3/12/2024

Testing Plan

Version 1.0

Online Medical Consultation System (OMCS)

Submitted in partial fulfillment of the requirements of Final Project CS 310 Software Engineering

Tuhin Mondal | 22CS10087
Diganta Mandal | 22CS30062
Aritra Maji | 22CS30011

1. Test Plan Identifier

Project Name: Online Medical Consultation System (OMCS)
Test Plan Identifier: OMCS_TestPlan_2024_v7

2. References

- Software Requirements Specification (SRS) document
- Design documents
- Functional and Non-functional requirements documents

3. Introduction

The Online Medical Consultation System (OMCS) is a groundbreaking platform aimed at revolutionizing healthcare delivery by providing accessible and efficient online medical consultations. In the wake of the COVID-19 pandemic, which has strained conventional healthcare systems, the development and implementation of a robust OMCS have become imperative. This comprehensive test plan delineates a meticulous approach to evaluating the OMCS, ensuring its functionality, reliability, security, and performance meet the highest standards.

Our software aims to connect patients and doctors who are living far away from each other. It aims to allow patients to know the closest available doctor to ensure speedy medical care and allows them to book the doctor of their choice.

4. Software Risk Issues

1. **Data Security:** Protecting patient and doctor data from unauthorized access, breaches, and cyber threats to maintain confidentiality and comply with data protection regulations. Allowing users to set weak passwords, insecure data handling practices or data transmission procedures etc are some of the common points where any software may fail.
2. **Performance:** Ensuring optimal system performance to handle a potentially large volume of concurrent users without experiencing degradation or downtime. A massive slowdown at times like epidemics would keep the clients waiting and discourage people from using our platform.
3. **Compatibility:** Validating seamless operation across various platforms, browsers, and devices to maximize accessibility and user experience. The application has to be compatible with all the popular modern browsers so that the viewer has very low chances of having the pages rendered incorrectly to him.

4. **Accuracy of Diagnosis:** Verifying the effectiveness of the system's algorithms in matching patients with suitable doctors based on doctor specialization and location to facilitate effective medical consultation. This is essential as it helps the patient to find the doctor he wants to consult quickly.

5. Features to be Tested

5.1 User Authentication and Profile Management

- **Registration:** Thoroughly validate the registration process to ensure accurate capture of user information and secure storage in the database. Test registration with valid and invalid inputs to ensure robust error handling. No one should be able to log in with improper credentials.
- **Login:** Rigorously test the authentication mechanisms to prevent unauthorized access. Verify password encryption and implement brute force protection measures to mitigate login attempts.
- **Profile Management:** Verify users can update and manage their profiles securely. Test profile update functionalities such as changing passwords, updating personal information, and uploading profile pictures to ensure that they all work as expected.

5.2 Location-based Filtering

- **Location Detection:** Test the system's ability to accurately determine the user's location. Verify location accuracy under different network conditions. Also check whether the software is able to detect the approximate location of the user when he’s in a remote village or rural area not properly documented on most maps.
- **Search Results:** Ensure search results for doctors are relevant and comprehensive based on the user's location. Test filtering algorithms should be able to prioritize doctors in proximity to the user while considering other factors such as specialization and availability. Also, there should not be mistakes like doctors of other specializations getting displayed as this may lead to the wrong doctor getting booked.

5.3 Search for Specialized Doctors

- **Specialization Selection:** Validate users can effectively filter doctors based on medical specialization. Test the presentation of a comprehensive list of medical specializations and ensure users can select their desired specialization accurately.
- **Search Accuracy:** Verify search results align with the user's chosen specialization and location. Test algorithms to accurately match users with doctors specializing in the selected medical field and available within the user's vicinity.
- **Special Situations:** Test corner cases like whether the displayed information is correct even if no doctor is available for a particular specialization.

5.4 Appointment Scheduling

- **Appointment Request:** Test the appointment scheduling process to enable users to request appointments with preferred doctors. Validate the user interface to ensure it allows the patient to select the date and the doctor to select the time.

- **Doctor Availability:** Verify the system accurately displays the availability of doctors based on their schedules. Test appointment slot allocation algorithms to ensure users can only schedule appointments during days when the selected doctor is available.
- **Appointment Cancellation:** The patient should be allowed to cancel an appointment until a doctor has not confirmed it. Once the doctor confirms an appointment, the patient should not be able to cancel it any more.

5.5 Doctor Availability Management

- **Working Hours Update:** Test the functionality for doctors to update their availability for appointments. Verify doctors can specify their working hours for each day of the week and make adjustments as needed.

5.6 Appointment Management

- **Pending Appointments:** Comprehensive evaluation of the system's ability to manage pending appointments. Test the user interface for doctors to view a list of pending appointments, review patient details, and take appropriate action, such as confirming or rescheduling appointments.
- **Prescription Issuance:** Test the process of issuing online prescriptions or scheduling physical appointments following consultation. Verify doctors can generate prescriptions electronically or recommend physical appointments as necessary, ensuring proper medical care for patients.

5.7 Feedback System

- **Messaging Functionality:** Thorough assessment of the functionality of the feedback messaging system between doctors and patients. Test secure communication channels, message delivery, and ensure users can ask questions, provide feedback, and receive follow-up instructions.
- **Feedback Period:** Verify the messaging system remains active for the specified duration following consultation. Test message retention policies and ensure ongoing communication between doctors and patients for any follow-up inquiries or concerns.

5.8 Update Doctor Details

- **Profile Information:** Comprehensive testing of the ability for doctors to update their profile information securely. Test profile update functionalities such as editing specialization, uploading medical certificates, updating contact details, and specifying clinic/hospital location.

6. Approach

6.1 Testing Levels

- **Unit Testing:** Rigorously test individual components and functions to ensure correctness, reliability, and adherence to specifications.

Unit Testing Details:

Unit testing involves testing individual components of the application in isolation.

1. **Doctor's Interface**:
   We have to test the following individual functionalities :
   - Whether doctor can sign up
   - Whether doctor can log in
   - Whether the doctor can view the list of appointment requests
   - Whether doctor can confirm or reject appointments
2. **Patient's Interface**:
   We have to test the following functionalities :
   - Whether patient can sign up
   - Whether patient can log in
   - Whether the patient can send an appointment request to the doctor of his choice
   - Whether he can know the status of his appointment request after it has been confirmed/rejected by the doctor.

- **Integration Testing:** Evaluate interactions between different modules to validate seamless integration and interoperability.

Integration Testing Details

Integration testing involves testing the interaction between different components of the application.

1. **Data Flow**:
   - We basically test whether the information entered at a particular page enters the database correctly
   - We also have to check whether a functionality is able to correctly access information from the database and send the results to the frontend functionalities to be displayed.

2. **Changes in states of components:**
   - The stat es of components should change as we desire them to after various actions like clicking, submission etc. The disappearance and appearance of components of a page should also work fully in control of the backend functionalities as expected.

- **System Testing:** Conduct end-to-end testing of the entire OMCS application to ascertain its functionality, performance, security, and usability. The application should run as desired o all popular and widely used systems to have maximum user base.
- **Acceptance Testing:** Simulate real-world scenarios to gauge user satisfaction and validate compliance with user requirements and expectations.

6.2 Defect Reporting

- **Defect Tracking:** Implement a robust defect tracking system to log, prioritize, and monitor the resolution of reported issues.

- **Severity and Priority:** Assess defect severity and priority accurately to expedite critical issue resolution and maintain system stability.

7. Item Pass/Fail Criteria

- **User Authentication:** Pass if users can register, login, and manage their profiles without encountering errors or security vulnerabilities.
- **Location-based Filtering:** Pass if the system accurately filters doctors based on the user's location, providing relevant search results.
- **Appointment Scheduling:** Pass if users can successfully schedule appointments with doctors, and appointments are allocated based on availability.
- **Doctor Availability Management:** Pass if doctors can update their working hours and emergency availability without issues.
- **Feedback System:** Pass if the messaging system between doctors and patients functions smoothly, facilitating effective communication.
- **Update Doctor Details:** Pass if doctors can update their profile information without encountering errors.

8. Test Deliverables

- **Test Plan Document:** A comprehensive document delineating the testing strategy, objectives, scope, and methodologies.
- **Test Cases Document:** Detailed test cases covering all identified features and functionalities of the OMCS application.
- **Test Execution Reports:** Reports documenting the results of test case execution, including pass/fail status, test coverage, and any issues encountered.
- **Defect Reports:** Logs of reported defects, including descriptions, steps to reproduce, severity, priority, and status updates.

9. Environmental Needs

- **Hardware:** Testing devices encompassing desktop computers, laptops, tablets, and smartphones, representing various platforms, configurations, and screen sizes.
- **Software:** A diverse array of web browsers such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge for comprehensive compatibility testing.
- **Database Server:** A robust, scalable, and high-performance database server hosting the OMCS database to facilitate data storage, retrieval, and manipulation during testing.

- **Internet Connection:** A stable, high-speed, and reliable internet connection to access the OMCS application and conduct online testing activities seamlessly.

10. Staff and Training Needs

- **Testers:** Proficient, experienced, and certified testers with in-depth knowledge of software testing methodologies, techniques, tools, and best practices.
- **Developers:** Competent and skilled developers possessing a deep understanding of the OMCS application architecture, design, and codebase to assist in defect resolution, bug fixing, and troubleshooting.
- **Training:** Comprehensive training sessions, workshops, and seminars for testers and developers to familiarize them with the OMCS application, testing tools, methodologies, and specific test scenarios.

11. Responsibilities

- **Testers:** Responsible for meticulously planning, preparing, executing, and documenting test cases, conducting thorough defect analysis, and providing timely feedback to stakeholders.
- **Developers:** Tasked with promptly addressing reported defects, rectifying bugs, and ensuring the quality, stability, and reliability of the OMCS application.
- **Project Manager:** Overarching responsibility for orchestrating testing activities, allocating resources judiciously, managing timelines, and ensuring strict adherence to the test plan and project objectives.

12. Schedule

- **Test Preparation:** Meticulous review of requirements, design documents, and meticulous creation of comprehensive test cases
- **Test Execution:** Methodical execution of test cases, meticulous logging of defects, and comprehensive reporting of findings
- **Defect Resolution:** Swift resolution of reported defects by developers, followed by rigorous retesting and verification
- **Regression Testing:** Comprehensive retesting of fixed issues and rigorous validation to ensure no new regressions arise

13. Planning Risks and Contingencies

13.1 Delay in Data Collection

- **Risk:** Potential delays in obtaining necessary data for testing, such as user profiles, medical records, and geographical information.

- **Contingency:** Prioritization of testing based on available data, proactive communication with stakeholders regarding delays, and agile adjustment of the testing schedule as required to accommodate data availability.

13.2 Server being down

- **Risk:** The server can be down at a particular instant due to excess users that exceeds its capacity, technical faults etc.
- **Contingency:** An alternate server should always be available to take on the responsibilities of the crashed server to ensure the users don't experience serious problem.

This comprehensive test plan, enriched with detailed descriptions and exhaustive testing scenarios, ensures a meticulous approach to evaluating the Online Medical Consultation System (OMCS). By adhering to this plan diligently, we aim to identify, mitigate, and address any potential issues or risks, thereby ensuring the successful deployment of a reliable, secure, and user-centric OMCS application.