<Team Z>

<GradSchoolZero>
Design Report
For <Graduate Program Management System>

Version <2.0>

Table of Contents

1. Introduction

2. All use cases

    2.1 Scenarios for each use case

    2.2 Collaboration or sequence class diagram for each use case

3. E-R diagram for the entire system

4. Detailed design

5. System screens

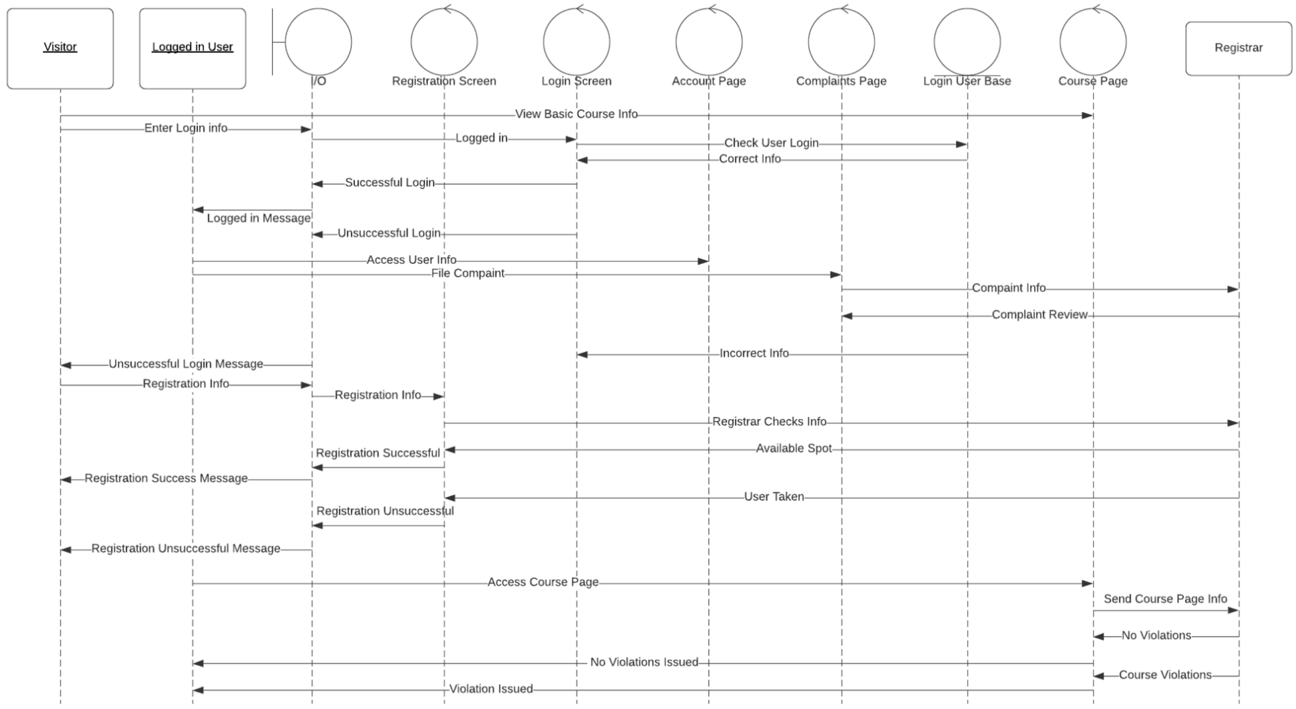
6. Memos of group meetings

7. Address of the git repo

Software Requirements Specification

1. Introduction

The Phase II report will provide the data structure and logic of the functionalities in the entire system. The image below will be an overall picture of the system using a sequence class diagram. It outlines the typical interaction between the system and the users.



2. Overall Description

2.1 Scenarios for each use case

Login: Students and instructors will have to login to their own page with all the basic information. New students who logged in for the first time will have to change the login information.

Register/Apply: a visitor can apply to be either a student or instructor

Continue as Visitor: visitors can access the course page without a need to be a student or instructor.

Basic Course Info: basic introduction of the program and anyone can view the basic information of highest/lowest rated classes and students with the highest GPA

Review Application: restistars are able to view, accept, and decline visitor applications to become a student or instructor.

Class Setup: During class set-up period, registrars set up classes, class time, class size, and course instructor on register course.

Register Course: All matriculated students can register 2-4 courses while following the rules during the course registration period.

Drop Course: Students can drop courses between registration and grading period for a grade W.

Student/Instructor Violations Handling: The handling is done by registrars who will give out warnings and suspensions.

Review Course: Students can write reviews for their classes and assign stars (1-worst to 5-best), which will be summarized in the class. Only registrars know who wrote which review. Students can not rate the class after the instructor posted their grade.

Assign Grades: instructors assign grades during the grading period and registrars are able to see them to check the requirements before giving out warnings to instructors.

Current Course Grades: lets students see their current course grades in their course page.

All Academic Record: Instructors can see basic information and academic records of the student.

Personal Record: Students page to see their own version of basic information and academic records which is different from the instructors view.

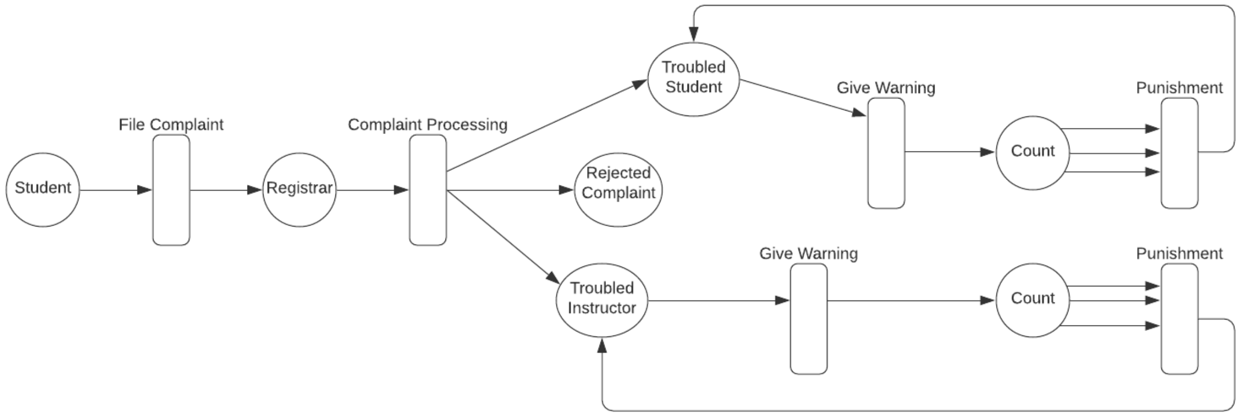
Apply for Graduation: Students finishing 8 classes can apply for graduation and registrars will check if students covered all required courses before graduating them with a masters degree.

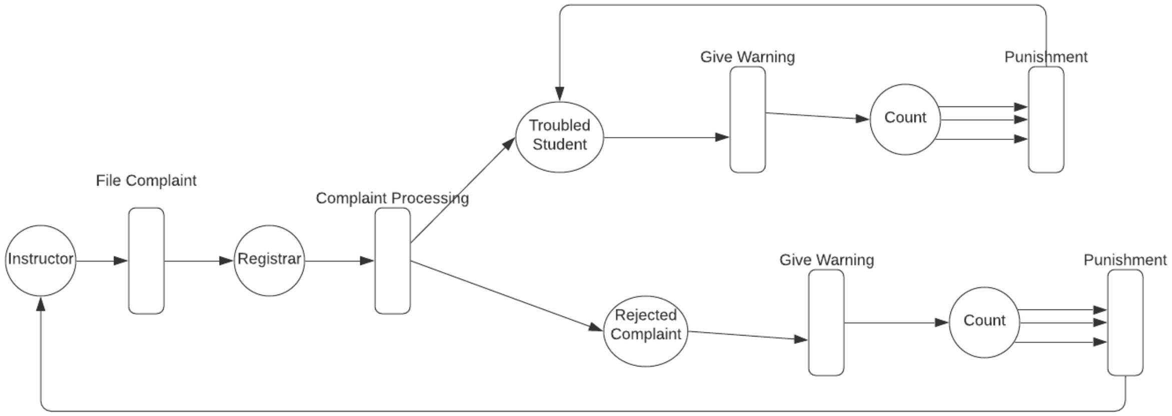
File Complaint: Students can complain about other students or instructors to the registrar. Instructors can complain about a student to the registrars to warn or deregister the student.

Complaint Review-Penalty: Registrars will process student complaints and issue a warning after an investigation. Registrars also process instructor complaints accordingly with a warning or deregister the student or punish the instructor with 1 warning.

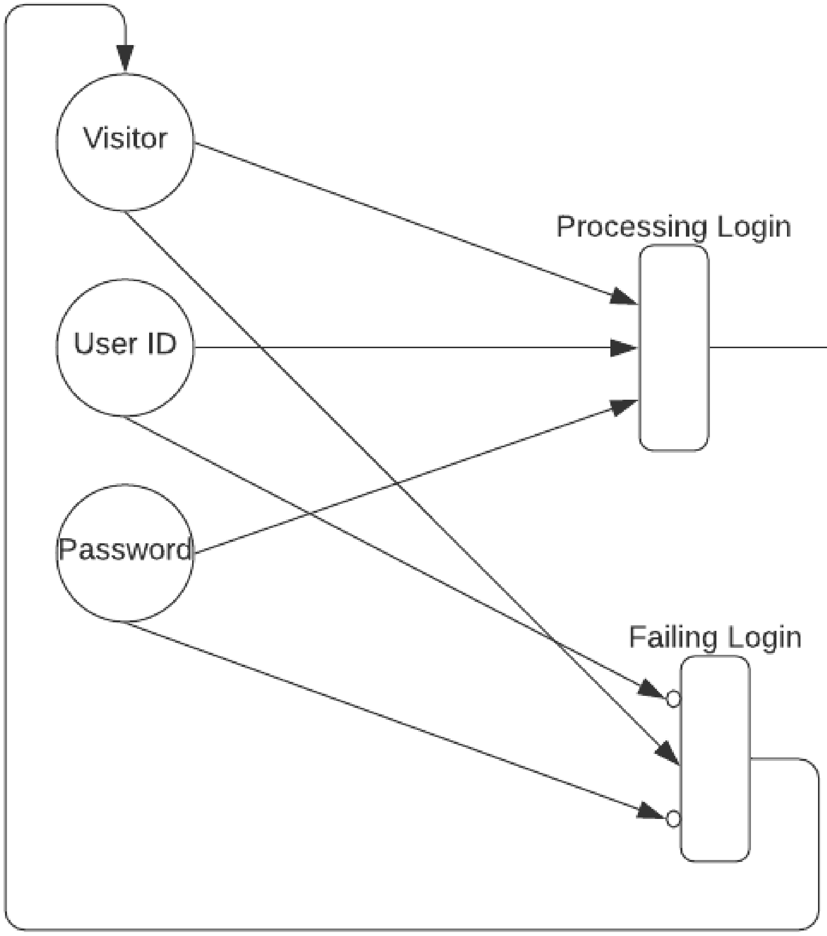
2.2 Collaboration or sequence class diagram for each use case

COMPLAINTS PAGE

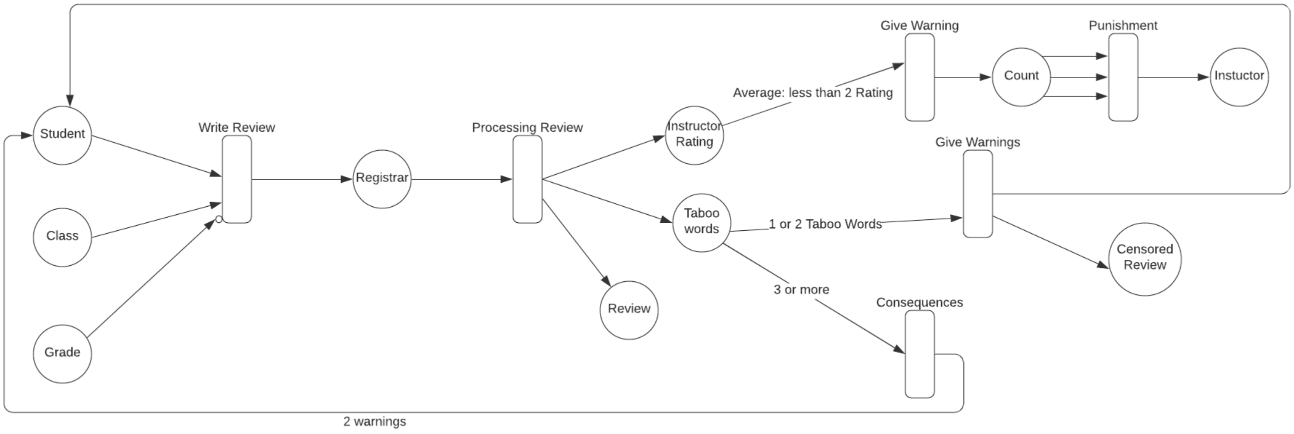




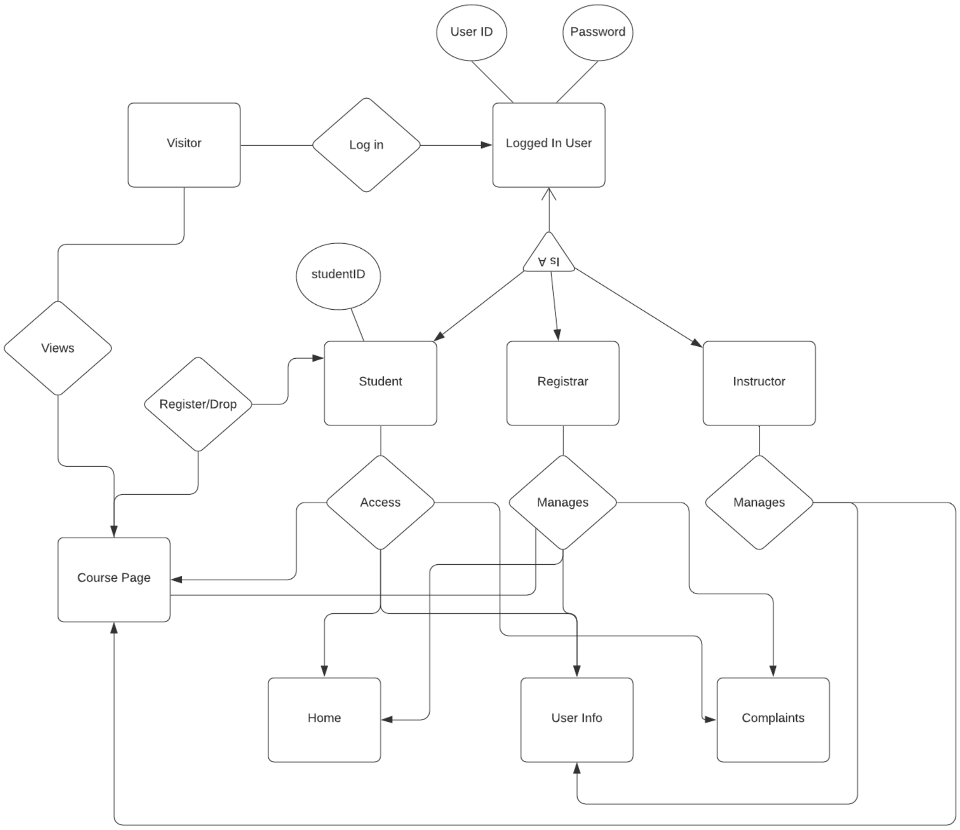
LOG-IN SCREEN



REVIEW COURSE SCREEN



3. E-R diagram for the entire system



4. Detailed design

Register/Apply For Student

```
def registerStudent(first_name, last_name, gpa, registrar_approval):
    # registrar_approval is a bool -> True means we have approval to enroll student
    if gpa > 3.0 and registrar_approval:
        new_student = Student(first_name, last_name, gpa)
        new_student.generateID()
        new_student.generatePassword()
        # .... less unimportant details follow here
        new_student_status = true
    else:
        new_student_status = false
    return new_student_status
```

```
def registerInstructor(first_name, last_name):
    If registrar_approval:
        new_instructor = Instructor(first_name, last_name)
        # approved instructors will be assigned class(es)
        new_instructor.getApprovedClassList()
        new_instructor_status = true
    else:
        new_instructor_status = false

def login(username, password):
    if login_form_is_valid():
        user = db.query.get(user).first()
        # .first() gets the 1st result from database or returns None if not found
        if user and user.check_password_hash(password):
            login_status = true
        else:
            print("invalid login")
            redirect("/login")
```

Review application

```
Def reviewApplication(student, application):
    student = db.query.get(student).first()
    if student and student.checkRequiredCourses():
        istudent.graduationStatus = True
        graduationList.append(student)
    else:
        student.warning +=1
        if student.checkForPunishment:
            Registrar.get_Registrar()
```

Personal Record

```
Def getPersonalRecord(user):
    if(user == student):
        display(student_record)
    elif(user == instructor)
        display(instructor_record)
```

File Complaint

```
Def fileComplaint(user):
    if user == student:
        investigation = studentComplaint(subject_user)
        if investigation:
            Subject_user.warning +=1
            subject_user.complaintReviewPenalty
            if subject_user.complaintReviewPenalty:
                Registrar.getRegistrar()
    elif user == instructor:
        Investigation = instructorComplaint(subject_student)
        if investigation:
            punishStudent(subject_student)
        else:
            user.warning += 1
            user.complaintReviewPenalty
```

Complaint Review-Penalty

```
Def complaintReviewPenalty(warning):
    while(warning > 3):
        registrar_fine += fine
        drop student
```

All Academic Records

```
Def acadRecord(login):
    if(user == instructor):
        display(basicInfo)
        display(allStudentInfo)
    elif(user == student):
        display(basicInfo)
    else:
        display(everything)
```

Basic Course Info

```
Def basicCourseInfo():
    Display highest rated classes
    Display lowest rated classes
    Display students with the highest GPA
```

Student/Instructor Violations Handling

```
Def violationHandling(user):
    warnings = user.getWarnings
    if warnings >= 3:
        punishUser(user)
        callForRegistrar()
```

Register Course

```
Def registerCourse(student_user):
    if(student_user == wantClass):
        insert class
```

Drop Course

```
Def dropCourse(studentID, courseID, period):
    if period = class_running OR registration:
        Drop course
        course_count–
    if course_count = 0:
        suspend for 1 semester
```

Assign Grades

```
Def assignGrade(student):
    if student in courseList:
        total = sum(course.getStudentGrades(student))
        final_grade = ( total // len(course.getStudentGrades(student)-1) )
        student.grade(course, final_grade)
```

Current Course Grades

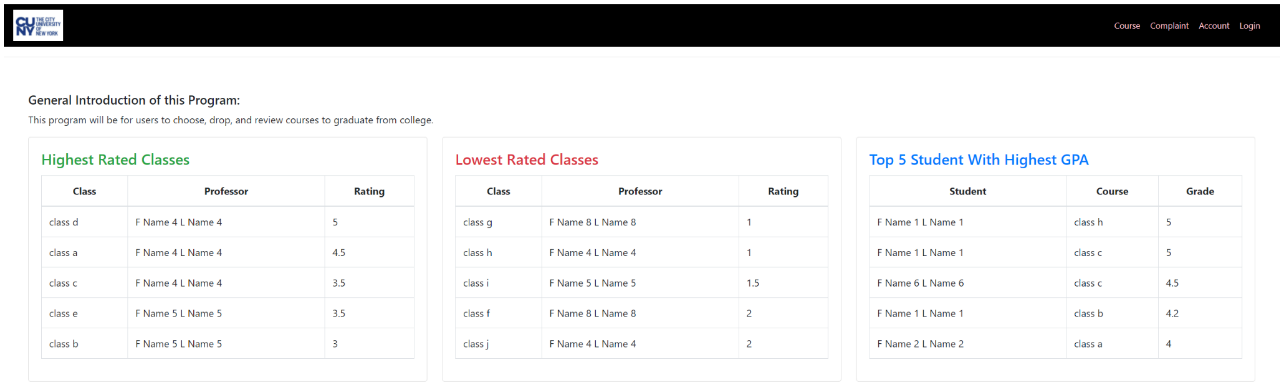
```
Def currentCourseGrade(currentSemester):
    While(currentSemester == True):
        if(studentGrades == NULL):
            assign(grades)
        else:
            warning += 1
```

Class Set Up

```
Def classSetUp(registrarID, period):
    if period = class_set_up AND registrarID = valid:
        set up classes, class time, course instructors and class size
```

5. System screens

HOME



General Introduction of this Program:

This program will be for users to choose, drop, and review courses to graduate from college.

Highest Rated Classes

Class	Professor	Rating
class d	F Name 4 L Name 4	5
class a	F Name 4 L Name 4	4.5
class c	F Name 4 L Name 4	3.5
class e	F Name 5 L Name 5	3.5
class b	F Name 5 L Name 5	3

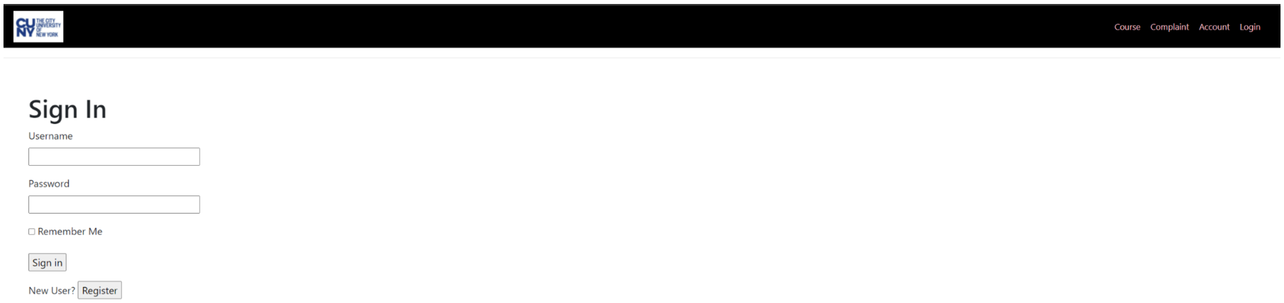
Lowest Rated Classes

Class	Professor	Rating
class g	F Name 8 L Name 8	1
class h	F Name 4 L Name 4	1
class i	F Name 5 L Name 5	1.5
class f	F Name 8 L Name 8	2
class j	F Name 4 L Name 4	2

Top 5 Student With Highest GPA

Student	Course	Grade
F Name 1 L Name 1	class h	5
F Name 1 L Name 1	class c	5
F Name 6 L Name 6	class c	4.5
F Name 1 L Name 1	class b	4.2
F Name 2 L Name 2	class a	4

LOG-IN SCREEN



Sign In

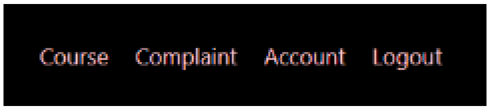
Username

Password

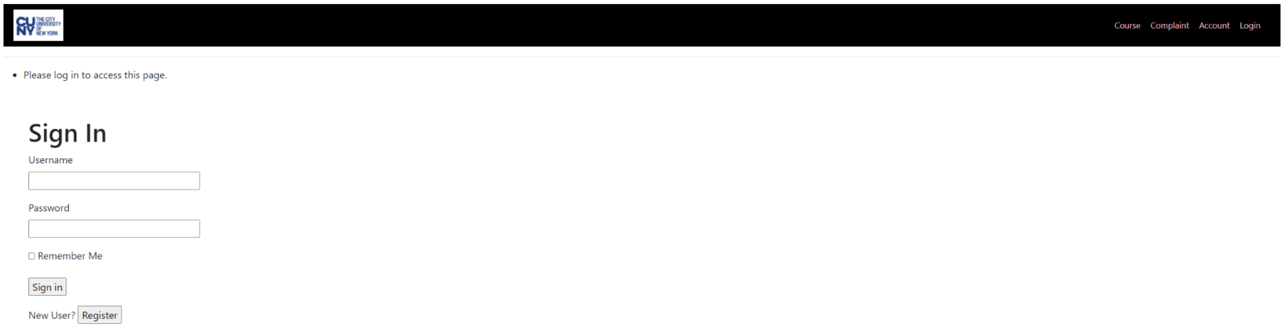
□ Remember Me

New User?

NAVIGATION BAR



LOG-IN REQUESTED TO VIEW COMPLAINTS



- Please log in to access this page.

Sign In

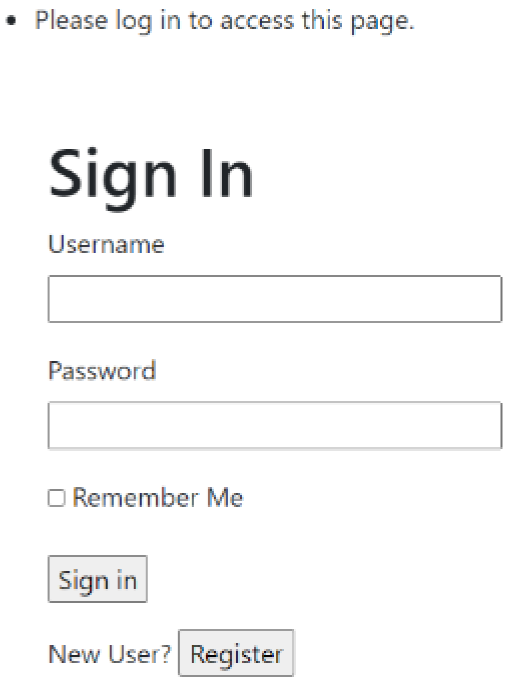
Username

Password

□ Remember Me

Sign in

New User? Register



- Please log in to access this page.

Sign In

Username

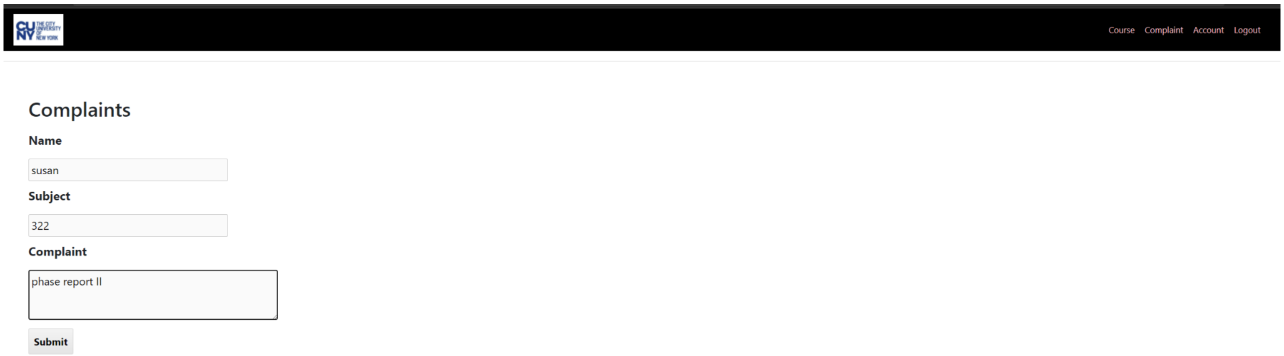
Password

□ Remember Me

Sign in

New User? Register

COMPLAINTS PAGE



Complaints

Name

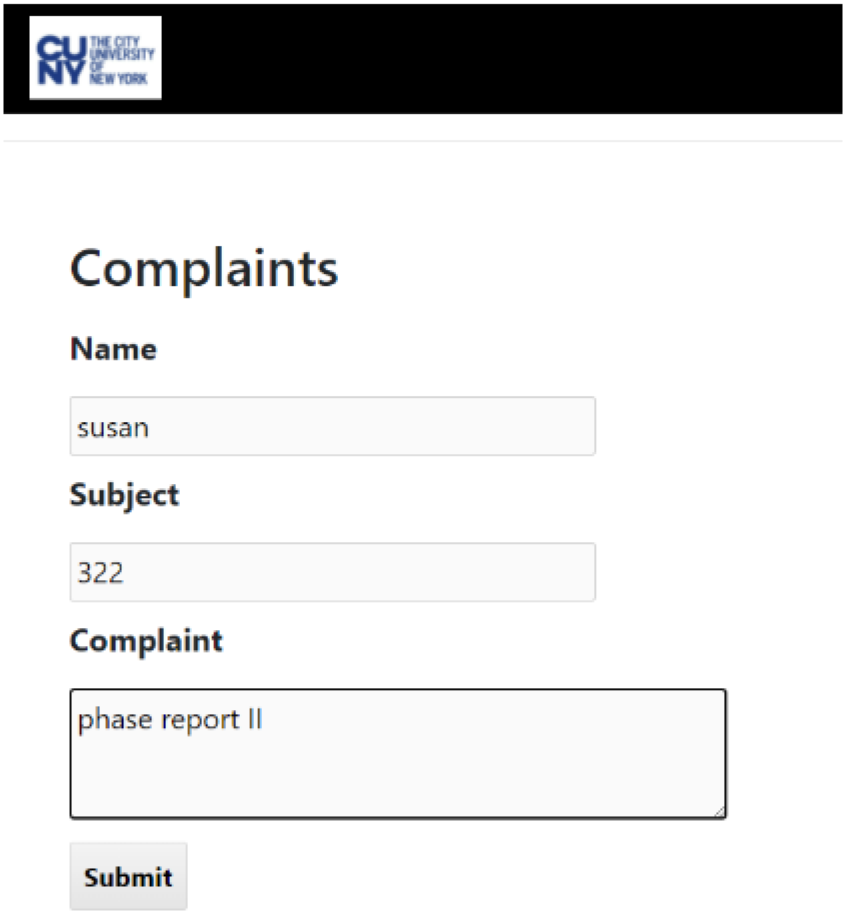
susan

Subject

322

Complaint

phase report II



Complaints

Name

susan

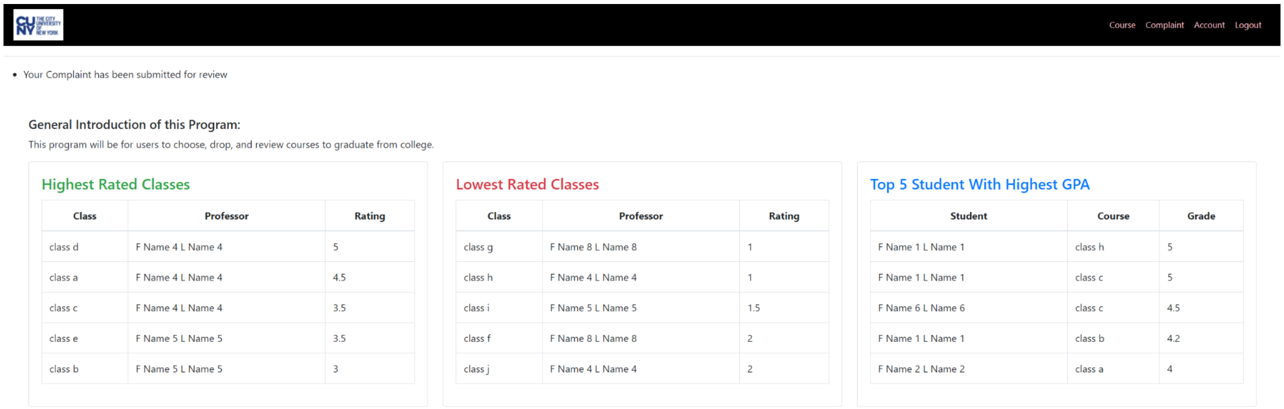
Subject

322

Complaint

phase report II

COMPLAINT PAGE SUBMITTED



- Your Complaint has been submitted for review

General Introduction of this Program:

This program will be for users to choose, drop, and review courses to graduate from college.

Highest Rated Classes

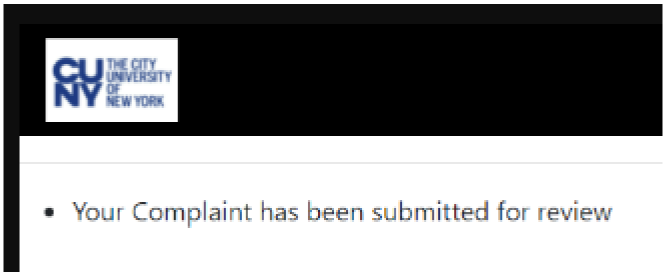
Class	Professor	Rating
class d	F Name 4 L Name 4	5
class a	F Name 4 L Name 4	4.5
class c	F Name 4 L Name 4	3.5
class e	F Name 5 L Name 5	3.5
class b	F Name 5 L Name 5	3

Lowest Rated Classes

Class	Professor	Rating
class g	F Name 8 L Name 8	1
class h	F Name 4 L Name 4	1
class i	F Name 5 L Name 5	1.5
class f	F Name 8 L Name 8	2
class j	F Name 4 L Name 4	2

Top 5 Student With Highest GPA

Student	Course	Grade
F Name 1 L Name 1	class h	5
F Name 1 L Name 1	class c	5
F Name 6 L Name 6	class c	4.5
F Name 1 L Name 1	class b	4.2
F Name 2 L Name 2	class a	4



- Your Complaint has been submitted for review

6. Memos of group meetings

Date	Time	Description	Time Limit
<dd/mmm/yy>	<hour/min>	<details>	<hour/min>
<28/Sep/2021>	5:00PM	Confirming team members Team member introductions	30mins
<01/Oct/2021>	5:30PM	Brief discussion on Project Spec Assigning roles to project	60mins
<10/Oct/2021>	4:30PM	Setting up environments	60mins
<17/Oct/2021>	5:00PM	Discussion and planning of phase I report	20mins

<24/Oct/2021>	4:15PM	Phase I report UML diagram	75mins
<26/Oct/2021>	5:00PM	Cleaning and proofreading report Phase I report completed	30mins
<31/Oct/2021>	5:00PM	Git repo active and contributing to repository Make sure everyone is familiar with Git	60mins
<07/Nov/2021>	4:30PM	Check up on what everyone has contributed and further discussion of project	20mins
<15/Nov/2021>	5:00PM	Discussion on whether or not to use real data base or dummy data	30mins
<23/Nov/2021>	4:30PM	Working on our assigned task together Commiting to git repo	60mins
<24/Nov/2021>	5:00PM	Working on Phase II report	30mins
<25/Nov/2021>	5:00PM	Working on Phase II report	4hours
<26/Nov/2021>	3:00PM	Finishing and proofreading Phase II report	6hours

7. Address of the git repo

Address of the git repo (github, gitlab, bitbucket, etc) of your team's work so far - put all materials including this report there
https://github.com/tryingtolearn11/322_project.git