PART A Capturing Requirements with a USE-Case Diagram

Actors:

- Student
- Prof
- Admin
- Dean
- President

Student:

STUDENT ITEMS

- Student ID: UINT, 8 digits
- Name: String, 20 char
- Grades:List of courses, 2 char percentage
- Tuition Fees, unsigned long
- Eng Specialization: 20 char string
- Enrolled courses: list of courses, char string course number eg. APSC
- Username
- Password

USE CASE:

Enroll into UBC Engineering Year 1

- User requests action from system
- System checks:
    - User grades
    - Program eligibility (pre-reqs, transcripts, etc)
- If eligible, system creates request for administration to approve
- If not eligible, System does nothing
- Print student registration/program status to user
- INCLUDES:
    - Identify User (Login)

Selection/admittance into year 2 program

- System checks:
    - User data
    - Program size/average/eligibility
- If eligible, system creates request for administration to approve
- If not eligible, System does nothing
- Print student registration/program status to user
- INCLUDES:
    - Identify User (Login)
    - (Access Admin database)

View Student Details

- Student has access to any of their own data
- Simple request and display from the system
- Not allowed to write.
- INCLUDES:
    - Identify User (Login)
    - Browse student information
        - EXTENDS:
            - Enroll in a Course
            - Drop a Course
    - (Access Student Database)

View course details

- Student may access any non-confidential course data of a course they are registered in.
- Not allowed to write.
  INCLUDES:
    - Identify User (Login)
    - Browse course information
    - (Access course database)

<u>Prof:</u>

PROFESSOR ITEMS

- course List : vector <Course>
- Faculty ID
- Name
- Username
- Password

USE CASE:

Modify Course Information

- Professor will request to create folder in the database (assignments, labs, quizzes, etc)
- Professor can manage what users have access to the folder eg. students or professor only
- All folders will default to professor only unless permission is explicitly shared.
- INCLUDES:
    - Identify User (Login)
    - (Access course database)
- EXTENDS:

        - Create Assessment Proposals
        - Remove Assessment Proposals

Submit course grade and standing for each student

- In course database, each registered student will have data stored within the course
- When a graded assignment is added to the database, each student in the course database will have individual data for that assignment
- The prof may assign grades to each student individually
- INCLUDES:
    - Identify User (Login)
    - Post student assignment grade
    - Post student cumulative course grade
    - Access course database

Admin (enrollment services, advising, etc) :

ADMIN ITEMS:

- Faculty ID
- Name: String, 24 char
- Username
- Password

USE CASE:

Browse Course Information

- INCLUDES:
    - Identify User (Login)
    - (Access Course Database)
- EXTENDS:
    - Assign prof to a course
    - Create Course
    - Delete Coures

Adjudicate eligibility of student to proceed to next year

- Is able to view open requests from students to register in 1st or 2nd year, approve or deny them.
- INCLUDES:
    - Identify User (Login)
    - Approve/Deny student proceeding to next year standing
    - (Access admin/student database)

Graduates student at end of program

- Is able to view graduation requests, approve or deny them

INCLUDES:

- Identify User (Login)
- Approve/Deny student proceeding to next year standing
- (Access student/admin database)

President

PRESIDENT ITEMS

- Faculty Members ID:
- Name: String, 24 char

USE CASE:

Browe Student Information

- INCLUDE:
  - Identify User (Login)
- EXTENDS
  - Suspend student when academic misconduct occurs
  - (Access student database)

Manages all University departments (Deans) and Programs

- Administrative database
- INCLUDES:
  - Identify User (Login)
  - (Access student/admin database)

System Overview:
Every user (prof, student, etc) will need to use the (identify user) use case to do any of their required actions. All actions (view grades, create exams, etc) will assume the user has already been authenticated.

Databases:
User Database

- contains all students/prof/admin/president data
- student information
- student standing and specialization
- student/prof courses

Course Database

- all courses
- course details

- enrolled students

Specialization Database

- list of specializations
- required courses
- required grades

USE CASE: LOGIN - Identify user (required for every situation)

- The user inputs their credentials (Account name, password)
- The system reads the credentials and identifies the user & user data
    - If unable to identify, the system will request new credentials
- System confirms user and prompts action
- Recover last password via email(?)

PART B Capturing Use-Case Behaviour using a Sequence Diagram

Use Cases:

- **2. Admittance into 2nd Year Program (Student)**
- **8. President suspends student based on cheating (President)**
- **Create Assignment Proposal (Prof)**
- **Enroll in a Course (Student)**

CWL Login (allow or reject)

- User logs in, system identifies user as student, prof, admin, dean, or president. Can allow or deny access based on credentials

User Input

- Allows user to enter information into system (login, options, etc)

Computer (select option)

- Displays list of options available based on user privileges (student, prof, admin, etc)

System Database

- Stores and contains all user, standing, and course information, can browse or update based on user privileges (student, prof, admin, etc)

Specialization

- Display specialization information
- Students can request to get in a 2nd year specialization to admin
- Admin can approve or reject request if student has prerequisites

Course

- Display Course list
- Display Course Information
- Has Enrolled student
- Admin can assign prof to a course or remove prof from course
- Admin can add student to course or remove student from course
- Course can be a prereq for another course, and a prereq for a specialization

PART C Classes, Class diagrams and Class Relationships

PRESIDENT ITEMS

- Faculty Members ID:
- Name: String, 24 char

ADMIN ITEMS:

- Faculty ID
- Name: String, 24 char
- Username
- Password

PROFESSOR ITEMS

- course List : vector <Course>
- Faculty ID
- Name
- Username
- Password

STUDENT ITEMS

- Student ID: UINT, 8 digits
- Name: String, 20 char
- Grades:List of courses, 2 char percentage
- Tuition Fees, unsigned long
- Eng Specialization: 20 char string
- Enrolled courses: list of courses, char string course number eg. APSC
- Username
- Password

CLASSES:

- User
    - President
    - Admin
    - Professor

    - Student
- Course
- Specialization
- Computer
- System Database