

Healthcare System

A PROJECT REPORT

Submitted by

**DIVY PATEL [220SBECE30014]
KISHAN PATEL [220SBECE30018]**

*In fulfillment for the award of the degree
of*
BACHELOR OF ENGINEERING
in
COMPUTER ENGINEERING



LDRP Institute of Technology and Research, Gandhinagar

Kadi Sarva Vishwavidyalaya

April, 2021-22

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH

GANDHINAGAR

CE-IT Department



CERTIFICATE

This is to certify that the Project Work entitled **“Healthcare System”** has been carried out by **Divy Patel (220SBECE30014)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-6 of Kadi Sarva Vishwavidyalaya University during the academic year **2021-22**.

Name of Guide

Jalpa Khamar

Internal Guide

LDRP ITR

Name of HOD

Prof. Shivanghi Surati

Head of the Department

LDRP ITR

**LDRP INSTITUTE OF TECHNOLOGY AND
RESEARCH
GANDHINAGAR**

CE-IT Department



CERTIFICATE

This is to certify that the Project Work entitled **“Healthcare System”** has been carried out by **Kishan Patel (220SBECE30018)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-6 of Kadi Sarva Vishwavidyalaya University during the academic year 2021-22.

Name of Guide

Jalpa Khamar

Internal Guide

LDRP ITR

Name of HOD

Prof. Shivanghi Surati

Head of the Department

LDRP ITR

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide for her exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by her time to time shall carry me a long way in the journey of life on which I am about to embark. I also take this opportunity to express a deep sense of gratitude to University for cordial support, valuable information and guidance, which helped me in completing this task through various stages.

Divy Patel (220SBECE30014)

Kishan Patel (220SBECE30018)

ABSTRACT

Generally, the process of getting Doctor Appointment is time consuming. The Patient needs to visit the Doctor's Clinic and there also they have to travel a long distance and wait for a long queues. So, the main purpose of this application is to provide the online Doctor Appointment and Lab Test Appointment. The purpose of the project is to build an application program to reduce the manual work for managing Doctor Appointment and Lab Test.

Healthcare system is a one stop mobile application an individual required to fulfil their health needs. User can do the online consultation & solve their queries via chat, audio /video call conferencing. Also user can book an appointment to run the laboratory tests at his/her convenience. In a nutshell, Healthcare application is the easiest solution and most convenient way to add your health related concerns.

LIST OF FIGURES

NO.	NAME	PAGE NO.
1.	Class Diagram	9
2.	Top Level Use Case Diagram	11
3.	Sequence Diagram	12
4.	0-Level DFD	14
5.	1-Level DFD	15
6.	Entity Relationship Diagram	17
7.	Activity Diagram of Patient	23
8.	Activity Diagram of Doctor	24
9.	Activity Diagram of Lab Manager	25
10.	Activity Diagram of Admin	25
11.	Test Report Diagram	26

LIST OF TABLES

NO.	NAME	PAGE NO.
1.	Patient	18
2.	Doctor	19
3.	Admin	20
4.	Appointment	20
5.	Lab Manager	21

TABLE OF CONTENTS

NO.	CHAPTER NAME	PAGE NO.
	Acknowledgement	I
	Abstract	II
	List of Figures	III
	List of Tables	IV
	Table of Contents	V
1.	Introduction	1
	1.1 Introduction	1
	1.2 Scope	1
	1.3 Key Features of the System	1
2.	Technology and Literature Review	2
	2.1 Tools and Technology	2
	2.1.1 HTML/CSS	2
	2.1.2 PHP	2
	2.2 Project Planning	3
	2.2.1 Project Development Approach	3
	2.2.2 Milestones and Deliverables	3
	2.3 Cost Estimation	4
3.	System Requirements Study	6
	3.1 User Characteristics	6
	3.2 Software and Hardware Requirements	6
	3.2.1 Software Requirements	6
	3.2.2 Hardware Requirements	6
	3.2.3 Functional Requirements	7
	3.2.4 Non-Functional Requirements	7

3.3 Assumptions and Dependencies	8
3.3.1 Assumptions	8
3.3.2 Dependencies	8
4. Function of System	9
4.1 Class Diagram	10
4.2 System Activity	11
4.3 Sequence Diagram	12
4.4 Data Flow Diagram	13
5. System Design	16
5.1 Entity Relationship Diagram	16
5.2 Data Dictionary	17
5.2.1 Farmer	18
5.2.2 Product	18
5.2.3 Admin	19
5.2.4 Sell Product	19
5.2.5 Sell Crop	20
5.2.6 Crops	21
5.3 Activity Diagram	22
5.3.1 Activity diagram for Farmer	22
5.3.2 Activity diagram for Depo Manager	23
5.3.3 Activity diagram for Yard Manager	24
5.3.4 Activity diagram for Admin	25
6. Testing	26
6.1 Test Report	26
6.2 Testing Planning Steps	26
6.2.1 Functionality Testing	27
6.2.2 Usability Testing	27
6.2.3 Interface Testing	27

6.2.4 Performance Testing	28
6.2.5 Security Testing	29
6.3 Testing Strategies	30
6.3.1 White Box Testing	32
6.3.2 Black Box Testing	33
6.4 Test Case Scenario	34
6.4.1 Test Case for Inconsistent Database	34
6.4.2 Test Case for Foreign Key Constraint	34
6.4.3 Test Case for Order Details Addition	35
6.4.4 Test Case for Deleting Log-in Credentials	36
7. Conclusion	37
7.1 Conclusion	37
7.2 Bibliography	37

1. INTRODUCTION

1.1 Introduction:

Healthcare system is a one stop mobile application an individual required to fulfil their health needs. User can do the online consultation & solve their queries via chat, audio /video call conferencing. Also user can book an appointment to run the laboratory tests at his/her convenience. In a nutshell, Healthcare application is the easiest solution and most convenient way to add your health related concerns.

1.2 Scope

The Healthcare System is Very Helpful to Manage the Time of all users like Doctor, Lab manager and Patient. Also Doctor can see the Patient History Reports and Problems. Lab Manager also have easy to manage appointment and Upload the Reports.

1.3 Key features of the System:

Generally, the process of getting Doctor Appointment is time consuming. The Patient needs to visit the Doctor's Clinic and there also they have to travel a long distance and wait for a long queues. So, the main purpose of this application is to provide the online Doctor Appointment and Lab Test Appointment. The purpose of the project is to built an application program to reduce the manual work for managing Doctor Appointment and Lab Test.

2. TECHNOLOGY AND LITERATURE REVIEW

2.1 Tools and Technology:

2.1.1 React Native:

React Native lets you create truly native apps and doesn't compromise your users' experiences. It provides a core set of platform agnostic native components

2.1.2 Project Development Approach:

The model that is referred for the development of the project is INCREMENTAL model. It combines elements of the waterfall model applied in an iterative fashion. In this process the phases are same as waterfall but the advantage is that when first phase is done it is incremented and then the other phases are carried with the same cycle. Here in this add ones on each phase can be added according to the need of the client and the project. Phases are as follows:

- Communication
- Planning
- .Modeling: Includes Designing
- Construction
- Deployment: Feedback, Delivery

Each phases are iteratively carried out. Main reason for using this than any other is waterfall has the drawback of iterations, if there is any other requirement added later on then this is not possible to add up in it, Spiral model has disadvantage that it need more manpower and even it is for multiple transactions or multiple tasks handling projects and so does the time consumption is more in it for those projects.

Planning is essential because multiple software teams works in parallel on different system functions. Scalability should be obtained in any of the project selected but it is not available in waterfall cause of few drawbacks.

2.1.2 Milestones and Deliverables:

- **Month 1: Milestones & Deliverables**

Milestones	Deliverables
Study about our web application requirement, planning	Analysis Report
Understand a project definitions and basic terms and logic for Parameter Evaluation.	
Gathering the requirements of the project using different fact-finding techniques.	Analysis Report
Still Continue with Requirement's study.	

Month 2-4: Milestones & Deliverables

Milestones	Deliverables
System Analysis	Analysis Report
System Design including various diagrams	SRS

- **Month 5 to 7: Milestones & Deliverables**

Milestones	Deliverables
Integrating techniques of bootstrap and Django	Designing/Coding
Database creation and Procedures	Designing/Coding
Admin Module of Client Support System	Designing/Coding
Consultant Module of Client Support System	Designing/Coding
Accountant Module of Client Support System	Designing/Coding

- **Month 8: Milestones & Deliverables**

Milestones	Deliverables
Client Module of Client Support System	Designing/Coding
Website Testing	Testing
Required changes after testing	Designing/Coding

2.2 Cost Estimation:

Several models are used to find the estimate costing of the software but we are using the well-known model known as COCOMO model.

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- **Effort:** Amount of labour that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

- Organic
- Semi-detached
- Embedded

SOFTWARE PROJECTS	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Since we are unfamiliar with some aspects of the system like server handling and some more aspects so our project comes under semi-detached model. Therefore

$$\begin{aligned}\text{Effort} &= a * (\text{KLOC})^b \text{ PM} \\ &= (3.0) * (12.500)^{1.12} \\ &= 50.78 \text{ person-month approx.}\end{aligned}$$

$$\begin{aligned}\text{Schedule} &= c * (\text{Effort})^d \\ &= 2.5 * (50.78)^{0.35} \\ &= 9.80 \text{ months approx.}\end{aligned}$$

Here as we see 5 months estimate time is taken by the team of 2 people to make this website if we take average salary of each person as 20,000rs/month hence the website will cost $2 * 20,000 * 5 = 2,00,000\text{rs}$ approx. where Transportation, hardware and other cost have not been included.

3. SYSTEM REQUIREMENTS STUDY

3.1 User Characteristics:

Analysing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure that we are still developing the system for the end users.

The user must have following characteristics:

- User must have basic knowledge of Computers.
- User should understand the use of all modules.
- User can easily interact with the proposed system.
- User must know the technical terms used in the company for performing different tasks specially related to call logs, payment details, transportation details and report retrieval.
- User should be also being aware about the running process of the system.

3.2 Software and Hardware Requirements:

Software and Hardware Requirements are used to describe the minimum hardware and software requirements to run the Software. These requirements are described below.

3.2.1 Software Requirements:

- **Client:**
 - Operating System: Android or IOS
 - Android 6.0+

3.2.2 Hardware Requirements:

- **Client:**
 - Smartphone that have Play Store
 - 4GB of RAM
 - Internet Connection

3.2.3 Functional Requirements:

The basic features available in the system are as follows:

1. Patient can search for the specialist doctor's profile and book an appointment.
2. Patient can view their past visit details under History section
3. Patient can run health check up (AI Program) and order medication from linked pharmacy
4. Patient can book an appointment for laboratory tests
5. Patient gets reminder for medication every day
6. Doctor can manage patient's appointments
7. Also doctor can view patient history
8. Doctor can consult patient on online appointment
9. Doctor can prepare the prescription and send it to patient
10. Patient can view the laboratory test results in the patient profile

3.2.4 Non-Functional Requirements:

Following is a list of nonfunctional requirements:

- **Performance:**

This system should remain accessible 24x7. At least 100 users should be able to access the system altogether at any given time.

- **Security:**

The database of system should not store any password in plain text rather the ceasercipher text has to be stored.

- **Reliability:**

It can be accessed by the end users 24*7 as needed hence is highly reliable for endusers.

- **Availability:**

Internet connection for the nodes with the database server is ensured and hence the application will be available any time for access.

- **Portability:**

The developed web application is portable as it can be accessed from any operationsystem regardless Windows, Mac, Linux provided they have a browser to access Internet.

3.3 Assumptions and Dependencies:

3.3.1 Assumptions:

- Database transactions are assumed to be secure and reliable.
- User is the person having enough knowledge for the traversing operation.
- We will provide a user-friendly interface so that any user can easily navigate through the system, but he/she should be capable of providing valid credentials for successful login.
- The server used for data storing is always secured.

3.3.2 Dependencies:

- The system is dependent upon the user's valid credentials. If user inputs wrongusername or password, he/she will not be allowed to login to the system.
- This application depends on the server and internet as all the information is collected and then stored in the server through secure internet connection.
- All the users of the system will be assigned a specific role. According to these roles each and every user will be allowed to access predefined set of feature.

4 FUNCTION OF SYSTEM

4.1 Class Diagram:

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

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as –

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

4.2 System Activity:

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating.

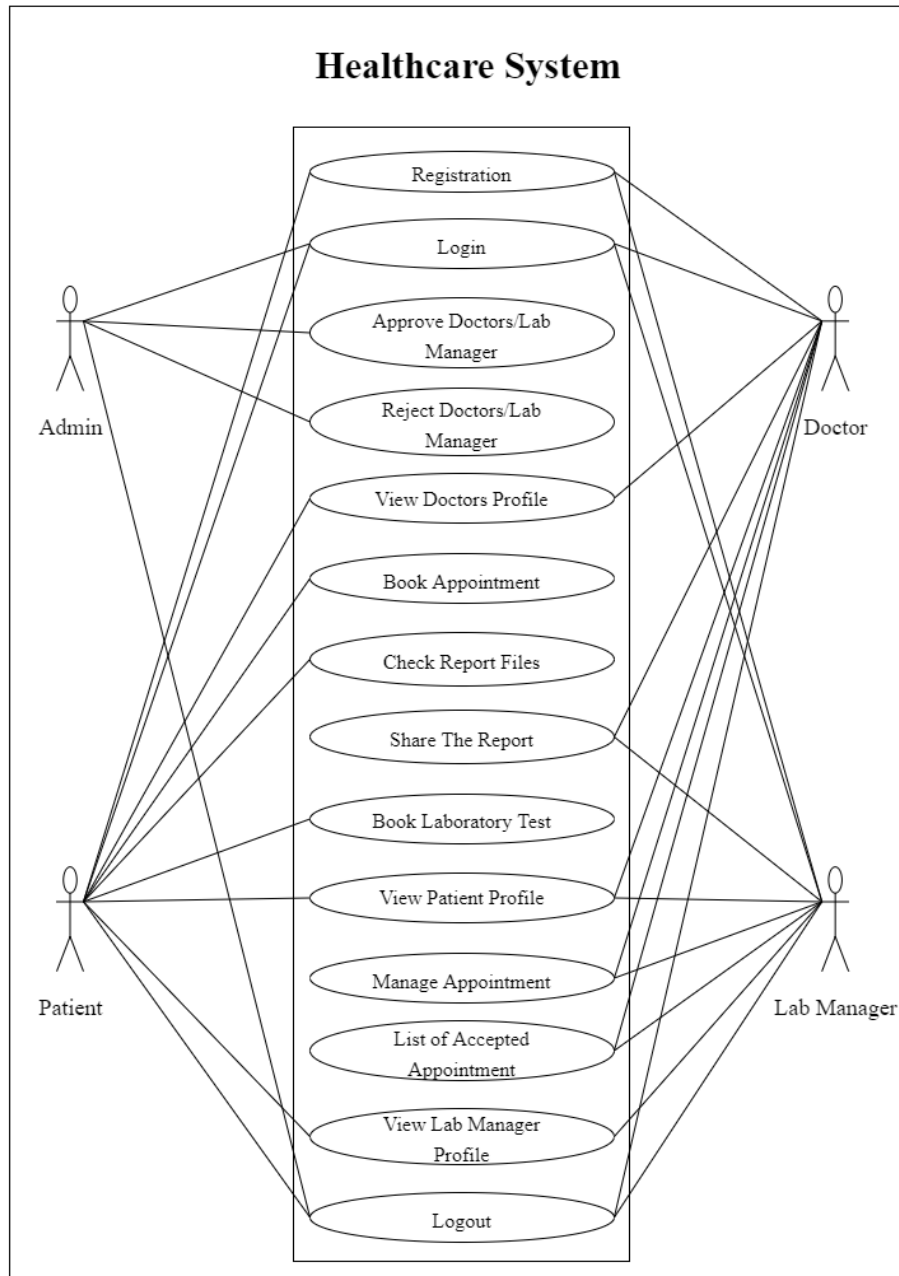
Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements and actors.

Figure 4.2.1: Use Case Diagram of Healthcare System



4.3 Sequence Diagram:

Uml Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence Diagrams captures:

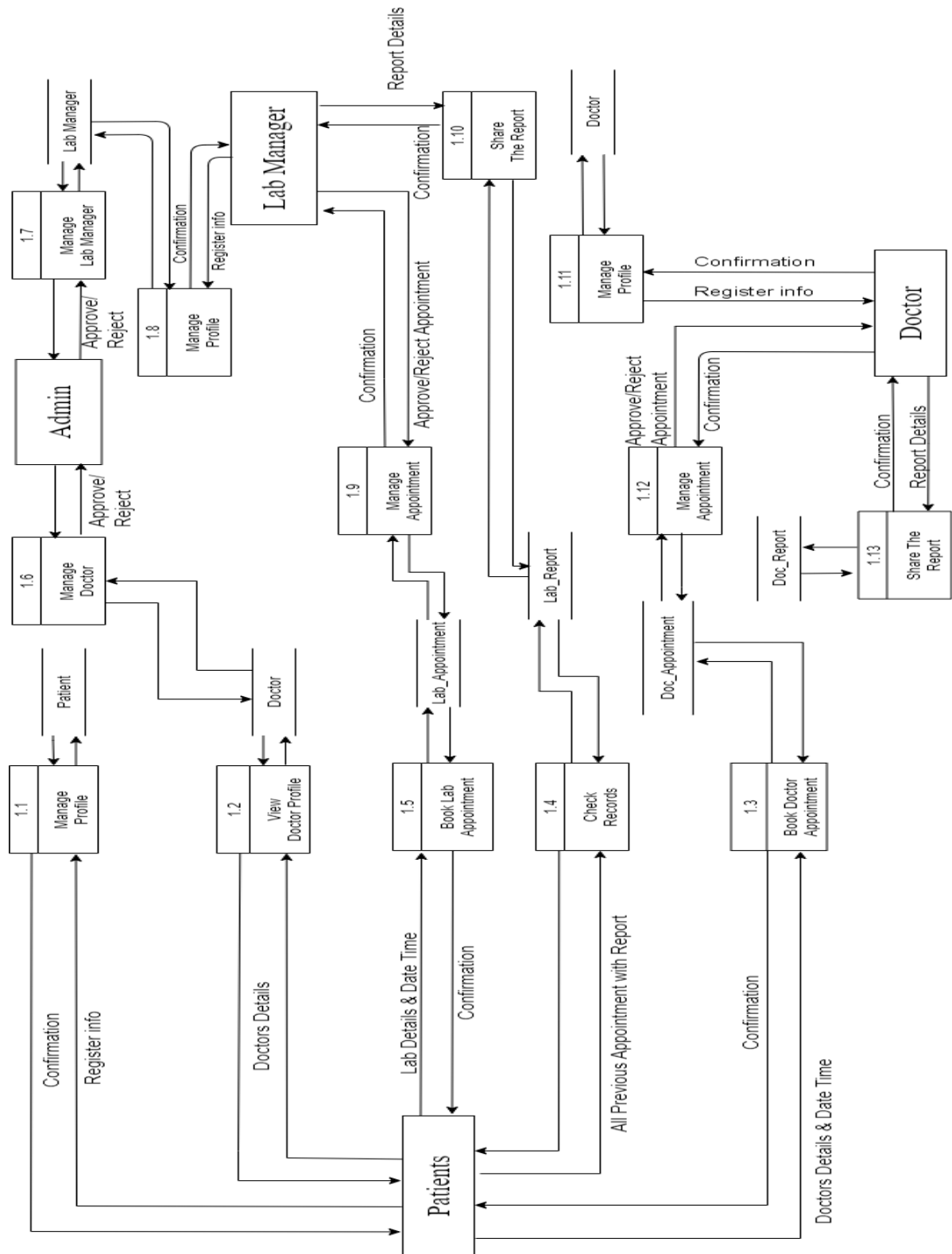
- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

4.4 Data Flow Diagram:

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expandit to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements

Figure 4.4.2: 1-level Data-Flow diagram of Healthcare System



5 SYSTEM DESIGN

5.1 Entity Relationship Diagram:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties by defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.

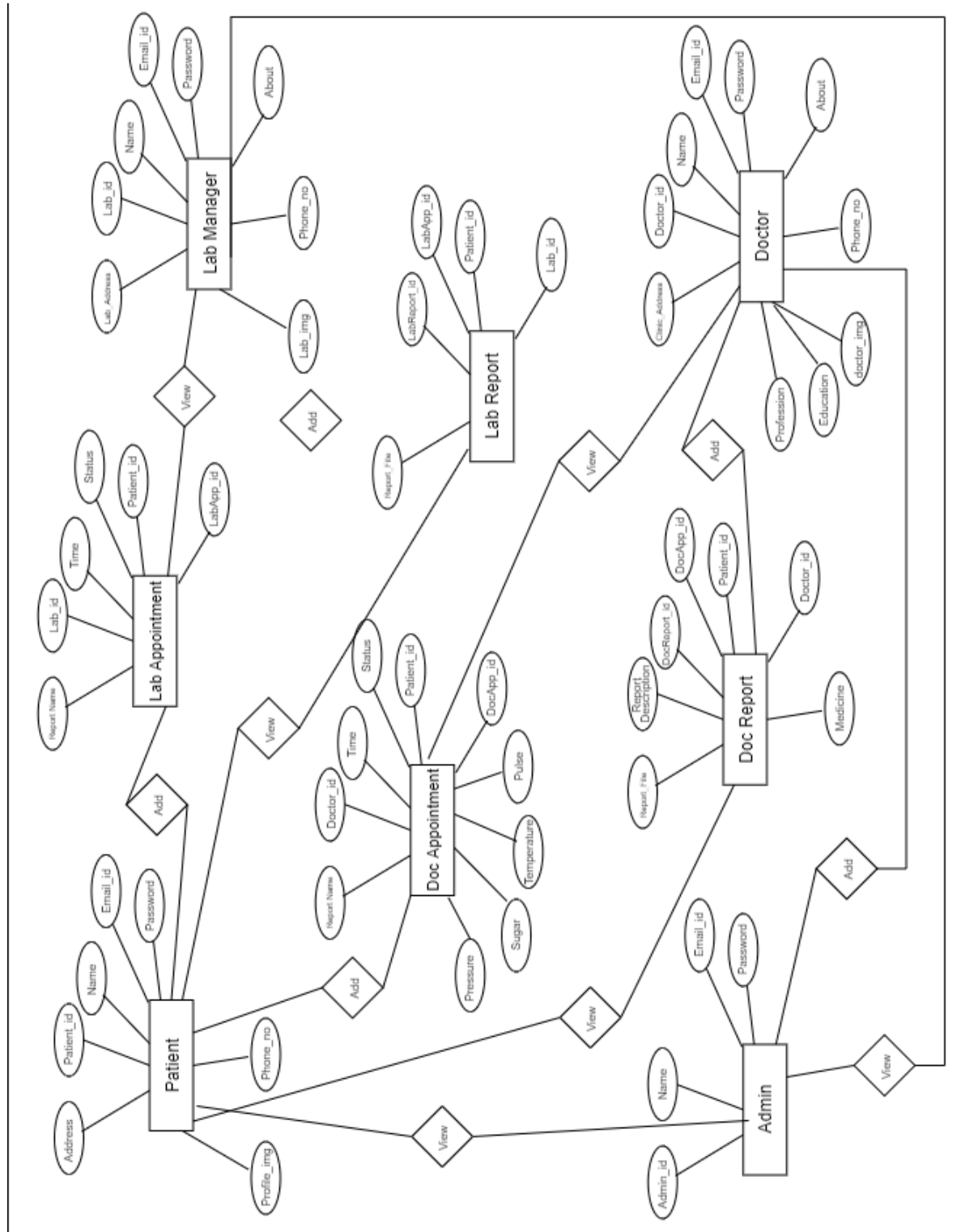


Figure5.1.1: ER diagram of Healthcare System

5.2 Data Dictionary:

A data dictionary contains metadata i.e., data about the database. 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 users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.

The data dictionary in general contains information about the following –

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

5.2.1 Patient:

Primary Key:- P_id

Description:- Used to Register/Login and Add the Patient.

Sr_no	Field name	Data Type	Length	Constraint	Description
1	P_id	Int	10	Primary	Patient id
2	Pname	Varchar	20	Not Null	Patient Name
3	Email	Varchar	20	Not null	Email id
4	Password	Varchar	20	Not null	Password
5	Mo_number	Int	10	Null	Mobile number
6	Address	Varchar	40	Null	Address

5.2.2 Doctor:

Primary Key:- D_id

Description:- Used to Get Details of the Doctor.

Sr_No	Field Name	Data Type	Length	Constraint	Description
1	D_id	Int	10	Primary	Doctor id
2	D_name	Varchar	20	Not Null	Doctor Name
3	D_address	Int	10	Not null	Doctor price
4	D_photo	Blob	10	Null	Photo
5	D_About	Varchar	40	Null	About

5.2.3 Admin:

Primary Key:- A_id

Description:- Store Details of the Admin.

Sr_No	Field Name	Data Type	Lengh	Constraint	Description
1	A_id	Int	10	Primary	id
2	A_name	Varchar	20	Not Null	Name
3	Email	Varchar	20	Null	Email id
4	Password	Varchar	20	Not null	Password
5	Mo_number	Int	10	Null	Mobile number

5.2.4 Appointment:

Foreign Key:- D_id, P_id

Description:- Make Records of Appointments.

Sr_No	Field Name	Data Type	Length	Constraint	Description
1	app_id	Int	10	Foreign Key	App_id
2	D_id	Int	20	Foreign Key	D Id
3	Desc	Int	20	Not null	desc
4	P_name	Varchar	20	Not Null	P_Name

5.2.5 Lab Manager:

Primary Key:- L_id

Description:- Lab Manager.

Sr_No	Field Name	Data Type	Length	Constraint	Description
1	L_id	Int	10	Primary	Lab id
2	L_name	Varchar	20	Not Null	Lab Name
3	L_photo	Blob	10	Null	Photo
4	L_address	Int	10	Not Null	Address
5	L_description	Varchar	40	Null	Description

5.3 Activity Diagram:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

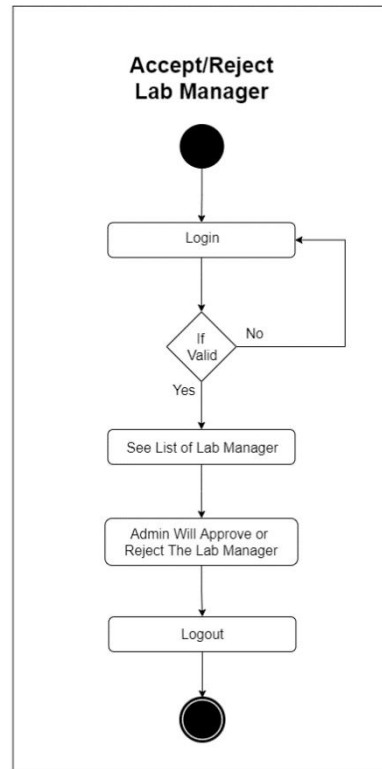
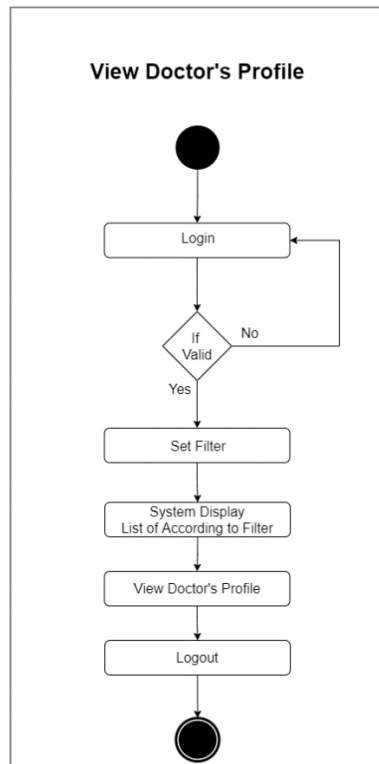
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

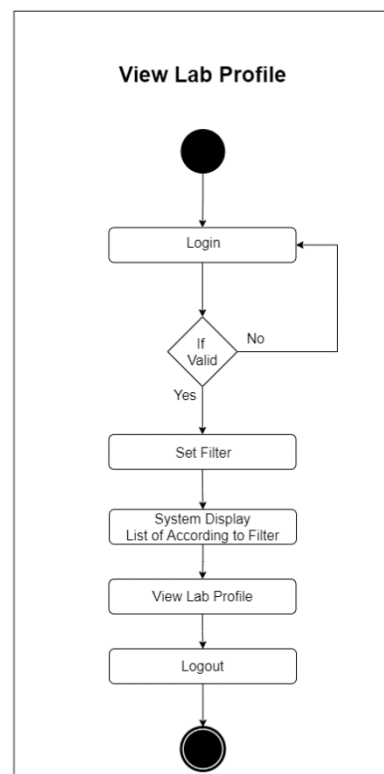
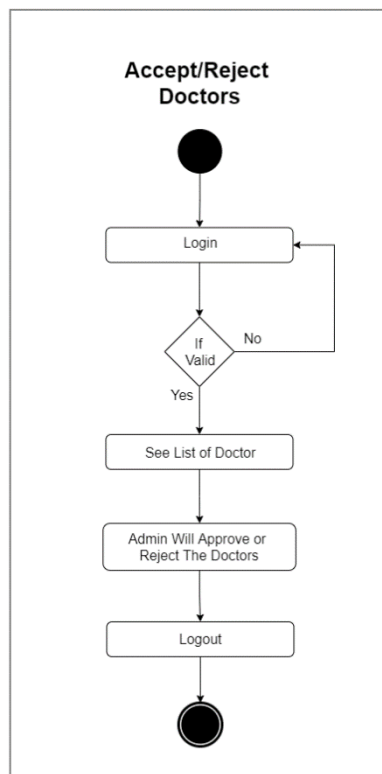
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The purpose of an activity diagram can be described as –

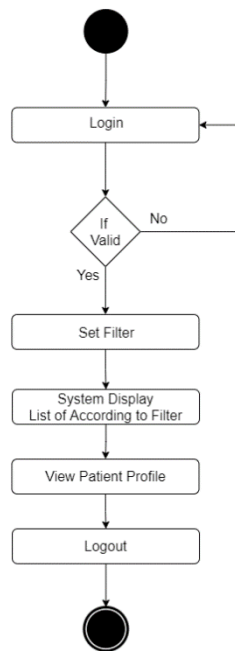
- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

There are 2 activity diagram for this system one for user side and another for administration side.

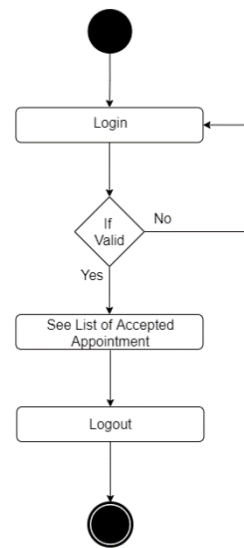




View Patient Profile



List of Accepted Appointment



6 TESTING

6.1 Test Report:

Test Report is a document which contains a summary of all test activities and final test results of a testing project. Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release

For example, if the test report informs that there are many defects remaining in the product, stakeholders can delay the release until all the defects are fixed.

- **What does a test report contain?**



Fig. 6.1.1 test report diagram

Project Information:

All information of the project such as the project name, product name, and version should be described in the test report.

6.2 Testing Planning Steps:

- 1) Functionality Testing
- 2) Usability testing
- 3) Interface testing
- 4) Performance testing
- 5) Security testing

6.2.1 Functionality Testing:

Test for – all the links in web pages, database connection, forms used in the webpages for submitting or getting information from user, Cookie testing. Check all the links:

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from web pages.
- Test to check if there are any orphan pages.
- Lastly in link checking, check for broken links in all above-mentioned links.
- Test forms in all pages: Forms are the integral part of any web site. Forms are used to get information from users and to keep interaction with them. So, what should be checked on these forms?
- First check all the validations on each field.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.
- Let's take example of the search engine project currently I am working on, In this project, we have advertiser and affiliate signup steps. Each sign up step is different but dependent on other steps. So, sign up flow should get executed correctly.

There are different field validations like email Ids, User financial info validations. All these validations should get checked in manual or automated web testing.

- **Cookies testing:** Cookies are small files stored on user machine. These are basically used to maintain the session mainly login sessions. Test the application by enabling or disabling the cookies in your browser options. Test if the cookies are encrypted before writing to user machine. If you are testing the session cookies (i.e. cookies expire after the session ends) check for login sessions and user stats after session end. Check effect on application security by deleting the cookies. (I will soon write separate article on cookie testing)
- **Validate your HTML/CSS:** If you are optimizing your site for Search engines then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if site is crawlable to different search engines.
- **Database testing:** Data consistency is very important in web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality. Check, if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More on database testing could be load on DB, we will address this in web load or performance testing below.

6.2.2 Usability Testing:

- **Test for navigation:** Navigation means how the user surfs the web pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages. Usability testing includes: Web site should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy purpose. Main menu should be provided on each page. It should be consistent.
- **Content:** Content should be logical and easy to understand. Check for spelling errors. Use of dark colours annoys users and should not be used in site theme. You can follow some standards that are used for web page and content building. These are common accepted standards like as I mentioned above about annoying colours, fonts, frames etc.
- Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes. These are some basic standards that should be followed in web development. Your task is to validate all for UI testing.

6.2.3 Interface Testing:

The main interfaces are:

- Web server and application server interface
- Application server and Database server interface.

Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users.

Check what happens if user interrupts any transaction in-between? Check what happens if connection to web server is reset in between?

6.2.4 Performance Testing:

Web application should sustain to heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing

Test application performance on different internet connection speed. In the web load testing test if many users are accessing or requesting the same page. Can system sustain in peak load times? Site should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

Stress testing: Generally, stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes.

Stress is generally given on input fields, login and sign up areas. In web performance testing web site functionality on different operating systems, different hardware platforms are checked for software, hardware memory leakage errors.

parameter to different site ID which is not related to logged in user. Access should be denied for this user to view others stats.

- Try some invalid inputs in input fields like login username, password, input textboxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given download option.
- Test if SSL is used for security measures. If used proper message should get displayed when user switch from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on web server.

6.3 Testing Strategies:

6.3.1 White Box Testing:

If we go by the definition, “White box testing” (also known as clear, glass box or structural testing) is a testing technique which evaluates the code and the internal structure of a program.

White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised.

White box testing coverage specifications:

1. Code coverage

2. Segment coverage: Ensure that each code statement is executed once.

3. Branch Coverage or Node Testing: Coverage of each code branch in from all possible was.

4. Compound Condition Coverage: For multiple conditions test each condition with multiple

5. Data Flow Testing (DFT): In this approach you track the specific variables through each possible calculation, thus defining the set of intermediate paths through the code. DFT tends to reflect dependencies but it is mainly through sequences of data manipulation. In short, each data variable is tracked and its use is verified. This approach tends to uncover bugs like variables used but not initialize, or declared but not used, and so on.

5. Path Testing: Path testing is where all possible paths through the code are defined and covered. It's a time-consuming task.

6. Loop Testing: These strategies relate to testing single loops, concatenated loops, and nested loops. Independent and dependent code loops and values are tested by this approach.

Limitations:

Not possible for testing each and every path of the loops in the program. This means exhaustive testing is impossible for large systems.

This does not mean that WBT is not effective. By selecting important logical paths and data structure for testing is practically possible and effective.

Steps to Perform WBT:

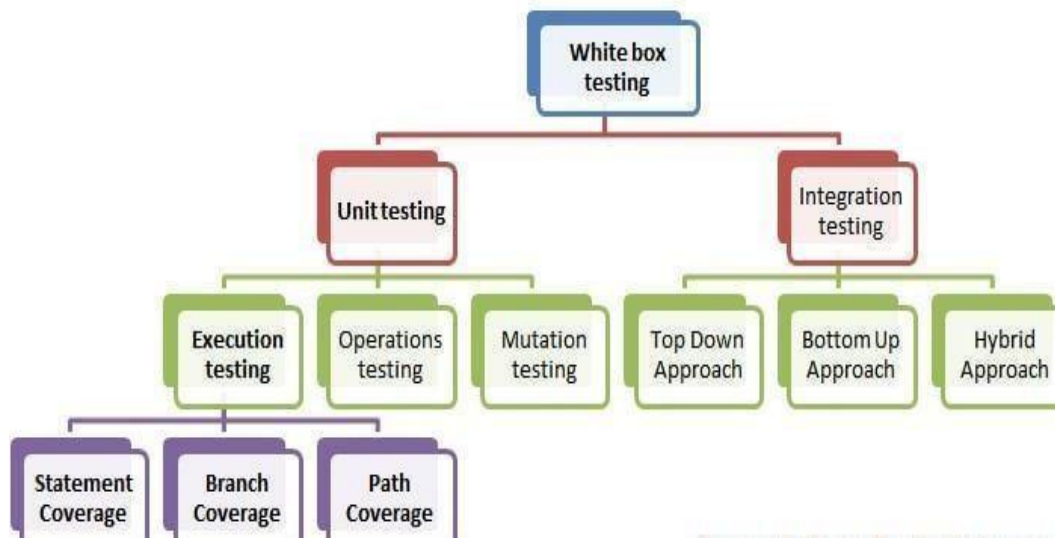
Step 1 – Understand the functionality of an application through its source code. Which means that a tester must be well versed with the programming language and the other tools as well techniques used to develop the software.

Step 2– Create the tests and execute them.

When we discuss the concept of testing, “coverage” is considered to be the most important factor. Here I will explain how to have maximum coverage from the context of White box testing.

- **Types and Techniques of White Box Testing:**

Types of White Box Testing



Today, we are going to focus mainly on the execution testing types of 'Unit testing white box technique'.

3 Main White Box Testing Techniques:

1. Statement Coverage
2. Branch Coverage
3. Path Coverage

Note that the statement, branch or path coverage does not identify any bug or defect that needs to be fixed. It only identifies those lines of code which are either never executed or remains untouched. Based on this further testing can be focused on.

Let's understand these techniques one by one with a simple example.

1) Statement coverage:

In a programming language, a statement is nothing but the line of code or instruction for the computer to understand and act accordingly.

In case of an “IF statement”, there will be two test conditions: One to validate the true branch and,

Other to validate the false branch.

Hence, in theory, Branch Coverage is a testing method which is when executed ensures that each and every branch from each decision point is executed.

2) Path Coverage

Path coverage tests all the paths of the program. This is a comprehensive technique which ensures that all the paths of the program are traversed at least once. Path Coverage is even more powerful than Branch coverage. This technique is useful for testing the complex programs.

6.3.2 Black Box Testing:

Black Box Testing is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing.

It is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.

The main focus in Black Box Testing is on the functionality of the system as a whole. The term ‘Behavioral Testing’ is also used for Black Box Testing. Behavioral test design is slightly different from the black-box test design because the use of internal knowledge isn’t

Types of Black Box Testing:

1) Functional Testing:

This type deals with the functional requirements or specifications of an application. Here, different actions or functions of the system are being tested by providing the input and comparing the actual output with the expected output.

For Example, when we test a Dropdown list, we click on it and verify that it expands and all the expected values are showing in the list. **Few major types of Functional Testing are:**

- Smoke Testing
- Sanity Testing
- Integration Testing
- System Testing
- Regression Testing
- User Acceptance Testing

2) Non-Functional Testing:

Apart from the functionalities of the requirements, there are several non-functional aspects as well that are required to be tested to improve the quality and performance of the application.

Few major types of Non-Functional Testing include:

- Usability Testing
- Load Testing
- Performance Testing
- Compatibility Testing
- Stress Testing

6.4 Test Case Scenario:

6.4.1 Test Case for Inconsistent Database:

TEST CASE ID: TC001		NAME: Database Inconsistency
PURPOSE	Checking out the consistency of database in oracle server through SQL Developer	
INPUT	Inserting new data through form	
EXPECTED OUTPUT	Reflection/addition of data in database.	
STEPS:		
Open client registration form.		
2.	Fill data and click submit it to add to the database	This should be reflected in database tables.

6.4.2 Test Case for Foreign Key Constraint Violation:

TEST CASE ID: TC002		NAME: Foreign Key Violation
PURPOSE	To assure that The Sell Product details are not going to delete	
INPUT	Product and Farmer Details	
EXPECTED OUTPUT	Prevention to delete the records	
STEPS:		
Farmer Buy Product from yard manager.		
Buying Process.		

6.4.3 Test Case for Order Details Addition:

TEST CASE ID: TC003		NAME: Check for Database File
PURPOSE	To allow addition of order details of Crops	
INPUT	Selection of Onsite option in Support type option	
EXPECTED OUTPUT	Allow addition of order details.	
STEPS:		

Select Onsite option in support type in order status form

Add your details and click submit.

6.4.4 Test Case for deleting Log-in Credentials details:

TEST CASE ID: TC004		NAME: Deleting login id	
PURPOSE		delete the account would not hinder the order detail of the farmer as in case of any replacement that detail will be required	
INPUT		Add user id and password for confirmation	
EXPECTED OUTPUT		Deletion of account on our website	

7 CONCLUSION

7.1 Conclusion:

The Healthcare System is Very Helpful to Manage the Time of all users like Doctor, Lab manager and Patient. Also Doctor can see the Patient History Reports and Problems. Lab Manager also have easy to manage appointment and Upload the Reports.

7.2 Bibliography:

Web Resources:

- <https://reactnative.dev/docs/>
- <https://docs.expo.dev/>
- The Complete React Native Course 2021 (<https://youtu.be/ANdSdIlgsEw>)

Reference Books:

- Software Engineering: A practitioner's approach by Roger S.
- Fundamentals of Software Engineering by Rajib Mall.
- React Native: The Complete Reference