Low Level Design Document

Online Doctor Visit Appointment (Web Application)

Shubham Gantayat, Suraj Kumar

Version 1.0

14/08/2021

Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Responsible Party | Date |
| 1.0 | Initial version | Shubham Gantayat,  Suraj Kumar | 14-08-2021 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## What is Low Level Design Document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Online Doctor Visit Appointment Application. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

# Architecture

The architecture document is available at –

<https://github.com/shubhamgantayat/MedHub/blob/main/Documents/ProcessView.jpg>

# Architecture Description

## Login

First thing anyone will see is the login window, which will ask the user for username and password to login. Sign Up toggler will redirect to signup window.

## Signup

The signup window opens when you click the signup button at the login page. Sign up page is different for both doctor and patients.

## Doctor – Appointments

This page will show the appointments for upcoming 15 days along with patient details.

## Doctor – Patients

This section allows the doctor to see the past patient details whom they have diagnosed.

## Doctor – Take a leave

Doctor can take a leave for a specific period. During that time all the appointments will be rescheduled to the nearest possible opening and the patients will be notified.

## Doctor – Profile

Doctor can visit their profile for setting up payment details and edit their initial information.

## Patient – Home

This section allows patients to search for a doctor by name or speciality and allows them to book an appointment.

This window shows a list of doctors. On clicking doctor image the booking window opens

## Patient – Appointments

This allows patients to see their past as well as upcoming appointments.

## Patient – Notification

This section displays reminder, rescheduling and cancellation of appointments.

## Patient – Profile

Patients can visit their profile for setting up payment details and edit their initial information.

## Admin – Active

Admin can view all the active doctors and can cancel their active license on the site in case of a dispute.

## Admin – Inactive

Admin can activate the account of a doctor after verifying the account details and doing a background check.

# Unit Test Cases

|  |  |  |
| --- | --- | --- |
| Test Case Description | Prerequisite | Expected result |
| Verify whether the Application URL is accessible to the user | 1. Application URL should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads completely for the user when the URL is accessed. | 1. Application URL is accessible 2. Application is deployed | The Application should load  completely for the user when the URL is accessed |
| Verify whether the User can sign up in the application | 1. Application is accessible | The User should be able to sign up in the application |
| Verify whether user can successfully login to the application | 1. Application is accessible 2. User is signed up to the application | User should be able to successfully login to the application |
| Verify whether the user can see the doctor’s dashboard | 1. Application is accessible. 2. User is logged in to application. 3. User is a doctor. | User should be able to see the doctor dashboard. |
| Verify whether the doctor can see the appointments. | 1. Application is accessible 2. User is logged in to application 3. User is a doctor 4. User has set time for appointments 5. User has provided his account information 6. User is activated by admin | Doctor should be able to see appointments. |
| Verify whether the doctor can see patients-section. | 1. Application is accessible   2. User is logged in to application   1. User is a doctor 2. User has set time for appointments 3. User has provided his account information.   6. User is activated by admin  7. User has treated any patients | Doctor should be able to see patients |
| Verify whether doctor can take a leave | 1. Application is accessible  2. User is logged in as doctor | Doctor should be able to see take a leave section. |
| Verify whether doctor can visit profile | 1. Application is accessible  2. User is logged in as doctor | Doctor should be able to see profile section. |
| Verify whether user can see patient dashboard | 1. Application is accessible.  2. User is logged in to application.  3. User is a patient | User can see patient dashboard. |
| Verify whether patient can see home section | 1. Application is accessible.  2. User is logged in to application.  3. User is a patient | Patient can see home section. |
| Verify whether patient can see appointment section. | 1. Application is accessible.  2. User is logged in to application.  3. User is a patient  4. User has booked appointments. | Patient can see appointment section. |
| Verify whether patient can see notification section | 1. Application is accessible.  2. User is logged in to application.  3. User is a patient | Patient can see notification section. |
| Verify whether patient can see profile section. | 1. Application is accessible.  2. User is logged in to application.  3. User is a patient | Patient can see profile section. |
| Verify whether user can see admin dashboard. | 1. Application is accessible.  2. User is logged in to application.  3. User is an admin | User can see admin dashboard |
| Verify whether admin can see active doctors. | 1. Application is accessible.  2. User is logged in to application.  3. User is an admin  4. Active doctors present in database | Admin can see active doctors |
| Verify whether admin can see inactive doctors. | 1. Application is accessible.  2. User is logged in to application.  3. User is an admin  4. Inactive doctors present in database | Admin can see inactive doctors. |