Contents

[BTR490 System 2016 2](#_Toc453357801)

[(1) System Name and Team Number 2](#_Toc453357802)

[(2) Team Members 2](#_Toc453357803)

[(3) Client and/or Mentor 2](#_Toc453357804)

[(4) System Overview 2](#_Toc453357805)

[(5) Technology to be used 2](#_Toc453357806)

[(6) Stakeholders 2](#_Toc453357807)

[(7) A Diagrammatic Representation of the System 3](#_Toc453357808)

[(8) System Components 4](#_Toc453357809)

[(9) Who Did What Log 8](#_Toc453357810)

# BTR490 System 2016

## System Name and Team Number

* Seneca College Academic Honesty Procedure
* Team #19

## Team Members

* Mona Alkhulaqi
* Jagmeet Bhamber
* Shawn Mathew

## Client and/or Mentor

* Mentor: Marcel Jar, Professor at the School of Information & Communications Technology at Seneca College
* Clients: Tim McKenna, Professor at the School of Information & Communications Technology at Seneca College

## System Overview

A web platform to automate the academic honesty procedures at Seneca College. After completing the proposed system, it will help the committee make the processing and tracking of academic honesty violations less cumbersome. Also, it should help faculty submit any incidents to the committee, making the process quicker and fairer. The system will allow instructors to enter an offense, allow committee and faculty members to view student’s previous incidents, allow faculty to edit incidents, and allow students to appeal for accused offences.

## Technology to be used

* ASP.NET
* SQL
* Javascript

## Stakeholders

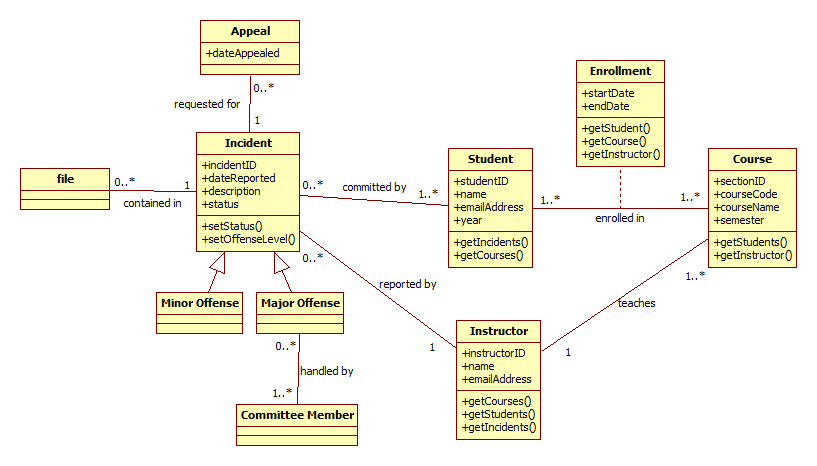
|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Chairman of academic honesty | Authority – the final decision maker of the situation |
| Faculty | Authority – will decide to use and upload the suspicious documents and release any warning if available |
| ICT | Sponsor of the project |
| Students | End user – their documents or assignments will be uploaded if the faculty have any suspicions about their work and a warning will be sent to them if applicable |
| Registration Office | End user – the final decision will be sent to them from Chairman to be officially released |
| Technical | Project developer and builder |

## A Diagrammatic Representation of the System

See diagrams below in 8) System components

## System Components

Class Diagram:



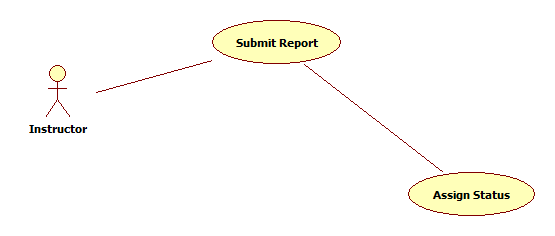
* Incident Reporting
* Viewing Student’s Previous Incidents
* Editing Existing Incidents
* Requesting Appeal

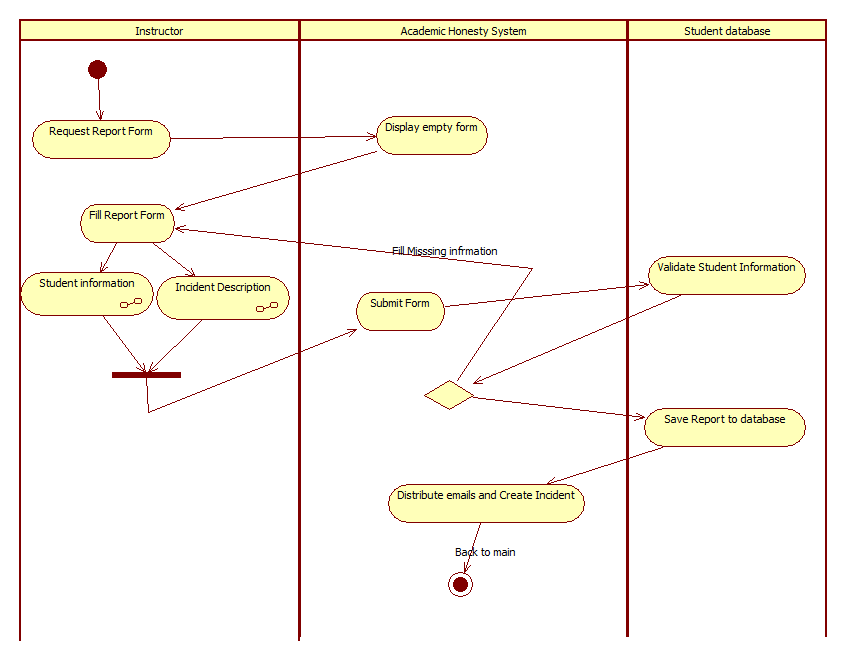
### Component 1: Incident Reporting

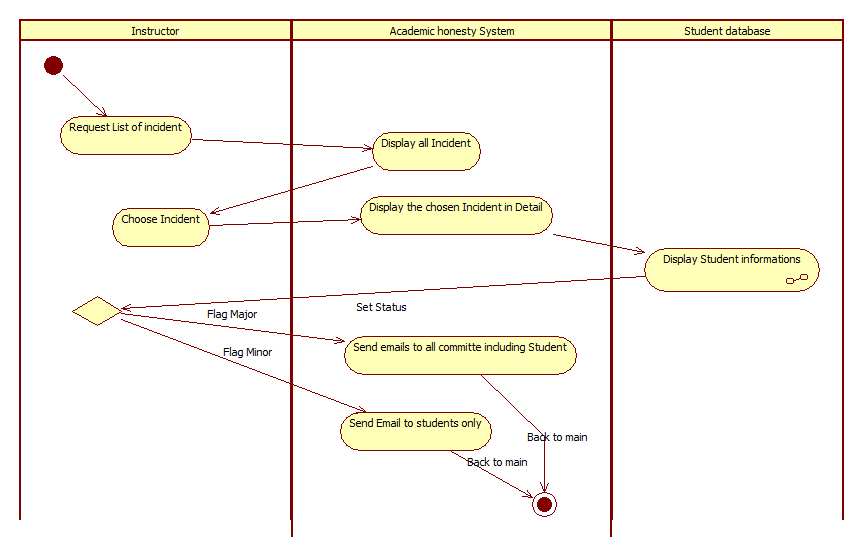
An instructor reports an incident of academic dishonesty. He/she will later flag it as minor/major once the incident is resolved.

Business rules:

* An instructor may only submit reports for students in his/her classes.
* A report consists of the following mandatory information: At least 1 student (with name, student number, and email address) and a description of the incident.







Submit Report:

Main flow:

Precondition: Instructor logged in, system active

|  |  |  |
| --- | --- | --- |
|  | Actor: Instructor | System |
| 1 | Request to submit report | Display form consisting of fields for student(s), description, and file uploads. Form contains mandatory fields of student (with name, number, and email), and description. Additional students and pdf/text documents are optional. |
| 2 | Fill in form with at least 1 student (name, number, email) and a description, then submit | Save report to database, adds incident to involved students’ records, and sends email to involved student(s) |

Alternate flows:

|  |  |  |
| --- | --- | --- |
|  | Alternate flow | Description |
| A1 | Not all mandatory fields were filled | User provided no student and/or no description. System re-displays form (keeping the previous input) and prompts user that there must be at least 1 student and a description. |
| A2 | Invalid student info was entered. | Student info (name, number, email) did not match any student found in the database. System re-displays form (keeping previous input) and highlights name, number, and email for the student not found. |
| A3 | Unsupported file type was uploaded | User uploads a file type that is not a pdf or text file. System re-displays form (keeping previous input) and notifies user that only pdf and text files are allowed |

Assign status:

Main flow:

Precondition: Instructor logged in, system active, incident has been created with no status

|  |  |  |
| --- | --- | --- |
|  | Actor: Instructor | System |
| 1 | Request list of incidents | Display list of all incidents involving the instructor, with description, and whether they have a status applied already or not. |
| 2 | Select an incident to view in detail | Display incident and all of its data: all students (with their names, student numbers, emails), description, attached files. |
| 2 | Flag offense level as major | Incident is flagged as a major offense. An email is sent to all committee members and involved students stating that this incident was flagged as a major offense. This email also contains description and students’ names. |

Alternate flows:

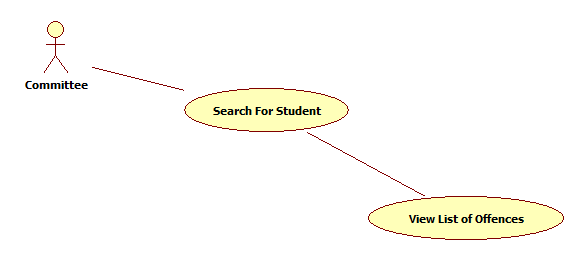
|  |  |  |
| --- | --- | --- |
|  | Alternate flow | Description |
| A1 | Flag offense level as minor | Incident is flagged as a minor offense. An email is sent to involved students only stating that this incident was flagged as a minor offense. Email also contains description and students’ names. |

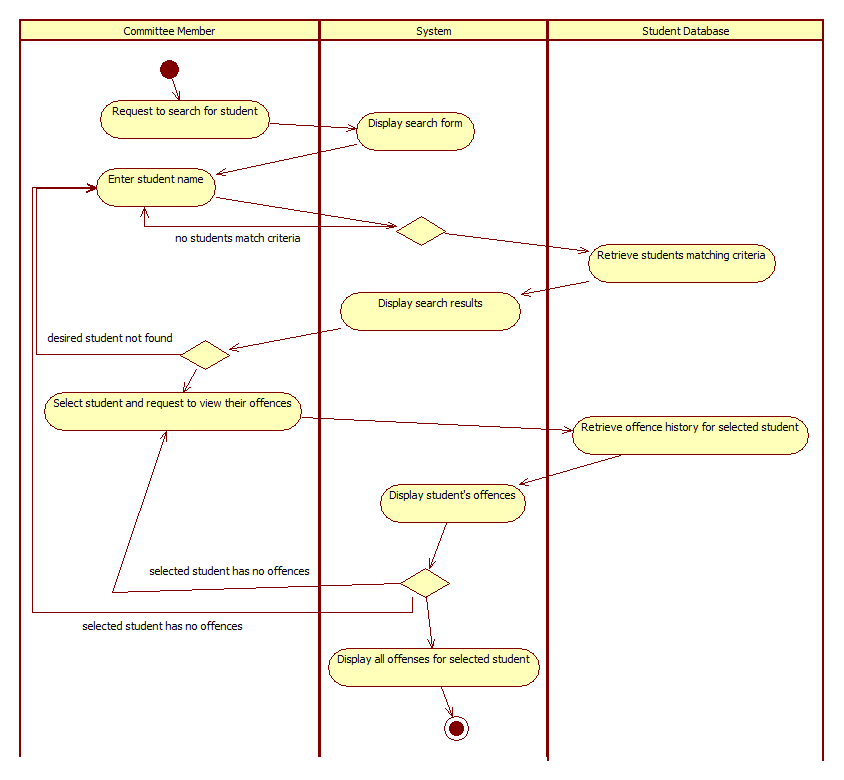
### Component 2: Viewing Student’s Previous Incidents

A committee member views a student’s history of offenses.

Business rules:

* Only committee members have access to this component





Search for Student:

Main flow:

Precondition: Committee member logged in, system active

|  |  |  |
| --- | --- | --- |
|  | Actor: Committee member | System |
| 1 | Requests to search for student | Displays search bar |
| 2 | Enters student name and searches | Displays all students that match search criteria, with their names and student numbers |
| 3 | Selects student and requests to view their offenses | Displays all offenses (with description, instructor, other students, and flag) recorded for the selected student |

Alternate flows:

|  |  |  |
| --- | --- | --- |
|  | Alternate flow | Description |
| A1 | No student matches searched name | System displays “no users found” and prompts actor to search again (step 2) |
| A2 | Desired student not found in search | If the user does not find the desired student, they may choose to search again (step 2) |
| A3 | Student has no offenses | System displays that the student has aclean record, and prompts actor to return to search results (step 3) or search again (step 2) |

Test Cases

1. Test case Summary

|  |  |
| --- | --- |
| **Scenario Name** | **Scenario Description** |
| Main: Successful Search | User successfully reaches desired student’s offence history page |
| Alt: no students found | No students match user’s search criteria |
| Alt: desired student not found | Desired student not found in search results |
| Alt: selected student has no offences | The student is found but they do not have any offences recorded |

1. Test Case Details

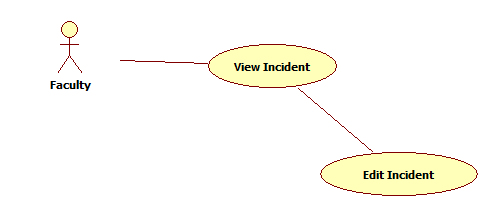
| **Test Case** | **Scenario** | **Data values / conditions being tested** | **Expected Result** |
| --- | --- | --- | --- |
|  | Main: Successful Search | Entered student’s name exists in database, student has committed offences | Student displayed among search results; student profile displays all of their offences |
|  | Alt: No students found | Requested student does not match any names in student database | Empty search results displayed; option to re-search displayed |
|  | Alt: Desired student not found | Requested student not found in search results | Search results displayed; desired student is not found in search results; option to re-search displayed |
|  | Alt: Selected student has no offences | Entered name exists in database, student has no offences | Student displayed in search results; student profile displays “no offences committed”; option to return to search results displayed |

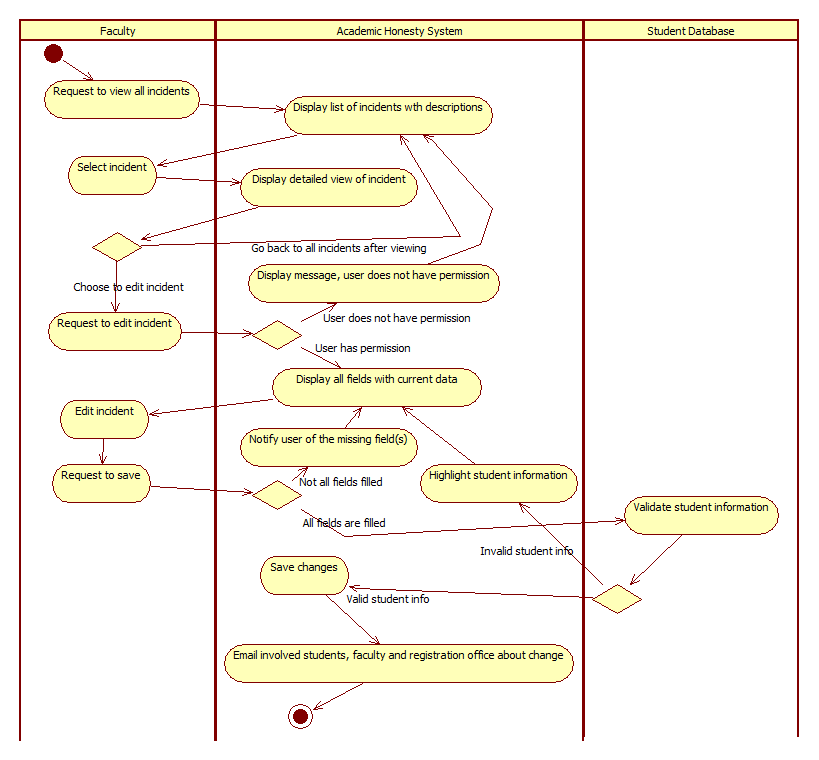
### Component 3: Editing Existing Incidents

A committee or faculty member edits any existing incidents.

Business rules:

* Open cases may be edited by faculty and committee
* Closed cases may only be edited by the committee
* Instructor may not be edited





Edit Incident:

Main flow:

Precondition: Faculty member logged in, system active

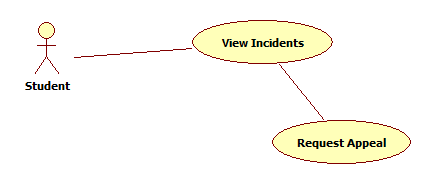
|  |  |  |
| --- | --- | --- |
|  | Actor: Faculty member or Committee member | System |
| 1 | Request to see all incidents | Display list of all incidents, with their descriptions only. |
| 2 | Select incident to view in detail | Displays detailed view of incident, including student names and numbers, instructor, course, description, and attached files |
| 3 | Requests to edit incident | System displays all fields (student names and numbers, instructor, course, description, and files) available to be edited, and loads them with current data. Instructor is greyed out and cannot be edited. |
| 4 | Makes adjustments to any fields that need to be updated, and requests to save | System saves changes to database, and sends email to involved students, committee and faculty members, and registration office informing that the incident (identified in the email by its description) has been edited. |

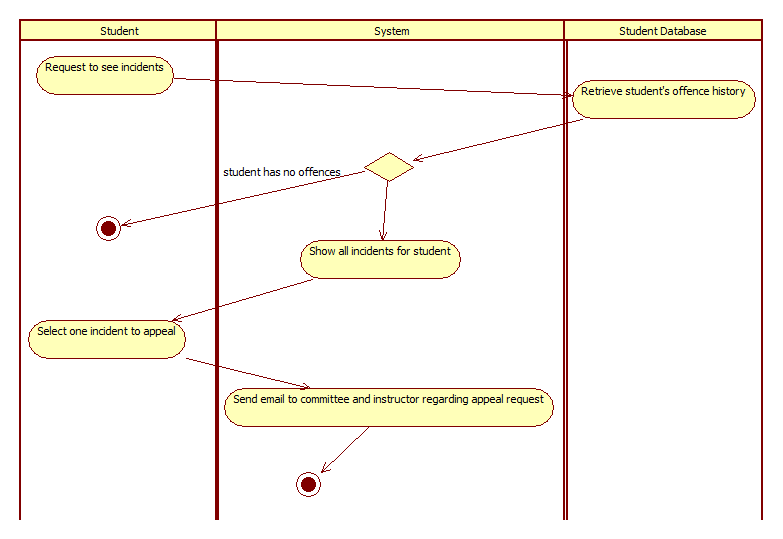
Alternate flows:

|  |  |  |
| --- | --- | --- |
|  | Alternate flow | Description |
| A1 | Incident does not have all required fields filled (no description, only 1 student) | User leaves blank for description and/or removes students. System redisplays form and notifies user that there is no info for the missing field, and that it is required (step 4) |
| A2 | Invalid student info was entered. | Student info (name, number, email) did not match any student found in the database. System re-displays form (keeping previous input) and highlights the name, number, and email for the student not found. (step 4) |
| A3 | Actor does not want to edit the incident, only view it | After viewing the details for the incident, the actor may choose to return to list of all incidents (step 1) |
| A4 | Actor does not have permission to change the case (only committee members may edit closed cases) | System notifies actor that they do not have permission to edit incident, redirects them to Incident Details page (step 2) |

### Component 4: Requesting Appeal

A student requests an appeal for any offenses they’ve committed





Request Appeal:

Main flow:

Precondition: Student logged in, system active

|  |  |  |
| --- | --- | --- |
|  | Actor: Student | System |
| 1 | Request to see all incidents | Display list of all incidents that the student was involved in, with their description, instructor, course, flags, and names of other involved students. |
| 2 | Request to appeal a selected incident | Sends an email to the involved instructor, as well as committee members that the student would like to appeal the case. Email contains student name and number, and incident description. |

Alternate flows:

|  |  |  |
| --- | --- | --- |
|  | Alternate flow | Description |
| A1 | No incidents for the student | Student is not convicted of any offenses, system displays message that the user has a clean record and prompts them to exit the system. |

## Who Did What Log

|  |  |
| --- | --- |
| Team Member | Work completed on this Project (be specific please) |
| Jagmeet Bhamber | Use case specifications |
| Shawn Mathew | Use case Diagrams |
| Mona Alkhulaqi | Stakeholders and their roles |
| Prabhjot Kaur | Technology to be used and detailed explanation of all use cases |