e-Valuate

Iteration Plan 3

Version 1.0.0

Revision History

Date	Version	Description	Author
29/Oct/19	0.1.1	Initial Draft	Jackie Ye
16/Jan/20	0.2.1	Move UC08 from Iteration 4 to 3	Kenny Houston
08/Feb/20	0.3.1	Write Sections 1 through 3	Kenny Houston
09/Feb/20	0.3.2	Write Sections 4 and 5	Kenny Houston
10/Feb/20	0.4.1	Assign use cases Added site map	Kenny Houston
19/Feb/20	0.5.1	Update class diagram	Kenny Houston
06/May/20	1.0.0	Final touch ups and corrections	Kenny Houston

Table of Contents

Iteration Plan 3

1. Introduction

1.1 Purpose

This purpose of Iteration 3 is to add the final use cases to e-Valuate which tie everything together. In addition to implementing 4 new use cases, Iteration 3 will weave the previous use case web pages together to build a fluid experience. And finally, we will be adding and modifying smaller, but still important bits of contents to make e-Valuate whole.

1.2 Scope

This iteration will include 4 use cases in addition to the continued work of testing modifying the system. We will be modifying the chair/judge user experience by adding new functionality which allows more refined management of an event. We will also be expanding upon the functionality of the Score Report, making it interactive and more comprehensive.

1.3 Definitions, Acronyms, and Abbreviations

All terms can be found in the glossary.

1.4 References

1.4.1 Glossary

1.4.2 UC05 Invite Judges to an Event

1.4.3 UC07 Analyze a Score Report

1.4.4 UC08 Assign Judges to Group

1.4.5 UC11 Use an Event Invitation Link

1.4.6 UC12 Access a List of Assigned Entries

1.5 Overview

The use cases of Iteration 3 focus on adding judge management functionality in a way that can be tailored by the chair. The chair may invite judges to their event, and then organize them into grading groups. Also, the chair will have access to a Score Report which will comprehensively show all the judges grades for each entry. Lastly, the website will become a more fluid and unified experience as previous use cases are brought together and improvents become ongoing.

2. Plan

2.1 07/Feb/20 - Design

The team will be completing finalizing requirements for this iteration in this phase. The use case diagram and class diagram will be updated to account for the new requirements. They are obviously flexible as more feedback is received from the client. However, this is our starting point.

2.2 10/Feb/20 - Development

Any development on new use cases or modification to old use cases is done in this phase. All front end and back end modifications will also be completed here.

2.3 24/Feb/20 - Testing

With functional development completed, further refinement is possible in this phase. Ideally, the developer believes their use cases to be perfect. Other developers will help to verify that it is, or suggest final touches which improve upon the quality of the use case.

2.4   02/Mar/20 - Completion

The use cases of this iteration will be completed by the developers and approved by the client by 02/Mar/20. However, all other types of changes will be ongoing.

3.   Resources

3.1   New Tools

3.1.1   Shiro

Implementing Shiro as our backend security framework in Iteration 2 was not possible given the time constraints. Now that fundamentals of Shiro are understood and the account creation and login front end pages are otherwise complete, implementing Shiro will be possible.

3.1.2   AWS

AWS (Amazon Web Services) is currently our choice for deployment. With Iteration 2 bringing much of the code together, we will begin deploying the project in order to become acquainted with the processes and hassle of deployment.

3.2   Cost

3.2.1   Shiro

Shiro is a free security software, so it will have no monetary cost. Just the time costs for learning and implementation.

3.2.2   AWS

AWS only charges for run time at a rate of $0.34/hr. We intend to use it only for testing during this iteration, so hopefully, that will keep costs down. 4hrs/day for a month is approximately $40.

There are also free and academic tiers we can take advantage of, so paying out of pocket is only in the worst case.

4.   Use Cases

4.1   UC05 Invite Judges to an Event

Some discussion around this use case is still necessary. Do we want this to be an invitation link, or a one time use code? I believe one time use codes to be a simpler implementation. However, I can see the utility in an invitation link bringing the user to the app. TODO discuss implementation.

This use case has been assigned to Jackie.

4.2   UC07 Analyze a Score Report

This will be an extension of the alpha implementation from Iteration 1. The alpha version was merely a read-only implementation. Meanwhile, this implementation will be a much more in depth interactive score report with read and write abilities.

This use case has been assigned to Megan.

4.3   UC08 Assign Judges to Groups

This use case will likely require a front end heavy implementation to make for an experience which is satisfactory to the user. The Chair must be able to specify multiple groups and move judges between different groups easily, while being able to change their mind on the fly.

This use case will involve some modifications to UC12, which is currently splitting Judges into 2 groups automatically.

This use case has been assigned to Hau.

4.4   UC11 Use an Event Invitation Link

Some decisions regarding the design of this use case still need to be made (See UC05 above) and the use case's documentation will need to be ironed out accordingly.

This use case has been assigned to Action

5.   Evaluation Criteria

5.1   Responsive

The system needs to react to the user's inputs quickly. If a user were to become frustrated with slow response times, they would be less likely to continue using the system in the future. 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 will be deemed easy enough to use.
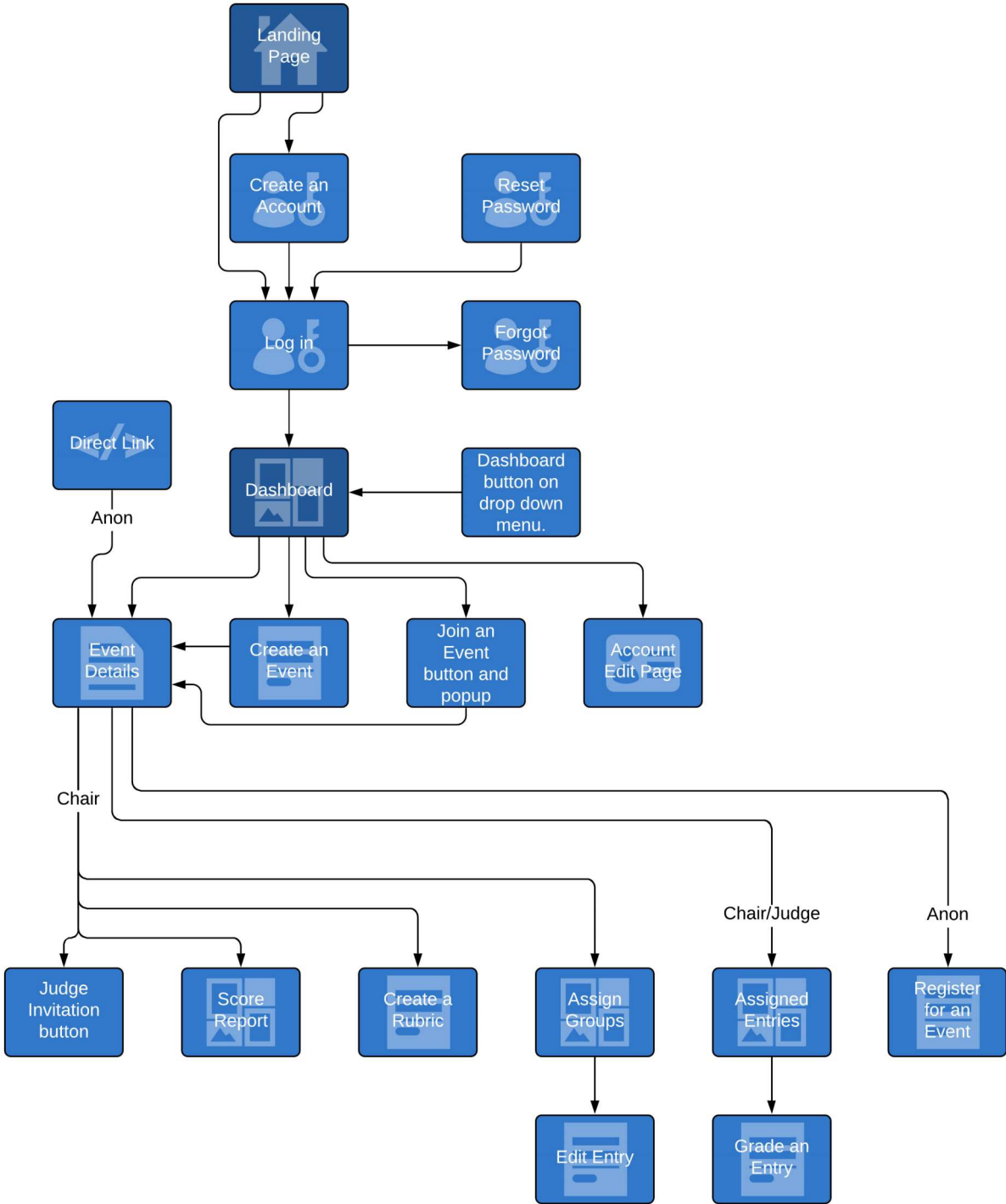
5.3   Reliability

The user must be able to have full confidence that the system will save all the required information without fail. Or at the very least, convey clearly to the user when information has failed to save properly.

6.   Diagrams

6.1   Site Map

6.2   Use Case Diagram
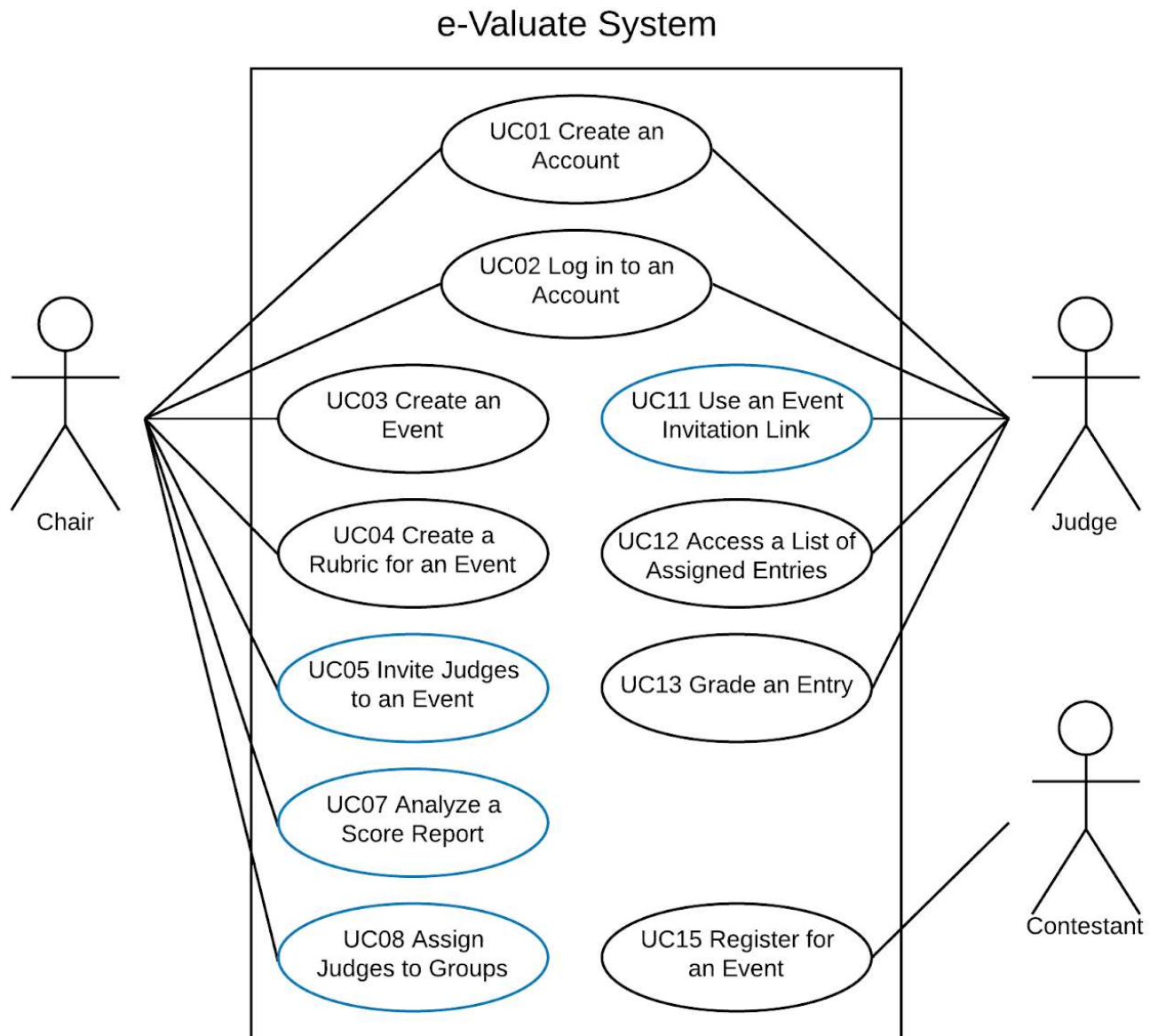
6.3 Class Diagram
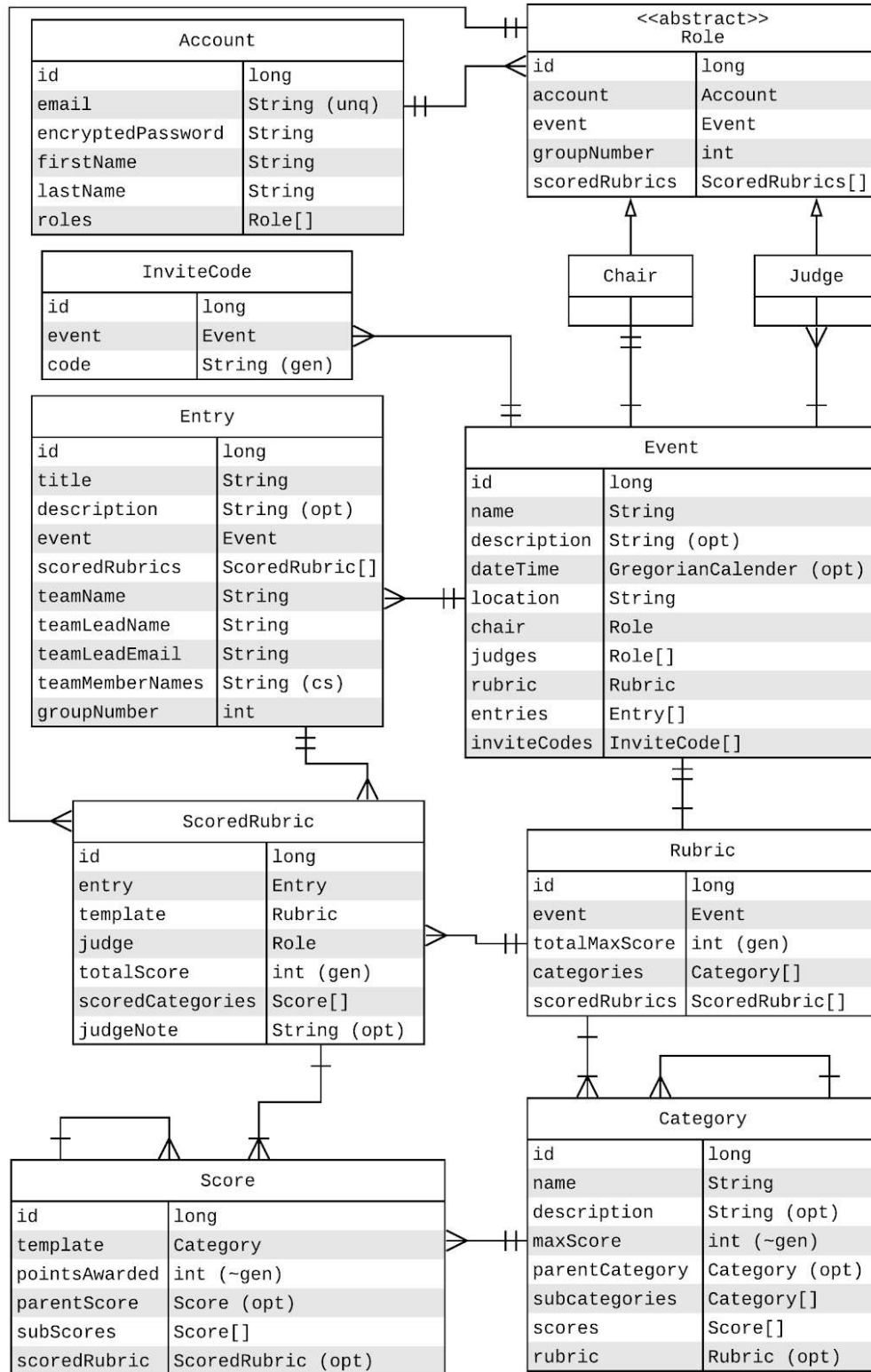
Key

cs	Comma Separated
gen	Generated
~gen	Possibly Generated
opt	Optional
unq	Unique

Account

id	long
email	String (unq)
encryptedPassword	String
firstName	String
lastName	String
roles	Role[]

<> Role

id	long
account	Account
event	Event
groupNumber	int
scoredRubrics	ScoredRubrics[]

Chair

Judge

InviteCode

id	long
event	Event
code	String (gen)

Entry

id	long
title	String
description	String (opt)
event	Event
scoredRubrics	ScoredRubric[]
teamName	String
teamLeadName	String
teamLeadEmail	String
teamMemberNames	String (cs)
groupNumber	int

Event

id	long
name	String
description	String (opt)
dateTime	GregorianCalender (opt)
location	String
chair	Role
judges	Role[]
rubric	Rubric
entries	Entry[]
inviteCodes	InviteCode[]

ScoredRubric

id	long
entry	Entry
template	Rubric
judge	Role
totalScore	int (gen)
scoredCategories	Score[]
judgeNote	String (opt)

Rubric

id	long
event	Event
totalMaxScore	int (gen)
categories	Category[]
scoredRubrics	ScoredRubric[]

Score

id	long
template	Category
pointsAwarded	int (~gen)
parentScore	Score (opt)
subScores	Score[]
scoredRubric	ScoredRubric (opt)

Category

id	long
name	String
description	String (opt)
maxScore	int (~gen)
parentCategory	Category (opt)
subcategories	Category[]
scores	Score[]
rubric	Rubric (opt)