

Florida International University

School of Computing and Information Sciences

Final Deliverable

GraderFinder

Team Members: Leila Adaza, William Padron, David Paez, Cheska Sy, Enrique Rodicio, Justin Cabrera, Nicholas Delamo, Damian Torrens, Phillip Nguyen

Product Owner: Deng Pan

Instructor: Masoud Sadjadi

Table of Contents

1. Introduction	4
2. User Stories	5
2.1 Implemented User Stories	5
2.2 Pending User Stories	6
3. Project Planning	6
3.1 Hardware and Software Resources	6
3.2 Sprints Plan	8
3.2.1 Sprint 1 Planning	8
3.2.2 Sprint 2 Planning	8
3.2.3 Sprint 3 Planning	9
3.2.4 Sprint 4 Planning	10
3.2.5 Sprint 5 Planning	10
3.2.6 Sprint 6 Planning	11
3.2.7 Sprint 7 Planning	11
4. System Design	12
4.1 Architectural Patterns	12
4.2 System and Subsystem Decomposition	12
4.3 Deployment Diagram	12
4.4 Design Patterns	12
5. User Interface and Diagrams	13
5.1 User Interface Prototype Design	13
5.2 User Interface Final Design	14
5.3 Use Case Diagram	15
5.4 Class Diagram	16
5.5 Sequence Diagrams	17
4.5.1 Grader View	17
5.5.2 Professor View	18
6. Appendix	19
6.1 Appendix A - Sprint Review	19
6.1.1 Sprint 1 Review	19
6.1.2 Sprint 2 Review	20
6.1.3 Sprint 3 Review	21
6.1.4 Sprint 4 Review	21

6.1.5 Sprint 5 Review	22
6.1.6 Sprint 6 Review	23
6.2 Appendix B - User Manual and Installation	23
6.2.1 User Manual	23
7. Closing Remarks	24

1. Introduction

GraderFinder is a Learning Tools Interoperability (LTI) application integrated into Canvas with the purpose of allowing professors to easily find graders for courses using the automated system. In addition, it provides students a streamlined platform to sign up to be a grader for assignments that professors post on the GraderFinder app, removing the need for students who meet the relevant grader prerequisites to communicate with a middleman in charge of manually sending out mass emails. The application is built off of the open-source Canvas LMS and uses technologies such as Ruby for the frontend and backend as well as PostgreSQL to store registered grader and course information.

2. User Stories

2.1 Implemented User Stories

1. As a developer, I want to set up a GitHub repository to be able to collaborate with other code-base contributors.
2. As a developer, I want to read the LTI specifications to understand how to use it in the code of our project.
3. As a developer, I want to learn how to use the Ruby programming language to code the frontend and backend of our project.
4. As a developer, I want to implement functionality to allow new users to use GraderFinder.
5. As a user who is a student, I want to have a way to input my GPA, prerequisite courses, and grades for those courses to become a grader.
6. As a developer, I want to create a use case diagram to understand and document the overall design of the system such as the actors and functions.
7. As a developer, I want to create sequence diagrams to document how the functionality of the system works.
8. Create methods to select assignments the grader is eligible for. As a developer, I want to create methods to select assignments a grader is eligible for to automate the process for professors in need of graders.
9. As a developer, I want to refine our use case diagram to document the edge cases that were not included in the initial design.

10. As a user who is a grader, I want to simply click a button to easily apply to grade an assignment.
11. As a developer, I want to create a class diagram to understand and document how the separate classes and functions of the system interact with each other.
12. As a user who is an instructor, I want to select what grader I would like to grade my course's assignments to choose who I prefer instead of leaving it up to the automated system.
13. As a developer, I want to refine the class diagram's fields with fields from the Canvas database table to achieve a clearer and more accurate depiction of how the system works.
14. As a teacher or grader, I would like GraderFinder's UI to look clean and organized

2.2 Pending User Stories

15. As a developer, I want to follow the LTI app demo tutorial to gain a better understanding of how to make an LTI app and integrate into GraderFinder.
16. As a user who is a professor, I would like to integrate GraderFinder as an LTI application on my school's Canvas LMS.
17. As a user who is a professor, I would like to have GraderFinder return grades to Canvas LMS automatically, together with submission comments.

3. Project Planning

3.1 Hardware and Software Resources

The following are software resources we used to design this project:

- **Microsoft Teams** - We used Microsoft Teams to organize meetings and communicate any schedule changes. Most of our communication occurred in Zoom meetings where we discussed any issues and potential hurdles. Microsoft Teams also allowed us to communicate with the product owner when need be.
- **Zoom** - A video conferencing platform used to host our daily scrum, sprint planning, sprint review, and backlog grooming meetings. Both teams would discuss current progress, issues, and hurdles on a daily basis. The product owner would join for sprint review, sprint planning, and backlog meetings and assist in generating user stories and guiding students.
- **Google Drive** - Platform used to store any documentation and enable team members to collaborate on live reports. All our diagrams, meeting minutes, and wireframes were placed here for all team members to access.
- **Canvas** - A learning management system used by professors to upload assignments, quizzes, and exams for students to take. Learning Tools Interoperability (LTI) applications like the ones we are developing can be used to add bonus functionality to a professor's course.
- **Ruby on Rails** - A server-side web application framework written in Ruby. Ruby on Rails provides default structures for databases, web services, and web pages. Canvas is built using Ruby on Rails.

- **PostgreSQL** - A free and open-source relational database management system emphasizing extensibility and SQL compliance. Our custom GraderFinder database uses PostgreSQL and holds information on graders, courses, and assignments.

3.2 Sprints Plan

3.2.1 Sprint 1 Planning

Attendees: Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo, William Padron, David Paez, Leila Adaza, Cheska Sy

Product Owner: Deng Pan

Date: 8/31/20

Time: 5:30 PM - 6:00 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Set up a GitHub repository
2. Read about LTI specification and find out how it works
3. Learn how to use Ruby programming language
4. Determine how to notify graders/do the matching (for example, Uber)
 - a. Notifications for assignments available, instructor choosing them, and reviewed
5. Create basic documentation
 - a. Overview, Purpose, Problem it tries to solve, refer to documentation used

3.2.2 Sprint 2 Planning

Attendees: Cheska Sy, William Padron, David Paez, Leila Adaza, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 9/14/20

Time: 7:45 PM - 8:11 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Generate a dummy database of students, ta accounts, and teachers that can be retrieved by the canvas system
2. Create methods to assign a TA to a teacher's assignment within their class
3. Create a page to display the methods mentioned in User Story 2 on the front end for users
4. Create page that takes new grader's GPA, prerequisite courses, and grades for those courses
5. Create documentation for use case, class, and sequence diagrams

3.2.3 Sprint 3 Planning

Attendees: David Paez, Leila Adaza, Cheska Sy, William Padron, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 9/28/20

Time: 8:10 PM - 8:25 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Return grades with comments to instructor in Canvas
2. Create methods to select assignments grader is eligible for
3. Continue working on Use Case diagrams to make more comprehensive and cover edge cases; think about all possible scenarios

3.2.4 Sprint 4 Planning

Attendees: Leila Adaza, Cheska Sy, William Padron, David Paez, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 10/12/20

Time: 7:46 PM - 7:53 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Implement way for graders to apply to grade an assignment
2. Implement Master courses rather than generic courses for teachers
3. Create additional documentation for GraderFinder workflow
 - a. Alternative use case diagram for scenario in which app is fully integrated in Canvas
 - b. Sequence diagram for professor's view
 - c. Class diagram
4. Return grades with comments to instructor in Canvas

3.2.5 Sprint 5 Planning

Attendees: William Padron, David Paez, Leila Adaza, Cheska Sy, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 10/26/20

Time: 8:02 PM - 8:15 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Allow instructor to select grader instead of first come first serve
2. LTI App demo by next Monday (Backlog grooming Meeting)
3. Improve class diagram based on PO's previous feedback
 - a. Replace current fields with fields from SQL Canvas table
 - b. Create alternative, streamlined use case diagram for LTI implementation

3.2.6 Sprint 6 Planning

Attendees: David Paez, Leila Adaza, Cheska Sy, William Padron, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 11/10/20

Time: 7:36 PM - 7:46 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Begin preparing Final Documentation for Senior Showcase
2. LTI App demo
3. Final GraderFinder LTI app

3.2.7 Sprint 7 Planning

Attendees: David Paez, Leila Adaza, Cheska Sy, William Padron, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 11/24/20

Time: 7:30 PM - 7:52 PM

Estimated Team Velocity: 24 points

Stories Pulled Into Sprint

1. Finalize Final Documentation for Senior Showcase
 - a. Write information for user manual
 - b. Fill out content for installation
 - c. Finish sprint reviews
 - d. Finalize user design interface by compiling the screenshots of the product
 - e. Reformat user stories to follow “As a user/developer, I want to do X to achieve Y.”
2. Explain additional tables and describe the Canvas system architecture for System Design documentation section
3. Work on individual posters
4. Finish LTI app demo along with documentation
5. Work on UI of GraderFinder

3.2.8 Sprint 8 Planning

Attendees: Nicholas Delamo, Enrique Rodicio, Phillip Nguyen, Damian Torrens, William Padron, David Paez, Leila Adaza, Cheska Sy

Product Owner: Deng Pan

Date: 12/7/20

Time: 7:30 PM - 7:35 PM

Estimated Team Velocity: 8 points

Stories Pulled Into Sprint

1. Finish compiling files for Final Deliverable

4. System Design

4.1 Architectural Patterns

MVC (Model, View, Controller):

The GraderFinder system follows the MVC architecture, where the model represents an object in the database, and is what is in charge of communicating with the database. Controller is in charge of providing information to the view, and also with receiving user input and making changes to the database. Views are the front-end which users see and interact with.

4.2 System and Subsystem Decomposition

GraderFinder System:

The GraderFinder system is composed of two major subsystems: the Grader Subsystem and the Teacher Subsystem.

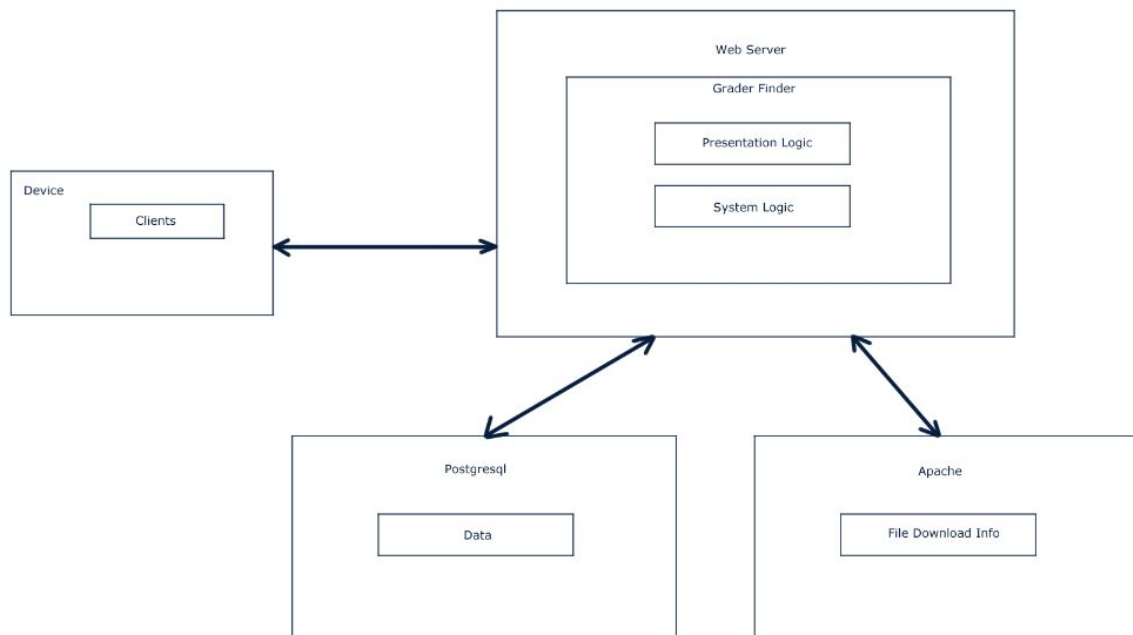
Grader Subsystem:

The Grader Subsystem is responsible for any functions and features concerning grader users. This includes registering new grader users to the GraderFinder database with GPA and prerequisites included, applying for assignments the Grader is eligible to grade once their application has been accepted by an instructor, downloading and scoring assignments while grading, and using SpeedGrader to grade assignments. The Grader Subsystem relies on the Teacher Subsystem for created assignments by teachers to grade, grader applications for assignments being accepted, assignment scores and submissions, and prerequisites being used to determine the eligibility of a grader.

Teacher Subsystem:

The Teacher Subsystem is responsible for any functions and features concerning teacher users. This includes registering new teacher users to the GraderFinder database, allowing the teacher users to create a new course and enter prerequisites for that course, creating and publishing assignments for that course, and viewing grader applications for assignments in the course. In addition, teachers can assign graders to different sections of a course for an assignment, submit assignments from canvas into GraderFinder for graders to apply for, and view assignments a grader has finished grading. Also, teachers will be able to integrate GraderFinder into a LMS via LTI. The Teacher Subsystem is reliant on the Grader Subsystem for grader applications for an assignment, assigning graders to different sections of a course, receiving scores for a graded assignment from a grader, and course prerequisites being used to determine the eligibility of a grader.

4.3 Deployment Diagram



4.4 Design Patterns

The design patterns used throughout the project are the main ones used by Rails framework originally. This can be seen in the Template design pattern, which creates a skeleton to be used during the independent pages that will display different information.

Also, the MVC (Model-View-Controller) as part of the Rails framework, is the pattern that most impacted our project as it let us create and control the user interface. This helped to separate and work independently in the different parts of the project and the way information is presented, processed, and saved. The files in the view directory are the ones that present the information to the user and let them interact with the application. The controller directory has the functions that accept the data from the user and process it to the models or the views, depending on the use case. And the model directory contains the database where all the information about the graders is stored.

Another design pattern used in the project is the Facade. This pattern allows the user to have a simple User Interface to interact with, like forms, leaving the deep functionality and operations behind the scenes. All the forms through the application like login, registration, and creation of courses are presented as a simple interface with a complex subsystem behind.

The Observer design pattern is also present in this application, and it can be seen when the graders are notified about new assignments available to be graded. Also, from the professor's side it is present when the professor can see a list of available graders that submitted a request to grade an assignment.

5. User Interface and Diagrams

5.1 User Interface Prototype Design


Home


Account


Dashboard


Courses


Calendar


Inbox

ACH4302 > Modules

Home

Discussions

Grades

People

Syllabus

GraderFinder

GraderFinder

GraderFinder Registration Form

* Required

Email address *

Your email

Full Name *

Your answer

Panther ID *

Your answer

What courses have you taken where you've received a B or higher? *

Choose

GRA *

Choose

Submit


Home


Account


Dashboard


Courses


Calendar


Inbox

ACH4302 > Modules

Home

Discussions

Grades

People

Syllabus

GraderFinder

GraderFinder

Available Assignments	Add Course
1. COP2210 HW Assignment 1	Apply
2. COT3010 Ch 03 Quiz	Apply
3. COP3337 Lab 3	Apply
4. COP4338 Programming Assignment 2	Apply
5. CDA3103 Exam 1	Apply
6. CDA4101 In-class Lab Questions 2	Apply
7. CEN4010 Web Project	Apply
8. COP3530 Data Structures Quiz 2	Apply
9. COT3541 Logic Assessment 1	Apply

ACH4302 > Modules

Home

Discussions

Grades

People

Syllabus

GraderFinder

Available Assignments

1. COP2210 HW Assignment 1

2. COT3010 Ch 03 Quiz

3. COP3337 Lab 3

4. COP4338 Programming Assignm

5. CDA3103 Exam 1

6. CDA4101 In-class Lab Questions

7. CEN4010 Web Project

8. COP3530 Data Structures Quiz 2

9. COT3541 Logic Assessment 1

CIS 4951: Capstone I

Masoud Sadjadi

Instructor Information

Email: sadjadi@fsu.edu

Phone: 305-710-6598

Office: ECS 212 E or Online

Office Hours: Online by Appointment

Website: www.cis.fsu.edu/~sadjadi

Teaching Assistant: Sumesh Kumar, skuma027@fsu.edu, Office Hours Tu/Th 12-2pm

Course Time Zone | Eastern Standard Time (EST).

Course due dates are according to this time zone.

Add Course

Apply

Apply

Apply

Apply

Apply

Apply

Apply

Apply

Apply

ACH4302 > Modules

Home

Discussions

Grades

People

Syllabus

GraderFinder

Assignment 1

Due: Dec 17, 2019 at 11:59pm - COP2210

6/7 Graded

13.67 / 20 (68%) Average

1/7

Emily Boone

Download

Zoom

Copy

Print

Fullscreen

```
// if you want to draw graphics on the screen, use the paintComponent method
// it give you a graphic context to draw on
public void paintComponent(Graphics g){

    super.paintComponent(g);

    // when the player is still in the game
    if(!inGame){
        g.drawImage(apple, apple_x, apple_y, this);
        //tell the graphic context to draw the
        for (int z = 0; z < dots; z++) {
            if (z == 0)
                g.drawImage(head, collisionX[z], collisionY[z], this);
            else g.drawImage(dot, collisionX[z], collisionY[z], this);
        }
        Toolkit.getDefaultToolkit().sync();

        // if you do not dispose
        g.dispose();
    }
    else gameOver(g);
}
```

Submitted: Jun 23 at 7:50am **LATE**

Submitted Files: (click to load)

Jupiter.docx

Assessment

Grade out of 20

Assignment Comments

Add a Comment

Submit

Download Submission Comments

5.2 User Interface Final Design

Applications for course

Account

Dashboard

Courses

Calendar

Inbox

TestCourse2

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Applications

Prerequisites

Settings

This course has the following applications for grading:

Assignment Name	Number of Applicants	View Applications
Test3	1	View

Grader information modification

Account

Dashboard

Courses

Calendar

Inbox

GPA, Current GPA: 3.6

Enter new GPA, or leave blank to keep the same

PID, Current PID: 34234

Enter new Student ID, or keep blank to leave the same

Taken Courses:

Course Name	Grade	
CCC333	A	Delete
BBB222	B	Delete

Add more prerequisites:

Course Name	Grade
None	None

Add course

Remove course

Save

Course Prerequisite Setup

Account

Dashboard

Courses

Calendar

Inbox

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Applications

Prerequisites

Settings

TestCourse4

Course Name

Grade

None

None

Add course

Remove course

Save Course prereqs

Change Course Prerequisites

Account

Dashboard

Courses

Calendar

Inbox

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Applications

Prerequisites

Settings

TestCourse2

Current Prerequisite Courses:

Course Name	Grade	Delete
AAA111	A	Delete
BBB222	A	Delete
CCC333	B	Delete

Add more prereqs:

Course Name

Grade

None

None

Add course

Remove course

Save Course prereqs

Accept Grader Application

TestCourse2

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Applications

Prerequisites

Settings

This course has the following applications for grading:

Applicant Name	Applicant Details	Assign Section
dg	Grader Info	None

Save Grader application

Grader Application Confirmation Page

TestCourse2

Account

Dashboard

Courses

Calendar

Inbox

Are you sure you would like to grade the following assignment:

Course Name	Assignment Name	Check to Confirm
TestCourse2	Test3	☐

Confirm

Grader Prerequisite Page

Account

Dashboard

Courses

Calendar

Inbox

GPA

Enter your current GPA

PID

Enter your Student ID

Course Name

Grade

None

None

Add course

Remove course

Create Cf grader

Eligible Assignments on Grader Homepage

Account

Dashboard

Courses

Calendar

Inbox

Grader Homepage

Assignments grader is eligible for:

Course Name	Assignment Name	
TestCourse	TestAssignment	Apply
TestCourse2	Assignment Test 2	Apply
TestCourse2	Test3	Apply
TestCourse3	AssignmentTest	Apply
TestCourse4	yty	Apply

Modify Grader Information

Grader Information Available to Teacher

TestCourse2

Grader Information

Name: dg
GPA: 3.8
PID: 2334523

Courses Taken:

Course	Grade
AAA111	A
BBB222	A
CCC333	A

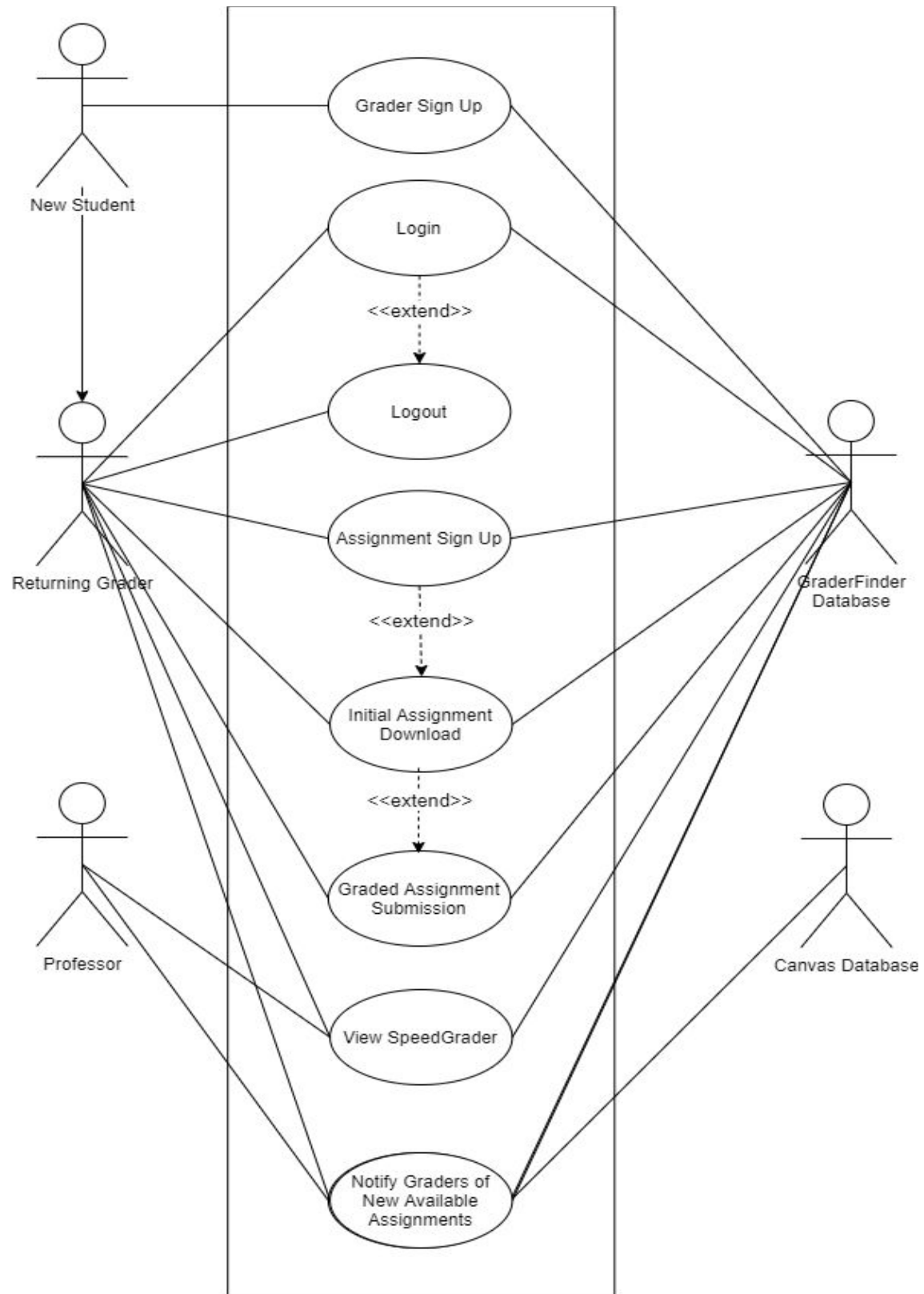
Home
Announcements
Assignments
Discussions
Grades
People
Pages
Files
Syllabus
Outcomes
Quizzes
Modules
Applications
Prerequisites
Settings

Grader Application Pending

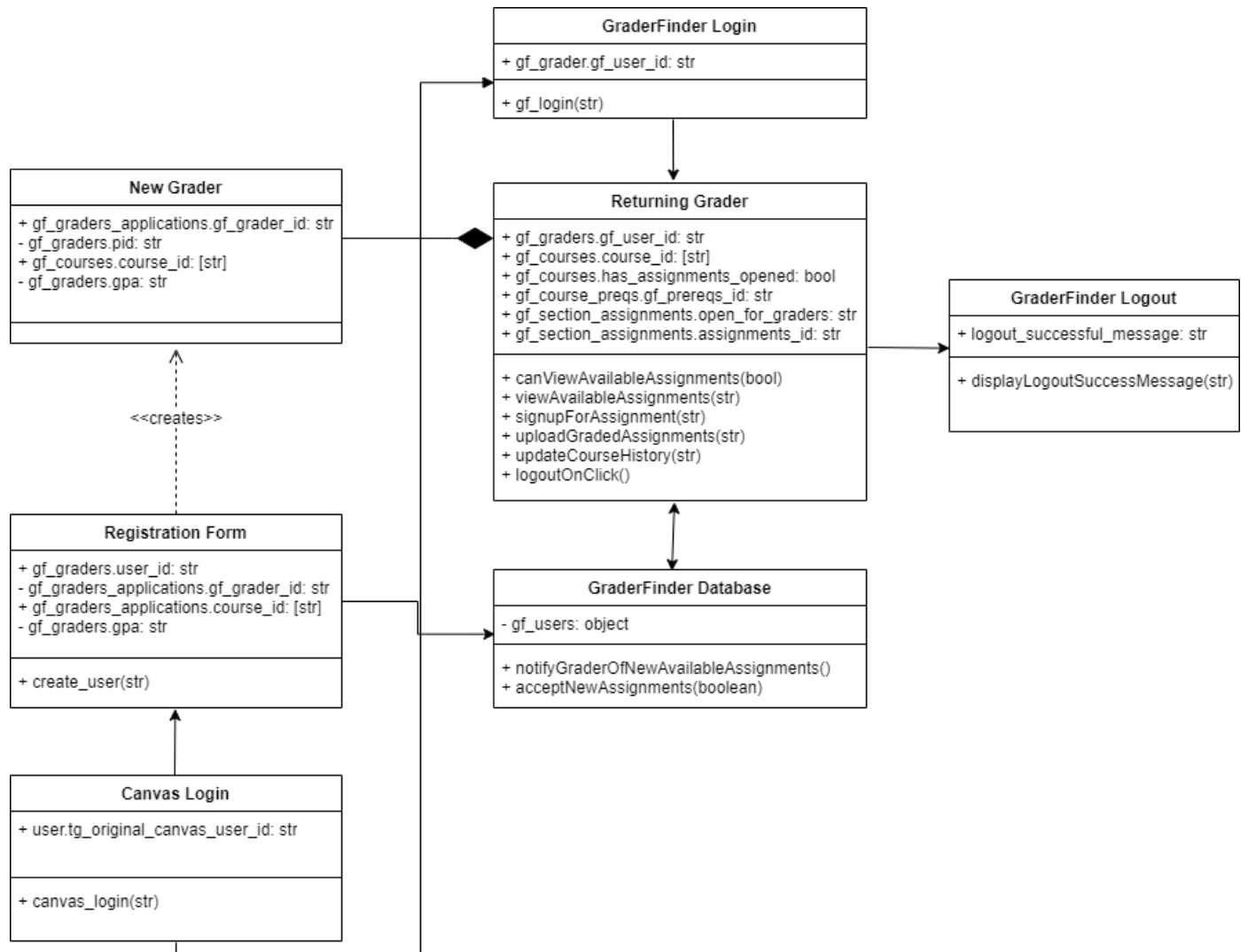
Your application is being reviewed by the course's professor.

Account
Dashboard
Courses
Calendar
Inbox

5.3 Use Case Diagram

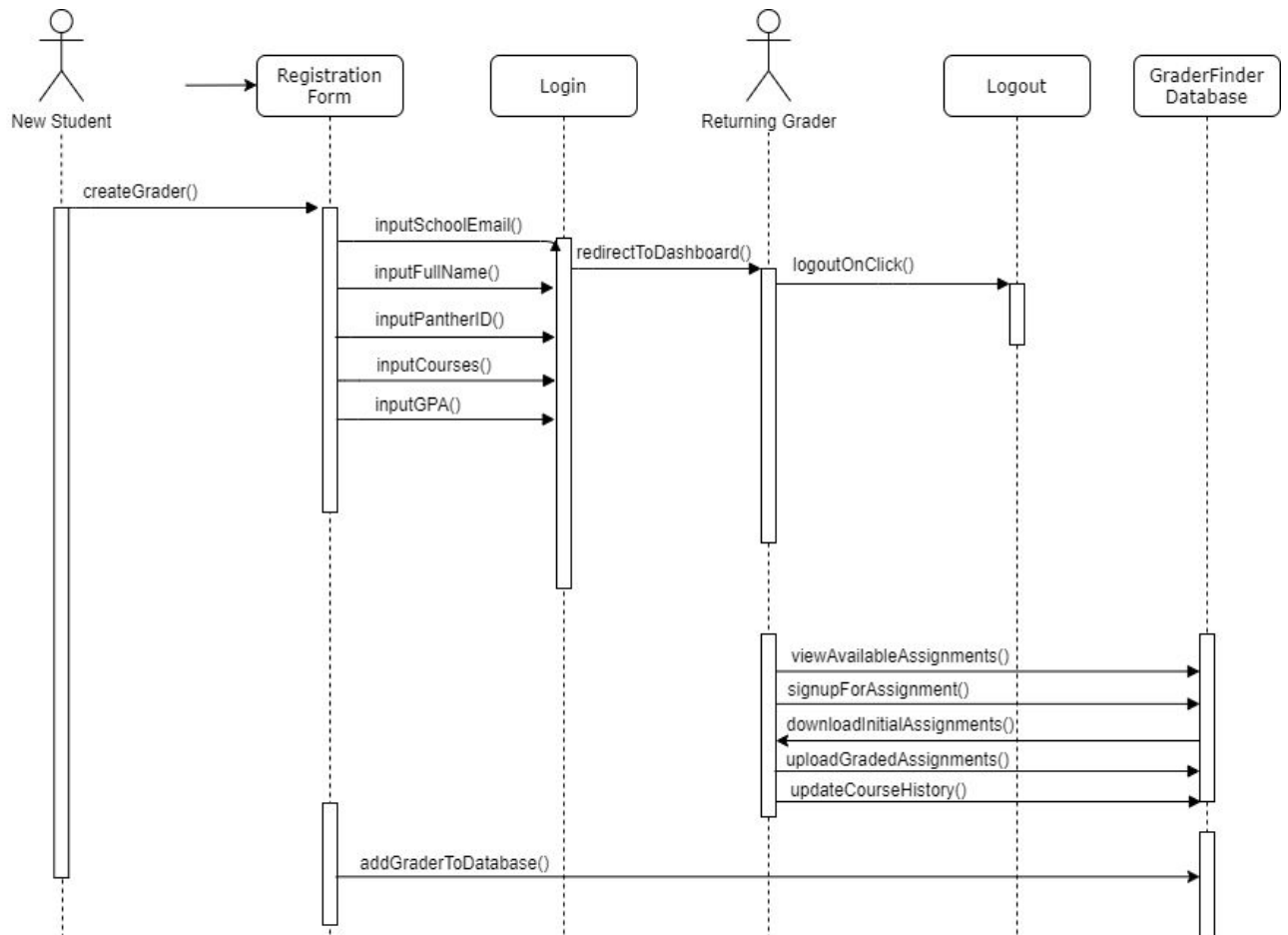


5.4 Class Diagram

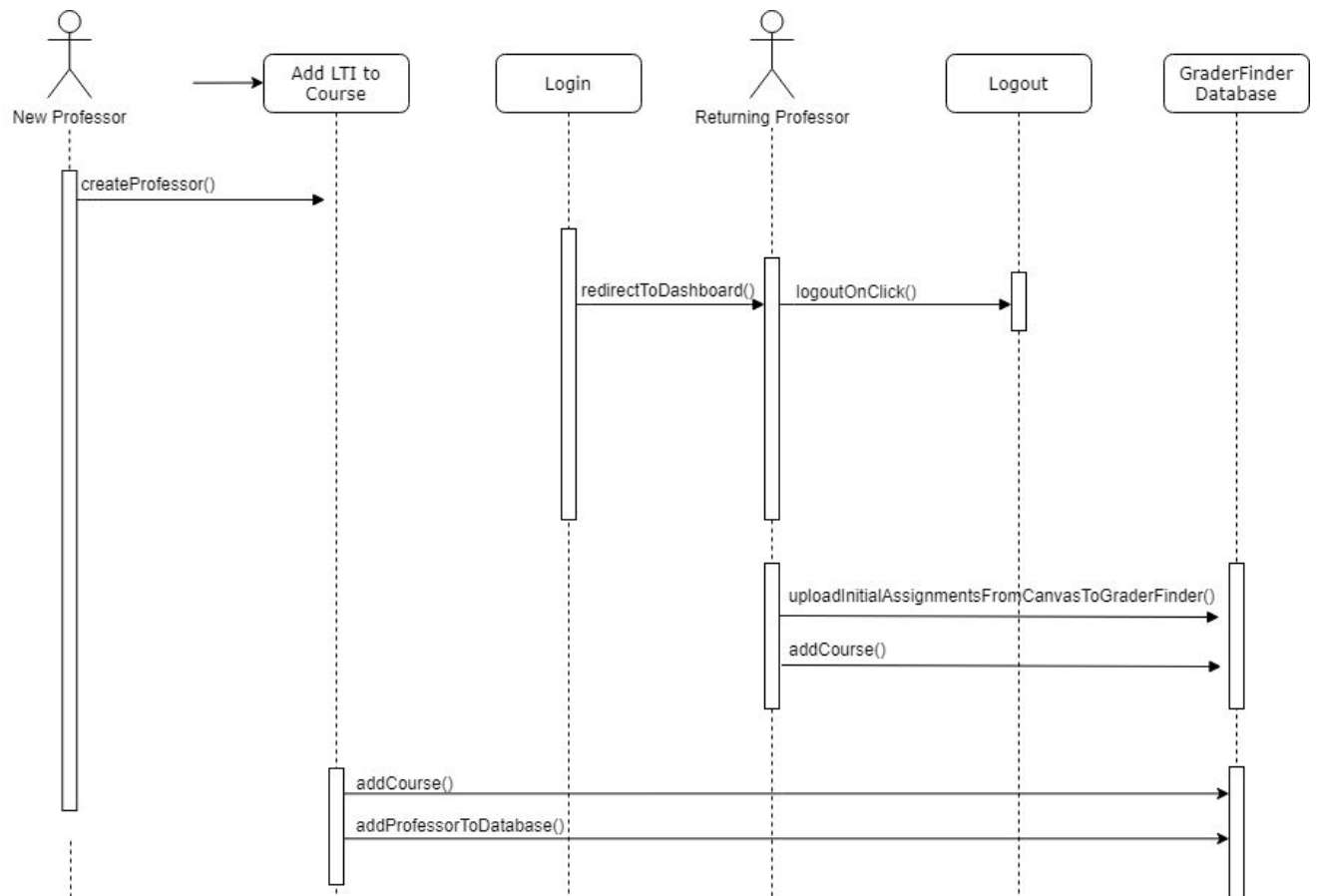


5.5 Sequence Diagrams

4.5.1 Grader View



5.5.2 Professor View



6. Appendix

6.1 Appendix A - Sprint Review

6.1.1 Sprint 1 Review

Attendees: Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo, William Padron, David Paez, Leila Adaza, Cheska Sy

Product Owner: Deng Pan

Date: 9/11/20

Time: 7:46 PM - 7:52 PM

User Stories Accepted by Product Owner

1. Set up a GitHub repository
2. Read about LTI specification and find out how it works
3. Learn how to use Ruby programming language
4. Determine how to notify graders/do the matching (for example, Uber)
 - a. Notifications for assignments available, instructor choosing them, and reviewed
5. Design grader rating system
6. Create GraderFinder website that is hosted separately/in its own domain

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 1 - System returns/screens a list of eligible students
- User Story 2 - Button on students' Canvas dashboard to register as a potential grader rather than in the Course module alone
- User Story 3 - Return grades from Canvas
- Other user stories were postponed due to too big of a scope. These will be refined and broken down for the next sprint.

6.1.2 Sprint 2 Review

Attendees: William Padron, David Paez, Leila Adaza, Cheska Sy, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 9/25/20

Time: 7:30 PM - 8:00 PM

User Stories Accepted by Product Owner

1. User Story 1 - Add new users
2. User Story 4 - Create form to take new grader's GPA, prerequisite courses, and grades for those courses
3. User Story 6 - Use case diagram
4. User Story 10 - Sequence diagram

User Stories Moved to the Product Backlog for Future Sprint Planning

- Enrique Rodicio, Justin Cabrera
 - User Story 2 - Create methods to assign a ta to assignment
 - User Story 3 - Create a page to display the User story 2 methods
- Nicholas Delamo, Damian Torrens, Phillip Nguyen
 - User Story 11 - Save progress of the grading in a separated database (Either partial or by parts) (7 pts.)
 - User Story 12 - Re-upload submissions button (with GraderFinder annotation perhaps) OR Canvas Grade Web API:
 - <https://canvas.instructure.com/doc/api/submissions.html> (8 pts.)
 - User Story 13 - Return grades with comments to instructor in Canvas (7 pts.)
 - User Story 14 - Have notifications set for deadlines with assignments that graders have agreed to grade
- Capstone 1 - Leila Adaza, Cheska Sy, David Paez, William Padron

- User Story 5 - Documentation/diagrams for Student Grader Registration

Button:

- User Story 7 - Class diagram/s
- User Story 8 - Database schema/s
- User Story 9 - Activity diagram/s

6.1.3 Sprint 3 Review

Attendees: Leila Adaza, Cheska Sy, William Padron, David Paez, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens

Product Owner: Deng Pan

Date: 10/9/20

Time: 7:40 PM - 8:07 PM

User Stories Accepted by Product Owner

1. User Story 2 - Create methods to select assignments grader is eligible for
 - a. UI less important
2. User Story 5 - Continue working on Use Case diagrams to make more comprehensive and cover edge cases; think about all possible scenarios

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 13 - Return grades with comments to instructor in Canvas (7 pts.)
 - UI less important
 - How grades can actually be returned back to Canvas
 - How this should be reflected on the user story definition
 - Refer to Canvas API documentation more often

6.1.4 Sprint 4 Review

Attendees: William Padron, David Paez, Leila Adaza, Cheska Sy, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 10/23/20

Time: 7:30 PM - 7:45 PM

User Stories Accepted by Product Owner

1. User Story 1 - Implement way for graders to apply to grade an assignment
2. User Story 3 - Create additional documentation for GraderFinder workflow
 - a. Sequence diagram for professor's view
 - b. Class diagram

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 1 - System returns/screens a list of eligible students
- User Story 13 - Return grades with comments to instructor in Canvas
 - Showcase LTI app in Backlog Grooming meeting Monday 11/2

6.1.5 Sprint 5 Review

Attendees: William Padron, David Paez, Leila Adaza, Cheska Sy, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 11/06/20

Time: 7:40 PM - 7:49 PM

User Stories Accepted by Product Owner

1. User Story 1 - Allow instructor to select grader instead of first come first serve
2. User Story 3 - Improve class diagram based on PO's previous feedback
 - a. Replace current fields with fields from SQL Canvas table
 - b. Create alternative, streamlined use case diagram for LTI implementation

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 2 - LTI App demo

- Start with simple functions
- Challenge in gradebook with LTI app

6.1.6 Sprint 6 Review

Attendees: William Padron, David Paez, Leila Adaza, Cheska Sy, Justin Cabrera, Enrique Rodicio, Phillip Nguyen, Damian Torrens, Nicholas Delamo

Product Owner: Deng Pan

Date: 11/20/20

Time: 7:35 PM - 7:54 PM

User Stories Accepted by Product Owner

1. User Story 1 - Begin preparing Group Poster Materials for Senior Showcase
2. User Story 2 - LTI App demo
3. User Story 3 - Actual GraderFinder LTI app
 - a. Integrated in Canvas LMS

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 3 - Actual GraderFinder LTI app
 - Only need to implement assignment upload integration component
- User Story 2 - LTI App demo

6.1.7 Sprint 7 Review

Attendees: Nicholas Delamo, Justin Cabrera, Phillip Nguyen, Damian Torrens, Enrique Rodicio, William Padron, David Paez, Leila Adaza, Cheska Sy

Product Owner: Deng Pan

Date: 11/4/20

Time: 7:35 PM - 7:41 PM

User Stories Accepted by Product Owner

1. User Story 1
 - Write information for user manual
 - Fill out content for installation
 - Finish sprint reviews
 - Finalize user design interface by compiling the screenshots of the product
 - Reformat user stories to follow “As a user/developer, I want to do X to achieve Y.”
2. User Story 2
 - Explain additional tables and describe the Canvas system architecture for System Design documentation section
3. User Story 3
 - Work on individual posters
4. User Story 5
 - Work on UI of Graderfinder

User Stories Moved to the Product Backlog for Future Sprint Planning

- User Story 4 - Moved to Pending*
 - Finish LTI app demo along with documentation

6.1.8 Sprint 8 Review

Attendees: Damian Torrens, Nicholas Delamo, Phillip Nguyen, Justin Cabrera, Enrique Rodicio, William Padron, David Paez, Leila Adaza, Cheska Sy

Time: 7:47 PM - 7:49 PM

User Stories Accepted by Product Owner

1. User Story 1 - Finish compiling files for Final Deliverable

6.2 Appendix B - User Manual and Installation

6.2.1 User Manual

The full manual can be found in UserGuide.pdf in the Documents Folder

6.2.2 Installation

The full installation guide can be found in `InstallationGuide.pdf` in the Documents Folder

6.2.3 WishList

The following are features that should be implemented into GraderFinder in the future:

- LTI integration into Canvas, where GraderFinder can be given access to a teacher's gradebook to automatically update grades.
- Make code more uniform with Canvas's code
- Disable irrelevant Canvas features
- Prevent graders from accessing assignments they did not apply to grade. Currently, they have access to the whole course as Tas
- Add input validation for GPA
- Prevent duplicate prerequisites from being entered

7. Closing Remarks

Although this is the first version of the GraderFinder LTI application, our team was able to make notable progress including the implementation of a new grader registration page, an assignment signup page for registered graders who meet given prerequisites with course/assignment details and interactive, functional UI elements, and a working backend comprised of a custom GraderFinder database using PostgreSQL that currently holds grader, course, and assignment information. Components that students next semester will most likely have to implement would be the integration of the SpeedGrader app for graders to utilize for a more streamlined grading process that interacts directly with the Canvas database instead of having to go through the GraderFinder database during as well.