

Assignment 2

Program Functioning Part D video: <https://youtu.be/KKUDWvJMqoM>

Program Testing Part E video: <https://youtu.be/jL5laSpULJQ>

Part A

Selection and admittance into a year 2 program

Actors:

- Student
 - The student that wants to apply for a year 2 program (for engineering)
- Engineering Advisor
 - The advisor places the student into the program of choice (makes final decision)

Student Stories:

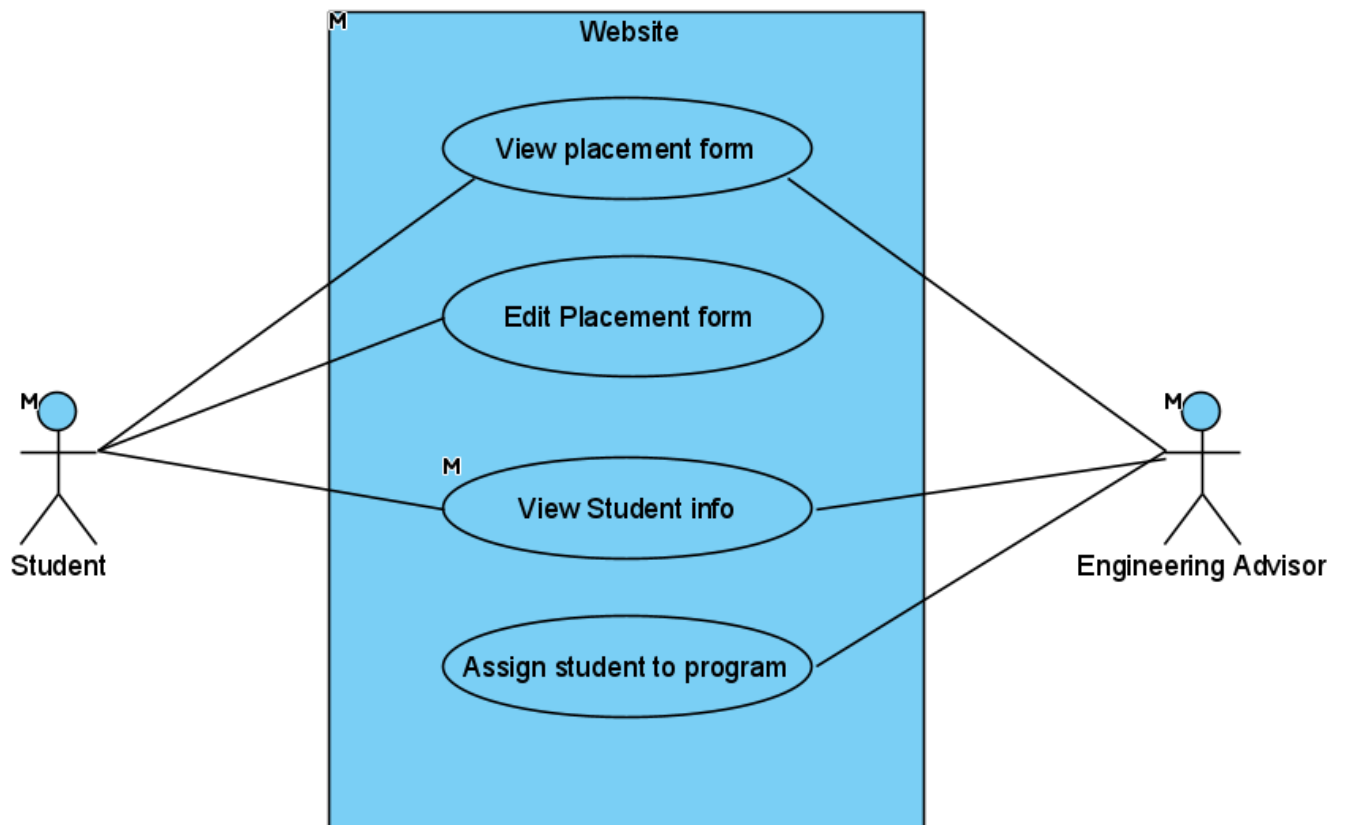
- Edit placement form
 - Data: [Ranked list = array]

Both stories:

- View ranked list of programs
 - Data: [Ranked list = array]
- View student info (grades, avg, student number) / good standing
 - Data: [Student ID = Number, 8 digits]
 - Data: [Name = String, 24 chars]
 - Data: [Avg = Number, 3 digits]
 - Algo: course marks added up divided by total courses taken in the year
 - Data: [Ranked list = array]
 - Data: [Year = Number, 1 digit]
 - Data: [Eligible for promotion = Bool]
 - Algo: 80% courses passed (over 50%) for current year

Engineering Advisor stories:

- Assign student to program
 - Data: [Program = string]
 - Algo: removes all non-eligible promotion students, give student above threshold their first choice and gives the rest their next choice.



Adjudicate the eligibility of a student to graduate from year 2 to year 3 and year 3 to year 4 etc. (what are the rules about this).

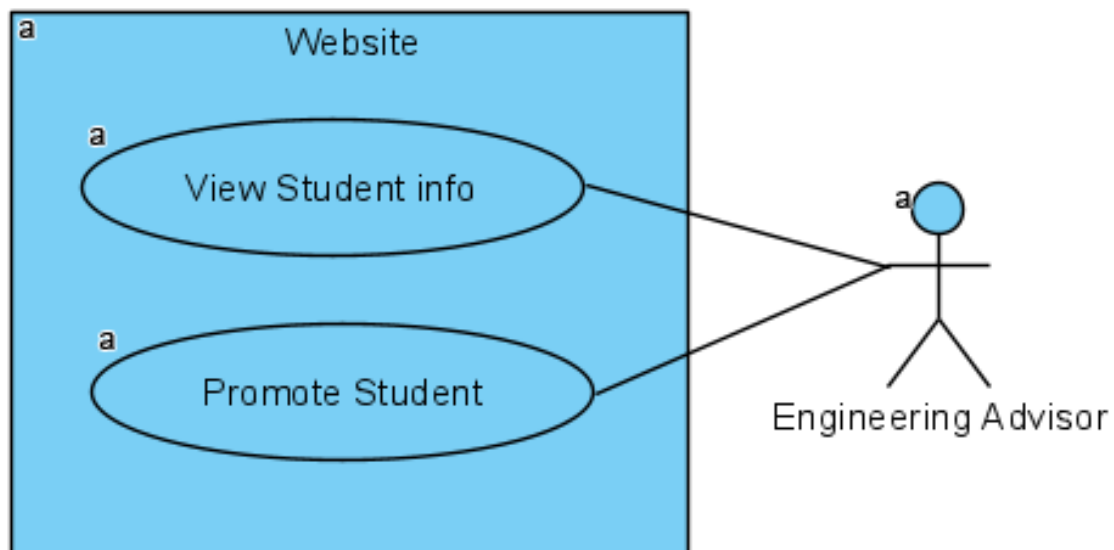
Basically, degree navigator

Actors:

- Engineering Advisor
 - The advisor promotes the student

Engineer advisor stories:

- View student info (grades, avg, student number) / good standing
 - Data: [Student ID = Number, 8 digits]
 - Data: [Name = String, 24 chars]
 - Data: [Avg = Number, 3 digits]
 - Algo: course marks added up divided by total courses taken in the year
 - Data: [Program = Tuple, (2 digits int, string)]
 - Data: [Year = Number, 1 digit]
 - Data: [Eligible for promotion = Bool]
 - Algo: 80% courses passed (over 50%) for current year
- Promote student
 - Data: [Student info -> Year]
 - Algo: if [Student info -> Eligible for promotion] is true then promote.



Graduate (or not) a student at the end of the program (lots of checks involved here)

<https://ssc.adm.ubc.ca/sscportal/servlets/SSCMain.jsp?function=GraduationService>

Actors:

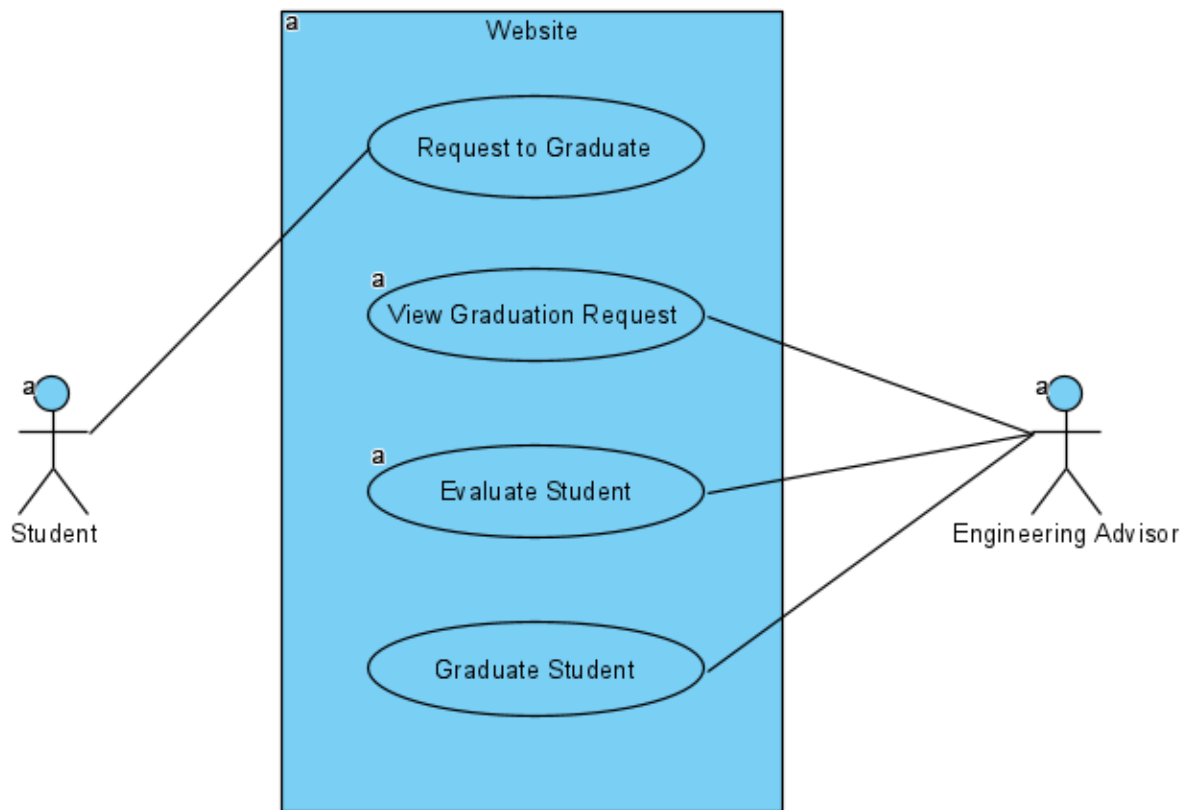
- Student
 - The student has to apply for graduation
- Engineering Advisor
 - The advisor approves graduation of the student

Student stories:

- Request for graduation
 - Data: [Student -> I want to graduate] goes true

Engineer advisor stories:

- View/process request for graduation
 - Algo if [Student -> I want to graduate] is true then run Process Graduation
- Evaluate student info (grades, avg, student number) / good standing
 - Data: [Student ID = Number, 8 digits]
 - Data: [Name = String, 24 chars]
 - Data: [Program = Tuple, (2 digits int, string)]
 - Data: [Year = Number, 1 digit]
 - Data: [I want to graduate = Bool]
 - Student's request for graduation
 - Data: [Eligible for graduation = Bool]
 - Algo: Good standing and all degree requirements fulfilled and no outstanding fees
 - First year completed
 - Algo: all courses passed
 - Second year completed
 - Algo: all courses passed
 - Third year completed
 - Algo: all courses passed
 - Fourth year completed
 - Algo: all courses passed
 - Complementary Studies courses completed
 - Algo: all courses passed
 - Fees ok
 - Algo: no parking tickets or library fees
 - Data: [Graduation approved = bool]
- Process graduation
 - Algo: if [Student info -> Eligible for graduation] is true then [Student info -> Graduation approved] is true.



President suspends student for 1 or 2 terms (or indefinitely) based on number of incidences of cheating in a course while at UBC. This requires that students be banned from registering during suspension, a zero is entered in the course and a record of the incident is entered on their transcript and recorded in Deans office

Actors:

- President's office
 - The president's office suspends the student based on number of cheating incidents
- Dean's office
 - Notified of the incident and records it.
- Prof
 - Has a course that the student cheated in

All stories:

- View student info (grades, avg, student number) / good standing
 - Data: [Student ID = Number, 8 digits]
 - Data: [Name = String, 24 chars]
 - Data: [Standing = Number, 1 digit]
 - Data: [cheating status = Bool]
 - Algo: if suspended cannot register, then true, otherwise false
-

President stories:

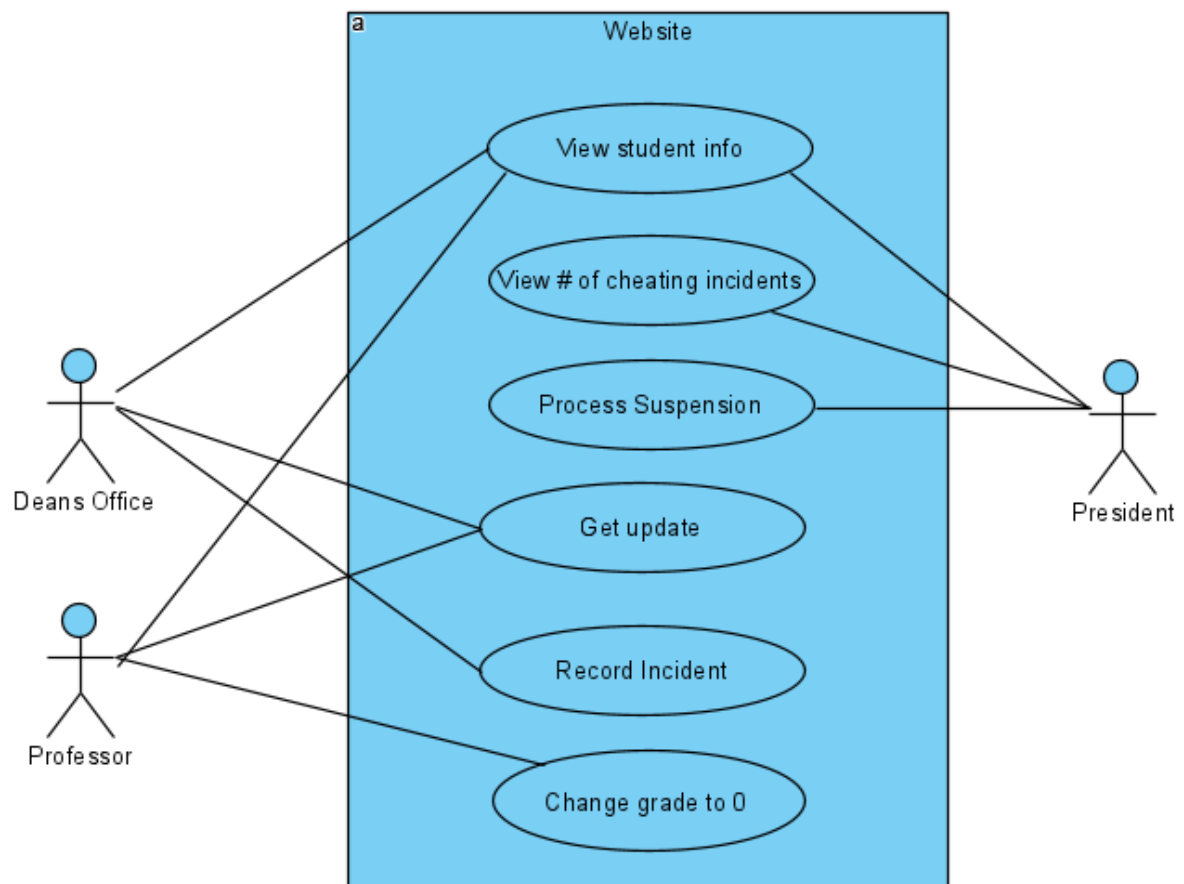
- Process suspension
 - Algo: if not currently suspended
 - 0: 0 for the course, no need to suspend
 - 1: 1 term suspension
 - 2: 2 terms suspension
 - 3+: Expulsion + indefinite suspension
 - Notifies Dean to record incident

Dean:

- Record incident
 - Data: [Student -> cheating incidents (dean) = Number, 1 digit]

Prof:

- Change grade to 0
 - Data: [student id grade-> zero]



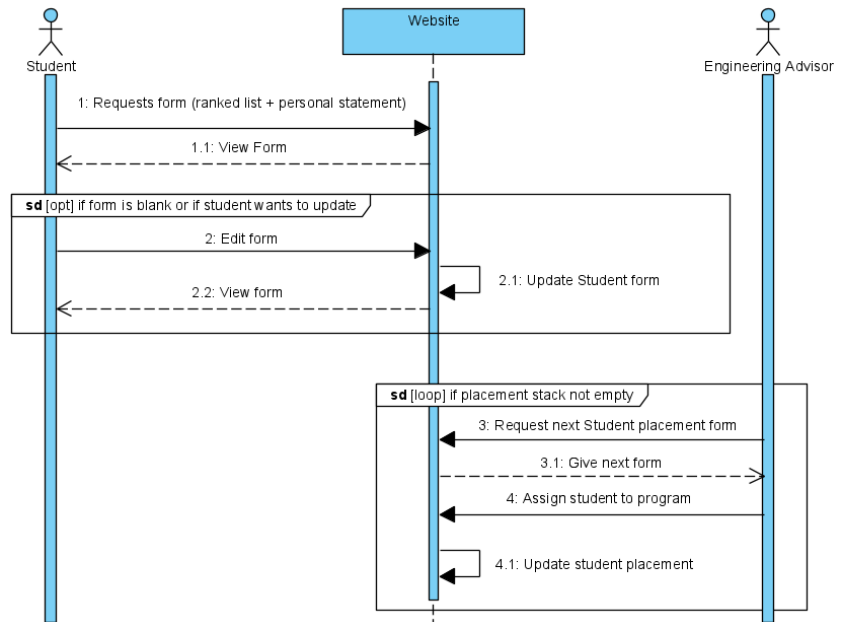
Part B

2

Student requests form to view. The website queries itself for student form, then returns it for display.

In a loop, if the form is blank or if student wants to update, the student will edit the form, which in turn updates the website.

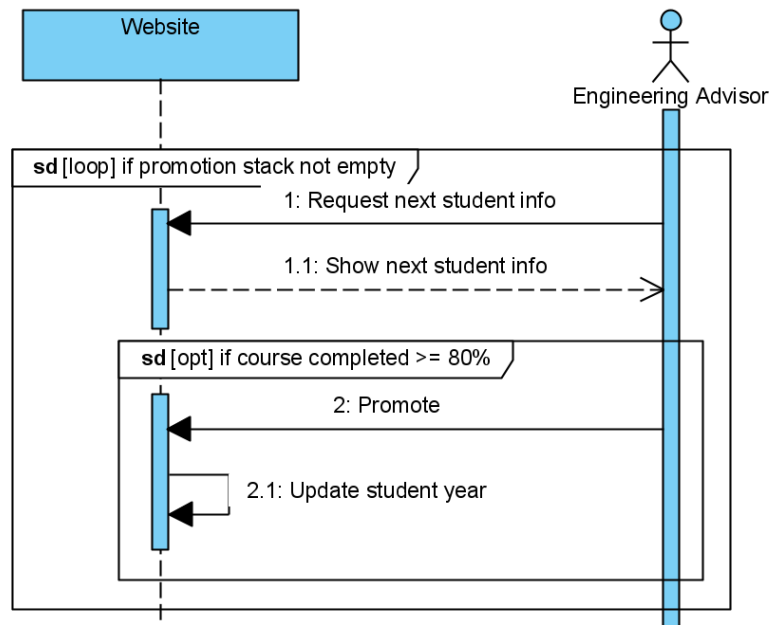
If the placement stack is not empty, then the advisor requests to see the next student form, gets the next form, then assigns student to a program based on internal algorithm of the advisor.



6

In a large loop, the website displays the next student's info of the promotion stack.

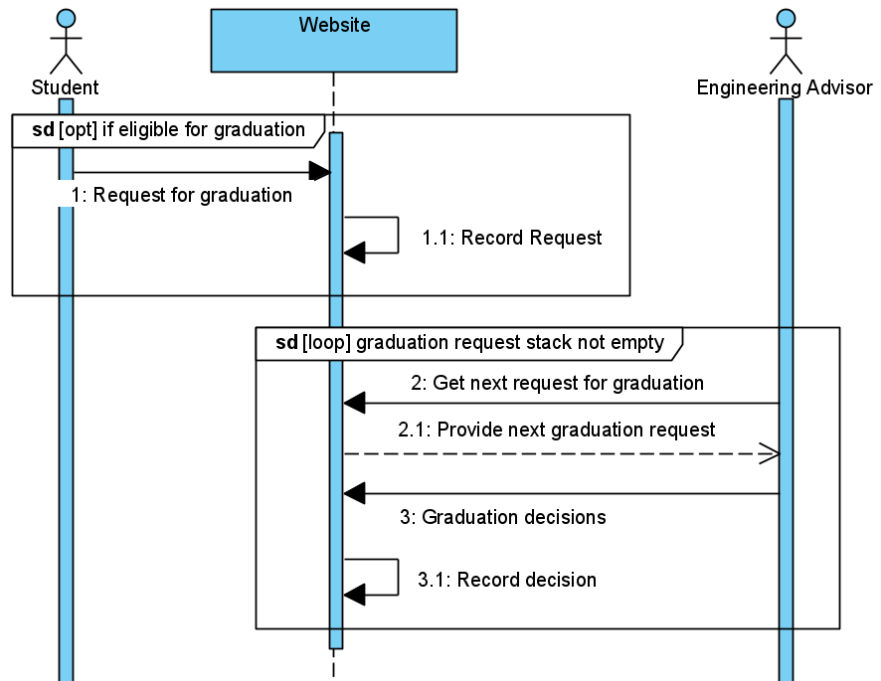
If the student has completed over 80 % of courses for the year, then there is a deliberation by the advisor for a potential promotion. The advisor can elect to promote depending on the student's info, and it updates the website.



7

If the student is eligible for graduation, then the student will request for graduation.

If the graduation request stack is not empty, the engineering advisor looks into the stack of requests. Gets the next request and processes graduation decision, then updates the website (record).

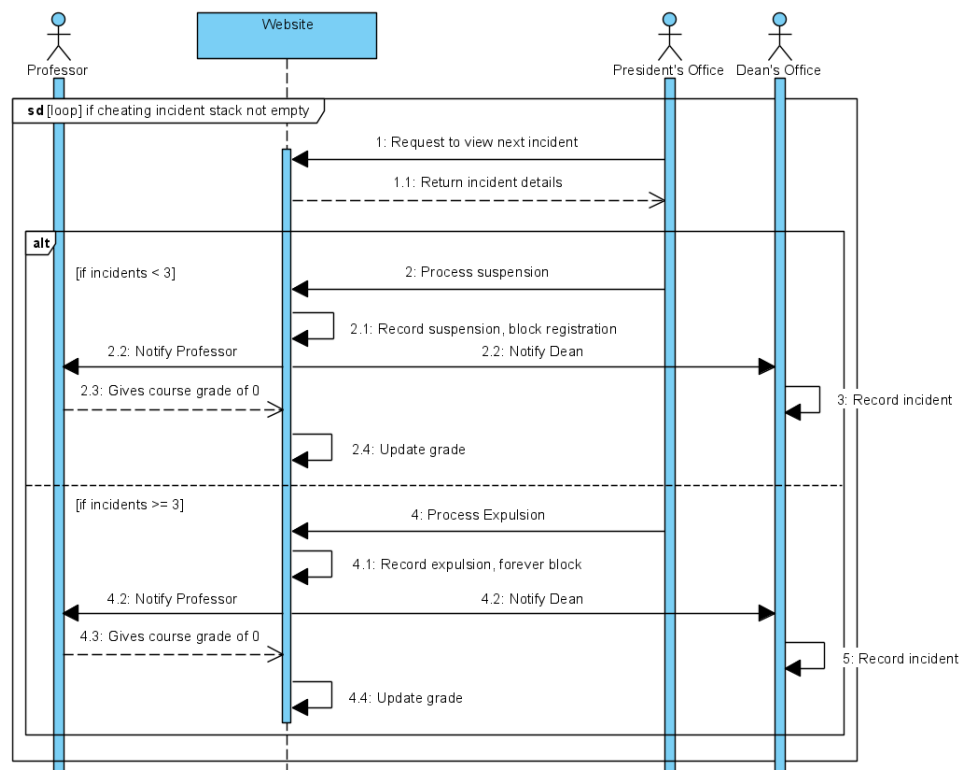


8

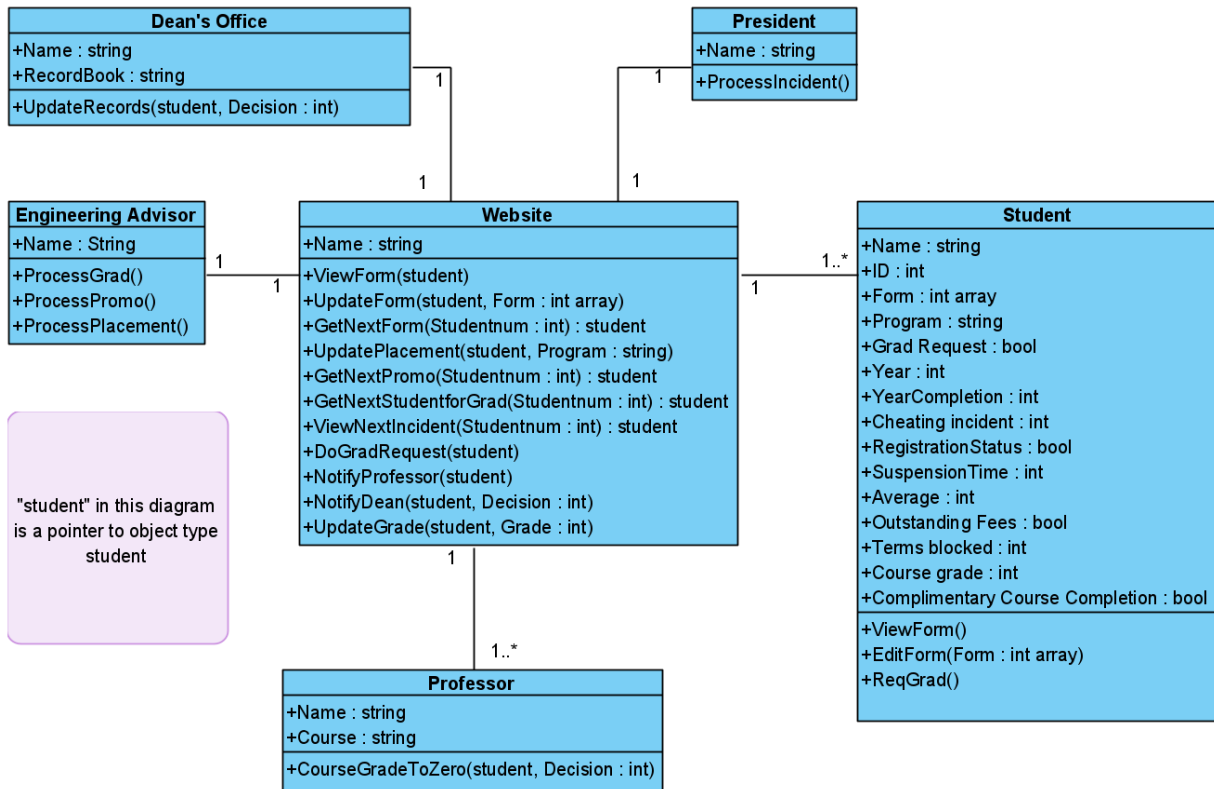
If there is a cheating incident in the stack, the president's office requests to view the incident.

Depending on the number of incidents, the student is either suspended or expelled. Either way it will be recorded by the Dean's office and the course mark will be 0.

The detail in suspension is within the algorithm of processing the suspension.



Part C



PART E

GitHub Demonstration:

master		
Commits on Dec 4, 2020		
Merge branch 'master' of https://github.com/mchuahua/CPEN333	bb287f7	<>
varunrajana committed 33 minutes ago		
testing complete	152d12d	<>
varunrajana committed 34 minutes ago		
edited readme	8e48b25	<>
mchuahua committed 36 minutes ago		
partE setup	b7e9e7f	<>
varunrajana committed 22 hours ago		
Commits on Dec 3, 2020		
partD working	b89359d	<>
varunrajana committed 23 hours ago		
partD 2 working	afacd4a	<>
varunrajana committed yesterday		
working on part D	0ff5fc4	<>
varunrajana committed 2 days ago		
Commits on Dec 2, 2020		
working on part C&D	152e54e	<>
varunrajana committed 2 days ago		
stuf	129622b	<>
mchuahua committed 2 days ago		
updated part B again	895bbee	<>
varunrajana committed 2 days ago		
updated part B	88d4878	<>
varunrajana committed 2 days ago		
Commits on Dec 1, 2020		
Part B done and added part A stuff	b33c3b8	<>
mchuahua committed 3 days ago		
Part A done	2c3e7c1	<>
mchuahua committed 3 days ago		
Commits on Nov 29, 2020		
basic usse cases done	4d2828b	<>
mchuahua committed 6 days ago		
Commits on Nov 28, 2020		
basic usse cases done	1d60c3e	<>
mchuahua committed 6 days ago		
Added descriptions for part A	4ec2051	<>
mchuahua committed 6 days ago		
new vpp	5160305	<>
mchuahua committed 6 days ago		
sfd`	a4c2b94	<>
mchuahua committed 6 days ago		