e-Valuate

Iteration Plan 1

Version 1.3.0

Revision History

Date	Version	Description	Author
29/Oct/19	0.0.1	Completed sections 4 and 5 Began section 1	Kenny Houston
30/Oct/19	0.0.2	Complete section 1, 2, and 3	Kenny Houston
01/Nov/19	0.0.3	Made edits to grammar	Jordyn Dent
04/Nov/19	0.1.1	Added database ER diagram	Kenny Houston
05/Nov/19	0.1.2	Correct links to Use Cases Add tasks to UC 04 implementation	Kenny Houston
05/Nov/19	0.1.3	Added tasks to UC 07 in section 4.3	Jordyn Dent, Megan Phan
05/Nov/19	0.1.4	Added tasks to UC 13 in section 4.2	Action Ghimire, Hau Ha
08/Nov/19	0.1.5	Added Use Case Diagram	Kenny Houston
09/Nov/19	1.0.0	Remove database diagram	Kenny Houston
10/Nov/19	1.0.1	Add "Actual" dates to tables Change date formatting	Kenny Houston
11/Nov/19	1.0.2	Add Class Diagram	Kenny Houston
17/Nov/19	1.0.3	Revise UC04 tasks	Kenny Houston
20/Dec/19	1.1.0	Update UC04 actual dates Add "Restructure front end" tasks to UC04	Kenny Houston
29/Dec/19	1.1.1	Update UC07 actual dates	Megan Phan
05/Jan/20	1.2.0	Update UC13 actual dates	Hau Ha
06/May/20	1.3.0	Final touch ups and corrections	Kenny Houston

e-Valuate	Version: 1.3.0
Iteration Plan 1	Date: 06/May/20

Table of Contents

Iteration Plan 1

1. Introduction

1.1 Purpose

In the first iteration, we want the most basic and most necessary features to be implemented. Since e-Valuate is an interactive rubric platform, we will only be implementing the features which directly cater to this goal.

1.2 Scope

All layers of the stack will be touched in this iteration. There will be changes to the front end and back end, to accommodate the implementation of the first set of use cases.

1.3 Definitions, Acronyms, and Abbreviations

All terms can be found in the glossary.

1.4 References

1.4.1 Glossary

1.4.2 UC04 Create a Rubric for an Event

1.4.3 UC07 Analyze a Score Report

1.4.4 UC13 Grade an Entry

1.5 Overview

This iteration is planned to build the most fundamental pieces of the system while touching all layers. This first set of features will focus on rubrics: Their creation by the chair and their use by the judges. Then there will be a minimal version of a feature that will be implemented in full during a later iteration. This minimal feature will just be used to verify that judges' scores are saved into the system correctly.

2. Plan

2.1 04/Nov/19 - Begin

By this time, the details of the use cases will be wholly solidified, so we may begin work on the implementation.

2.2 08/Nov/19 - Design Complete

Shortly after starting, we will have solidified database requirements and general architecture in visual models.

2.3 22/Nov/19 - Beta and client testing

Two weeks after having the design laid out, we will have rough versions of our features, to begin testing both internally and in front of the client. This client check-in is necessary to assure we do not stray too far away from their vision.

2.4 09/Dec/19 - Complete

The iteration will be complete by 09/Dec/19. The implementation will be fully tested both internally and with the client.

3. Resources

3.1 Cost

This first iteration should be of fairly low financial cost but may have the greatest time cost. Since all layers of the system will be constructed for the first time in this iteration, it will likely take longer to implement.

3.2 Tools

3.2.1 Spring boot

3.2.2 Maven

3.2.3 MySQL

3.2.4 Hibernate

3.2.5 ReactJS

4. Use Cases

4.1 UC04 Create a Rubric for an Event

This iteration will not have accounts or events created by any particular account. Any and all events at this time will be hardcoded into the system's database, simply to have this use case built on top of them. All these events and the rubrics associated with them will be accessible by anyone who has access to the system as a whole. We may implement a set of codes to access the front end of the system, but the complexity of this will likely be quite limited.

Jackie and Kenny are assigned to this use case implementation.

Task	Assignee	Expected Start	Expected End	Actual Start	Actual End
Build and test Score and Entry class in backend and test	Kenny	12/Nov/19	15/Nov/19	11/Nov/19	25/Nov/19
Build and test Category class in backend and test	Jackie	12/Nov/19	15/Nov/19	11/Nov/19	25/Nov/19
Build Rubric Creation Controller and Service classes	Kenny	16/Nov/19	22/Nov/19	26/Nov/19	02/Dec/19
Build front end Rubric Creation page	Jackie	16/Nov/19	22/Nov/19	26/Nov/19	02/Dec/19
Restructure front end	Kenny	N/A	N/A	20/Dec/19	05/Jan/20
Testing period	All members	22/Nov/19	04/Dec/19	05/Jan/20	16/Jan/20

4.2 UC13 Grade an Entry

The implementation of this use case is heavily tied to the purpose of the program as a whole. As such, it is being implemented within the first iteration. The registered entries, much like the events, will be hard-coded into the system at this time. Entries will not be assigned to any particular judge, because there are no judge accounts and group assignments will be handled later.

Hau and Action are assigned to this use case implementation.

Task	Assignee	Expected Start	Expected End	Actual Start	Actual End
Design/draw the page	Action & Hau	08/Nov/19	08/Nov/19	08/Nov/19	08/Nov/19

Build frontend Grade an entry page	Action	08/Nov/19	13/Nov/19	08/Nov/19	15/Nov/19
Build GradeAnEntry Controller and Services classes	Hau	10/Nov/19	15/Nov/19	26/Nov/19	03/Dec/19
Implement frontend Grade an Entry use case	Hau & Action	18/Nov/19	21/Nov/19	26/Nov/19	05/Dec/19
Connect backend to frontend	Hau	18/Nov/19	21/Nov/19	30/Nov/19	05/Dec/19
Testing Period	All Member	28/Nov/19	04/Dec/19	05/Dev/19	16/Jan/20

4.3   UC07 Analyze a Score Report (alpha version)

The implementation of this feature will be severely cut down compared to the full use case. This use case only exists in this iteration because we need a means of verifying that the entries graded in UC13, are saved successfully by the system.

Jordyn and Megan are assigned to this use case implementation.

Task	Assignee	Expected Start	Expected End	Actual Start	Actual End
Draw web page wireframe	Jordyn & Megan	06/Nov/19	06/Nov/19	06/Nov/19	06/Nov/19
Build and test the ScoreReportRepository and ScoreReportService	Megan	14/Nov/19	20/Nov/19	20/Nov/19	30/Nov/19
Build and test the ScoreReportController	Megan	14/Nov/19	20/Nov/19	22/Nov/19	02/Dec/19
Implement the frontend of the ScoreReport Page	Megan	22/Nov/19	2/Dec/19	25/Nov/19	05/Dec/19
Connect backend to the frontend	Megan	22/Nov/19	02/Dec/19	25/Nov/19	05/Dec/19
Testing period	All members	28/Nov/19	04/Dec/19	05/Dec/19	17/Jan/20

5.   Evaluation Criteria

5.1   Responsive

The system needs to react to the user's inputs quickly, as to aid them in their grading efforts, rather than hinder them. A slow electronic system can quickly become more frustrating than doing everything by hand on physical paper.

5.2   Ease of Use

The users need to be able to navigate the system without the guidance of a third party. We will present the web app to the client, and provide no other guidance other than what the built-in system provides. If they can accomplish all the desired tasks without becoming frustrated, then the system is easy enough to use.
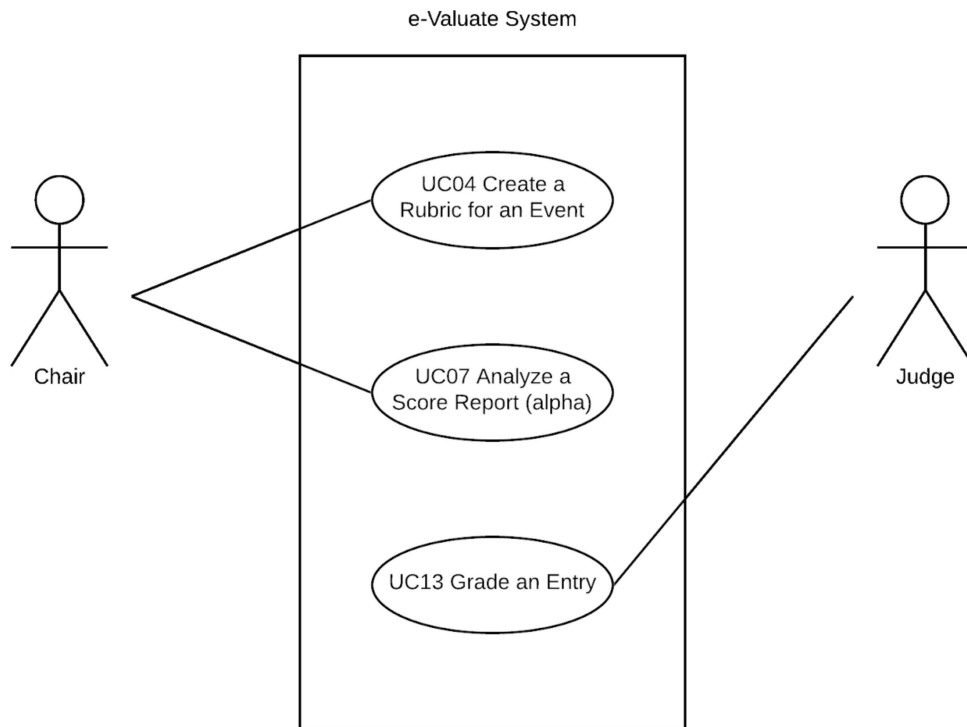
5.3 Reliability

The user must be able to have full confidence that the system will save rubrics and grades reliably. Or at the very least, convey clearly to the user when information has failed to save properly.

6. Diagrams

6.1 Use Case Diagram

e-Valuate System

UC04 Create a Rubric for an Event

UC07 Analyze a Score Report (alpha)

UC13 Grade an Entry

Chair

Judge

6.2   Class Diagram