CCNY ZERO

Phase II: Design Report

For Web Application

Version <1.0.1>

Revision History

Date	Version	Description	Author
24/OCT/21	1.0	Phase I – Software Requirements SpecificationNOV	Haroon Syed, Andrew Persaud, Josue Flores, Tufayel Ahmed, Christopher Lall
13/NOV/20	1.0.1	Phase II – Design report & updated UseCase Diagram from Phase I	Haroon Syed, Andrew Persaud, Josue Flores, Tufayel Ahmed, Christopher Lall

Table of Contents

©Questionable, 2021

Online Graduate Program	Version: 1.0.1
Software Requirements Specification	Date: 16/NOV/21
Phase II Report	

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Use Cases

2.1 Use-Case Model Survey

2.2 Use-Case Reports

3. E-R Diagram

4. Detailed design

5. System screens

6. Group Meetings

7. Github Repositories

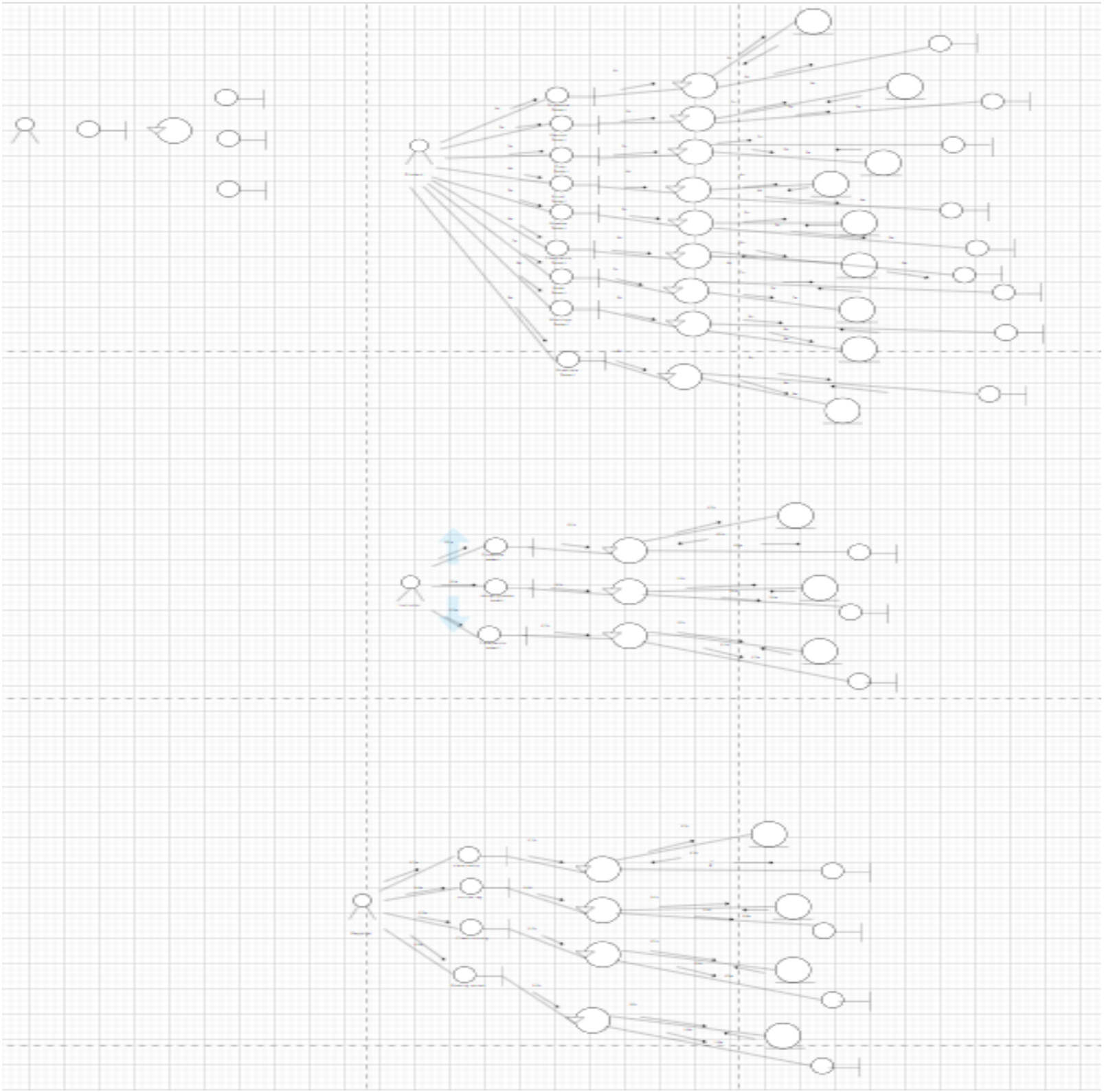
Design Report & updated Use-Case Diagram from Phase I

1. Introduction

Collaboration diagram:

https://drive.google.com/file/d/17vW-DEhoitJXzhbKTTD8qJ1GEEe_Mr2U/view?usp=sharing

Online Graduate Program	Version: 1.0.1
Software Requirements Specification	Date: 16/NOV/21
Phase II Report	



1.1 Purpose

The Graduate Education System (GES) will be fully described in this software requirement specification. This report will contain the following components: System Requirements/Specifications, Use-Case Models and Detailed description of use cases, Petri Net Diagram for each use case, ER Diagram for all entities and their relationships within the database, and pictures of the System screens for the frontend of our website can all be found in this document.

1.2 Scope

The Class Enrollment Program (CEP) will be a web based application where new or old users can register for classes.

This GES includes two case diagrams: the first use-case diagram describes the student functionalities (register, manage the course, check grades). The second use-case describes the faculty class: registrars and instructors. Each faculty has its separate specifications according to the task description provided. This document includes the software requirements such as frameworks that will be used, as well as relevant dependencies used.

1.3 Definitions, Acronyms, and Abbreviations

Terms	**Definitions**
ReactJS	React is a front end JavaScript library for building user interfaces maintained by Facebook.

NodeJS	Javascript runtime environment that runs outside the web browser
Google Firebase	Database management system
Javascript (JS)	JS is an object-based, object-oriented programming language.
GES	Graduate Education System
CEP	Class Enrollment Program

1.4 References

Software Requirements Specification, www-cs.ccny.cuny.edu/~csjie/322/spec_sample.pdf.

Phase II: Design report, https://www-cs.ccny.cuny.edu/~csjie/322/design.txt

1.5 Overview

- The “Use Case” section of this document illustrates the GES features of the class enrollment system application and its accessibility for different types of users. The “Use Case Model Service” goes into detail about each type of user permission and privileges. It contains a complete overview of the user's scope throughout the application. The GES diagram is also included as a top down view of the system as a whole through use-case diagrams which link different types of users to features in the GES.

The “Use Case Reports" section of the document illustrates an overview of the system and its relation to users. It includes the Class Diagrams and Petri-Nets for each of the use cases. There are normal and exceptional scenarios containing a brief overview of the cases.

The “ER Diagram” section contains the attributes and keys for each class encompassing the entire system.

The “Detailed Design” section contains critical methods from services and controllers in the application. For simplicity and readability, pseudo code is displayed instead of the code used in the application.

Finally, the “Screenshots” section displays some progress with the frontend. All events are temporarily hardcoded. At the conclusion of the use cases, the data from the database will be used to populate the fields.

2. All Use Cases

The GES features an online graduate education system that is accessible by different types of users. Visitors have the ability to browse basic information about the program such as highest rated classes, lowest rated classes and students with the highest GPA. Visitors also have the ability to sign up to become a student or an instructor. Failing to meet the college’s requirement will end up with a rejection by the registrars. However, accepted students will receive a unique id and must create a new password upon first login. Accepted instructors will be assigned classes by the registrar. Students and instructors can see all of their basic information on their own page. Students can see their records while instructors can see the academic records of the students in their course. The registrar can see everything. The school system goes through four periods in a semester which is all monitored and managed by the

registrars. The first period is the class set-up period where registrars will be creating classes and providing the necessary information for all of those courses. The second period is the course registration period where students are able to sign up for classes. Students can only register between 2-4 courses and will be accepted if there is no time conflict or is not exceeding the student limit in that course. Students also have the ability to retake the same class if he/she got an F before. The third period is the class running period where students can't register for any more courses unless they are given another chance to choose other courses which are special circumstances. These circumstances are if the instructor course was cancelled and they were in that course. Students must have a minimum of 2 courses or they will be warned. Courses with less than 5 students will be cancelled and the instructors of those courses will be warned. If all the instructor's courses are cancelled, he/she will be suspended and will not be able to teach next semester. The last period is the grading period where instructors will be assigning grades. Instructors need to assign grades to all the students or they will be warned. During the registration and grading period, if students drop any course, they will receive a W as their final grade. Dropping all courses will result in a suspension for one semester. Class GPA of 3.5> or <2.5 will be questioned by the registrars. Without proper justification, the instructor will be warned or fired right away. Students with GPA below 2 or failed the same course twice will be terminated from the system. Students with GPA higher than 3.75 will be labeled as an honor student and have the ability to have one warning removed from the system. Students who have completed 8 courses can apply for graduation and receive their Master degree. False application will receive a warning. The graduate education system also features a review and complaint system. Students can write reviews of the course and rate it from 1 through 5. Students will have to rate the course before they receive their grade. Registrars are the only people to see who the author of that review is. Instructor's courses whose reviews are lower than a 2 rating will be warned. Three warnings

result in a suspension. Authors who write 1 or taboo words will receive warnings and the taboo words will be censored. However, 3 or more taboo words will not be shown in the system and the author will receive 2 warnings. The complaint system is designated for both the student and the instructor. Students can complain about other students and instructors to the registrars. Based on the investigation, the registrars will either warn the student or the instructor. An instructor can complain a student to the registrar to either warn or deregister the student. The registrar will then choose the punishment for that student accordingly. Any students with 3 warnings will be suspended for 1 semester and must pay a fine to the registrars.

2.1 Use-Case Model Survey

In our use case diagram, we have 2 types of ordinary users and 2 types of privileged users. The ordinary users are the Visitors (users with no account) and Students (users with account). The privileged users are Instructors and Registrars.

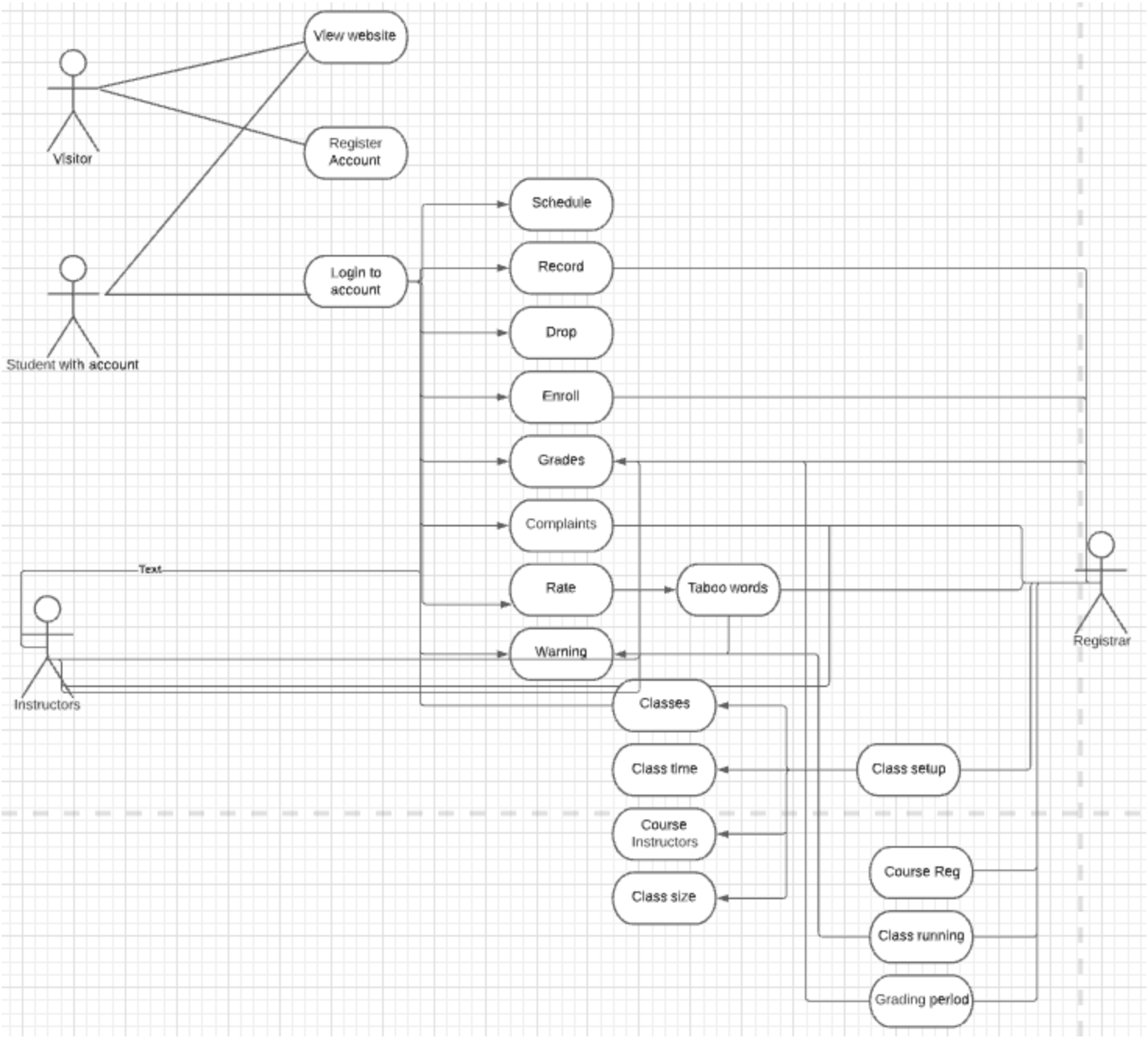
Visitors: Visitors are users with no accounts and can apply to become a student or an instructor at the college. Visitors can also see the general introduction of the program which also contains the highest rated classes, lowest rated classes and students with the highest GPA.

Student: A student is a user who is new to the website and is looking for courses to enroll into . Students can browse and view the website criteria listed just like visitors but also enroll into courses with their created account. Once the enrollment is successful they have to confirm that they are in the class. They can rate both

the teacher, and the class from 1 to 5 (1 worst to 5 best). Students can write comments with their reviews and if they write words that are detected as taboo (1 or 2) then the author receives one warning. Reviews with more than 3 taboo words are not shown in the systems and the author will receive 2 warnings. The student cannot rate the class after the instructor posts the grades. 3 warnings deregister a customer.

Instructor: The graduate education system can have many instructors. Each instructor has its own page where he/she can see all of their basic info and the academic records of their students in his/her current classes. An instructor can also complain about their students to the registrar who can then warn or deregister the student. Instructors can also be suspended if all of their courses are cancelled and won't be able to teach next semester. Instructors can also be suspended if they accumulate three warnings which can be obtained by receiving an average rating of <2, canceling a course or not assigning grades for all of their students.

Registrars: The registrar has access to everything. They can accept or reject student or instructor applications. However they must provide a justification to the applicant if he/she is a student. Registrars also manage the four periods that a semester goes through. During the class set-up period, they are responsible for setting up classes, class time, course instructors and class size. Registrars also have the power to view the reviews from anyone. They can also approve/reject students that apply for graduation. Registrars are in charge of processing complaints from students or instructors. They can either issue warnings to students or instructors. They also have the power to suspend students and instructors.


View website
Visitor
Register Account
Schedule
Login to account
Record
Drop
Student with account
Enroll
Grades
Complaints
Text
Rate
Taboo words
Registrar
Warning
Instructors
Classes
Class time
Class setup
Course Instructors
Course Reg
Class size
Class running
Grading period

2.2 Use-Case Reports

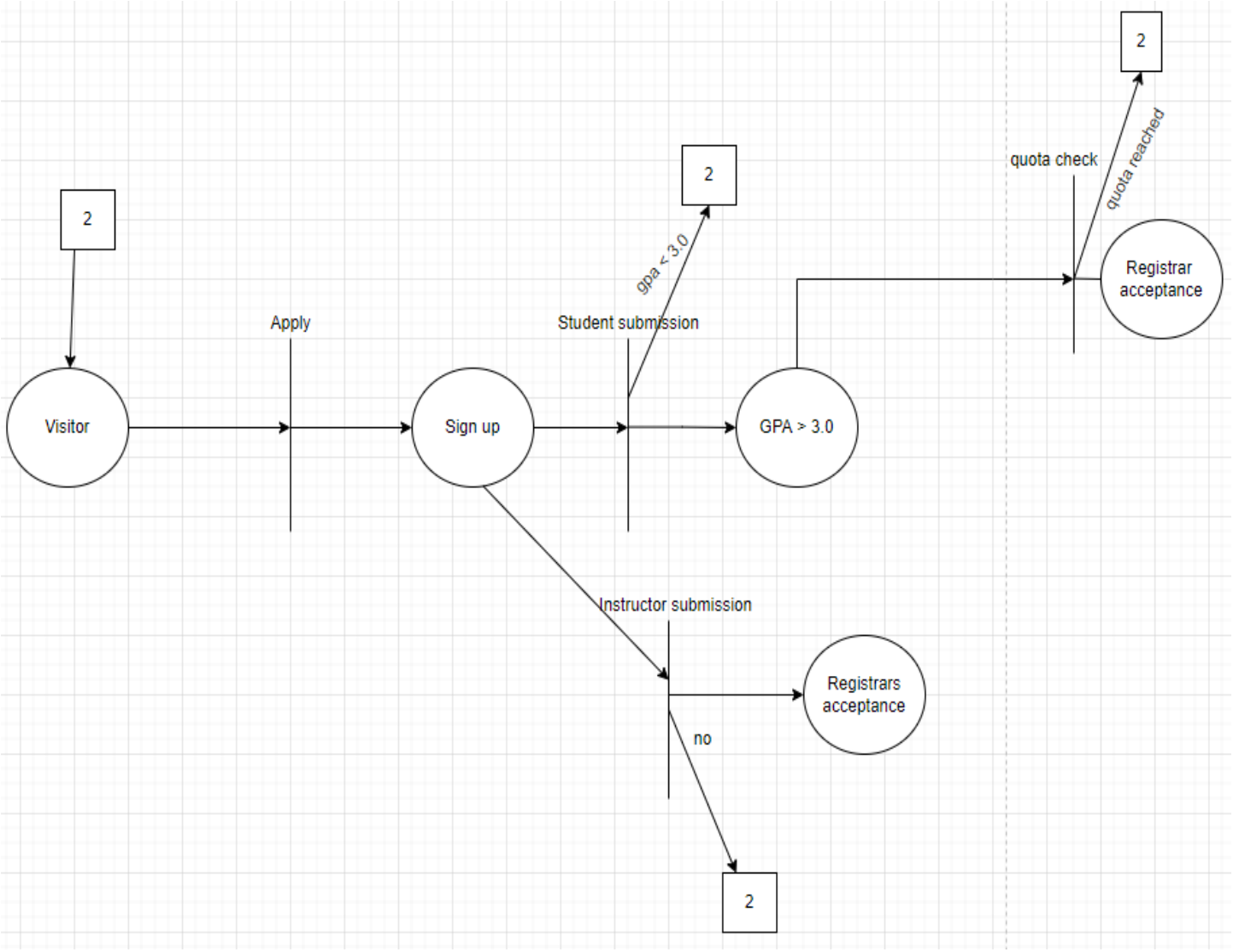
2.2.1 [Visitor Use Case]

Use-Case: Apply

Description: Visitors can apply to attend the graduate school program as an instructor or student by filling out the form.

Normal Use Case: Visitor selects “Not Registered, Create an Account” option and a sign-up form appears on the screen. The visitor is then prompted to enter information into their respective form fields: First Name, Last Name, GPA, Date of Birth, Email, and password. Afterwards, they are prompted to select a radio button to enroll into the program as a Student or an Instructor. Once this is selected, the visitor needs to select the enroll button. If the program quote is not reached, the visitor is automatically accepted as a student if they meet the graduate program’s requirements:

Exceptional Scenario: Program quota is reached. At this point, the user is rejected from the program regardless of their GPA. In the event the visitor’s GPA is too low, < 3.0, the form will not allow them to submit their application for the program.



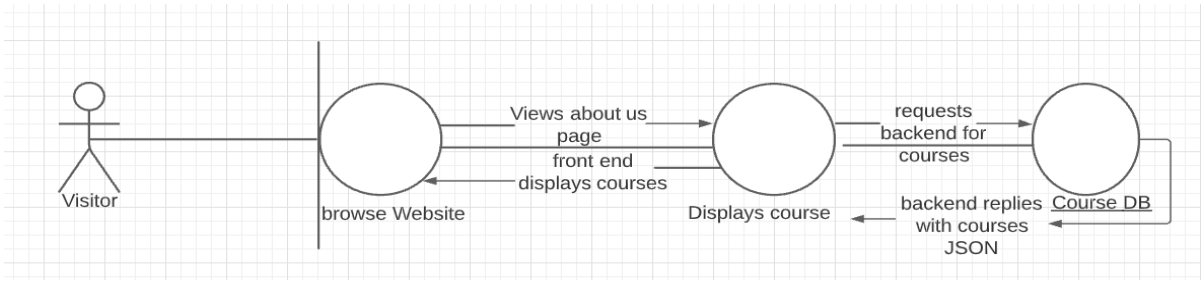
Petri net for Use case-Apply.

Use-Case: About Us

Description: Visitors can visit the About Us Page.

Normal use case : Visitor can observe the basic information for the graduate program: the college, view the program overview video, the top students with their associated GPAs, and the Best/Worst classes based on their ratings.

Exceptional Scenario: None



2.2.2 [Students Use Case]

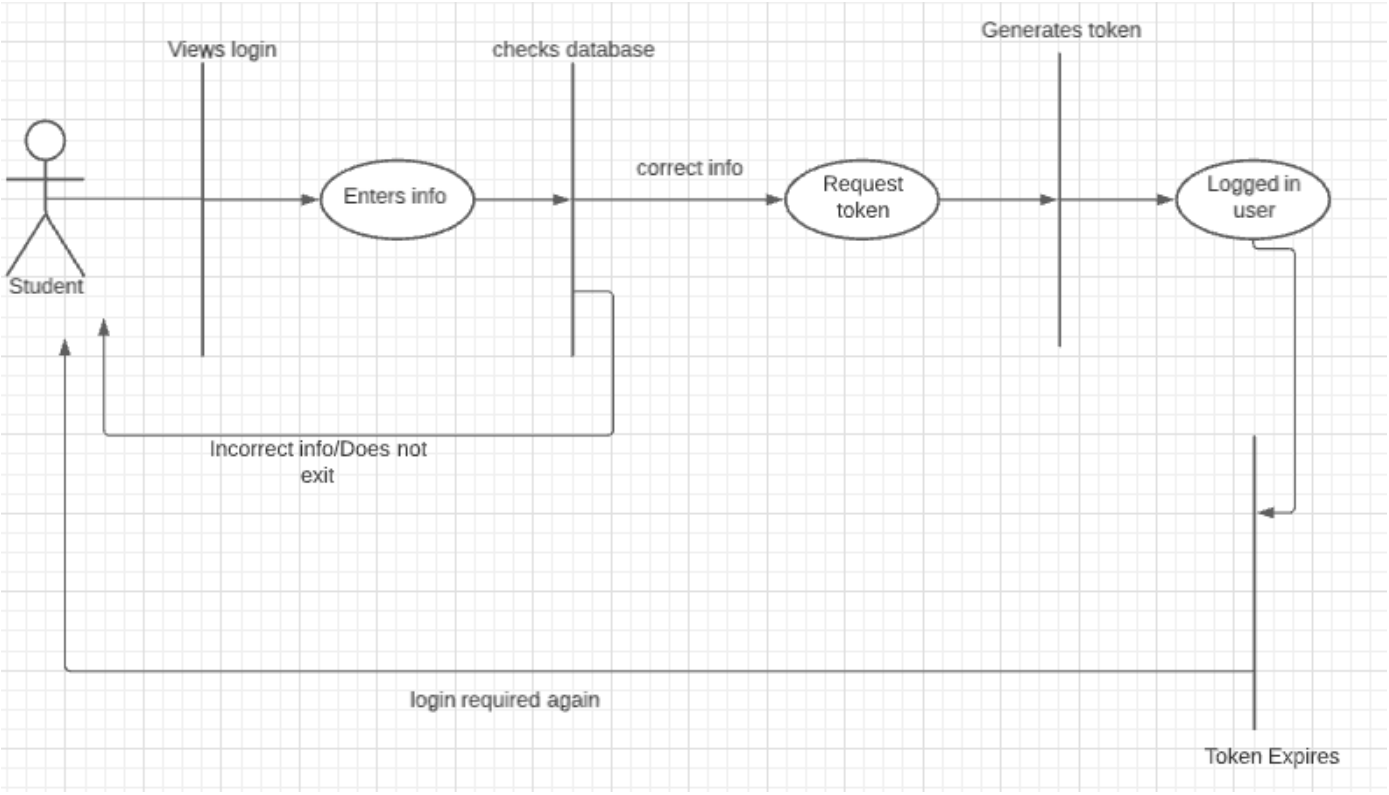
Use-Case: Login

Description: Students are able to sign into the graduate program via the Login Portal on the home page.

Normal Use Case: Students enter their credentials within the sign-in page in the corresponding form boxes and click the Login button. If successful, the student is granted access to the student view of the website. Otherwise, an error message appears saying “Account information was entered incorrectly or the account does not exist.” In the event the student is currently awaiting admission into the program, an alert will appear on the screen stating that the application is still under review.

Furthermore, if a student is rejected by the Registrar, they will receive an alert whenever they try to login containing a message that entails their rejection from the program with the reason being stated. In all unsuccessful cases, they will remain on the same sign-in page.

Exceptional Scenarios: If the program quota is already reached, their application is automatically rejected by the system.



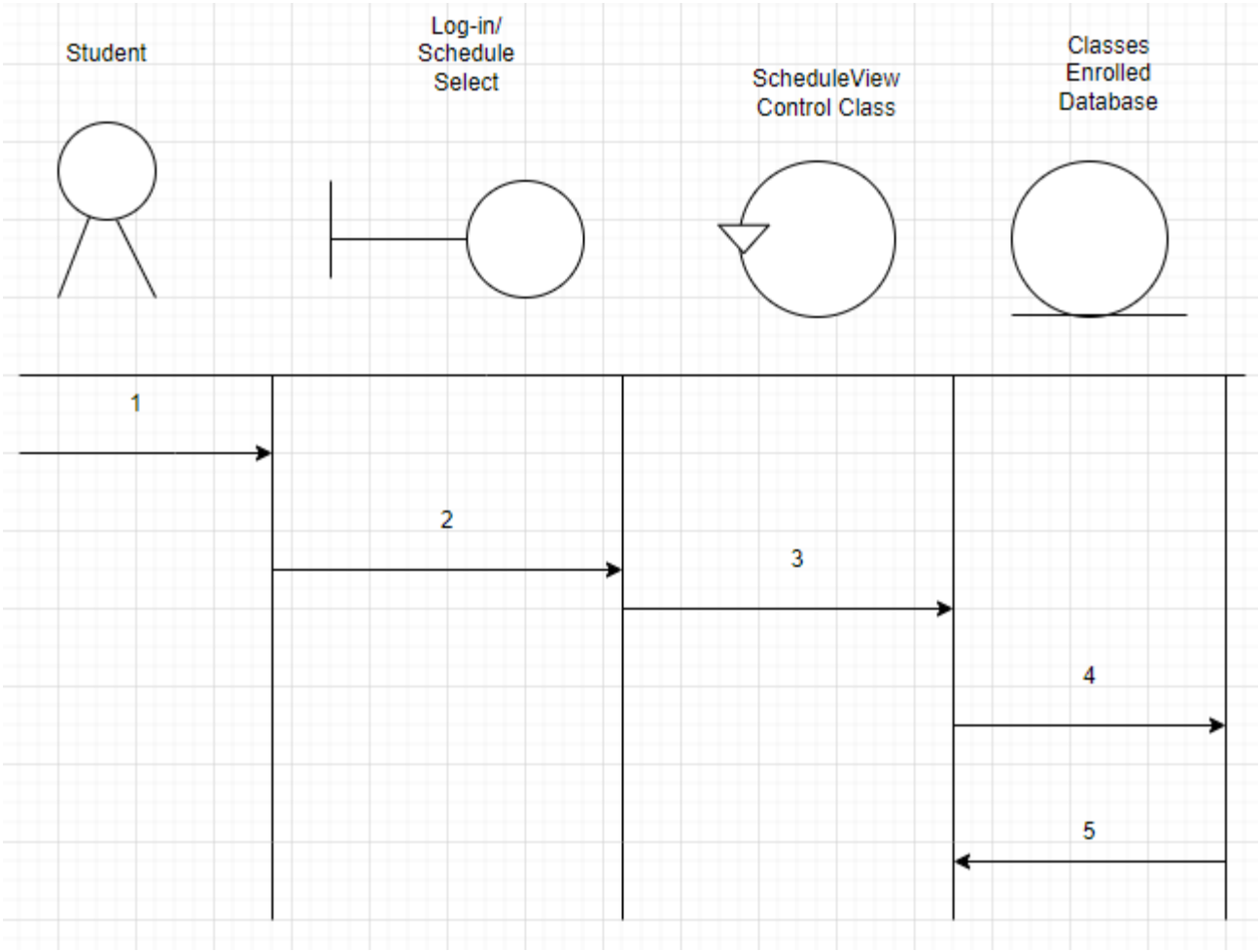
Use-Case: Schedule

Description: Students can see their current semester schedule composed of the classes where they are currently enrolled in the roster.

Normal Use Case: This is the default view for the students after they log-in. Within the largest UI window to the right, students can see a table of their currently enrolled courses. The headers of each section of the table include: Course, Instructor, Room, Days/Time, Section. As one could surmise, the

course information for each class would appear within the respective row entry, with each field being aligned with the corresponding column of the headers. For example, CSC 322 would appear underneath the course header.

Exceptional Cases: The table will not display any information if the student has not enrolled in any courses or the student is currently suspended, which occurs when they incur 3 warnings to their account.

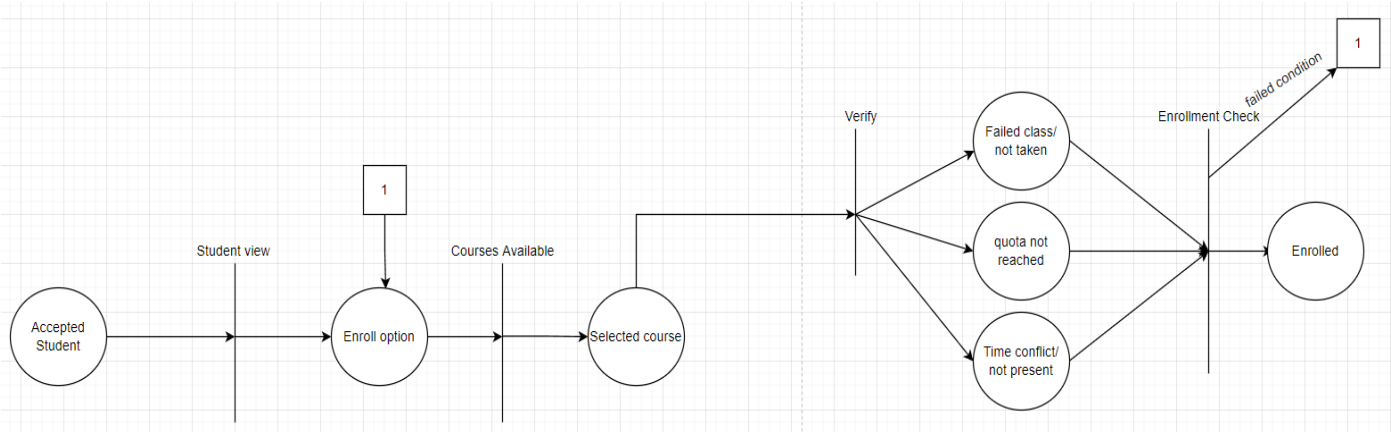


Use-Case: Enroll

Description: Students are able to register/enroll for classes for the current semester during the course registration period.

Normal Use Case: Students are able to register for courses by selecting the enroll option from the drop-down menu. After this option is selected, the corresponding available courses appear within the largest UI window to the right of the screen. At this point, the students are granted the option to select and enroll into 2-4 courses given no time conflict exists between the currently enrolled courses and the class enrollment quota for the course is not fulfilled. In the event of the latter condition being realized, students are granted the option to join the wait-list for the respective course.

Exceptional Scenario: Students cannot retake courses they have completed in the past. The only instance that allows them to retake a course is the student previously failing the course with a grade of F. Furthermore, students who do not enroll in at least 2 classes by the end of the course registration period (including a special short registration period) will receive a warning towards their account.



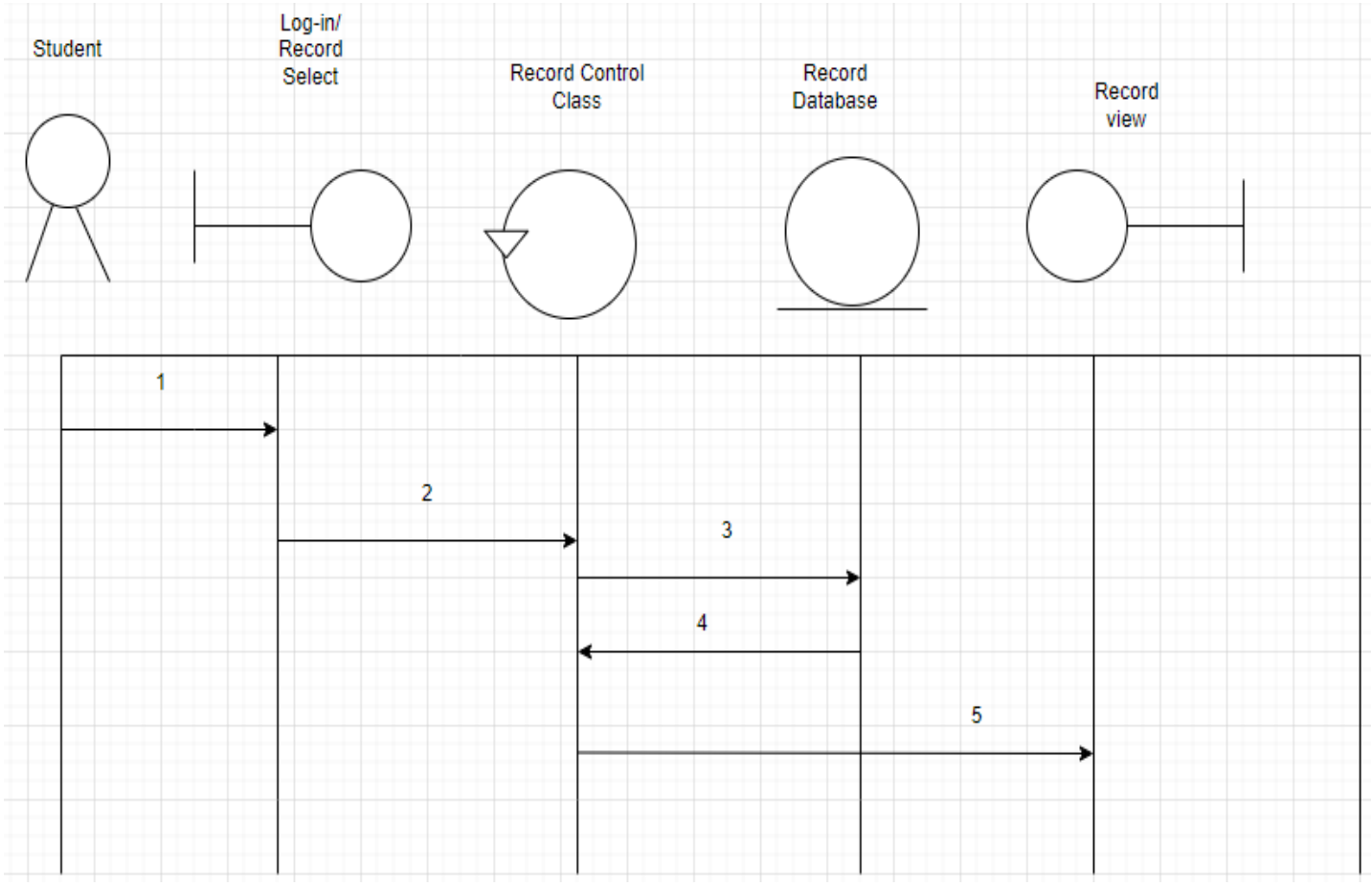
Petri net for Enroll use case.

Use-Case: Record

Description: Students can see their past course history from their graduate program career

Normal Use Case: Students select the record option from the drop-down menu and the student's corresponding academic record appears in the largest UI window to the right of the screen. Within this record, the students will be able to see the name of the previous courses, their grade in the course, and the name of the instructor for that respective course.

Exceptional Scenario: The table contains no information when the student has yet to complete/drop any classes past the running period.



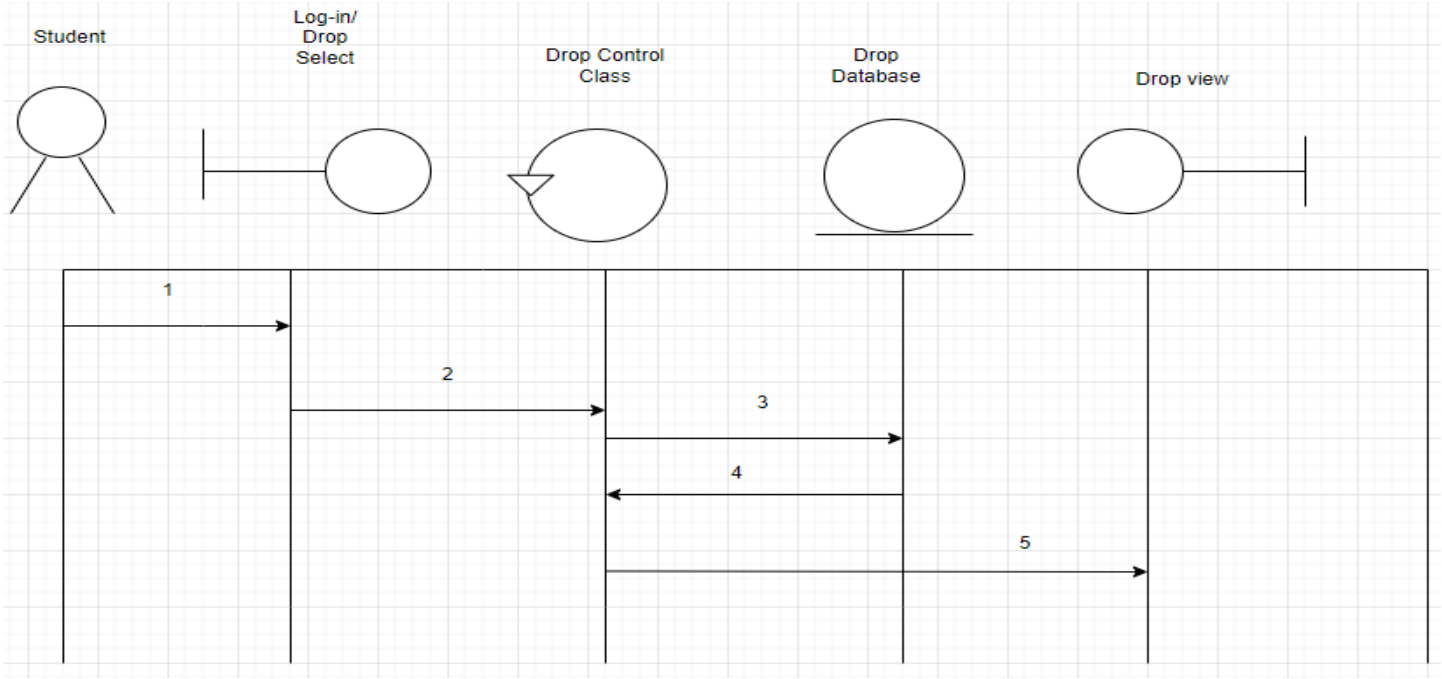
Petri net for Use-Case Record

Use-Case: Drop

Description: Students can drop a course after they are enrolled in the roster for the course.

Normal Use Case: Students select the Drop option from the drop-down menu and the Drop UI appears in the largest window to the right of the screen. Within this container, students will see their currently enrolled course information within a table and be provided a “Drop” button corresponding to each entry. When this button is selected, the student will be removed from the roster of the course and that entry will disappear from the table in the Drop UI. In addition, this dropped class will not show in the current schedule for the student, but it will also appear in the student’s academic record if the course is dropped post-Registration Period.

Exceptional Scenario: Dropping a course, pre-running period, would simply remove the student from the roster and remove the course entry from the student’s schedule.

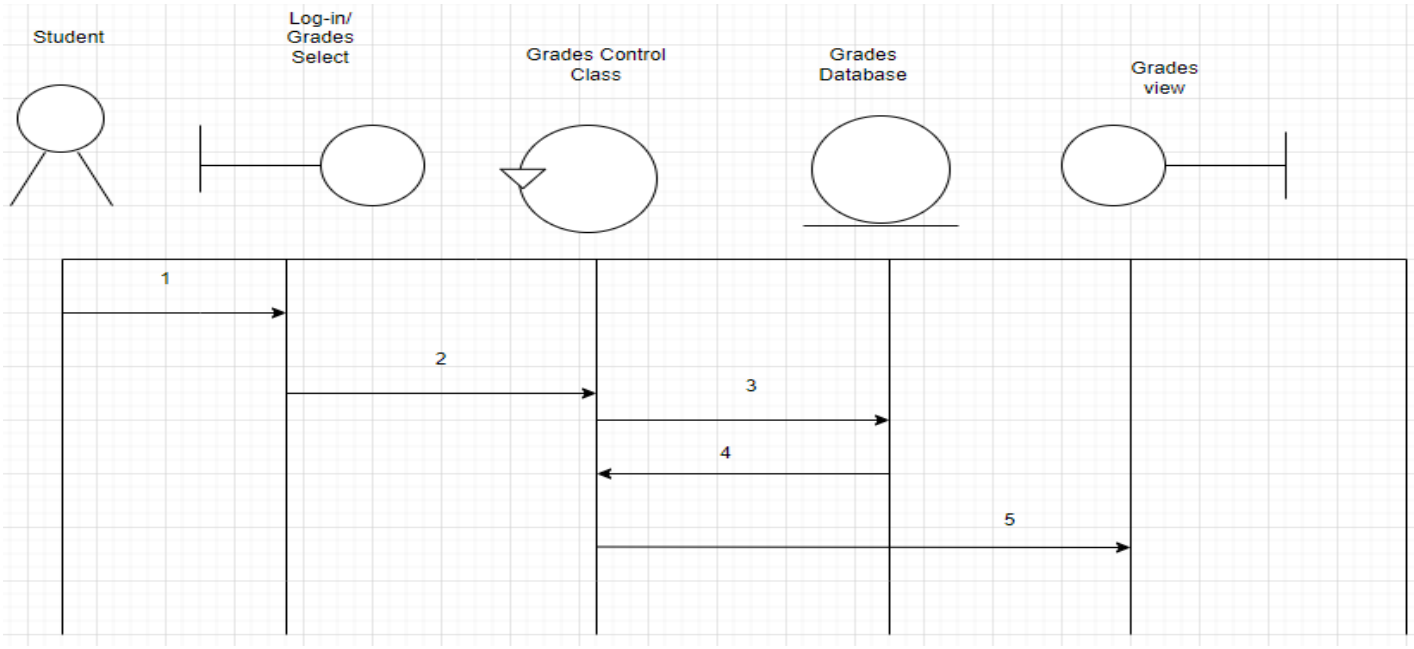


Use-Case: Grades

Description: Students can see their respective semester course grades once the grades are assigned by their corresponding instructors.

Normal Use Case: Students select the Grades option from the drop-down menu and the Grade UI appears in the largest window to the right of the screen. Within this container, students will see their current semester course information and the currently inputted grade by the instructor within the table. The field containing the grade should be empty until the grade is entered by the instructor during the grading period.

Exceptional Scenario: If a student drops a course post-course registration period. Students whose semester GPA is above 3.75 or overall GPA(>1 semester) higher than 3.5 will be labeled as honor roll students.

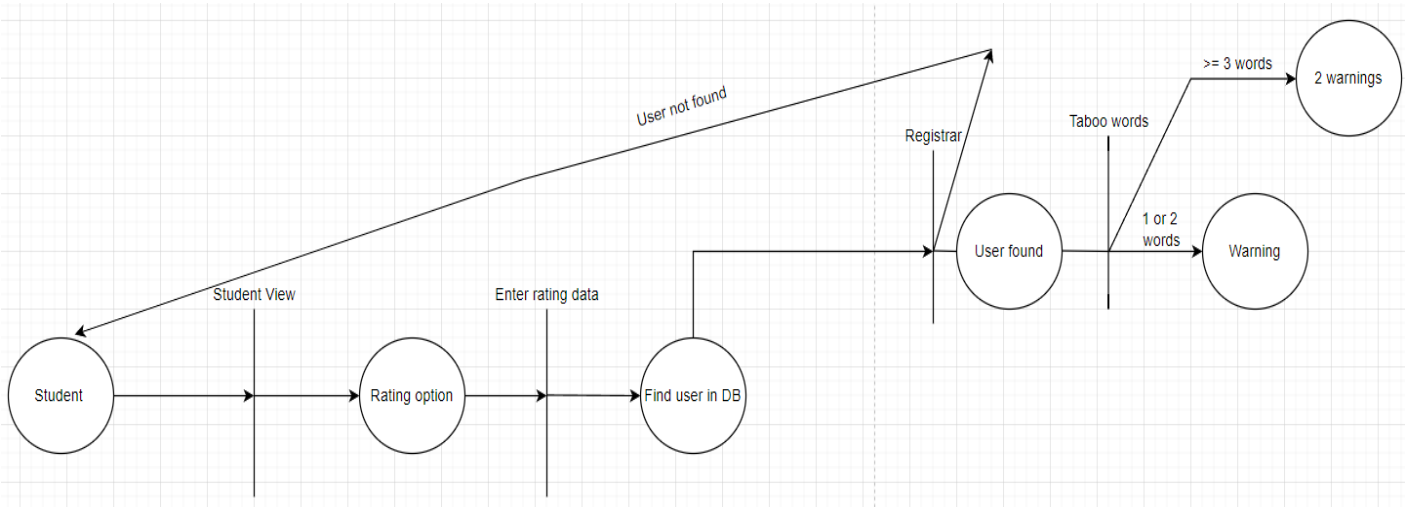


Use-Case: Rating

Description: A student has the privilege to leave a rating about the class they are taking.

Normal Use Case: Once a student logs in, they will be able to see a rating option within the dropdown on the left. Once clicking on this option, the large UI window on the right side should show 5 stars (for the student to rate 1-5), the name of the class, and a text box for a student to fill out and get into detail.

Exceptional Cases: The author of the taboo written review will receive one warning whereas reviews with >= 3 taboo words are not shown in the systems and the author will receive 2 warnings.

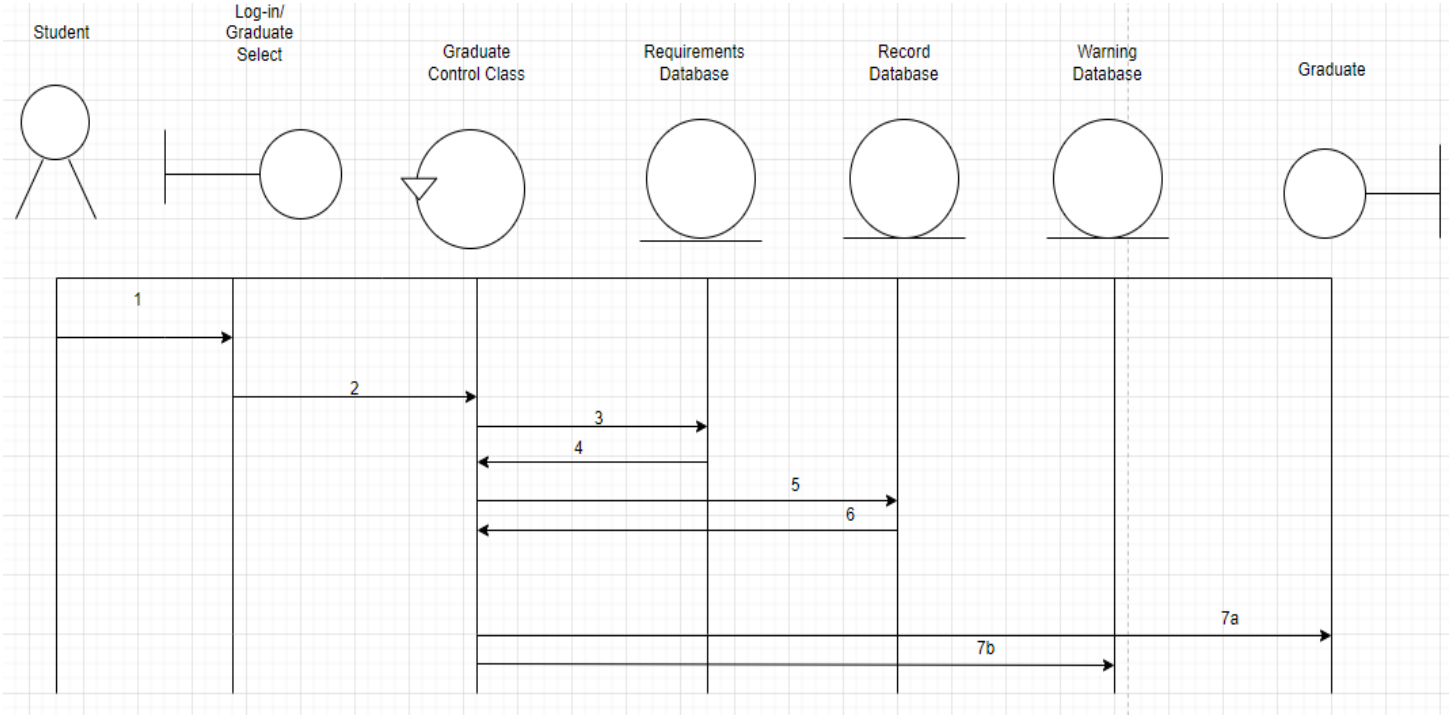


Use-Case: Graduate

Description: Students apply for graduation.

Normal Use Case: Students finishing 8 classes can apply for graduation, if the registrar finds all required courses are covered the student will graduate and leave the system with a masters degree.

Exceptional Scenario: If the student does not satisfy the required courses then they will receive a warning for reckless graduation application.



2.2.3 [Instructor Use Case]

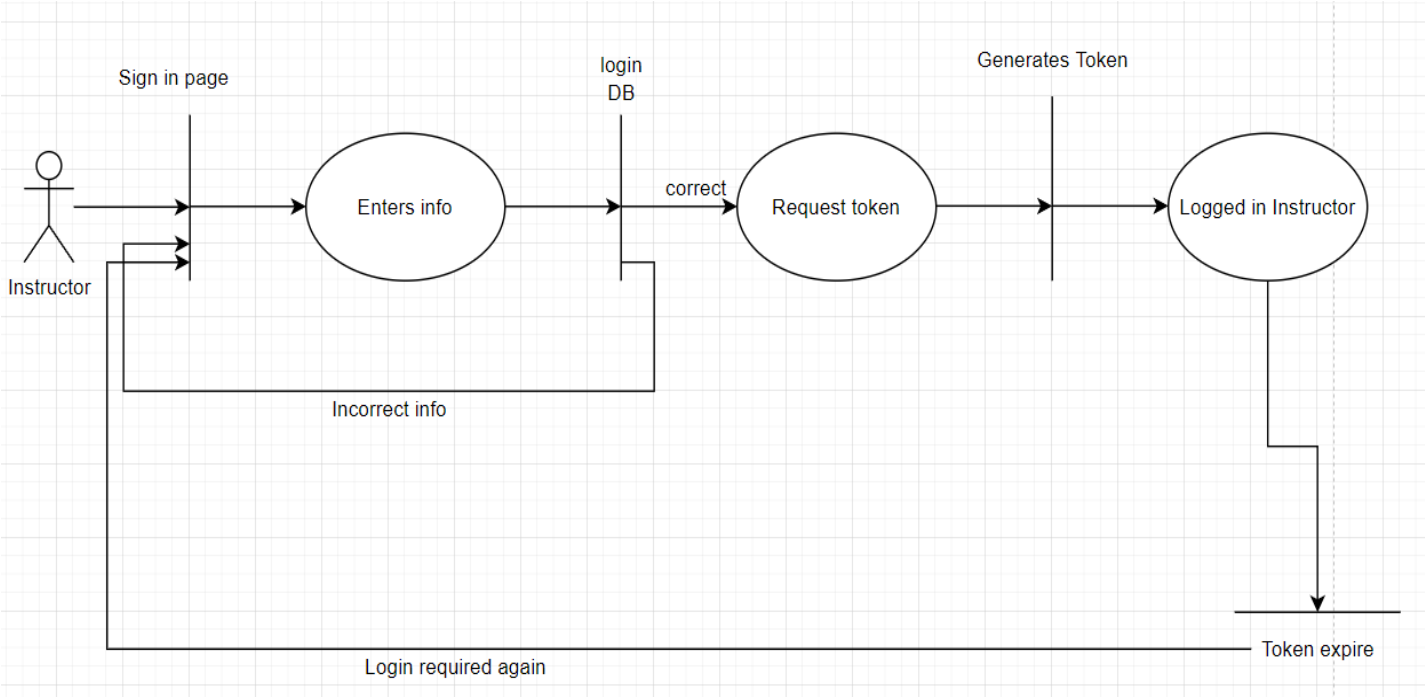
Use-Case: Login

Description: Instructors can login to the account they created on the home website page.

Instructors log into the account they have created with a valid username and password. If successful, the Instructor is granted access to the Instructor view of the website. Otherwise, an error message appears saying “Account information was entered incorrectly or the account does not exist.” In the event the Instructor is currently awaiting admission into the program, an alert will appear on the screen stating that the application is still under review.

Furthermore, if an Instructor is rejected by the Registrar, they will receive an alert whenever they try to login containing a message that entails their rejection from the program with the reason being stated. In all unsuccessful cases, they will remain on the same sign-in page.

Exceptional Scenarios: none

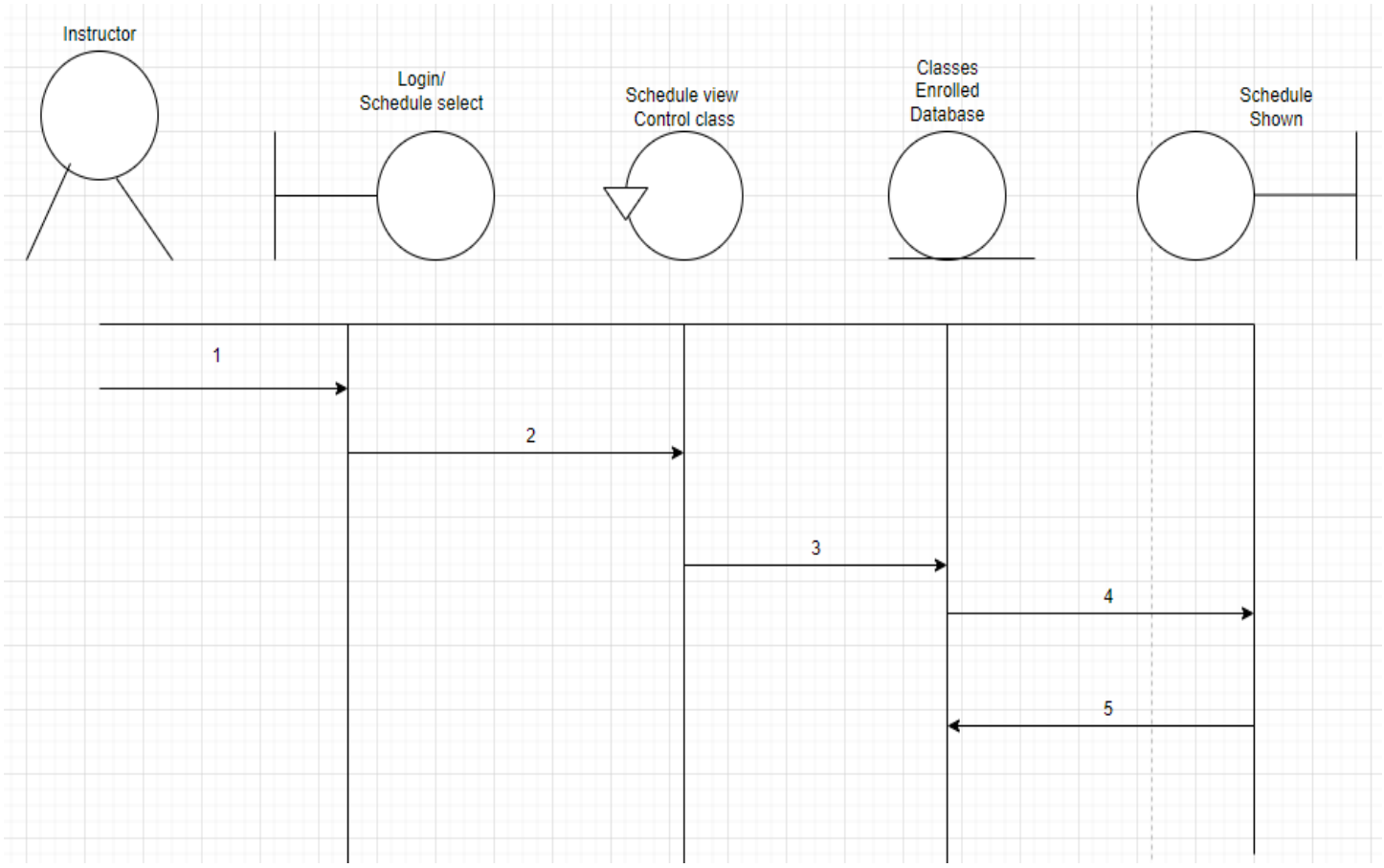


Use-Case: Schedule

Description: Instructors can see their current semester schedule composed of the courses they are teaching.

Normal Use Case: This is the default view for the instructors after they log-in. Within the largest UI window to the right, instructors can see a table of their currently taught courses. The headers of each section of the table include: Class , Time, Room, Section, and Instructor. As one could surmise, the course information for each class would appear within the respective row entry, with each field being aligned with the corresponding column of the headers. For example, CSC 342 would appear underneath the course header.

Exceptional Cases: The instructor will receive a warning if a course they teach is cancelled. The table will not display any information if the Instructor does not teach any courses. If the instructor has every course cancelled , then they will be suspended as well as not being able to teach in the next semester.

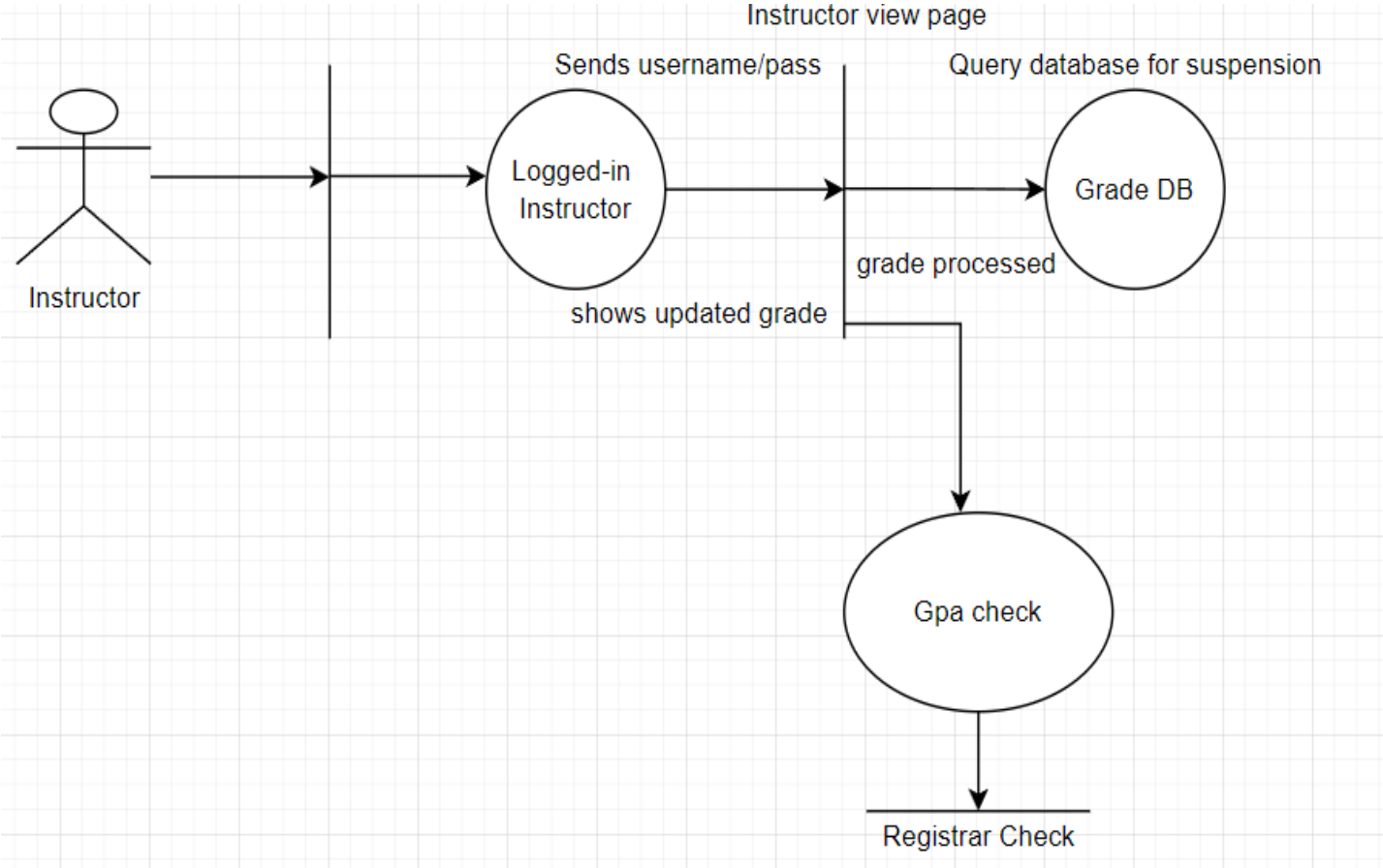


Use-Case: Assigning Grades

Description: Instructors can assign grades to the students who are currently taking classes they teach.

Normal Use Case: During the grading period, instructors assign grades.

Exceptional Cases: After the grading period ends, the instructors who did not assign grades for all students will be warned. Any instructor whose class GPA is above 3.5 or below 2.5 will be questioned by registrars, without adequate justifications the instructor will be warned or fired right away.

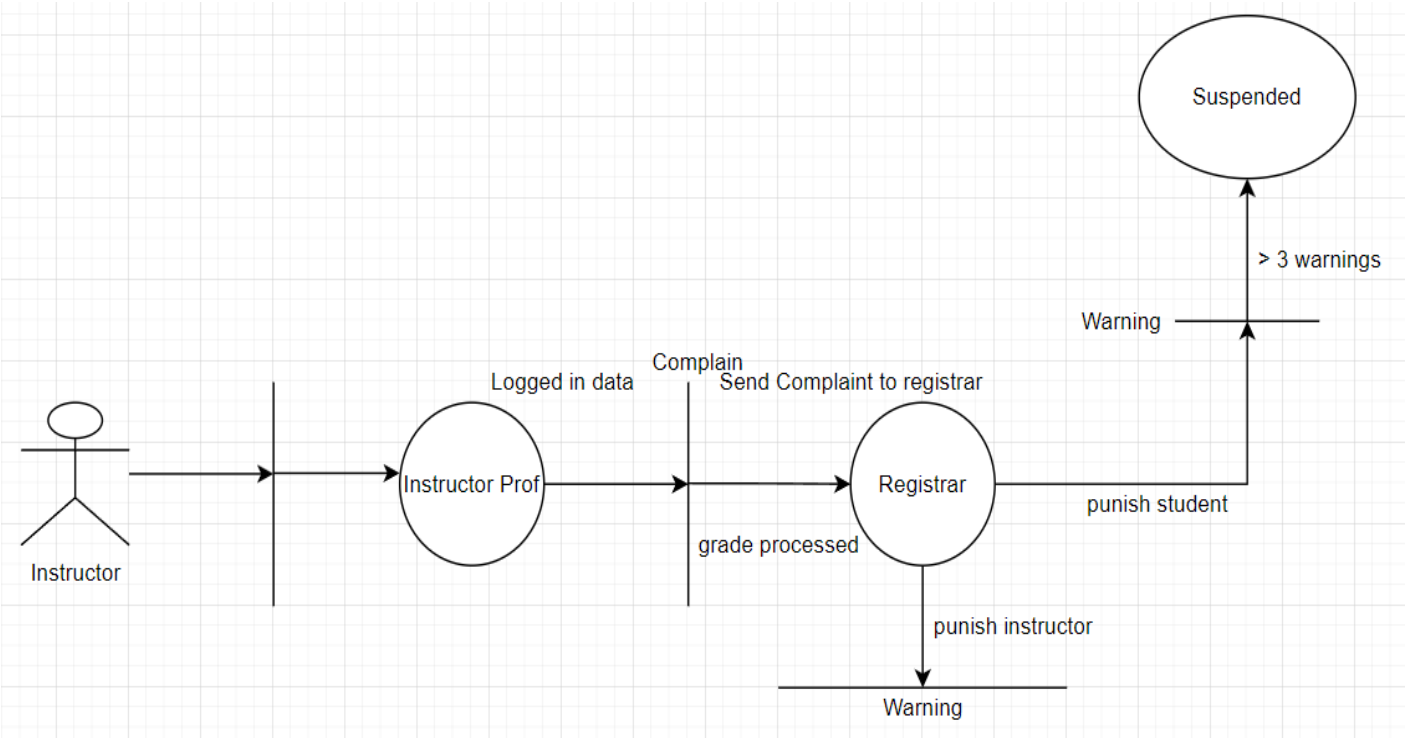


Use-Case: Complaints

Description: Instructors can complain to the registrar about a student.

Normal Use Case: An instructor can complain to the registrars to warn or de-register the student, the registrars must take action: either punish the student accordingly or punish the instructor by one warning.

Exceptional Cases: Any students receiving up to 3 warnings will be suspended for 1 semester and must pay a fine to the registrars.

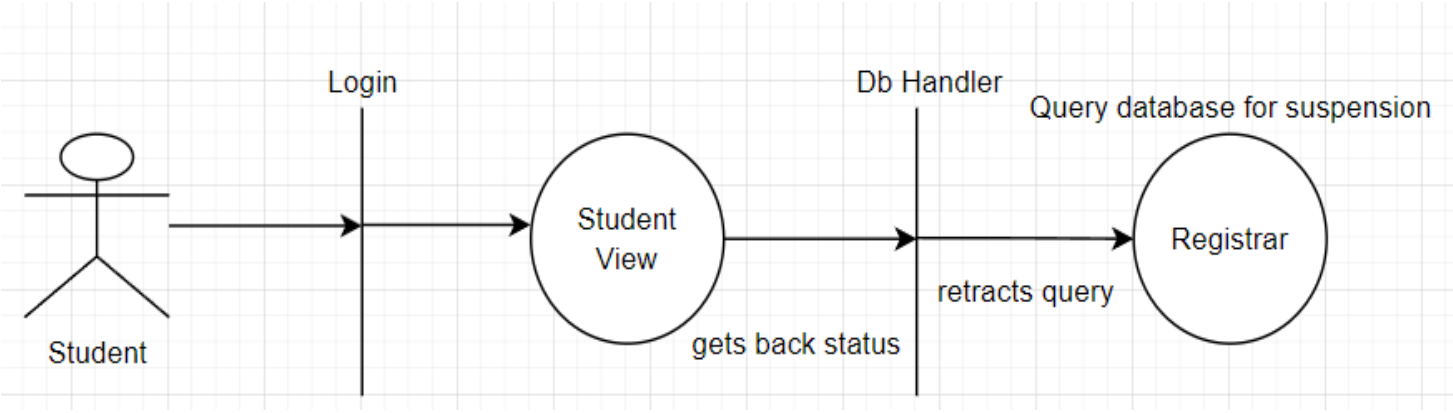


2.2.4 [Student Suspension and warning Use Case]

Use-Case: Request suspension to be taken off.

Description: Suspended students can contact the registrars to remove the suspension so they can take class within the following semester. The above use cases are restricted unless the registrars take action. Any students whose GPA is below 2 or who failed the same course twice will be terminated automatically.

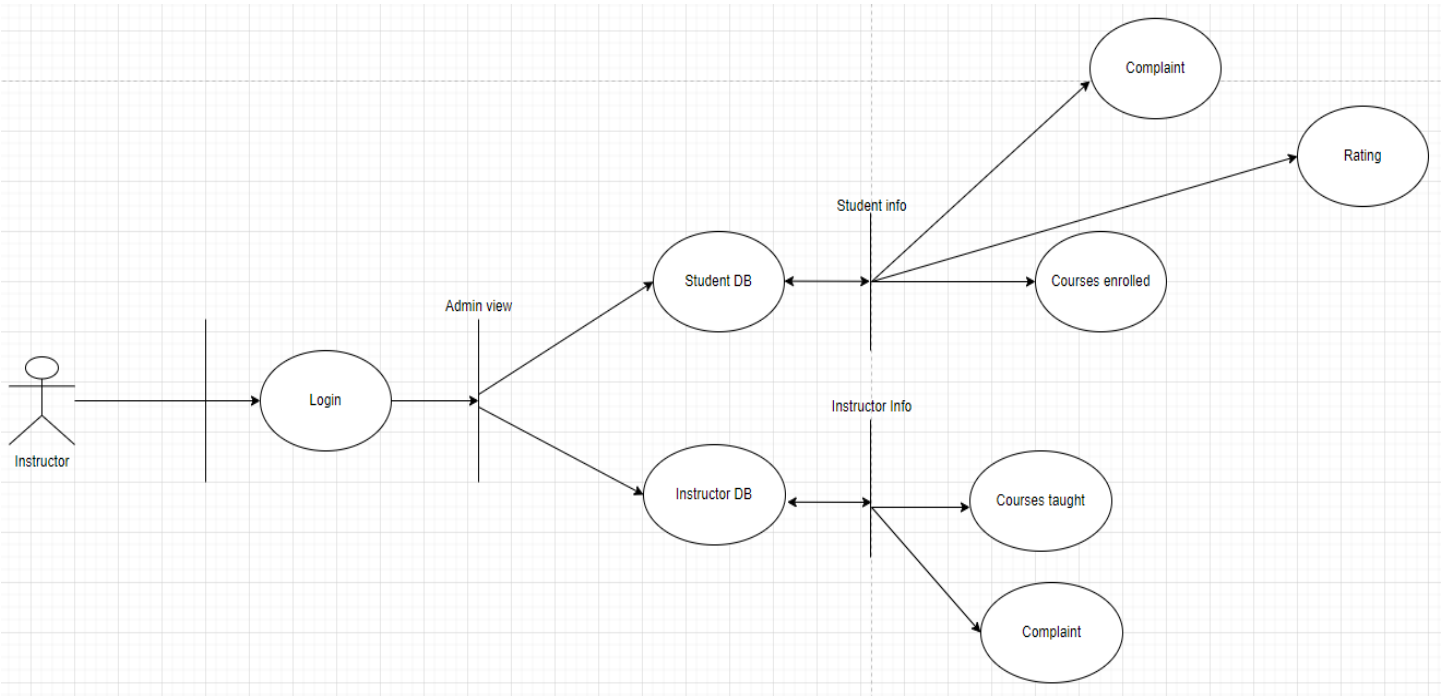
Exceptional Cases: Those whose GPA is between (2,2.25) will receive a warning demanding an interview with the registrars, must pay a fine to the registrars.



2.2.5 [Registrar Use Case]

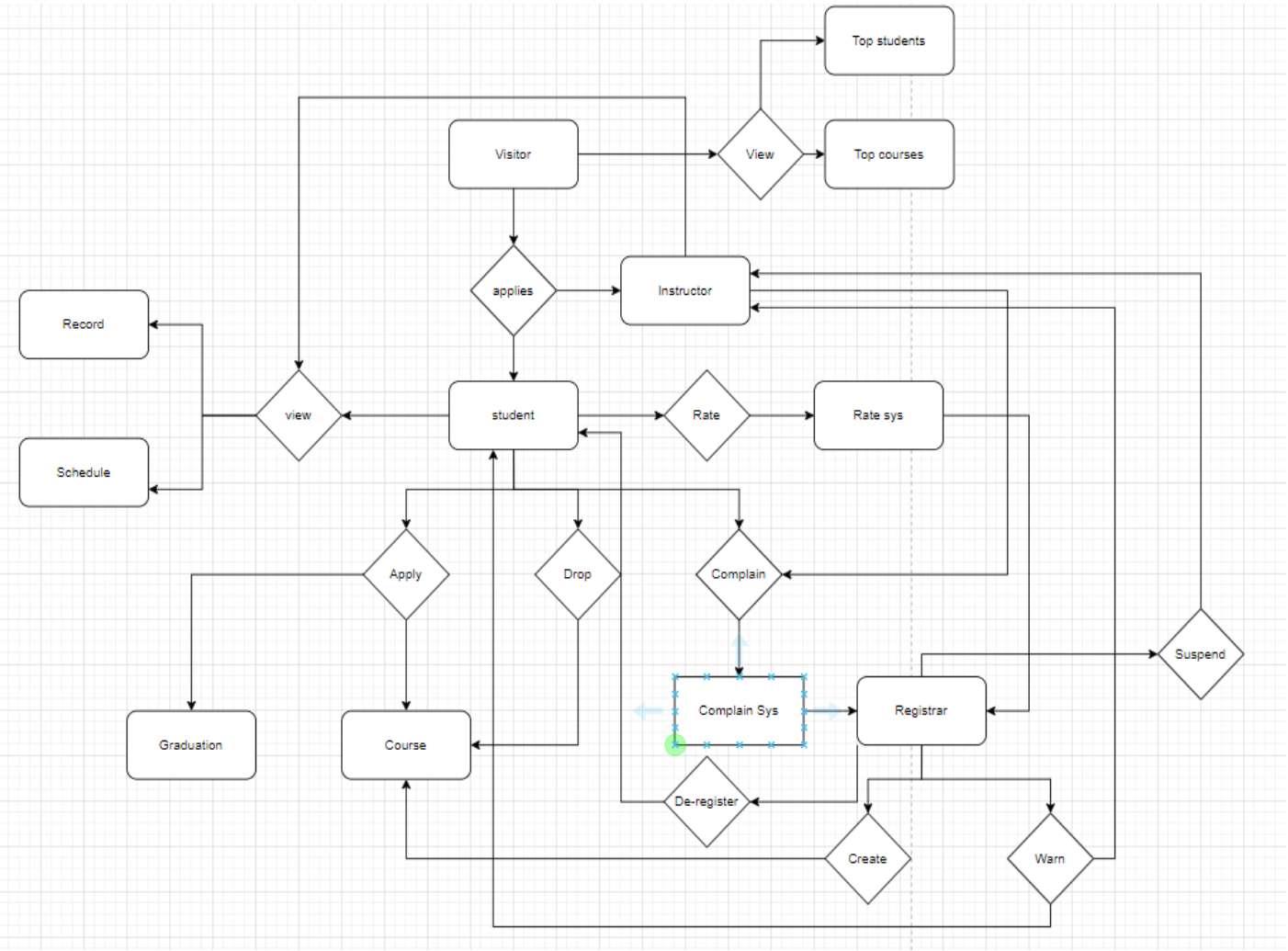
Use-case: Class setup period

Description: The registrars set up classes, class time, course instructors and class size.



3. E-R Diagram

link: https://drive.google.com/file/d/1Nz3PJSVIuCYsjnz19tBBwljwLGc0bBgJ/view?usp=sharing



4. Detailed design

Login:

Login(email, password){

Checks the system for the email and corresponding password within the database and logs in.

Returns the UserID for the user.

}

Apply:

Apply (event) {

Student/Instructor fills out a document form with the corresponding info: (firstName, LastName, GPA, DateOfBirth, email, password, role) and an EMPLID is randomly generated for them.

Data gets read from the form and it is pushed into the Users Database.

Student/Instructor awaits the registrar to accept/reject them.

If accepted, the student/instructor data is moved into the corresponding student/instructor database.

Returns nothing.

}

About Us:

getStudents(db){

Checks firebase collection with the students.

Pushes the data within a student array.

Sorts the array by GPA.

Slice the array into the top 5 students and set those students into the useState variable “students”.

Returns nothing.

}

getTClasses(db){

Checks firebase collection with the Classes.

Pushes the data within a topratingclass array.

Sorts the array by Rating,

Slices the array into the top 5 classes and sets those classes into the useState variable “tClasses”

Returns nothing.

}

getLClasses(db){

Checks firebase collection with the Classes.

Pushes the data within a lowratingclass array.

Sorts the array by Rating,

Slices the array into the lowest 5 classes and sets those classes into the useState variable “lClasses”

Returns nothing.

}

Students:

getStudentCourses(db){

Checks firebase collection with the students.

Finds the signed-in students userID and their current enrolled courses data.

Pushes the data within a student array.

Sets the CurrentClasses useState variable with the data that was pushed into the student array from the previous step.

Returns nothing.

}

getWarnings(db){

Checks firebase collection with the students.

Finds the signed-in student's userID within the collection and their Warnings data.

Pushes the data within a warning array.

Sets the Warnings useState variable with the data that was pushed into the warning array from the previous step.

Returns nothing.

}

enrollCourse(class,daytime,room,section,size,instructor,instructorUuid){

Checks the firebase collection called Courses.

Finds the specific class within the Courses collection.

Checks if the student is already enrolled in the course.

If not, Checks the number of enrollments within the class and the remaining seats.

If remaining seats are 0, place the student within the waitlist.

Otherwise, place the student within the Instructor's roster for the Course and put the student within the course's database.

Increment the current size of the class' roster by 1.

Returns nothing.

}

Complain(className,Instructor){

Get the Instructor referenced in the parameters from the database.

Get the students within the class specified by className from the database.

Set the classStudents array to contain the respective data of each student in the same course.

Returns nothing.

}

ComplainFind(UserID){

Checks to see if the UserID is within the Instructor array that was previously populated with Instructor data.

If so, variable ComplainUuid is assigned the first and last name of the instructor as a string data type.

Checks to see if the UserID is within the students array that was previously populated with Student data.

If so, variable ComplainUuid is assigned the first and last name of the instructor as a string data type.

Returns nothing.

}

SubmitReview() {

The Student submits a review of an instructor with the text and rating values being saved accordingly.

The previous rating of the instructor is called and the current rating is added to the total rating.

The total rating is averaged once more and stored within the specific Instructor's database collection.

The text of the review is then analyzed word by word to check for any taboo words listed within a tabooWords array.

If so, the student who submitted the review will receive a warning for 1-2 Taboo words found with the taboo words being censored by "*".

Similarly, the student would receive two warnings for reviews that contain 3+ taboo words and the review would not submit.

Updates the student's warning collection and warning count if they incurred a warning due to their review.

Returns nothing.

}

SubmitComplaint(){

The student's first name, last name, complaint target, and complaint is saved and added to the firebase collection "Complaints".

An alert message appears notifying that the complaint was submitted.

Returns nothing.

Instructors:

getStudents(db) {

Retrieves the full list of students from the student collection within the database.

Sets the students array to contain the data from each student.

Returns nothing.

}

getWarnings(db){

Retrieves the full set of warnings given to the instructor that is currently signed into website.

Sets each warning's respective data into an warn array for that said instructor.

Returns nothing.

}

getInstructorCourses(db){

Retrieves the full list of Courses that the currently signed-in instructor has under their jurisdiction during this semester from the database.

The data for each course they are assigned is pushed into a course array.

Returns nothing.

}

Complain(ClassName){

Finds the class of the student that the instructor would like to issue a complaint towards.

Pushes the data of each student into an array instComp.

Returns nothing.

}

Complain(studentName){

Sets the variable studentComplainName with the value of the name for the student that the instructor would like to issue a complaint about.

Returns nothing.

}

SubmitComplaint(){

The instructor's first name, last name, complaint target, and complaint is saved and added to the firebase collection "Complaints".

An alert message appears notifying that the complaint was submitted.

Returns nothing.

}

AssignGrades(){

Instructor assigns grades for each student in their roster for their courses.

Returns nothing.

}

Registrar:

getStudents(db) {

Retrieves the full list of students from the student collection within the database.

Sets the students array to contain the data from each student.

Returns nothing.

}

getInstructors(db) {

Retrieves the full list of students from the student collection within the database.

Sets the students array to contain the data from each student.

Returns nothing.

}

WarningCheckStdCourses() {

Checks the number of Courses that each student is enrolled in the moment the course running period commences.

Warns students that have less than 2.

Returns nothing.

}

SuspensionCheckInstrCourses(){

Checks whether all of an instructors courses in the semester were canceled.

If so, Suspend the Instructor

Returns nothing

}

SuspensionCheckInstrWarnings(){

Checks if the number of the warning of an instructor is less than 3.

If so, nothing happens.

Otherwise, Suspend the Instructor.

Returns nothing.

}

CancelCourses(){

Cancel all courses that have less than 5 students enrolled in the course.

Give the students a second chance to enroll into courses if cancelled.

Warn the instructor of the course that was cancelled.

Returns nothing.

}

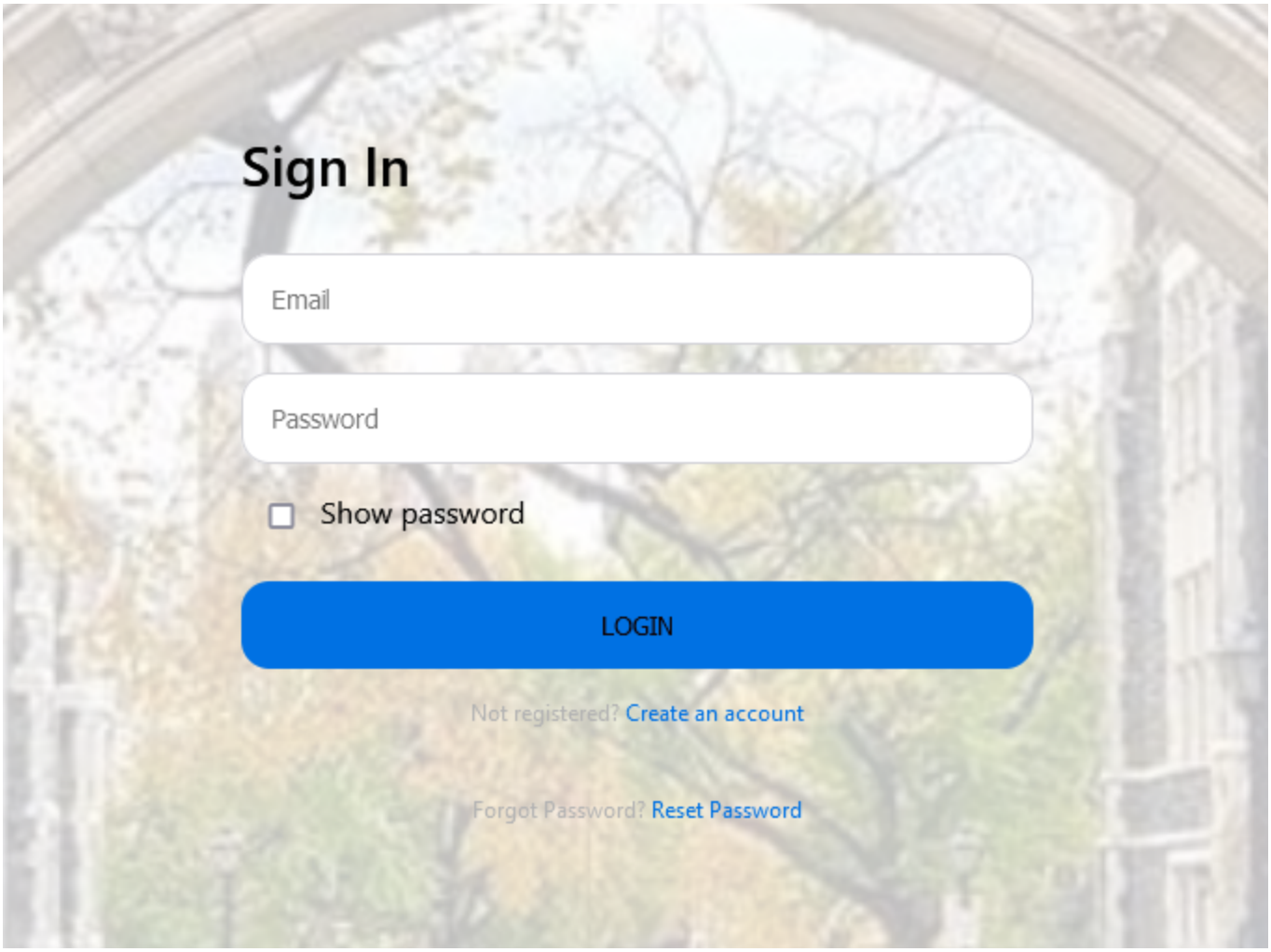
Accept Students/Instructors

```
async function Accept(a, b, c, d, e, f, g ,useruiid, h){
    if(f === "0"){
        const payload = {firstname: a, lastname: b, GPA: c, DateofBirth: d, Email: e, Role: "Student", password: g, useruiid:useruiid, empl: h, numWarn
        console.log(useruiid);
        await setDoc(doc(db, "Students", (parameter) useruiid: any
        await deleteDoc(doc(db, "Users",useruiid ));
    }else{
        const payload = {firstname: a, lastname: b, DateofBirth: d, Email: e,Role: "Instructor", password: g, useruiid:useruiid, Review: 1, numReview:
        const userid = useruiid;
        firtname = a;
        lastname = b;
        togglecourseAssignPopup(useruiid);
        await setDoc(doc(db, "Instructor", useruiid), payload);
    }
}
```

Assign Courses

```
async function Assign(a,b,c,d,f){
    classes=a;
    try{
    await setDoc(doc(db, "Instructor", ud, "Courses", a), {
        DayTime: b,
        Room: c,
        Secion: d,
        Size: f,
        Instructor: firtname + " " + lastname,
        Class: classes
      });
      await setDoc(doc(db, "AssignedClasses", a), {
        Class: classes,
        DayTime: b,
        Room: c,
        Secion: d,
        Size: f,
        Instructor: firtname + " " + lastname,
        Instructoruiid: ud
      });
    }catch{
        alert("Error");
    }
}
```

5. System screens



CCNYZero About Us

About Us

The City College of New York (established as 'The Free Academy' in 1847) is the founding institution of the City University of New York and home to eight schools and divisions, each dedicated to the advancement of research and knowledge. We're the place where Albert Einstein first presented his theory of general relativity outside of Europe, our alumni discovered the polio vaccine, helped build the Internet, and designed the Panama Canal. And, our alumni have gone on to become Supreme Court Justices, Secretaries of State, leaders of industry, world-renowned researchers and award winning actors and musicians. We're the City College of yesterday, today and tomorrow. See yourself here. City College.

CCNY Schools and Divisions

The Bernard and Anne Spitzer School of Architecture

CUNY School of Medicine

The Grove School of Engineering

Division of Humanities & the Arts

Colin Powell School for Civic and Global Leadership

School of Education

Division of Interdisciplinary Studies at the Center for Worker Education

The Division of Science

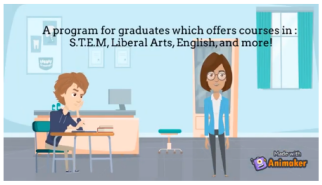
©Questionable, 2021

$5M+
Total student scholarships, stipends, and fellowships awarded

115+
Number of undergraduate majors and master's programs

150+
Total number of countries that CCNY students represent

95.6%
Full-time faculty members who hold terminal degrees in their fields



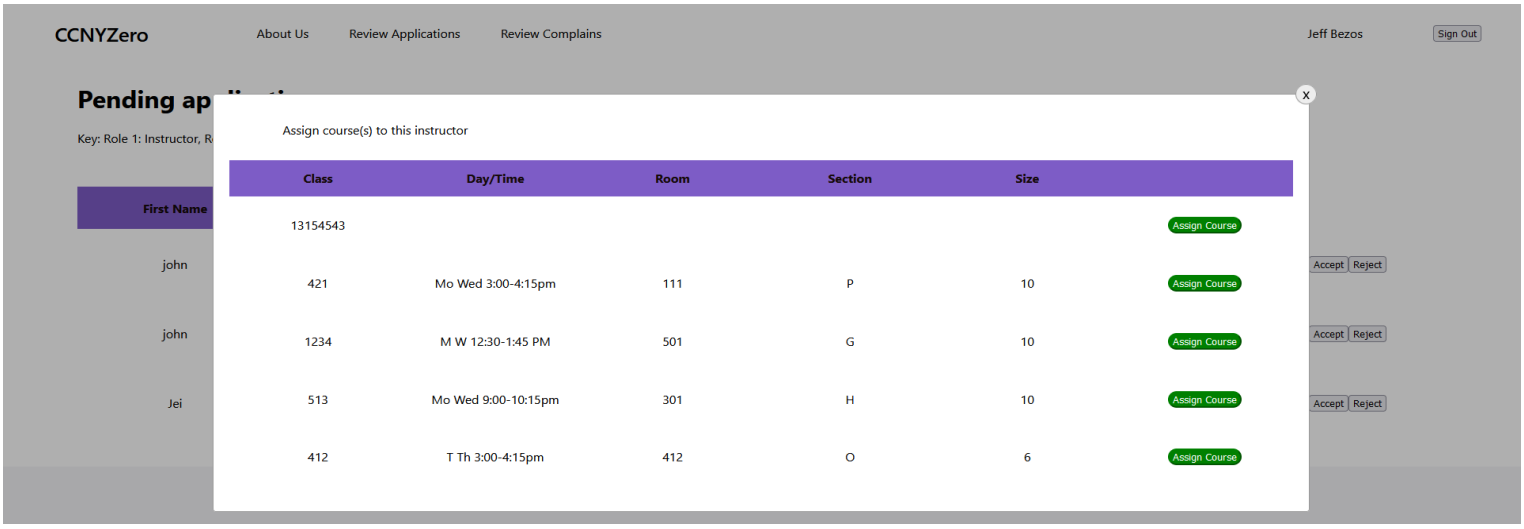
Top students and Top Classes

First Name	Last Name	GPA
Deja	Vu	4
Andrew	Syed	3.954
Adam	Smith	3.896
Kelly	Johnson	3.584
Miss	Jackson	3.456

Course	Rating
Math210000	4.79
Math203000	4.52
HIST24100	4.5
Anth20100	4.2
EE20500	4.04

Course	Rating
ENGR10100	2.6
CHEM10200	3.06
CSC10400	3.4
PHYS20700	3.71
CSC22000	3.83


CCNYZero
About Us
Review Applications
Review Complains
Jeff Bezos
Sign Out
Pending applications
Key: Role 1: Instructor, Role 0: Student
First Name
Last Name
GPA
DOB
Email
Role
john
doe
2.4
2020-10-30
johndoe@gmail.com
0
Accept
Reject
john
cena
3.5
1975-09-25
johncena@gmail.com
0
Accept
Reject
Jei
Wei
4.0
1995-05-25
jeiwei@gmail.com
1
Accept
Reject
Copyright © 2021 CCNYZero


CCNYZero
About Us
Review Applications
Review Complains
Jeff Bezos
Sign Out
Assign course(s) to this instructor
Class
Day/Time
Room
Section
Size
13154543
Assign Course
421
Mo Wed 3:00-4:15pm
111
P
10
Assign Course
1234
M W 12:30-1:45 PM
501
G
10
Assign Course
513
Mo Wed 9:00-10:15pm
301
H
10
Assign Course
412
T Th 3:00-4:15pm
412
O
6
Assign Course
First Name
john
Jei
Accept
Reject

6. Group Meetings

Meeting #	Date	Time	Topic
1	09/26/2021	2 HRS	Introduction
2	10/03/2021	5 HRS	Project initialization
3	10/10/2021	4 HRS	Discussion/ Group Coding
4	10/17/2021	3 HRS	Group Coding
5	10/24/2021	4 HRS	Phase 1 Report
6	10/31/2021	3 HRS	Discussion

7	11/07/2021	4 HRS	Debugging and programming
8	11/14/2021	4 HRS	Phase report 2

7. Github Repositories

- Repo: https://github.com/NishanthPrajith/Csc-322-Final-Project