VLOL Master Test Plan

1. Test Plan Identifier

|  |  |  |
| --- | --- | --- |
| Identifier | Author/s | Date |
| Virtual Letter of Life Testing – MTP\_1.0 | M. Allibalogun, R. Garcia, S. Urgessa, M. Debela, C. Baisi, A. Mwamba, J. Bourne | June 27, 2020 |
|  |  |  |
|  |  |  |
|  |  |  |

1. References

List of documentation that will be used to conduct a thorough test plan:

* Project Plan.
* Technical design document.
* Software Requirement Specification (provides Functional and Non-Functional requirements).
* IEEE 829 (Test Plan Template).

1. Introduction

This document will provide on overview test plan for the interoperable Virtual Letter of Life system (VLOL) that will exchange information through the interdisciplinary Emergency Medical Services (EMS), allowing information to be updated and viewed simultaneously by various authorized end users, while also granting patients with the accessibility to continuously update their medical information. This Master Test Plan addresses all the components and elements related to the VLOL. The test plan objectives are:

* define the tools that will be used to conduct testing throughout the SDLC of the application
* assigns roles and responsibilities
* provide a schedule for testing with deadlines
* establish expectation for both tester and stockholders
* list all elements needed for the environment to function.

The application must be delivered in 11 weeks; thus, the testing approach will be done parallel between each test levels. The strict timeline doesn’t warrant any delay, for it will severely set back the test plan schedule. The EMPlus team is under the assumption that all third parties will ensure that their services/products will not hinder the test plan schedule. The test plan will have four levels of testing:

* Unit test (J-Unit)
* System & Integration Testing
* Acceptance Testing (Contextual Inquiries)
* Automation Testing (Selenium)

1. Test Items

The following items are products that will be tested. The list provides names, versions, and releases.

1. Window Server 2019
2. Heroku or Microsoft Internet Information Services (IIS) 10.0
3. JDK and JRE 11.0.7
4. Maven 3.6.3
5. UMGC GitHub Repository
6. Microsoft Server 2019
7. Selenium

1. Software Risk Issues

Majority of the software risk that could impact the project where listed in Project Plan Risk Management section. All the risks were categorized as low-level risk. Below a list of those risk that pertain to testing.

* System incompatibility when deployed on the client server (Salisbury IT Department)
* Inability for the users to access their personal account and perform update/deletion/creation
* The application unable to read the barcode when printed of when its scanned
* The system fail to generate a barcode when the user creates an account

1. Features to be Tested

Below is the outline of features that will be tested within those test unit levels. Additionally, we’ll have patch management plan to address issues that we did not discover in testing.

* Unauthenticated and Unauthorized User
  + Access landing page
  + Access registration page
  + Access contact page for both general information and Sys Admin about issues
* Program Participant
  + Create an account as program participant
  + Access to VLOL login page
  + View, add, edit, delete letter of life information
    - Allergies
    - Condition
    - Medication
    - Contact information
    - Mailing address
  + Generate QR (quick response) barcode
  + Print barcode
  + Access contact page for general information, Sys Admin about issues, and Patient Advocate
* Medical Services Provider
  + Access to Letter of Life system
  + Search and view patient through search engine
  + Retrieve patient medical information history
  + Search and view patient medical information by scanning bar code
  + Access contact page for both general information and Sys Admin about issues
* Patient Agent
  + Access Letter of Life system login page
  + Edit “My Profile” page to ensure patient information is up to date
  + Search patient by scanning bar code
  + Access contact page for both general information and Sys Admin about issues
* System Administrator
  + Access VLOS system page
  + Search for user account via laptop or mobile device
  + View, add, edit, delete user accounts
  + Grant limit permission to Medical Service Provides and Patience Advocates
  + View, add, edit, delete Medical Service Provides and Patience Advocates accounts
  + View system error log
  + View system activity log
  + Access user guide

1. Features Not to Be Tested

List of features that will not be directly address or tested for the reminder of the next 11 weeks:

1. Network security
2. Approach

8.1 Testing Levels

As stated in the introduction section, testing for the VLOL application will involve Unit, System/Integration, Acceptance, and Automation testing. For each level of testing there will be a designated tester that will test particular section of the application (classes, interface, et) and a second personal that will play as an overseer. The time restriction for the completion of this application may require for additional help, and that’s where project manager will assist as test managers to ensure full test coverages are completed.

UNIT TESTING – developers will utilize the JUnit framework to conduct the back-end testing. Tester will utilize a Whitebox documentation to create test cases and keep track of all the testing that are being executed. The test person may also utilize this form as a prof of the testing by documenting sample output, data printout, and bug found outside the list of test cases. To keep track of bugs and assigned them, project manager will utilize Jira to assign bugs to developers.

SYSTEM/INTEGRATION – the system/integration will require a collective team effort between the front end, back end, and DevOps Team. The product owner will be playing the role of development team leader and all the developers will assist. It should be noted that no system integration will be performed until all the JUnit test cases have passed and the majority of bugs found have be mitigated. Ultimately, as long as no major defect impede the application from working, a system/integration will be executed at the end of every sprints.

ACCEPTANCE TESTING – the acceptance testing will require the participation of the Client and a user who will play the role of “Patient” while being supervised by the Product Owner. The product owner will observer and analyze the interaction between the users and the application and determine the acceptance by following the 10 Usability Heuristics by Jakob Nielsen.

AUTOMATION TESTING – developers will use Selenium to conduct automating testing on the application is run on the web browser. The implementation of this tool is warranted and will ensure the application will be ready by the deadline. With automation testing we’ll increate test coverages, the speed of test execution, as well as saving time.

8.2 Test Tools

a. Selenium 3.141.59

b. JUnit 4.13

c. Jira 8.10.0

d. OWASP Top 10

e. SonarQube 8.3.1

8.3 Meetings

Meetings between testers and developers will take place on a weekly basis via Zoom. Those meetings have been scheduled and can be found in the WBS calendar. The Product Owner may request additional meetings for testing for reasonable reasons, as long as the PMs are notified.

1. Item Pass/Fail Criteria

The criteria used to determine if each test item has passed or failed testing is categorized by the different test levels. The ultimate goal it to have between 90-100% test coverage for all functionalities. That entails that all JUnit test cases created pass testing, the SonarQube prevents the duplications and potential bugs that may not be detected when creating JUnit test cases, and for Selenium removing manual redundancy and human prone errors.

1. Suspension Criteria and Resumption Requirements

The testing activities will be put into alt when the below scenario occur:

* 1. All actors are unable to reach the website via browser or mobile application
  2. All actors are unable to perform certain execution on the application such as updating information, patient unable to add a person as a patient agent, medical provider unable to scan patient barcode via the app, or the system administrator unable to grant users action on their accounts
  3. Failure to access database information

Once those events have been resolved by the development team testing will resume, however the protocol in this case will require that all functionality must be retested and the Product Owner will be responsibility to schedule a meeting with the client to ensure that everything is working properly.

1. Test Deliverables

List test deliverables, and links to them if available, including the following:

Test Plan (this document itself)

Test Cases (White Box)

Test Scripts

Defect/Enhancement Logs

Test Reports

1. Test Environment

Hardware: Windows or Mac PC

Software: IDE (NetBeans or IntelliJ), JUnit, Selenium, and SonarQube

Network:

1. Environmental needs
2. Staffing and Training Needs

Our short man power does not permit to have one specific tester assigned for all the four level of testing. Our resolution is to have both back-end developers split the unit test (Junit) by having one of the PM conducting testing and the Product owner overseeing each test cases. The front-end team will conduct the automation testing via Selenium. The contextual inquiry will be conduct by the Product Owner as he analysis the interaction between the application and all external members who will utilize the VLOL system (Client, Consultant, and EMTs). Lastly, both backend and front-end developers will work alongside the DevOps team to ensure that system integration functionality are being tested.

The successful completion and testing execution are predicated by having each member trained on those test areas. Here are the test training requirements:

* Back end developers will need training and acquire knowledge on Junit and Maven
* Front end developers will need training and acquire knowledge on Selenium.
* The EMTs will require training on the new screens and reports
* At least two members must have an idea of the requirements that must be met when conducting the contextual inquiry

1. Roles & Responsibilities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Client | PM | Dev Team | Test Team | DevOps  Team |
| Documentation and Execution test acceptance | X | X |  | X | X |
| Unit test documentation and execution |  | X | X | X |  |
| System Design reviews |  | X | X | X |  |
| Detail Design Reviews | X | X | X | X |  |
| Test Procedures and rules |  | X | X | X | X |
| Test Report |  |  |  |  | X |
| Screen prototype reviews | X | X | X | X |  |
| Regression testing |  | X | X | X | X |
|  |  |  |  |  |  |

The above table provides a clear view of each members role and responsibility in the testing phase. The Lead development member will be in charge to ensure that all unit test cases from the test plans and documentation are being met. The project managers will write and update the test plans and documentations. All members of the project, including stakeholders and clients, will take part of the application system review. In this review all members will have the final says over any new change request or updates. The DevOps team will provide the Dev Team test report whenever any new feature is being pushed to the main repository. The test report will be generated by one of the DevOps pipelines (i.e. Jenkins). The client and few EMT members will take part in the final execution acceptance test plan.

1. Schedule

The project plan provides a well design Gantt chart that has allocated testing activities for each sprint. The schedule has the specific dates for all Unit Testing, Static Code Analysis, Integration testing, and so forth. The below table will list the test activities, the dates, and the people/team in charge.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Task | Due Date/Duration | Responsibility |
| 1 | Static Code Analysis | June 27, 2020 – 1 day |  |
| 1 | Unit Testing | June 28, 2020 – 1 day |  |
| 2 | Review Technical Design Document & Test Plan | June 30, 2020 – 1 day |  |
| 2 | Sprint Review with Client | July 2, 2020 – 1 day | Project Owner |
| 2 | Static Code Analysis |  |  |
| 2 | Unit Testing |  |  |
| 3 | Sprint Review with Client |  | Project Owner |
| 3 | Static Code Analysis |  |  |
| 3 | Unit Testing |  |  |
| 4 | Sprint Review with Client |  | Project Owner |
| 4 | Static Code Analysis |  |  |
| 4 | Unit Testing |  |  |
| 5 | Sprint Review with Client |  | Project Owner |
| 5 | Static Code Analysis |  |  |
| 5 | Unit Testing |  |  |
| Final Testing | Integration Testing | July 21, 2020 – 1 day |  |
| Final Testing | Vulnerability Scanning | July 22, 2020 – 1 day |  |
| Final Testing | Penetration Testing | July 23, 2020 – 1 day |  |
| Final Testing | Review Test Report | July 26, 2020 – 1 day |  |
| Final Testing | Test and Documentation Review with Client | July 27, 2020 – 1 day |  |
| Final Testing | Review Final Test Report | July 28, 2020 – 1 day |  |
|  |  |  |  |
|  |  |  |  |

1. Planning Risks and Contingencies

List the assumptions that have been made during the preparation of this plan.

List the dependencies.

1. Approvals

|  |  |
| --- | --- |
| Consultant | Mir Assadullah |
| Client/Project Sponsor | Christopher L. Truitt |
| Project Manager | Sefanit Urgessa, Mohammed Allibalogun |
| Product Owner | Rob Garcia |
| VLOL Test Team (Front & Back end Dev) | M. Allibalogun, R. Garcia, C. Baisie, J. Bourne |
| DevOps Team | Lawrence Green, Anietie Williams |