

Project Test Plan

Guidance is indicated using red text. Examples are indicated using blue text.

Project Name	Plan Name	Last Approved Date	Last Approved By
VA T4NG MCCF	Project Test Plan	10/19/2018	Michael King

Page Version History

Version	Date	Comment
Current Version (v. 8)	Oct 19, 2018 17:06	Michael King: Approved
v. 7	Sep 28, 2018 14:35	Janie Qu
v. 6	Sep 28, 2018 14:27	Janie Qu
v. 5	Sep 19, 2018 13:03	AJ DeLisle (Deactivated)
v. 4	Aug 28, 2018 23:30	Michael King
v. 3	Aug 14, 2018 10:46	AJ DeLisle (Deactivated)
v. 2	Jul 20, 2018 10:02	Colin Winchester
v. 1	Jul 20, 2018 09:26	Copy Page Tree

Table of Contents:

- [Test Overview](#)
- [Test Dependencies](#)
- [Test Approaches](#)

Test Overview

Component	Details/Approach
Test Objectives	<ul style="list-style-type: none">• To provide test coverage for 100% of the documented Sprint requirements and derived requirements• To provide coverage for System/ Software Design Document elements• To execute 100% of the test cases during User Functionality Testing• To execute 100% of the Performance testing• To create, maintain and control the test environment• To create and manage test cases• To create and manage defects• To create automation of test cases to increase code coverage• To execute automation on a regular basis by Sprint and shorter cycles
Test Scope (Inclusions and Exclusions)	<p>Test Inclusions:</p> <p>Testing for the project will include all necessary test areas to ensure a quality product is released to users on Production. The entire system from unit testing, to end-to-end testing, through user acceptance testing. Sprint stories will be pulled from the backlog and will be tested for meeting the acceptance criteria defined in Sprint planning. Priority of testing is set by the customer and user agents so that the highest priority items are tested first. UX testing will be compared against approved wireframes as soon as they are available.</p> <p>Test Exclusions</p> <p>N/A</p>

Entry Criteria	<ul style="list-style-type: none"> • All stories should have acceptance criteria written for them with tester, development, and business analyst approvals • All unit tests should have passed • All integration tests should have passed • All test cases have been written for the new features to be tested • The build is deployed to test (or pre) and passes a smoke test run by either automation or testers
Exit Criteria	<ul style="list-style-type: none"> • All Sprint test cases have been completed • No priority one or two bugs were found • No severity one bugs were found (as approved by BAs) • Less than five priority three bugs were found • 100% of all functional tests were run
Test Reports	<ul style="list-style-type: none"> • Testing results can be found in Rally. • Reports will be run on each Sprint showing the test cases passed, test cases failed, and status of the stories tested. Test cases will be pass or fail based on the issues found. • Stories passing will be determined by bugs found and priority of such bugs. A story will fail if a priority one or two bug is discovered in the story. A story will also fail if five or more priority three or below bugs are found.
Link to Test Case Repository	<p>TASCore- https://rally1.rallydev.com/#/120708787896d/testcases?tpsSI=0&tpsV=qv%3A0</p> <p>eBilling- https://rally1.rallydev.com/#/79590444732d/testcases?tpsSI=0&tpsV=qv%3A0</p> <p>ePayments- https://rally1.rallydev.com/#/79590436908d/testcases?tpsSI=0&tpsV=qv%3A0</p> <p>ePharmacy- https://rally1.rallydev.com/#/79590445676d/testcases?tpsSI=0&tpsV=qv%3A0</p> <p>eInsurance- https://rally1.rallydev.com/#/79590442620d/testcases?tpsSI=0&tpsV=qv%3A0</p>
Test Case Management Approach	<p>Once elaboration is complete, test cases will be required to record the testing needed to be done for each Sprint. A group of sprints will make up the sum of a release. A release version is the key Rally data attribute that establishes requirements traceability between user stories, test cases, test executions, and defects. For that reason, much of the test reporting is at the release version level.</p> <p>Test cases will be stored in Rally.</p> <p>Test cases are derived from stories, derived requirements, and escaped bugs.</p>
Test Schedule	The test schedule is maintained in Rally. Testing is an on-going effort, but regression testing is always scheduled for the last sprint of a build.

Test Dependencies

Component	Details/Approach
General Dependencies	<p>Test data must be in place. It should be the same data that will be used on the test, pre, and production systems. In early sprints, the data will be 'lorum ipsum' fake text, but this need to be replaced by real text as soon as possible. Not having real data as prevented release to production on previous projects and must be taken very seriously.</p> <p>Automation will require access to the test server from which it is run. Also, the automation will need a connection to the application under test and the Jenkins server.</p>
Test Environment Requirements	<p>The test team uses the Dev/Test stack that is present in MAG. The latest unit tested code base must be first deployed to MAG DEV and cleared by QA. The code is then promoted up the stack (MAG CI, MAG CIT, MAG SQA, MAG UAT).</p> <p>TASCore conducts testing in MAG CI and MAG CIT. Eventually all future work to transition to the MAG.</p> <p>Currently, MUMPS teams have their own testing instances in AITC.</p>

Test Approaches

Component	Test Technologies	Details/Approach
Obtaining and Maintaining Test Data	N/A (Manual)	<p>Initial test data for the first few sprints will be 'lorem ipsum' fake text content and template images. This will be expedite development and test work, where functionality will be higher priority. As the process moves closer to data on test will be populated from Pre. At this point, there is no automation to do this, so the data will need to be populated.</p> <p>It is imperative that the data is created as early as possible to avoid issues with data migration. Testers will need time to review content and determine that the data is correct and properly moved.</p>

Integration Testing	N/A (Manual)	Integration testing will occur in the functional test area for now. This area of testing can be created in automation should be created at a higher acceptance area, as it is hard to maintain. Automation will cover a subset of the function but should include the main use cases.
Functional Regression Testing	N/A (Manual)	<p>Each story will be assessed for acceptance criteria (AC) and test cases will be created based on this. This testing will be completed during the Sprint. Stories will be considered done when all sprint tests are complete, no priority ones are found, and no severity one bugs are found.</p> <p>Testing against existing features created prior to the Sprint will be run concurrently with sprints. A full regression running all tests on existing functionality, will be run before release of the project.</p>
Static Code Analysis Testing	Fortify	The developers and DevOps will run code analysis software on the code to ensure code quality. Developers will be pairing and code reviews to ensure the quality of their code.
Security Testing	Fortify	IA testers will run a series of penetration and stig compliance testing to ensure the product does not have security issues that allow improper access. IA will run IA automation tools as well.
Performance Testing	JMeter	Tests on performance will be done toward the end of the release testing to ensure it is stable before testing and to handle the expected load on the system.
Accessibility Testing	JAWS	Each story will include WCAG compliance testing, where applicable, and will be completed before completion of the sprint. WCAG compliance testing for UI built prior to the Sprint will be considered part of regression testing. WCAG compliance testing will be done using AInspector for each area. By contractual obligation, this tool will be used in Production, so we will ensure all tests it runs work.
User Acceptance Testing (UAT)	N/A (Manual)	UAT will be conducted by the customer after each build. UAT will be scheduled with the customer and will be done with all major stakeholders.
Full Regression Testing	N/A (Manual)	<p>Each story based off a visual comparison (comp) will get reviewed by the visual QA team to ensure the views are matching the comparisons, before the end of each Sprint. Visual bugs will follow the rules of the bug manager of this document, given a priority and a severity. Both visual QA bugs and stories with a visual QA component will be tested by the second Friday of each Sprint, as the testing is done per page, with consideration of the stories it will allow the developers to work on any issues found before the end of the sprint</p> <p>Visual QA will provide a report of issues found that will be recorded in Rally for development to fix. Testers will provide information based on the report, but any questions should be directed to Visual QA as testers will just be entering information as is.</p> <p>As comps are malleable until approval, any issues found are not considered stoppers until the comps are approved by the customer. Development will be working to the latest comp stored in Invision, as we will be starting development once comps are finalized. It is understood that changes in the comps may require new stories to be generated to update them. These stories will be put through evaluation and prioritization before being placed in the backlog.</p>
Web Validation Testing	validator.w3.org	<p>All production sites pass Web Validation (https://validator.w3.org/)</p> <p>For sites that generate HTML or are not accessible by the tool, we upload the code to the site for validation.</p> <p>This will check the HTML and CSS against the standard.</p> <p>JavaScript errors are caught in the browser console and logged as defects.</p>
SEO Best Practices	N/A	The MCCF website is for internal VA use only and will not be indexed by any search engines.
Defect Handling	Rally & RTC	<p>Finding defects is highly important to ensure quality software. When defects are discovered, the sequential events below define the current Test Team defect management process.</p> <ol style="list-style-type: none"> 1. Discover a defect in the Test, Pre, or Test/Pre Liferay environment 2. Create a Rally defect. Please refer later in this section for instruction on how to properly create a defect to meet these traceability and reporting standards 3. Notify the Agile Team in the MCCF HipChat room for defect visibility and clarification and prioritization in sprints 4. Defects that can be fixed during the Build can be recorded as subtasks in the applicable test case 5. The Team can also call for Ad Hoc Bug Triage Meetings if there is the need to review all bug statuses. 6. Testing results for bug fixes are stored on Rally

User Story
to Test Case
Traceability

Rally

Every test case created in Rally is linked to a story or 508 requirement.

Details for TC2674

Defects (3)

Test Sets (0)

Results (0)

Discussion (0)

Revisions (12)

General

ID: TC2674

Name: FHIR Account Search by ID

Tags:

Description:

1. Response is JSON

2. Search Response is a Bundle

3. Search Parameters conform <https://www.hl7.org/fhir/search.html>

1. _id

2. _format

4.

Attachments:

Owner: [Justin Ballerstein](#)

Project: TASCore

Test Folder:

Expedite: False

Rank:

Test Case

Objective:

Work Product: US3811: Implement FHIR Conformance Testing (CM) (T) (D)

Type: Compliance

Priority:

Pre-Conditions:

M