

Hospital Visiting System

User Requirements

The system should allow employees to book visits for specific patients by selecting the patient, procedure and specific date for the visit. Optionally a ward and a discount might be applied to the visit, based on the discount the total is calculated (if the discount isn't provided the total is the original procedure price). A staff user can also view, create and edit medical records for specific users. The system should also allow to calculate an employee's yearly income, which is specific to each employee.

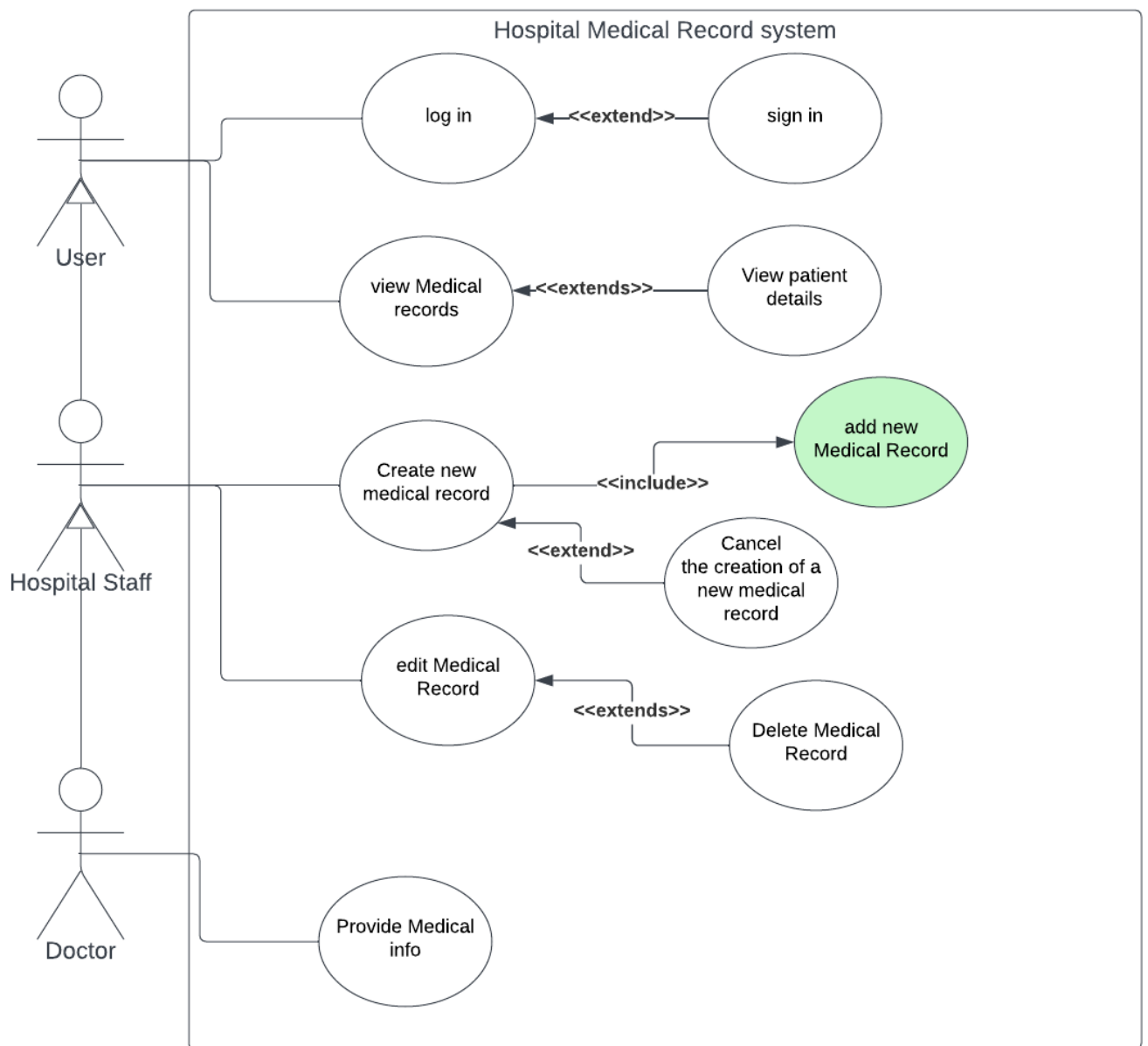
Doctors can perform procedures and remove them from the list of performed procedures, view and edit procedure details, join a department, become a head of a department

Administrators should be able to perform either IT or Management tasks as well as remove them from the task list.

Patients should be able to book and cancel visits and view their own Medical Records.

Doctors can give the medical staff the information necessary to fill into the medical record for as well as enter the information themselves.

Figure 1. Use Case Diag



Use case Scenario: Add medical Record

Precondition: The hospital staff worker is authorized and logged into the system. The "Patients" window is accessible, and the patient for whom the medical record will be added exists in the system.

Points:

The hospital staff worker navigates to the "Patients" window.

The staff worker selects the patient for whom a new medical record will be added by clicking on the "view" button in the "Medical record" column of the patient.

The system redirects the staff worker to the "Medical Records" window specific to the selected patient.

The staff worker clicks on the "add record" button.

The system presents the "Add New Medical Record" form.

The staff worker selects the patient from the list of patients in the form.

The staff worker enters the weight, height, and blood sugar levels of the patient.

The staff worker enters the date of the record.

The staff worker has the option to cancel the process by clicking the "cancel" button.

If the staff worker decides to proceed, they click the "add" button to submit the data and create a new medical record.

The system validates the entered data and creates the new medical record.

A pop-up message appears, confirming that the record was submitted successfully.

The system navigates the staff worker back to the "Medical Records" window.

The staff worker can view the newly added record along with the previously existing records.

Postcondition: The medical record for the selected patient is successfully added to the system and is visible in the "Medical Records" window.

Figure 2. Class Diagram – analytical

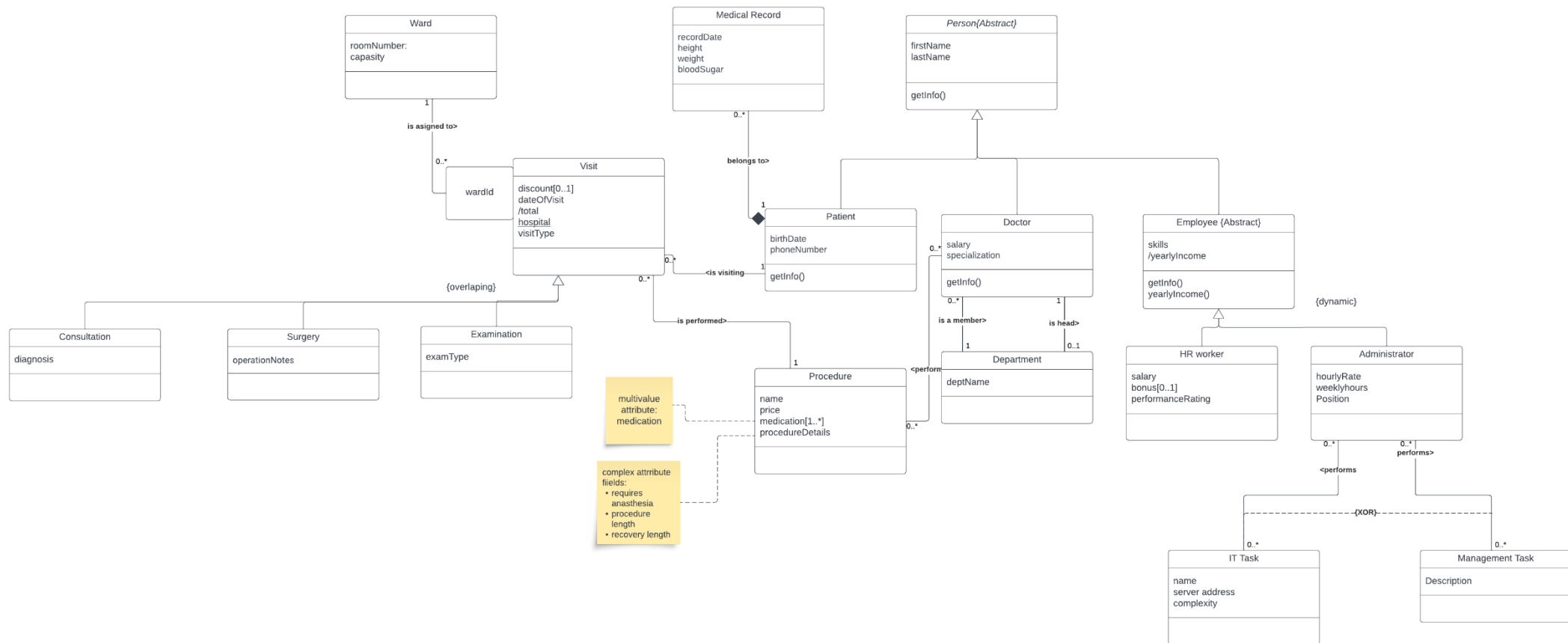


Figure 3. Class Diagram – design

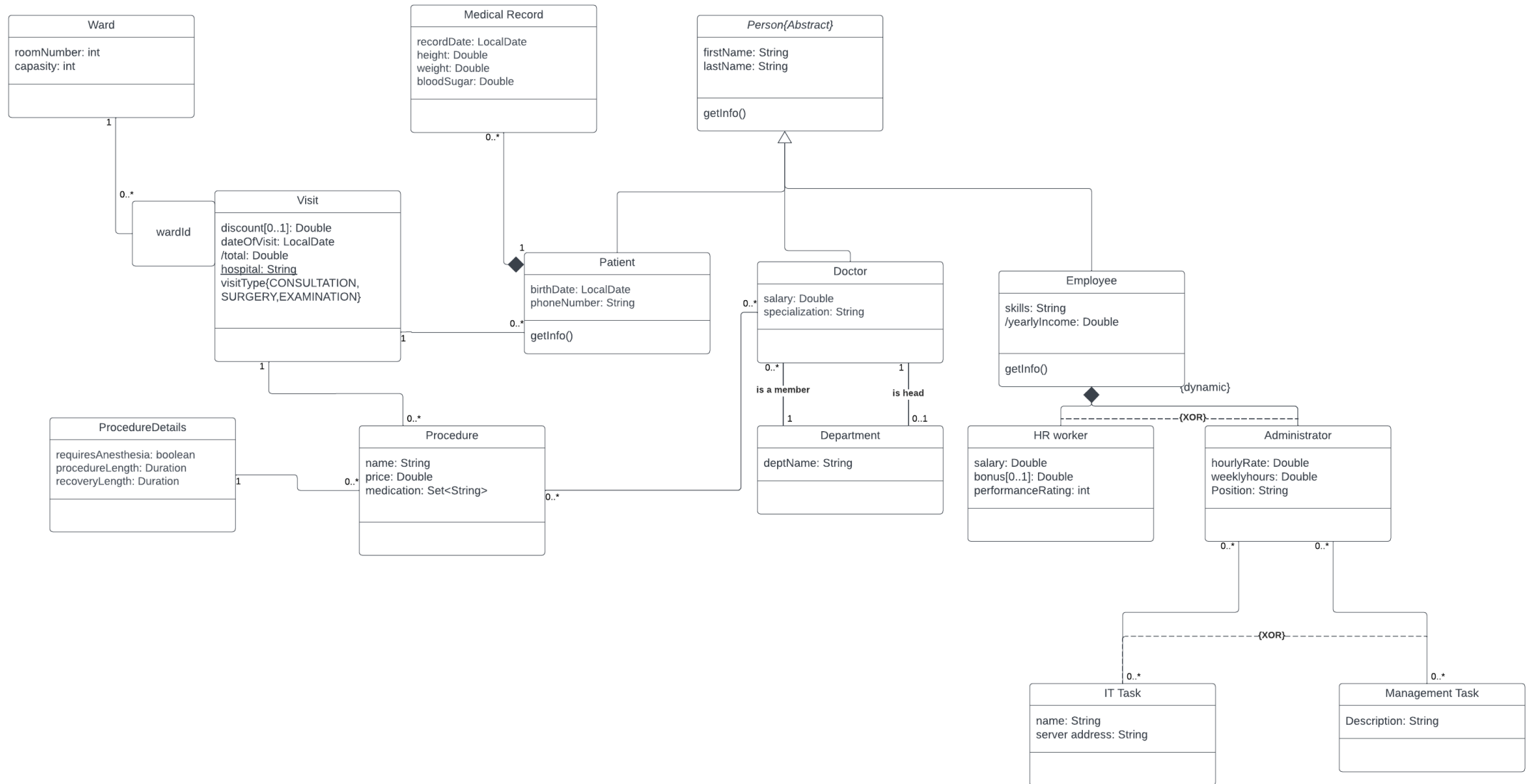


Figure 4. Activity Diagram for Medical Record

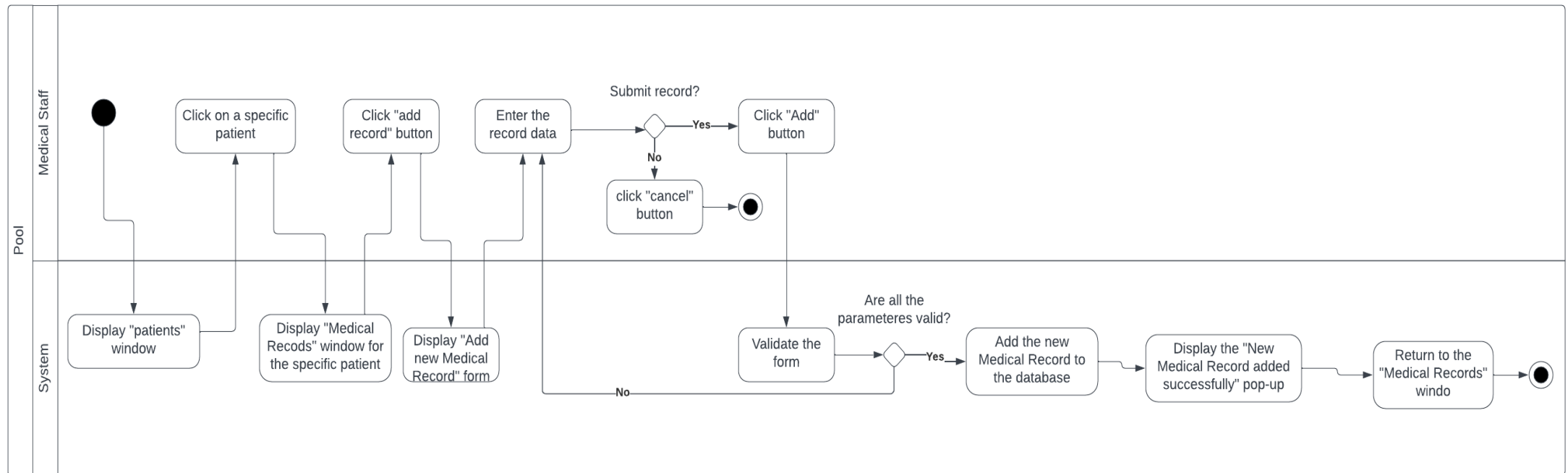


Figure 5. State Diagram

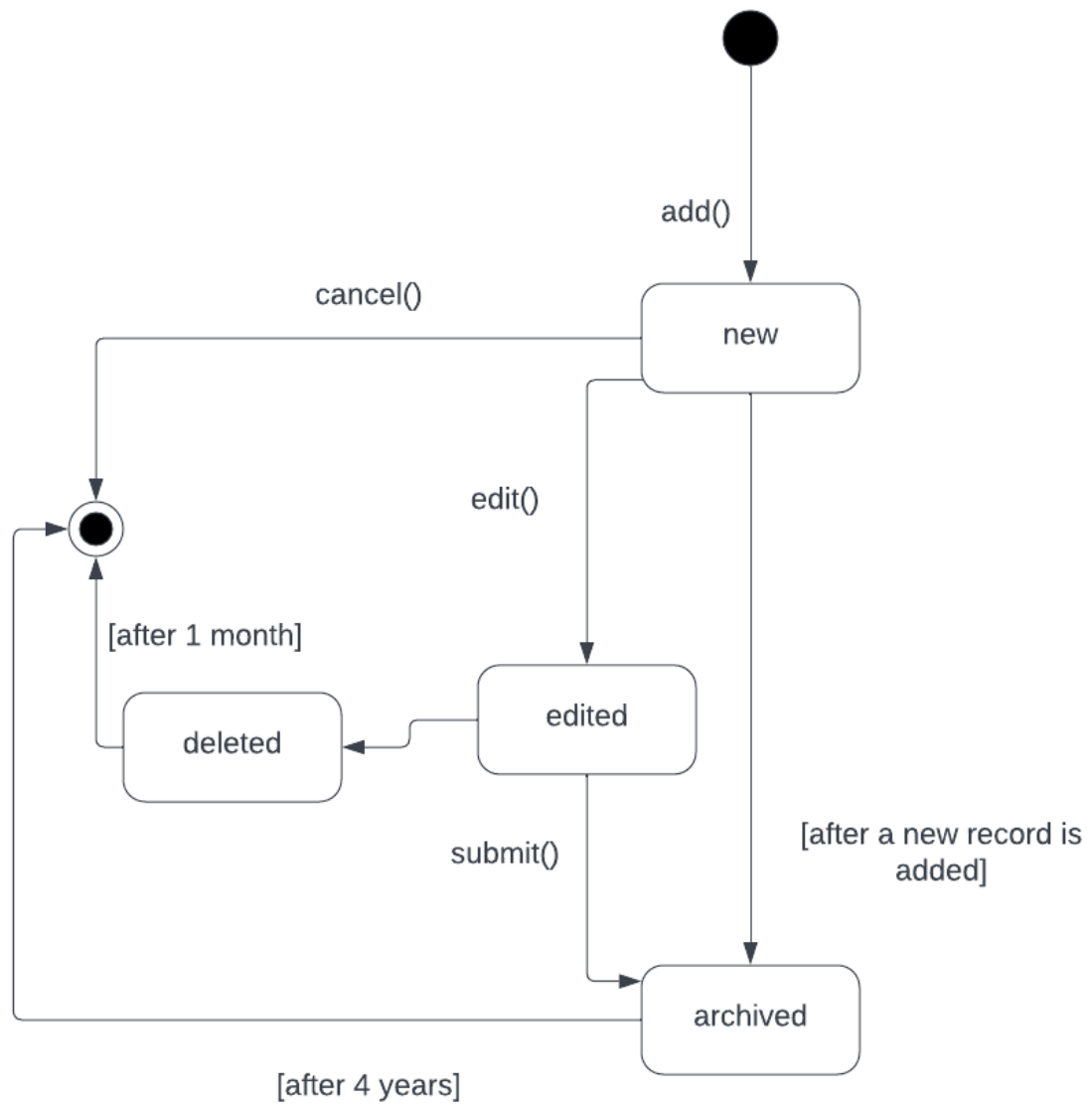


Figure 6. The interaction (sequence) diagram

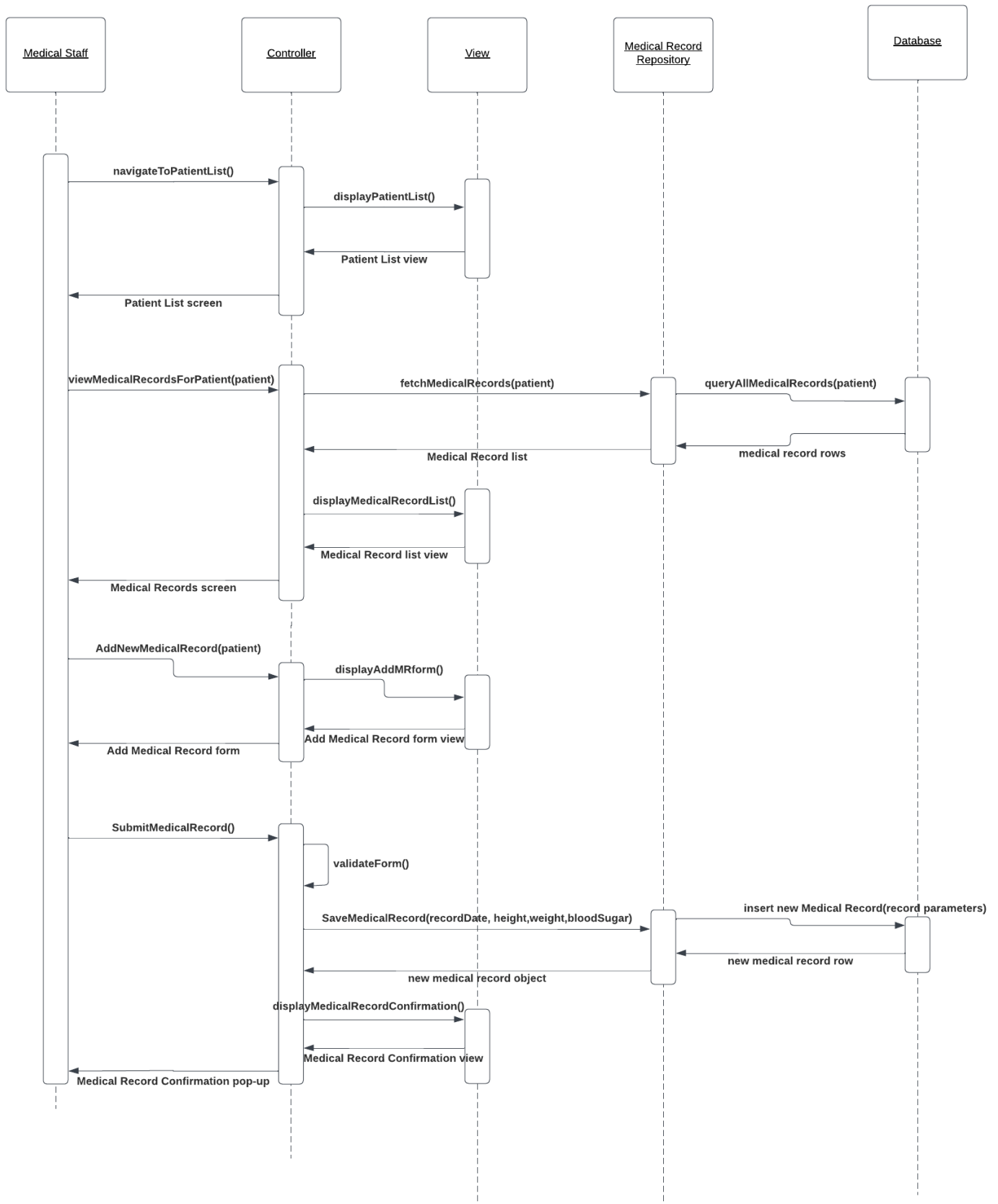


Figure 7. GUI design for “Add Medical Record” use case flow

“Patients” list screen with “view” Medical record button:

Patients				
Patient List				
First name	Last name	Birth date	Phone number	Medical Record
Jay	Keigh	01.10.2000	233090321	View
Elle	E mayo	07.03.1996	723095224	View
Axe	Dee	23.12.1963	950066632	View

Figure 8. GUI design for “Add Medical Record” use case flow

List of “Medical Records of Patient” screen with “add new” button

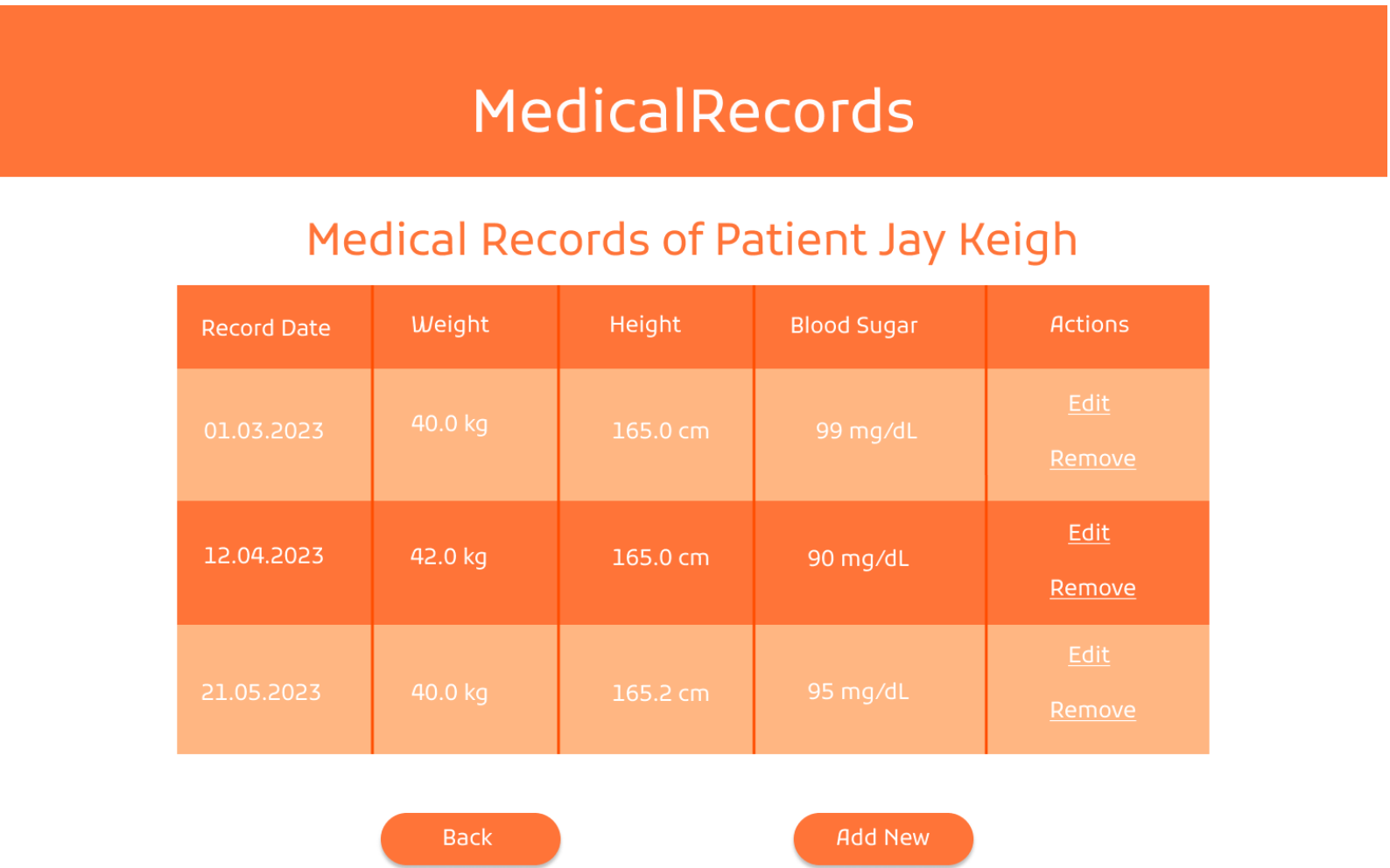



Figure 9. GUI design for “Add Medical Record” use case flow


“Add Medical Record form for specific Patient” screen with “add” and “cancel” buttons

New Medical Record

Add new Medical Record for Patient Jay Keigh

Record date
 

Patient

Jay Keigh 

Weight

Height

Blood Sugar

Cancel

Add

Figure 10. GUI design for “Add Medical Record” use case flow

“Add Medical Record form for any Patient” screen with “Select Patient” dropdown box, which allows to choose a patient

New Medical Record

Record date

mm/dd/yyyy



Patient

Select Patient



Jay Keigh

Elle Emayo

Axe Dee

Height

Enter height (cm)

Blood Sugar

Enter blood sugar levels (mg/dL)

Cancel

Add

Discussion of design decisions

The Class Diagram is specifically designed for implementing the Hibernate ORM framework, specifically using POJOs as models for ORM entities. The main focus of the diagram is to illustrate the structure of data types and associations, without including any logic utilization classes such as Services or Controllers.

To keep the diagram less cluttered, certain elements are intentionally omitted. For example, the "id" field, which is required for all classes in Hibernate entities, is not shown in the diagram as it would be the same for all classes. Additionally, the private fields annotations are not included, assuming that all fields in all classes would be private and accessed through public getters and setters.

Overall, the Class Diagram focuses on the structural representation of entities and associations, while leaving out specific implementation details related to ORM framework utilization and logic implementation.

In class Procedure complex attribute "procedure details" is made to be stored in a separate entity for more effective querying and the added benefit of reusing the same details for other procedures.

Visit class attribute "discount" is optional and when it is applied, the derived attribute total is calculated by subtracting the price of the procedure multiplied by the discount from the price of the procedure. If the discount is not set, the total just returns the "price" attribute from the Procedure class

HR worker attribute "bonus" is also optional and can only be added if the performance rating is 4 or more, this attribute is used when calculating the derived attribute "yearlyIncome"