**Test Plan**

**NASA EVA Gamification**

*Group 3*

**Prepared By**

Okechukwu Ogudebe

Victoria Guadagno

Jacqueline Macfadyen

Michael Salgo

Kevin Fortier

Table of Contents

[Introduction 3](#_Toc508555377)

[Objectives and Tasks 3](#_Toc508555378)

[Objectives 3](#_Toc508555379)

[Tasks 3](#_Toc508555380)

[Scope 4](#_Toc508555381)

[Scope of Testing 4](#_Toc508555382)

[Tactics 4](#_Toc508555383)

[Testing Strategy 5](#_Toc508555384)

[Unit Testing 5](#_Toc508555385)

[Definition 5](#_Toc508555386)

[Participants 5](#_Toc508555387)

[Methodology 5](#_Toc508555388)

[System Testing 5](#_Toc508555389)

[Definition 5](#_Toc508555390)

[Participants 5](#_Toc508555391)

[Methodology 6](#_Toc508555392)

[Environment Requirements 6](#_Toc508555393)

[Control Procedures 6](#_Toc508555394)

[Problem Reporting 6](#_Toc508555395)

[Level 1 6](#_Toc508555396)

[Level 2 7](#_Toc508555397)

[Error Reporting Form 7](#_Toc508555398)

[Features to Be Tested 7](#_Toc508555399)

[Features Not to Be Tested 7](#_Toc508555400)

[Schedules 7](#_Toc508555401)

[Major Deliverables 8](#_Toc508555402)

[Tools 8](#_Toc508555403)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Description of Change** | **Author** | **Date** |
| 1.0 | Initial Creation of Document | Okechukwu Ogudebe | 03/10/2018 |
| 1.1 | Revised Document | Victoria Guadagno | 03/11/2018 |

# Introduction

This Test Plan documents and tracks all the changes made to the NASA Gamify Test Plan. All the information recorded is vital in testing the final product. The Test Plan document was created as part of the second milestone by the team. Its intended recipients are the team lead, project team, instructor, and the testing team. Some portions of this plan will be shared with the user and other stakeholders whose input approval is required in the testing phase.

Our primary purpose is to enhance user-interaction on a MediaWiki environment through gamification. Gamification is the technique of applying game principles and design elements in non-game contexts. It improves the user experience by breaking down complex tasks into manageable tasks. Every user hopes to rise to the most prestigious level on the system. Other users can track their progress and use them as benchmarks on where they want to be. Users will receive badges based on their activities.

Testing is vital to the success of this project and helps identify significant loopholes. Project documentation and test cases will act as the blueprint for future developers and users. MediaWiki is used as a data repository and will be the backbone of this project.

# Objectives and Tasks

## Objectives

Gamification has positive effects on individuals as it encourages participation. Users can earn badges based on the number of posts made, the number of pages edited, the content shared, and interactions with other users, just to name a few generic examples – not to be included in this current test plan. This basic desire for recognition contributes positively towards the growth of the game. Creating a Test Plan is essential at ensuring that all the functions are working perfectly. We seek to accomplish the following objectives in the Test Plan:

|  |  |
| --- | --- |
| **ID** | **Objectives** |
| O1 | Database design and test |
| O2 | Game test |
| O3 | User Profile test |

## Tasks

Here are the tasks identified in this Test Plan

|  |  |
| --- | --- |
| **ID** | **Tasks** |
| T1 | Overall game testing |
| T2 | User-interface testing – No CSS and JavaScript errors |
| T3 | Reward system testing – Allocation of badges works seamlessly |
| T4 | Database testing |
| T5 | Penetration testing – To ensure user security |
| T6 | Game render testing |
| T7 | Post-testing |
| T8 | Problem reporting |
| T9 | Unit testing |

# Scope

## Scope of Testing

With the introduction of new modules in the development phase, software bugs are inevitable in the game. The team will conduct the following tests to ensure system integrity:

1. User interface testing
2. Reward system testing
3. Database testing
4. Testing of automated features
5. Functional and non-functional testing
6. User tracking
7. System stability tests

New changes made to the code could contribute to system instability. Our team will also compare MediaWiki performance before and after modifications. Performance tests include page load index, SEO performance, browser compatibility and security.

## Tactics

The following tactics will be employed in accomplishing the tasks listed in the scope section:

1. User interface testing – Modern web browsers offer web development tools vital for benchmarking a web page's performance. Since our game is web based, we can use Mozilla and Google Chrome tools for benchmarking the game’s UI.
2. Reward system testing – Badge allocation is a vital component for the game. Testers will first use dummy data and will be under the supervision of one of the team members.
3. Database testing – Modules of the database will be subject to tests. Tables, data models, and the database schema has to conform to referential integrity. The database developer and test lead will run tests on the database to ensure everything works perfectly.
4. Testing of automated features – Automated processes such as badge allocation will be checked for errors by the team. Automated functions can boost the game’s interaction levels and reduces the administrator’s workload.
5. Functional and non-functional testing – The game is to be tested to ensure that it conforms to modern software requirements. The product should meet its functional requirements.
6. User tracking – Tracking user’s activities is useful for the rewards system. Cookies and session data should not be used to spy on the users. The integrity of the user tracking module is key for game’s success.
7. System stability tests – The overall security and performance of the system has to be analyzed before the deployment of the game. Added modules are to be checked for compatibility problems with the game.

A master test schedule will be unveiled for team, developers, and testers.

# Testing Strategy

Testing will be sub-divided into two main testing categories. One category will be unit testing while the other handles system testing. The testing team is to prepare a comprehensive report on their findings and their proposed solutions.

## Unit Testing

### Definition

Unit testing is a software development process which tests the smallest possible sections of code to be sure each section works correctly. Various components are tested for functionality and to be sure there are no errors. System integration testing of the modules will be conducted to ensure the new user reward system functions perfectly together with the game.

### Participants

Okechukwu Ogudebe – Testing Lead

### Methodology

In unit testing, individual pieces of code are checked for errors. Okechukwu Ogudebe, the testing lead, will create the test plan, database scripts, game scripts, and user profile scripts. Tools required for this exercise include PHP code editors (i.e. Sublime, Notepad, PHPStorm, and Netbeans).

## System Testing

### Definition

System Testing evaluates the system as a whole. This includes meeting the requirements and all functionality. Inner knowledge of the design and code is not required. In this section, awarding badges will be tested for its compliance. This can be done using dummy data and user interaction.

### Participants

Okechukwu Ogudebe – Test Lead

### Methodology

Okechukwu Ogudebe is tasked with the duty of creating test scripts for system and integration testing. System testing will begin after unit testing has been completed. This is to reduce the workload on the system testing team if there are any changes in the game’s unit.

# Environment Requirements

The game’s environment is as follows:

* IIS or IIS express (7 or higher)
* 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/>
* MediaWiki (1.27.4)
  + <https://www.mediawiki.org/wiki/Download)https://www.mediawiki.org/wiki/Download>
  + Requires 7-Zip to unpack .tar.gz file on Windows
    - <http://www.7-zip.org/)http://www.7-zip.org/>

Tools needed for testing

* Code editor – Can be Notepad, Sublime, Eclipse or Netbeans
* Browsers – Firefox and Google Chrome
* Penetration testing software – Wireshark, Kali Linux

# Control Procedures

## Problem Reporting

Problem reporting helps in the identification of complications developed while gamifying the NASA EVA MediaWiki. Problematic areas need to be highlighted and the corresponding corrective actions implemented. Different procedures are laid out to ensure that the team identifies and tracks system problems. Problem reporting will be broken into two levels.

### Level 1

Level 1 constitutes all high-level problems that would contribute to system breakdown. These are high-risk errors that require technical solutions. They include:

* Software bugs – MediaWiki is written in PHP and gamifying the framework may contribute to system breakdown. Software bugs can affect the usability of the system and in extreme cases lead to system shutdown. Each new module introduces to MediaWiki will be checked for compatibility, PHP language standards, code cleanliness, and data types.
* Database errors – As discussed above the introduction of new modules introduce new complexity to the system. It will be quite easy for the developers to break MediaWiki by altering the database without first understanding the underlying features. These errors will most likely be highlighted by the compiler while trying to parse data from the database.
* Security – Integrating new modules on MediaWiki will most likely introduce new security challenges to the system. Tweaking user session tracking can compromise user’s passwords to third-parties. Security problems can lead to lawsuits and ultimately taint the company’s image.

### Level 2

In level 2, we will be tracking problems that are less likely to break down the game and require little or no technical know-how when fixing. They include:

* User unable to post or edit content – Could be a problem with browser or user has insufficient privileges.
* Web page is not loading – This can be as a result of an incorrect setup of the development environment. Microsoft’s IIS services cannot render PHP without the necessary configuration.
* Tracking badges – Database errors and session data could be tampered with.

Each member of the team is required to document any problems that they encounter during the development of the system. Regular meetings are scheduled to ensure that any problem with the game are detected early.

## Error Reporting Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Error ID** | **Name** | **Error Level** | **Error Description** | **Date** |
|  |  |  |  |  |

# Features to Be Tested

The following gamification features will be tested:

* User’s Badges
* Badge and user profile tracking
* User administrative privileges
* Development environment
* Database integrity
* User interface tests

# Features Not to Be Tested

The following functionality will not be tested:

* Web page optimization
* Authenticity of the content
* Game experience

# Schedules

When it comes to game design, unforeseen stumbling blocks can affect the schedule and delivery date. To counter this, extra time should be added into the schedule. Team members will work overtime if they fall behind on their tasks.

## Major Deliverables

These are the testing-related major deliverables.

* Test Plan – The entire schedule of what is to be tested. Planning eases the process and helps identify overlapping tests. Coordinate the key in identifying all potential problems with the game.
* Test Cases – Tests cases are fundamental in finding problems with the application.
* Test Incidence Reports – When a problem is noted with the game. The tester is required to create a simple report on what they have encountered. Incidence reports will be compiled based on their intensity and go into the final report.
* Test Summary Reports – Summarized version of the Test Plan report that can be shared with the general public.

# Tools

Tools used in the testing of the game include a Windows PC configured to run the following software:

1. IIS or IIS express (7 or higher)
2. PHP (5.6) (http://windows.php.net/download#php-5.6)
3. MariaDB (5.5.59) (https://downloads.mariadb.org/mariadb/5.5.59/)
4. MediaWiki (1.27.4) (https://www.mediawiki.org/wiki/Download)

The PHP development environment automatically logs bugs that it encounters. Manual bug tracking will also be done by the team through spreadsheets. All problems should be recorded in the appropriate tracking spreadsheet. The logs will be helpful in preventing further system breakdown.