GRAD SCHOOL ZERO TEAM T

Daniel Aguilar-Rodriguez, Julia Berdecía, Baishaki Debi, Iqra Qumar, Farnaz Zinnah

Software Requirements Specification

CCNY Grove School of Engineering| CSC 322 Software Engineering

Revision History

Date	Version	Description	Author
10/24/21	draft	First pass - draft	J. Berdecía
10/25/21	1.0	Incorporate teammates' review comments	J. Berdecía

Table of Contents

Introduction

Purpose

GradSchoolZero is a web-based university management system designed for students and faculty to manage courses, view grading, and access curricula.

Scope

Major Project Activities

- *Dividing Responsibilities*

Tasks are divided according to team members' strengths and preferences. However, no team member should feel confined to one specific role.

- *Clarifying Requirement Specifications*

Requirement specifications must be broken-down, paraphrased, and simplified due to the elaborate nature of *GSZ*. Regular discussions and consistent communication ensure that the team is collectively clear on specifications.

- *Arranging Functionality Workflow*

It is vital to understand the functionality and workflow of *GSZ* prior to building the program. Although time-consuming, this activity will help the team avoid excessive modifications and confusion throughout the building process.

- *Drafting Wireframes*

The team would like to plan and test a variety of user interfaces before integrating into the program. Select team members devise and provide samples of wireframes to access the arrangement of functionality and ease of use.

- *Software Installation*

All team members must ensure that the required software is downloaded as soon as possible. This allows for time to troubleshoot for bugs or installation issues.

- *Software Familiarization*

Team members will devote time to learning to work with the required software and share related resources.

- *Web Deployment*

Team member will deploy Flask application to Heroku.

Deliverables

- *Blank Flask Project **(COMPLETE)***

Create a blank Flask Python project and execute the program from the terminal.

- *Repository* ***(COMPLETE)***

Create team repository for version control and the integration of project changes.

- *Flask Project Skeleton* ***(COMPLETE)***

Create generic/mostly blank project web pages in Flask application. Ensure program executes from the terminal and push changes to repository.

- *Wireframe Draft for Home Page* ***(COMPLETE)***

Create wireframe draft for home page with online tools. Share with teammates for approval and discussion.

- *Phase 1 Report* ***(COMPLETE)***

Create Phase 1 report to report project progress thus far.

- *Software Resources* ***(COMPLETE)***

Share software resources with teammates to expedite familiarization.

- *Wireframe Drafts for Remaining Pages*

If needed, teammates will draft wireframes for the layout of the remaining pages to be composed. <u>This deliverable will be broken down by individual web page in the next report.</u>

- *Flask Project Home Page*

Complete the home page for *GSZ* and execute from the terminal.

- *Flask Project Remaining Pages*

Teammates will complete the remaining web pages for the *GSZ* program. <u>This deliverable will be broken down by individual web page in the next report.</u>

<u>Acceptance Criteria</u>

- Each role in *GSZ* must have the correct corresponding access according to requirement specifications.
- A storage method must be implemented to keep track of student rosters, grades, and other features in *GSZ* (text, CSV, JSON files are currently being considered).
- Changes made by a particular role must successfully be reflected in the dashboard of another role. For example: If an instructor fails a student, this 'F' will be reflected in the student's record.

Definitions, Acronyms, and Abbreviations

GSZ: *GradSchoolZero*

Flask: A Python-based web framework

Wire Frame: A blueprint representing the framework of a website

Role: Visitor, Student, Instructor, Registrar

Dashboard: A display of one's data

Skeleton: A basic outline

References

Flask Tutorial

Project Specs

Overview

GradSchoolZero is a web-based university management system designed for students and faculty to manage courses, view grading, and access curricula.

There are four roles in GSZ:

1. Visitors
2. Students
3. Instructors
4. Registrar

There are four periods within a semester:

1. Class Set-Up Period
2. Course Registration Period
3. Class-Running Period
4. Grading Period

All Users may access the home page to view an overview of the program, highest rated classes, lowest rated classes, and students with the highest GPAs.

Visitors may apply to study or instruct at *GSZ*.

Students may register for courses, drop courses, check grades, review courses, issue complaints, and file for graduation.

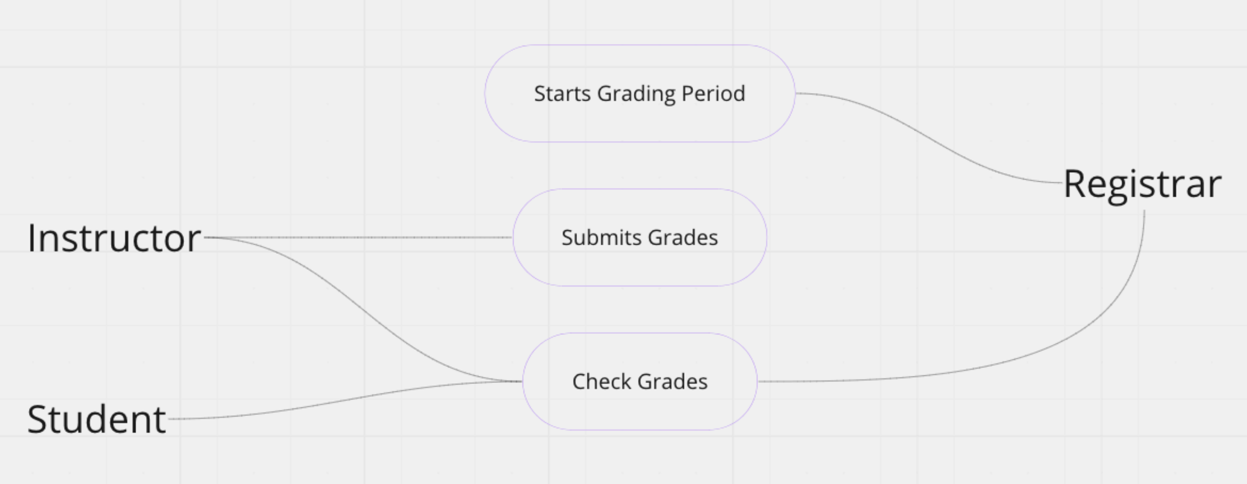
Instructors may issue grades, issue complaints, and view academic records of students.

Registrar may set up courses, change the period within the semester, access all student records, access all instructor information, and have all super user privileges.

Overall Description

Use-Case Model Survey

Example: Simple Grading Period Process



Assumptions and Dependencies

Assumptions

- Team T has access to all resources to complete this project, including sufficient team members, technology, and documentation
- The Flask application will be operational from the terminal throughout the work cycle
- The scope GSZ will not expand throughout the work cycle

Dependencies

- Software download
- Software familiarization
- Wireframes
- Git familiarization

Specific Requirements

Use-Case Reports

Case 1: Student drops class

1. Student logs in to *GSZ* and is redirected to the dashboard
2. Student scrolls down to current course list
3. Student clicks button displayed as 'Drop Class' next to corresponding course
4. Student dashboard refreshes with corresponding course specified as 'DROPPED'

Case 2: Instructor submits grades

1. Instructor logs in to *GSZ* and is redirected to dashboard
2. Instructor clicks button displayed as 'Assign Grades'
3. Instructor is redirected to page with assigned courses and corresponding students
4. Instructor assigns grades to students using dropdowns next to each student name
5. Instructor presses submit and the page is refreshed with corresponding students with their grades assigned

Case 3: Registrar changes period from Class Running Period to Grading Period

1. Registrar logs in to *GSZ* and is redirected to dashboard
2. Registrar clicks button displayed as 'Grading Period'
3. Registrar is redirected to Grading Period web page

Supplementary Requirements

- Usability
    - Would a user instantly recognize the purpose of this application?
    - How easy would it be for a user to learn to navigate the application?
    - How can we ensure that individuals with special needs can use this application?

- Resource Conservation
    - How much data should be stored?
    - Should we minimize the number of redirects?

Supporting Information

Miro Board | Functionality Outline

GitHub Repository

Wireframe Draft

Home Page Draft