**McMaster Gamification**

Submitted by: InsertTeamNameHere

Submitted to: Hao Lac

ICET Department,

School of Engineering Technology and Applied Science

Progress Campuse, Block A

Centennial College

Discipline: Software Engineering (insertSubDisciplineHere(e.g.,IG\_HIT))

Date: InsertDueDateHere

# Declaration of Sole Authorship

We, insertTeamNameHere, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed: insertTeamMemberSignaturesHere (programOfStudy)

Date: insertDueDateHere

# Abstract

Provide an abstract in this section in 200 words or less in paragraph format.

An accurate condensation of the TR. State the main idea or thesis by answering questions such as: 

* What is the TR about? 
* Why is it significant? 

What should I do about it?

# Table of Contents

[**Declaration of Sole Authorship**](#_ivxuupdnryj1) **2**

[**Abstract**](#_png97umuc7ht) **3**

[**Table of Contents**](#_vcrqkmh6x3mj) **4**

[**List of Illustrations, Diagrams, and Tables**](#_ojv08r1u8pbx) **5**

[**1.0 INTRODUCTION**](#_31xl715q4yg7) **6**

[**2.0 METHODOLOGY AND RESULTS**](#_2ogpv41edulb) **7**

[2.1 Literature Review](#_izv5z0jacpg0) 7

[2.2 Proposed solution](#_3bul1jrlvxfl) 7

[2.3 User Role Modeling](#_u0no92w2rcdx) 8

[2.3.1 Brainstorm and Group](#_e4nfk61o61s) 8

[2.3.2 Consolidated user roles](#_w4hgvqexy3zn) 9

[2.3.3 Description of user roles and persona](#_z3qo7946mip2) 9

[2.3.4 Additional documentation](#_obcon58zr5lj) 10

[2.4 Release 1.0](#_fk2ypsmaw662) 11

[2.4.1 User stories](#_kivqu7ubwxwp) 11

[2.4.2 Additional documentation](#_v403sbskn5fq) 11

[2.5 Release plan 1.0](#_wmksujunf6w7) 11

[2.6 Iteration plan (up to Release 1.0)](#_6gbhuro3t1gx) 12

[2.6.1 Additional documentation](#_buo0mktrwmn9) 12

[2.7 Design documentation](#_wshdz52kjcx4) 12

[2.8 Progress monitoring](#_j8s8dg7qe5ax) 13

[2.9 Acceptance Tests for Release 1.0](#_op3tfrtua5i0) 14

[2.10 Release 2.0](#_ije0jl7y3wof) 16

[**3.0 CONCLUSIONS**](#_qgmam4m4tkry) **17**

[**4.0 RECOMMENDATIONS**](#_hsn9qjh7mzvx) **18**

[**CREDITS, LICENSE, AND REFERENCES**](#_8scyq3f3rbv8) **19**

[Credits](#_f0k94rrcszez) 19

[License](#_i54klyw4sgny) 19

[References](#_nvvc6a9ytxd4) 19

[**Appendix A (Manuals)**](#_ywl0m951u4x9) **20**

[**Appendix B (Test Plan)**](#_7d16if3e51ie) **21**

# List of Illustrations, Diagrams, and Tables

[Figure 1: Organizing the user role cards on a table [1]](#9tht8m96xzzo).

[Figure 2: The consolidated role cards [1].](#39pqc6401kkb)

[Table 1: Acceptance Tests for Release 1.0.](#2qi9vcw338tt)

# 1.0 INTRODUCTION

Introduction (Including the problem statement):

* What is the technical problem?

To create a more engaging environment for students to learn Health Informatics Principles; in conjunction with McMaster University our aim will be to produce a framework and working application for the gamification of the program. To produce an interesting presentation of the program, we would like to present a method of delivery that would improve the interest and interaction between the courses and students.

* Why was the work described in the TR undertaken?

The work undertaken is primarily to provide a more engaging method of instruction and delivery for the students of the Health Informatics Program at McMaster University. This is necessary for the University to consider other methods of learning and to maintain the attention of the students. This method will also allow the students to become much more involved in the learning process so that feedback may be given and the course improved.

* What is included and/or omitted? And what procedures are used?

The project was proposed to be completed using the MEAN Stack, primarily using Node JS framework and MongoDB as the backend database. We decided to not use SQL server because of its integration difficulty. The project should be able to assign work, to assign value to the work and to advance the students along the course.

* What unique problems were encountered in doing or interpreting the work?

As we are game programming students we do not have distinct knowledge or a though understanding of Health Informatics. This is remedied by asking questions and the open process of communication with McMaster University. The initial concept was to create a mind map stakeholder and badge schema which has since been reworked to a different method. The issues encountered allowed us to learn the process needed for a successful team and a well-rounded application.

* Are there unique approaches in the study?

Gamification was the major concept utilized for the project. The idea is to make learning about Health Informatics more fun and engaging for the students. The project was to design and implement a system that would satisfy the curiosity and function of the quantitative assessment for the students while also maintaining a slight competitive nature and a sense of personal achievement upon completion.

# 2.0 METHODOLOGY AND RESULTS

## 2.1 Literature Review

There has been an increased interest in the term "gamification" over the past several years. This recent term presents the opportunity to make certain aspects of a system fun and more interactive with its users. This would allow for more interaction between the teachers (Game Masters) and Students. With the teachers at the forefront of the courses, they would be capable of changing the framework with something that they believe would better suit their needs for their teaching methods.

## 2.2 Proposed solution

The project will utilize existing evaluations of the Health Informatics program to create a gamified system for the students. The framework consists of each aspect of the program being separated into specific categories known as Stakeholders. These Stakeholders as an area of study will have numerous topics attached to it known as Quests. The Quests will comprise of assignment material for the course.

Professors will act as Game Masters, who mark the assignments and award Badges based on the mark. Professors may also add additional incentives such as grades or perhaps a physical reward like gift cards.

As students complete Quests, Badges will be awarded. These Badges will be public and known by the class as a way of initiative and as a way for the students to be aware of their own progress. Completing a number of Badges that relates to a certain Stakeholder will earn Titles based on that Stakeholder, eventually earning enough Badges to achieve the Master Title. Special Titles will also be awarded upon completion if certain requirements are met. These are not required Badges and are similar to achievements.

The Badges themselves will have a ranking within them Gold, Silver, or Bronze which will correlate with the grade that the student achieved for that assignment. These designations assist the Game Master and allow the students an opportunity to understand the position they are in. These Badge ranks also allow for Special Titles such as All Gold Title.

Architecture (developer's view)

* HTML Website
* Pages for students to view their badges and potential badges
* Pages for the Professor to add badges, check student status, view assignment marks
* A page showing potential badges to earn and how to earn them
* Database of students
* Includes marks, badges, names, semester, courses

## 2.3 User Role Modeling

### 2.3.1 Brainstorm and Group

Show the results of your brainstorming session for identifying initial user roles and how they are organized (see [Figure 1](#9tht8m96xzzo)). Discuss each user role identified and the arrangement of [Figure 1](#9tht8m96xzzo).

Quests

Titles

Badges

Game Master

Player

|  |
| --- |
| Figure 1: Organizing the user role cards on a table [1]. |

## 

## 

### 2.3.2 Consolidated user roles

Show the consolidated user roles (see [Figure 2](#tfh2u3u06ecm)). Discuss the results of [Figure 2](#tfh2u3u06ecm), focusing on why some roles were merged, removed, and/or added.

|  |
| --- |
| Figure 2: The consolidated role cards [1]. |

### 2.3.3 Description of user roles and persona

For each consolidated role from the above [section](#j8k4v18q0uv8), include detail that answer at least the following questions:

* The frequency with which the user will use the software.
* The user's level of expertise with the domain.
* The user's general level of proficiency with computers and software.
* The user's level of proficiency with the software being developed.
* The user's general goal for using the software. Some users are after convenience, others favor a rich experience, and so on.

Include personas here (optional).

### 2.3.4 Additional documentation

For this section, include the video(s) from your workshop showing how your team:

1. Brainstormed for the initial set of user roles.
2. Organized the initial set of roles.
3. Consolidated and condensed the roles.
4. Generated detailed description of each consolidated role.

Provide the file name and URL to the video(s) in your shared folder.

Submit all index cards used in this session (use a rubber band). Clearly label the index card stack with your team name and an appropriate identifier (e.g. “User role identification”).

## 2.4 Release 1.0

### 2.4.1 User stories

The following are required for this section, excluding additional deliverables as outlined in sub-section [Additional documentation](#q7jvj24ek8zv) below:

1. Show and discuss the results of your low-fidelity prototype generated during your story writing workshop.
2. Provide your definition of story point (i.e. the number of ideal developer hour to a story point).
3. Show the stories created during the story-writing workshop. You can submit scanned images of your index cards (both front and back).
4. Prioritized stories based on the MoSCoW rule.[[1]](#footnote-1)

### 2.4.2 Additional documentation

For this section, include the video(s) from your workshop showing how your team:

1. Brainstormed for stories and generated the low-fidelity prototype (story writing workshop).
2. Estimated stories using the Wideband Delphi approach.
3. Prioritized stories using the MoSCoW rule.

Provide the file name and URL to the video(s) in your shared folder.

Submit all index cards used in this session (use a rubber band). Clearly label the index card stack with your team name and an appropriate identifier (e.g. “Pre-release 1.0 user stories”). Also, please ensure that the front, bottom-right side of each non-constraint user story index card is labeled with the team's estimate in *ideal developer hours.*

## 2.5 Release plan 1.0

The following are required for this section:[[2]](#footnote-2)

1. Provide the iteration length and the release date.
2. The refine priorities of the Must- and Should-Have stories by organizing the stories into groups that have a high likelihood of being performed together.
3. The actual release plan.

## 2.6 Iteration plan (up to Release 1.0)[[3]](#footnote-3)

The following are required for this section, excluding additional deliverables as outlined in sub-section [Additional documentation](#eijocz7ycp9n) below:

1. Present each iteration plan with tables showing disaggregated tasks per story and discuss any discrepancies between the estimated and actual ideal time required to complete the tasks (see Table 1 in document [Iteration Planning](https://drive.google.com/open?id=1x2SYm8QOJqti2qqwro8cF4cJQs3mS-CHmAAlihN9UAc)).

### 2.6.1 Additional documentation

For this section, include at least one video from your iteration planning meeting[[4]](#footnote-4):

1. Showing how your team disaggregated stories into their constituent tasks.
2. How developers on your team volunteer and take responsibilities for tasks.

Provide the file name and URL to the video(s) in your shared folder.

## 2.7 Design documentation

Traditional approaches to software development, in contrast to that of Agile approaches, place a great deal of emphasis on upfront design. The Agile approach to design are quick sessions that seek the simplest solution and then incrementally build on that solution. A quick design session can include the use of CRC cards that can ultimately lead to the generation of UML diagrams.

Using Agile approaches to software development does not mean you are limited to using only Agile techniques. If you feel that a technique (e.g., use case or interaction design scenario) is more suitable, or better conveys the features of your system to your users, then use it.

In this section, you are required to submit and discuss the following:

* A paper prototype of your application/system.
* Any design work your team has done in developing your system including CRC cards, UML diagrams, ERD diagrams, use cases, interaction design scenario, etc.

## 2.8 Progress monitoring

The following are required for this section:[[5]](#footnote-5)

A table summarizing progress and changes during a release with supporting discussion (see Table 1 in document [Measuring and Monitoring Progress](https://drive.google.com/open?id=1c-cIhvv_pYoAV3sADxto8fO9ljDvU_pZXFf6cp2lcl8)).

## 2.9 Acceptance Tests for Release 1.0

The following are required for this section:

1. The [table of Acceptance Tests](#2qi9vcw338tt) for this Release as shown below and the link to your video demo.

Video Demo Link: https://drive.google.com/file/d/0B2LdTKFM\_KMWZkRUblhMX0JoQk0/view

|  |  |  |
| --- | --- | --- |
| User Story | Acceptance Test | Task |
| Story 1:  User can register on the  website | Test with new login  Test with a login already used  Test with weak password  Test with strong password | Registration web-form  Registration routes (get/post) in controller  User table in database |
| Story 2:  User can login on the website | Test with correct login and password  Test with incorrect login  Test with incorrect password | Log in web-form  Login routes (get/post) in controller  User table in database |
| Story 3:  User can reset password | Test with correct email  Test with non-existing email | Email reset web-form  Email reset routes (get/post) in controller  User table in database |
| Story 4:  Admin can create new quest (Assignment) | Test creation of new quest  Test creating of the quest already exists | Quest creation web-form  Quest creation routes in controller  Quest table in database |
| Story 5:  Admin can create new badge | Test creation of new badge  Test creating of the badge already exists | Badge creation web-form  Badge creation routes in controller  Badge table in database |
| Story 6:  Admin can create new stakeholder | Test creation of new stakeholder  Test creation of the stakeholder already exists | Stakeholder creation web-form  Stakeholder creation routes in controller  Stakeholder table in database |
| Story 7:  Player can take a quest | Test successful completion of quest  Test failure of the quest | Quest web-forms  Quest routes in controller  Quest table in database |
| Story 8:  Admin can accept results of the quest and badge assigning | Test acceptance process for successful quest | Quest acceptance web-form  Quest acceptance routes in controller  Quest detail table in database |
| Story 9:  Admin can accept results of the quest and stakeholder assigning | Test acceptance process for successful quest | Quest acceptance web-form  Quest acceptance routes in controller  Quest detail table in database |
| Story 10:  Admin assigns quest to stakeholder for multi topic quests | Test acceptance of quest assignment to stakeholder | Quest allocation web form, Quest allocation routes in controller |

## 2.10 Release 2.0

Release 2.0 has essentially the same structure as Release 1.0 except for a minor difference. If your team wrote enough stories to cover up to or beyond Release 2.0 during your first story-writing workshop as described in the [User stories](#a12dw6d04xyp) section, then your team will not need to hold a second workshop.

Another difference is that the release date for Release 2.0 would have a fixed deadline; that is, the week before the final exam week. Every other deliverable as described in Release 1.0 applies to this release and any subsequent releases.

# 3.0 CONCLUSIONS

A conclusion interprets the data found in the Body. It is reasoned judgment and not opinions. Consider the variables. Relate cause and effect. Analyze, evaluate, make comparisons and contrasts. Base the conclusion on fact.

# 4.0 RECOMMENDATIONS

Recommendations are not required for all studies. They suggest a course of action and would generally be provided when there are additional areas for study, or if the reason for the TR was to determine the best action going forward.

# CREDITS, LICENSE, AND REFERENCES

## Credits

Provide any credits here. The following are examples:

Author of the template graphic layout : Hao Lac <haolac.at.centennial@gmail.com>

Author of the template explanation text : John Doe <[john.doe@centennialcollege.ca](mailto:john.doe@centennialcollege.ca)>

## License

State the license granted with your system. For example:

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License".

## References

[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

# Appendix A (Manuals)

In this section, include a user manual for your system/application. The user manual should include the following items:

1. Instructions on how to install and configure your system/application, documenting all external software dependencies that need to be setup manually.
2. A user guide for the administrator (use screen shots of your system/application and briefly discuss each screen shot).
3. A user guide for the normal user (use screen shots of your system/application and briefly discuss each screen shot).

# Appendix B (Test Plan)

In this section, include a test plan that discusses how you will test your system (unit testing, integration testing, and acceptance testing). Focus on automating your test as much as possible.

1. See the [User Stories](https://drive.google.com/open?id=1caa0hWkLmSex3YytaVzHIPz9Lup9mQCWaS_339IImNE) deliverable. [↑](#footnote-ref-1)
2. See the [Release Plan](https://drive.google.com/open?id=1L8lKu8QLqNfpaAntcHeWtLMxik9OrmBszS9Nrg0XEUI) deliverable. [↑](#footnote-ref-2)
3. *x* represents integer values. Teams typically have a total of four iteration plan (i.e. Iteration plan 1, Iteration plan 2, and so on). [↑](#footnote-ref-3)
4. Indicate which iteration the video corresponds with. Note: your team is required to submit one video for the iteration planning meeting for the entire project. If you decide to submit a video in Release 1.0, then you do not need to include an [Additional documentation](#eijocz7ycp9n) section for Release 2.0. [↑](#footnote-ref-4)
5. See the [Measuring and Monitoring Progress](https://drive.google.com/open?id=1c-cIhvv_pYoAV3sADxto8fO9ljDvU_pZXFf6cp2lcl8) deliverable. [↑](#footnote-ref-5)