COVID TESTING MANAGEMENT SYSTEM

(SOFTWARE REQUIREMENT SPECIFICATION (SRS))

1.INTRODUCTION:

1.1 PURPOSE

The purpose of this document is to build an online COVID testing management system to manage COVID testing Reports, patients and Phlebotomist to easy access of the system.

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.

DB	Database
DDB	Distributed Database
ER	Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the COVID testing management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the COVID patients and as well as to the Phlebotomist.

1.4 PROJECT SCOPE

The purpose of the COVID testing management system is to ease testing management and to create a convenient and easy-to-use application for patients trying to test the COVID . The system is based on a relational database with its testing management test slot Booking functions. We will have a database server supporting hundreds of major cities around the world .Above all, we hope to provide a comfortable user experience along with the best slots available.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A distributed COVID testing database system stores the following information.

In this project, I used **PHP** and **MySQL** database. It has two 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.

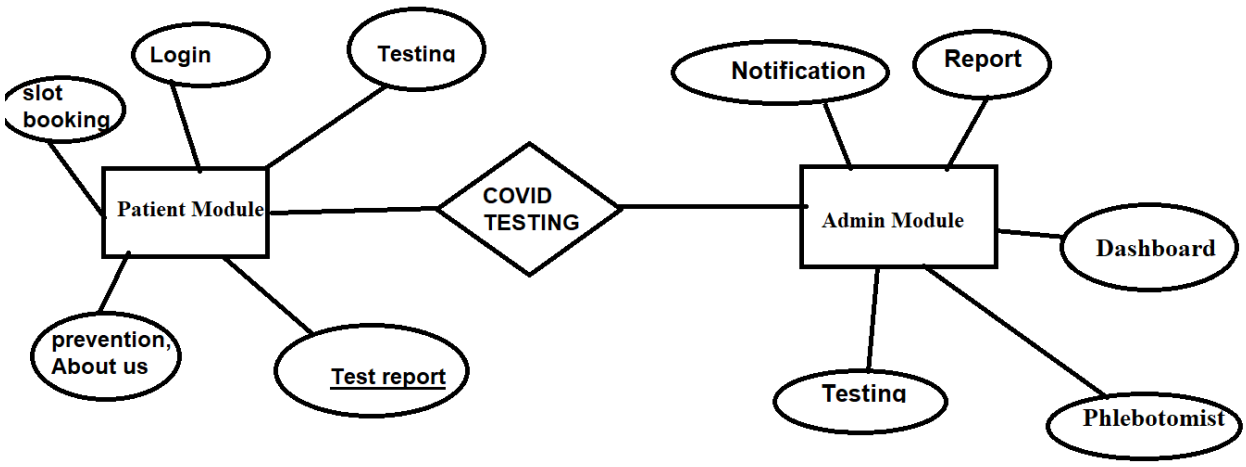
- **Notification:** In this section, the admin will get a notification for every new test request (notification bell).
- Admin can also update his profile, change the password and recover the password.

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.

2.2 PRODUCT FEATURES

The major features of COVID testing management system as shown in below **entity–relationship model** (**ER model**)



The diagram shows the layout of COVID testing management system – entity–relationship model

2.3 OPERATING ENVIRONMENT

Operating environment for the COVID Testing management system is as listed below.

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

2.5 ASSUMPTION DEPENDENCIES

● The functioning of the application relies on MySQL being available on the host machine.

● It is also dependent on a web browser to access the application.

3. SYSTEM FEATURES

- **DESCRIPTION and PRIORITY:**

The COVID Testing management system maintains information on patients, Phlebotomist, and slot bookings.

Of course, this project has a high priority because it is very difficult to take covid test during the pandemic situation.

- **STIMULUS/RESPONSE SEQUENCES**

  - Enter Personal information .
  - Select the test type
  - Book the convenient time slot.

- **FUNCTIONAL REQUIREMENTS**

Other system features include:

- **CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software : Html, css, JavaScript
- Back-end software: php, MySQL

4.2 HARDWARE INTERFACES

- Windows.
- A browser which supports CGI, HTML & Javascript.

4.3 SOFTWARE INTERFACES

Following are the software used for the COVID Testing Mnagement system.

Software used	**Description**
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the patients records, testing records we have chosen MYSQL database.
Front end	Html, css, JavaScript

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers.

5. NONFUNCTIONAL REQUIREMENTS

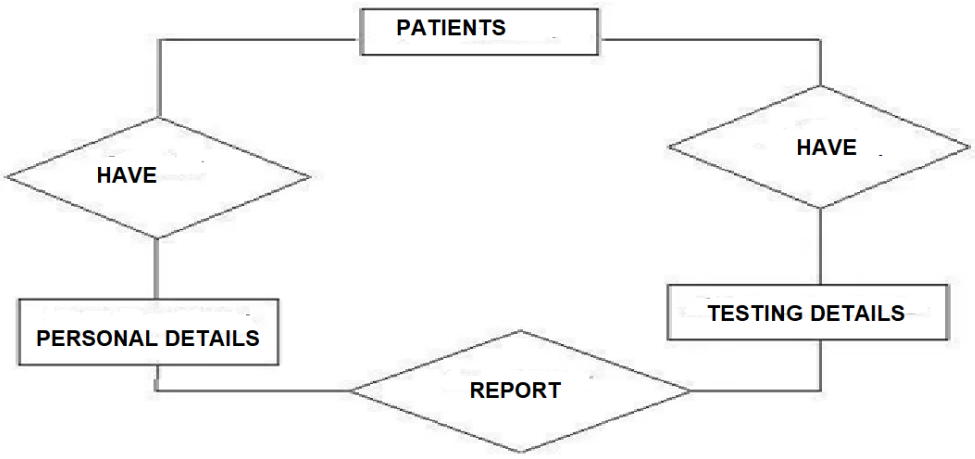
5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of Patients database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.



the diagram shows the ER diagram of Patients database

B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The Testing slot should be available on the specified date and specified time as many patients are doing advance Booking.

- **MAINTAINABILITY:** The administrators should maintain correct schedules of Testing.
- **USABILITY:** The Testing schedules should satisfy a maximum number of patients needs.