
TEST PLAN DOCUMENT

For

ONLINE MEDICAL CONSULTATION SYSTEM

Prepared by

Bratin Mondal(21CS10016)

Swarnabh Mandal(21CS10068)

Somya Kumar(21CS30050)

Instructors

Prof. Sourangshu Bhattacharya

Prof. Abir Das

CONTENTS

1.0 Introduction.....	4
2.0 References.....	4
3.0 Features to be tested.....	5
3.1 Registration process.....	5
3.2 Login process.....	5
3.3 Patient.....	5
3.4 Doctor.....	6
3.5 Admin.....	6
4.0 Test Plans.....	6
4.1 Home page.....	6
4.2 Registration.....	7
4.3 Login.....	7
4.4 Patient.....	8
4.5 Doctor.....	8
4.6 Admin.....	9
4.7 Logout.....	9
5.0 Approach.....	9
5.1 Methodology.....	10
5.2 Test Types.....	10
5.3 Meetings.....	11
6.0 Environment needs.....	11
7.0 Software risk issues.....	12

1.0 Introduction

The purpose of this test plan is to verify the functionality of the Online Medical Consultation System. The main objective of this plan is to ensure seamless operation of the System application, with accurate information delivery to users, without any data loss or inconsistencies. Emphasis will be placed on testing for potential issues that may impact the application's performance or user experience.

2.0 References

The following documents were referred for the construction of this test plan:

1. https://moodlecse.iitkgp.ac.in/moodle/pluginfile.php/40527/mod_resource/content/1/ieee-829.pdf
2. SRS document that had been submitted by our team
3. <https://www.indeed.com/career-advice/career-development/how-to-write-test-plan>
4. <https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>

3.0 Features to be tested

3.1 Registration process

1. Test the registration process of both patient and doctor.
2. Test whether redirection to respective dashboards function properly.
3. Test whether the welcome message is delivered on successful registration.
4. Test whether the error message(s) are shown on the screen if the user is already registered or if he/she has provided incomplete information.

3.2 Login process

1. Redirection to user dashboard depending on the type of user.
2. Test whether an error message is shown if the user provides incorrect credentials or if he/she has not registered.

3.3 Patient

1. Test whether the patient is redirected to the different pages(profile, appointment booking and appointment history) properly.
2. Test whether the filtering of hospitals on the basis of pin code and date works properly.

3. Test whether the patient is able to book an appointment.

3.4 Doctor

1. Test whether the doctor is redirected to the different pages (profile, appointment history, appointment requests) properly.
2. Test whether the doctor is able to accept/reject appointments successfully.

3.5 Admin

1. Test whether the admin is redirected to the different pages.
2. Test whether admin is able to successfully add or remove hospitals from the list of hospitals that are registered with OCMS.

4.0 Test Plans

4.1 Test plan for home page

1. There are 4 options in the home page : one for patient register, one for doctor register, one for viewing the list of hospitals and lastly for logging in. Clicking on each button should lead to proper redirection.

4.2 Test plan for registration(both doctor and patient)

1. There are two buttons referring to the two different types of sign up (patient and doctor) these buttons will redirect the user to the respective page.
2. User will need to provide all the required credentials
3. Error handling during registration,
 - a.) Possible errors that can occur during registration:
 - User provides an Email ID that is already in use.
 - User leaves a field empty.
 - User provides incorrect type of information like a text in the field of contact number
 - b.) Appropriate messages should be displayed and the user will be redirected to the same sign up page.

4.3 Test Plan for Login Process

1. User will be required to enter their registered email id and password for logging in
2. If incorrect credentials are filled, then user will be shown an error message and asked to enter correct credentials
3. Appropriate error message should be displayed when any of the required fields are not filled.
4. After the login is successful the system checks the user profile and redirects the user to their respective dashboard.

4.4 Test Plan for Patient features

1. The patient after logging in is redirected to the patient dashboard. He/She can then select one of the three buttons(profile, book appointment, appointment history) at the top of the page. The patient should be then redirected to the respective pages.
2. For booking an appointment the patient can choose the date and location(based on pin code), he/she should then be shown the list of hospitals that match the criterias.
3. Upon booking an appointment the patient will be shown a message if and only if the doctor accepts the appointment.

4.5 Test Plan for Doctor features

1. The doctor after logging in is redirected to the patient dashboard. He/She can then select one of the three buttons(profile, appointment requests, appointment history) at the top of the page. The doctor should be then redirected to the respective pages.
2. The doctor should be able to accept appointment requests.
3. The doctor should be able to decline appointment requests.

4. The Doctor should be able to view his/her appointment history.

4.6 Test Plan For Admin features

1. The admin after logging in is redirected to the admin dashboard he/she should be able to then go to the different pages and this redirection should work properly.
2. The admin should be able to view the list of registered hospitals, doctors.
3. The admin should be able to add/remove hospitals and doctors.
4. The admin should be able to accept registration of new doctors.

4.7 Logout

1. The different users have a logout button in their respective dashboards on clicking this the user should be redirected to the home page.

5.0 Approach

5.1 Methodology

In this project, we will adopt the Scrum Development Methodology to achieve agility. We will prioritize continual planning, learning, improvement, team collaboration, evolutionary development, and timely completion. Our approach emphasizes flexible responses to change, with an emphasis on individual and team interactions over processes and tools.

5.2 Test Types

The software testing for this project will be conducted internally by the developers due to the project's small scale. It is not feasible to perform exhaustive testing, therefore, the optimal level of testing will be determined by a risk assessment of the application.

The testing process will include:

1. Unit Testing: This will involve testing smaller units of the software design, focusing on individual units or groups of interrelated units. The programmer will use sample input to observe corresponding outputs.

2. System Testing: The software will be tested on different operating systems, primarily Linux, Windows, and macOS, using the black box testing technique. The emphasis will be on the required input and output, without focusing on internal workings. Special attention will be paid to page protection during security testing.

3. Regression Testing: The addition of new files to the program leads to changes in the program. Therefore, the entire component will be tested to ensure proper functionality, even after adding new components to the complete program.

5.3 Meetings

Test meetings are done after the completion of each substantial part of the project. Emergency meetings are done whenever the necessity arises.

6.0 Environment needs

1. The application does not require any specialized hardware to run.
2. The software will receive test data through demonstration or and user creations.
3. Testing the software requires only a functional email address and internet connection.
4. There are no specific data ranges, but maintaining a database in an online deployment service will lead to additional charges. Additionally, unique data is preferred during testing to avoid errors from repeated data in user creation.
5. Each component of a multipart feature must be tested at least once, by creating mock users, patients, etc.

6. During testing, there are no restrictions on the usage of the web app. Each user session is separate and will be handled simultaneously by the server using PostgreSQL. Therefore, testing the software will not have any impact on other users.

7.0 Software Risk Issues

All packages should be in the versions as mentioned in the requirements.txt file. If that didn't work , you can troubleshoot by updating the packages to their latest version.