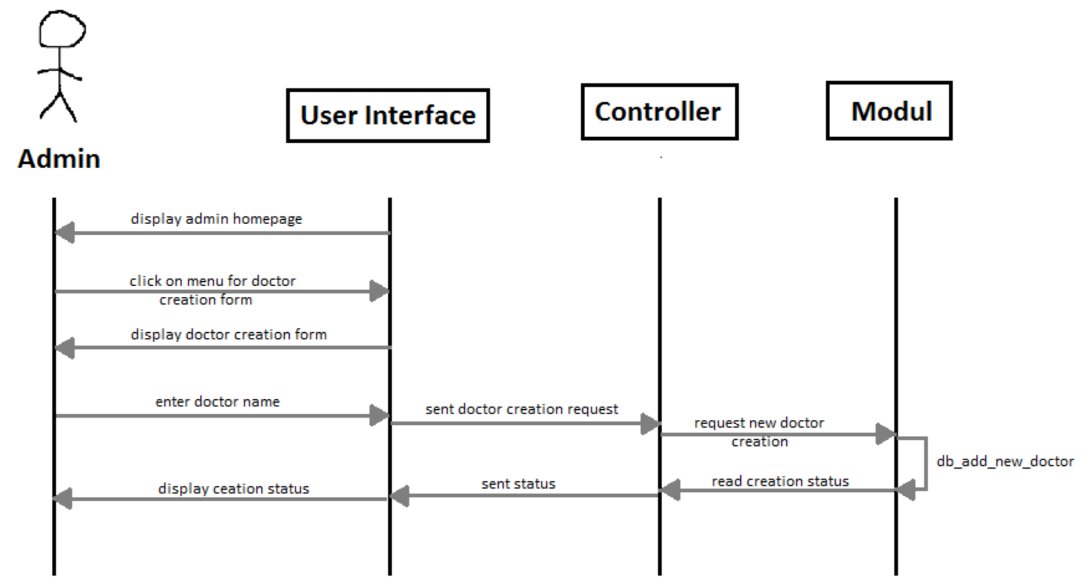
Information Systems in Healthcare – Michel Kana, PhD - Homework

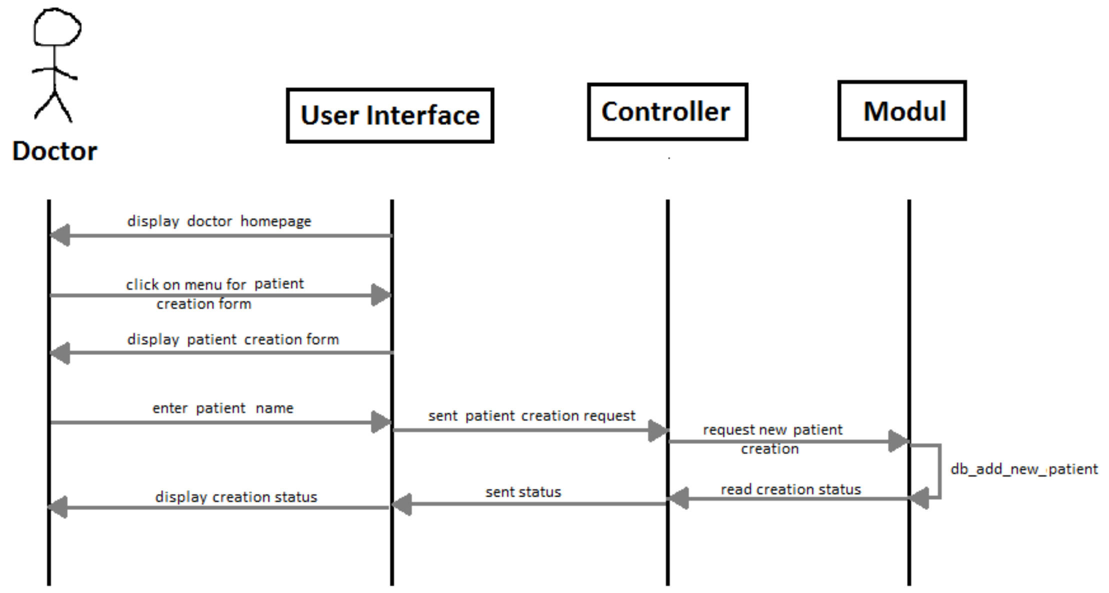
Solution by Katarzyna Dunikowska

1. Draw the sequence diagrams for the remaining use cases.

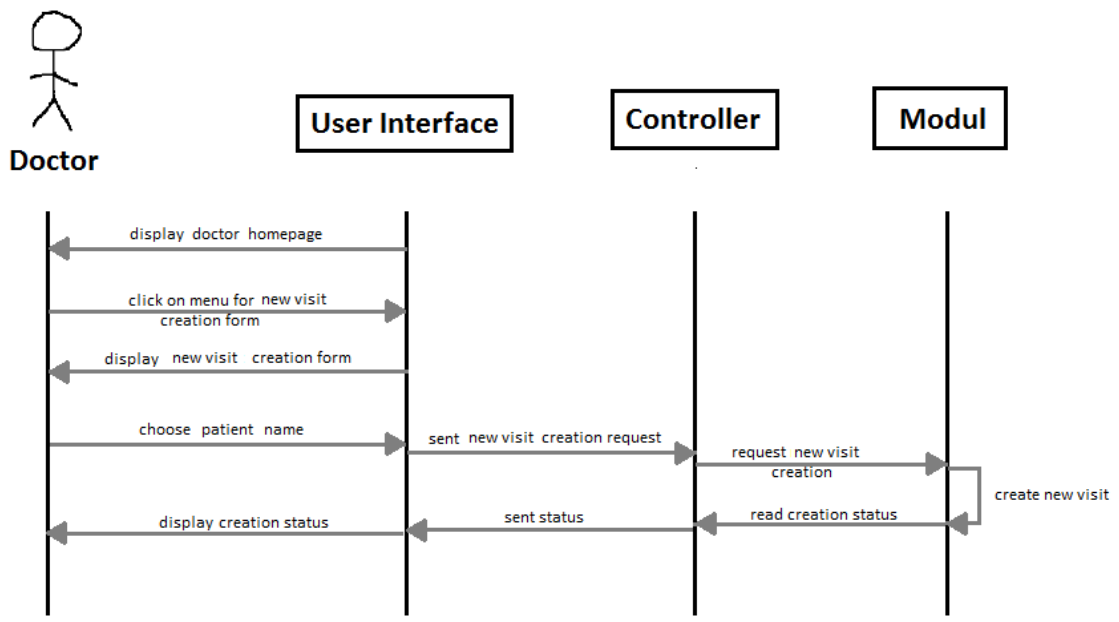
- Create new doctor



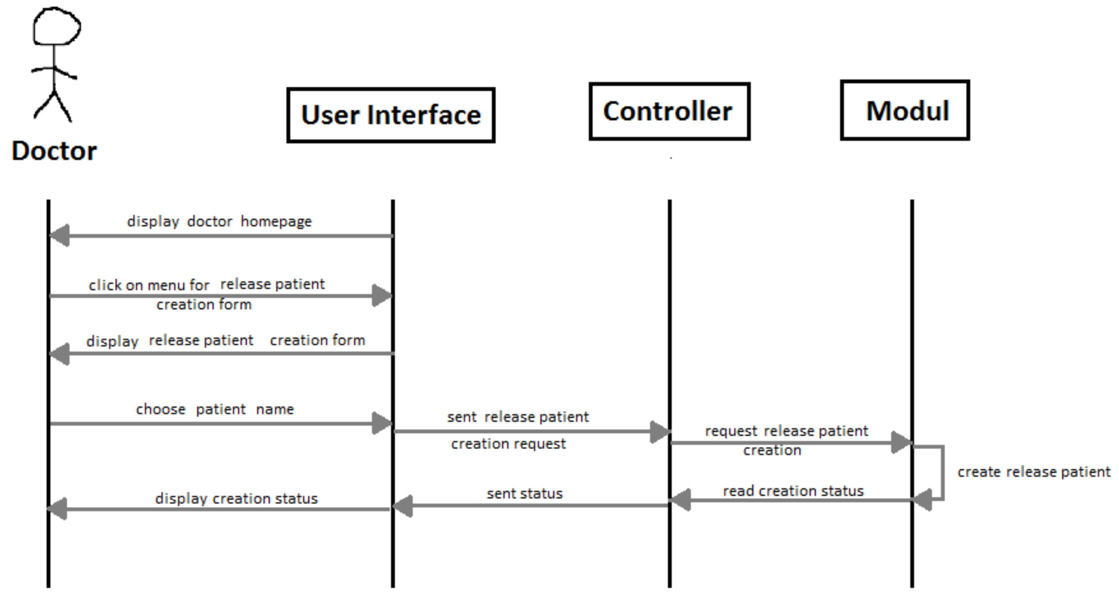
- Create new patient



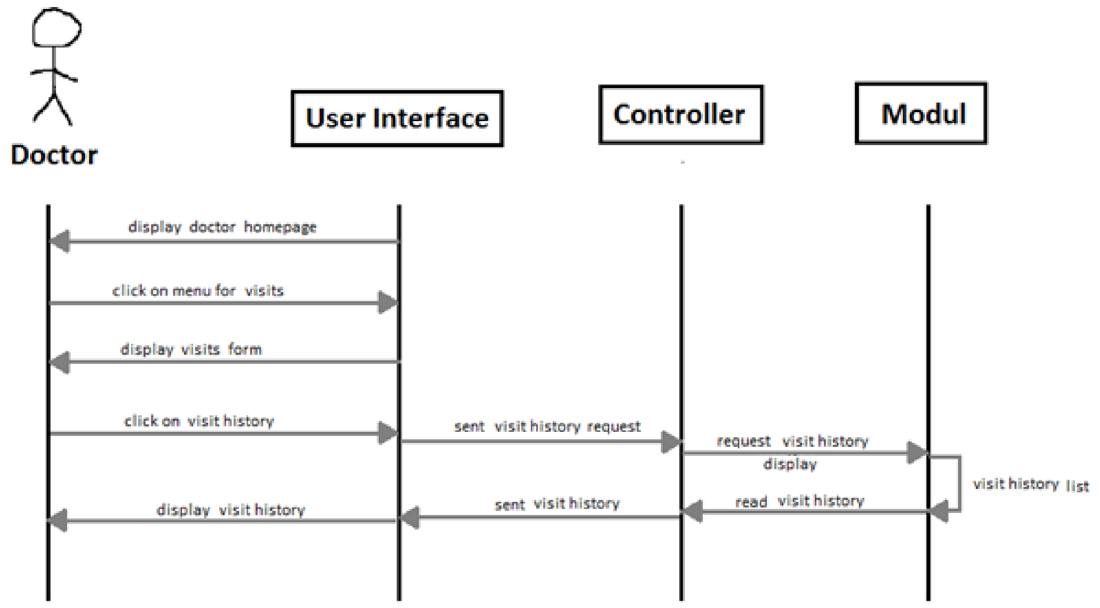
- Create new visit



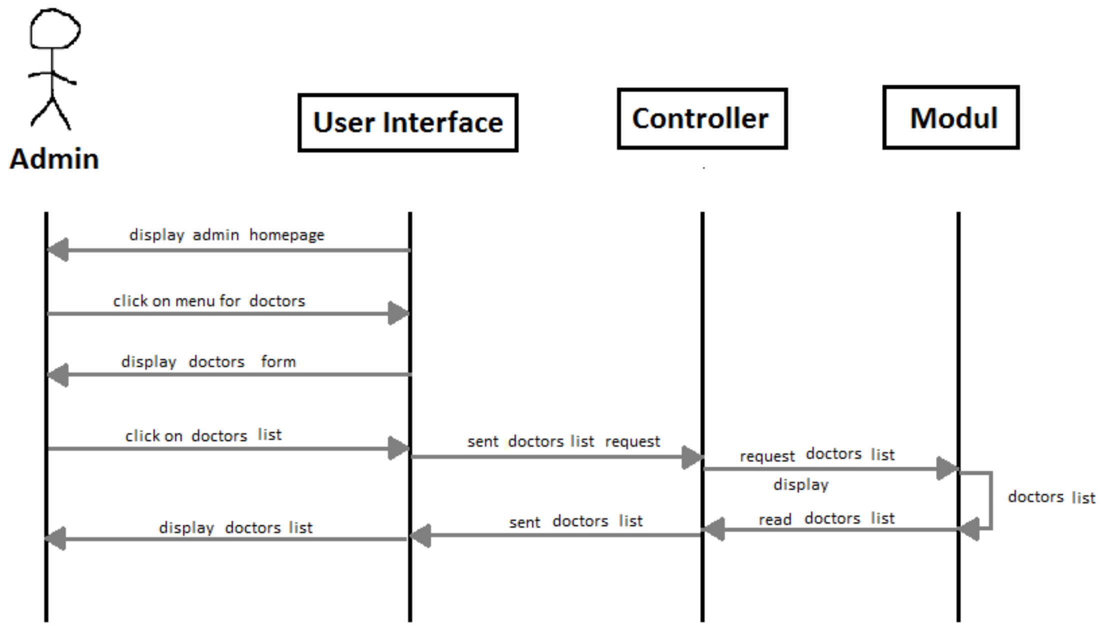
- Release patient


Doctor
User Interface
Controller
Modul
display doctor homepage
click on menu for release patient creation form
display release patient creation form
choose patient name
sent release patient creation request
request release patient creation
create release patient
read creation status
sent status
display creation status

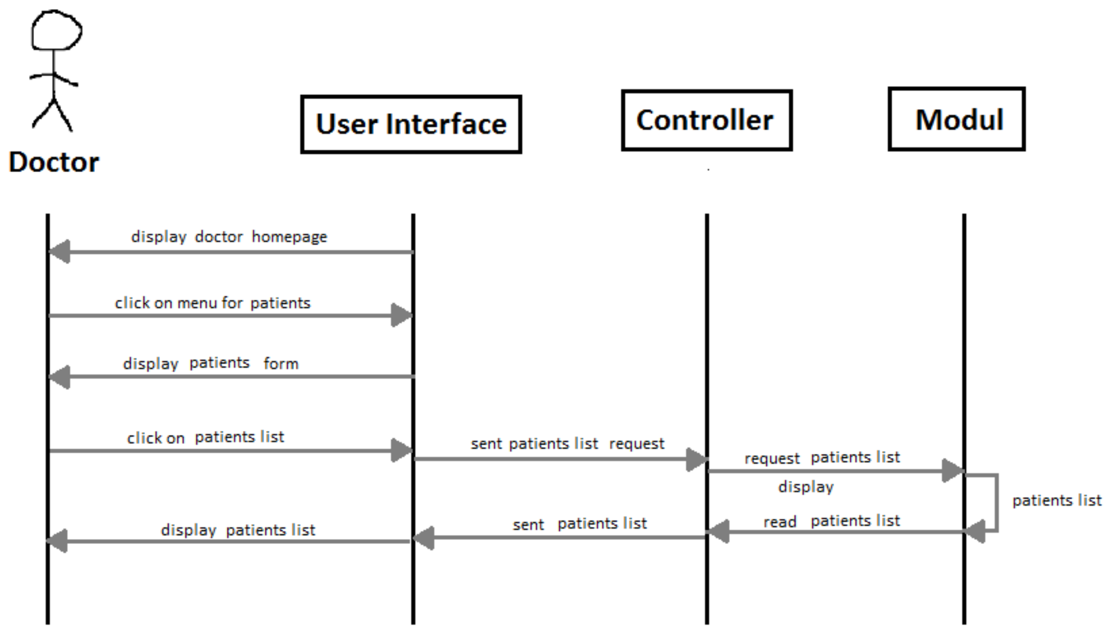
- Display visit history



- Display doctors list



- Display patients list



2. Write an essay about UML (modeling methodology, diagrams)

Unified Modeling Language (UML) - general-purpose modeling language in the field of software engineering. The Unified Modeling Language includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.

UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.

Modeling:
It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The model also contains documentation that drives the model elements and diagrams.

UML diagrams represent two different views of a system model:

- Static (or structural) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- Dynamic (or behavioral) view: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

Diagrams:

UML has two types of diagrams. Diagram types represent structural information, and the other represent general types of behavior, including four that represent different aspects of interactions.

Structure diagrams

Structure diagrams emphasize the things that must be present in the system being modeled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems.

- Class diagram: describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes.
- Component diagram: describes how a software system is split up into components and shows the dependencies among these components.
- Composite structure diagram: describes the internal structure of a class and the collaborations that this structure makes possible.
- Deployment diagram: describes the hardware used in system implementations and the execution environments and artifacts deployed on the hardware.
- Object diagram: shows a complete or partial view of the structure of an example modeled system at a specific time.
- Package diagram: describes how a system is split up into logical groupings by showing the dependencies among these groupings.
- Profile diagram: operates at the metamodel level to show stereotypes as classes with the <<stereotype>> stereotype, and profiles as packages with the <<profile>> stereotype. The extension relation (solid line with closed,

filled arrowhead) indicates what metamodel element a given stereotype is extending.

Behavior diagrams

Behavior diagrams emphasize what must happen in the system being modeled. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

- Activity diagram: describes the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.
- UML state machine diagram: describes the states and state transitions of the system.
- Use Case Diagram: describes the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases.