



SYSTEMARCHITEKTUR MIT VAADIN

MHC-PMS Team YELLOW

Michelle Lüscher, Simon Oppliger,
Yaron Walter, Lucien Heuzeveldt,
Nadine Siegfried

VERSION	BESCHREIBUNG	NAME	DATUM
0.1	Initiales Dokument	Simon Oppliger	01.05.2019
0.2	DeploymentDiagramm	Michelle Lüscher	01.05.2019
0.3	ComponentDiagramm	Michelle Lüscher	02.05.2019
0.4	Komponentendiagramm	Yaron Walter	05.05.2019
1.0	Freigabe des Dokuments und letzte Anpassungen	Team	06.05.2019

Inhalt

1. Vorwort.....	2
2. Glossar	2
3. Deployment Diagramm.....	3
4. Component Diagram	4
5. Klassendiagramme	5
5.1 Model.....	5
5.2 Main View	6
6. Anhang.....	6
6.1 Abbildungsverzeichnis	6
6.2 Tabellenverzeichnis	6

1. Vorwort

In diesem Dokument wird Task 5 aus dem «Software Engineering and Design»-Kurs der BFH abgehandelt. Im Rahmen dieses Tasks wurde folgendes gefordert:

For your application

- model the software architecture
 - o high-level: UML component and deployment diagram
 - o detailed: UML class diagram for main view of your application
- taking into account
 - o system requirements
 - incl. the need for persistent data storage
 - o architectural design pattern: MVP
 - o the Vaadin framework

Die Erzeugnisse dieser Aufgabe werden hier in derselben Reihenfolge präsentiert.

Das Dokument richtet sich an technische Fachleute, es werden Fachwörter verwendet. Das Wissen aus den vorhergehenden Task-Dokumenten wird ebenfalls vorausgesetzt. Aufgrund von früheren Rückmeldungen der Lehrpersonen wurde das Wiki in ein FAQ umbenannt

2. Glossar

Begriff	Erklärung
PMS	Patientenmanagementsystem
MHC	Mental Health Care
SoED	Software Engineering and Design (Modul an der BFH-Bern)
BFH	Berner Fachhochschule
Vaadin	Framework für Web-Applikationen in Java
UML	Unified Modelling Language – Ein Standard für Diagramme im Bereich der Software-Modellierung.
Use Case	Anwendungsfall einer Applikation
Framework	Programmiergerüst für eine Applikation mit vielen Hilfsmethoden
FAQ	Frequently Asked Questions

Tabelle 1: Glossar

3. Deployment Diagramm

Im Deployment-Diagramm ist ersichtlich, wie ein einfacher Betrieb der Applikation vorgesehen ist.

Anmerkungen:

- Es handelt sich um eine Aufzeichnung der mindestens lauffähigen Konfiguration.
- Themen wie redundante Datenspeicherung können mit einer umfangreicheren Konfiguration der Datenbankstruktur erreicht werden.

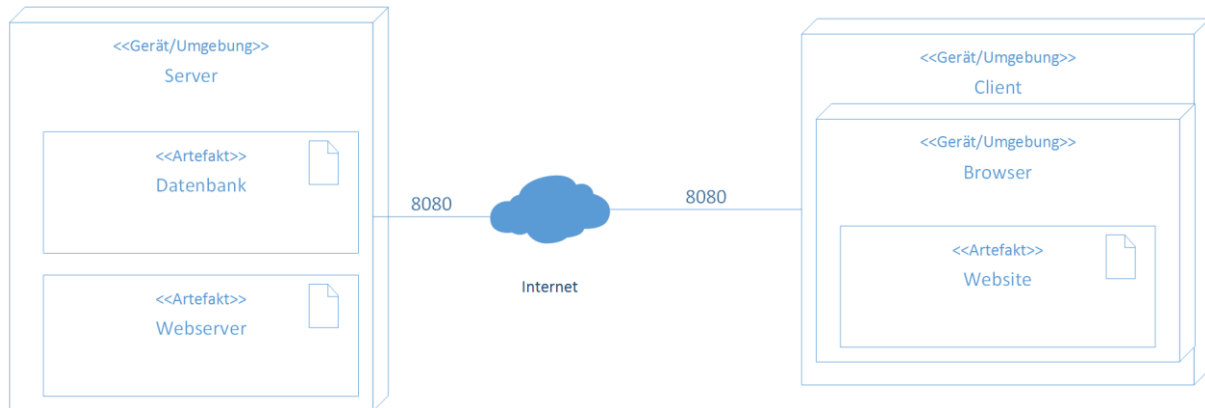


Abbildung 1: Deployment-Diagramm

Das Diagramm ist als Bild in höherer Auflösung im GIT-Verzeichnis zu Task 5 zu finden.

4. Component Diagramm

Das Komponentendiagramm bildet aus einer groben Sicht die relevanten Teile unserer Applikation ab.

Anmerkungen:

- Die User-Komponente ist in erster Linie aufgrund der Rechte mit den Komponenten Faq und Seminar verbunden.
- Es wurde zugunsten der Lesbarkeit keine Unterteilung der View in die einzelnen Ansichten vorgenommen. Trotzdem wird die Applikation aus verschiedenen Ansichten bestehen.
- Die Komponenten Seminar, Faq und Forum beinhalten jeweils weitere Komponenten Manager, DTO und Repositories (vom Repository Pattern), welche aber der Übersicht wegen weggelassen wurden.

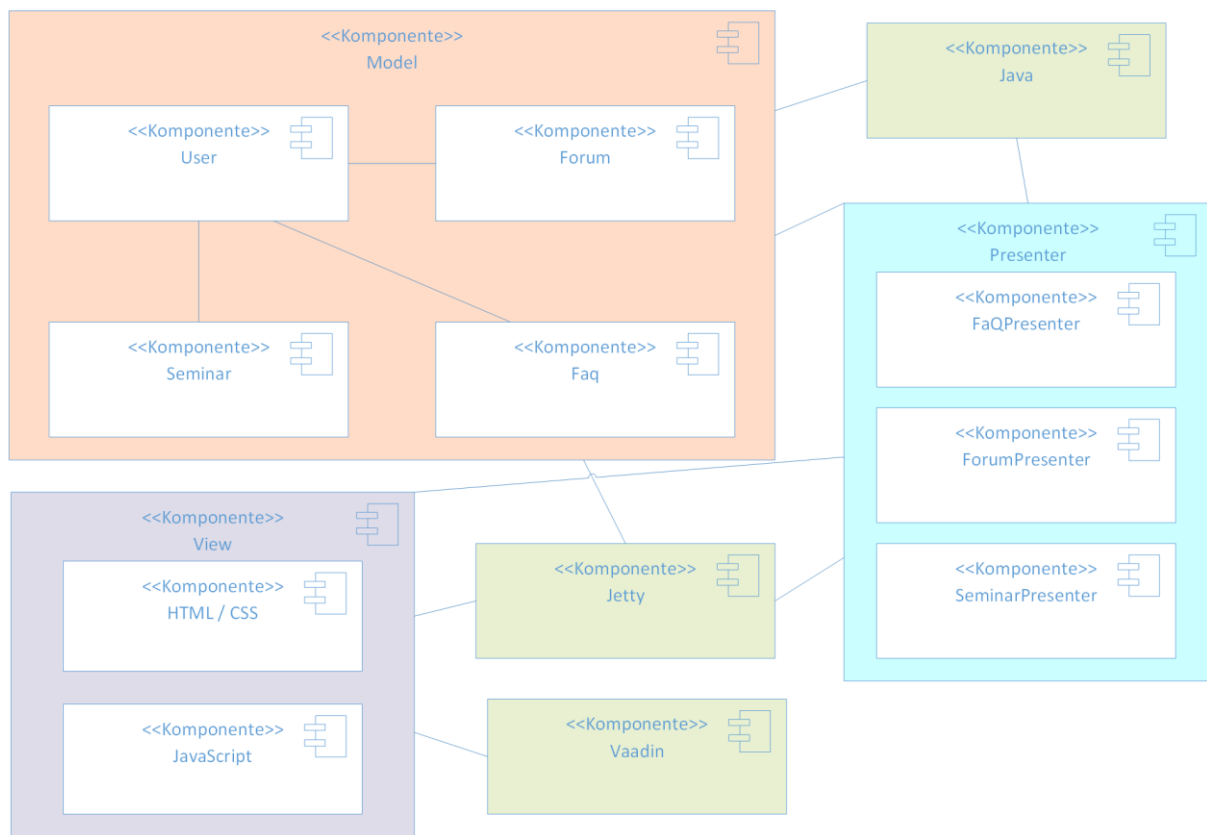


Abbildung 2: Komponentendiagramm

Das Diagramm ist als Bild in höherer Auflösung im GIT-Verzeichnis zu Task 5 zu finden.

5. Klassendiagramme

5.1 Model

Das Model-Package beinhaltet die inhaltlichen Objekte unserer Applikation.

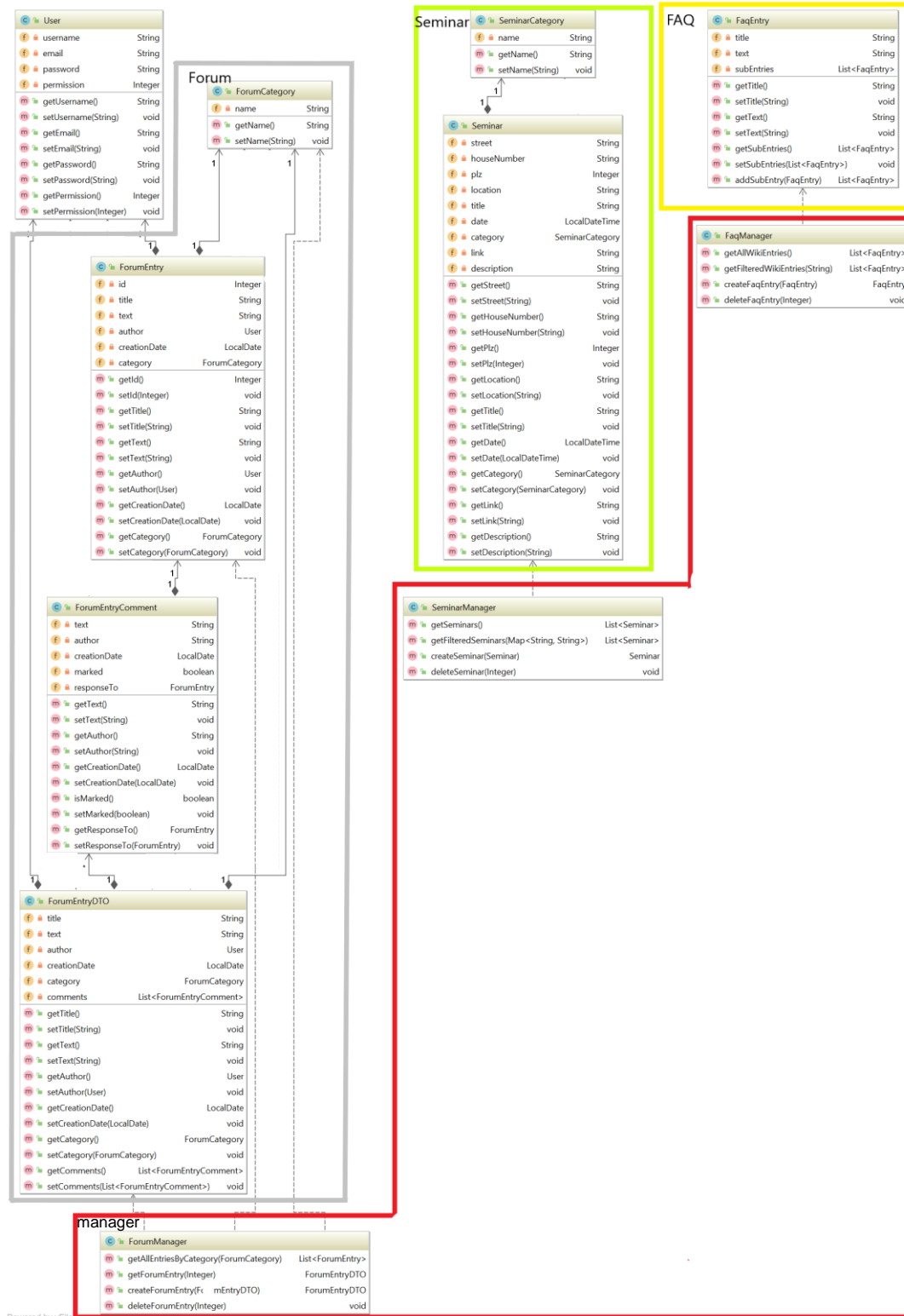


Abbildung 3: Modell-Package

Das Diagramm ist als Bild in höherer Auflösung im GIT-Verzeichnis zu Task 5 zu finden.

5.2 Main View

In unserem View-Package finden sich bis anhin folgende Klassen. In der Layout-Klasse befindet sich der grundlegende Seitenaufbau mit der Menubar, das den Wechsel zwischen den Features ermöglicht.

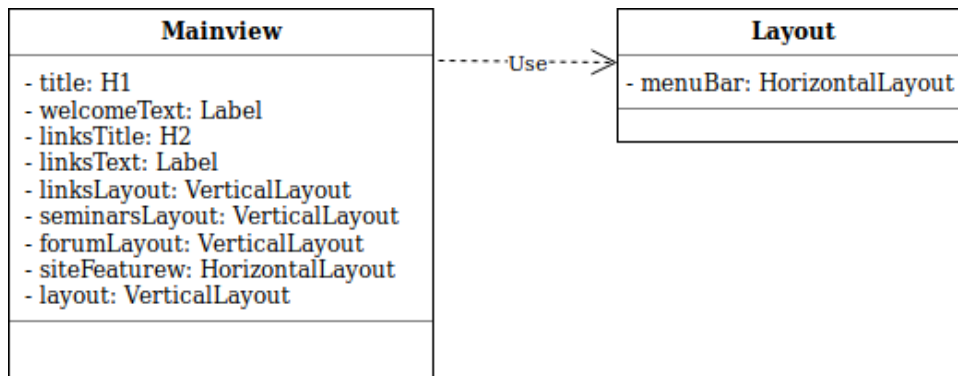


Abbildung 4: View package

Das Diagramm ist als Bild in höherer Auflösung im GIT-Verzeichnis zu Task 5 zu finden.

6. Anhang

6.1 Abbildungsverzeichnis

Abbildung 1: Deployment-Diagramm	3
Abbildung 2: Komponentendiagramm	4
Abbildung 3: Modell-Package	5
Abbildung 4: View package	6

6.2 Tabellenverzeichnis

Tabelle 1: Glossar	2
--------------------------	---