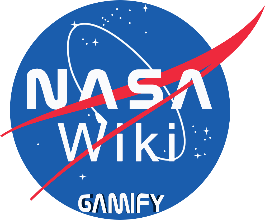
**NASA EVA**

**Gamification**

**G A M E D E S I G N**

**PHASE II**



**Prepared By**

Laura Addiego

Samia Alam

Kelli Corey

Charles Milk

Adeola Odusola

Hung Pham

# Table of Contents

[Table of Contents 3](#_Toc518218882)

[Revision Table 4](#_Toc518218883)

[Introduction 4](#_Toc518218884)

[Purpose 4](#_Toc518218885)

[Scope 4](#_Toc518218886)

[Document Overview 4](#_Toc518218887)

[Software Architecture Overview 4](#_Toc518218888)

[Data Design 4](#_Toc518218889)

[Software Architecture Design 5](#_Toc518218890)

[Software Design Description 5](#_Toc518218891)

[Component Name: Point Configuration 5](#_Toc518218892)

[Point Configuration Design Description: 5](#_Toc518218893)

[Workflows: 5](#_Toc518218894)

[Point configuration Pseudocode: 6](#_Toc518218895)

[Component Name: Badge Configuration 6](#_Toc518218896)

[Badge Configuration Design Description: 7](#_Toc518218897)

[Workflows: 7](#_Toc518218898)

[Badge configuration Pseudocode: 7](#_Toc518218899)

[Component Name: Leaderboard 8](#_Toc518218900)

[Leaderboard Design Description: 8](#_Toc518218901)

[Workflows: 8](#_Toc518218902)

[Leaderboard Pseudocode: 8](#_Toc518218903)

[User Interface Design 9](#_Toc518218904)

[Overview of User Interface 9](#_Toc518218905)

[Leaderboard User Interface Image 9](#_Toc518218906)

[Requirements Mapping 9](#_Toc518218907)

[Regulatory Requirements 10](#_Toc518218908)

[Non-Functional Requirements 10](#_Toc518218909)

[Functional Requirements 10](#_Toc518218910)

# Revision Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Description of Change** | **Author** | **Date** |
| 1.0 | Initial Creation of Document | Samia Alam | 6/28/18 |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this software design document is to describe the architecture and system design of NASA EVA Gamification project phase 2. The document describes component design, data design, user interface design, and requirement mapping of the features that will be developed during the project. The intended audiences are developers, designers, and testers.

## Scope

The primary focus of this phase of NASA EVA gamification project is to build an extension that provides a consolidated infrastructure, allowing for the addition of gamification elements to the NASA Wiki environment. This phase of the project will focus on three areas of enhancements: point measurement, badge assignment, and a leaderboard. Users will be given points for contributing to the mediawiki. Phase 2 will only focus on earning points for adding a new page or editing an existing page. Different badges will be earned based on different amount of point accumulation. The overall points, contributions, badges, and current position among other users will be viewable through the leaderboard that shows the relative position of the user. Overall, the enhancements will fulfill different aspects of gamification including encouraging and motivating users to contribute more.

## Document Overview

This document describes the system architecture, Software design, data design, user interface design, and requirement matrix associated with Phase 2 of NASA EVA Gamification project .

# Software Architecture Overview

The phase 2 of NASA EVA gamification project architecture has four main components:

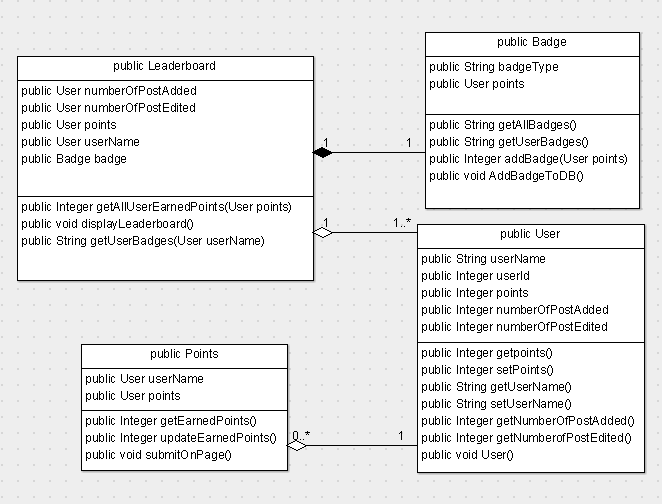
* Updating the gamification database table
* Point Assignment Configuration
* Badge Configuration based on point accumulation
* Scaled Leaderboard creation

## Data Design

All MediaWiki data will be stored in specified tables within MariaDB database. See the database design documentation for further details.

## Software Architecture Design

This software architecture design provides a basic concept of different classes that are interdependent on each other to fulfill the expected functionalities defined in the requirements document.



# Software Design Description

## Component Name: Point Configuration

Point configuration is the base of the gamification of NASA EVA MediaWiki. User will earn points based on their contribution to mediawiki. Point assignment will vary based on the type if action that user does. Different amount of point will be assigned to a user based on whether they have added or edited a page.

### Point Configuration Design Description:

Different point value will be set for a defined add and edit action taken by user. If the action is an add, then 100 points will be added to the user. If the action taken by the user is an edit, then 50 point will get added to the user. The points will be stored in user’s table associated with points.

### Workflows:

* When a user creates a page, 100 points will be assigned to the user
* When a user edits a page, 50 points will be assigned to the user

**Point Configuration Class name:** Points

**Class Description:** Points class will include the attributes and functions needed for point assignments to users.

**Functions:**

* getEarnedPoints($user): this function will get points earned by a user.
* updateEarnedPoints ($user, $points\_earned): This function will update point counts for users when more points are earned.
* onPageSubmitComplete ($user): This function will allow the system to submit the points to the MediaWiki page.

### Point configuration Pseudocode:

Class Points {

Function getEarnedPoints ($user) {

Get points\_generated in table revision from database.

}

Function updateEarnedPoints ($user, $points\_earned) {

update total\_points\_generated to table user from the database.

Class Badges.addBadge($user, $points\_Earned)

}

Function onPageSubmitComplete ($user) {

initial $points\_Earned = 0;

$points\_Earned = $this->getEarnedPoint ($user);

if (page is a new page) {

$points\_Earned += 100;

else

$points\_Earned += 50;

}

$this->updateEarnedPoint ($user, $points\_Earned)

}

}

## Component Name: Badge Configuration

The badge feature is added as a reward system for the users based on their contribution. Badges will be configured based on total point accumulation. This will fulfill the gamification aspect of encouraging users to increase contribution to earn different badges.

### Badge Configuration Design Description:

Badges will be configured in an array based on different values. Different images of the badge will be stored in a file. Based on different badge selection, the specified image of the badge will be set in associated user’s profile.

### Workflows:

* When a user earns 50 point, bronze badge gets assigned to the user
* When a user earns 100 point, silver badge gets assigned to the user
* When a user earns 200 point, gold badge gets assigned to the user
* When a user earns 500 point, platinum badge gets assigned to the user
* When a badge is earned, it is displayed on top of the user’s profile page

**Badge Configuration** **Class name:** Badge

**Class Description:** badge class will include the attributes and functions needed for different badges earned by users.

**Functions:**

* getAllBadges (): this function will get all badges from the database.
* getUserBadges (): This function will get badges for a specific user from the database
* addBadgeToDB ($user, $badge\_id): This function will allow badge information associated with a user to be added to the database.
* addBadge ($user, $points\_Earned): this function will allow adding badge to user based on different points accumulation.

### Badge configuration Pseudocode:

Class Badge {

Function getAllBadges () {

Get badges from table mv\_badges in database.

}

Function getUserBadges () {

Get badges from table mw\_user\_badges in database for each user.

}

Function addBadgeToDB ($user, $badge\_id) {

insert badge into table mv\_user\_badges in database.

}

Function addBadge ($user, $points\_Earned) {

Initial $points\_Earned = 0;

Initial $badges array = empty;

$points\_Earned = Class Point.getEarnedPoints($user);

$badges = $this->getAllBadges();

foreach $badges as $badge\_id=>$value {

if $points\_Earned >= $value

$this->addBadgeToDB($user, $badge\_id);

}

}

}

## Component Name: Leaderboard

Leaderboard will allow the users to view their position or status in the game as they can see the name of the other users with 100 points above or below them. It will also allow them to see how many pages they have added or edited.

### Leaderboard Design Description:

The design of the leaderboard would be in a table format as the desired data would be pulled by running a query. A query will be run to get the count of add, edit, and points associated with a user. The result from the query will be made visible in the leaderboard in user’s profile page.

### Workflows:

* When users navigate to their profile page, then can see a leaderboard which show the Add count, edit count, points earned, the user’s position in the leaderboard, and other users 100 points over and below.
* When Users click on the add count, they can view the name of the pages created.

**Leaderboard** **Class name:** LeaderBoard

**Class Description:** Leaderboard class will include the attributes and functions needed to display a leaderboard in the MediaWiki profile page.

**Functions:**

* getAllUserEarnedPoints (): this function will get points earned by different users from the database.
* getUserBadges ($user): This function will get badges for a specific user from the database
* displayLeaderBoard(): This function will allow displaying of different information including username, added post count, edited post count, points, and badges in the leaderboard.

### Leaderboard Pseudocode:

Class LeaderBoard {

Function getAllUserEarnedPoints (){

get point\_generated from the database for all users.

}

Function getUserBadges ($user) {

get all badges from database for each user.

}

Function displayLeaderBoard() {

Initial $point\_generated array = empty;

Initial $user\_badges array = empty;

Initial $user\_points\_Earned array = empty;

$point\_generated = $this->getAllUserEarnedPoints;

foreach ($point\_generated as $user) {

$user\_badges = Class Badges->getUserBadges ($user);

$user\_points\_Earned =Class Points->getEarnedPoints($user);

}

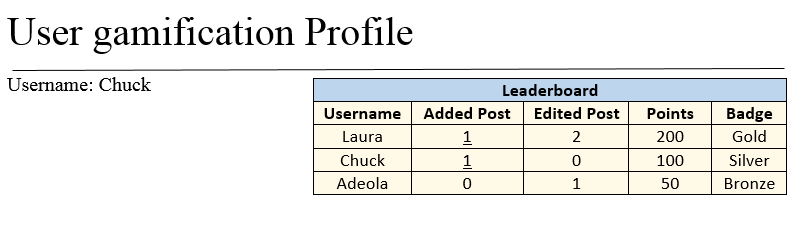
}

# User Interface Design

## Overview of User Interface

The User Interface will allow users to view their earned points, badges, and leaderboard associated with the EVA gamification. When user logs into their account, they will view the leaderboard in the profile page by default. The added post count will appear in link format. When user clicks on the linked value, the name of the pages the user added will be displayed. The different criterion of the leaderboard will be displayed in different columns. When user goes to a different page, the leaderboard will not be displayed anymore. The badges earned by the user will be displayed on the top right corner of the page.

## Leaderboard User Interface Image



# Requirements Mapping

The design satisfies the following business and functional requirements:

## Regulatory Requirements

|  |  |
| --- | --- |
| **ID** | **Business Requirements** |
| R5 | All mediawiki gamification related metadata and records should be managed in MariaDB or MySQL |
| R4 | All Code and extensions should be written in PHP |

## Non-Functional Requirements

|  |  |
| --- | --- |
| **ID** | **Non Functional Requirements** |
| NF1 | Configuring any variables should be allowed through a config PHP file |

## Functional Requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Component** | **Functional Requirements** |
| T-PA1 | Points | Users shall earn 100 points when a page is created |
| T-PA2 | Points | Users shall earn 50 points when a page is edited |
| T-BG1 | Badges | User with 50 total points shall earn a bronze badge |
| T-BG2 | Badges | User with 100 total points shall earn a silver badge |
| T-BG3 | Badges | User with 200 total points shall earn a gold badge |
| T-BG4 | Badges | User with 500 total points shall earn a platinum badge |
| T-BG5 | Badges | Badges shall be displayed on top of the user’s profile page in a tabular form with the earned badge images |
| T-LB1 | Leaderboard | Add count, edit count, points earned, and the user’s position among others will be displayed in leaderboard |
| T-LB2 | Leaderboard | Other users with 100 points more or less than the current user will be displayed in the leaderboard |
| T-LB3 | Leaderboard | Users shall be able to click on the edit count to view the name of the pages created |
| T-LB4 | Leaderboard | Leaderboard will only be visible in user profile page |

## 