Master Test Plan

Virtual Letter of Life (VLOL) Application

Version 2.0

Sefanit Urgessa

Mohammed Allibalogun

Robert Garcia

Meron Getachew Debela

Charles Baisie

Augustin Mwamba

Jamal Bourne

SWEN 670 Software Engineering Project

University of Maryland Global Campus

Date: 07/21/2020

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 06/30/2020 | 1.0 | Initial Master Test Plan |
| 07/21/2020 | 2.0 | Version includes feedback from Dr. Mir Assadullah |
|  |  |  |
|  |  |  |

**Table of Contents**

1. [References 34](#_Toc517588950)
2. [Introduction 34](#_Toc517588950)
3. [Test Items 34](#_Toc517588950)
4. [Software Risk Issues 36](#_Toc517588950)
5. [Features To Be Tested 36](#_Toc517588950)
   1. [Roles 36](#_Toc517588950)
   2. User Stories (Functional Requirements) 36
   3. Non-Functional Requirements 39
6. [Features Not To Be Tested 40](#_Toc517588950)
7. [Approach 40](#_Toc517588950)
   1. Testing Levels 40
   2. Test Tools 41
   3. Meetings 41
   4. Source of Test Data 41
   5. Inputs and Outputs 41
   6. Testing Priorities 41
8. Item [Pass/Fail Criteria 42](#_Toc517588950)
   1. [Evaluation Team 42](#_Toc517588950)
   2. [Exit Criteria 42](#_Toc517588950)
   3. [Evaluation Process 42](#_Toc517588950)
9. [Suspension Criteria and Resumption Requirements 43](#_Toc517588950)
10. [Test Deliverables 43](#_Toc517588950)
11. [Environmental Needs 44](#_Toc517588950)
12. Staffing [and Training Needs 44](#_Toc517588950)
13. [Roles & Responsibilities 44](#_Toc517588950)
14. [Schedule 45](#_Toc517588950)
15. [Planning Risk and Contingencies 46](#_Toc517588950)
16. [Approvals](#_Toc517588950) 47
17. [Glossary – Acronym and Abbreviation](#_Toc517588950) 47

**Table of Tables**

Table 1 – Roles & Responsibilities 44

Table 2 – Schedule 45

Table 3 – Risk Matrix 46

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).

1. **Introduction**

This document will provide an overview test plan for the interoperable Virtual Letter of Life system (VLOL) that will exchange information through the interdisciplinary Emergency Medical Services (EMS). The VLOL application allows 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 Stakeholders
* 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 Testing
* Integration Testing
* System Testing
* Acceptance Testing

1. **Test Items**

The list below identifies all the items that will be used as a target for testing:

* Unauthenticated and Unauthorized User
  + Access to an informative Landing page to learn about the Letter of Life program.
  + Access to a Contact page for information solicitation about the Letter of Life program or contact the System Administrator about an issue.
  + Access to a secure Registration page to enroll in the Letter of Life program.
* Program Participant
  + Access to secure Letter of Life Login Page.
  + Create or delete the VLOL account.
  + View/Add/Edit/Delete Letter of Life information.
  + Print a barcode or quick response (QR) code with a link to Letter of Life (which may be affixed to a bracelet, card, or necklace), so medical personnel can quickly access patient’s medical history in case of an emergency.
  + Access to a secure Contact page to contact Patient Advocate (if designated) and/or the System Administrator about an issue.
* Medical Services Provider
  + Access the Letter of Life system via a Login Page, so unauthenticated and unauthorized users cannot access a patient's medical history.
  + Edit profile information via an online form to ensure profile information is up to date.
  + Search & View for patients on a laptop or mobile device via an online form to retrieve their medical history and provide correct medical care.
  + Search & View for a patient by scanning a barcode or quick response (QR) code, using a mobile device or authorized accessory.
  + Print a MIEMSS-approved Short Form, populated with patient's information, to provide non-system medical facilities and staff with patient's medical history.
  + Access to a secure Contact page to contact the System Administrator about an issue.
* Patient Agent
  + Access the Letter of Life system via a Login Page.
  + Edit my profile information via an online form to ensure profile information is up to date.
  + Search for patients on a laptop or mobile device via an online form, to view, edit, and print patient information, as necessary.
  + Search for a patient by scanning a barcode or quick response (QR) code, using a mobile device or authorized accessory, to view, edit, and print patient information, as necessary.
  + Access to a secure Contact page to contact the System Administrator about an issue.
* System Administrator
  + Access the VLOL system via a Login Page.
  + Search for user accounts on a laptop or mobile device via an online form to add, view, edit, and delete user accounts, as necessary.
  + View/Add/Edit/Delete accounts for Medical Services Providers and Patient Advocates to limit non-patient access to patient information.
  + View a system error log to inform the development team of problems or issues with the application.
  + View a system activity log, so I can audit access and use of the system.
  + Access to a user guide to provide users with instructions on how to use the system.

1. **Software Risk Issues**

Majority of the software risk that could impact the project were listed in Project Plan Risk Management section. Majority of the risks were categorized as low-level risk. The list below provides risks that will impede testing:

* System incompatibility when deployed on our first client server
* Inability for the users to access their personal account and perform update/deletion/creation
* The application unable to print or scan the Quick Response (QR) code
* The system fails to generate a QR code 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. There will be a patch management plan to address issues that are not discovered in testing.

* 1. **Roles:**
* Unauthenticated and unauthorized user
  + Any user who accesses the page and has not enrolled or created an account. Guest users.
* Program Participant
  + A user who has enrolled in the VLOL and has successfully logged in his/her account.
* Medical Services Provider
  + A user who enrolls and plays the provider role for multiple Program participants. They ensure a participant medical history is up to date.
* Patient Advocate
  + A user enrolled has an advocate by the program participant. They will be assisting the user by making sure their account is up to date.
* System Administrator
  + A user who ensures the correct credentials are assigned to users based on their roles. This user has universal access to everything expect creating an account for a program participant.
  1. **User Stories (Functional Requirements):**

Table 1 below shows User Stories (functional requirement) and acceptance criteria for each requirement, which is also available in the Software Requirement Specification document. Additionally the table displays associated priority for each requirement. The test priority indicators that will be used are:

* H - must be tested
* M - should be tested, will test only after all H items are tested
* L - might be tested, but not until all H and M items have been tested

Table 1 User Stories and Acceptance Criteria

| **ID** | **User Stories** | **Associated Priority** | **Acceptance Criteria** |
| --- | --- | --- | --- |
| REQ-1.1 | As an unauthenticated and unauthorized user, I want access to an informative Landing page, so I can learn about the Letter of Life (LOL) program. | H | * View a homepage that introduces the LOL program. |
| REQ-1.2 | As an unauthenticated and unauthorized user, I want access to a Contact page, so I can solicit information about the LOL program or contact the System Administrator about an issue. | H | * View a Contact page for information solicitation about the LOL. * Contact the System Administrator about an issue. |
| REQ-1.3 | As an unauthenticated and unauthorized user, I want access to a secure Registration page, so I can enroll in the Letter of Life program. | H | * Enroll in the LOL program using a secure Registration page. |
| REQ-1.4 | As a registered, authenticated, and authorized Program Participant, I want to access my LOL via a Login Page, so unauthenticated and unauthorized users cannot access my medical history. | H | * Login to LOL account via the Login Page by providing username and password. |
| REQ-1.5 | As a registered, authenticated, and authorized Program Participant, I want to add my LOL information to the database via an online form, so medical personnel can have access to my medical history in case of an emergency. | H | * Add medical history into LOL database. |
| REQ-1.6 | As a registered, authenticated, and authorized Program Participant, I want to view my LOL information via an online form, so I can validate my information or print my LOL. | H | * View and validate medical history. |
| REQ-1.7 | As a registered, authenticated, and authorized Program Participant, I want to edit my LOL information via an online form, so I can ensure my medical history is up to date. | H | * Edit medical history. |
| REQ-1.8 | As a registered, authenticated, and authorized Program Participant, I want to delete my LOL information from the database via an online form, so I may stop participating in the program. | H | * Delete medical history. |
| REQ-1.9 | As a registered, authenticated, and authorized Program Participant, I want to print a barcode or quick response (QR) code with a link to my LOL (which I may affix to a bracelet, card, or necklace), so medical personnel can quickly access my medical history in case of an emergency. | H | * Print QR code or barcode with a link to LOL account. |
| REQ-1.10 | As a registered, authenticated, and authorized Program Participant, I want access to a secure Contact page, so I can contact my Participant Advocate (if designated) and/or the System Administrator about an issue. | H | * Contact Participant Advocate or System Administrator about LOL. |
| REQ-1.11 | As a registered, authenticated, and authorized Medical Services Provider, I want to access the LOL system via a Login Page, so unauthenticated and unauthorized users cannot access a Participant's medical history. | H | * Login to VOL system via Login Page. |
| REQ-1.12 | As a registered, authenticated, and authorized Medical Services Provider, I want to edit my profile information via an online form, so I can ensure my information is up to date. | H | * Edit provider profile information. |
| REQ-1.13 | As a registered, authenticated, and authorized Medical Services Provider, I want to search for Participants on a laptop or mobile device via an online form, so I can retrieve their medical history and provide correct medical care. | H | * Search for Participants medical history. |
| REQ-1.14 | As a registered, authenticated, and authorized Medical Services Provider, I want to search for a Participant by scanning a barcode or quick response (QR) code, using a mobile device or authorized accessory, so I can retrieve their medical history and provide correct medical care. | H | * Scan QR code and retrieve Participant medical history. |
| REQ-1.15 | As a registered, authenticated, and authorized Medical Services Provider, I want to view a Participant's medical history on a laptop or mobile device via an online form, so I can provide correct medical care. | H | * View Participant’s medical history. |
| REQ-1.16 | As a registered, authenticated, and authorized Medical Services Provider, I want to print a MIEMSS-approved Short Form, populated with my Participant's information, so I can provide non-system medical facilities and staff with my Participant's medical history. | H | * Print a MIEMSS-approved Short Form with populated Participant’s information. |
| REQ-1.17 | As a registered, authenticated, and authorized Medical Services Provider, I want access to a secure Contact page, so I can contact the System Administrator about an issue. | H | * Contact System Administrator about an issue. |
| REQ-1.18 | As a registered, authenticated, and authorized Participant Advocate, I want to access the LOL system via a Login Page, so unauthenticated and unauthorized users cannot access the medical history of my Participants. | H | * Login to LOL via Login Page. |
| REQ-1.19 | As a registered, authenticated, and authorized Participant Advocate, I want to edit my profile information via an online form, so I can ensure my information is up to date. | H | * Edit profile information. |
| REQ-1.20 | As a registered, authenticated, and authorized Participant Advocate, I want to search for Participants on a laptop or mobile device via an online form, so I can view, edit, and print Participant information, as necessary. | H | * Search for Participant information. |
| REQ-1.21 | As a registered, authenticated, and authorized Participant Advocate, I want to search for a Participant by scanning a barcode or quick response (QR) code, using a mobile device or authorized accessory, so I can view, edit, and print Participant information, as necessary. | H | * Scan QR code or barcode to search for Participant’s information. |
| REQ-1.22 | As a registered, authenticated, and authorized Participant Advocate, I want to view the LOL information of my Participants via an online form, so I can validate their information or print their LOL. | H | * View LOL information of Participants. |
| REQ-1.23 | As a registered, authenticated, and authorized Participant Advocate, I want to edit the LOL information of my Participants via an online form, so I can ensure their medical history is up to date. | H | * Edit LOL information of Participants. |
| REQ-1.24 | As a registered, authenticated, and authorized Participant Advocate, I want to print a barcode or quick response (QR) code with a link to my Participant's LOL (which I may affix to a bracelet, card, or necklace), so medical personnel can quickly access their medical history in case of an emergency. | H | * Print a QR code or barcode with a link to Participant’s LOL. |
| REQ-1.25 | As a registered, authenticated, and authorized Participant Advocate, I want access to a secure Contact page, so I can contact the System Administrator about an issue. | H | * View Contact page * Contact System Administrator about issue. |
| REQ-1.26 | As a registered, authenticated, and authorized System Administrator, I want to access the LOL system via a Login Page, so unauthenticated and unauthorized users cannot access Participant medical history or user account information. | H | * Login to LOL using username and password via the Login Page. |
| REQ-1.27 | As a registered, authenticated, and authorized System Administrator, I want to search for user accounts on a laptop or mobile device via an online form, so I can add, view, edit, and delete user accounts, as necessary. | H | * Search for user accounts. |
| REQ-1.28 | As a registered, authenticated, and authorized System Administrator, I want to add accounts for Medical Services Providers and Participant Advocates, so I can limit non-Participant access to Participant information. | H | * Add accounts for EMS Providers and Participant Advocates. |
| REQ-1.29 | As a registered, authenticated, and authorized System Administrator, I want to view accounts for Medical Services Providers and Participant Advocates, so I can validate their information. | H | * View EMS Providers and Participant Advocates accounts. |
| REQ-1.30 | As a registered, authenticated, and authorized System Administrator, I want to edit accounts for Medical Services Providers and Participant Advocates, so I can ensure their information is up to date. | H | * Edit EMS Providers and Participant Advocates accounts. |
| REQ-1.31 | As a registered, authenticated, and authorized System Administrator, I want to delete accounts for Medical Services Providers and Participant Advocates, so I can limit non-Participant access to Participant information. | H | * Delete EMS Providers and Participant Advocates accounts. |
| REQ-1.32 | As a registered, authenticated, and authorized System Administrator, I want to search for Participants on a laptop or mobile device via an online form, so I can add, view, edit, and delete Participant information, as necessary. | H | * Search for Participants information. |
| REQ-1.33 | As a registered, authenticated, and authorized System Administrator, I want to add accounts for Participants, so I can enroll them in the program. | H | * Register Program Participants. |
| REQ-1.34 | As a registered, authenticated, and authorized System Administrator, I want to view accounts for Participants, so I can validate their information. | H | * View Participants account information. |
| REQ-1.35 | As a registered, authenticated, and authorized System Administrator, I want to edit accounts for Participants, so I can ensure their information is up to date. | H | * Edit Participants account information. |
| REQ-1.36 | As a registered, authenticated, and authorized System Administrator, I want to delete accounts for Participants, so I can disenroll them from the program. | H | * Delete Participants account information. |
| REQ-1.37 | As a registered, authenticated, and authorized System Administrator, I want to view a system error log, so I can inform the development team of problems or issues with the application. | H | * View system error log. |
| REQ-1.38 | As a registered, authenticated, and authorized System Administrator, I want to view a system activity log, so I can audit access and use of the system. | H | * View system activity log. |
| REQ-1.39 | As a registered, authenticated, and authorized System Administrator, I want to have access to the user guide, so I provide users with instructions on how to use the system. | H | * Access the user guide. |

* 1. **Non-Functional Requirements:**

NF-1.1: As a system, I want to limit Create-Read-Update-Delete (CRUD) access to authenticated and authorized users only, so I may preserve the confidentiality and integrity of Personally Identifiable Information (PII).

NF-1.2: As a system, I want new registrants to verify their accounts via a link sent via email (2FA), so I may mitigate the creation of fake accounts.

NF-1.3: As a system, I want all non-verified accounts to be deleted within 24 hours, so I may prevent database bloating.

NF-1.4: As a system, I want all user input validated, so I may prevent injection and scripting attacks.

NF-1.5: As a system, I want user sessions to timeout within 30 minutes, so I may prevent inadvertent disclosure of PII.

NF-1.6: As a system, I want to limit multi-account access to authorized devices only (MAC Whitelisting), so I may preserve the confidentiality and integrity of PII, as well as the availability of the system.

NF-1.7: As a system, I want the following NIST SP 800-53 controls enforced throughout the application, so I may preserve the confidentiality and integrity of PII, as well as the availability of the system:

* + AC-2: ACCOUNT MANAGEMENT
  + AC-3: ACCESS ENFORCEMENT
  + AC-7: UNSUCCESSFUL LOGON ATTEMPTS
  + AC-8: SYSTEM USE NOTIFICATION
  + AC-11: SESSION LOCK
  + AC-12: SESSION TERMINATION
  + AU-2: AUDIT EVENTS
  + AU-8: TIME STAMPS
  + IA-2: IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)
  + SC-13: CRYPTOGRAPHIC PROTECTION
  + SC-23: SESSION AUTHENTICITY
  + SI-10: INFORMATION INPUT VALIDATION
  + SI-11: ERROR HANDLING

NF-1.8: As a system, I want all SQL queries to use prepared statements, so I can prevent SQL injection attacks.

NF-1.9: As a system, I want all tables to be normalized to 3rd Normal Form, so I can improve data integrity and reduce data redundancy.

NF-1.10: As a system, I want to unit test all functions and methods, so I can ensure boundary conditions are not violated.

NF-1.11: As a system, I want to incorporate static code analysis, using both automated and manual methods, so I can ensure there are no weaknesses due to code syntax.

NF-1.12: As a system, I want to incorporate dynamic code analysis and penetration testing, so I can ensure there are no vulnerabilities due to code semantics.

NF-1.13: As a system, I want to style and comment all code per the appropriate style guide, so I can properly maintain and turn-over the application, as necessary.

NF-1.13: As a system, I want to incorporate version control, so I may audit and track code generation.

NF-1.14: As a system, I want all CRITICAL, HIGH, MEDIUM, and STANDARD issues and risks, identified during static and dynamic analysis, corrected before uploading to a repository, so I can ensure the repository's code meets the bug bar.

NF-1.15: As a system, if a CRITICAL, HIGH, MEDIUM, and STANDARD issue and risk cannot be corrected immediately, I want it recorded in a bug log, so I can ensure the issue or risk is corrected before deployment.

NF-1.16: As a system, I want all presentation code to be HTML5 and W3C compliant, so I can ensure the application provides the same functionality across different platforms.

1. **Features Not To Be Tested**

The following domain are outside of the scope of this project and will not be tested by the EMSPlus team:

1. Network security
2. System deployment
3. **Approach**
   1. **Testing Levels**

As stated in the introduction section, testing for the VLOL application will involve Unit, System, Integration, and Acceptance 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 be an overseer. The time restriction for the completion of this application may require for additional help, and that is where project manager will assist as test managers to ensure full test coverages are completed. The Bottom-Up technique will be the selected approach that developers will use to conduct Unit testing. Developers will start by testing small units of code and work their way to larger codes.

UNIT TESTING – developers will utilize the JUnit framework to conduct the back-end testing. Tester will utilize a White-box Test documentation to create test cases and keep track of all the testing that are being executed. Testers may also utilize this form as a point of reference for testing by documenting sample outputs, data printout, and bug found outside the list of test cases. Project managers will utilize Jira to keep track of bugs and assigned them to the developers for resolution.

INTEGRATION TESTING – the objective of integration testing is to ensure that objects in both the Model and Control are working together. This form of testing will eliminate any potential memory leaks.

SYSTEM TESTING - the system testing 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 with the assistance of all the developers. 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. Developers will use Selenium against the web application.

* 1. **Test Tools**
* **Selenium 3.141.59** – an automated testing tool that will be used by developers to test against the VLOL web application. This software will run a multiple set of unit and acceptance tests, ensure coded modules don’t break when their working together, as well as performing regression testing.
* **JUnit 4.13** – a unit testing framework for all applications writing in Java language. This framework will be used by developers to conduct 100% test coverages for all classes and methods used for the VLOL application by providing annotation for test methods, conduct assertion for expected results, and provide test runners.
* **Jira 8.10.0** – this project management software tool will be used by Project Managers and developers to post, update, track, and fix bugs and issues in the VLOL application through a Kanban board.
* **OWASP ZAP 2.9.0** – the web application testing security tool that will be used to test the VLOL application. Developers will use this tool to perform fuzzing, scripting, spidering, and proxying against the VLOL web application.
* **SonarQube 8.3.1** – a software tool used by developers during Code Reviews for code quality and security.

* 1. **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 as well as the communication metrics on project plan section 10.3. The Product Owner may request additional meetings for testing for reasonable reasons, as long as the PMs are notified.

* 1. **Source of Test Data**

The source data that will be used for our test cases will come from our H2 development embedded database.

* 1. **Testing Priorities**

Testers will categories test cases within four parameters

* **Blockers** – any and all test cases that will stop the VLOL application from running. Any test cases under this category will be addressed immediately by the developers,
* **Critical** – any test cases under this category will also be address ASAP in order for the application to properly function.
* **Major** – test cases that follow under this section are not a high priority. PM will assign theses type of cases to developers as they see fit.
* **Minor** – the lowest level of test cases which do not impede the lifeline of the application, nor will they impact the consumer day to day activity while they are using the application. Test cases under this category should be address at the very end, and if there’s a deadline, developers should leave these cases and focuses on those tests from the categories above.

1. **Item Pass/Fail Criteria**

The criteria used to determine if each test item has passed or failed testing is categorized by three sections.

* 1. **Evaluation Team**

In general, these criteria are less technical and more a combine subjective decision formulated by the Project Managers, Product Owner, Lead Tester, and the Client. There may come a time when the application maybe released with flawed software that may still benefit the organization for the current time being.

* 1. **Exit Criteria**

The exit criteria will reflect the terms and condition set by the Approach section in the Test Plan documentation. These criteria will be set by the test team, which will consider the following aspect of the application:

* Are the main functionalities of the application available?
* Are all critical faults cleared from the application?
* Have both the test summary report and test plan been updated with the current founding?

The test team will ensure the following list have been met:

* Cleared all incident that will impact the application lifecycle
* Assess, mitigate, and record all remaining incidents
* Assign personal to update the necessary documentation before moving to the next test phase
* Ensure the Client has confirmed the Acceptance testing and create a sequential test report with what has occurred
  1. **Evaluation Process**

The evaluation process will take into consideration four separate data and make a final decision based on those evaluation

* **Test Result Summary** – evaluate all incident reports and compared with what requirement they have impacted
* **Business Scenario** – this evaluation compares those incident reports and see how it impacted the application at a business or operational level
* **Estimate Business Impact** – this evaluation point will outline the effect of the incident, the frequency impact on the application, and measures that can be taken to mitigate with those defects on the application when it’s released
* **Make Acceptance Decision** – this final decision will be made after taking into consideration the outcome of the previous 3 data results

The ultimate goal is to have between 90-100% test coverages for all functionalities, which entails that all JUnit test cases created pass testing. The SonarQube prevents the duplications and remove potential bugs that may not be detected when creating Junit test cases. As for Selenium, it will remove 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:

* All actors are unable to reach the website via browser or mobile application
* 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 QR code via the app, or the system administrator unable to grant users action on their accounts
* Failure to access database information

Once the development team has resolved those events, testing will resume. However, the protocol in this case will require that all functionality must be retested and the Product Owner will be responsible to schedule a meeting with the client to ensure that the application is properly working.

1. **Test Deliverables**

The acceptance criteria for these deliverables will be directly linked to the Product Deliverables in the Project Plan under section 5.4. The test deliverables go as follow:

* **Test Plan** (this document)
* **User Acceptance Test** (UAT) plan
  + Black Box Testing
    - Reduced time spent searching through accurate patient information
    - Enhance security to ensure the security of patients’ sensitive data storage and accessibility: allowing information to be viewable to the users with regard to their level of access based upon their level of position.
    - Accessibility to patients to easily update their medical information to ensure accurate information sharing between first responders and other authorized personnel
    - Accessibility to patients to easily update their medical information to ensure accurate information sharing between first responders and other authorized personnel
* **Unit Test plan**
  + White Box Testing
    - Personal Health Data will be date and time stamped, inclusive to medication management and continuity of care.
    - The system will monitor who is accessing patient information and provide data for audit.
* **Test Reports**
  + Level Test Log report
  + Test Incident report
  + Test summary report

1. **Environmental Needs**

Hardware: Windows or Mac PC

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

1. **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 all developers split the unit test (Junit) by having one of the PM conducting testing and the Product owner overseeing each test cases. The contextual inquiry will be conduct by the Product Owner to analyze the interaction between the application and all external members who will utilize the VLOL system (Client, Consultant, and EMTs). Lastly, all developers will work alongside the DevOps team to ensure that system integration functionality are being tested through a pipeline.

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 be aware of the minimal requirements for the application to pass a contextual inquiry testing

1. **Roles & Responsibilities**

Table 2 – 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 member’s role and responsibility in the testing phase. The Lead development member will be in charge of ensuring that all unit test cases from the test plan 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 an input over any new change request or updates, as stated in the Project Plan section 8 Change Management Plan. The DevOps team will provide developers a 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. 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.

Table 3 – Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Task | Due Date/Duration | Responsibility |
| 1 | Static Code Analysis | June 22, 2020 – 1 day | Developers |
| 1 | Unit Testing | June 22, 2020 – 1 day | Developers |
| 2 | Review Technical Design Document & Test Plan | June 23, 2020 – 1 day | PM, Project Owner, Developers |
| 2 | Sprint Review with Client/Acceptance Testing | June 27, 2020 – 1 day | Project Owner |
| 2 | Static Code Analysis | June 29, 2020 – 1 day | Developers |
| 2 | Unit Testing | June 29, 2020 – 1 day | Developers |
| 3 | Sprint Review with Client/Acceptance Testing | July 3, 2020 – 1 day | Project Owner |
| 3 | Static Code Analysis | July 6, 2020 – 1 day | Developers |
| 3 | Unit Testing | July 6, 2020 – 1 day | Developers |
| 4 | Sprint Review with Client/Acceptance Testing | July 11, 2020 – 1 day | Project Owner |
| 4 | Static Code Analysis | July 13, 2020 – 1 day | Developers |
| 4 | Unit Testing | July 13, 2020 – 1 day | Developers |
| 5 | Sprint Review with Client/Acceptance Testing | July 18, 2020 – 1 day | Project Owner |
| 5 | Static Code Analysis | July 20, 2020 – 1 day | Developers |
| 5 | Unit Testing | July 20, 2020 – 1 day | Developers |
| Final Testing | Integration Testing | July 21, 2020 – 1 day | Project Owner, Developers |
| Final Testing | Vulnerability Scanning | July 22, 2020 – 1 day | Project Owner, Developers |
| Final Testing | Penetration Testing | July 23, 2020 – 1 day | Project Owner, Developers |
| Final Testing | Review Test Report | July 26, 2020 – 1 day | PM, Project Owner, Developers |
| Final Testing | Test and Documentation Review with Client | July 27, 2020 – 1 day | PM, Project Owner, Developers |
| Final Testing | Review Final Test Report | July 28, 2020 – 1 day | PM, Project Owner, Developers |

1. **Planning Risks and Contingencies**

Table 4 below displays potential risks, risk level, and contingencies.

Table 4 – Risk Matrix

|  |  |  |
| --- | --- | --- |
| Potential risk | Risk Level | Contingencies |
| Lack of personnel resources when testing is to begin. | Low | Acquire additional resources from clients and external project teams |
| Lack of availability of required data | Low | Sample data will be utilized for testing |
| Changes to the original requirements or designs. | Low | Enforce all additional request to be document separately and approved by the Change Control Board (CCB) |
| Inability for the users to access their personal account and perform update/deletion/creation | Medium | Project manager will select will create a Jira ticket and assign it to one of the developers who will work to fix the issue |
| The application unable to print or scan the Quick Response (QR) code | High | Developers will terminate all activities and redirect their focus to resolve the issue at hand |
| The system fails to generate a QR code when the user creates an account | High | Developers will terminate all activities and redirect their focus to resolve the issue at hand |

1. **Approvals**

* Consultant → M. Assadullah
* Client/Project Sponsor → C. L. Truitt
* Project Manager → S. Urgessa, M. Allibalogun
* Product Owner → Rob Garcia
* VLOL Test Team (Front & Back end Dev) → M. Allibalogun, R. Garcia, C. Baisie, J. Bourne
* DevOps Team → L. Green, A. Williams

1. **Glossary**

**Acronym and Abbreviation**

* CCB – Change Control Board
* EMS – Emergency Medical System
* EMT – Emergency Medical Team
* IDE – Integration Development Environment
* PC – Personal Computer
* PM – Project Manager
* QR – Quick Response
* SDLC – Software Development Life Cycle
* VLOL – Virtual letter of Life