**User stories**

“As a user, I want to rank my choices for second year discipline so that I can be placed in my preferred program.”

“As a program coordinator, I want to place students into engineering disciplines so that they can continue their degrees.”

“As a professors, I want to be able to report a cheating incident so that the cheating student can be disciplined.”

“As a faculty Dean, I want to be able to suspend students who have cheated so that they will not want to cheat again.”

|  |
| --- |
| Use case: **Login**   * Ask user for username/password   + Extend use case: **Register user** * User enters their username and password * User presses “Login” button * System checks database for username * If username not found <<Scenario 1>> * System compares the entered password with the one in the database, * If password incorrect <<Scenario 2>> * User granted access   Scenario 1: System display “username not found” message  Scenario 2: System display “incorrect password” message |

|  |
| --- |
| Use case: **Register user**   * User clicks “Register” button * User prompted for email address, new password, and UBC student/employee ID * User clicks “Submit” button * System requests verification of UBC student/employee ID from *UBC Information Database* * If *UBC Information Database* does not have the entered ID <<Scenario 1>> * Account is created and user is returned to login screen   Scenario 1: System display “student/employee ID not found” message |

|  |
| --- |
| Use case: **View disciplines**   * Include **Login** * System checks user’s program and year level * If user is not in first year engineering <<Scenario 1>> * Student clicks “View Disciplines” button * System shows student a list of all available engineering disciplines, with a description of each   Scenario 1: System hides “View disciplines” button |

|  |
| --- |
| Use case: **Rank disciplines**   * Include **View Disciplines** * If student does not have academic standing to continue to second year <<Scenario 1>> * If student has already submitted their rankings <<Scenario 2>> * Student clicks “Rank disciplines” button * Student uses drop down boxes to rank each discipline from first choice to last choice * Student clicks “submit rankings” button * System stores user’s program rankings in a database   Scenario 1: System hides “Rank disciplines” button  Scenario 2: System displays “Rankings submitted” message |

|  |
| --- |
| Use case: **View courses**   * Include **Login** * If user is not a student who is eligible to register for courses <<Scenario 1>> * User clicks “View courses” button * System shows student a list of available courses at UBC with description of each   Scenario 1: System hides “View courses button” |

|  |
| --- |
| Use case: **Register course**   * Include **View courses** * User clicks on course they wish to register in * If user’s program, prerequisites, and academic standing do not make them eligible for the course <<Scenario 1>> * If registration period for selected course is not open <<Scenario 2>> * Student clicks “Register course” * System adds student to course and updates database   Scenario 1: System tells student why they are ineligible to register this course and hides “Register course” button  Scenario 2: System displays “Registration period closed” message and hides “Register course” button |

|  |
| --- |
| Use case: **View financial summary**   * Include **Login** * User clicks “Financial summary” button * System displays user’s outstanding dues and payment history |

|  |
| --- |
| Use case: **Pay tuition**   * Include **View financial summary** * User clicks “Pay now” button * System shows user payment page * User inputs payment information * If payment fails, <<Scenario 1>> * System displays “payment successful” message and updates student’s outstanding dues in database to reflect payment   Scenario 1: System notifies user that payment has failed and gives them the option to try again |

|  |
| --- |
| Use Case: **View list of first year students**   * Include **Login** * If user is not Engineering program coordinator <<Scenario 1>> * User clicks “View list of first year students” button * System displays a list of the names, student IDs, and academic standing of every first year student in engineering   + Extend use case: **View list of students ranked by grades**   + Extend use case: **View students’ ranked program choices**   Scenario 1: System hides “View list of first year students” button |

|  |
| --- |
| Use Case: **View list of students ranked by grades**   * System displays “Rank students by grades” button * User clicks “Rank students by grades” button * System shows user list of students ranked from highest to lowest GPA across all first year classes |

|  |
| --- |
| Use Case: **View student’s ranked program choices**   * System displays “View ranked program choices” button next to the name of each student * User clicks “View ranked program choices” button for a student * System shows user a list of the student’s program rankings from first to last |

|  |
| --- |
| Use Case: **Assign student to discipline**   * Include **View list of first year students** * System displays “Assign student to discipline” button next to the name of each student * User clicks “Assign student to a discipline” button for a student * System shows user a list of engineering disciplines that have not yet reached capacity * User clicks on the discipline they want to assign the student to * System saves the user’s selection in a database and updates the student list to show the student’s assigned discipline |