TEST PLAN FOR

<<SMART CARE>>

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0	06/11/23	Khushi	Black box and white box testing

1 Introduction

The Smart Care website which aims to solve many problem of patients and doctors i.e it provides the facility of saving the medical reports online. Sometimes when patients changes their doctors, then some of patients lost their prescriptions and are unable to tell doctors about their previous health issues so they can save all of their reports and documents on the platform for their better cure. The platform is also helpful for doctor as doctor can save all their patients history in efficient manner. Doctors can also refer prescriptions of other doctor for betterment.

1.1 Scope

1.1.1 In Scope

The Smart Care- Health based web-app has a wide scope in the upcoming world as smart health have gained significant attention in recent years as an innovative approach to healthcare. This system helps medical professionals in making accurate and timely diagnoses, as well as provide the correct treatment based on patient data. Connecting patients and doctors through a user-friendly interface will make it easier for patients to use in emergency situations. Integrating many platforms under one roof, such that blood availability, medicine availability, path-Labs services and also provide the storage of documents.

A) Functional Requirements

1) **Home page:** The homepage should provide users with a clear and concise overview of the e-healthcare services on the navbar.

2) **Services:** The Smart Care provides different types of services including Path-lab , Blood bank , Health education (Yoga & Gym).

3) **Doctor's profile:** Doctors should be able to manage their profile information, view their patient's medical history, and can see a list of doctors of same field .

4) **Patient's profile:** Patients should be able to manage their profile information, view their medical history, and can see a list of doctors.

B) Non-Functional Requirements

1. **Scalability:** The smart care should be able to handle a large volume of traffic without compromising its performance. It should be able to accommodate a large number of users and handle a high volume of traffic.

2. **Speed:** The Smart care should be fast and responsive. operations should be processed quickly, and users should not have to wait long periods.

3. **Security:** Security is crucial in any e-Healthcare based system. The platform should have robust security measures in place to protect patient's data and prevent unauthorized access of doctors and patients.

4. **User-friendly interface:** The Smart care should be easy to use and navigate. The interface should be intuitive and user-friendly, allowing users to quickly find the services they are looking for.

5. **Reliability:** The Smart care should be available and accessible at all times. Downtime or disruptions can result in lost opportunities for users, and damage the reputation of the platform.

1.1.2 Out of Scope

Out Of Scope defines the features, functional or non-functional requirements of the software that will NOT be tested :

1. **Cloud Computation Offloading: -** Ensure that computational tasks are efficiently offloaded to the cloud for smooth performance on client devices.
2. **Scalability and Cloud Hosting: -** Load testing to ensure the platform can handle increased user loads efficiently when hosted on the cloud.
3. **Custom Knowledge Database: -** Ensure the database provides accurate and reliable descriptions and context to users.

1.2 Quality Objective

Here make a mention of the overall objective that you plan to achive without your testing
Some objectives of your testing project could be :

• Ensure the Application Under Test conforms to functional and non-functional requirements
• Ensure the AUT meets the quality specifications defined by the client
• Bugs/issues are identified and fixed before go live

1) **Usability:**

- User Interface: Evaluate the user interface for simplicity and ease of use.
- Responsiveness: Verify that the website is responsive on various devices and screen sizes.
- Accessibility: Ensure the website complies with accessibility standards.

2) **Performance:**

- Load Testing: Test how the website handles concurrent user loads.
- Response Times: Measure and optimize response times for various actions.
- Scalability: Ensure the website can scale to handle an increased number of users.

3) **Security:**

- Data Encryption: Verify that user data and transactions are encrypted.
- Smart Contract Audits: Conduct security audits of the smart contracts to avoid vulnerabilities.
- DDoS Protection: Implement measures to prevent Distributed Denial of Service (DDoS) attacks.
- Authentication and Authorization: Ensure that user authentication and authorization mechanisms are robust.

4) **Compatibility:**

- Browser Compatibility: Test the website on various web browsers.

5) **Documentation:**

- Ensure comprehensive and up-to-date documentation for users and developers.

6) **Recovery and Backup:**

- Implement recovery mechanisms in case of data loss or system failure.

1.3 Roles and Responsibilities

- QA Analyst – Prof. Harsh Khatter
- Test Manager – Prof. Shreela Pareek
- Configuration Manager – Prof. Neha Shukla
- Developers – Hanu Agarwal, Chhayank Tyagi , Khushi.
- Installation Team – Prof. Harsh Khatter, Prof. Shreela Pareek, Prof. Neha Shukla, Hanu Agarwal, Chhayank Tyagi , Khushi.

2 Test Methodology

2.1 Overview

- Waterfall Model

The waterfall model is a software development model used in the context of large, complex projects, typically in the field of information technology. It is characterized by a structured, sequential approach to software development.
The waterfall model is useful in situations where the project requirements are well-defined and the project goals are clear. It is often used for large-scale projects with long timelines, where there is little room for error and the project stakeholders need to have a high level of confidence in the outcome.

Features of the Waterfall Model:

- The waterfall model involves a sequential approach to software development, where each phase of the project is completed before moving on to the next one.
- The waterfall model relies heavily on documentation to ensure that the project is well-defined and the project team is working towards a clear set of goals.
- The waterfall model places a high emphasis on quality control and testing at each phase of the project, to ensure that the final product meets the requirements and expectations of the stakeholders.
- The waterfall model involves a rigorous planning process, where the project scope, timelines, and deliverables are carefully defined and monitored throughout the project lifecycle.

Overall, the waterfall model is used in situations where there is a need for a highly structured and systematic approach to software development. It can be effective in ensuring that large, complex projects are completed on time and within budget, with a high level of quality and customer satisfaction.

2.2 Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

We aim to test our project at the following levels :

1) Unit Testing: This is the lowest level of testing and focuses on individual components or functions within the software. Developers often perform unit tests to verify that specific parts of the code work correctly.

2) Integration Testing: This level of testing checks how different components or modules of the software work together. It ensures that integrated parts of the software function as intended.

3) System Testing: At this level, the entire system is tested as a whole. It verifies that the software meets its specified requirements and functions properly in its intended environment.

2.3 Test Completeness

Here you define the criteria that will deem your testing complete.
For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

3 Test Deliverables

Here are the deliverables

- Test Plan
- Test Cases
- Bug Reports
- Test Strategy

4 Test Cases

1) Unit Testing

Test Cases	Type Verification	Input	Expected O/P	Actual O/P	Status
1	Data Verification	Enter valid data in all the fields and submit	Successful submission	Successful registration	Success
2	Data Verification	Leave the "Name" field empty in Doctor/ Patient registration	Error is displayed	Unsuccessful registration	Failue
3	Data Verification	Leave the "Email" field empty in Doctor/Patient registration	Error is displayed	Unsuccessful registration	Failue
4	Data Verification	Enter an incorrect "Email" format	Error is displayed	Unsuccessful registration	Failue
5	Data Verification	Password is too short	Error is displayed	Unsuccessful registration	Failue
6	Data Verification	Password does not contains special characters, at least one capital letters and alphanumerics	Error is displayed	Unsuccessful registration	Failue
7	Data Verification	Enter the "Contact" number of more than 10 or less than 10 digits	Error is displayed	Unsuccessful registration	Failue
8	Data Verification	Enter the alphabets in "Contact" field	Error is displayed	Unsuccessful registration	Failue
9	Image Verification	Not adding the Profile picture	Error is displayed	Unsuccessful registration	Failue
10	Image Verification	Adding the Profile picture of more than 1 MB	Error is displayed	Unsuccessful registration	Failue
11	Image Verification	Adding the Profile picture less than 100 KB	Error is displayed	Unsuccessful registration	Failue
12	Image Verification	Adding the Profile picture of objects(like table,chiar,etc)	Error is displayed	Unsuccessful registration	Failue
13	Data Verification	Enter the age whose DOB year is less than 2000	Error is displayed	Unsuccessful registration	Failue
14	Data Verification	Enter zero in "Age" field of Patient Registration	Error is displayed	Unsuccessful registration	Failue
15	Data Verification	Leave the "Specialty" field in Doctor Registration	Error is displayed	Unsuccessful registration	Failue
16	Data Verification	Leave the "Hospital" field empty in Doctor registration	Error is displayed	Unsuccessful registration	Failue
17	Data Verification	While the doctor adds hospitals, enter "Location" and "Contact"	Successful submission	Successful adding	Success
18	Data Verification	Leave the "Location" field in add hospital from Doctor's Dashboard	Error is displayed	Unsuccessful submission	Failue

2) Equivalence Testing

Image uploaded in Patient / Doctor Registration

Partitions	Test Scenario	Expected Outcome
Partition 1	100kB<Image<1MB	Valid
Partition 2	Image<100KB	Invalid
Partition 3	1MB<Image	Invalid

Test Cases	Expected Outcome	Status	Remarks
9	Invalid	Failure	Important to upload profile picture
10	Invalid	Failure	Size is more than 1 MB
11	Invalid	Failure	Size is less than 100 KB
12	Invalid	Failure	Object pic is uploaded

DOB in Patient Registration

Partitions	Test Scenario	Expected Outcome
Partition 1	2000<Year	Valid
Partition 2	Year<2000	Invalid

Test Cases	Expected Outcome	Status	Remarks
13	Invalid	Failure	Year is less than 2000
14	Invalid	Failure	Age cannot be zero

3) Decision Table

Condidtions	Username Incorrect & Password Incorrect	Username Incorrect & Password Correct	Username Correct & Password Incorrect	Username Correct & Password Coorect
Usename (Valid/Invalid)	Invalid	Invalid	Valid	Valid
Password (Valid/Invalid)	Invalid	Valid	Invalid	Valid
Login (Successful/Failure)	Failure	Failure	Failure	Successful Login

4) Boundary Value Analysis

Image uploading

100 KB < Image < 1 Mb

Min = 100 KB
Min + 1 = 101 KB
Max = 1 Mb
Max – 1 = 1023 KB
Nominal = 500 KB

Boundary Values	Test Cases	Outcome	Status
Min = 100 Kb	Profile Picture is 100 KB	Successful uploaded	Success
Min + 1 = 101 KB	Profile Picture is 101 KB	Successful uploaded	Success
Max = 1 MB	Profile Picture is 1 MB	Successful uploaded	Success
Max - 1 =1023 KB	Profile Picture is 1023 MB	Successful uploaded	Success
Nominal = 500 KB	Profile Picture is 500 KB	Successful uploaded	Success

Year in DOB

2000 < Year

Min = 2000
Min + 1 = 2001
Max = 2050
Max – 1 =2049
Nominal = 2025

Boundary Values	Test Cases	Outcome	Status
Min = 2000	Year is 2000, age is calculated	Valid	Success
Min+1 = 2001	Year is 2001, age is calculated	Valid	Success
Max = 2050	Year is 2050, age is calculated	Valid	Success
Max - 1=2049	Year is 2049, age is calculated	Valid	Success
Nominal = 2025	Year is 2025, age is calculated	Valid	Success

3 Resource & Environment Needs

3.1 Testing Tools

List of Tools like

1) Selenium
2) Mentis BT
3) Automation BT

3.2 Test Environment

Following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Office 2013 and above
- MS Exchange, etc.

4 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test