Clinical Automation System

This document details the requirement specifications for the above-named project. Reach out to your SME / Trainer for any query.

Technology	MEAN / MERN Stack
Document Type	RLL Requirement Specifications
Author	MLA
Current Version	1.0
Current Version Date	18-December-2023
Status	Active

Document Control

Version	Change Date	Change Description	Changed By
1.0	28-September-2021	Document Creation	Shrinivasan Shridharan
1.1	18-December-2023	Document Update	Shabarinath KP

Mphasis Learning Academy

Domain: HealthCare

Project Objective:

Create a dynamic and responsive web application which allows patients to book appointments with doctor/s and manage their appointment schedule

Tools and Technologies:

- **Front-End:** Html, CSS, Javascript, React,Angular etc
- **Server-side:** ExpressJS
- **Back-end:** MongoDB
- **Middleware : Node.js**

Background of the project:

XYZ Ltd is a company which builds a software system catering to various business needs.

XYZ Ltd plans to develop travel based - web application, which enables users of various roles to book, manage appointments with healthcare providers.

The team decided to hire a Full Stack developer who could develop a web application with a rich and user-friendly interface. You are hired as the Full Stack developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

Functional Requirements:

Below are the key responsibilities and functionalities to be implemented in the portal.

The Patient should be able to:

1. Login to system with Captcha
2. Registration with Captcha
3. View doctor visitation and timings
4. Book an appointment with a particular doctor.
5. If visited doctor can send message (DB messaging system) to doctor regarding queries with one week of visitation which doctor will respond
6. Only two queries are allowed for a visitation within one week.
7. View appointment history for self.
   View medicines list or search for a particular medicine form the pharmacy.

Mphasis Learning Academy

The front-office executive should be able to:

1. Login to system
2. Book appointment for a patient.
3. Approve the appointment which are booked by patients online.
4. Can view all patient appointment.
5. Search for appointment for a patient by PatientID or Name

The Doctor should be able to:

1. Login to system
2. View patient appointment approved by front office.
3. View inbox and respond to messages sent by patients
4. View medicines list or search for a particular medicine form the pharmacy

The Pharmacist should be able to

1. Login to system
2. Manage medicines (CRUD) in the database.
3. Search medicines on name.
4. Medicine price will be Price + Tax – discount
5. Give medicine at 10% discount to patients

Common Features

1. Welcome Page
2. Login Screen
3. Registration Page
4. Forgot Password / Reset Password Page
5. All users should be able to View and edit profile.
6. Implement validations wherever required for example: login page, registration page etc.
7. UI-UX should be user friendly.
8. Back button should be disabled after logout / sign-out

Phase 1: Database Schema Design

1. Identify domain objects and their attributes as per the requirement.
2. Create a Database tables with necessary relationship as per the requirement.

Mphasis Learning Academy

Phase 2: Front End Development

Develop web pages as per requirements for the web application using Front-End Technologies.

Phase 3: Back End Development

Develop a RESTful Web API to perform CRUD operations on Domain objects as per requirements and store the data in MongoDB database

Steps to develop a Restful Web API.

1. Identify the domain objects and their attributes as per requirements.
2. Design Database Schema as per requirements.
3. Create Entity class for each domain object with required attributes.
4. Create DAL class for performing CRUD operations for each Entity.
5. Create a Service class to invoke DAO class methods for each Entity.
6. Create a Controller class to build the RESTful Web API using Service class using required annotations.

Phase 4: Unit Testing

1. Perform Unit Testing using postman / swagger / insomnia tools for all the functional requirements.
2. Perform Functional Testing using POSTMAN for all REST end points.

Note:

- Use proper Naming Conventions (package, class and interface, variable names)
- Use Interfaces for loose coupling as and when required.
- Use standard HTTP status codes: 404,500,200,201,401,403 while implementing RESTful Web API

Mphasis Learning Academy