Balham College Requirements

**There may be errors or ambiguities in this information – please seek clarification from the college if necessary. Also, please note that this information may change later.**

Balham College requires an application to administer its student intake, enrolments, and programmes. The college wants an application that is easy for non-technical people to use and able to handle and recover any data lost in the event of a failure.

The application system is to be used by enrolments clerks, human resources clerks, course administrators, research administrators, student support clerks, and programme administrators. They all agree that they require the size of text boxes to reflect the maximum size of the corresponding data.

The human resources clerks are responsible for adding lecturers’ details and producing the lecturers report. The human resources clerks are also responsible for updating existing lecturers’ details and deleting the details of lecturers who have left the college.

The enrolments clerks are responsible for adding students’ details and producing the report on students (which lists all of the details of all the students). Once students have been added, the enrolments clerks can enrol them onto courses. The enrolments clerks are also responsible for updating existing students’ details and for deleting the details of students. Students who have at least one enrolment cannot be deleted.

The programme administrators are responsible for adding courses and producing the courses report. Also the programme administrators are responsible for updating existing course and for deleting courses. After a course is added, the programme administrators can assign lecturers to it. The programme administrators are also responsible for maintaining the details of the programmes (programme name and level).

The course administrators are responsible for adding assessments, entering results, and producing the assessments report. Also the course administrators are responsible for updating existing assessments and for deleting assessments. Research administrators are responsible for adding, updating and deleting research topics, adding and removing research projects and producing the research projects report.

The student support clerk is responsible for recording students’ issues and for producing the students’ issues report.

A lecturer can be associated to zero to many assignments and an assignment is associated to one lecturer. A course can be associated to zero to many assignments and an assignment is associated to one course.

A course can be associated to zero to many enrolments and an enrolment is associated to one course. A student can be associated to zero to many enrolments and an enrolment is associated to one student.

A course can be associated to zero to many assessments and an assessment is associated to one course. A programme can be associated to zero to many courses and a course is associated to one programme.

An assessment can be associated to zero to many results and a result is associated to one assessment. An enrolment can be associated to zero to many results and a result is associated to one enrolment.

A lecturer can be associated to zero to many research projects and a research project is associated to one lecturer. A research topic can be associated to zero to many research projects and a research project is associated to one research topic.

An enrolment can be associated with zero to many issues and an issue is associated to one enrolment.

1. Add Student

Use Case ID: 1

Description: This use case enables the enrolment clerk to add a student’s details.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Add Student” function.
* The system displays the “Add Student” form with all fields blank.
* The enrolments clerk enters the student id (1 to 99999999 inclusive), last name, first name, street address, suburb, city, email address, phone number, and status (part-time or full-time).
* The system either confirms that all necessary fields are filled in correctly and saves the student’s details or the system identifies any missing or incorrect fields and prompts for the completion of all fields.
* The system displays the “Student added successfully” message.
* The system displays the “Add another student?” prompt.
* The enrolments clerk either elects to end the use case and the system closes the form, or elects to add another student and the system displays the “Add Student” form with all fields blank.

1. Update Student

Use Case ID: 2

Description: This use case enables the enrolments clerk to update a student’s details.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Update Student” function.
* The system displays the “Update Student” form and displays a list of all the students (student id, last name and first name).
* The enrolments clerk either selects the student whose details need updating or elects to cancel the operation and the system closes the form.
* The system displays the student’s details (student ID, last name, first name, street address, suburb, city, email address, phone number, and status).
* The enrolments clerk updates the relevant details (last name, first name, street address, suburb, city, email address, phone number, and status only) and elects to update the student.
* The system validates the entries in the fields and, either asks for confirmation to change the student’s details, or identifies missing or incorrect fields and prompts for completion of the entry.
* The enrolments clerk either confirms the change of details or elects to cancel the changes and the system closes.
* The system saves the student’s details.
* The system displays the “Student updated successfully” message.
* The system displays the “Update another student?” prompt.
* The enrolments clerk either elects to end the use case and the system closes the form, or elects to update another student and the system displays the “Update student” form and displays a list of all the students (student id, last name and first name).

1. Delete Student

Use Case ID: 3

Description: This use case enables the enrolments clerk to delete a student’s details.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Delete Student” function.
* The system displays the “Delete Student” form and displays a list of all the students (student id, last name and first name) that have no enrolments.
* The enrolments clerk selects the student who requires deleting or the enrolments clerk elects to cancel the operation and the system closes the form.
* The system displays the student’s details (student ID, last name, first name, street address, suburb, city, email address, phone number, and status).
* The enrolments clerk either elects to delete the student or elects to cancel the operation and the system closes the form.
* The system deletes the student.
* The system displays the “Student deleted successfully” message.
* The system displays the “Delete another student?” prompt.
* The enrolments clerk either elects to end the use case and the system closes the form, or the enrolments clerk elects to delete another student and the system displays the delete student form and displays a list of all the students (student id, last name and first name) that have no enrolments.

1. Produce Students Report

Use Case ID: 4

Description: This use case enables the enrolments clerk to produce the students report.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Produce Students Report” function
* The system displays the “Students Report” form
* The enrolments clerk either selects the option to generate the report or elects to cancel, without generating the report, and the system closes the form.
* The system gets the details (student ID, last name, first name, street address, suburb, city, phone number, email address, and status) of each student.
* The system gets the details (year, semester, status, course name, and programme name) of each enrolment for each student.
* The system then generates the students report (student ID, last name, first name, street address, suburb, city, phone number, email address, and status and year, semester, status, course name, and programme name of each student’s enrolment) with the students sorted by first name within last name.
* The system concludes the use case by closing the form.

1. Enrol Student

Use Case ID: 5

Description: This use case enables the enrolments clerk to enrol a student onto a course.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Enrol Student” function.
* The system displays the “Enrol Student” form and displays a list of all the students (student id, last name and first name).
* The enrolments clerk either selects the student who is enrolling or elects to cancel the operation and the system closes the form.
* The system displays the student’s details (student id, last name, and first name, street address, suburb, and city).
* The system displays the list of the enrolments (enrolment number, year, semester, and course name) for the selected student.
* The system displays a list of the courses (course id, and name) that are currently running.
* The enrolments clerk selects a course.
* The system displays the course’s details (course id, name, credits and programme name).
* The enrolments clerk enters the enrolment number (1 to 999999999999 inclusive), year (2020 to 2050 inclusive), semester (1 to 3 inclusive), and status (pending, paid, or complete) of the enrolment.
* The enrolments clerk elects to enrol the student on to the course or elects to cancel the operation and the system closes the form.
* The system saves the enrolment’s details (enrolment number, student, year, semester, status, and course) unless the student has already enrolled on the same course for the same year and same semester in which an error message is displayed.
* The system displays the “Student enrolled successfully” message.
* The system displays the “Enrol another student?” prompt.
* The enrolments clerk either elects to end the use case and the system closes the form, or the enrolments clerk elects to enrol another student and the system displays the “Enrol Student” form and displays a list of all the students (student id, last name and first name).

1. Withdraw Student

Use Case ID: 6

Description: This use case enables the enrolments clerk to withdraw a student from a course.

Priority: Medium

Pre-conditions: The enrolments clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The enrolments clerk selects the “Withdraw Student” function.
* The system displays the “Withdraw Student” form and displays a list of all the students (student id, last name and first name) that have enrolments.
* The enrolments clerk either selects the student who is withdrawing or elects to cancel the operation and the system closes the form.
* The system displays the student’s details (student id, last name, and first name, street address, suburb, and city).
* The system displays the list of the enrolments (enrolment number, year, semester, and course name) for the selected student.
* The enrolments clerk selects an enrolment.
* The enrolments clerk elects to withdraw the student from the course or elects to cancel the operation and the system closes the form.
* The system deletes the enrolment details.
* The system displays the “Student withdrawn successfully” message.
* The system displays the “Withdraw another student?” prompt.
* The enrolments clerk either elects to end the use case and the system closes the form, or the enrolments clerk elects to withdraw another student and the system displays the “Withdraw Student” form and displays a list of all the students (student id, last name and first name) that have enrolments.

1. Add Lecturer

Use Case ID: 7

Description: This use case enables the human resources clerk to add a lecturer’s details.

Priority: Medium

Pre-conditions: The human resources clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The human resources clerk selects the “Add Lecturer” function.
* The system displays the “Add Lecturer” form with all fields blank.
* The human resources clerk enters the lecturer’s id (1 to 9999 inclusive), last name, first name, street address, suburb, city, phone number, email address, ranking (A or B or C), and type (academic or contract).
* The system either confirms that all necessary fields are filled in correctly and saves the lecturer’s details, or identifies any missing or incorrect fields and prompts for the completion of all fields.
* The system displays the “Lecturer added successfully” message.
* The system displays the “Add another lecturer?” prompt.
* The human resources clerk either elects to end the use case and the system closes the form, or elects to add another lecturer and the system displays the “Add Lecturer” form with all fields blank.

1. Update Lecturer

Use Case ID: 8

Description: This use case enables the human resources clerk to update a lecturer’s details.

Priority: Medium

Pre-conditions: The human resources clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The human resources clerk selects the “Update Lecturer” function.
* The system displays the “Update Lecturer” form and displays a list of all the lecturers (lecturer id, last name and first name).
* The human resources clerk either selects the lecturer whose details require updating or elects to cancel the operation and the system closes the form.
* The system displays the lecturer’s details (lecturer ID, last name, first name, street address, suburb, city, phone number, email address, ranking, and type).
* The human resources clerk updates the relevant details (last name, first name, street address, suburb, city, phone number, email address, ranking, and type only).
* The system validates the entries in the fields and either asks for confirmation to change the lecturer’s details or identifies missing or incorrect fields and prompts for completion of the entry.
* The human resources clerk either confirms the change of details or cancels the changes and the system closes the form.
* The system saves the lecturer’s details.
* The system displays the “Lecturer updated successfully” message.
* The system displays the “Update another lecturer?” prompt.
* The human resources clerk either elects to end the use case and the system closes the form, or elects to update another lecturer and the system displays the “Update lecturer” form and displays a list of all the lecturers (lecturer id, last name and first name).

1. Delete Lecturer

Use Case ID: 9

Description: This use case enables the human resources clerk to delete a lecturer’s details.

Priority: Medium

Pre-conditions: The human resources clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The human resources clerk selects the “Delete Lecturer” function.
* The system displays the “Delete Lecturer” form and displays a list of all the lecturers (lecturer id, last name and first name) who have no course assignments.
* The human resources clerk either selects the lecturer who requires deleting or elects to cancel the operation and the system closes the form.
* The system displays the lecturer’s details (lecturer ID, last name, first name, street address, suburb, and city).
* The human resources clerk either elects to delete the lecturer or elects to cancel the operation and the system closes the form.
* The system deletes the lecturer.
* The system displays the “Lecturer deleted successfully” message.
* The system displays the “Delete another lecturer?” prompt.
* The human resources clerk either elects to end the use case and the system closes the form, or elects to delete another lecturer and the system displays the “Delete Lecturer” form and displays a list of all the lecturers (lecturer id, last name, and first name) who have no course assignments.

1. Produce Lecturers Report

Use Case ID: 10

Description: This use case enables the human resources clerk to produce the lecturers report.

Priority: Medium

Pre-conditions: The human resources clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The human resources clerk selects the “Lecturers Report” function
* The system displays the “Lecturers Report” form.
* The human resources clerk either selects the option to generate the report or elects to cancel without generating the report and the system closes the form.
* The system gets the details (lecturer id, last name, first name, street address, suburb, city, phone number, email address, ranking, and type) of each lecturer.
* The system counts the number of course assignments for each lecturer.
* The system then generates the lecturers report (lecturer id, last name, first name, street address, suburb, city, phone number, email address, ranking, type, and count of assignments) with the lecturers sorted by first name within last name.
* The system concludes the use case by closing the form.

1. Add Course

Use Case ID: 11

Description: This use case enables the programme administrator to add a course’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Add Course” function.
* The system displays the “Add Course” form with all fields blank.
* The programme administrator enters the course’s details (course id (1 to 9999 inclusive), course name, credits (5 to 120 inclusive), fee (200.00 to 8000.00 inclusive), and status (current or suspended)).
* The system either confirms that the details are filled in correctly or identifies that the details are missing or incorrect and prompts for the completion of the details.
* The system displays a list of programmes (programme name and level).
* The programme administrator selects a programme.
* The programme administrator either elects to add the course or elects to cancel the operation and the system closes the form.
* The system saves the course’s details (course id, course name, credits, status, and programme).
* The system displays the “Course added successfully” message.
* The system displays the “Add another course?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to add another course and the system displays the “Add Course” form with all fields blank.

1. Update Course

Use Case ID: 12

Description: This use case enables the programme administrator to update a course’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Update Course” function.
* The system displays the “Update Course” form and displays a list of all the courses (course id and course name).
* The programme administrator either selects the course that has details that needs updating or elects to cancel the operation and the system closes the form.
* The system displays the course’s details (course id, course name, credits, status, fee, and programme name).
* The programme administrator updates the relevant details (name, credits, fee, and status only).
* The system validates the entries in the fields and either prompts for confirmation to change the courses details or identifies missing or incorrect fields and prompts for completion of the entry.
* The programme administrator either confirms the change of details or cancels the changes and the system closes the form.
* The system saves the course’s details.
* The system displays the “Course updated successfully” message.
* The system displays the “Update another course?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to update another course and the system displays the “Update Course” form and displays a list of all the courses (course id and course name).

1. Delete Course

Use Case ID: 13

Description: This use case enables the programme administrator to delete a course’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Delete Course” function.
* The system displays the “Delete Course” form and displays a list of all the courses (course id and course name) that have no lecturers, assessments, or enrolments assigned to them.
* The programme administrator either selects the course that requires deleting or elects to cancel the operation and the system closes the form.
* The system displays the course’s details (course id, course name, credits, and status).
* The programme administrator either elects to delete the course or elects to cancel the operation and the system closes the form.
* The system deletes the course.
* The system displays the “Course deleted successfully” message.
* The system displays the “Delete another course?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to delete another course and the system displays the “Delete Course” form and displays a list of all the courses (course id and name) that have no lecturers, assessments, or enrolments assigned to them.

1. Produce Courses Report

Use Case ID: 14

Description: This use case enables the programme administrator to produce the courses report.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Courses Report” function.
* The system displays the “Courses Report” form.
* The programme administrator either selects the option to generate the report or elects to cancel without generating the report and the system closes the form.
* The system gets the details (course ID, course name, credits, status, fee, and programme name) of each course.
* The system counts the number of enrolments for each course.
* The system counts the number of assessments for each course.
* The system then generates the courses report (course ID, course name, credits, status, and programme name) with the course sorted by name.
* The system concludes the use case by closing the form.

1. Assign Lecturer

Use Case ID: 15

Description: This use case enables the programme administrator to assign a lecturer to a course.

Priority: Medium

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Assign Lecturer” function.
* The system displays the “Assign Lecturer” form and displays a list of all courses (course id and name).
* The programme administrator either selects the course to which the lecturer is to be assigned or elects to cancel the operation and the system closes the form.
* The system displays the course’s details (course id, name, and credits).
* The system displays a list of all lecturers (lecturer id, last name, and first name).
* The programme administrator selects a lecturer.
* The system displays the selected lecturer’s details (lecturer id, last name, first name, and ranking.
* The programme administrator enters a role (primary or secondary).
* The programme administrator either elects to assign the lecturer to the course or elects to cancel the operation and the system closes the form.
* The system saves the assignment (course, lecturer, and role) unless the lecturer has already been assigned to the course in which case an error message is displayed and the system displays a list of all lecturers (lecturer id, last name, and first name).
* The system displays the “Lecturer assigned successfully” message.
* The system displays the “Assign another lecturer?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to assign another lecturer and the system returns displays the “Assign Lecturer” form and displays a list of all courses (course id and name).

1. Remove Lecturer

Use Case ID: 16

Description: This use case enables the programme administrator to remove a lecturer from a course.

Priority: Medium

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Remove Lecturer” function.
* The system displays the “Remove Lecturer” form and displays a list of all the courses (course id and course name) that have lecturers assigned.
* The programme administrator selects a course.
* The system displays the course’s details (course id, course name, status, and programme name).
* The system displays a list of the lecturers (lecturer id, last name, first name, and role) assigned to the selected course.
* The programme administrator selects a lecturer.
* The programme administrator either elects to remove the lecturer from the course or elects to cancel the operation and the system closes the form.
* The system deletes the assignment’s details.
* The system displays the “Lecturer removed successfully” message.
* The system displays the “Remove another lecturer?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to remove another lecturer and the system returns displays the “Remove Lecturer” form and displays a list of all courses (course id and course name) that have lecturers assigned.

1. Add Assessment

Use Case ID: 17

Description: This use case enables the assessment coordinator to add an assessment’s details.

Priority: Medium

Pre-conditions: The course administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The assessment coordinator selects the “Add Assessment” function.
* The system displays the “Add Assessment” form with all fields blank.
* The system displays a list of courses (course id and name).
* The course administrator selects a course.
* The course administrator enters the assessment id (1 to 200 inclusive) assessment number (1 to 5 inclusive), assessment name, type (Assignment, Written Test, Practical Test, or Final Exam), weighting (10 to 100 inclusive) and maximum mark (50 to 200 inclusive).
* The system either confirms that all necessary fields are filled in correctly and saves the assessment’s details (assessment id, assessment number, assessment name, type, weighting, maximum mark, and course id) or the system identifies any missing or incorrect fields and prompts for the completion of all fields.
* The system displays the “Assessment added successfully” message.
* The system displays the “Add another assessment?” prompt.
* The course administrator either elects to end the use case and the system closes the form, or elects to add another assessment and the system displays the “Add Assessment” form with all fields blank.

1. Update Assessment

Use Case ID: 18

Description: This use case enables the assessment coordinator to update an assessment’s details.

Priority: Low

Pre-conditions: The course administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The course administrator selects the “Update Assessment” function.
* The system displays the “Update Assessment” form and displays a list of all the assessments (assessment id, assessment number and assessment name).
* The course administrator either selects the assessment whose details need updating or elects to cancel the operation and the system closes the form.
* The system displays the assessment’s details (assessment id, number, assessment name, type, weighting, maximum mark, and course name).
* The course administrator updates the relevant details (assessment number, assessment name, type, weighting, and maximum mark only) and elects to update the assessment.
* The system validates the entries in the fields and, either asks for confirmation to change the assessment’s details, or identifies missing or incorrect fields and prompts for completion of the entry.
* The course administrator either confirms the change of details or elects to cancel the changes and the system closes.
* The system saves the assessment’s details.
* The system displays the “Assessment updated successfully” message.
* The system displays the “Update another assessment?” prompt.
* The course administrator either elects to end the use case and the system closes the form, or elects to update another assessment and the system displays the “Update assessment” form and displays a list of all the assessments (assessment id, assessment number and assessment name).

1. Delete Assessment

Use Case ID: 19

Description: This use case enables the assessment coordinator to delete an assessment’s details.

Priority: Medium

Pre-conditions: The course administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The course administrator selects the “Delete Assessment” function.
* The system displays the “Delete Assessment” form and displays a list of all the assessments (assessment number, assessment name and course name) that do not have any results.
* The course administrator selects the assessment who requires deleting or the course administrator elects to cancel the operation and the system closes the form.
* The system displays the assessment’s details (assessment id, assessment number, assessment name, type, and course name).
* The course administrator either elects to delete the assessment or elects to cancel the operation and the system closes the form.
* The system deletes the assessment.
* The system displays the “Assessment deleted successfully” message.
* The system displays the “Delete another assessment?” prompt.
* The course administrator either elects to end the use case and the system closes the form, or the course administrator elects to delete another assessment and the system displays the delete assessment form and displays a list of all the assessments that do not have any results.

1. Produce Assessments Report

Use Case ID: 20

Description: This use case enables the assessment coordinator to produce the assessments report.

Priority: Medium

Pre-conditions: The course administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The course administrator selects the “Produce Assessments Report function
* The system displays the “Assessments Report” form
* The course administrator either selects the option to generate the report or elects to cancel, without generating the report, and the system closes the form.
* The system gets the details (assessment id, assessment number, assessment name, type, weighting, maximum mark, and course name) of each assessment.
* The system gets the details (mark, result date, and student last name and first name) of each result for each assessment.
* The system then generates the assessments report (assessment id, assessment number, assessment name, type, weighting, maximum mark, and course name and mark, result date, and student last name and first name for each result for each assessment) with the students sorted by first name within last name within assessment name within course name.
* The system concludes the use case by closing the form.

1. Enter Result

Use Case ID: 21

Description: This use case enables the course administrator to enter a result for a student against an assessment.

Priority: High

Pre-conditions: The course administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The course administrator selects the “Enter Result” function.
* The system displays the “Enter Result” form and displays a list of all assessments (assessment id, assessment number, assessment name and course name).
* The course administrator either selects the assessment that is to be assigned or elects to cancel the operation and the system closes the form.
* The system displays the assessment’s details (assessment number, assessment name, maximum mark, and course name) and then displays a list of all enrolments (student name, year, semester, and status) for the course of the selected assessment.
* The course administrator either selects an enrolment and enters a mark (0 to 200 inclusive) and a result date or elects to cancel the operation and the system closes the form.
* The system saves the result (assessment, enrolment, result date, and mark) unless a result has already been entered for assessment and enrolment in which an error message is displayed.
* The system displays the “Result entered successfully” message.
* The system displays the “Enter another result?” prompt.
* The course administrator either elects to end the use case and the system closes the form, or elects to enter another result and the system displays the “Enter Result” form and displays a list of all assessments (assessment id, assessment number, assessment name and course name).

1. Add Programme

Use Case ID: 22

Description: This use case enables the programme administrator to add a programme’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Add Programme” function.
* The system displays the “Add Programme” form with all fields blank.
* The programme administrator enters the programme’s details (programme id (1 to 50 inclusive), programme name and level (1 to 10 inclusive).
* The system either confirms that the details are filled in correctly or identifies that the details are missing or incorrect and prompts for the completion of the details.
* The programme administrator either elects to add the programme or elects to cancel the operation and the system closes the form.
* The system saves the programme’s details (programme id, programme name, and level).
* The system displays the “Programme added successfully” message.
* The system displays the “Add another programme?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to add another programme and the system displays the “Add Programme” form with all fields blank.

1. Update Programme

Use Case ID: 23

Description: This use case enables the programme administrator to update a programme’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Update Programme” function.
* The system displays the “Update Programme” form and displays a list of all the programmes (programme id and programme name).
* The programme administrator either selects the programme that has details that needs updating or elects to cancel the operation and the system closes the form.
* The system displays the programme’s details (programme id, programme name, and level).
* The programme administrator updates the relevant details (programme name and level only).
* The system validates the entries in the fields and either prompts for confirmation to change the programmes details or identifies missing or incorrect fields and prompts for completion of the entry.
* The programme administrator either confirms the change of details or cancels the changes and the system closes the form.
* The system saves the programme’s details.
* The system displays the “Programme updated successfully” message.
* The system displays the “Update another programme?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to update another programme and the system displays the “Update Programme” form and displays a list of all the programmes (programme id and programme name).

1. Delete Programme

Use Case ID: 24

Description: This use case enables the programme administrator to delete a programme’s details.

Priority: High

Pre-conditions: The programme administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The programme administrator selects the “Delete Programme” function.
* The system displays the “Delete Programme” form and displays a list of all the programmes (programme id and programme name) that have no courses assigned to them.
* The programme administrator either selects the programme that requires deleting or elects to cancel the operation and the system closes the form.
* The system displays the programme’s details (programme id, programme name, and level).
* The programme administrator either elects to delete the programme or elects to cancel the operation and the system closes the form.
* The system deletes the programme.
* The system displays the “Programme deleted successfully” message.
* The system displays the “Delete another programme?” prompt.
* The programme administrator either elects to end the use case and the system closes the form, or elects to delete another programme and the system displays the “Delete Programme” form and displays a list of all the programmes (programme id and name) that have no courses assigned to them.

1. Add Research Project

Use Case ID: 25

Description: This use case enables the research administrator to add a research to a lecturer.

Priority: Medium

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Add Research Project” function.
* The system displays the “Add Research Project” form and displays a list of all the lecturers (lecturer id, last name, and first name).
* The research administrator either selects the lecturer to add the research project to or elects to cancel the operation and the system closes the form.
* The system displays the lecturer’s details (lecturer id, last name, first name, and type).
* The system displays the research projects’ details (output, research project description, and research topic description) for each research project already linked to the selected lecturer.
* The system displays a list of the research topics (research topic id, research topic description, and impact).
* The research administrator selects a research topic.
* The research administrator enters the research project’s id (1 to 9999 inclusive), output, start date, and research project description.
* The research administrator either elects to add the research project or elects to cancel the operation and the system closes the form.
* The system saves the research project’s details (research project id, lecturer, research topic, output, start date, and research project description).
* The system displays the “Research project added successfully” message.
* The system displays the “Add another research project?” prompt.
* The research administrator either elects to end the use case and the system closes the form, or the research administrator elects to add another research project and the system displays the “Add Research Project” form and displays a list of all the lecturers (lecturer id, last name, and first name).

1. Remove Research Project

Use Case ID: 26

Description: This use case enables the research administrator to remove a research to a lecturer.

Priority: Medium

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Remove Research Project” function.
* The system displays the “Remove Research Project” form and displays a list of all the lecturers (lecturer id, last name, and first name) who have research projects.
* The research administrator either selects the lecturer to remove the research project from or elects to cancel the operation and the system closes the form.
* The system displays the lecturer’s details (lecturer id, last name, first name, and type).
* The system displays the research projects’ details (research project id, output, research project description, and research topic description) for each research project linked to the selected lecturer.
* The research administrator selects the research project to remove.
* The research administrator either elects to remove the research project or elects to cancel the operation and the system closes the form.
* The system deletes the research project’s details.
* The system displays the “Research project removed successfully” message.
* The system displays the “Remove another research project?” prompt.
* The research administrator either elects to end the use case and the system closes the form, or the research administrator elects to remove another research project and the system displays the “Remove Research Project” form and displays a list of all the lecturers (lecturer id, last name, and first name) who have research projects.

1. Add Research Topic

Use Case ID: 27

Description: This use case enables the research administrator to add a research topic’s details.

Priority: Low

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Add Research Topic” function.
* The system displays the “Add Research Topic” form with all fields blank.
* The research administrator enters the research topic’s id (1 to 99 inclusive), research topic description and impact (high, medium or low).
* The system either confirms that all necessary fields are filled in correctly and saves the research topic’s details or the system identifies any missing or incorrect fields and prompts for the completion of all fields.
* The system displays the “Research topic added successfully” message.
* The system displays the “Add another research topic?” prompt.
* The research administrator either elects to end the use case and the system closes the form, or elects to add another research topic and the system displays the “Add Research Topic” form with all fields blank.

1. Update Research Topic

Use Case ID: 28

Description: This use case enables the research administrator to update a research topic’s details.

Priority: Low

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Update Research Topic” function.
* The system displays the “Update Research Topic” form and displays a list of all the research topics (research topic id and research topic description).
* The research administrator either selects the research topic whose details need updating or elects to cancel the operation and the system closes the form.
* The system displays the research topic’s details (research topic id, research topic description, and impact).
* The research administrator updates the relevant details (research project description and impact only) and elects to update the research topic.
* The system validates the entries in the fields and, either asks for confirmation to change the research topic’s details, or identifies missing or incorrect fields and prompts for completion of the entry.
* The research administrator either confirms the change of details or elects to cancel the changes and the system closes.
* The system saves the research topic’s details.
* The system displays the “Research topic updated successfully” message.
* The system displays the “Update another research topic?” prompt.
* The research administrator either elects to end the use case and the system closes the form, or elects to update another research topic and the system displays the “Update research topic” form and displays a list of all the research topics.

1. Delete Research Topic

Use Case ID: 29

Description: This use case enables the research administrator to delete a research topic’s details.

Priority: Low

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Delete Research Topic” function.
* The system displays the “Delete Research Topic” form and displays a list of all the research topics (research topic id and research topic description) that are not linked to any research projects.
* The research administrator selects the research topic that requires deleting or the research administrator elects to cancel the operation and the system closes the form.
* The system displays the research topic’s details (research topic id, research topic description, and impact).
* The research administrator either elects to delete the research topic or elects to cancel the operation and the system closes the form.
* The system deletes the research topic.
* The system displays the “Research topic deleted successfully” message.
* The system displays the “Delete another research topic?” prompt.
* The research administrator either elects to end the use case and the system closes the form, or the research administrator elects to delete another research topic and the system displays the delete research topic form and displays a list of all the research topics (research topic id and research topic description) that are not linked to any admissions.

1. Produce Research Report

Use Case ID: 30

Description: This use case enables the research administrator to produce the research report.

Priority: Medium

Pre-conditions: The research administrator has logged onto the system

Post Conditions: None

Assumptions: None

* The research administrator selects the “Produce Research Report function
* The system displays the “Research Report” form
* The research administrator either selects the option to generate the report or elects to cancel, without generating the report, and the system closes the form.
* The system gets the details of the research projects (lecturer last name, lecturer first name, research topic description, impact, research project description, output and start date).
* The system then generates the research report (lecturer last name, lecturer first name, research topic description, impact, research project description, output and start date) with the research projects sorted by research project description
* The system concludes the use case by closing the form.

1. Record Issue

Use Case ID: 31

Description: This use case enables the student support clerk to record an issue a student has with an enrolment.

Priority: Medium

Pre-conditions: The student support clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The student support clerk selects the “Record Issue” function.
* The system displays the “Record Issue” form and displays a list of all the students (student id, last name and first name) who have enrolments.
* The student support clerk either selects the student who has an issue or elects to cancel the operation and the system closes the form.
* The system displays the student’s details (student id, last name, and first name, street address, suburb, and city).
* The system displays the list of the enrolments (enrolment number, year, semester, and course name) for the selected student.
* The student support clerk selects an enrolment.
* The system displays the selected enrolment’s details (enrolment number, year, semester, and course name).
* The student support clerk enters the issue id (1 to 99999 inclusive), issue description, issue date, and action taken.
* The student support clerk elects to record the issue or elects to cancel the operation and the system closes the form.
* The system either confirms that all necessary fields are filled in correctly and saves the issue’s details (issue id, issue description, issue date, action taken, and enrolment number) or the system identifies any missing or incorrect fields and prompts for the completion of all fields.
* The system displays the “Issue recorded successfully” message.
* The system displays the “Record another issue?” prompt.
* The student support clerk either elects to end the use case and the system closes the form, or the student support clerk elects to record another issue and the system displays the “Record Issue” form and displays a list of all the students (student id, last name and first name) who have enrolments.

1. Produce Issues Report

Use Case ID: 32

Description: This use case enables the student support clerk to produce the issues report.

Priority: Medium

Pre-conditions: The student support clerk has logged onto the system

Post Conditions: None

Assumptions: None

* The student support clerk selects the “Produce Issues Report function
* The system displays the “Issues Report” form
* The student support clerk either selects the option to generate the report or elects to cancel, without generating the report, and the system closes the form.
* The system gets the details of all of the issues (issue id, enrolment number, issue description, issue date, action taken, course name and student last name and first name)
* The system then generates the issues report (issue id, enrolment number, issue description, issue date, action taken, course name and student last name and first name) with the issues sorted by issue id within issue date.
* The system concludes the use case by closing the form.