Team A

GradSchoolZero

Design Report

For GradSchoolZero

Version 2.0

Revision History

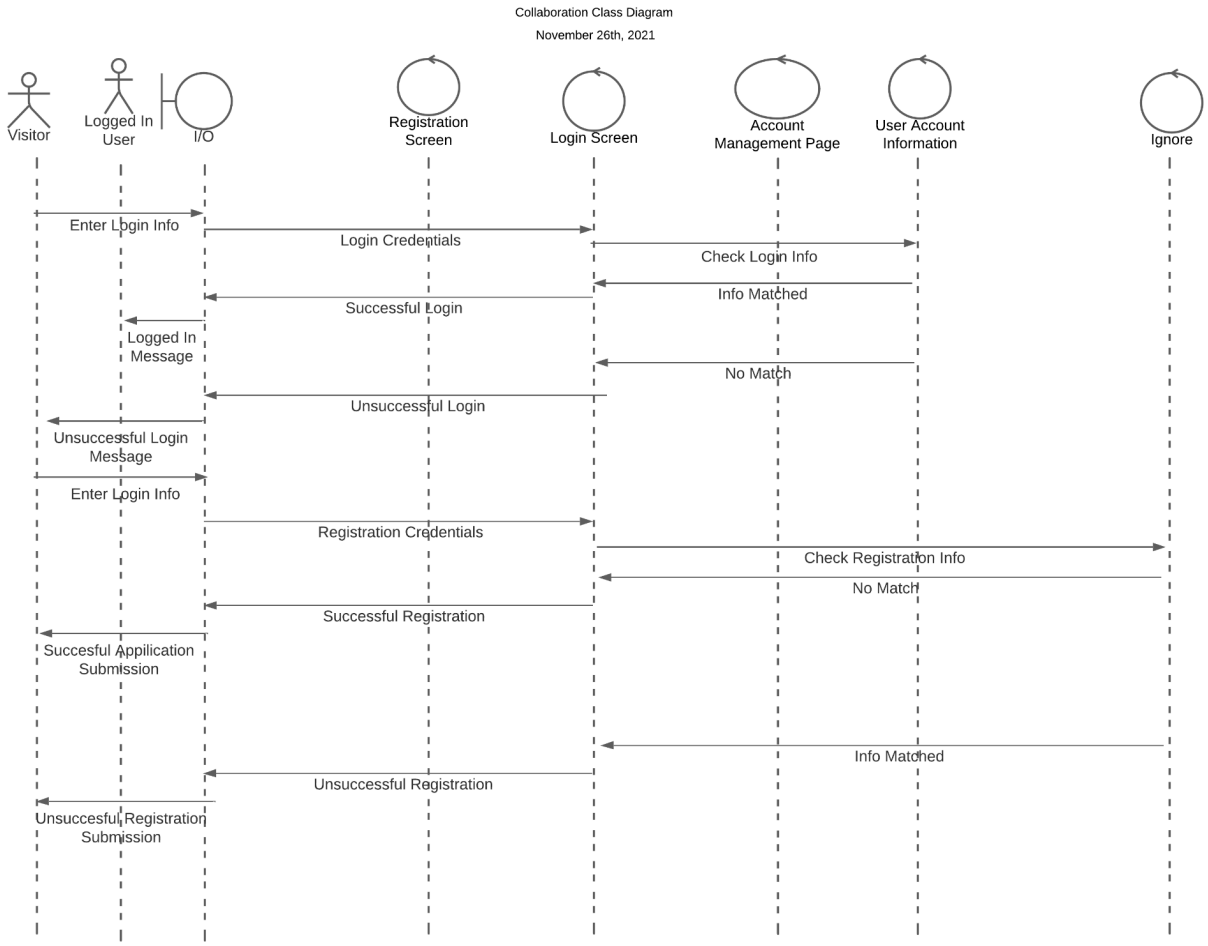
Date	Version	Description	Author
11/22/21	2.00	Write use-case scenarios (section 2.1), draw E-R diagram (section 3)	Atiya Mirza
11/23/21	2.01	Updated section 2.3, 5.1, and 6	Carlos Flores
11/26/21	2.02	Section 4	Willie Shi
11/26/21	2.03	Section 1.1 & 5.2	MdShahid Emdad
11/26/21	2.04	Final touches	Carlos Flores
11/26/2021	2.05	Even more final touches	Willie Shi

Table of Contents

Design Report

1. Introduction

1.1 Collaboration Class Diagram



2. Use Cases

2.1 Use-Case Scenarios

Available at Any Time

Homepage: Any type of user can visit the homepage to view some basic and statistical information about the graduate program. There is only a normal case which is that the user can view top courses, top students, and the program's mission statement.

Apply: Any visitor can apply to be a student or instructor in the graduate program, and only the registrars can approve or reject either type of application. For students, the normal case is that the visitor has GPA > 3.0 and the program quota is not reached, so the visitor is accepted by the registrar as a student and receives their unique id and password. Exceptional cases occur when the visitor has GPA < 3.0 and/or the program quota is already reached, so they are rejected with a justification. Another exceptional case is that the student's application is filled out incorrectly so it is

not submitted. For instructors, the normal case is that the visitor's application is submitted and either approved or rejected by the registrar without any needed justification. The exceptional case is that the visitor's application is filled out incorrectly so it is not submitted.

Application Status: Any visitor can check the status of their student or instructor application. The normal case is that the visitor's name is entered and the status is returned. An exceptional case is that the visitor's name does not correspond to any application so a status cannot be returned.

Login: This use-case is accessible to all student, instructor, and registrar users for the purposes of logging in to their respective types of accounts where they will gain access to their respective functions. The normal case is that the user provides a valid username and password. Exceptional cases occur when the user provides an invalid username and/or password.

Class Set-Up Period

course_management: This functionality is accessible to registrars for the purpose of establishing classes during the first period of a semester. There is only a normal case which is that the registrar successfully sets up classes, class dates/times, course instructors, and class sizes.

applications:

Course Registration Period

Search Classes: Matriculated students can search the listings of courses that they can potentially register in. There is only a normal case which is that the listings are available for students to see.

Course Registration: This use-case is available to matriculated students who can register in courses. The normal case is that the student successfully enrolls in 2-4 classes. Exceptional cases occur if there is a time conflict among chosen classes and if the pre-requisites were not taken and passed. If the student enrolls in less than 2 classes, they receive a warning. Another exceptional case is that the upper limit of the courses is reached so the student is placed in the wait-list which only the course instructor can take the student off of. An exceptional case also occurs if the student tries to retake a class they did not previously receive a F in, which is not allowed.

Class Running Period

Class Overview: Registered students can view their currently assigned classes.

View Records: Registered students can view their current courses and academic records (includes completed and uncompleted required courses), teachers can view the basic info and academic records of students in their current classes, and registrars can view all records of all students in all classes. There is only a normal case which is that the records are available for each type of user.

Drop Class: A student can drop a class between the registration and grading period with a grade W. The normal case is that a class is dropped and the student receives a grade W. An exceptional case is that all classes are dropped and the student is automatically suspended for one semester. A similar exceptional case can occur if all classes are dropped but the registrar takes another action.

Cancel Course: If a course has less than 5 students registered, the course is cancelled and the course instructor is warned. An exceptional case is that all of an instructor's courses have less than 5 students registered so they are all cancelled and the instructor is permanently fired.

Special Registration: This is accessible to students who were registered in cancelled classes. The normal case is that the students are given another chance to register in other courses to satisfy the

2-4 class requirement.

Ratings/Reviews: This is accessible to all students, instructors and registrars. A student who is in a class can write a review for that class and assign a 1 to 5 star rating. Instructors can view published reviews anonymously, and only the registrar can view who wrote the reviews.The normal case is that a student writes a review and it becomes part of the class summary. Exceptional cases occur if a student rates every class 5 stars or rates at least three classes only 1 star; the student will be questioned and if found wrong, the rating is removed and the student is warned. If an instructor has an average rating < 2, the instructor is warned. If an instructor accumulates 3 warnings, the instructor is permanently fired. If the student tries to rate the class after the instructor posts the grade, it will not be allowed. If a student's review has 1 or 2 taboo words, the words are changed to * and the student is warned once. If a review has 3 or more taboo words, the words are not shown and the student receives two warnings.

Complaints: This is accessible to all students, instructors, and registrars. A student can complain about another student to the instructor, or about the instructor to the registrar. An instructor can process student complaints, issue a warning, and complain about a student to the registrar. Registrars can process complaints and take any action needed. For student, the normal case is that the complaint is processed and a warning is issued by the instructor or registrar either to the complainer or complainee. For instructor, the normal case is that the complaint is processed and either the student is punished accordingly (warn or de-register) or the instructor is punished with one warning by the registrar. An exceptional case is that up to three warnings are issued to a student so they are suspended for 1 semester and pay a fine to the registrar.

Grading Period

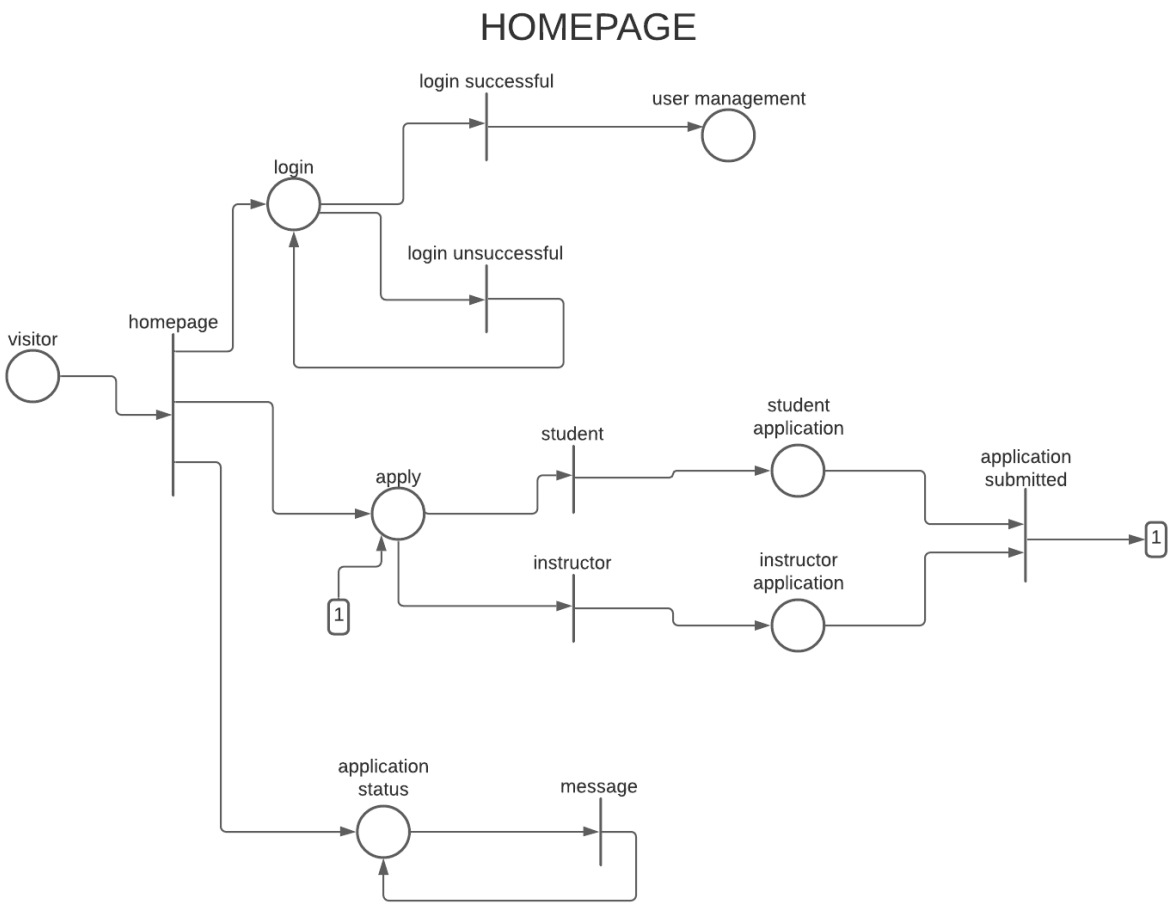
Assign Grades: This is accessible to instructors (and registrars) for the purpose of assigning grades to students during the grading period. The normal case is that an instructor assigns grades to all of their students with a class GPA between 2.5 and 3.5. An exceptional case is that an instructor doesn't assign grades for all students so they are warned. Another exceptional case is that an instructor has a class GPA <2.5 or >3.5, so the instructor is questioned by the registrar and either warned or fired right away if they don't have adequate justifications.

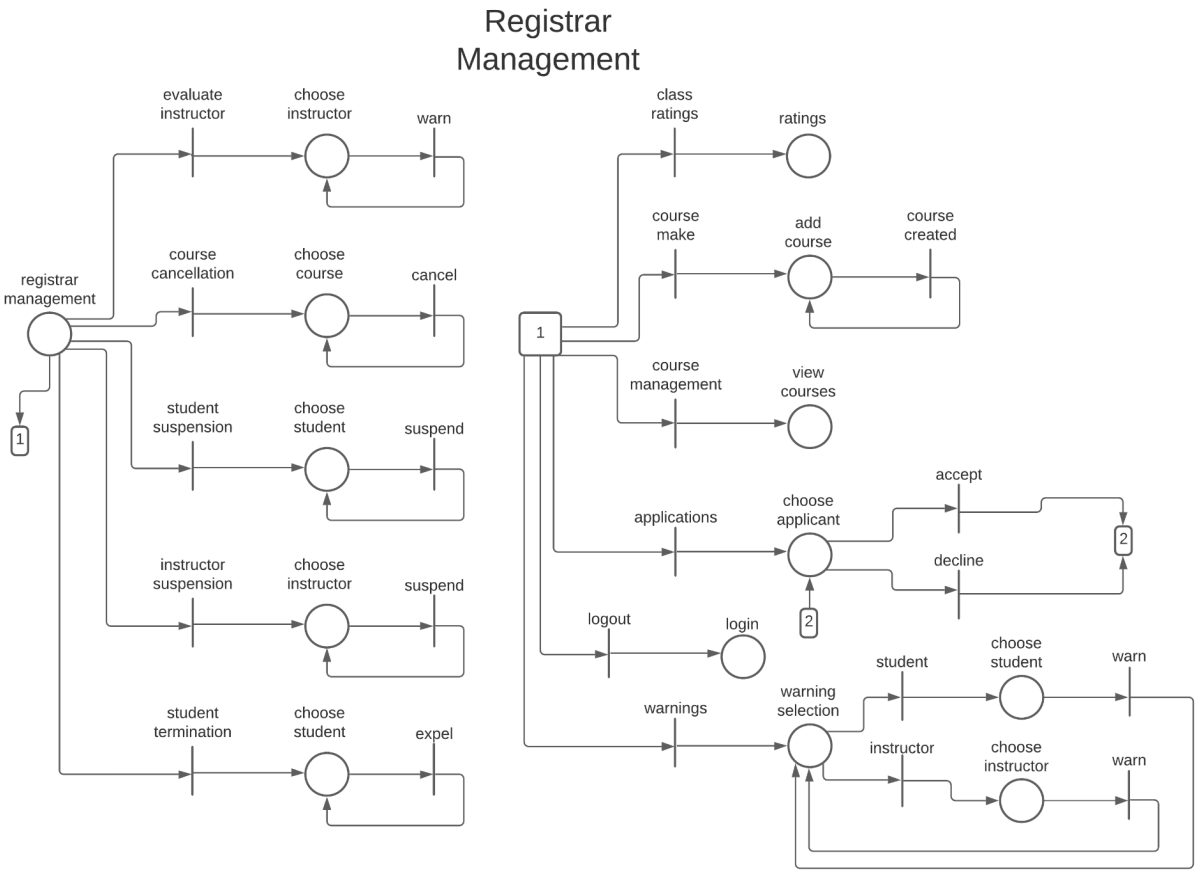
View Grades: This is accessible to students (and registrars) for the purpose of viewing grades assigned by instructors after the grading period ends. The normal case is that a student can see their grades and passing semester and overall GPA's. An exceptional case is that a student has GPA < 2.0 or they failed the same course twice so the student is automatically terminated from the system. Another exceptional case is that a student has GPA between 2 and 2.25 so the student receives a warning demanding an interview with the registrars. An exceptional case that also occurs is if a student has semester GPA > 3.75 or overall GPA > 3.5 so they are labeled as honor roll students automatically, which can be used to remove one warning if there is any.

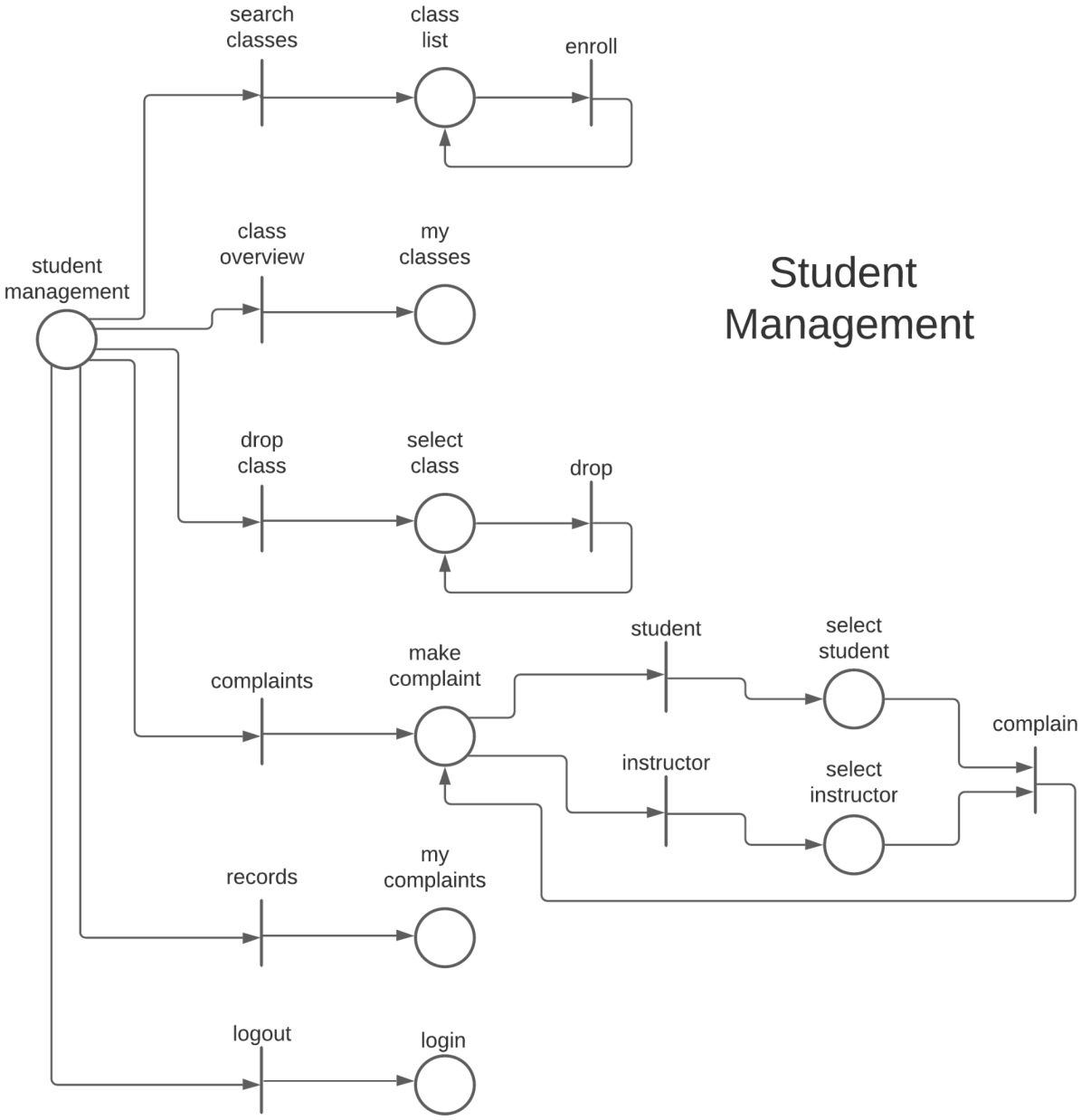
Apply For Graduation: This is accessible to students (and registrars) for the purpose of applying for graduation. The normal case is that a student has completed the required 8 classes and successfully applies for graduation, so the student leaves the system with a Master degree. An exceptional case is that a student applies for graduation but all required courses are not covered, so the student receives a warning from the registrars for a reckless graduation application.

2.2 Collaboration OR Sequence Class Diagrams for Use Cases

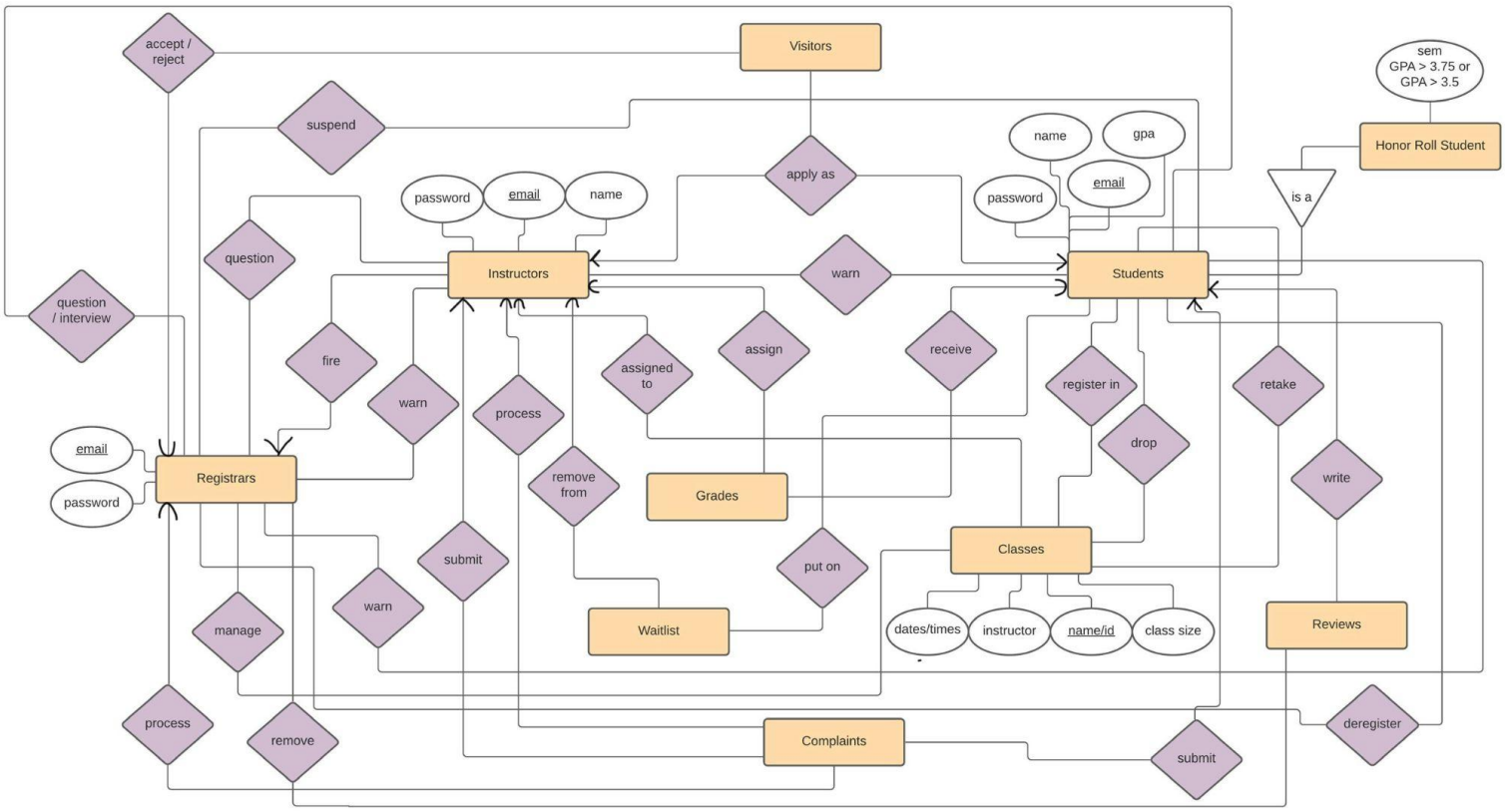
2.3 Petri-nets for Use Cases


HOMEPAGE
login successful
user management
login
login unsuccessful
visitor
homepage
student
student application
application submitted
apply
1
instructor
instructor application
1
application status
message


Registrar Management
registrar management
evaluate instructor
choose instructor
warn
course cancellation
choose course
cancel
student suspension
choose student
suspend
instructor suspension
choose instructor
suspend
student termination
choose student
expel
1
class ratings
ratings
course make
add course
course created
course management
view courses
applications
choose applicant
accept
decline
2
logout
login
warnings
warning selection
student
choose student
warn
instructor
choose instructor
warn


search classes
class list
enroll
class overview
my classes
student management
Student Management
drop class
select class
drop
student
select student
make complaint
complaints
complain
instructor
select instructor
my complaints
records
logout
login

3. E-R Diagram



4. Detailed Design

for EVERY method use pseudo-code to delineate the input/output and main functionalities

input: email and password
used to login the user when they have an account
def login():
    if POST:
        check user email if it is in the system
        if yes:
            if yes:
                logs them into their account
                return (their management page)
            if no:
                warn them of wrong password
        if no:
            flash “email not in registered
    else:
        return login page

input: logout command
used to log out the user from their account
def logout():

```
        will call the log_user() function
        returns to the home page

input: when user clicks on the apply button on drop down
used for people to apply (student and instructor)
def apply():
        if POST:
                checks which button user clicks
                student:
                        directs to student apply
                instructor:
                        directs to instructor apply
        else:
                renders the apply page

input: clicked the student apply button in apply()
used to direct the user to enter in their gpa and name for application for registrar
def apply_student():
        if POST:
                gpa = from user input
                name = from user input
                new user = applications (above info)
                render same page
        else:
                renders same page



input: instructor clicked the apply button in apply()
used to direct the user to enter their subject they want to teach and their name
def apply_instructor():
        if POST:
                subject = from user input
                name = from user input
                new user = applications (above info)
                render same page
        else:
                renders same page

input: users would click on application status on top drop down
used to check the user's application status by name (but we can add a code they write in the apply
        so it is more secure and others who know their name cannot make their account for them.
def application():
        if POST:
                try:
                        searches for name
                        orders by name
                        create a session so the user type can be passed to another function
                        show the status
                except:
                        flash "no application"
        else:
                render application status page
```

input: when user clicks on signup on top of drop down, will ask for email, first name, password, and confirm password
used to sign up users that already had their application approved
def sign_up():
    if POST:
        email = from user input
        first name = = from user input
        pass = from user input
        pass confirm = from user input
        user type = in application a session was created (will grab from there)
        check if email already in system
        if yes:
            flash “email already exist”
        else:
            creates account (User)
        checks (user type)
        if student:
            return student management page
        if instructor:
            return instructor management page
        else:
            return registrar management page

    else:
        return sign_up page

input: when you click courses on the top drop down
used to show the courses page
def courses():
    shows courses currently in the system
    return courses page

input: click
used to show the course management page
def course_management():
    return show course management page

input: click
used to show the student view page
def student_view():
    return student view page

input: either accept or decline button
used to accept or reject applications - located in registrar drop down tab
def applications():
    shows all the applications that are still pending (students first and then instructors)
    if accept is pressed:
        removes them from list by changing status
        status = 0
        return same page but with updated list
    if reject is pressed:

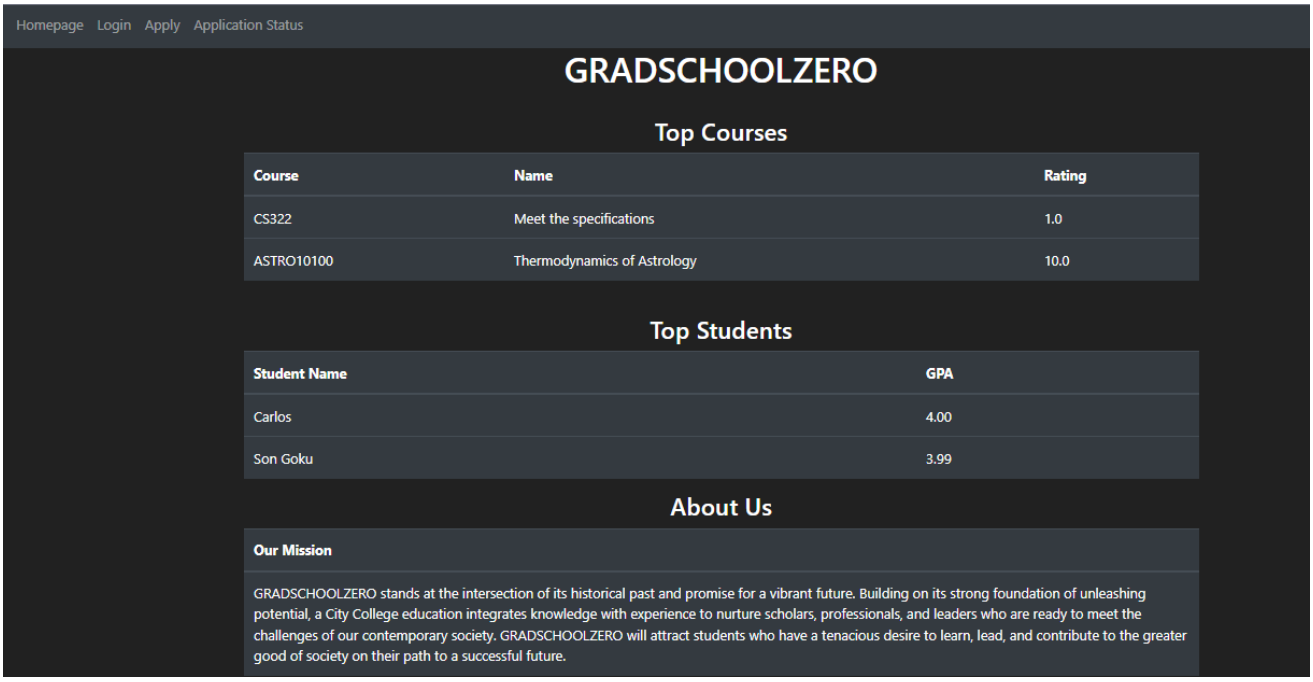
```
                removes them from list by changing status
                status = 1
                return same page but with updated list
        return applications page with current list

input: warning tab click
used to display all the “warnings” from other users directed to them
def warnings():
        returns warning page

input: button clicks for each corresponding button in student management
used to see what button the user clicks to direct them to the right url
def student_management():
        if POST:
                (buttons) linking to corresponding
                        return page
        else:
                return student management page

input: click
used to check over the overview and details regarding the classes
def class_overview():
        if POST:
                shows specific class that was clicked on
                return page with information displayed
        else:
                return class overview page


input: click
used to file complaints
def student_instructor_complaint(): → also same with def student_student_complaint()
        if POST:
                text = what user types in box
                submits text by creating new db entry
                flash status of complaint
                return student_instructor_complaints page

        else:
                return student_instructor_complaints page

input: click
used to view warnings that the student gets
def student_warnings():
        if POST:
                will show the warning you clicked on
                return page containing the complaint
        else:
                return student warnings page

input: click
used to show the students records (this includes the classes they are taking right now and classes
        they already took
def records():
```

```
        will show the students records in single html file
        return records page

input: logged in as an instructor
used to show options and controls instructors have
def instructor_mangagement(): → same with def registrar_management()
        if POST:
                (buttons) linking to corresponding
                        return page
        else:
                return instructor management page

input: click
used to grade the students in his class
def grade_students():
        if POST:
                will select which student to grade
        else:
                return grade students page

input: click
used to show all the courses either student or instructor
def my_courses():
        shows all the course
        return my courses page



input: click
used to show students records (grades and classes taken)
def student_records():
        shows student's records on a page
        return records page

input: click
used to show instructor's warnings on their page
def instructor_warning():
        shows all existing warnings
        return instructor warning page

input: click
used to evaluate an instructor (done by the registrar given the info of class and students complaints
def evaluate_instructor():
        if POST:
                select instructor given button
                text = entered text
                new = complaint(above info)
        else:
                return evaluate instructor page

input: click
used to remove courses in the course list
def cancel_course():
```

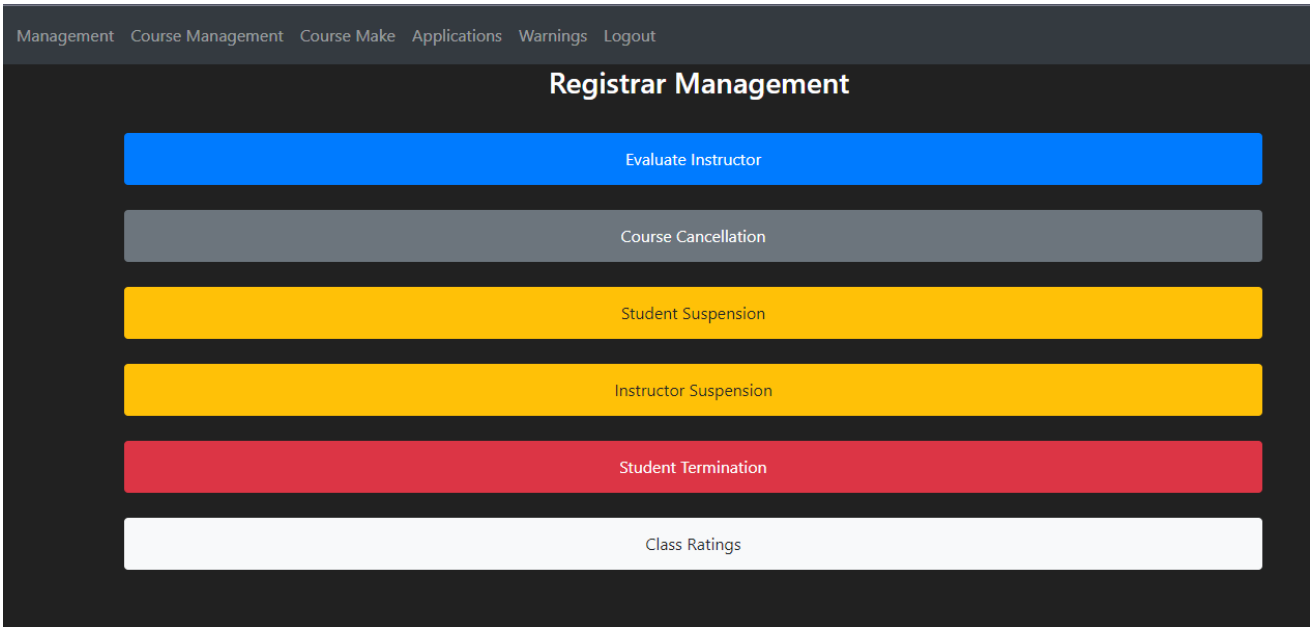
```
        if POST:
                remove class
        else:
                return cancel course page

input: click
used to remove a student from the school (giving them suspension)
def student_suspension(): → same with def instructor_suspension
        if POST:
                student = id of the one clicked
                time them out of class / semester
        else:
                return suspension page

input: click
used to Terminate a student (meaning they cannot apply no longer for classes) registrar
def student_termination():
        if POST:
                will select student for termination
                blacklist = student name appended
        else:
                return student termination page

input: click
used to see class ratings
def class_ratings():
        show all classes by passing info of classes into html
        return html of the class rating page
```

5. System Screens

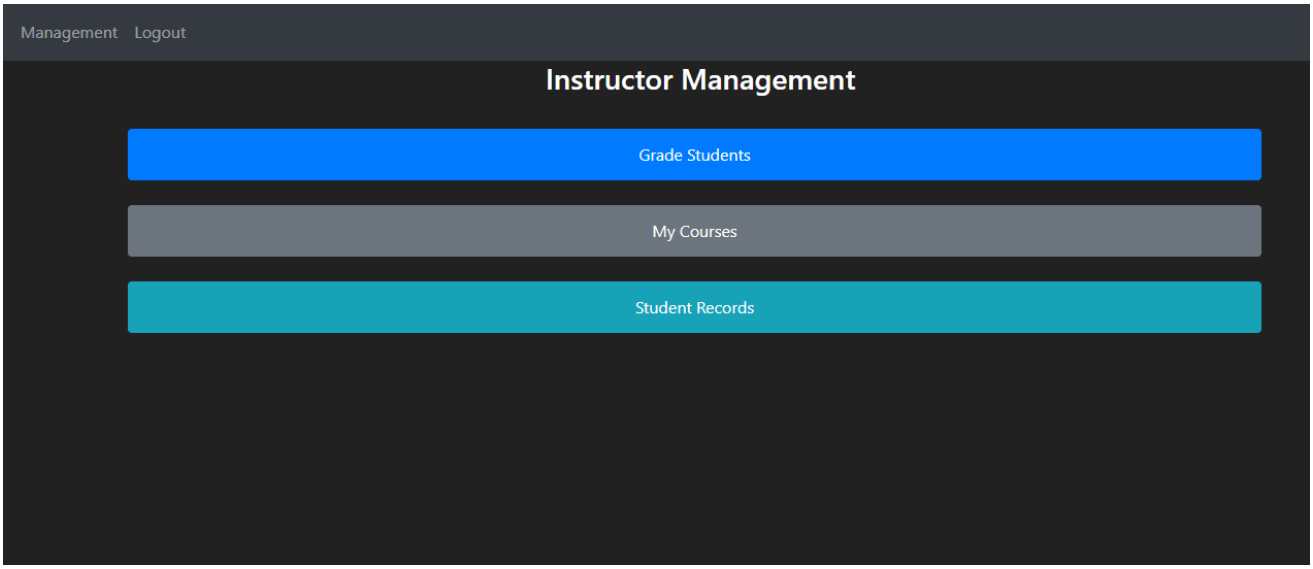
5.1 Major GUI Screens



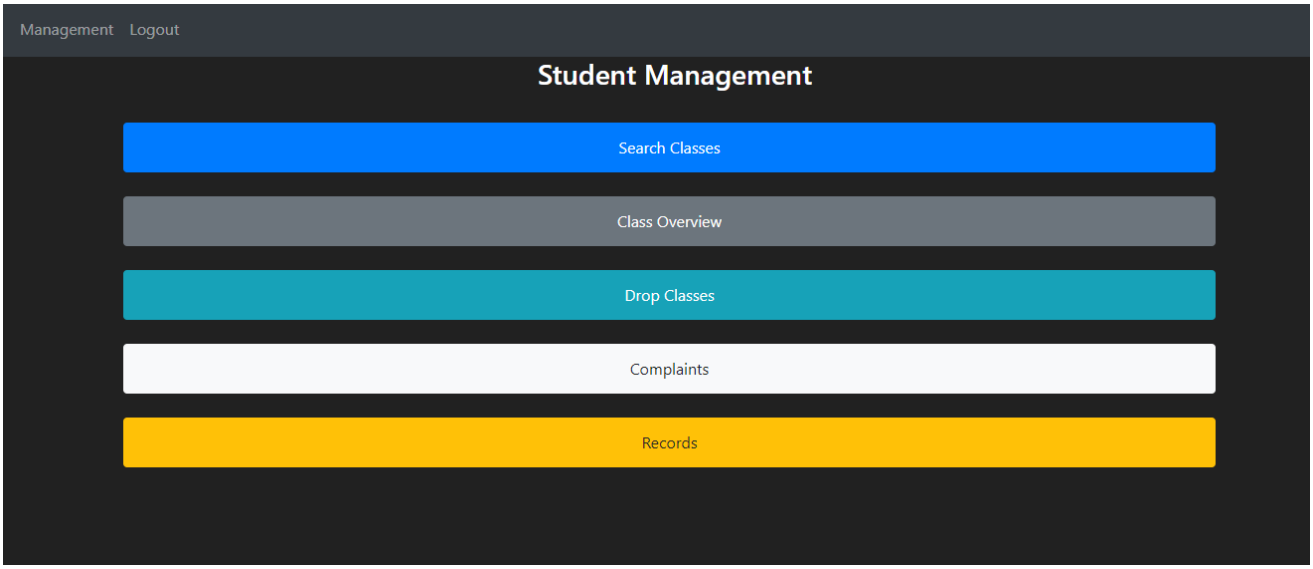
Landing page of the website where users can learn information about the school as well as navigate options to apply, see their application status and login.



Registrar main page after logging in. This page includes links to all the options the registrar has for the school management.

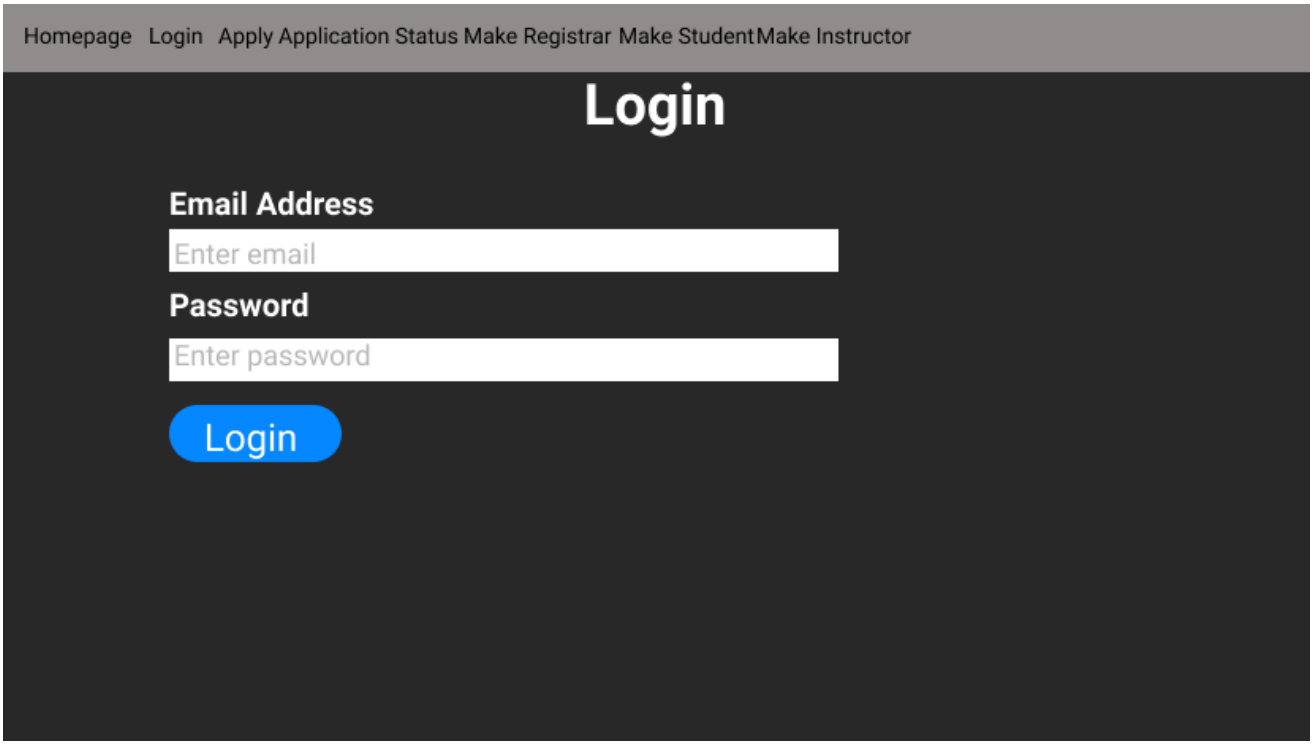


Instructor main page after logging in. Similarly to registrars, this page possesses all the available functions that the instructors have.



Student main page after logging in. Similarly to previous pages, this page possesses all the available tools that the student can choose from.

5.2 Sample Prototype



6. Group Meeting Memos

The first meeting that we had as a group was in the very beginning of the semester when the project description was finalized by the professor. We then met to socialize and establish bonds to further figure out what language and platforms we wanted to use for the project. We opted on using Python and HTML as the major components of our program as we were for the most part familiar with them, hence a common ground. Furthermore, the next meeting we had was to divide the phase 1 report to see what each person would do. We decided on the following:

Carlos & Willie: Section 2
Steven: Section 4
Atiya: Section 1 and formatting
Shahid: Section 3

Further, we had about 3 more meetings in which we discussed and analyzed the progress we have made so far. Later on, report 2 was assigned, resulting in our next meeting, where we discussed and assigned specific parts of the report, as present in the revision history section of this document.

Some general comments that are worth discussing is that throughout the whole process we were in constant communication on our group chat and if anybody needed any type of help we would all try to provide assistance. If we were unable to solve the problem through the chat, then we would hop on a quick voice call on either zoom or discord, sharing screen as needed to resolve the problem. The main concerns regarding this project that we have realized is that not all of us are free at the same time, and unfortunately we all have jobs and are taking hard classes, including but not limited to: CS 33500 which has a really bad reputation for being extremely hard. With that in mind, the current state of the project might be a bit delayed from where we would want it to be, but we are hopeful that with more grind in the remaining weeks, we can hopefully finish and have a good, working project to present.

7. Github Repository Address

Address to Team A's Github Repository: https://github.com/shahidemdad/GradSchoolZero