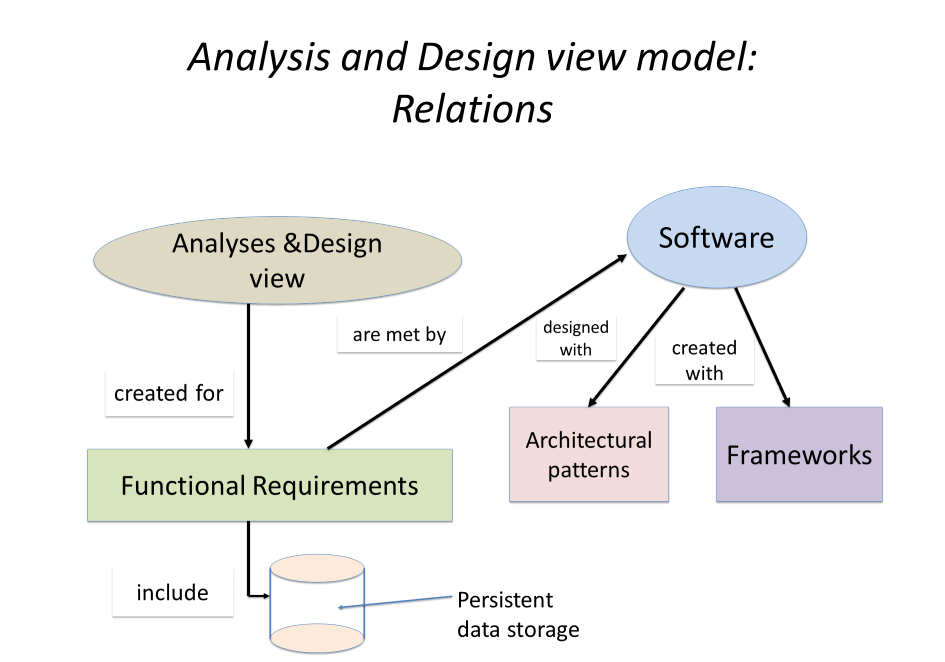
**CS1 Task 9: Software Architecture**

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

**"Analysis & Design View" model of software architecture.**



1. **Identifying software components and their relationships and interactions based on system requirements.**

Analysis and Design view model is for functional requirements.

*Functional system requirements:*

* 1. Our user (Doctor) can login in system

1.2 Our user can create and modify patient’s records

* can change master data (Stammdaten Patient)
* can change patient information (appointments, treatments, diagnosis, prescriptions)
* can write comments and notes and upload documents

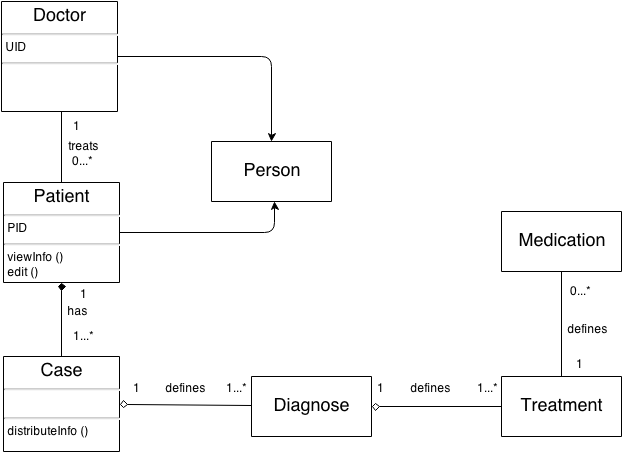
1.3 The user can retrieve data from the system, it means he or she can read patients records

* 1. The user can distribute patient information
  2. User can handle drug prescription

* 1. System provides access to persistent data storage (DB)
  2. System provides information storage

1. **Identifying software components and their relationships and interactions based on refined domain model.**

We have defined 7 classes and relationship between them. We have corrected domain model after correcting the sequence diagram.



Depending of system requirements more functions and class attributes could be defined and from domain model class diagram can be build.

1. **Framework Vaadin**

*Vaadin provides:*

- a UI for the user to interface with business logic and data of the application (realization of Form and Menu Manager)

- a data model for binding presented in field component (text fields, check boxes) to a data source (realization of Patient-Info Manager)

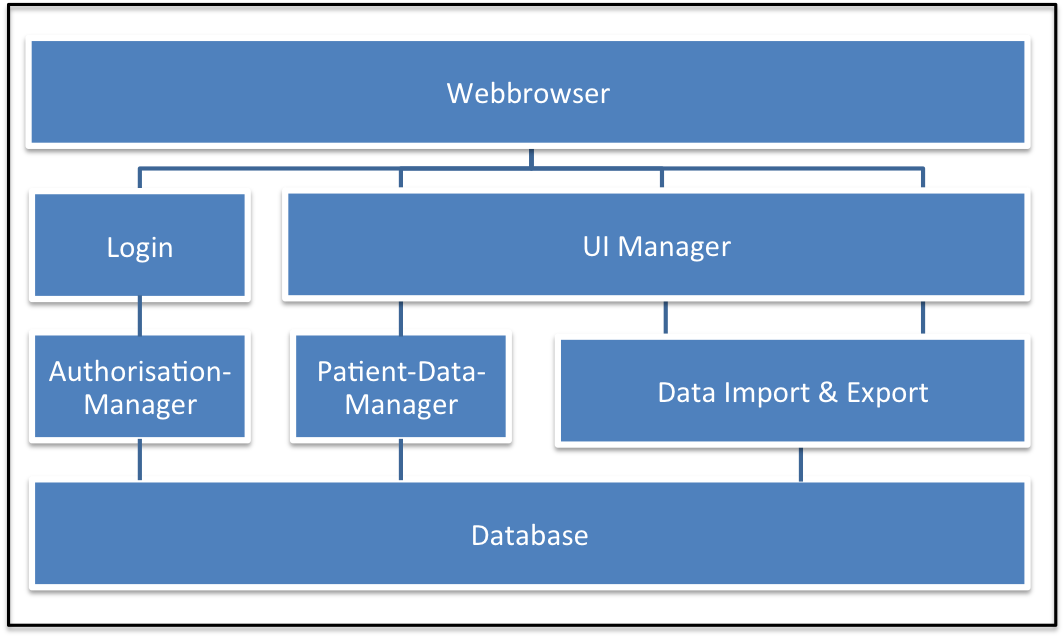
- a class for Login checking (Authorization Manager), we have to implement using the frame of Vaadin

1. **Access to DB**

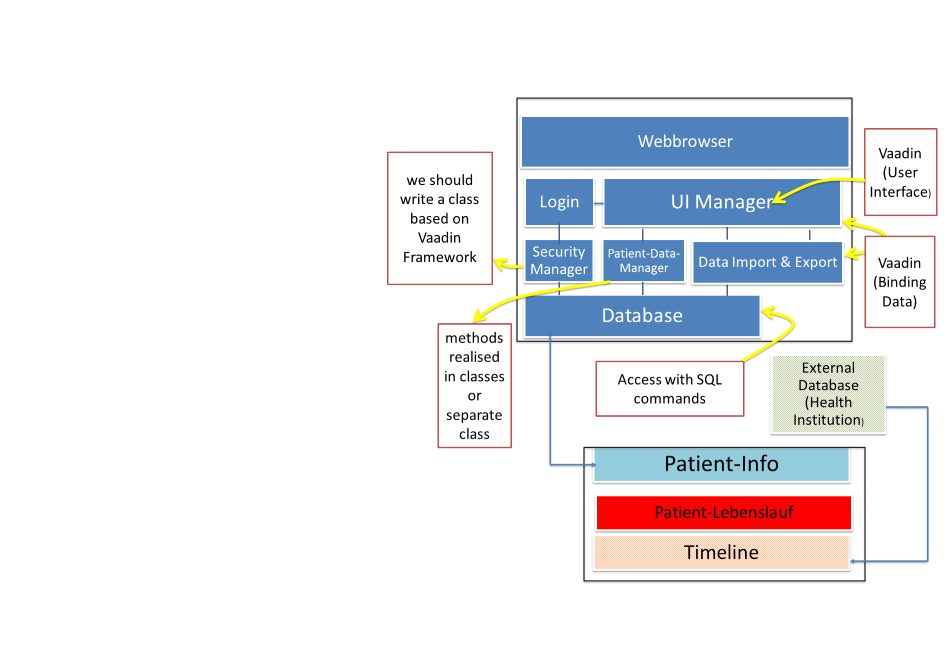
Software should provide an access to DB, for this we need *SQL requests* (Data import and export)

1. **Architectural design patterns**

Our software is web based information system which can be based on a Layer Architecture Pattern



We also discussed the design and the architecture as a group and enhanced the diagram with some additional information:

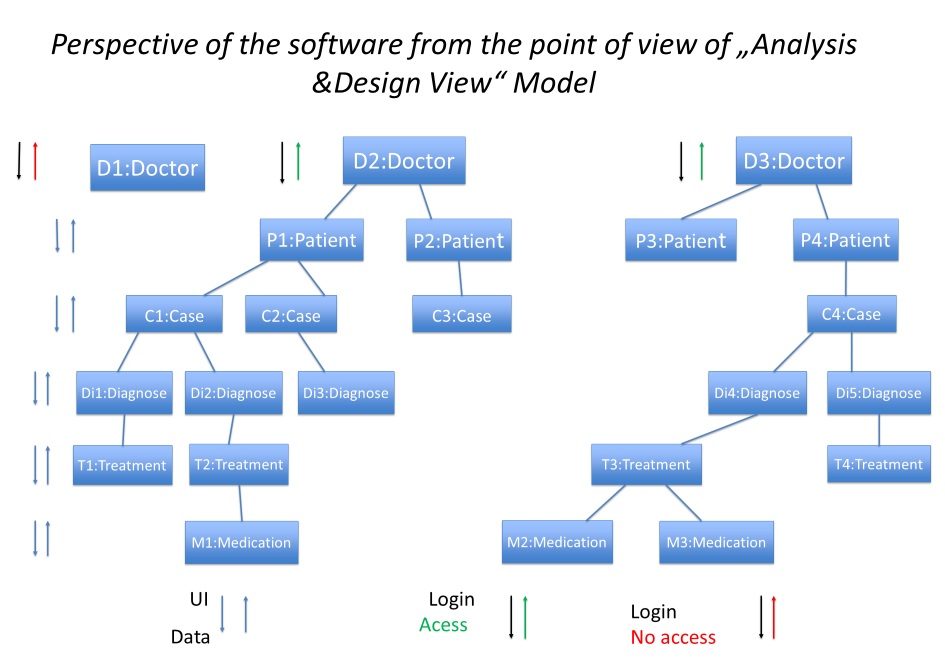


Explanation: in the left box are our main components or features of the product. The Patient-Info where the doctor can see the diagnose, treatment, medication, etc. of the patient. The feature of the Patient CV (Lebenslauf) is where the doctor can add notes about life events of the patients (for example an incident in childhood, marriage, etc.) The Timeline is generated by information from an External Database with information of several Health Institutions. It contains the medical information of the patient, for example a surgery, hospital visit, etc.)

The red boxes describe our thoughts about the different tasks.

1. **Analysis and Design view model**

Based on the analysis above, we can represent a view of the relations between the abstract instances as objects of different classes through the URL Object diagram.



With this diagram it is possible to image which processes occur according functional requirements. And looking at architectural pattern it is easier to image how exactly these processes are going. It is also possible to see how User Interface and Data (in database or in between data storage) combine.