1. Problem statement

We need a software to manage a hospital’s information through user interface connected to a database and controllers that manage different system requests base on the selected request.

1. Use cases

* Add patient
* Checkout patient
* Search patient
* Modify patient
* Schedule appointment
* Reserve bed for patient
* Add doctor
* Remove doctor
* Add department
* Delete department

1. Use case diagram
2. Use case description

|  |  |
| --- | --- |
| Use Case Name | Add Patient |
| Goal in Context | Adding a new patient to the hospital |
| Pre-Conditions | Administrator is logged in |
| Successful End Condition | Patient added |
| Failure End Condition | Patient wasn’t added |
| Primary Actor | Administrator |
| Secondary Actor | None |
| Trigger | “Add patient” button clicked |
| Input | Patient information |
| Output | None |
| Main Flow | 1. Administrator selects “Add patient” button  2. System shows “Add patient” form  3.Administrator adds patient information into the form  4.Administrator clicks “Add” button  5 Systems confirms if patient information is valid  6. Patient added to hospital system  7.Success message appears |
| Extensions | 5.1 Systems rejects invalid patient information  5.2 Show error message “Invalid patient information |

|  |  |
| --- | --- |
| Use Case Name | Checkout patient |
| Goal in Context | Checking out patient from hospital |
| Pre-Conditions | Administrator is logged in and patients exists and is currently an active patient |
| Successful End Condition | Patient checked out |
| Failure End Condition | Patient wasn’t able to checkout |
| Primary Actor | Administrator |
| Secondary Actor | None |
| Trigger | “Checkout patient” button clicked |
| Input | Patient information, services offered to patient and medical costs. |
| Output | Medical bill |
| Main Flow | 1. Administrator selects “Checkout patient” button  2. System shows “Checkout patient” form  3. Administrator adds patient information and medical costs into the form  4.Administrator clicks “Checkout” button  5. Systems confirms if patient exists  6. Systems confirms that patient is currently an active patient  7. Patient is marked as an inactive patient in the hospital system  9.Medical bill printed  8.Success message appears |
| Extensions | 5.1. No patient found in system  5.2 Failed message appears showing “No patient found”  6.1 Patient is found to be an inactive patient.  6.2 Failed message appears showing “Patient is currently inactive and has no bills”  8.1 Printer not working  8.2 Show error message “Problem with printer” |

|  |  |
| --- | --- |
| Use Case Name | Schedule appointment |
| Goal in Context | Schedule an appointment between a patient and a doctor |
| Pre-Conditions | Administrator is logged in, patients exists, doctor exists |
| Successful End Condition | Appointment reserved |
| Failure End Condition | Appointment not reserved |
| Primary Actor | Administrator |
| Secondary Actor | None |
| Trigger | “Schedule appointment” button clicked |
| Input | Patient information and doctor’s information and an appointment note |
| Output | None |
| Main Flow | 1. Administrator selects “Schedule appointment” button  2. System shows “Schedule appointment” form  3. Administrator adds patient information and doctor’s information into the form  4.Administrator clicks “Schedule” button  5. Systems confirms if patient exists  6. Systems confirms if doctor exists  7. System reserves appointment  8. Success message appears showing appointment date |
| Extensions | 5.1. No patient found in system  5.2 Failed message appears showing “No patient found”  6.1. No doctor found in system  6.2 Failed message appears showing “No doctor found” |

|  |  |
| --- | --- |
| Use Case Name | Add doctor |
| Goal in Context | Adding a new doctor to the hospital |
| Pre-Conditions | Administrator is logged in and there is a department for the doctor |
| Successful End Condition | Doctor added |
| Failure End Condition | Doctor wasn’t added |
| Primary Actor | Administrator |
| Secondary Actor | None |
| Trigger | “Add doctor” button clicked |
| Input | Doctor’s information and department’s name |
| Output | None |
| Main Flow | 1. Administrator selects “Add doctor” button  2. System shows “Add doctor” form  3.Administrator adds doctor’s information and department’s information into the form  4.Administrator clicks “Add” button  5. Systems confirms if department exists  6. Doctor added to hospital system  7. Success message appears |
| Extensions | 5.1. No department found for doctor  5.2 Failed message appears showing “Department doesn’t exist” |