is validated against the software that has been constructed. This test provides the final assurance that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are corrected during this phase.

### **Output Testing**

No system could be useful if it does not produce the required output in the specific format. Output testing is performed to ensure the correctness of the output and its format. The output generated or displayed by the system is tested asking the user about the format required by them.

### **User Acceptance Testing**

The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing. The testing of the software began along with the coding. The unit testing was done for each module in the software. For various inputs such that each line of code is executed at least once.

## 5.2 TEST CASES

A test plan document the strategy that will be used to verify and ensure that a product or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from test Engineers. Depending on the product and the responsibility of the organization to which the test plan applies.

### Unit Testing:

| Form              | Procedure  | Expected Result  | Actual Result                                       | Status |
|-------------------|--|--|---|--------|
| Entry Form        | Choose whether to login, view services or view about us. |  |   |        |
| Login Form        | Enter valid username and password                        | Should validate the user and provide link to user accounts.                                | Got the entry into user accounts.                   | Pass   |
| Registration form | Select the type of user and enter all mandatory fields   | Should validate all entered fields and flash a message indicating successful registration. | Message indicating successful registration is shown | Pass   |

|                         |   |  |  |      |
|-------------------------|---|--|--|------|
| Service Requesting Form | Select the required service, pincode, date and quantity | Should validate all entered fields and flash a message indicating successful requesting. | Message indicating successful requesting is shown. | pass |
| Service Accepting Form  | Desired request is accepted                             | Details of customer and date should be shown.  | Details of customer and date are shown             | Pass |
| Payment Form            | Enter all the mandatory fields                          | Should validate all entered fields and flash a message indicating successful payment.    | Message indicating successful payment is shown.    | pass |

**Integration Testing:**

| <b>Form combined</b>                        | <b>Expected result</b>                               | <b>Actual result</b>                | <b>Status</b> |
|---|--|-------------------------------------|---------------|
| Login and user account forms.               | Get entry to the appropriate user page.              | Appropriate user page is displayed. | Pass          |
| Registration Forms.                         | Must register the user successfully                  | Registration is successful          | Pass          |
| Service Requesting Form                     | Must add the specified entry in the database         | Specific entry in added.            | Pass          |
| Service Accepting Form                      | Must add the specified entry in the database         | Specific entry in added.            | Pass          |
| Payment Form                                | Must add the specified entry in the database         | Specified entry added.              | Pass          |
| Administration page and updating forms.     | Must pass updated details to corresponding database. | Update successful                   | Pass          |
| Administration page and calculating results | Must calculate and display results                   | Results displayed                   | Pass          |

|                           |  |   |      |
|---------------------------|--|---|------|
| Main page and other forms | Log out from all other forms should lead to the main page and disable all previous activities. | Main page is displayed and all previous activities are disabled | Pass |
|---------------------------|--|---|------|

**Validation Testing:**

| Test Case   | Expected result   | Actual result   | Status |
|-------------|---|---|--------|
| Create user | Check that all mandatory fields and validate all entered data fields. | If any error found display message and the same screen is displayed else record saved and confirms.                 | Pass   |
| Edit User   | Edit the row corresponding to the value entered                       | If the value entered is invalid error message is thrown otherwise message indicating successful deletion is flashed | Pass   |

## 6.1. INTRODUCTION

Implementation is that state in the project plan where the theoretical design is put into real test. All the theoretical and practical works are now implemented as a working system. This is the most crucial stage in the life cycle of a project, the project may be accepted or rejected depending on how it gathers confidence among the users. If the user has achieved satisfaction with the new project, then the project can be termed as successful and then onwards its maintenance and other subsequent works can be commenced. The system goes for implementation only after passing through some rigorous testing, especially when it comes to operating system and other system software, the testing and implementation phase assumes greater significance. The implementation stage involves following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the change cover.
- Evaluation of the changeover method.



## **6.2. INSTALLATION PROCEDURE**

Installation of software refers to the final installation of the package in the real environment, to the satisfaction of the intended users and the successful operation of the system. In many organizations, those who commission the software development project will not be the one to operate them. In the initial stage, the person who is not sure that the software will make the jobs easier will doubt about the software. But we have to ensure that the resistance does not build one makes sure that

- The active user must be aware of the benefits of using the system
- Their confidence in the software is built up
- Proper guidance is imparted to the user so that he is comfortable in using the application

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system into actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification of the system.

### **6.3. IMPLEMENTATION PLAN**

Implementation is the most crucial stage in achieving a successful system and for us it is the processing of bringing “Patient Health Tracker” into operational use and training it over to the user. Implementation includes all those activities that take place to convert from the old system to new one. The basic requirements for implementing the proposed system are already mentioned above. This software provides total security for the operations. That is, it prevents any unauthorized access. After successful login the user can go to the form according to the situation. After completion of the Patient Health Tracker System design and coding, the analyst, the user and the management evaluate the system to ensure that it fulfil all its goals. Thus, with the implementation of the project the critical design is turned into a working system. System implementation plan is concerned with writing program, creating databases, testing programs and operational plans.

## **7.1 FUTURE ENHANCEMENT**

The system has been developed with flexibility in mind. The requirement of the company is bound to change as and when new operations are included. Keeping in view advancements that are being made in technology it is necessary that the system be able to cope up with the changes that are bound to happen. So, in today's world of mobile technology the software “Patient Health Tracker” if integrated with the mobile will be an added advantage. The mobile users will get instant alerts from this site. The software if we create a mobile app or an alert system for more interaction with the user and also widening the reach of the system to its users. The system entitled “Patient Health Tracker” provides maximum user interaction and flexibility. The system users stored procedures on the database. This also can be enhanced in the future.

## **BIBLIOGRAPHY**

[illegible]