|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Patient | **USE CASE TYPE** |
| **USE CASE ID:** | 1 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case allows the assistant administrator to add a patient’s details. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Add Patient” function.  Step 2: The system displays the “Add Patient” form with all fields blank.  Step 3: The assistant administrator enters the last name, first name, street address, suburb, city, email address, phone number, and insurance code.  Step 4: The assistant administrator clicks on the “Add Patient” button.  Step 5: The system confirms that the details are filled in correctly.  Step 6: The system generates a unique patient ID for the new patient.  Step 7: The system saves the patient’s details (patient id, last name, first name, street address, suburb, city, email address, phone number, and insurance code).  Step 8: The system displays the “Patient added successfully” message.  Step 9: The system displays the “Add another patient?” prompt.  Step 10: The assistant administrator clicks on the “Return” button.  Step 11: The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a1: The assistant administrator clicks on the “Return” button.  Step 4a2: The system goes to step 11. | |
| Step 5a1: The system identifies that the details are missing or incorrect, displays the “Please fill in the fields correctly” message.  Step 5a2: The system goes to step 3. | |
| Step 10a1: The assistant administrator chooses to add another patient.  Step 10a2: The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE NAME:** | | | Update Patient | **USE CASE TYPE** | |
| **USE CASE ID:** | | | 2 | **Design Requirements: 🗹** | |
| **PRIORITY:** | | | High |  | |
| **PRIMARY BUSINESS ACTOR:** | | | Assistant Administrator | | |
| **OTHER PARTICIPATING ACTORS:** | | | None | | |
| **DESCRIPTION:** | | | This use case allows the assistant administrator to update a selected patient’s details. | | |
| **PRE-CONDITIONS:** | | | The assistant administrator has logged onto the system | | |
| **TYPICAL COURSE** | | | Step 1: The assistant administrator selects the “Update Patient” function.  Step 2: The system displays the “Update Patient” form.  Step 3: The system gets the list of all patients.  Step 4: The system displays the list of all the patients (patient ID, last name and first name) in a list box.  Step 5: The assistant administrator selects the patient that has details that needs updating.  Step 6: The system gets the selected patient’s details.  Step 7: The system displays the selected patient’s details (patient id, last name, first name, street address, suburb, city, phone number, email address, and insurance code).  Step 8: The assistant administrator updates the relevant patient’s details (last name, first name, street address, suburb, city, phone number, email address, and insurance code only).  Step 9: The assistant administrator clicks on the “Update Patient” button.  Step 10: The system validates the entries in the fields and prompts for confirmation to change the patients details.  Step 11: The assistant administrator clicks on the “Confirm” button.  Step 12: The system saves the patient’s details.  Step 13: The system displays the “Patient updated successfully” message.  Step 14: The system displays the “Update another patient?” prompt.  Step 15: The assistant administrator clicks on the “Return” button.  Step 16: The system closes the form to end the use case. | | |
| **OF EVENTS:** | | |
|  | | |
|  | | |
|  | | |
|  | | |
| **ALTERNATE COURSES:** | | | Step 4a1: The assistant administrator clicks on the “Return” button.  Step 4a2: The system goes to step 16. | | |
| Step 7a1: The system identifies missing or incorrect fields, displays the “Please fill in the fields correctly” message.  Step 7a2: The system returns to step 8. | | |
| Step 9a1: The assistant administrator clicks on the “Return” button.  Step 9a2: The system goes to step 16. | | |
| Step 15a1: The assistant administrator chooses to update another patient.  Step 15a2: The system returns to step 3. | | |
| **POST CONDITIONS:** | | | None | | |
| **ASSUMPTIONS:** | | | None | | |
| **USE CASE NAME:** | Delete Patient | | **USE CASE TYPE** |
| **USE CASE ID:** | 3 | | **Design Requirements: 🗹** |
| **PRIORITY:** | High | |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | | |
| **OTHER PARTICIPATING ACTORS:** | None | | |
| **DESCRIPTION:** | This use case allows the assistant administrator to delete a selected patient’s details. | | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Delete Patient” function.  Step 2: The system displays the “Delete Patient” form.  Step 3: The system gets the list of patients who have no admissions.  Step 4: The system displays the list of all the patients (patient ID, last name and first name) who have no admissions.  Step 5: The assistant administrator selects the patient that requires deleting.  Step 6: The system gets the selected patient’s details.  Step 7: The system displays the patient’s details (patient ID, last name, first name, street address, suburb, city, phone number) in read-only textboxes.  Step 8: The assistant administrator clicks on the “Delete” button.  Step 9: The system deletes the patient.  Step 10: The system displays the “Patient deleted successfully” message.  Step 11: The system displays the “Delete another patient?” prompt.  Step 12: The assistant administrator clicks on the “Return” button.  Step 13: The system closes the form to end the use case. | | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a1: The assistant administrator clicks on the “Return” button.  Step 5a2: The system goes to step 13. | | |
| Step 8a1: The assistant administrator clicks on the “Return” button.  Step 8a2: The system goes to step 13. | | |
|  | Step 12a1: The assistant administrator chooses to delete another patient.  Step 12a2: The system goes to step 3. | | |
| **POST CONDITIONS:** | None | | |
| **ASSUMPTIONS:** | None | | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Patients Report | **USE CASE TYPE** |
| **USE CASE ID:** | 4 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case allows the assistant administrator to produce the patients report. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Produce Patients Report” function.  Step 2: The system displays the “Produce Patients Report” form.  Step 3: They assistant administrator clicks on the “Display Report” button.  Step 4: The system gets the details (patient ID, last name, first name, street address, suburb, city, phone number, email address, and insurance code) of each patient.  Step 5: The system counts the complete admissions for each patient.  Step 6: The system counts the current admissions for each patient.  Step 7: The system displays the patient’s report (patient ID, last name, first name, street address, suburb, city, phone number, email address, insurance code, number of complete admissions and number of current admissions for each patient) sorted by first name within last name.  Step 8: The assistant administrator clicks on the “Return” button.  Step 9: The system closes the form to end the use case. | |
| **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 | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Record Payment | **USE CASE TYPE** |
| **USE CASE ID:** | 5 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case allows the assistant administrator to record a payment made by a patient against a selected admission. | |
| **PRE-CONDITIONS:** | The assistant administrator has logged onto the system | |
| **TYPICAL COURSE** | Step 1: The assistant administrator selects the “Record Payment” function.  Step 2: The system displays the “Record Payment” form.  Step 3: The system gets the list of all patients that have billed admissions.  Step 4: The system displays the list of all the patients (patient ID, last name and first name) that have billed admissions in a list box.  Step 5: The assistant administrator selects the patient who is making the payment.  Step 6: The system gets the details of the selected patient.  Step 7: The system displays the selected patient’s details (patient ID, last name and first name) in a textbox.  Step 8: The system displays a list of the patient’s billed admissions (admission ID and description) in a combo box.  Step 9: The assistant administrator selects an admission.  Step 10: The assistant administrator enters the payment amount.  Step 11: The assistant administrator clicks the “Record Payment” button.  Step 12: The system checks that the field is completed correctly.  Step 13: The system generates a unique value for the payment code.  Step 14: The system saves the payment (payment code, admission, amount, and today’s date).  Step 15: The system displays the “Payment recorded successfully” message.  Step 16: The assistant administrator clicks on the “Return” button.  Step 17: The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a1: The assistant administrator clicks on the “Return” button.  Step 5a2: The system goes to step 17. | |
| Step 11a1: The assistant administrator clicks on the “Return” button.  Step 11a2: The system goes to step 17. | |
| Step 12a1: The system identifies that the field input is missing or incorrect, displays the “Please fill in the field correctly” message.  Step 12a2: The system goes to step 10. | |
|  | Step 16a1: The assistant administrator chooses to record another payment.  Step 16a2: The system returns to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Topic | **USE CASE TYPE** |
| **USE CASE ID:** | 29 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case allows the research administrator to add a research topic’s details. | |
| **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 description and level.  Step 4: The system confirms that the details are filled in correctly.  Step 5: The research administrator clicks the “Add Research Topic” button.  Step 6: The system generates a unique value for the research topic ID.  Step 7: The system saves the research topic’s details (research topic ID, description, and level).  Step 8: The system displays the “Research topic added successfully” message.  Step 9: The system displays the “Add another research topic?” prompt.  Step 10: The research administrator clicks on the “Return” button.  Step 11: The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a1: The system identifies that the details are missing or incorrect, displays the “Please fill in the fields correctly” message.  Step 4a2: The system goes to step 3. | |
| Step 5a1: The assistant administrator clicks on the “Return” button.  Step 5a2: The system goes to step 11. | |
| Step 10a1: The assistant administrator chooses to add another research topic.  Step 10a2: The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **USE CASE NAME:** | Update Research Topic | | **USE CASE TYPE** | | |
| **USE CASE ID:** | 30 | | **Design Requirements: 🗹** | | |
| **PRIORITY:** | High | |  | | |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | | |
| **DESCRIPTION:** | This use case allows the research administrator to update a selected research topic’s details. | | | | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system | | | | |
| **TYPICAL COURSE**  **OF EVENTS:** | Step 1: The research administrator selects the “Update Research Topic” function.  Step 2: The system displays the “Update Research Topic” form.  Step 3: The system gets a list of all the research topics.  Step 4: The system displays a list of all the research topics (research topic ID and description) in a list box.  Step 5: The research administrator selects the research topic whose details need updating.  Step 6: The system gets the selected research topic details.  Step 7: The system displays the research topic’s details (research topic ID, description and level).  Step 8: The research administrator updates the relevant details (description and level only).  Step 9: The research administrator clicks on the “Update Research Topic” button.  Step 10: The system validates the entries in the fields and asks for confirmation to change the research topic’s details.  Step 11: The research administrator clicks on the “Confirm” button.  Step 12: The system saves the research topic’s details.  Step 13: The system displays the “Research topic updated successfully” message.  Step 14: The system displays the “Update another research topic?” prompt.  Step 15: The research administrator clicks on the “Return” button.  Step 16: The system closes the form to end the use case. | | | | |
| **ALTERNATE COURSES:** | Step 5a1: The research administrator clicks on the “Return” button.  Step 5a2: The system goes to step 16. | | | | |
| Step 10a1: The system identifies that the details are missing or incorrect, displays the “Please fill in the fields correctly” message.  Step 10a2: The system returns to step 8. | | | | |
| Step 11a1: The research administrator clicks on the “Return” button.  Step 11a2: The system goes to step 16. | | | | |
| Step 15a1: The research administrator chooses to update another research topic.  Step 15a2: The system returns to step 3. | | | | |
| **POST CONDITIONS:** | None | | | | |
| **ASSUMPTIONS:** | None | | | | |
| **USE CASE NAME:** | | Delete Research Topic | | **USE CASE TYPE** |
| **USE CASE ID:** | | 31 | | **Design Requirements: 🗹** |
| **PRIORITY:** | | High | |  |
| **PRIMARY BUSINESS ACTOR:** | | Research Administrator | | |
| **OTHER PARTICIPATING ACTORS:** | | None | | |
| **DESCRIPTION:** | | This use case allows the research administrator to delete a selected research topic’s details. | | |
| **PRE-CONDITIONS:** | | The research administrator has logged onto the system | | |
| **TYPICAL COURSE** | | Step 1: The research administrator selects the “Delete Research Topic” function.  Step 2: The system displays the “Delete Research Topic” form.  Step 3: The system gets a list of all the research topics not linked to any research projects.  Step 4: The system displays a list of all the research topics (research topic ID and description) that are not linked to any research projects.  Step 5: The research administrator selects the research topic that requires deleting.  Step 6: The system gets the selected research topic details.  Step 7: The system displays the research topic’s details (research topic ID, description and level).  Step 8: The research administrator clicks on the “Delete” Button.  Step 9: The system deletes the research topic’s details.  Step 10: The system displays the “Research topic deleted successfully” message.  Step 11: The system displays the “Delete another research topic?” message.  Step 12: The research administrator clicks on the “Return” button.  Step 13: The system closes the form to end the use case. | | |
| **OF EVENTS:** | |
|  | |
|  | |
|  | |
|  | |
| **ALTERNATE COURSES:** | | Step 5a1: The research administrator clicks on the “Return” button.  Step 5a2: The system goes to step 13. | | |
| Step 8a1: The research administrator clicks on the “Return” button.  Step 8a2: The system goes to step 13. | | |
| Step 12a1: The assistant administrator chooses to delete another research topic.  Step 12a2: The system returns to step 2. | | |
| **POST CONDITIONS:** | | None | | |
| **ASSUMPTIONS:** | | None | | |