CS1 Task 7 Domain Model and Sequence Diagram

**Team Green (Melisa Zoronjic, Manuel Pfister, Katsiaryna Mlinaric, Jonathan Drewlow und Philipp Noser)**

1. **UML Domain Model**

With the help of Responsibility Driving Design, the possible classes for our application were defined. As a result, the Domain Model Diagram was created, which presents conceptual or domain classes and defines the relations between them.

* In the first line, the objects *Patient* and *Doctor* were defined because such people play a central role in processes, occurring within MNS-PMS. In advance, it was admitted that the objects of Patient and Doctor domain classes can have some common attributes. That is why the conceptual class *Person* has been created and the **generalisation** has been used to show the relations between these three concepts (Person-Doctor-Patient).
* Because Patient and Doctor must collaborate with each other the new domain class named *Case* was defined. This is a central concept because interactions between Patient in a way that one doctor has many cases, and on the other hand many doctors can work on one case. The association between Patient and Case is defined like a **composition** (every Case belongs to one Patient, there is no Case without Patient, if Patient does not exist anymore, Case also does not exist).

Case

Patient

* Every Patient must be treated by Doctor. That means he should have a diagnosis and get some medication. To describe exactly how it is possible, the collaboration within the domain class Case were three more concepts defined, named *Diagnose*, *Treatment* and *Medication*. The association between the domains Case and Diagnose is defined as **aggregation,** between Diagnosis and Treatment too, and between Treatment and Medication, is the associations with defined cardinalities.

Case

Diagnose

The whole domain model is shown on the Picture 1.

1..\* is a

1

Doctor

1

treats

0..\*

Person

1

1..\* is a

Medication

0..\*

defines

1

defines

1 1..\*

defines

1 1..\* 1..\*

1

has

1..\*

Treatment

Diagnose

Case

Patient

Picture 1. Domain Model.

1. **UML sequence diagram**

**2.1 Information update**

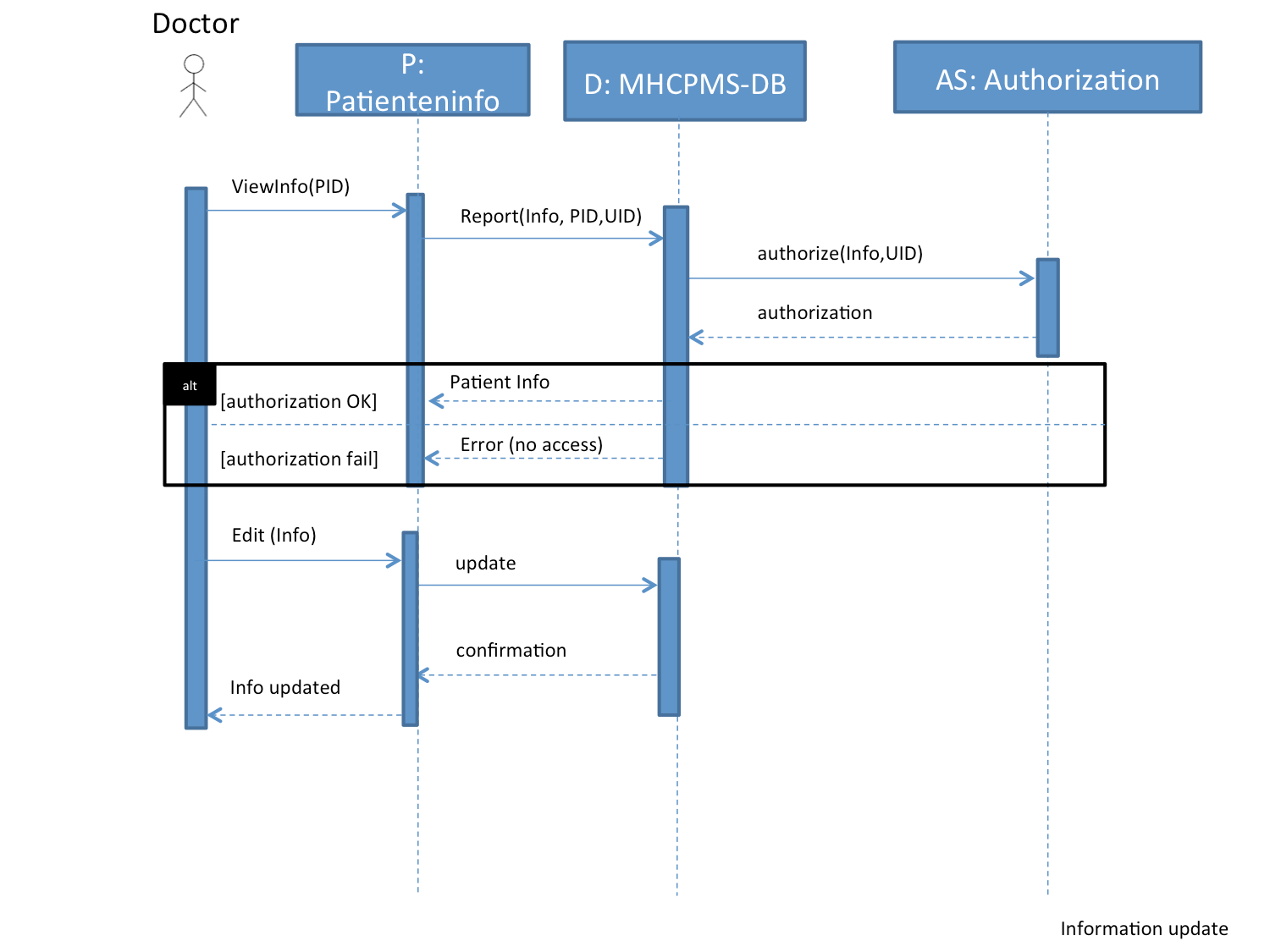


Diagram 1

The sequence diagram of Diagram 1 describes the process when the user updates some information in our software. The user starts with:

1. Enter the PID and get access to Patient Information
2. With the PID,UID and Patient Information we get access to our database
3. The system tests if the user has permission to the information in our database
4. If the user has the necessary rights, information is shown out of our database in the Patient information system
5. If the user does not have the necessary rights, the information out of the database will not be shown on the user and he is out of system.
6. In situation 4 the user has the right to edit patient Information’s.
7. New information will be added to the database
8. The data base let the patient Information system know that information is now up to date.
9. The user gets a message, that all information is up to date.

**2.2 Information about some specific event**

The sequence diagram of Diagram 2 describes the process when system alerts the user about the specific even. The user starts with:

1. Enter the PID and get access to Patient Information system.
2. With the PID, UID the Patient Information system checks on database if new information is available.
3. The system tests if the user has permit to the information in our database
4. If no new information is available, the database tells that fact on the Patient information system. No system alert has to turn on to the user.
5. If the user does not have the necessary rights, the information out of the database will not be shown to the user and user is out of system.
6. If the user has the necessary rights, and new information is available, the database tells that fact to Patient Information system and then to the user.

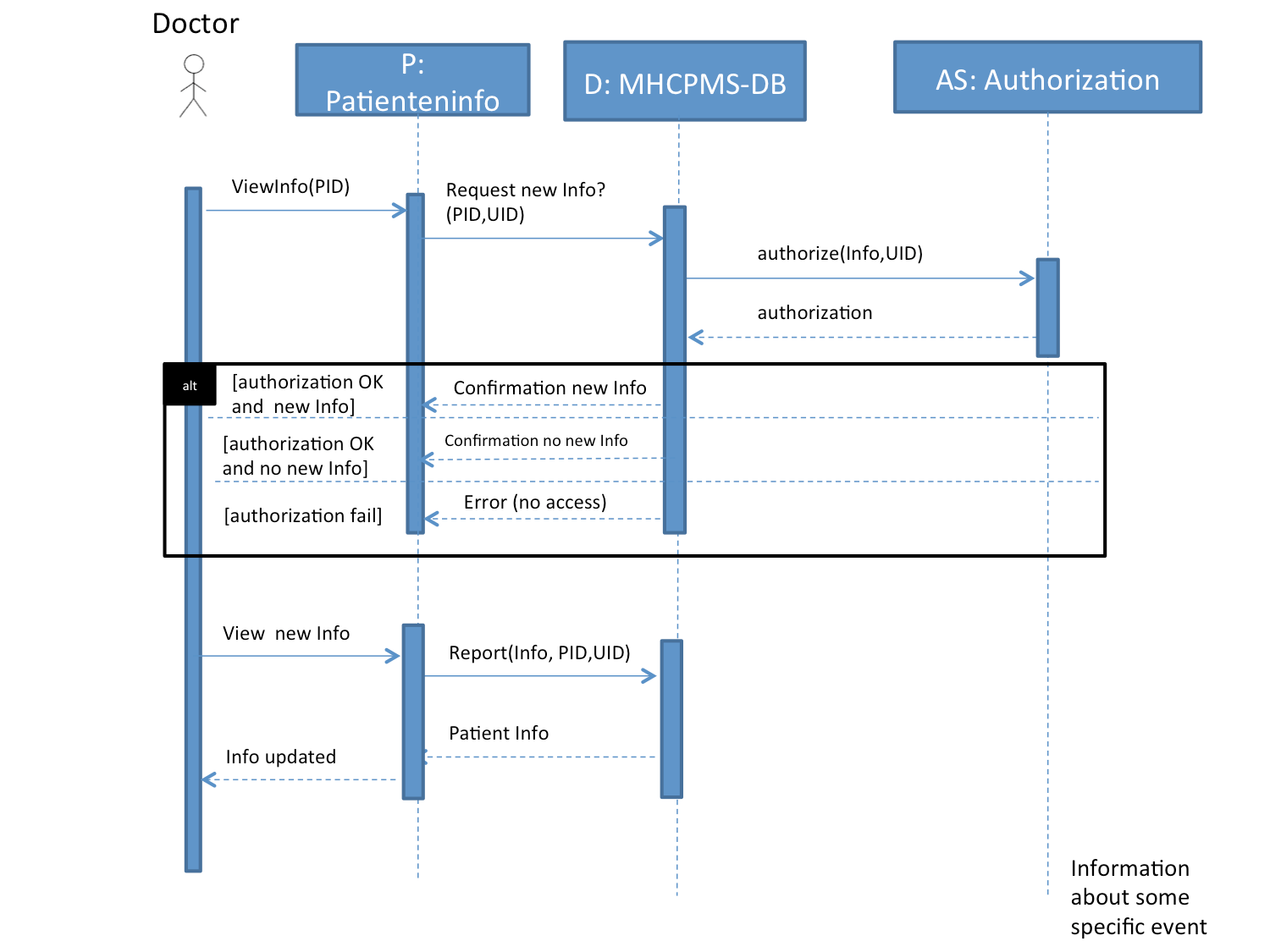
****

Diagram 2

1. **Refined UML domain model**

After creating two sequence diagrams describing two different situations, the Domain Model has been refined.

There are attributes and operations which can be used at a later stage – creation of classes’ diagram.

**Attributes**:

*PID* –patient’s identification number (belongs to class Patient)

*UID*- user’s identification number (belongs to class Doctor)

**Operations can be used later as methods of classes**:

*viewInfo()*

*edit(argument)*

*distributeInfo*

With this it is possible to modify three classes on Domain Model:

Doctor

Patient

**()**

UID

PID

viewInfo()

edit(argument)

Case

distributeInfo()