|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 10 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator adding an admission to the system. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Add Admission” function.  Step 2: The system displays the “ Add Admission” form with all fields blank.  Step 3: The assistant administrator enters the admission’s details (description, and admission date).  Step 4: The system displays a list of patients (patient id, last name, and first name).  Step 5: The assistant administrator selects a patient from the list.  Step 6: The system displays a list of wards (ward id, and ward name).  Step 7: The assistant administrator selects a ward from the list.  Step 8: The assistant administrator clicks on the “Add Admission” button.  Step 9: The system validates the details are filled in correctly.  Step 10: The system generates a unique value for admission id.  Step 11: The system saves the admission’s details (admission id, description, ward, patient, admission date, and status: “current”).  Step 12: The assistant administrator clicks on the “Return” button.  Step 13: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 8a1: The assistant administrator clicks on the “Return” button.  Step 8a2: The system goes to step 13. | |
| Step 9a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 9a2: The system returns to step 3. | |
| Step 12a1: The assistant administrator elects to add another admission.  Step 12a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Add Admission

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 11 | **Design Requirements: 🗹** |
| **PRIORITY:** | low |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator updating an admission in the system. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Update Admission” function.  Step 2: The system displays the “Update Admission” form with a list of all the current admissions (admission id, and description).  Step 3: The assistant administrator selects the admission which is to be updated.  Step 4: The system displays the admission’s details (admission id, description, admission date, status, patient last name, patient first name, and ward name).  Step 5: The assistant administrator modifies the relevant details (description, status (either current or complete only), and admission date only).  Step 6: The assistant administrator clicks on the “Update Admission” button.  Step 7: The system validates the details are filled in correctly.  Step 8: The system prompts for confirmation to change the details.  Step 9: The assistant administrator confirms the change of details.  Step 10: The system saves the admission’s details.  Step 11: The assistant administrator clicks on the “Return” button.  Step 12: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 12. | |
| Step 7a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 7a2: The system returns to step 5. | |
| Step 9a1: The assistant administrator clicks on the “Return” button.  Step 9a2: The system goes to step 12. | |
| Step 11a1: The assistant administrator elects to update another admission.  Step 11a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Update Admission

# Delete Admission

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 12 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator deleting an admission from the system. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Delete Admission” function.  Step 2: The system displays the “Delete Admission” form with a list of all the closed admissions (admission id, and description).  Step 3: The assistant administrator selects the admission which is to be deleted.  Step 4: The system displays the admission’s details (admission id, description, admission date, and status).  Step 5: The assistant administrator clicks on the “Delete Admission” button.  Step 6: The system deletes all payments associated with the admission.  Step 7: The system deletes the admission.  Step 8: The assistant administrator clicks on the “Return” button.  Step 9: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 9. | |
| Step 5a1: The assistant administrator clicks on the “Return” button.  Step 5a2: The system goes to step 9. | |
| Step 8a1: The assistant administrator elects to delete another admission.  Step 8a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Produce Admissions Report

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Admissions Report | **USE CASE TYPE** |
| **USE CASE ID:** | 13 | **Design Requirements: þ** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator producing admissions report. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Admissions Report” function.  Step 2: The system displays the “Admissions Report” form.  Step 3: The assistant administrator clicks on the “Produce Report” button to generate the report.  Step 4: The system gets the details (admission ID, description, admission date, and status) of each admission.  Step 5: The system gets the patient’s last name and first name for each admission.  Step 6: The system gets the name of each medication prescribed to each admission.  Step 7: The system then displays the admissions report on the screen.  Step 8: The assistant administrator clicks on the “Return” button.  Step 9: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 9. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Allocate Doctor

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Allocate Doctor | **USE CASE TYPE** |
| **USE CASE ID:** | 20 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator allocating a doctor to an admission. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Allocate Doctor” function.  Step 2: The system displays the “Allocate Doctor” form with a list of all the current admissions (admission id, and description).  Step 3: The assistant administrator selects an admission to allocate the doctor.  Step 4: The system displays the admission’s details (admission id, description, patient last name, and patient first name).  Step 5: The system displays the list of doctors (doctor id, last name, first name, and role) already allocated to the admission.  Step 6: The system displays a list of all the doctors (doctor id, last name, and first name).  Step 7: The assistant administrator selects a doctor from the list of all the doctors.  Step 8: The assistant administrator enters the role (primary or secondary) and fee for the allocation.  Step 9: The assistant administrator clicks on the “Allocate Doctor” button.  Step 10: The system saves the allocation record (doctor, admission, fee, and role).  Step 11: The assistant administrator clicks on the “Return” button.  Step 12: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 12. | |
| Step 9a1: The assistant administrator clicks on the “Return” button.  Step 9a2: The system goes to step 12. | |
| Step 11a1: The assistant administrator elects to allocate another doctor.  Step 11a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Remove Doctor

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Doctor | **USE CASE TYPE** |
| **USE CASE ID:** | 21 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator removing a doctor from an admission. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Remove Doctor” function.  Step 2: The system displays the “Remove Doctor” form with a list of all the current admissions (admission id, and description) with allocations.  Step 3: The assistant administrator selects the admission from which the doctor is to be removed.  Step 4: The system displays the admission’s details (admission id, description, patient last name, and patient first name).  Step 5: The system displays the list of doctors (doctor id, last name, first name, and role) allocated to the admission.  Step 6: The assistant administrator selects a doctor from the list.  Step 7: The assistant administrator clicks on the “Remove Doctor” button.  Step 8: The system deletes the allocation record.  Step 9: The assistant administrator clicks on the “Return” button.  Step 10: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 10. | |
| Step 7a1: The assistant administrator clicks on the “Return” button.  Step 7a2: The system goes to step 10. | |
| Step 9a1: The assistant administrator elects to remove another doctor.  Step 9a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Add Ward

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Ward | **USE CASE TYPE** |
| **USE CASE ID:** | 22 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Facilities Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a facilities administrator adding a ward to the system. | |
| **PRE-CONDITIONS:** | The facilities administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The facilities administrator selects the “Add Ward” function.  Step 2: The system displays the “Add Ward” form with all fields blank.  Step 3: The facilities administrator enters the ward’s details (name, location, and capacity ).  Step 4: The system validates the details are filled in correctly.  Step 5: The system generates a unique value for ward id.  Step 6: The system saves the ward’s details (ward id, name, location, and capacity).  Step 7: The facilities administrator clicks on the “Return” button.  Step 8: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 4a2: The system returns to step 3. | |
| Step 7a1: The facilities administrator elects to add another ward.  Step 7a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Update Ward

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Ward | **USE CASE TYPE** |
| **USE CASE ID:** | 23 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Facilities Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a facilities administrator updating a ward in the system. | |
| **PRE-CONDITIONS:** | The facilities administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The facilities administrator selects the “Update Ward” function.  Step 2: The system displays the “Update Ward” form with a list of all the wards (ward id, and name).  Step 3: The facilities administrator selects the ward which is to be updated.  Step 4: The system displays the ward’s details (ward id, name, location, and capacity).  Step 5: The facilities administrator modifies the relevant details (name, location, and capacity only).  Step 6: The facilities administrator clicks on the “Update Ward” button.  Step 7: The system validates the details are filled in correctly.  Step 8: The system prompts for confirmation to change the details.  Step 9: The facilities administrator confirms the change of details.  Step 10: The system saves the ward’s details.  Step 11: The facilities administrator clicks on the “Return” button.  Step 12: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The facilities administrator clicks on the “Return” button.  Step 3a2: The system goes to step12. | |
| Step 7a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 7a2: The system returns to step 5. | |
| Step 9a1: The assistant administrator clicks on the “Return” button.  Step 9a2: The system goes to step 12. | |
| Step 11a1: The facilities administrator elects to update another ward.  Step 11a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Delete Ward

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Ward | **USE CASE TYPE** |
| **USE CASE ID:** | 24 | **Design Requirements: 🗹** |
| **PRIORITY:** | Low |  |
| **PRIMARY BUSINESS ACTOR:** | Facilities administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a facilities administrator deleting a ward from the system. | |
| **PRE-CONDITIONS:** | The facilities administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The facilities administrator selects the “Delete Ward” function.  Step 2: The system displays the “Delete Ward” form with a list of all the wards (ward id, and name) that are not linked to any admissions.  Step 3: The facilities administrator selects the ward who is to be deleted.  Step 4: The system displays the ward’s details (ward id, name, location, and capacity).  Step 5: The facilities administrator clicks on the “Delete Ward” button.  Step 6: The system deletes the ward.  Step 7: The facilities administrator clicks on the “Return” button.  Step 8: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The facilities administrator clicks on the “Return” button.  Step 3a2: The system goes to step 8. | |
| Step 5a1: The facilities administrator clicks on the “Return” button.  Step 5a2: The system goes to step 8. | |
| Step 7a1: The facilities administrator elects to delete another ward.  Step 7a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Close Admission

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Close Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 25 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes an assistant administrator closing an admission in the system. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Close Admission” function.  Step 2: The system displays the “Close Admission” form with a list of all the billed admissions (admission id, and description) that have payments.  Step 3: The assistant administrator selects an admission from the list.  Step 4: The system displays the admission’s details (admission id, description, and admission date).  Step 5: The assistant administrator clicks on the “Close Admission” button.  Step 6: The system gets the amount, medication name and cost from each prescription for the admission.  Step 7: The system gets the fee for each doctor allocated to the admission.  Step 8: The system calculates the amount due by summing the product of each prescription’s amount and medication’s cost and adding it to the sum of the doctors’ fees.  Step 9: The system calculates the amount paid (the sum of all payments made) on the admission.  Step 10: The system checks that the amount due is equal to or less than the amount paid.  Step 11: The system updates the status of the admission to closed.  Step 12: The system deletes all allocation records related to the admission.  Step 13: The system deletes all prescription records related to the admission.  Step 14: The system displays the message “Admission closed successfully”.  Step 15: The assistant administrator clicks on the “Return” button.  Step 16: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The assistant administrator clicks on the “Return” button.  Step 3a2: The system goes to step 16. | |
| Step 11a1: The system displays the message “Full payment has not been made yet. The admission cannot be closed”.  Step 11a2: The system goes to step 15. | |
| Step 15a1: The assistant administrator elects to close another admission.  Step 15a2: The system returns to step 2. | |
| **POST CONDITIONS:** | The admission’s status is set to closed if the amount due is equal to or less than the amount paid. | |
| **ASSUMPTIONS:** | None | |

# Add Research Project

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 26 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a research administrator adding a research project to a doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The research administrator selects the “Add Research Project” function.  Step 2: The system displays the “Add Research Project” form with a list of all the doctors (doctor id, last name, and first name).  Step 3: The research administrator selects a doctor from the list.  Step 4: The system displays the doctor’s details (doctor id, last name, first name, and specialty).  Step 5: The system displays the research projects’ details (outcome, budget, and research topic description) for each research project already linked to the selected doctor.  Step 6: The system displays a list of all the research topics (research topic id, description, and level).  Step 7: The research administrator selects a research topic from the list.  Step 8: The research administrator enters the research project’s details (outcome, end date, and budget).  Step 9: The research administrator clicks on the “Add Research Project” button.  Step 10: The system saves the research project record (doctor, research topic, outcome, end date, and budget).  Step 11: The research administrator clicks on the “Return” button.  Step 12: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The research administrator clicks on the “Return” button.  Step 3a2: The system goes to step 12. | |
| Step 9a1: The research administrator clicks on the “Return” button.  Step 9a2: The system goes to step 12. | |
| Step 11a1: The research administrator elects to add another research project.  Step 11a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Remove Research Project

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 27 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a research administrator removing a research project from a doctor. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The research administrator selects the “Remove Research Project” function.  Step 2: The system displays the “Remove Research Project” form with a list of all the doctors (doctor id, last name, and first name) who have research projects.  Step 3: The research administrator selects the doctor whose research project is to be removed.  Step 4: The system displays the doctor’s details (doctor id, last name, first name, and specialty).  Step 5: The system displays the research projects’ details (outcome, budget, and research topic description) for each research project linked to the selected doctor.  Step 6: The research administrator selects the research which is to be removed.  Step 7: The research administrator clicks on the “Remove Research Project” button.  Step 8: The system deletes the research project record.  Step 9: The research administrator clicks on the “Return” button.  Step 10: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The research administrator clicks on the “Return” button.  Step 3a2: The system goes to step 10. | |
| Step 7a1: The research administrator clicks on the “Return” button.  Step 7a2: The system goes to step 10. | |
| Step 9a1: The research administrator elects to remove another research project.  Step 9a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Add Research Topic

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Topic | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a research administrator adding a research topic to the system. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The research administrator selects the “Add Research Topic” function.  Step 2: The system displays the “Add Research Topic” form with all fields blank.  Step 3: The research administrator enters the research topic’s details (description, and level).  Step 4: The system validates the details are filled in correctly.  Step 5: The system generates a unique value for research topic id.  Step 6: The system saves the research topic’s details (research topic id, description, and level).  Step 7: The research administrator clicks on the “Return” button.  Step 8: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 4a2: The system returns to step 3. | |
| Step 7a1: The research administrator elects to add another research topic.  Step 7a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Update Research Topic

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Research Topic | **USE CASE TYPE** |
| **USE CASE ID:** | 29 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a research administrator updating a research topic in the system. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The research administrator selects the “Update research topic” function.  Step 2: The system displays the “Update research topic” form with a list of all the research topics (research topic id, and description).  Step 3: The research administrator selects the research topic which is to be updated.  Step 4: The system displays the research topic’s details (research topic id, description, and level).  Step 5: The research administrator modifies the relevant details (description, and level only).  Step 6: The research administrator clicks on the “Update Research Topic” button.  Step 7: The system validates the details are filled in correctly.  Step 8: The system prompts for confirmation to change the details.  Step 9: The research administrator confirms the change of details.  Step 10: The system saves the research topic’s details.  Step 11: The research administrator clicks on the “Return” button.  Step 12: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The research administrator clicks on the “Return” button.  Step 3a2: The system goes to step12. | |
| Step 7a1: The system identifies the details are missing or incorrect and displays the message “Please fill in all fields correctly”.  Step 7a2: The system returns to step 5. | |
| Step 9a1: The research administrator clicks on the “Return” button.  Step 9a2: The system goes to step 12. | |
| Step 11a1: The research administrator elects to update another research topic.  Step 11a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

# Delete Research Topic

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Research Topic | **USE CASE TYPE** |
| **USE CASE ID:** | 30 | **Design Requirements: 🗹** |
| **PRIORITY:** | Medium |  |
| **PRIMARY BUSINESS ACTOR:** | Research administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case describes a research administrator deleting a research topic from the system. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1: The facilities administrator selects the “Delete Research Topic” function.  Step 2: The system displays the “Delete Research Topic” form with a list of all the research topics (research topic id, and description) that are not linked to any research projects .  Step 3: The research administrator selects the research topic who is to be deleted.  Step 4: The system displays the research topic’s details (research topic id, description, and level ).  Step 5: The research administrator clicks on the “Delete Research Topic” button.  Step 6: The system deletes the research topic.  Step 7: The research administrator clicks on the “Return” button.  Step 8: The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a1: The research administrator clicks on the “Return” button.  Step 3a2: The system goes to step 8. | |
| Step 5a1: The research administrator clicks on the “Return” button.  Step 5a2: The system goes to step 8. | |
| Step 7a1: The research administrator elects to delete another research topic.  Step 7a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |