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.

*System requirements*.

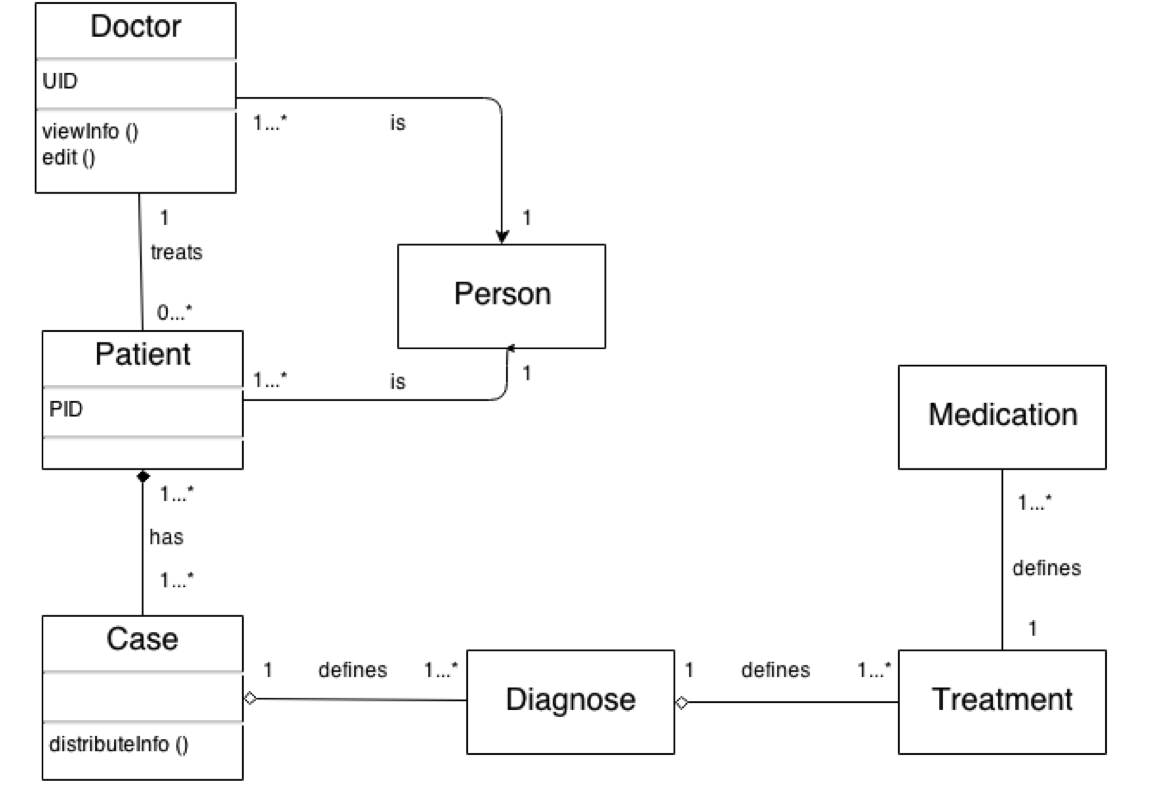
* 1. Functional system requirements:
     1. Our user (doctor) can create and modify patients records
* can change master data
* can make appointments?
* can change patient information ( appoinments, treatments, diagnosis, prescriptions)
* can write comments(using TimeLine?)
  + 1. The user can retrieve data from the system, it means he or she can read patients records
    2. The user can distribute patient information
    3. User can handle drug prescription
    4. System generates reports(on requests by Mental Health Service management)
    5. System provides access to persistent data storage (DB)
    6. System provides data storage
  1. Non functional system requirements:
     1. Security requirements:
* No access for not authorized users
* Patients privacy (no personal data for medical records staff)
  + 1. Usability requirements
* Friendly UI

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.

/\*in class Patient public void viewInfo(PID)\*/ /\* edit(PID)\*/

/\*in class Doctor can be public boolean getAccess(UID)\*/



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

1. Architectural design patterns

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

/\*we can combine this picture from Folien with picture of Jonathan\*/

Web browser

Login Form &Menu Manager Data validation

Security Manager PatientInfo Manager Data import&export Report generation

Patient DB

4. 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 sourse

(realisation of PatientInfo Manager)

- a class for Login checking (Security Manager), we must implement it 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. Resulting “Analyses and Design view”

As a result of analyses above we present the class diagram:

/\*Bild\*/