VIRTUAL CONSULTANT

TEST PLAN (IEEE 829 FORMAT)

Version 1.1
<20/10/2021>

Aditya Chandrasekhar (U1923951A) : Project Manager / Lead Developer
Aratrika Pal (U1922069F) : Backend Developer
Shruthi Srinivas (U1923611G) : Backend Developer / Release Engineer
Chong Zhe Ming (U1920757K) : Frontend Developer
Khush Kothari (U1922279J) : Frontend Developer
Kushal Sai Gunturi (U1923232F) : QA Manager
Yi Jia Xin, Joceline (U1920057J) : QA Engineer
Lim Yun Han Darren: (U1921275J) : Frontend Developer

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Yi Jia Xin Joceline	15/10/2021		25/9/2021	Initial Plan and Template
1.1	Yi Jia Xin Joceline	20/10/2021		05/10/2021	Final Test Plan

Table of Contents:

1. Test Plan Identifier

This test plan for Virtual Consultant contains the scope, approach and schedule of testing activities to be carried out for the application throughout its development lifecycle as well as its maintenance phase. This test plan will be the Master plan for the application and contains all relevant information on the resources and procedures of testing for Virtual Consultant. It also contains information in relation to how the testing process is controlled and configured throughout the product lifecycle.

2. Introduction

The Test Plan provides a comprehensive overview of the scope, approach, resources and schedule of all testing procedures and activities for Virtual Consultant. It also highlights the items required for testing, features to be tested, type of testing to be performed, personnel responsible for testing, resources and schedule required to complete testing and the associated risks for the plan.

Four types of testing will be carried out for Virtual Consultant and a black-box approach will be used. The four types are listed as below: -

I. Unit testing

Unit testing will be performed to test the individual units of Virtual Consultant. This will enable us to validate each component of the application independently and this process will be carried out in the development phase of the product life cycle.

II. Integration testing

After performing unit testing, the individual components of the application will be tested together to test the performance of the functionality of the application.

III. System testing

The Quality Assurance team will perform independent testing to test the system as a whole and give feedback to the development team.

3. Test Items

The following items have been identified for testing through the different testing techniques mentioned in the introduction:

3.1 Unit testing

3.1.1 Sign In Navigation

The user should be redirected to the login page when the 'Sign In' button at the top of Virtual Consultant's homepage.

3.1.2 View Chat History

Users must be able to view their chat history with patients/doctors respectively.

3.1.3 Send New Request

The user must be redirected to the consultation request page upon selection of the 'Send a New Request' button in the 'Consultations' page. A consultation request must be sent upon successful submission of the request form.

3.1.4 View Profile

Users must be redirected to their own profile when they select their account icon on the top right hand corner of the application. The user must be able to view their personal information in their profile page.

3.1.5 Create/Delete Posts

Doctor users must be able to create their own posts when they select the 'New Post' button in their 'Posts' page. Doctor users must also be able to delete their own posts if they choose to do so.

3.1.6 Messages

Users (patients and doctors) must be able to send text messages and attachments in their chat window upon successful matching of patients to doctors and vice versa.

3.1.7 Acceptance of Consultation Requests

Doctor users must be able to receive consultation requests from patients and they must be able to accept/reject the incoming consultation requests.

3.2 Integration testing

3.2.1 Authentication

The user (patient and doctor) must be able to log into their respective systems using their credentials.

3.2.2 Feed

The patient user must be able to view all posts from doctors that they have followed in their 'Posts' page.

3.2.3 Communication

Users must be able to communicate with each other seamlessly through text messaging, voice calls and video calls in their chat window upon successful matching.

3.3 System testing

3.3.1 Smoke Testing

Users must be able to login to the application and navigate across all features as designed.

3.3.2 Stress Testing

Multiple users must be able to use the application at the same time.

3.3.3 Scalability Testing

The database should be able to process queries seamlessly without having any problems.

3.3.4 Recoverability Testing

Data stored for any user must be stored in the database if the application crashes.

4. Features to be Tested

The features to be tested are given in the table below and will be given a risk rating which consists of: High, Medium and Low. High level risks have the highest importance and must be tested and debugged as soon as possible. Medium level features are to be tested after high level features are tested and low level features will only be tested after all high level and medium level features have been tested and working properly.

Item	Risk
Authentication	High
Feed	Low
Communication	Medium

5. Features Not to be Tested

All features implemented in the application will be tested since all features are pivotal in the overall function of our application.

6. Approach

A test approach specifies how testing would be performed and consists of two techniques: reactive and proactive. A reactive approach conducts testing after the design and implementation of the code is completed whilst a proactive approach initiates testing as early as possible in order to find and fix defects before the build is created.

We will be following a methodological approach for Virtual Consultant's test plan. A methodological approach consists of all methods which are pre-determined and pre-defined, to carry out the testing activity. The methods covered under this approach are used to test a software product from different perspectives and requirements, ranging from static analysis of the source code to dynamic testing of the application itself. Tests will be carried out as documented under the Test Summary Report document. The Quality Assurance engineers are held responsible for the task of creating test runs. The tester will execute the tests and denote each case as Pass/Fail and record the results down in the Test Summary Report. Should any test cases be marked as Failed, they must be highlighted to the developing team immediately for ramification.

The Quality Assurance engineers will review the test runs in the Test Summary Report and review it with the Quality Assurance Manager before the test runs are marked as Passed. When testing is deemed to be completed by the QA Manager, a test report will be submitted to the project manager.

The following are some factors that were considered when selecting the test approach:

- Nature of the product and domain
- Risks of failure of the team and environment
- Regulatory and legal aspects
- Experience and expertise of the team in the proposed tools and techniques

7. Item Pass/Fail Criteria

This section details the definition if an item has passed or failed its test in the project.

For each test case, the tester will be equipped with the appropriate input and its respective output. The test case will be deemed to Pass only if the result of the test run matches perfectly with the output defined in the test case. If the test run result does not match perfectly with the output, it will be marked as Failed. In any case of discrepancies, where it is not clear if the result matches with the ideal output, the QA Manager will be contacted for a consultation.

The completion criteria for our test plan is as follows:

- 100% of the unit test cases completed
- 90% of the integration test cases completed and <10% of cases having minor defects
- System testing done with at least 10 external testers

8. Suspension Criteria and Resumption Requirements

The suspension criteria lists out the reasons why suspension of the testing process would occur whilst the resumption requirements specifies when testing can resume after suspension.

8.1 Suspension Criteria

Suspension in the software testing process refers to a period where the testing team suspends testing activities based on a criteria. The test team will decide whether to suspend all testing activities in its entirety or just a part of it. Suspension can occur when external components are not readily available or when a major defect has been detected.

For Virtual Consultant, the testing team may choose to suspend any testing processes if there is sufficient reason and documentation to support the suspension. A few common criteria have been identified as shown below:

- Hardware/software unavailability
- Major defects that may negatively impact the testing progress
- Defects in the build
- Connectivity issues
- Unavailable test resources
- Project scheduling prioritises deliverables
- Unexpected output
- Functionality is not working as defined in the System Requirements Specification

Should there be more than 40% of failed test cases, suspension of testing will be triggered until the development team fixes all the bugs. The Lead Developer must be kept in the loop with regards to all failed test cases.

8.2 Resumption Requirements

Resumption refers to the restarting or continuation of the process invoked after the suspension criteria (as listed above) has been met. This process involves verification of the correction of the defect by which suspension was invoked during the testing process.

A list of common requirements for the resumption of the testing process is shown below:

- Hardware or software resources are available as per requirements
- Issue due to which suspension occurs is resolved
- No further defect has been found
- Output meets what is expected

9. Test Deliverables

Test deliverables are the test artifacts which are given to the stakeholders of a software project during the Software Development Life Cycle (SDLC). Some of the deliverables are provided before the testing phase commences and some are provided during the testing phase.

For this project, the deliverables are as follows:

9.1 Test Cases

Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected results, post-conditions and actual results.

9.2 Test Report

The test report contains the test results and the summary of test execution activities.

9.3 Test Plan Document

The test plan document contains the plan for all testing activities to be completed and is derived from the Project Proposal, SRS and Use Case documents of the project. It is usually prepared by the Testing Manager.

9.4 Revision Logs

The revision logs contain the revision history of all documents.

9.5 Defect Logs

The defect logs contain all defects identified during testing as well as the solutions implemented for the aforementioned defects.

10. Test Tasks

The following tasks are the set of tasks to be completed:

I. Prepare test plan
II. Identify items to be tested
III. Identify method of conducting tests
IV. Assign personnel to each test case
V. Conduct testing
VI. Fix bugs and errors
VII. Create defects log
VIII. Create test report

11. Environmental Needs

In order to conduct testing on Virtual Consultant, the following requirements have to be met:

- Linux, Mac OS X, or Windows
- Git
- Visual Studio Code

12. Staffing and Training Needs

Training will be required to set up the application on the testers' machines. The Lead Developer will be the one responsible for the provision of training on how to run the application. The QA Engineers can also consult the front-end and backend developers if they run into any problems. The QA Manager will be the one responsible for finding methods to train the testers.

13. Responsibilities

Role	Responsibilities
Lead Developer	Provide required training in code details. Provide solutions for running the code. Conduct white-box testing on the system.
Project Manager	Decide which features should be tested. Monitor the errors found when collecting and analysing the final testing report.
Release Manager	Collect details of test runs to ensure software meets requirements before deployment.
Frontend Developers	Conduct white-box testing on frontend components.
Backend Developers	Conduct white-box testing on backend components.
QA Engineer	Conduct black-box testing. Report test logs to QA Manager. Generate additional test cases as needed.
QA Manager	State the risk and contingency plan for the different testing phases. Monitor testing activities and ensure all testing resources are available. Set overall testing strategy.

14. Schedule

The testing phase for Virtual Consultant will last for up to 2 weeks, details of which are supplemented in the Project Plan. The activities planned for the testing phase are as follows:

- A daily backlog will be maintained to keep track of the items which are yet to be completed. The QA Manager will religiously monitor the backlog and help the QA Engineers clear the backlogs should they get too large.
- The number of days assigned to each test case is dependent on their risk level. Test cases with High risk will require 3 days, Medium risk will require 2 days and Low risk test cases will require 1 day. The assignment is to ensure that time is not wasted on testing low risk items.
- For any bugs that are discovered, the result shall be informed immediately to the team.

15. Risks and Contingencies

Risk	Contingency
Shortage of testers	Members of the development as well as management team will be deployed to help with the testing process.
Improper training for testers	The QA Manager and Lead Developer will be responsible for the training of all testers.
Improper communication between testers and developers	The daily backlog is to ensure that the testing is kept on track. Testers are required to report any bugs that they discover immediately to the Lead Developer.

16. Approvals

Role	Test Type	Approval Criteria
Project Manager	Overall	Application works as required.
Lead Developer	White Box	System performs in accordance with functional requirements.
QA Manager	Black Box	All test cases are covered.

17. References

Document	Link
Project Plan	https://docs.google.com/document/d/177FZoE_UKM6KDhG6PwX8By5yym5UP28emRq6WYnAGUw/edit
System Requirement Specification	https://docs.google.com/document/d/1uz3NmF5Cf5cTb-JBC9nPleMj_9obNEWgQIeOB8JPF0Q/edit
IEEE Test Plan Standard	https://learn-ap-southeast-1-prod-fleet02-xythos.content.blackboardcdn.com/5dcb73f79ba4c/64997?X-Blackboard-Expiration=1634574600000&X-Blackboard-Signature=URY86tlQGg93OGLZgNXABIZso%2Bn1laUPwy0qdlJgVpA%3D&X-Blackboard-Client-Id=100085&response-cache-control=private%2C%20max-age%3D10800&response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27ieee829Handout.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20211018T133000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=10800&X-Amz-Credential=AKIAZH6WM4PL2RCDY4NT%2F20211018%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=1e95a61699ac89826b4860a3a85f9702441e68f7b18482d78ea0d68c76652c0e