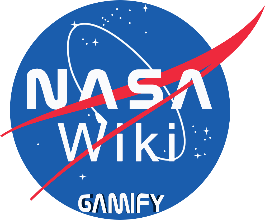
**NASA EVA**

**Gamification**

**TEST PLAN**

**PHASE II**



**Prepared By**

Laura Addiego

Samia Alam

Kelli Corey

Charles Milk

Adeola Odusola

Hung Pham

Table of Contents

[Revision Table 3](#_Toc518219743)

[1. Introduction 3](#_Toc518219744)

[1.1 Scope 3](#_Toc518219745)

[1.1.1 In Scope 3](#_Toc518219746)

[1.1.2 Out of Scope 4](#_Toc518219747)

[1.2 Objective 4](#_Toc518219748)

[1.2.1 Primary Objective 4](#_Toc518219749)

[1.2.2 Secondary Objective 4](#_Toc518219750)

[1.3 Roles and Responsibilities 4](#_Toc518219751)

[1.4 Assumptions and Constraints 5](#_Toc518219752)

[1.4.1 Assumption 5](#_Toc518219753)

[1.4.2 Constraints 5](#_Toc518219754)

[2. Test Strategy 5](#_Toc518219755)

[2.1 Unit Testing 5](#_Toc518219756)

[2.1.1 Definition 5](#_Toc518219757)

[2.1.2 Participants 6](#_Toc518219758)

[2.1.3 Methodologies 6](#_Toc518219759)

[2.2 System Testing 6](#_Toc518219760)

[2.2.1 Definition 6](#_Toc518219761)

[2.2.2 Participants 6](#_Toc518219762)

[2.2.3 Methodologies 6](#_Toc518219763)

[2.3 Database Testing 6](#_Toc518219764)

[2.3.1 Definition 7](#_Toc518219765)

[2.3.2 Participants 7](#_Toc518219766)

[2.3.3 Methodologies 7](#_Toc518219767)

[2.4 Test Completeness Criteria 7](#_Toc518219768)

[2.5 Bug Severity and Priority Definition 7](#_Toc518219769)

[Severity Levels 7](#_Toc518219770)

[Priority Levels 8](#_Toc518219771)

[2.6 Bug Reporting 8](#_Toc518219772)

[2.7 Features to be Tested 9](#_Toc518219773)

[2.8 Features Not to be Tested 9](#_Toc518219774)

[3. Test Deliverables 9](#_Toc518219775)

[4. Resources and Environments 10](#_Toc518219776)

[4.1 Test Environment 10](#_Toc518219777)

[4.1.1 Hardware 10](#_Toc518219778)

[4.1.2 Software 10](#_Toc518219779)

[4.2 Testing tools 10](#_Toc518219780)

[5. Test Schedule 10](#_Toc518219781)

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Change Description** | **Author** | **Date** |
| V1.0 | Initial Test Plan Draft | Samia Alam | 6/17/18 |
|  |  |  |  |

# 1. Introduction

This Test Plan demonstrates all the processes associated with phase 2 of NASA EVA Wikimedia project testing. The purpose of this document is to describe strategies, workflows, processes and methodologies used to plan, execute, and manage testing of NASA EVA Mediawiki application. This document is intended to be used by all members of the development team including Project manager, QA team, Developers, and business analysts.

## 1.1 Scope

### 1.1.1 In Scope

The NASA EVA Media wiki Phase 2 Test Plan defines testing approach associated with unit testing, functional testing, Database testing, system testing, and Usability testing. The test scope includes:

* Performing Unit Testing to ensure the codes fulfill all the requirements.
* Testing of all functional requirements listed in the Software Requirements Document.
* Doing System Testing to ensure the entire system fulfills the requirements provided.
* Completing Database Testing to ensure the database fulfills all the database-specific requirements.
* Doing Performance Testing testing to make sure the new features did not negatively impact the User Interface and performance of the overall application.
* Doing User Interface Testing to ensure adequate usability of the features in the front end

### 1.1.2 Out of Scope

The following testing areas are out of scope for the NASA EVA Media wiki test plan and test scope:

* User acceptance testing.
* Functional requirements testing for features outside of NASA EVA wiki project phase 2.

## 1.2 Objective

### 1.2.1 Primary Objective

The Primary objective of testing the new features of the application is to ensure that the system meets all the requirements including functional and non-functional requirements. Additionally, the objective is to verify that the system satisfies all the use case scenarios and maintains acceptable performance and quality of the product. The end user should be able to see and confirm that all requirements have been met at the end of the project.

Any changes to the requirements document, functional or design specifications will be documented and tested within the remaining time of the project,

### 1.2.2 Secondary Objective

The secondary objective of testing is to identify all issues, risks, or design flaws, and communicate all the issues to the team to make sure that the issues are addressed before going out to production.

## 1.3 Roles and Responsibilities

Different team members will fulfill the following roles and responsibilities:

a) Developers will develop the new features of the application

b) Developers will perform unit testing

c) QA Analysts will perform functional testing, System testing, and Usability Testing.

d) The database development lead will complete Database testing

e) QA Analysts will create and maintain test cases and defect tracking processes.

f) QA Lead will provide and manage the test plan, test schedule and the final sign off for the product to be released to production.

g) Business Analysts will develop and update Requirements Documents as needed based on gaps or issues identified throughout development and testing.

## 1.4 Assumptions and Constraints

### 1.4.1 Assumption

Some of the assumptions during testing phase are:

a) The developers have completed all unit testing to ensure code fulfills are requirements.

b) User acceptance testing will be done by the end-user.

c) Test results will be reported using Test result summary document

d) Test Cases or Scripts are developed and approved by the time test execution begins.

e) Prior features of the application have already been tested and validated.

### 1.4.2 Constraints

Some of the constraints associated with testing are:

a) Short project schedule may put testing completion at risk

b) Developers may not have enough time to add all the unit tests due to lack of time.

c) Some team members not knowing PHP would prevent them from helping with Unit Testing

# 2. Test Strategy

The testing strategy for this project is to thoroughly test each requirement in the Software Requirements Specification document. The strategy is to include both valid and invalid scenarios to ensure that the system functions properly during valid scenario, and rejects invalid paths. The overall testing will be divided into following categories: Unit Testing, System testing, and Database Testing.

## 2.1 Unit Testing

### 2.1.1 Definition

Unit test is a white-box testing activity which analyzes internal structure of the application by using program source code as the test basis. The primary intention of Unit test is to determine errors in programming logic. The goal of unit test is to have 100% code coverage and reasonable data coverage to ensure individual parts of codes are working correctly. It is based on functional and data requirements mentioned in the requirements specifications. The objective of unit test is to make sure that the individual components or units of codes work properly and meet their requirements. Overall, Unit test is used to improve code quality by verifying each unit of code used to fulfill functional requirements.

### 2.1.2 Participants

Developers are responsible for completing unit testing.

* Adeola Odusola,
* Hung Pham, and
* Laura Addiego

### 2.1.3 Methodologies

Each line of new code will be tested to search for error in programming logic. Test Script will be created to execute Unit Testing. A PHP code editor such as notepad, NetBeans, PHPStorm or any other code editor is required to execute unit testing

## 2.2 System Testing

### 2.2.1 Definition

System Testing is a form of black-box testing which examines the complete and integrated system. System Testing includes evaluating the system’s compliance with requirements from the software specification document. There are various types of system testing such as functional testing, performance testing, usability testing, and many others. Following types of system testing will be done for the project: Functional testing, performance testing, and regression testing.

### 2.2.2 Participants

QA Lead is responsible for completing System testing.

* Samia Alam

Business Analyst is also responsible for validating the functionality, hence may participate in functional testing

* Charles Milk (Chuck)

### 2.2.3 Methodologies

System Testing will begin after Unit testing has been completed, and the new code has been integrated into the current application. Test script will be created to execute System testing. The script may be created in Microsoft Word Document.

## 2.3 Database Testing

### 2.3.1 Definition

Database testing involved evaluating schema, tables, triggers, and many other aspects of database. Overall data integrity and consistency based on requirements are tested through Database Testing.

### 2.3.2 Participants

Database Development Lead is responsible for completing Database testing.

* Laura Addiego

### 2.3.3 Methodologies

Database Testing will be done after Database updates in Phase 2 have been completed. Test script will be created to execute Database Testing.

## 2.4 Test Completeness Criteria

Successful completion of testing will be determined by the application’s ability to fulfill all the functional and non-functional requirements from the requirement specification document. The application will be approved to be released to production only when all the test cases pass, and the application meets the client expectations based on the defined requirements.

## 2.5 Bug Severity and Priority Definition

Bug severity and priority levels are defined in the tables below. Testers should assign a severity and priority level when submitting a bug to ensure that severe and higher priority bugs get fixed right away. When bugs are identified, they will be addressed based on the level of severity and priority. The ultimate goal is to fix all the bugs, but due to time constraints, bugs with low severity and low priority might not get fixed. Any issue identified during Unit testing and database testing will get fixed right away.

### Severity Levels

|  |  |  |
| --- | --- | --- |
| **Severity ID** | **Level** | **Description** |
| 1 | Critical | The application or a feature crashes due to the bug. There is no workaround to bypass the bug. Example: System crashes; database or file gets corrupted; data loss. Bugs with severity level of 1 must get fixed. |
| 2 | High | Critical features become unusable, or show incorrect functionality, or other areas of the system become unusable due to the bug. Even if a work around exists, it is inconvenient or difficult. |
| 3 | Medium | Incorrect functionality of the application or feature, but easy workaround exists. |
| 4 | Low | Documentation or cosmetic error that does not disrupt any functionality |

### Priority Levels

|  |  |  |
| --- | --- | --- |
| **Priority ID** | **Level** | **Description** |
| 4 | Highest Priority (Must Fix) | This bug must be fixed in order to be released to production |
| 3 | Should Fix | Allowing this bug to be out in production would cause customer dissatisfaction |
| 2 | Fix If Have Time | Fix the issue if it does not delay the release date to production. |
| 1 | Low Priority | Fix these bugs when all other higher priority bugs have been fixed. |

## 2.6 Bug Reporting

When a bug or defect is identified, it should be reported using the following bug report template:

|  |  |
| --- | --- |
| **Bug Report** | |
| **Bug ID** | <<may start with BI01>> |
| **Title** | <<Title indicates the issue being reported in the bug report>> |
| **Reporter** | <<Name of the tester>> |
| **Date** | <<Current date when the bug is being reported>> |
| **Details** | <<Description:  Steps to Reproduce:  Expected outcomes (Associated requirement):  Actual Outcomes:  Additional Details if Available: >> |
| **Severity** | <<Specifies severity level>> |
| **Priority** | <<Specifies Priority of the issue>> |
| **Assigned To** | <<Name of the developer who should fix the issue>> |

## 2.7 Features to be Tested

Following features will be tested during phase 2 of the NASA EVA gamification project:

* Point Assignment System based on adding or editing a page
* Badge Configuration based on total point accumulation
* Leaderboard Behaviors and configurations
* Database Integrity after updating the existing database

## 2.8 Features Not to be Tested

Following features will not be tested during phase 2 of the the NASA EVA gamification project:

* Accuracy of the MediaWiki content
* Quality of the information added or edited by a user
* Point management for work done by users prior to implementation of the gamification features

# 3. Test Deliverables

The following test deliverables will be produced throughout the project:

|  |  |
| --- | --- |
| **Test Deliverables** | **Description** |
| Test Plan | Includes testing scope, strategies, schedule, tools, and overall processes that should be followed during testing |
| Test Cases/Script | Set of conditions and variables that will be used to test whether the features fulfill the design requirements |
| Bug Report | When an issue is found during testing, the tester will create a bug report using the bug report template to submit the issue to the developer |
| Test Result Summary Report | Summarized version of test plan and testing outcomes, indicating overall status of testing |

# 4. Resources and Environments

## 4.1 Test Environment

### 4.1.1 Hardware

Any Windows PC desktop or laptop that can be configured to fulfill the software requirements can be used for testing.

### 4.1.2 Software

The following environment requirements should be fulfilled to be able to test:

* PHP (5.6) (<http://windows.php.net/download#php-5.6>)
* MariaDB (5.5.59) ([https://downloads.mariadb.org/mariadb/5.5.59/)https://downloads.mariadb.org/mariadb/5.5.59/](https://downloads.mariadb.org/mariadb/5.5.59/)https:/downloads.mariadb.org/mariadb/5.5.59/))
* MediaWiki (1.27.4) ([https://www.mediawiki.org/wiki/Download)https://www.mediawiki.org/wiki/Download](https://www.mediawiki.org/wiki/Download)https:/www.mediawiki.org/wiki/Download))
* IIS or IIS express (7 or higher)

## 4.2 Testing tools

* WireShark will be used to test the website’s performance.
* Microsoft Word Document will be used to create and maintain Test Cases, and bug reporting.
* Any test scripts related to unit testing or database testing may be created using notepad, NetBeans or any other tool that the team may choose to use.

# 5. Test Schedule

Test Cases will be prepared for the future execution while the developers finish up coding. After completion of the coding phase, all the test cases will be executed during the testing phase. Testing phase will be completed by 08/05/2018. Upon the Final QA sign off, the product will be approved for deployment. The Dates in Test Schedule may change based on unplanned roadblocks or delays in code delivery.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testing Tasks** | **Start date** | **Planned End Date** | **Actual End Date** | **Responsible Person** |
| **Create Test Cases/Scripts** | 7/2/2018 | 7/29/18 |  | Samia |
| **Execute Test Cases** | 7/30/18 | 8/5/18 |  |  |
| Unit Testing | 7/30/18 | 7/31/18 |  | Adeola, Hung, Laura |
| Database Testing | 7/30/18 | 7/31/18 |  | Laura |
| System Testing |  |  |  |  |
| Functional Testing | 8/1/18 | 8/3/18 |  | Samia, Chuck |
| Performance Testing | 8/2/18 | 8/3/18 |  | Samia |
| Regression Testing | 8/4/18 | 8/5/18 |  | Samia, Chuck |
| **QA Sign Off for Release** | 8/5/18 | 8/5/18 |  | Samia |