**DIAGNOSTIC CENTER CLIENT MANAGEMENT SYSTEM**

**Document Owner – Rhythm Girdhar**

Table of Contents

[1. Overall Description 2](#_Toc22686628)

[1.1. Product Perspective 2](#_Toc22686629)

[1.2. Product Functions 2](#_Toc22686630)

[1.3. User Classes and Characteristics 2](#_Toc22686631)

[1.4. Operating Environment 3](#_Toc22686632)

[2. External Interface Requirements 3](#_Toc22686633)

[2.1. User Interfaces 3](#_Toc22686634)

[2.2. Hardware Interfaces 3](#_Toc22686635)

[2.3. Software Interfaces 3](#_Toc22686636)

[2.4. Communications Interfaces 4](#_Toc22686637)

[3. Functional Requirements 5](#_Toc22686638)

[3.1. Software 5](#_Toc22686639)

[3.2. Hardware 5](#_Toc22686640)

[4. Other Nonfunctional Requirements 6](#_Toc22686641)

[4.1. Performance Requirements 6](#_Toc22686642)

[4.2. Safety Requirements 6](#_Toc22686643)

[4.3. Security Requirements 6](#_Toc22686644)

[5. Use-Case Diagram 7](#_Toc22686645)

# Overall Description

# Product Perspective

While there are multiple diagnostic center management systems present in the market, this product is one of a kind. In such a case, it is required that the product be built from the ground up.

The product is self-contained and will be using multiple templates in order to make the overall design reusable.

The product aims at bridging the gap between diagnostic centers and the general public by providing an easy means of booking appointments.

# Product Functions

The app has multiple functionalities for different roles -

* Booking an Appointment
* Cancellation of an Appointment
* Uploading of Schedule by hospital
* Generation of Report

# User Classes and Characteristics

Users of this product include:

* End-users: These are typically patients who need to book an appointment to get the tests done.
* Client: Typically, diagnostic centers or hospitals who provide the services to the patients.

They are differentiated based on the subset of product functions used since patients will be provided booking services whereas the client will be provided functions to upload schedule and confirm appointments.

* Administrator: To manage the system and related databases.

# Operating Environment

This web application will work on all recent versions of Google Chrome browser, Firefox and Microsoft Edge on Windows and Linux platforms.

# External Interface Requirements

# User Interfaces

* The application allows various users to log into their accounts and select tests, add slots and book appointments at different centers or book an appointment home, and receive the reports.
* Web application shall permit using complete functionality including navigation using the keyboard alone, in addition to using a mouse and keyboard combinations.
* Front-end: HTML, CSS, JavaScript, Bootstrap
* Back-end: PHP and MySQL
* Coding Language: Python

# Hardware Interfaces

* Compatible with any web browser.
* It will be working as a web application on all types of different electronic devices like mobile phones, laptops and tablets.
* If the data increases in volume, with great velocity, we will be shifting to cloud and adopt microservices architecture.
* For now, it requires Windows operating system.

# Software Interfaces

* Operating system: Windows. We have chosen Windows operating system for its best support and user-friendliness.
* Database - The System will communicate with a database through an interface for the following operations:
  + To store all user profile details when they sign up to the platform.
  + To store all the listing details for all the different diagnostic centers and hospitals, with the accompanying pictures, etc.
  + To store the transactions enabled on the platforms like transaction details on booking appointments.
  + To store the details of the details of all the past transactions with different types of users.
* Tools - HTML, CSS, JavaScript (AJAX) and Bootstrap
* Log in and Sign up through electronic forms.
* Add Tests, Book Slots, Show Tests, Show Report, Download Reports are all taken care of, by the UI and the php & MySQL backend together.

# Communications Interfaces

* It is a web application so it should work on all kinds of web browsers and systems like laptop, mobile phones, tablets etc.
* The system shall send an email message to confirm registration with the system.
* The system shall send an email message to the user every time they book an appointment from the platform. The diagnostic center is also sent an email message.
* Push notification to mobile devices is sent by the system when the diagnostic center/hospital uploads the reports
* A notification will be sent by the system whenever the unavailable slots/test types are available later.
* An email message will be sent to the patient and the diagnostic center when verification is started, midway and completed.
* As there are emails- we will be using HTTP, FTP and POP protocols.
* We use electronic forms for registration and booking appointments.

# Functional Requirements

# Software

* Test Selection: The system shall allow users to search for and select tests based on preferences they enter.
* Slot Addition: The system shall allow users to choose a slot.
* Book Appointments: The system shall allow the user to finally book an appointment
* Verification Information: The system shall provide verification information for hospitals, diagnostic centers and patients.
* Payment Portal: The system shall allow a user to make a payment
* Reports: The system should allow the user to see and download reports.

# Hardware

* Minimum Hardware Requirements:
  + CPU - 830MHz
  + Internal Memory - 32 MB
  + Cache - 16 KB
  + RAM - 16 MB
* Prefered Hardware Requirements:
  + CPU - 1GHz
  + Internal Memory - 64 MB
  + Cache - 32 KB
  + RAM - 32 MB

# Other Nonfunctional Requirements

# Performance Requirements

This quality attribute describes the responsiveness of the system to various user interactions with it.

The performance requirements of this project are:

* The front page load time must not be longer than 2secs for people using the website with LTE connection.
* The application must load within 5 secs for people using a 3G or 2G network.
* The system should have a higher throughput performance to perform a large number of transactions (up to 10) within 30 secs.
* The application can support customers up to 1000.
* The application must handle the incoming requests appropriately varying with time such as in typical hours or peak hours.
* The application will be available 24/7 for 365 days in any calendar year.
* Upon submitting a request for creating a new user profile, the profile shall be created within 2 seconds(+/- 0.5 seconds).
* The application shall transmit data at 9600 bits/second.

# 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.
* We follow IEC 61511 / ISA-84-00.01

# Security Requirements

User Authentication Requirements:

* The user is required to login with a username and password.
* The password and the username given to the user must be kept confidentially by the user and must not be shared with anyone at any cost.
* The user will be prompted by the system to change the password once every three months to a new password that hasn’t been used in the last 9 months.
* Unsuccessful attempts must be logged.
* More than three unsuccessful login attempts must be alerted to the user.
* In case of a forgotten password, the new password will be sent to the registered mail-id of the user.

Information Security Requirements

* Vendors must choose their database partner carefully.
* The passwords and the medical records of the user is stored in encrypted form in our database.
* Access permissions for particular system information must be changed by system’s data administrator.

System Security Requirements:

* Suspicious activities like a sudden and unexpected surge in the number of requests will be logged.
* Intrusion detection and prevention is employed to check for malicious attacks on the system.
* Firewalls and anti-virus software must be installed in the systems used by the client.

# Use-Case Diagram

