|  |  |  |
| --- | --- | --- |
| **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 enables the assistant administrator to add a selected patient’s admission details. | |
| **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 (admission description and admission date).  Step 4- The system checks that the details are filled in correctly.  Step 5- The system displays a list of patients (patient id, last name, and first name) in a combo box.  Step 6- The assistant administrator selects a patient.  Step 7- The system displays a list of wards (ward id and ward name) in a combo box.  Step 8- The assistant administrator selects a ward.  Step 9- The assistant administrator clicks on the “add admission” button.  Step 10- The system checks that the details are filled in correctly.  Step 11- The system saves the admission’s details.  Step 12- The system displays the message “Admission added successfully”.  Step 13- The system displays the “Add another admission?” prompt.  Step 14 - The assistant administrator clicks on the “Return” button.  Step 15- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5A1- The system identifies that some details are missing or incorrect, displays the “Please fill in all fields correctly” message.  Step 5A2- The system returns to step 3. | |
| Step 9A1- The assistant administrator clicks on the “Return button.  Step 9A2- The system closes the form. | |
|  | Step 14A1- The assistant administrator elects to add another admission.  Step 14A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 11 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to update a selected admission’s details. | |
| **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) in a combo box.  Step 3- The assistant administrator selects the admission that has details that needs updating.  Step 4- The system displays the admission’s details (admission id, description, admission date, status, patient last name, patient first name, and ward name) in text boxes.  Step 5- The assistant administrator updates the relevant details (description, status (either current or complete only), and admission date only).  Step 6- The assistant administrator clicks the “Update Admission” button.  Step 7- The system validates the entries in the fields.  Step 8- The system displays the message “Are you sure that you want to update this admission?”.  Step 9- The assistant administrator clicks “Yes”.  Step 10- The system saves the admission’s details.  Step 11- The system displays the message “Admission updated successfully”.  Step 12- The system displays the “Update another admission?” prompt.  Step 13- The assistant administrator clicks the “Return” button.  Step 14- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 8A1- The system identifies that some details are missing or incorrect, displays the “Please fill in all fields correctly” message.  Step 8A2- The system returns to step 5. | |
|  | Step 9A1- The assistant administrator clicks “No”.  Step 9A2- The system closes the form. | |
|  | Step 13A1- The assistant administrator elects to update another admission.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Admission | **USE CASE TYPE** |
| **USE CASE ID:** | 12 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Assistant Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to delete a selected admission’s details. | |
| **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) in a combo box.  Step 3- The assistant administrator selects the admission that requires deleting.  Step 4- The system displays the admission’s details (admission id, description, admission date, and status) in text boxes.  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 system displays the message “Admission deleted successfully”.  Step 9- The system displays the “Delete another admission?” prompt.  Step 10- The assistant administrator clicks on the “Return” button.  Step 11- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The assistant administrator clicks on the “Return” button.  Step 3A2- the system closes the form. | |
| Step 5A1- The assistant administrator clicks on the “Return” button.  Step 5A2- the system closes the form. | |
|  | Step 10A1- The assistant administrator elects to delete another admission.  Step 10A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Admissions Report | **USE CASE TYPE** |
| **USE CASE ID:** | 13 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | The assistant administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the assistant administrator to produce the 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 “Generate report” button.  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 (admission ID, description, admission date, status, patient last name, patient first name, and medication names for each admission).  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 closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Prescription | **USE CASE TYPE** |
| **USE CASE ID:** | 15 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Pharmacy Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the pharmacy administrator to remove a prescription from a selected admission. | |
| **PRE-CONDITIONS:** | The pharmacy administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The pharmacy administrator selects the “Remove Prescription” function.  Step 2- The system displays the “Remove Prescription” form with a list of all the current admissions (admission id and description) that have prescriptions in a combo box.  Step 3- The pharmacy administrator selects the admission to remove the prescription from.  Step 4- The system displays the admission’s details (admission id, description, and patient’s last name and first name) in text boxes.  Step 5- The system displays a list of the prescriptions (prescriptionID, medication Name, prescription date, and amount) prescribed to the admission in a list box.  Step 6- The pharmacy administrator selects a prescription.  Step 7- The pharmacy administrator clicks on the “Remove prescription” button.  Step 8- The system deletes the prescription details.  Step 9- The system displays the message “Prescription removed successfully”.  Step 10- The system displays the “Remove another prescription?” prompt.  Step 11- The pharmacy administrator clicks on the “Return” button.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The pharmacy administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 7A1- The pharmacy administrator clicks on the “Return” button.  Step 7A2- The system closes the form. | |
|  | Step 11A1- The pharmacy administrator elects to remove another prescription  Step 11A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None. | |
| **ASSUMPTIONS:** | None. | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 27 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None. | |
| **DESCRIPTION:** | This use case enables the research administrator to add research to a selected 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) in a combo box.  Step 3- The research administrator selects the doctor to add the research project to.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty) in text boxes.  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 in data grid view.  Step 6- The system displays a list of the research topic (research topic id, description, and level) in a combo box.  Step 7- The research administrator selects a research topic.  Step 8- The research administrator enters the research project’s outcome, end date and budget.  Step 9- The research administrator clicks on the “Add Research” button.  Step 10- The system saves the research project details (research project id, doctor, research topic, outcome, end date, and budget).  Step 11- The system displays the “Research project added successfully” message.  Step 12- The system displays the “Add another research project?” prompt.  Step 13- The research administrator clicks on the “Return” button.  Step 14- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 9A1- The research administrator clicks on the “Return” button.  Step 9A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to add another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Remove Research Project | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to remove research from selected 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 in a combo box.  Step 3- The research administrator selects the doctor to remove the research project from.  Step 4- The system displays the doctor’s details (doctor id, last name, first name, and specialty) in a text box.  Step 5- The system displays the research projects’ details (outcome, budget, and research topic description) for each research project linked to the selected doctor in a list box.  Step 6- The research administrator selects the research project to remove.  Step 7- The research administrator clicks on the “remove research” button.  Step 8- The system deletes the research project’s details.  Step 9- The system displays the message “Research project removed successfully”.  Step 10- The system displays the “Remove another research project?” prompt.  Step 11- The research administrator clicks on the “Return” button.  Step 12- The system ends the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| Step 7A1- The research administrator clicks on the “Return” button.  Step 7A2- The system closes the form. | |
|  | Step 13A1- The research administrator elects to remove another research project.  Step 13A2- The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Research Projects Report | **USE CASE TYPE** |
| **USE CASE ID:** | 32 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Research Administrator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the research administrator to produce the research projects report. | |
| **PRE-CONDITIONS:** | The research administrator has logged onto the system. | |
| **TYPICAL COURSE** | Step 1- The research administrator selects the “Produce Research Projects Report” function.  Step 2- The system displays the “Research Projects Report” form.  Step 3- The research administrator clicks the “Generate report” button.  Step 4- The system gets the details (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description) of each research project.  Step 5- The system then displays the research project report (research project id, outcome, budget, end date, doctor’s id, last name, first name, and research topic description of each research project.) sorted by research project id.  Step 6- The research administrator clicks on the “Return” button.  Step 7- The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3A1- The research administrator clicks on the “Return” button.  Step 3A2- The system closes the form. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |