**Handover Document – Future Recommendations**

**NASA EVA Gamification**

*Group 3*

**Prepared By**

Michael Salgo

Victoria Guadagno

Okechukwu Ogudebe

Jacqueline Macfadyen

Kevin Fortier

Montrell Nuble

Table of Contents

[Introduction 3](#_Toc511756088)

[Historical Date Earned for Badges 3](#_Toc511756089)

[SQL Code Improvements 3](#_Toc511756090)

[Generic Names for Badges 4](#_Toc511756091)

[Extention.json Config Changes 4](#_Toc511756092)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Description of Change** | **Author** | **Date** |
| 1.0 | Initial Creation of Document | Michael Salgo | 04/07/2018 |
| 1.1 | Revised Document for Formatting and content | Victoria Guadagno | 04/08/2018 |
| 1.2 | Revised Document for Formatting and content and added Introduction/SQL Code Improvements text | Michael Salgo | 04/09/2018 |
| 1.3 | Added Text detailing how the config may need to be changed | Kevin Fortier | 04/17/2018 |

# Introduction

The purpose of this document is to list items of future recommendations to teams that are going to continue work where we left off. The lessons learned from our semester will help future semesters in their research and with any possible questions they had regarding the design.

# Historical Date Earned for Badges

The installation of the NASA\_EVA\_Gamification extension creates the gamification\_badges table. It then gathers pre-existing data about users from MediaWiki's database. It uses that data to determine which badges users have earned retroactively and updates the gamification\_badges table with these badges.

The Spring 2018 semester implemented the email verification game. We had three options to store in the date\_badge\_earned field when the badge was earned retroactively:

* Null
* MediaWiki's email verification date (the user\_email\_authenticated field, as seen here: https://www.mediawiki.org/wiki/Manual:User\_table#user\_email\_authenticated)
* Date of extension installation

We debated this choice for some time before deciding to store null. We felt that using the date the user installed the extension would be confusing. This date would be displayed for all badges earned before the extension was installed and would quickly become meaningless.

The ideal scenario would be to use the date that the badge would have been earned. While this would have been easy to implement in this game, it would not be possible for all games created in the future. Storing inconsistent data in the date\_badge\_earned field would be confusing.

That left us with leaving the field with a null value. This was the most consistent and least confusing option.

# SQL Code Improvements

In designing the gamification\_badges table, we initially wanted to incorporate a foreign key on the user\_id field, but we had discovered that MediaWiki does not apply foreign keys in its database tables so we opted out of using it for consistency purposes, but perhaps some more research can be done by future teams. Additionally, we were looking at implementing an index on the user\_id field for optimal searching by user\_id, but this was not something we were able to fit in in our timeline and it is something additional that future teams can research and look into.

# Generic Names for Badges

We used NASA-themed wording in all configurable values. We recommend that you use generic wording when you submit the extension to MediaWiki. Using generic names will allow the extension to appeal to a wider audience. Instead of calling the email verification badge "Spacesuit Verified" you could call it "Email verified." You can do the same with the badge ranks. Instead of "Astronaut", "Spacewalker," and "Moonwalker" you could call them "Bronze", "Silver," and "Gold."

# Extention.json Config Changes

The initial Email Verification game added to the extension was a “one and done” style game; either a user has earned it or not. As such, the global config variable was constructed with only that style game in mind. Future efforts on adding more games will need to account for games that meet some threshold (i.e. a minimum of 50 edits for a rank 1 ‘editor’ type game).

Two possible implementation paths for adding “tiered” games include:

* Adding additional complexity to the GamesToRankMapping value for each game
* Adding an additional extension associative array variable, where the key:value pair would be the game’s internal i18n tag and numerical value to meet the requirement.

The first option adds a lot of complexity to the syntax of the GamesToRankMapping variable, where a novice administrator may not be sure where to open and close array boundaries. The second option opens the door to mismatch or missing values between the added array and the GamesToRankMapping variable.