Sharda Shaikshanik & Samajik Sanstha's

Vidhydhan Commerce & Science College Walwadi, Dhule(200111)

A PROJECT REPORT ON

"COVID19 Testing Management System"
Submitted By

Mr. Timirash S. Patil

PRN No:-2019015400321725

Seat Number 358269

GUIDE BY

PROF. MRS. LALITA NIKAM

IN PARTIAL FULLFILLMENT OF DEGREE OF
BACHORAL OF COMPUTER APPLICATION (BCA) 21-22

Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon

Certificate

This is to certify that **Mr. Timirash S. Patil** final year student of Bachelor in Bachelor of Computer Applications (**BCA**) has been successfully complete these project entitled "**COVID19 Testing Management System** " as the record of the work carried out by his is accepted in partial fulfillment of the requirement for degree of Bachelor in Bachelor of Computer Applications (**BCA**) in the Bachelor of Computer Applications (**BCA**) vidhydhan Commerce & Science college Dhule under the guidance of **Prof. Mrs. Lalita Baburao Nikam** in the academic year 2021-2022.

Place:- Dhule
Date 22/5/22

Prof. L. B. Nikam
(Project Guide)

Head of Department
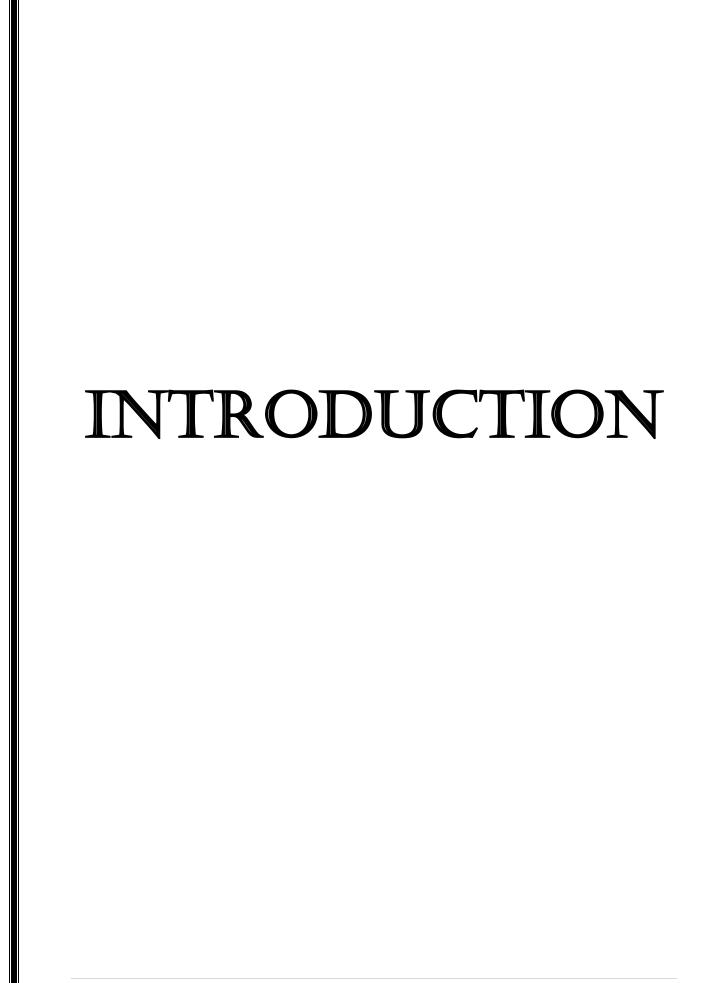Computer & Management
Vidyadhan College, Valwadi

Internal Examiner

Principal
Sharda Educ & Social Sanstha's
Vidyadhan Com.& Sci Colle
Valwadi (Dhule)

External Examiner

2

INDEX

"COVID19 Testing Management System"

Index

INTRODUCTION

Introduction

At the onset of COVID-19 pandemic, the challenge was to standardize the format of sample data from various Tests conducted across the country. ICMR designed the Specimen Referral Form (SRF) for use with every COVID-19 sample; NIC developed the RT-PCR and RATI mobile apps on Android, iOS and Windows mobiles for ICMR along with the web-portal to whitelist phlebotomists (sample collectors) for using the mobile apps and web-portal.

RT-PCR App is a handheld tool for the Medical Staff at Sample Collection Centres spread across the country. The Sample Collection Facility will be sending the sample for various type of specimen to ICMR labs conducting the RT-PCR test for confirmation of COVID-19. Advance intimation is being shared through the App with ICMR. On successful saving any sample, the Collection Centre and the User can view the collection details.

Salient features of the system include:

- Robust, reliable Cloud infra, with redundancy, zero down-time and region-wise Databases with responsive portal and apps, tested for 50,000 concurrent users

- Single sign-on using Government official email IDs and Mobile number-based access for Labs (including Private), Collection Centres and Sample Collectors

- Extensive training material, videos, FAQs

- SRF data accessible on real-time basis by ICMR Labs. Data Analytics, Auto Alert SMS/ Emails and GIS integrated

- Option to enter offline data after generation of SRF-ID

- Configurable- Skip Patient OTP for verification and warning on multiple use of same patient mobile

- RT-PCR, Rapid Antigen and Rapid Antibody tests covered

REQUIREMENT ANALYSIS

Requirement Analysis

- Project Name   COVID 19 Testing Management System

- Language Used   PHP5.6, PHP7.x

- Database   MySQL 5.x

- User Interface Design   HTML, AJAX,JQUERY,JAVASCRIPT

- Web Browser   Mozilla, Google Chrome, IE8, OPERA

- Software   XAMPP / Wamp / Mamp/ Lamp (anyone)

Hardware Requirement

♦ Hardware Requirement

- P-III or higher Processor.

- 40 GB HDD

- 256 SDD

- 8 GB RAM (min)

- Color VGA 800x600 resolution monitor

Feasibility study

A feasibility study specifies whether the proposed software project is practically possible or not. Whenever there arises a need for any software you don't directly jump in developing the particular software. Instead, we must first analyze certain facts to realize whether the software is worthwhile or not and this is called the feasibility study.

The feasibility study determines whether the software project can be achieved technically, organizationally, economically or not, and there are many other parameters to consider. Well, each organization performs the feasibility study in different ways. Some do it elaboratively and systematically, some might do it in an ad hoc fashion and some may not do it all.

Content: Feasibility Study

1. Importance
2. Types
3. Benefits

Importance of Feasibility Study

The feasibility study is of much more importance if done correctly. If the feasibility study is not done correctly it is of no use. Key points to determine the importance of feasibility study are:

Feasibility Study

✓   It prevents you from committing, investing your resources and capital for an impractical project.

✓   It might discover new ideas that might completely change the scope of your software.

✓   Improves the confidence of the team to concentrate on their project.

✓   Validate the need for the software.

✓   Estimates the risk involved in developing the software and analyzed if they can be minimized.

✓   Types of Feasibility Study

✓   There are three components of feasibility study i.e technical, organizational and economical. We have discussed each type in the section below:

The technical feasibility study determines whether the current technology and technical staff available with the organization will accomplish the software's development or not. The key issues that must be addressed during the technical feasibility study are as follow:

1.   Does the technical team available has knowledge to analyze, design, code for software? And they must also be able to install, operate and maintain the software.

2.   Can the proposed software meet the user's requirement?

3.   Technology is getting advanced with each passing day so the developed software product must sustain with the advancing technology in the future.

4.   Do we have the approved and accepted tools and equipment's to develop the software?

5.   Examine the complexity of the software project. A project with higher complexity must be provided with a high degree of technical skills and experience.

2. Organizational Feasibility

Though the software project is feasible from the technical aspect what if you need to make major organizational changes. It will make things more complex. To check if the software is organizationally or operationally feasible or not we must consider the following key points:

1.   Identify whether the organizational staff is ready to accept the new software or not

2.   Is the organizational staff technical sound to cope up with new technologies?

3.   Does the organization hardware infrastructure capable to support the new software?

Now, if the new software gets the staffs acceptance then it is more like that the software will be successful. If the organization has to make more changes in its policies, structures and standers then the software's failure risk gets increased. Apart from the above considerations, there are some more analyses as discussed below:

- **Competition:** Before we start developing the software we must consider the current trends of the market and the competitive software available in the market. This will help in the development of the software.

- **Uniqueness:** Though there are several competitor software available in the market the newly developed software must be known for its uniqueness. It must offer something that the existing software in the market doesn't offer.

- **Performance:** The performance of the software needs to be traced to whether it is performing financially and technically well corresponding to its requirement.

3. Economical Feasibility

For software to be economically feasible the cost expends to develop the software must exceed its benefits. The cost expends to develop a software includes:

1. An overall analysis of the system and its design.

2. Hardware purchased to develop the software

3. Software required in the development of the software.

4. Training required for the organization personals.

5. Cost invested in Installation of the software.

6. The cost required in operating the software

Now the benefits of the developed software must outweigh the cost required to develop the software. Some benefits can not be measured such as:

- Benefits of reduced labour cost.

- Benefits of faster processing.

- Benefits of improved customer service.

- Benefits of reduced errors.

- Benefits of Feasibility Study

A feasibility study gets you a clear-cut idea of whether the software will be a success even before you invest in the software.

- It increases the focus of the software development team.

- The software developer team may come across new opportunities.

- Confirms the developer team whether to accept the software project or not.

- Helps in evaluating the success rate of the software.

So, this is all about the feasibility study. We have learned about the feasibility study, in brief, we have also discussed its importance. Then we have discussed the types of components of the feasibility study. We have concluded our discussion with its benefits.

PHP

PHP is one of the most widely used server side scripting language for web development. Popular websites like Facebook, Yahoo, Wikipedia etc, and our very own Studytonight, are developed using PHP.

PHP is so popular because it's very simple to learn, code and deploy on server, hence it has been the first choice for beginners since decades.

In this tutorial series we will be covering all the important concepts of Php language from basics to advanced and will also share some ready-to-use, useful code sinppets for beginners to kickstart their web development project.

PHP stands for **Hypertext Pre-Processor**. PHP is a scripting language used to develop static and dynamic webpages and web applications. Here are a few important things you must know about PHP:

- PHP is an Interpreted language, hence it doesn't need a compiler.

- To run and execute PHP code, we need a Web server on which PHP must be installed.

- PHP is a server side scripting language, which means that PHP is executed on the server and the result is sent to the browser in plain HTML.

- PHP is open source and free.

If you are still confused about whether you should learn PHP or is PHP the right language for your web project, then here we have listed down some of the features and usecases of PHP language, which will help you understand how simple yet powerful PHP scripting language is and why you should learn it.

1. PHP is **open source** and **free**, hence you can freely download, install and start developing using it.

2. PHP has a very **simple and easy to understand syntax**, hence the learning curve is smaller as compared to other scripting languages like JSP, ASP etc.

3. PHP is **cross platform**, hence you can easily develop and move/deploy your PHP code/project to almost all the major operating systems like Windows, Linux, Mac OSX etc.

4. All the popular **web hosting services support PHP**. Also the web hosting plans for PHP are generally the amongst the cheapest plans becasue of its popularity.

5. Popular Content Management Systems like **Joomla**, **Drupal** etc are developed in PHP and if you want to start your own website like Studytonight, you can easily do that with PHP.

6. With PHP, you can create static and dynamic webpages, perform file handling operations, send emails, access and modify browser cookies, and almost everything else that you might want to implement in your web project.

7. PHP is **fast** as compared to other scripting languages like JSP and ASP.

8. PHP has in-built support for **MySQL**, which is one of the most widely used Database management system.

These are some of the main features of PHP, while as you will learn the language you will realise that apart from these features,

Uses of PHP

To further fortify your trust in PHP, here are a few applications of this amazing scripting language:

1. It can be used to **create Web applications** like Social Networks(Facebook, Digg), Blogs(Wordpress, Joomla), eCommerce websites(OpenCart, Magento etc.) etc.

2. **Comman Line Scripting**. You can write PHP scripts to perform different operations on any machine, all you need is a PHP parser for this.

3. **Create Facebook applications** and easily integrate Facebook plugins in your website, using Facebook's PHP SDK. Check this link for more information.

4. **Sending Emails** or building email applications because PHP provides with a robust email sending function.

5. Wordpress is one of the most used blogging(CMS) platform in the World, and if you know PHP, you can try a hand in **Wordpress plugin development**.

HTML

Html stands for Hyper Text Markup Language. Html mainly use for design a client side web pages, this is a static page it means you can only view Html page not give request and not get response from server. Using Html pages browser get user information through form (This is an Html element). Html provides so many elements (like <p>, <img>, <h1>) to design a web page.

Hypertext: is text which contains links to other texts. The hypertext pages are interconnected by hyperlinks, when mouse click on these link which brings you to a new webpage

Markup language: Any tag based language known as markup language, for example gml, sgml, html, xml etc.

Tim Berners-Lee is known as father of Html. The first publicly available description of HTML was a document called "HTML Tags", first described, on the Internet by Berners-Lee in late 1991.

Features for Html

- ❖ Html is a static page.

- ❖ Html is not a case sensitive.

- ❖ Html is a error free language.

- ❖ Html is simple and easy language.

- ❖ It provides facilities to add audio, video, image on web pages.

- ❖ Html is platform independent language, it can be run on any platform like window, linux, Mac.

- ❖ Each and every elements of html should be enclosed within the angular brackets (<>).

- ❖ Html programs are executed by the interpreter of the browser software.

- ❖ Html program save with **.htm** or **.html** extension.

- ❖ The current version of Html is **Html 5.0**

Ajax

Ajax stands for **Asynchronous JavaScript and XML**. The word "asynchronous" means that the user need not wait until the server replies.

AJAX is no programming or script language, no new invention and no separate Web service, module or plug-in. It is a group of inter-related technologies like javascript, dom, xml, html, css etc. It is an algorithm with 'old' technologies similar to the Dynamic Html.

Ajax allows to create server connections in the background while a user is interacting with a Web front-end. These connections can be created asynchronously, which means that the user need not wait until the server replies.

JQuery

Jquery is the library of JavaScript. It is simple and easy to learn. For perform any task jQuery required less code compare to javascript. For learning jquery's you need good knowledge of Html, css and javascript.

Jquery is created by **John Resig** in **2006** with a nice motto - Write less, do more.

There are lots of other JavaScript frameworks and libraries available, but jQuery is the most popular, and also the most extendable. It is very simple and easy to learn. For perform any task Jquery required less code compare to javascript. Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM

Javascript

JavaScript is a object-based scripting language and it is light weighted. It is first implemented by Netscape (with help from Sun Microsystems). JavaScript was created by Brendan Eich at Netscape in 1995 for the purpose of allowing code in web-pages (performing logical operation on client side).

It is not compiled but translated. JavaScript Translator is responsible to translate the JavaScript code which is embedded in browser.

Netscape first introduced a JavaScript interpreter in Navigator 2. The interpreter was an extra software component in the browser that was capable of interpreting JavaSript source code inside an HTML document. This means that web page developer no need other software other than a text editor of develop any web page.

Using HTML we can only design a web page but you can not run any logic on web browser like addition of two numbers, check any condition, looping statements (for, while), decision making statement (if-else) at client side. All these are not possible u

Where it is used?

It is used to create interactive websites. It is mainly used for:

- Client-side validation
- Dynamic drop-down menus
- Displaying date and time
- Build small but complete client side programs .

- Displaying popup windows and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box)

- Displaying clocks etc.

 using HTML So for perform all these task at client side you need to use JavaScript.

XAMPP

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

What is XAMPP?

XAMPP is an abbreviation where *X stands for Cross-Platform, A stands for Apache, M stands for MYSQL*

, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server,

, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl

is a programming language used for web development, PHP

is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components.

MySQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals.

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

Our MySQL tutorial includes all topics of MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

What is Database?

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc. In this section, we are going to focus on MySQL mainly.

What is MySQL?

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is **My Ess Que Ell**. *However, you can pronounce it in your way.* Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS,

etc. with C, C++, and Java languages.

MODULE DESCRIPTION

Modula Description

It has three modules

1. Admin
2. User(Patients)

Admin Module

Admin is the superuser of the website who can manage everything on the website. Admin can log in through the login page

- **Dashboard:** In this section, the admin can see all detail in brief like the total, assigned and the sample collected and completed tests.

- **Phlebotomist:** In this section, the admin can manage Phlebotomist (add, update, delete).

- **Testing**: In this section, the admin can manage all the tests like assign the test to Phlebotomist and update the history.

- **Report:** In this section, the admin can generate two types of report. One is between dates reports and another one is by search. Admin can search the report by order number, name and mobile number.

- **Notification:** In this section, the admin will get a notification for every new test request (notification bell).

- Admin can also update his profile, change the password and recover the password.

User(Patient) Module

- User can visit the application through a URL.

- **Testing:** This section divided into two parts. One is for new user and another one is for registered user. New user(First-time user) needs to provide personal and testing Information. A registered user only needs to provide test information, their personal information will be fetched from the database.

- **Report:** In this section, Users can search their test report using order number, name and registered mobile number.

- **Dashboard:** In this section, the User can see the in which State of how many tests are done.

DESIGN

Design

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.
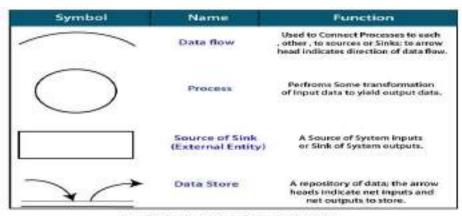
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.

2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.

3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.

4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:

Symbol	Name	Function
(arc)	Data flow	Used to Connect Processes to each, other, to sources or Sinks; to arrow head indicates direction of data flow.
(ellipse)	Process	Perfroms Some transformation of input data to yield output data.
(rectangle)	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
(data store)	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

Symbols for Data Flow Diagrams

Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.
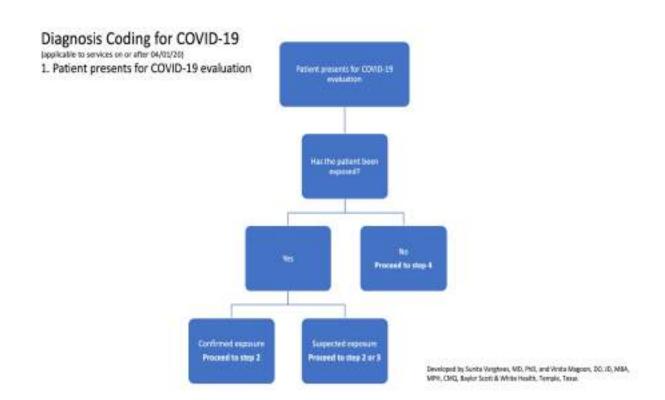
Data Flow: A curved line shows the flow of data into or out of a process or data store.

Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.
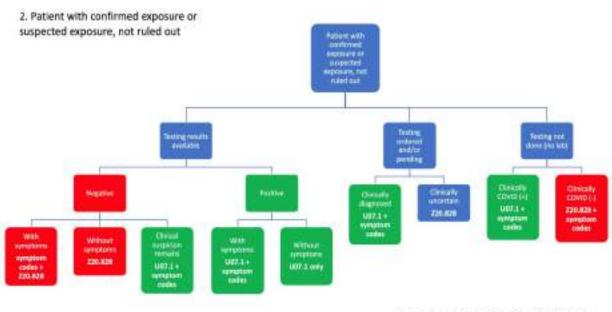
Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

Levels in Data Flow Diagrams (DFD)

The DFD may be used to perform a system or software at any level of abstraction. Infact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.
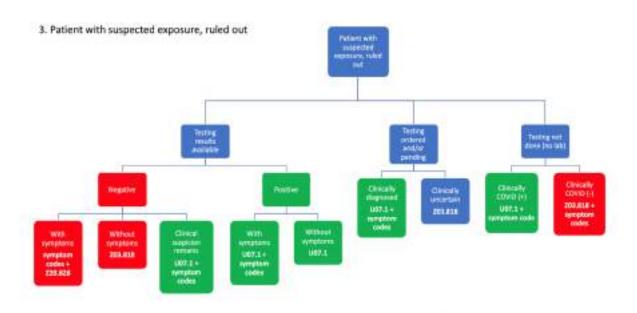
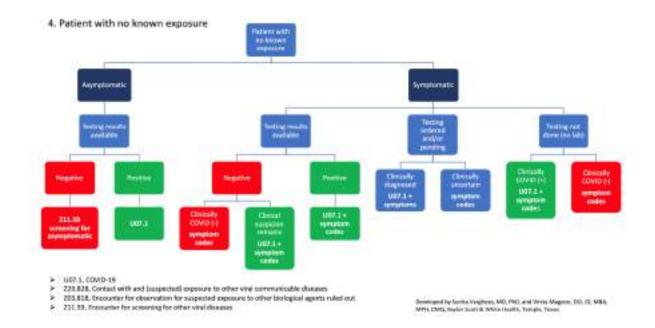## Patient with confirmed exposure or suspected exposure, not ruled out

2. Patient with confirmed exposure or suspected exposure, not ruled out

Patient with confirmed exposure or suspected exposure, not ruled out

- Testing results available
  - Negative
    - With symptoms — symptom codes + Z20.828
    - Without symptoms — Z20.828
    - Clinical suspicion remains — U07.1 + symptom codes
  - Positive
    - With symptoms — U07.1 + symptom codes
    - Without symptoms — U07.1 only
- Testing ordered and/or pending
  - Clinically diagnosed — U07.1 + symptom codes
  - Clinically uncertain — Z20.828
- Testing not done (no lab)
  - Clinically COVID (+) — U07.1 + symptom codes
  - Clinically COVID (-) — Z20.828 + symptom codes

Developed by Sunita Varghese, MD, PhD, and Anita Magown, DO, JD, MBA, MPH, CMQ, Baylor Scott & White Health, Temple, Texas

3. Patient with suspected exposure, ruled out

3. Patient with suspected exposure, ruled out

Patient with suspected exposure, ruled out

- Testing results available
  - Negative
    - With symptoms — symptom codes + Z20.828
    - Without symptoms — Z03.818
    - Clinical suspicion remains — U07.1 + symptom codes
  - Positive
    - With symptoms — U07.1 + symptom codes
    - Without symptoms — U07.1
- Testing ordered and/or pending
  - Clinically diagnosed — U07.1 + symptom codes
  - Clinically uncertain — Z03.818
- Testing not done (no lab)
  - Clinically COVID (+) — U07.1 + symptom code
  - Clinically COVID (-) — Z03.818 + symptom codes

Developed by Sunita Varghese, MD, PhD, and Anita Magown, DO, JD, MBA, MPH, CMQ, Baylor Scott & White Health, Temple, Texas

4. Patient with no known exposure

4. Patient with no known exposure

- U07.1, COVID-19
- Z20.828, Contact with and [suspected] exposure to other viral communicable diseases
- Z03.818, Encounter for observation for suspected exposure to other biological agents ruled out
- Z11.59, Encounter for screening for other viral diseases

Developed by Sonita Vaghnini, MD, PhD, and Venita Magann, DO, JD, MBA, MPH, CMIO, Baylor Scott & White Health, Temple, Texas.

ER-Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

Entity Relationship Diagram Symbols & Notations mainly contains three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes. There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

- Rectangles: This Entity Relationship Diagram symbol represents entity types
- Ellipses : Symbol represent attributes
- Diamonds: This symbol represents relationship types

- Lines: It links attributes to entity types and entity types with other relationship types

- Primary key: attributes are underlined

- Double Ellipses: Represent multi-valued attributes

ER Diagram Symbols

Components of the ER Diagram

This model is based on three basic concepts:

- Entities

- Attributes

- Relationships

ER Diagram Examples

For example, in a University database, we might have entities for Students, Courses, and Lecturers. Students entity can have attributes like Rollno, Name, and DeptID. They might have relationships with Courses and Lecturers.

WHAT IS ENTITY?

A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.

Examples of entities:

- Person: Employee, Student, Patient

- Place: Store, Building

- Object: Machine, product, and Car

- Event: Sale, Registration, Renewal

- Concept: Account, Course

Notation of an Entity

Entity set:

Student

An entity set is a group of similar kind of entities. It may contain entities with attribute sharing similar values. Entities are represented by their properties, which also called attributes. All attributes have their separate values. For example, a student entity may have a name, age, class, as attributes.

Example of Entities:

A university may have some departments. All these departments employ various lecturers and offer several programs.

Some courses make up each program. Students register in a particular program and enroll in various courses. A lecturer from the specific department takes each course, and each lecturer teaches a various group of students.

Relationship

Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.

Entities take part in relationships. We can often identify relationships with verbs or verb

Weak Entities

A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.

Ministry of Foreign Affairs Contract

Attributes	
Covid19_TravelGuidelines_Hash	bytes32

Functions
AddMoH(address,uint256)
RevokeMoH(address,uint256)

Ministry of Health Smart Contract

Attributes	
MoFA	address
Covid19Awareness_Hash	bytes32

Functions
AddTestingCenter(address,uint256)
RevokeTestingCenter(address,uint256)

Covid19 Testing Center Smart Contract

Attributes	
MoH	address
Covid19TestInfo_Hash	bytes32

Functions
PublishTestResults(bytes32, address,address,uint256)
PublishUpdates(address,address,uint256,string)

Patient Smart Contract

Attributes	
EA	address
Gender	enum
Name	string
DoB	uint256
Passport_IPFSHash	bytes32
Residency_IPFSHash	bytes32
ID_IPFSHash	bytes32
LastCountryVisitedDetails	struct
LastTestingCenterVisitDocs	bytes32
LastCovid19TestDate	uint256

Functions
Updated(string,uint256)
Attested(address,address,address, uint256)

DATABASE

Database

Collected information which is in an organized form for easier access, management, and various updating is known as a database.

Before going into a further discussion of databases, we must have a prior knowledge of exactly what is a DATA? Data can be defined as a collection of facts and records on which we can apply reasoning or can-do discussion or some calculation. The data is always easily available and is in plenty. It can be used for processing some useful information from it. Also, it can be in redundant, can be irrelevant. Data can exist in form of graphics, reports, tables, text, etc. that represents every kind of information, that allows easy retrieval, updating, analysis, and output of data by systematically organized or structured repository of indexed information.

Containers having a huge amount of data are known as databases, for example, a public library stores books. Databases are computer structures that save, organize, protect, and deliver data.Any system that manages databases is called a **database management system**, or DBM. The typical diagram representation for a database is a cylinder.Inside a database, the data is recorded in a table which is a collection of rows, columns, and it is indexed so that to find relevant information becomes an easier task. As new information is added, data gets updated, expanded and deleted. The various processes of databases create and update themselves, querying the data they contain and running applications against it.

The are several different types of database models have been developed so far, for example, *flat, hierarchical, network* and *relational*. These models describe the operations that can be performed on them as well as the structure of the conforming databases. Normally there is a database schema which describes the exact model, entity types, and relationships among those entities.

Flat Databases have the following characteristics –

- simple
- long and dominant
- useful for very small scale and simple applications.

A **Relational Database** has the following characteristics –

- organizes data such that it appears to the user to be stored in a series of interrelated tables
- used for high-performance applications
- efficient

Types of Databases

Here are some popular types of databases.

Distributed databases:

A distributed database is a type of database that has contributions from the common database and information captured by local computers. In this type of database system, the data is not in one place and is distributed at various organizations.

Relational databases:

This type of database defines database relationships in the form of tables. It is also called Relational DBMS, which is the most popular DBMS type in the market. Database example of the RDBMS system include MySQL, Oracle, and Microsoft SQL Server database.

Object-oriented databases:

This type of computers database supports the storage of all data types. The data is stored in the form of objects. The objects to be held in the database have attributes and methods that define what to do with the data. PostgreSQL is an example of an object-oriented relational DBMS.

Centralized database:

It is a centralized location, and users from different backgrounds can access this data. This type of computers databases store application procedures that help users access the data even from a remote location.

Open-source databases:

This kind of database stored information related to operations. It is mainly used in the field of marketing, employee relations, customer service, of databases.

Cloud databases:

A cloud database is a database which is optimized or built for such a virtualized environment. There are so many advantages of a cloud database, some of which can pay for storage capacity and bandwidth. It also offers scalability on-demand, along with high availability.

Data warehouses:

Data Warehouse is to facilitate a single version of truth for a company for decision making and forecasting. A Data warehouse is an information system that contains historical and commutative data from single or multiple sources. Data Warehouse concept simplifies the reporting and analysis process of the organization.

Table admin

Table patient

Table Phlebotomist

Table testing
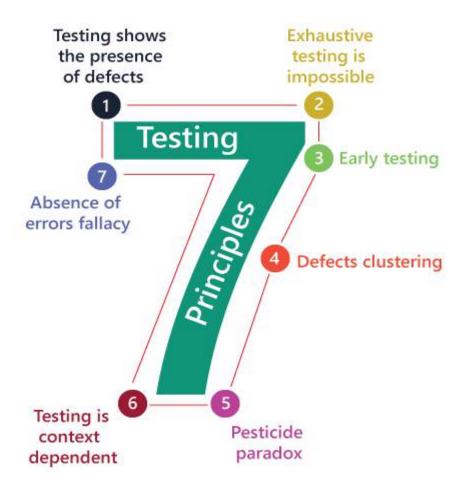

Table test record

TESTING

Testing

Software testing is a procedure of implementing software or the application to identify the defects or bugs. For testing an application or software, we need to follow some principles to make our product defects free, and that also helps the test engineers to test the software with their effort and time. Here, in this section, we are going to learn about the seven essential principles of software testing.

Let us see the seven different testing principles, one by one:

- o Testing shows the presence of defects

- o Exhaustive Testing is not possible

- o Early Testing

- o Defect Clustering

- o Pesticide Paradox

- o Testing is context-dependent

- o Absence of errors fallacy

Testing shows the presence of defects

The test engineer will test the application to make sure that the application is bug or defects free. While doing testing, we can only identify that the application or software has any errors. The primary purpose of doing testing is to identify the numbers of unknown bugs with the help of various methods and testing techniques because the entire test should be traceable to the customer requirement, which means that to find any defects that might cause the product failure to meet the client's needs.

By doing testing on any application, we can decrease the number of bugs, which does not mean that the application is defect-free because sometimes the software seems to be bug-free while performing multiple types of testing on it. But at the time of deployment in the production server, if the end-user encounters those bugs which are not found in the testing process.

Exhaustive Testing is not possible

Sometimes it seems to be very hard to test all the modules and their features with effective and non- effective combinations of the inputs data throughout the actual testing process.

Hence, instead of performing the exhaustive testing as it takes boundless determinations and most of the hard work is unsuccessful. So we can complete this type of variations according to the importance of the modules because the product timelines will not permit us to perform such type of testing scenarios.

Early Testing

Here early testing means that all the testing activities should start in the early stages of the software development life cycle's **requirement analysis stage** to identify the defects because if we find the bugs at an early stage, it will be fixed in the initial stage itself, which may cost us very less as compared to those which are identified in the future phase of the testing process.

To perform testing, we will require the requirement specification documents; therefore, if the requirements are defined incorrectly, then it can be fixed directly rather than fixing them in another stage, which could be the development phase.

Defect clustering

The defect clustering defined that throughout the testing process, we can detect the numbers of bugs which are correlated to a small number of modules. We have various reasons for this, such as the modules could be complicated; the coding part may be complex, and so on.

These types of software or the application will follow the **Pareto Principle**, which states that we can identify that approx. Eighty percent of the complication is present in 20 percent of the modules. With the help of this, we can find the uncertain modules, but this method has its difficulties if the same tests are performing regularly, hence the same test will not able to identify the new defects.

Pesticide paradox

This principle defined that if we are executing the same set of test cases again and again over a particular time, then these kinds of the test will not be able to find the new bugs in the software or the application. To get over these pesticide paradoxes, it is very significant to review all the test cases frequently. And the new and different tests are necessary to be written for the implementation of multiple parts of the application or the software, which helps us to find more bugs.

Testing is context-dependent

Testing is a context-dependent principle states that we have multiple fields such as e-commerce websites, commercial websites, and so on are available in the market. There is a definite way to test the commercial site as well as the e-commerce websites because every application has its own needs, features, and functionality. To check this type of application, we will take the help of various kinds of testing, different technique, approaches, and multiple methods. Therefore, the testing depends on the context of the application.

Absence of errors fallacy

Once the application is completely tested and there are no bugs identified before the release, so we can say that the application is 99 percent bug-free. But there is the chance when the application is tested beside the incorrect requirements, identified the flaws, and fixed them on a given period would not help as testing is done on the wrong specification, which does not apply to the client's requirements. The absence of error fallacy means identifying and fixing the bugs would not help if the application is impractical and not able to accomplish the client's requirements and needs.

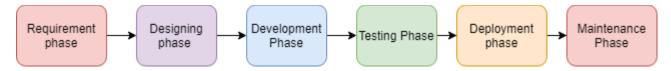**Software Development Life Cycle (SDLC)**

SDLC is a process that creates a structure of development of software. There are different phases within SDLC, and each phase has its various activities. It makes the development team able to design, create, and deliver a high-quality product.

SDLC describes various phases of software development and the order of execution of phases. Each phase requires deliverable from the previous phase in a life cycle of software development. Requirements are translated into design, design into

development and development into testing; after testing, it is given to the client.

Let's see all the phases in detail:

Different phases of the software development cycle

- o Requirement Phase
- o Design Phase
- o Build /Development Phase
- o Testing Phase
- o Deployment/ Deliver Phase
- o Maintenance

1. Requirement Phase

This is the most crucial phase of the software development life cycle for the developing team as well as for the project manager. During this phase, the client states requirements, specifications, expectations, and any other special requirement related to the product or software. All these are gathered by the business manager or project manager or analyst of the service providing company.

The requirement includes how the product will be used and who will use the product to determine the load of operations. All information gathered from this phase is critical to developing the product as per the customer requirements.

2. Design Phase

The design phase includes a detailed analysis of new software according to the requirement phase. This is the high priority phase in the development life cycle of a system because the logical designing of the system is converted into physical designing. The output of the requirement phase is a collection of things that are required, and the design phase gives the way to accomplish these requirements. The decision of all required essential tools such as programming language like Java, .NET, PHP, a database like Oracle, MySQL, a combination of hardware and software to provide a platform on which software can run without any problem is taken in this phase.

There are several techniques and tools, such as data flow diagrams, flowcharts, decision tables, and decision trees, Data dictionary, and the structured dictionary are used for

describing the system design.

3. Build /Development Phase

After the successful completion of the requirement and design phase, the next step is to implement the design into the development of a software system. In this phase, work is divided into small units, and coding starts by the team of developers according to the design discussed in the previous phase and according to the requirements of the client discussed in requirement phase to produce the desired result.

Front-end developers develop easy and attractive GUI and necessary interfaces to interact with back-end operations and back-end developers do back-end coding according to the required operations. All is done according to the procedure and guidelines demonstrated by the project manager.Since this is the coding phase, it takes the longest time and more focused approach for the developer in the software development life cycle.

4. Testing Phase

Testing is the last step of completing a software system. In this phase, after getting the developed GUI and back-end combination, it is tested against the requirements stated in the requirement phase. Testing determines whether the software is actually giving the result as per the requirements addressed in the requirement phase or not. The Development team makes a test plan to start the test. This test plan includes all types of essential testing such as integration testing, unit testing, acceptance testing, and system testing. Non-functional testing is also done in this phase.

If there are any defects in the software or it is not working as per expectations, then the testing team gives information to the development team in detail about the issue. If it is a valid defect or worth to sort out, it will be fixed, and the development team replaces it with the new one, and it also needs to be verified.

5. Deployment/ Deliver Phase

When software testing is completed with a satisfying result, and there are no remaining issues in the working of the software, it is delivered to the customer for their use.As soon as customers receive the product, they are recommended first to do the beta testing. In beta testing, customer can require any changes which are not present in the software but mentioned in the requirement document or any other GUI changes to make it more user-friendly. Besides this, if any type of defect is encountered while a customer using the software; it will be informed to the development team of that particular software to sort out the problem. If it is a severe issue, then the development team solves it in a short time; otherwise, if it is less severe, then it will wait for the next

version.After the solution of all types of bugs and changes, the software finally deployed to the end-user.

6. Maintenance

The maintenance phase is the last and long-lasting phase of SDLC because it is the process which continues until the software's life cycle comes to an end. When a customer starts using software, then actual problems start to occur, and at that time there's a need to solve these problems. This phase also includes making changes in hardware and software to maintain its operational effectiveness like to improve its performance, enhance security features and according to customer's requirements with upcoming time. This process to take care of product time to time is called maintenance.

White Box Testing

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

Black box testing

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.
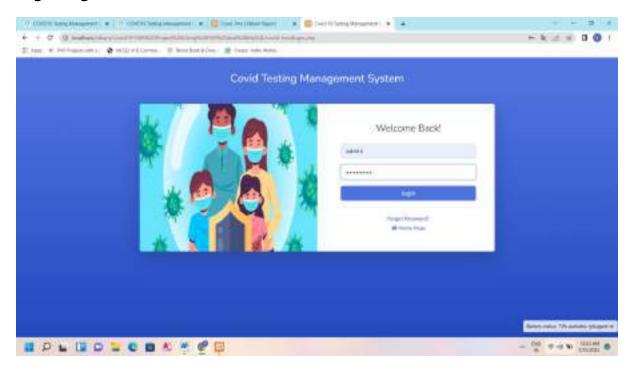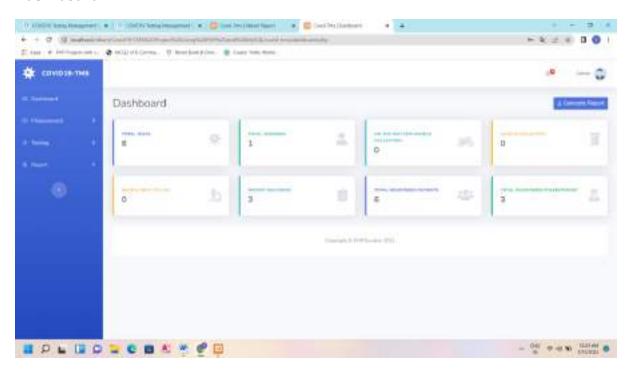

Generic steps of black box testing

- o The black box test is based on the specification of requirements, so it is examined in the beginning.

- o In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.

- o In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.

- o The fourth phase includes the execution of all test cases.

- o In the fifth step, the tester compares the expected output against the actual output.

- o In the sixth and final step, if there is any flaw in the software, then it is cured and tested again.

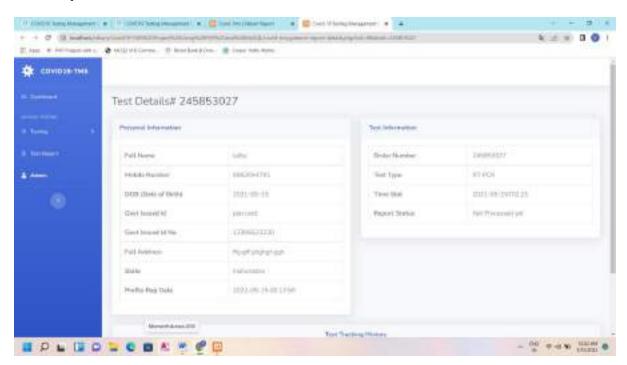# INPUT SCREEN

Login Page

Dash board

Covid Testing

State wise testing

Test Report

Test Report

OUTPUT SCREEN

All test

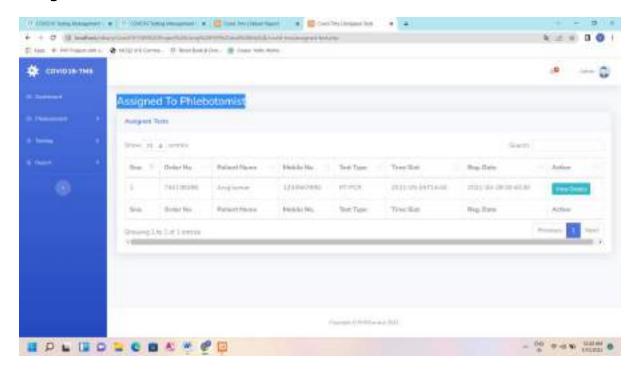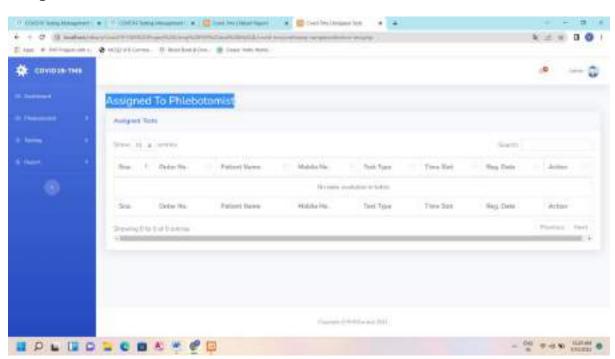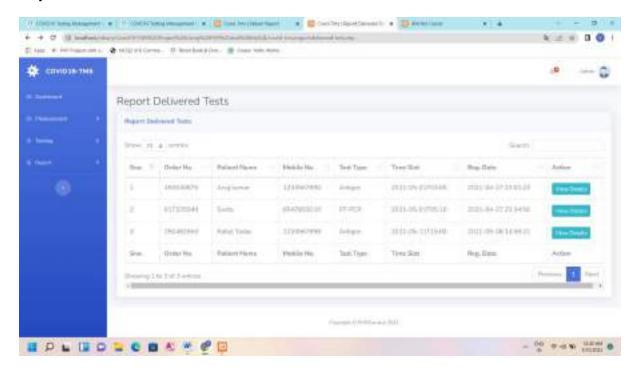**Assigned To Phlebotomist**

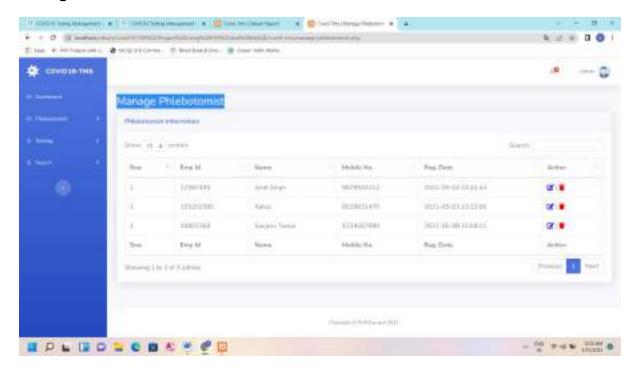**Assigned To Phlebotomist**
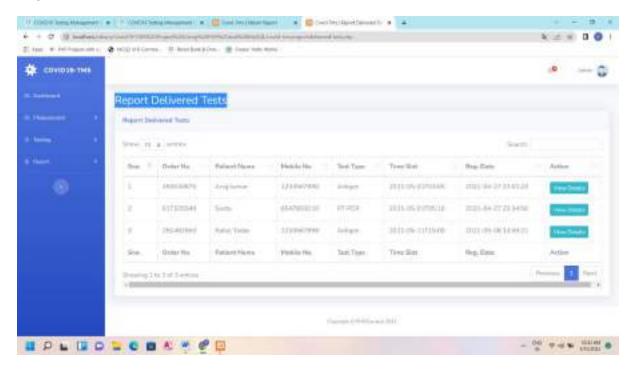
Report Delivered Tests

Manage Phlebotomist

Report Delivered Tests

CONCLUSION

Conclusion

COVID19 Testing Management System is very graceful and lively. Patients have to register to the portal by giving their details and then they can take appointments online with minimal effort. The Phlebotomist comes to the patient address to collect the sample. Once the test is done and the test report is generated, the patient can download the report by logging in to the portal. This system can be implemented in diagnostic labs and clinics.

• Automation of the entire system improves productivity.

• It provides a friendly graphical user interface which proves to be better when compared to the existing system.

• It gives appropriate access to the authorized users depending on their permissions.

• It effectively overcomes the delay in communications.

• Updating information becomes so easy.

• System security, data security and reliability are the striking features.

• The System has adequate scope for modification in future if it is necessary.

The main purpose of COVID19 Testing management system is to provide a platform where patients can book the test online and get their COVID19 test done at home. With the help of this project we are bringing the use of technology in the field of medical diagnosis where patients can avail all the diagnosis facilities at their door steps. Another purpose for developing this application is to generate the report automatically

BIBLIOGRAPHY

Bibliography

[1]. Laboratory testing for Middle East Respiratory Syndrome coronavirus, interim guidance (revised), January 2019, WHO/MERS/LAB/15.1/Rev1/2019, World Health Organization, 2018. (https://apps.who.int/iris/bitstream/handle/10665/259952/WHO-MERS-LAB-15.1-Rev1-2018-eng.pdf;jsessionid=08939A780A5A4552E C8279D0E6D650E4?sequence=1)

[2]. WHO Recommended Surveillance Standards WHO/CDS/CSR/ISR/99.2

[3]. Guideline for the collection of clinical specimens during field investigation of outbreaks WHO/CDS/CSR/EDC/200.4 (https://www.who.int/ihr/publications/WHO_CDS_CSR_EDC_2000_4/en/).

[4]. WHO laboratory biosafety manual, third edition. Geneva: World Health Organization; 2004.