TABLE of CONTENTS

REVISION HISTORY

VERSION	DATE	NAME	NOTE
0.1	2019/02/08	William Hsu	Initial a draft - list down all major items

1 INTRODUCTION

1.1 PURPOSE

The document will give an overview of where to begin the QA work and the end for tests. Therefore, it includes the resource plan, man-day estimation, test scope, testing techniques.

1.2 TERMINOLOGY

Terminology	Description

Table 1. Terminology

1.3 TEST ENVIRONMENT

1. System and applications - XXX
2. Test data - XXX
3. Database server - XXX
4. Client operating system - XXX
5. Browser - XXX
6. Hardware includes server operating system - XXX
7. Network - XXX

1.4 FEATURE INTRODUCETION

In order to have more information to look into features and provide correct information for stakeholders, this needs inputs from Product side to explain the major features and specifications.

1.5 TEST SCOPE

1.5.1 IN SCOPE

Here are overall test scopes. We separate test scope into the three parts

- API test
- functional test
- Performance test

1.5.2 OUT OF SCOPE

- L10N test
- Scan security hole
- Server - Hardware performance measurement

1.5.3 TEST DELIVERABLES

Phase	Deliverable	Owner	Contributors
Analysis	Triage blocking issues		
	Document Study		
	QA Test Plan		
Design	Test Case/Steps		
	Test Strategy		
	Integration Test Cases/Steps		
	Test Case Review/Refinement		
	Test Data Acquisition Plan		
Build	Prepared Integration Test Environments		
	Prepared QA Test Environment		
	Test Build		
Test	Unit Test		
	Smoke test		

	RAT test
	Fullrun
	Regression Test
	Automation Test/Scripts

1.5.4 ASSUMPTIONS

1. QA doesn't prefer to test local build unless it includes a high risk patch.
2. QA only tests the features which are informed. If anyone want to merge unauthorized patches. Please be responsible for all consequences.
3. Any features should follow these release criteria.
    1) No known blocking bugs associated (i.e. no "foot-in-the-door" feature landings, trailing dependent blocking bugs)
    2) Feature must meet all acceptance criteria before being verified fixed by QA
    3) Feature acceptance criteria should include any necessary UX, security and QA signoffs as identified during feature planning (for features that require the extra attention)
    4) All features should include tests (manual and automation) in their acceptance criteria
    5) Complex/high risk features should include a landing window in their acceptance criteria to avoid landing risky features towards the end of the feature development window

1.5.5 LIMITATIONS & RISK

1. Holiday impact
2. Product knowledge is not at the same level
3. Due to the web-based service, it's hard for QA to find the regression window

2 RESOURCE PLAN

2.1 ROLES & RESPONSIBILITIES

ROLES	RESPONSIBILITY	ASSIGNED TO
Director	To help communicate and coordinate high level information	
PM	To help triage bugs and negotiate showstoppers with partners	
EPM	To help triage bugs or trace the progress of project and engineer	
RD Manager	To work on bug triage, resource allocation, and HLA review	
Developer	To work on feature development and bug fixing	
QA Manager	To be in charge of QA resource allocation and quality estimation	
QA Engineer	Testing, automation, and help developers to smoothly release the build	
Taster	To help execute smoke test, RAT test, and find regression window	

2.2 ENVIRONMENT/BUILD

Stage: TBC (E.g, www-stage.XXXXXX.com)
Alpha/INT: TBC (E.g., www-int1.XXXXXX.com)
Beta/CT: TBC (E.g., www-ct.XXXXXX.com)
Pre-Prod: TBC (E.g., www-p2.XXXXXX.com)
Production: (E.g., www.XXXXXX.com)

2.3 TEST TOOLS

1. Automation framework: bok-choy
2. Chrome and Firefox web developer tool
3. Restlet Client (Chrome)
4. Fiddler

3 TEST PLAN

3.1 UNIT TEST

Skip. Developers should be in charge of unit tests.

3.2 SMOKE TEST

3.2.1 SCOPE

The purpose of smoke testing is to determine whether the new software build is stable or not so that the build could be used for detailed testing by the QA team and further work by the development team. So, test scope should cover all built-in functionalities but QA just have basic verification to make sure the release build is qualified. According to the smoke test result, we can decide if we can perform further tests.

3.2.2 SMOKE TEST ENTRANCE CRITERIA

1. All functionalities follow high-level design.
2. The test environment is available and ready for use.
3. Test tools installed in the environment are ready for use.
4. Testable code is available and reviewed.
5. Test Data is available and validated for correctness of Data.
6. Test case and suite are ready to use.

3.2.3 TEST CASES

TBC (E.g., www.github.com/XXXXXX/SmokeTestCases)

3.2.4 TEST DATA

TBC (E.g., www.github.com/XXXXXX/SmokeTestData)

3.2.5 DEFECT MANAGEMENT

A bug query which links to bug management system

3.2.6 TEST RESULTS

A hyperlink which links to test case management system

3.2.7 SMOKE TEST EXIT CRITERIA

1. All test cases should be completed.
2. The level of requirement coverage has been met.
3. There are NO Critical and high severity defects that are left outstanding.
4. All high risk areas are completely tested.
5. All software development activities are completed within the planned timelines

3.3 FUNCTIONAL TEST (Release Acceptance Test)

3.3.1 SCOPE

The release acceptance test is generally a short set of tests (10%~20% of fullrun test), which exercises mainstream functionalities. Any build that finds the critical or blocking issue during release acceptance test will be rejected.

3.3.2 RELEASE ACCEPTANCE TEST ENTRANCE CRITERIA

1. Test will be kicked off after all promised features are landed.
2. All features should follow feature release criteria.
3. The test environment is available and ready for use.
4. Test tools installed in the environment are ready for use.
5. Testable code is available and ready.
6. Test data is available and validated.

3.3.3 TEST CASES

TBC (E.g., www.github.com/XXXXXX/RATtestCases)

3.3.4 TEST DATA

TBC (E.g., www.github.com/XXXXXX/RATtestData)

3.3.5 DEFECT MANAGEMENT

A bug query which links to bug management system

3.3.6 TEST RESULTS

A hyperlink which links to test case management system

3.3.7 FUNCTIONAL TEST EXIT CRITERIA

1. All test cases should be completed.
2. All high risk areas are completely tested.
3. All software development activities are completed within the projected timelines.

4 TEST COMMUNICATION PLAN

4.1 TEST STATUS REPORTING

1. After tests are kicked off, all stakeholders will get daily test progress until the end of tests.
2. QA should immediately report P1 issue to PM & EPM & RD Manager.

3. Other, TBC/TBD

4.2 TEST DEFECT REPORTING

Severity	Definition	Handling
Show Stop/ Blocking Issue/ P1 issue	Defects that prevent further progress until resolved. In testing, the entire application component or function will not work. There is no work-around available, and testing cannot continue. This kind of defects will definitely impact testing schedule and project end-dates if not fixed immediately. Affected component cannot go into production without this defect resolved.	1. Report the blocking issue to PM, EPM, and RD Manager immediately. 2. Request developer to provide deadline of analysis and solution 3. QA needs to check if need to reschedule the test 4. QA runs tests to make sure there is no side effect before and after the code land on server
Serious/P2	Defects that occur in a major function and are grossly wrong. In testing, the entire application component or function will not work. There is a bypass available, and some testing can continue. In acceptance testing, defects that invalidate high or medium priority test cases but the end-date impact is manageable. Component can go into production, but requires manual user workaround.	1. Report the Serious issue to PM & EPM & RD Manager. 2. Request developer to provide deadline of solution 3. QA needs to verify patches after bug fixing
Moderate/P3	Defects that occur in a function but progress can continue. In testing, the function tested will not perform as expected but testing can continue. In acceptance testing, defects that invalidate low priority test cases but the end-date impact is manageable.	1. Triage these bugs to decide if we need to fix it on next build or put it on backlog 2. QA needs to verify patches after bug fixing
Low/P4	Defects are cosmetic, such as errors in format, spelling, wording, or UI defect. A defect in the work product under inspection which, if not fixed, would not cause a malfunction, would not prevent the attainment of a required result, and would not result in a Problem Report.	1. If it is UI defect, RD/QA needs to inform UX team to provide better solution. 2. If it is a minor functional defect, we can put it on backlog. 3. All defects need to be triaged

Table . Test Defect Reporting

5 TEST SCHEDULE

5.1 PROJECT SCHEDULE

5.1.1 MILESTONES/STATUS

Action Item / Milestone	Owner	Planned Date	Status	Notes/Next Steps

REFERENCE

[1] FEATURE_INTRODUCTION

[2] PRODUCT_SPECIFICATION

[3] HLD, PRD

[4] BUG_MANAGEMENT_SYSTEM

[5] TEST_CASE_MANAGEMENT_SYSTEM