

Team: The Iterators

Idea and Requirement for AUST Developer Hunt

Ashique Imran (AIUB)
Sazzed Rahman Choudhury (AUST)

Raihan Al-mamun (AIUB)
Ahmed Anando Anabil (AIUB)

Introduction:

We have studied various national and international academic websites. Here's our proposal for AUST Developer Hunt

Development tools

1. Laravel
2. Bootstrap
3. VueJs
4. HTML5 , Vanilla JavaScript and CSS

Overview

This web app is for both student and official personnel of AUST. There are 2 fundamental parts of the site.

First public landing page which is accessible for all without any signup or login where news and events, notices, research informations, messages from deans and vice chancellor, academic calendar will be available. This page will be maintained by one register.

Secondly there is a portal to get access into it login credentials will be necessary. Initial login credential will be provided by the super admin and admin. Total working process is given below.

Features and Sitemap

Total Actors:

- I. Super Admin (1 or 2)
- II. Admins (less than 10)
- III. Resistors (1 or 2)
- IV. Faculties (no limit)
- V. Accountants (less than 10)
- VI. Students (no limit)
- VII. Alumnus (no limit)

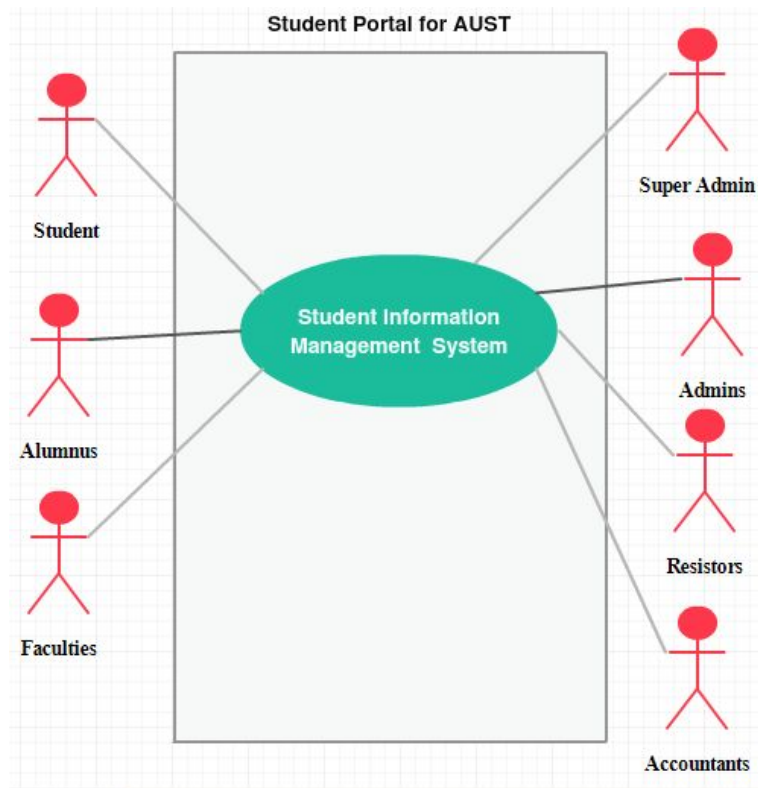
Use Case Scenario

Step-0	Step-1	Step-2	Main Actors
Student Information Management System	Authentication	Log in	Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus
		Log out	Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus
		Check Profiles	Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus
	Admission Processing	Publish admission notice	Admins
		Publish admission result	Admins
		Create student profiles	Admins
		Verify student profiles	Super admin
	Official works Maintenance	Publish general notice	Resistor
		Create faculty profiles	Admins
		Change profile informations	Admins
		Maintain financial records	Accountants
		Provide course	Resistor

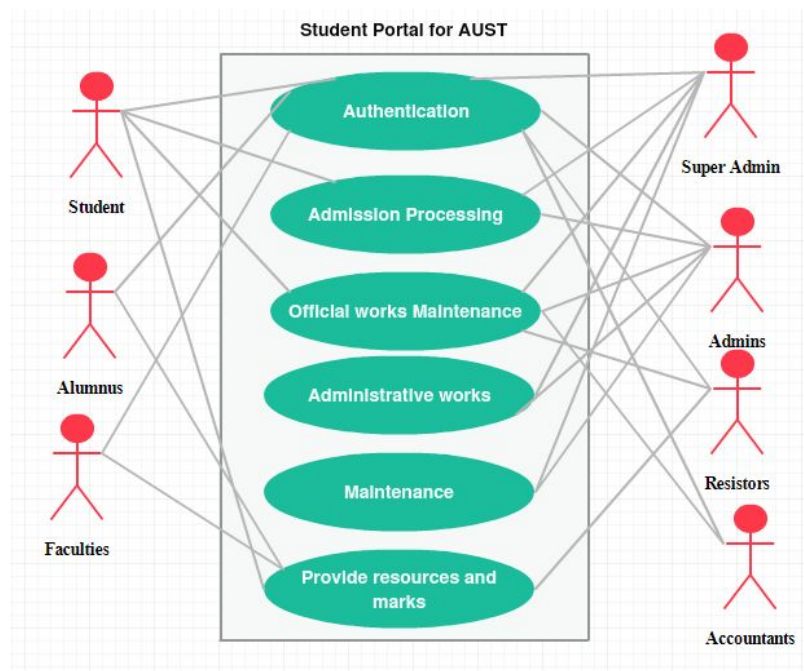
		outlines and academic calendars	
	Administrative works	Assigning students and faculties into courses	Admins
		Create Admin, Resistor and Accountants profiles	Super Admin
	Maintenance	Verify and approve all documents and profiles	Super Admin
		Block any profile	Super Admin, Admins
	Provide resources and marks	Provide Course Resource	Faculties
		Set marks submission deadline for faculties	Resistor
		Provide marks	Faculties

Use Case Description

To clarify the scenario we are going to illustrate use-case diagram, description and activity diagram. Use-case diagram given below:

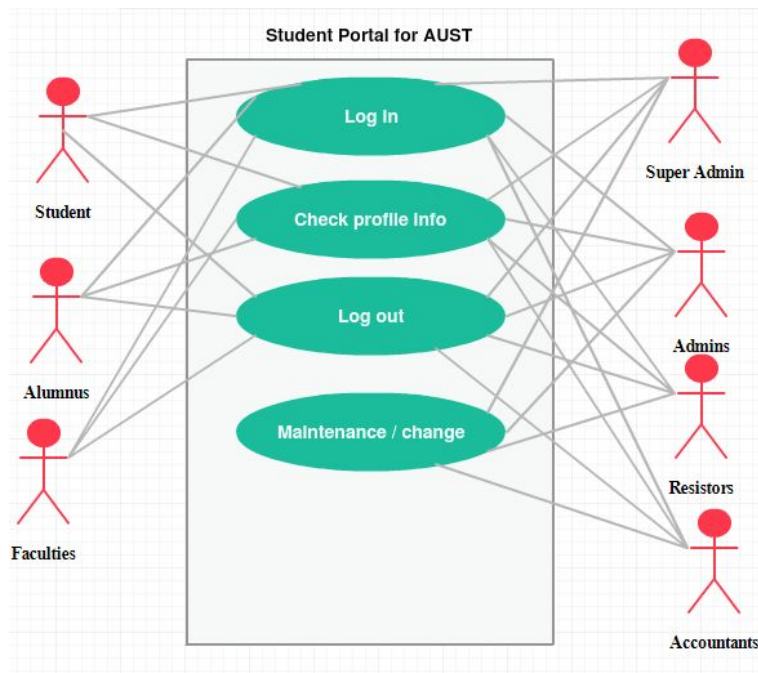


Following use-case diagram describing the relation between step-1 and all actors :



1. Authentication

This part can be divided into **4** subcategories : Login, Logout , Check self profile information(for all user), Maintenance or change



❖ Login:

Primary Actor: Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus

Precondition: Must have an account on this system.

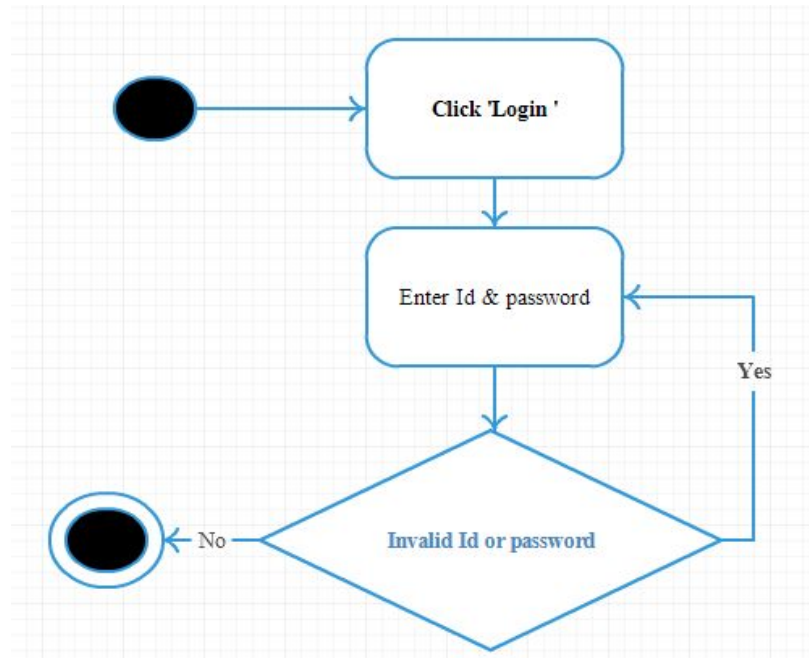
Goal in Context: To enter the system.

Scenario:

- A. All of the users will have an Id (unique)
- B. Visit the landing page .
- C. Click on 'Login' button.
- D. Input ID & password.
- E. Proceed to the next activity.

Exceptions:

- A. System error.
- B. Network error.
- C. Null information provided.
- D. Validation errors

**❖ Logout:**

Primary Actor: Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus

Goal in Context: To exit from the system.

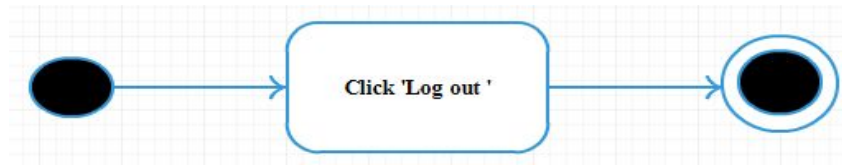
Scenario:

- A. Click the 'Logout' button.
- B. Get out of the system.

Priority: Essential, must be implemented.

Precondition: Must be logged in.

Frequency of Use: No time limits.



❖ Check Profile:

Primary Actor: Super Admin, Admins, Resistors, Faculties, Accountants, Students, Alumnus

Goal in Context: To Check one's information.

Precondition: Must be logged in.

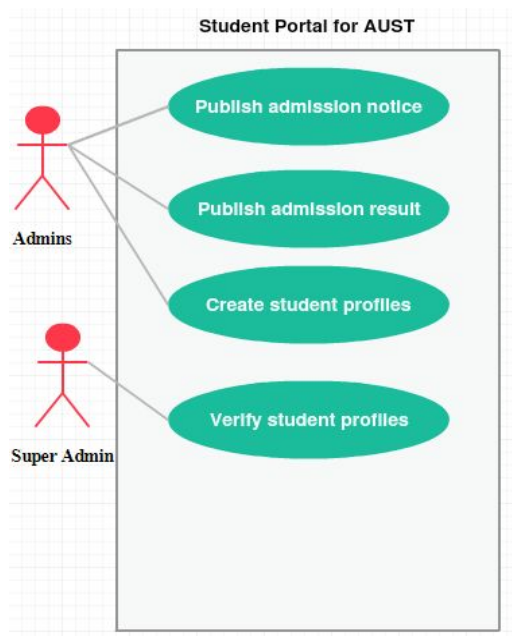
Scenario:

- A. Login to system
- B. To See the profile information press 'profile'
- C. To change password press 'change password'

Priority: Essential, must be implemented.

2. Admission Processing

This part has 4 parts as well.



❖ **Publish admission notice:**

Primary Actor: Admin

Goal in Context: Provide information on admission process.

Scenario:

- A. Sign in to the system.
- B. Click on 'Admission Process'.
- C. Select 'Publish Admission Notice'.
- D. Attach notice file.
- E. Upload the file.

Exceptions:

- A. System is not responding.
- B. File format mismatch.
- C. Unable to upload.

Priority: Essential, must be implemented.

Frequency of Use: Once per semester.

❖ **Publish admission result:**

Primary Actor: Admin

Goal in Context: Provide result of admission process.

Scenario:

- A. Sign in to the system.
- B. Click on 'Admission Process'.
- C. Select 'Publish Result'.
- D. Attach result file.
- E. Upload the file.

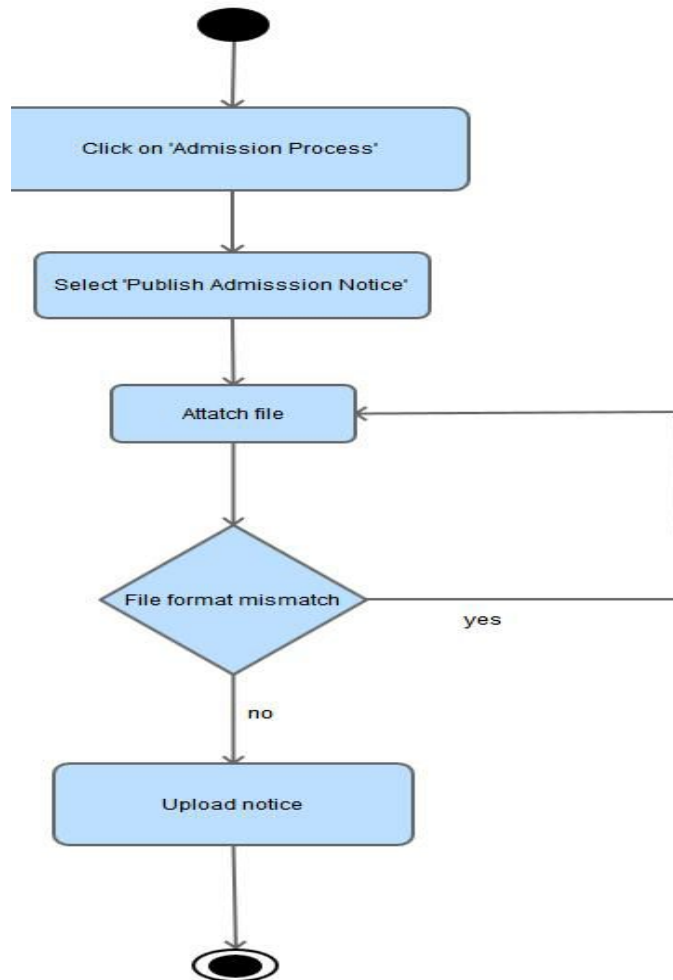
Exceptions:

- A. File size out of bound.
- B. File format mismatch.
- C. Unable to upload.

Priority: Essential, must be implemented.

Frequency of Use: Once per semester.

N.B For both of the parts diagram is same.



❖ **Create student profile:**

Primary Actor: Admin

Goal in Context: To register students to the system.

Scenario:

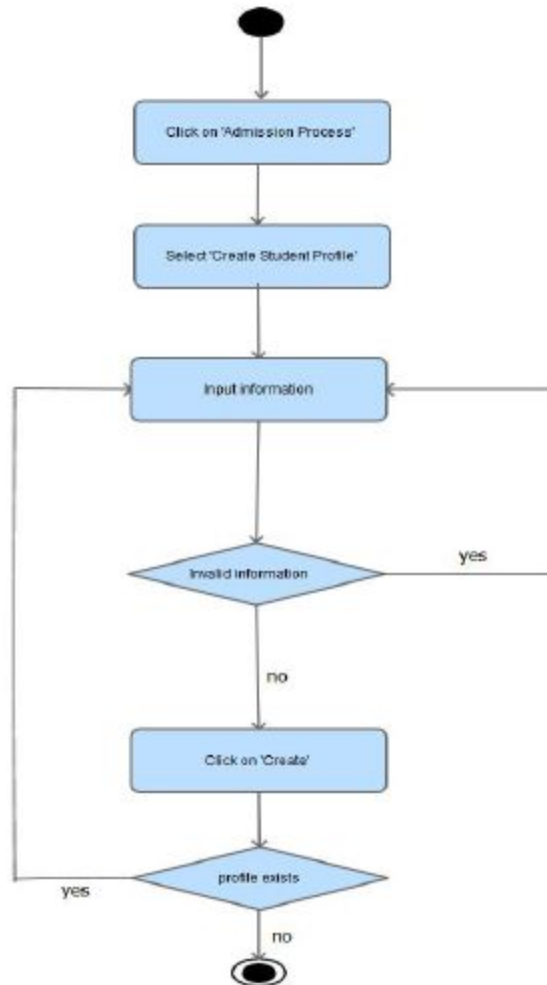
- A. Sign in to the system.
- B. Click on 'Admission Process'.
- C. Select 'Create Student Profile'.
- D. Input information.
- E. Click 'Create' button.

Exceptions:

- A. Profile already exists.

- B. Missing required information.
- C. Unable to create profile.

Priority: Essential, must be implemented.



❖ **Verify student profile:**

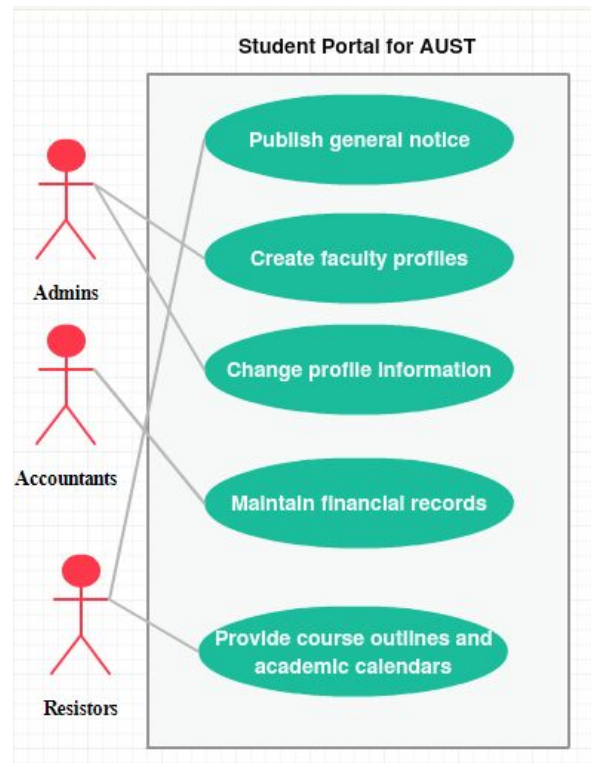
Primary Actor: Super Admin

Goal in Context: To register students to the system.

Scenario: Super admin checks all the students into and confirms the profile

3. Official works Maintenance

It has 5 partitions.



❖ **Publish general notice:**

Primary Actor: Resistor

Goal in Context: To provided academic information.

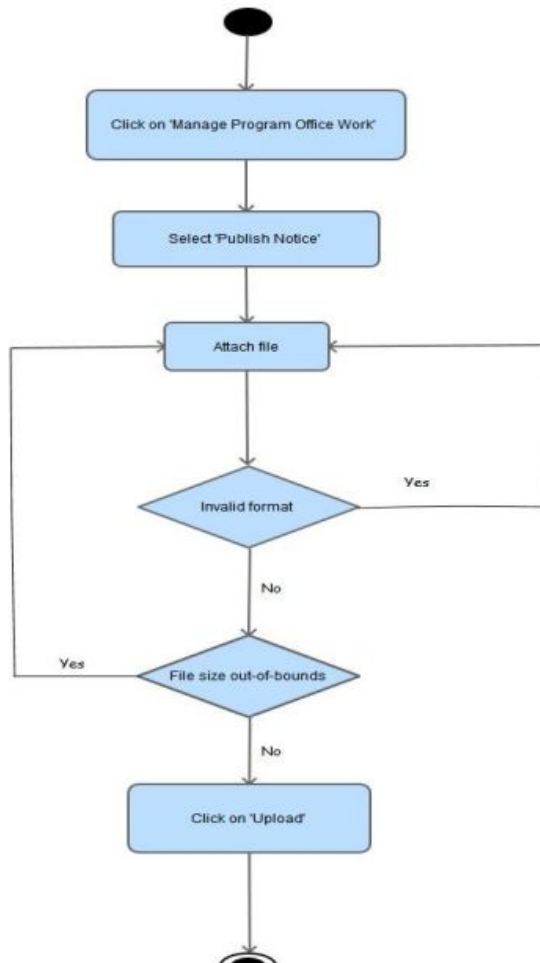
Scenario:

- A. Sign in to the system.
- B. Click on 'Manage Program Office Work'.
- C. Select 'Publish Notice'.
- D. Attach required files.
- E. Click 'Upload' Button.

Exceptions:

- A. File formats miss match.
- B. Unable to attach files.
- C. File size is too large to upload.

Priority: Essential, medium important to have dynamic notice board.



❖ **Add course outlines and academic calendar:**

Primary Actor: Resistor.

Goal in Context: Provide students and course teachers with course details.

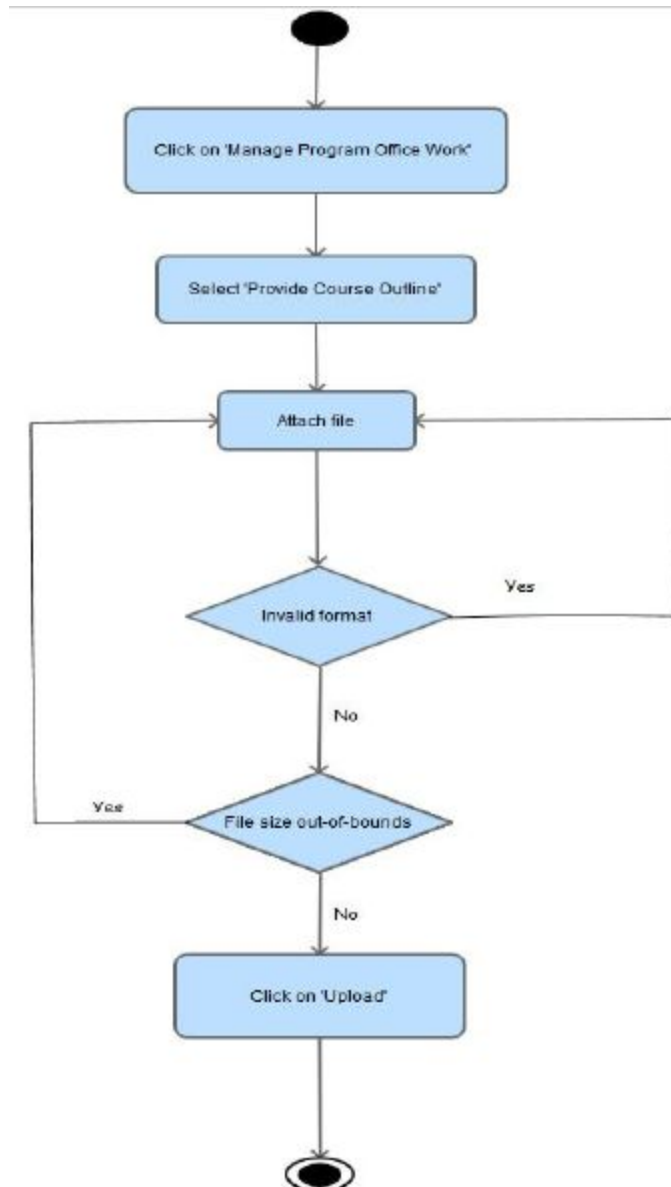
Scenario:

- A. Sign to the system.
- B. Click on 'Manage Program Office Work'.
- C. Select 'Provide Course Outline' or 'Academic calendar'.
- D. Attach file.
- E. Click 'Upload' button.

Exceptions:

- I. File format mismatch.
- II. Unable to attach file.
- III. File size is too large to upload.

Priority: Essential, must be implemented.



❖ **Create faculty profile:**

Primary Actors: Admin

Goal in Context: To register students to the system.

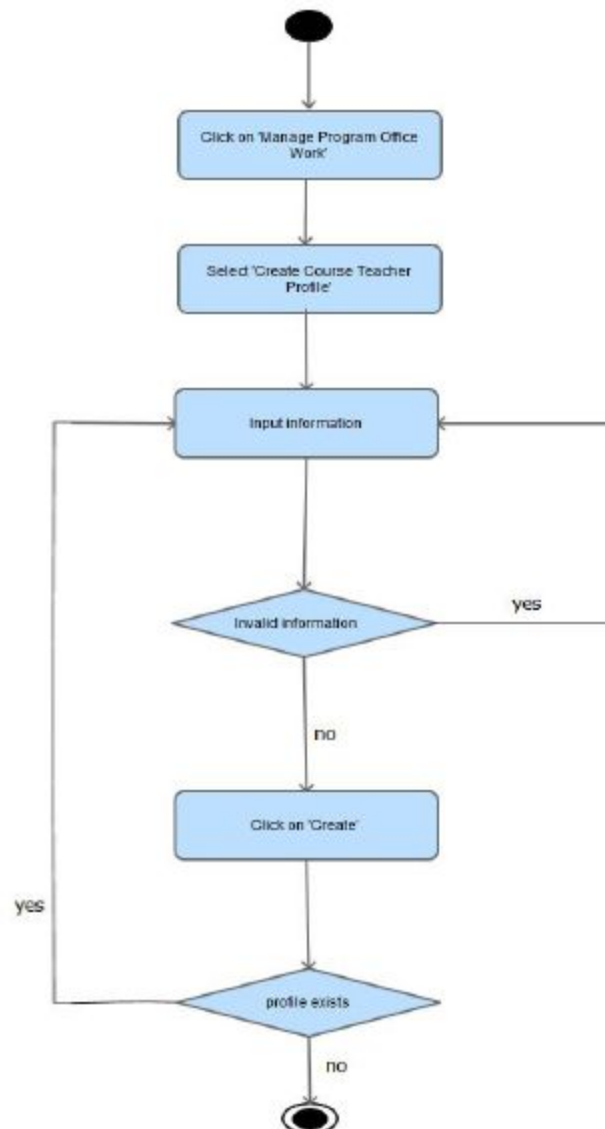
Scenario:

- A. Sign in to the system.
- B. Click on 'Manage Program Office Work'.
- C. Select 'Create Course Teacher Profile'.
- D. Input information.
- E. Click 'Create' button.

Exceptions:

- A. Profile already exists.
- B. Missing required information.
- C. System error.
- D. Unable to create profile.

Priority: Essential, must be implemented.



❖ **Change profile informations:**

Primary Actors: Admin

Goal in Context: To give the opportunity to edit profile information.

Scenario:

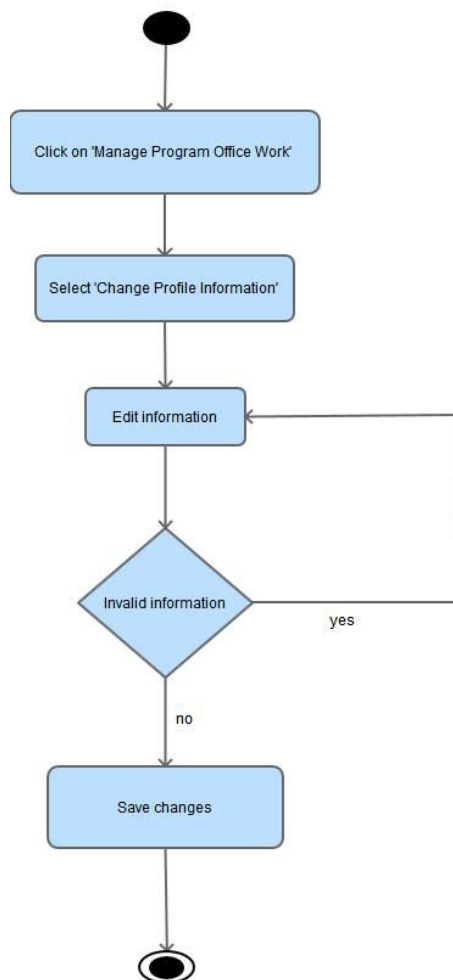
-
- A. Sign in to the system.
 - B. Click on 'Manage Program Office Work'.
 - C. Select 'Change Profile Information'.
 - D. Edit the information.
 - E. Confirm changes.

Exceptions:

- E. System error.
- F. Network error.
- G. Null information provided.
- H. Unable to update the changes.
- I. Validation errors

Priority: Essential, must be implemented.

Frequency of Use: Many times per day.



❖ **Maintain financial records:**

Primary Actor: Accountant.

Goal in Context: To keep the records of financial activities that students perform.

Scenario:

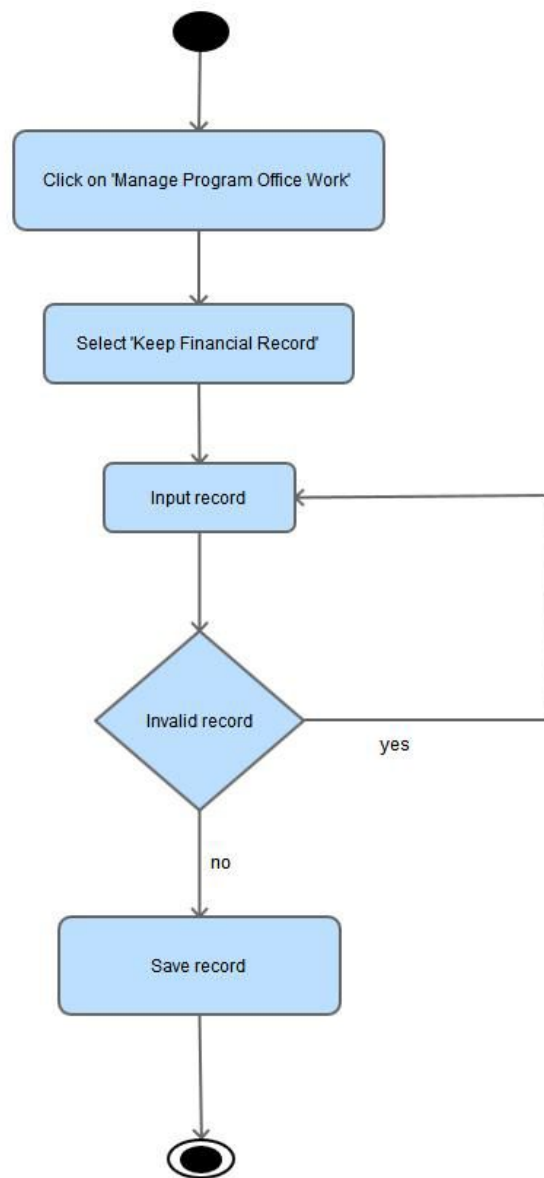
- A. Sign in to the system.
- B. Click on 'Manage Program Office Work'.
- C. Select 'Keep Financial Record'.
- D. Input the record.
- E. Update the record.

Exceptions:

- A. Null exception.
- B. System error.
- C. Wrong input.
- D. Unable to update.

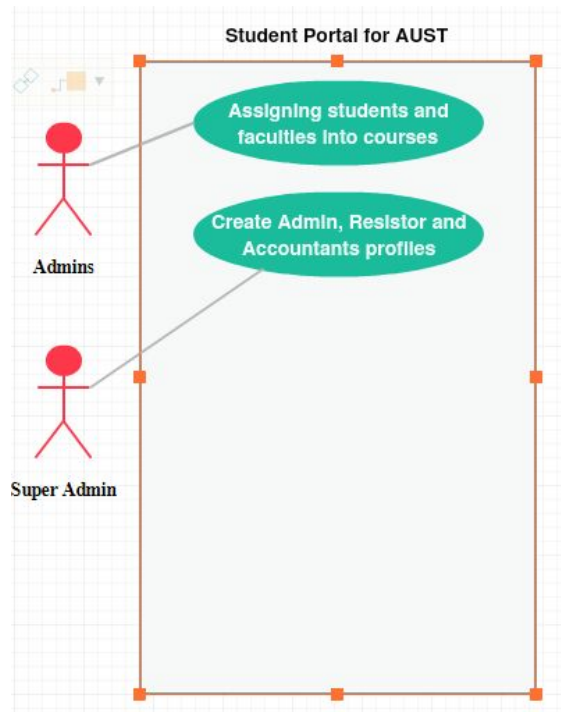
Priority: Essential, must be implemented.

Frequency of Use: Several times per day.



4. Administrative works

There are 2 partition in this section.



❖ **Assigning students and faculties into courses:**

Primary Actor: Admin

Goal in Context: To assign students and faculty to the system.

Scenario:

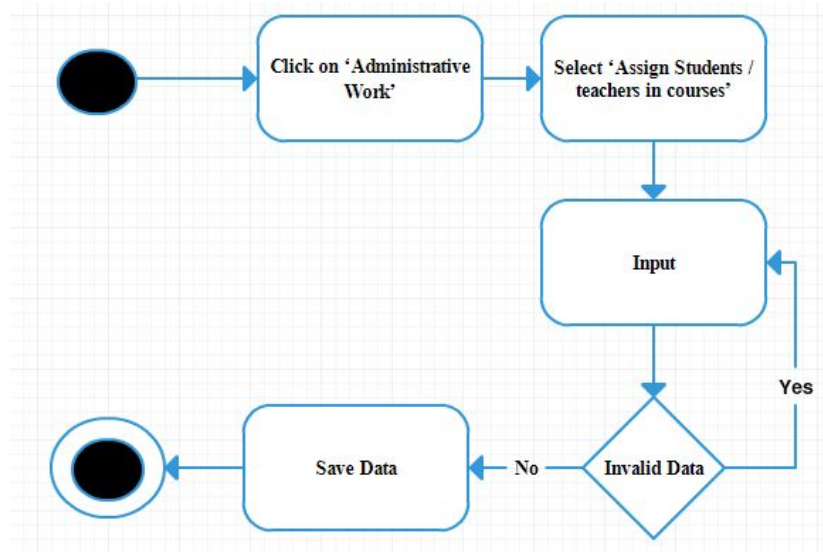
- A. Sign in to the system.
- B. Click on 'Administrative Work' button.
- C. Select 'Assign Students / teachers in courses'.
- D. Input information.
- E. Update the information.

Exceptions:

- A. Null point error.
- B. Wrong information provided.
- C. Cannot update the information.

Priority: Essential, must be implemented.

Frequency of Use: Once in every semester.



❖ **Create Admin, Resistor and Accountants profiles:**

Primary Actor: Super Admin

Goal in Context: To add admin, resistor and accountants profiles into the system.

Scenario:

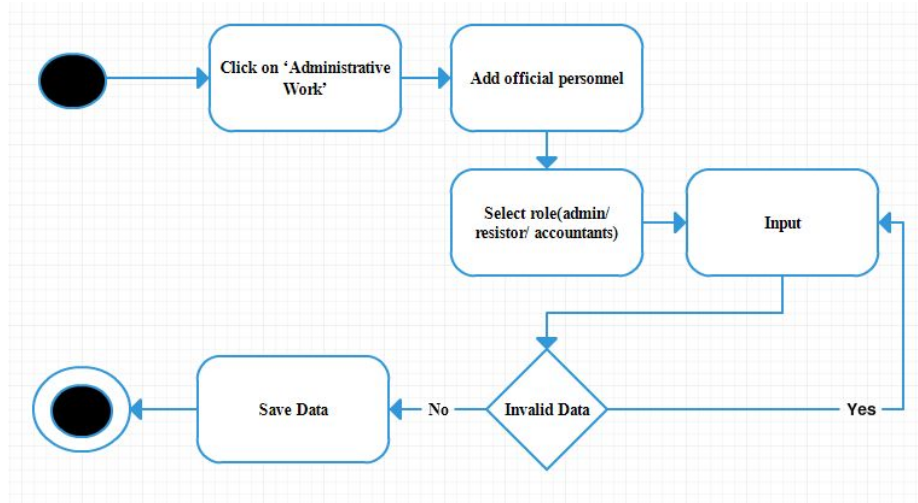
- A. Sign in to the system.
- B. Click on 'Administrative Work' button.
- C. Select 'Add official personnel'.
- D. Select the role from the dropdown (admin/ resistor /accountants)
- E. Input information.
- F. Update the information.

Exceptions:

- A. Null point error.
- B. Wrong information provided.
- C. Cannot update the information.

Priority: Essential, must be implemented.

Frequency of Use: Very limited use in a year.



5. Maintenance

Here we have 2 sub sections as well.

❖ **Verify and approve all documents and profiles:**

Primary Actor: Super Admin

Goal in Context: To Verify and approve all datum.

Scenario:

Any major changes by official personnel to be verified.

Exceptions:

- A. Null point error
- B. Validation error
- C. Connection error

Priority: Can be implemented.

Frequency of Use: Once in a while.

❖ **Block any profile:**

Primary Actor: Super admin , Admin

Goal in Context: To block any profile.

Scenario:

- A. Login to system
- B. Search the profile by Id
- C. Click 'Details'
- D. Click 'Block'
- E. Update information

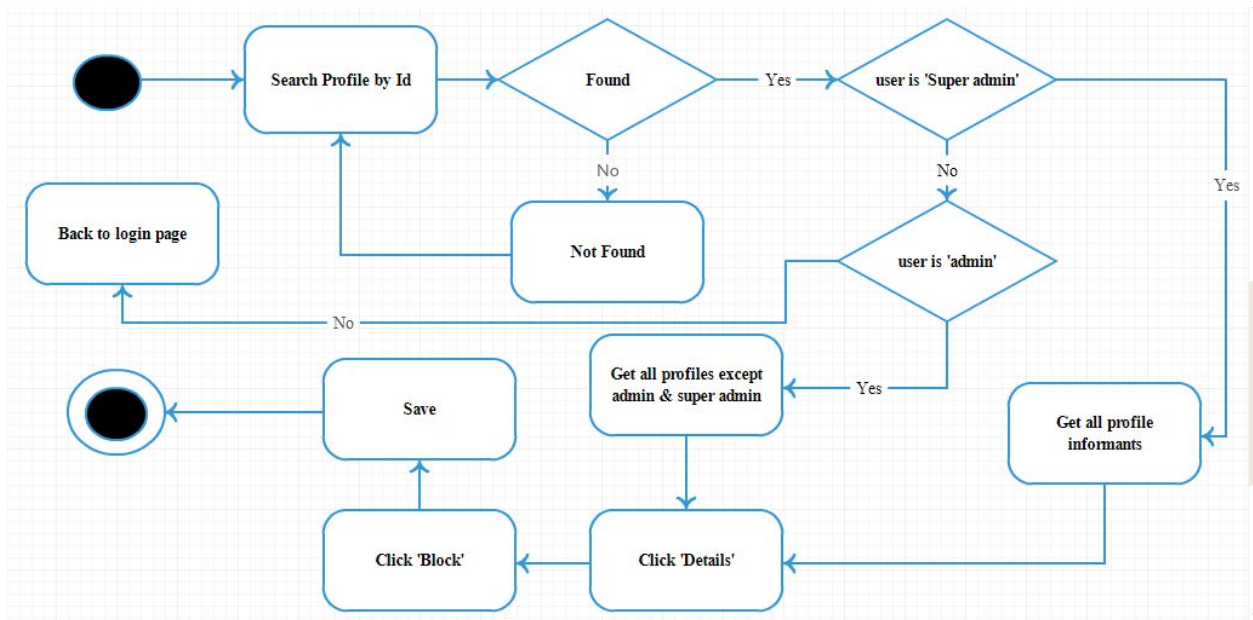
- F. A super admin can block all accounts but an admin can not block any admin or super admin profile

Exceptions:

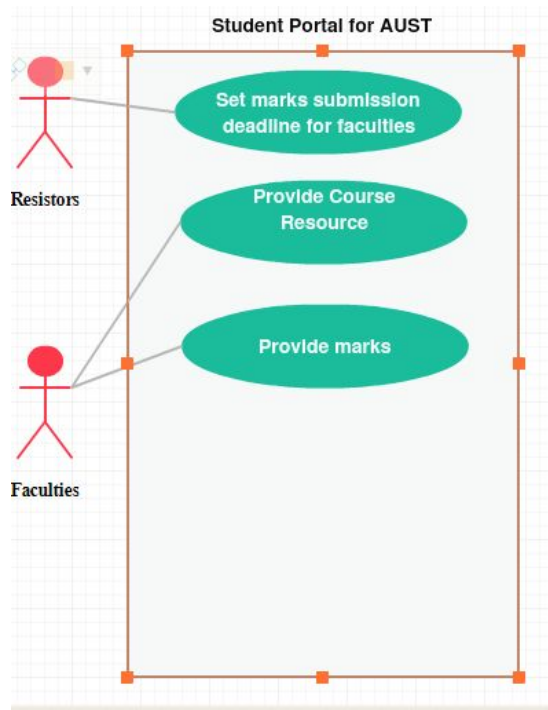
- A. Already blocked
- B. Deleted profile
- C. Connection error

Priority: Essential, must be implemented.

Frequency of Use: As per necessity, many times or not at all in a semester.



6. Provide resources and marks



Primary Actor: Course Teacher.

Goal in Context: Provide the students with course related resources like- *.pdf, *.txt, *.ppt, *.doc & *.jpg, *.png, *.jpeg

Scenario:

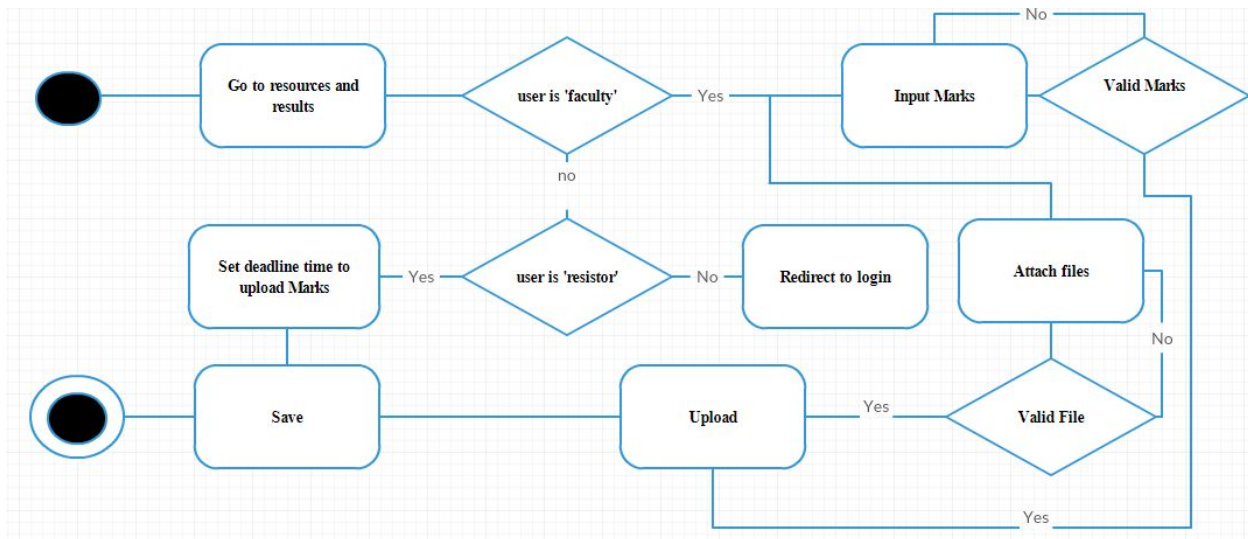
- A. Sign in.
- B. Resistor can set a deadline for uploading marks
- C. Click on 'Provide Course Resource & Marks'.
- D. Press 'Upload Resource' button.
- E. Press 'Attach' button.
- F. Browse and select file.
- G. Press 'Upload' button.

Exceptions:

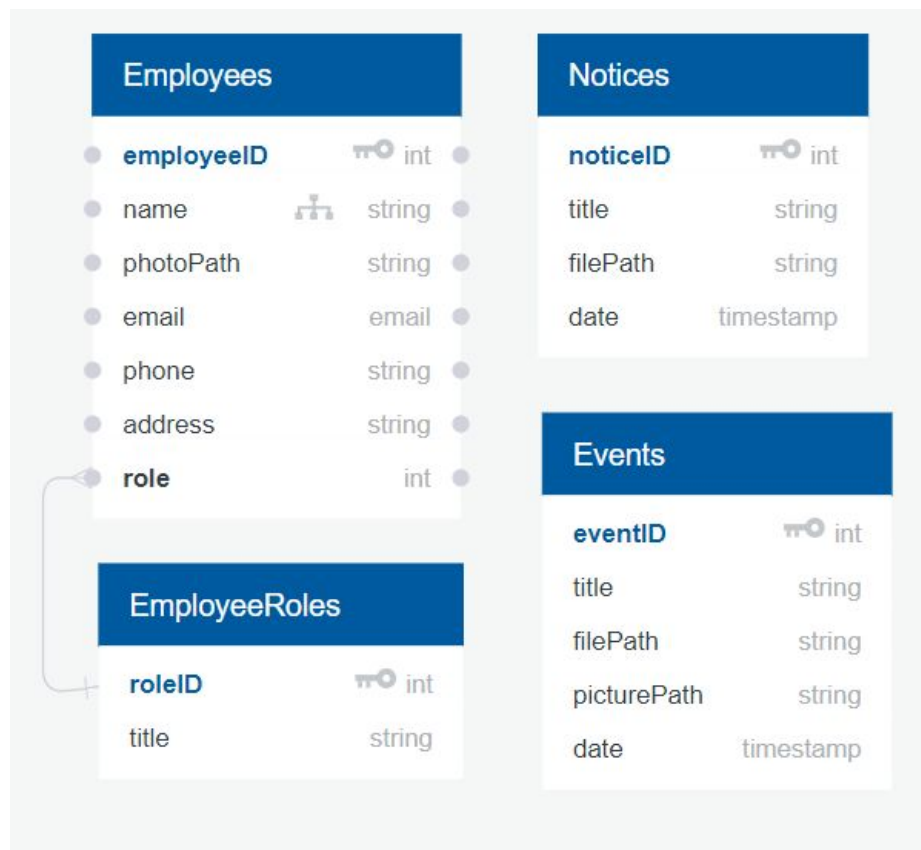
- A. Network problem.
- B. Connection problem.
- C. System problem.
- D. Data size is out-of-bound (4 MB).
- E. File is a possible threat (like *.exe or *.jar file).
- F. Negative marks input

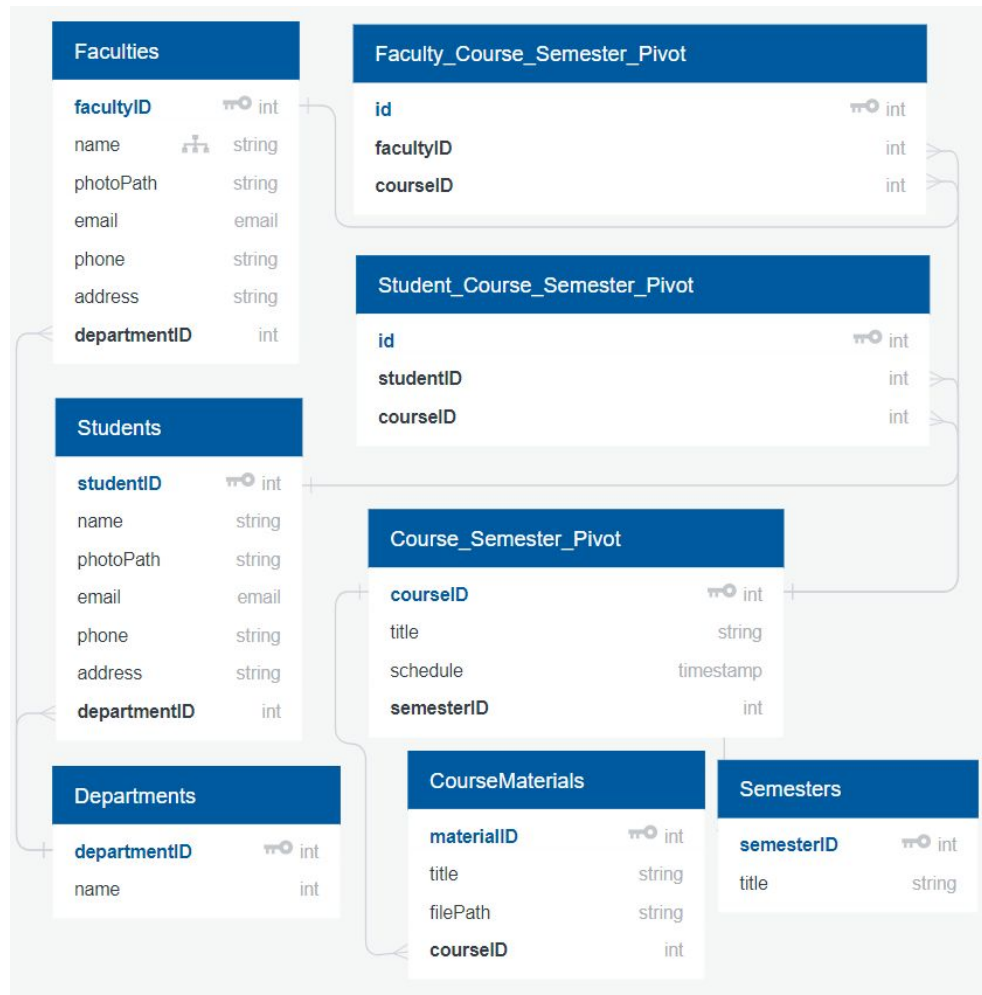
Priority: Moderate, may be implemented.

Frequency of Use: Several times during a course.



Schema Diagrams





Security issues

- ★ Advance data encryption will be used to protect passwords and session/cookie data.
- ★ Cross-site request forgery (CSRF) token to be used in every input form to prevent cross-site foreign request, malicious and unauthorized commands.
- ★ All data from the users will be validated both in frontend and backend.
- ★ Data validation, sanitization and Object-relational mapping (ORM) will minimize the chance of SQL injection.
- ★ Model view controller(MVC) architecture gives additional data and path security.
- ★ All notes uploaded by faculties will be kept in laravel resources directory which will not be accessed in public without system generated download link.