



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences



Task07

*David Gaupp, Marko Miletic, Sugeelan Selvasingham, Viktor Velkov,
Janahan Sellathurai, Alain Nippel*

Titelbild aktualisieren

Task07: Web Application

1. *Live Demo*
2. *Implementierte Features*
3. *Highlights*
4. *Gelernte Lektionen*
5. *Appendix*

Live Demo

Berner Fachhochschule | Haute école spécialisée bernoise | Bern University of Applied Sciences

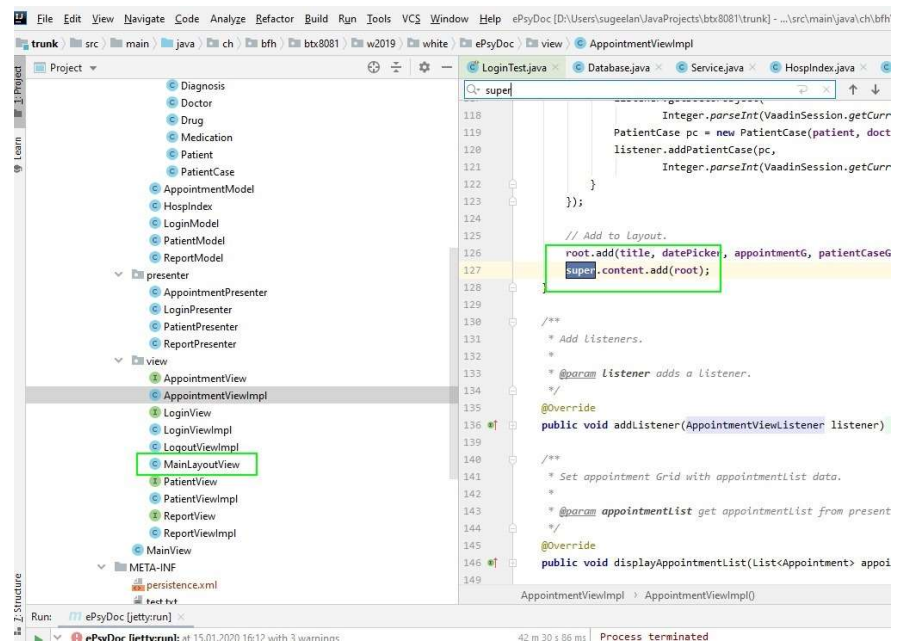
Deployment Diagramme werden verwendet, um die Hardware Knoten/Geräte eines Systems sowie die Verbindungen der Kommunikation zwischen ihnen darzustellen.

Implementierte Features

Design Thinking Ideas	Implemented Features
Patienten erfassen	☹️
Patientendatenabruf aus EPD	😊
Medikamenten verschreiben -> hospINDEX	😊
Medikamenten plausibilisieren like CDS	☹️
Terminverwaltung	😐
Externes Login	😐
Vorausgefüllte Formulare ausdrucken	☹️
Absturzsichere Anwendung	😊
Benutzerfreundliche Anwendung	?

Highlights

- ▶ Modularer Aufbau der Views
- ▶ MainLayoutView als Grundstruktur
- ▶ Restliche Views erben davon
- ▶ Bausteine durch «Super»-Aufruf hinzugefügt



Gelernte Lektionen

- ▶ *Allgemeine Kommunikation*
- ▶ *Umgang bei Krankheit*
- ▶ *Team*
 - ▶ *Vorgehen zu Beginn unklar*
 - ▶ *unterschiedlicher Wissenstand*
 - ▶ *Motivation*

Vielen Dank für eure Aufmerksamkeit

- *Noch Fragen?*



Bilderquellen

► **Titelbild:**

<https://image.jimcdn.com/app/cms/image/transf/dimension=697x10000:format=jpg/path/sd47613ceb1db53f4/image/ib674891e8fb5286a/version/1515513892/patientenzimmer-untersuchungszimmer-praxis-hausarzt-friedeck-rothenburg.jpg>

Anhang

Berner Fachhochschule | Haute école spécialisée bernoise | Bern University of Applied Sciences

Liste der persönlichen Beiträge der Teammitglieder

- ▶ **Gaup David:**
 - ▶ *Testen(N), Datenbank-Logik(N), Diagnose-Klasse(H)*
- ▶ **Milketic Marko:**
 - ▶ *Report(H), Medikation(H), MVP-Umsetzung(H), Diagnose-Klasse(N)*
- ▶ **Selvasingham Sugeelan:**
 - ▶ *Scrum(H), Datenbank-Logik(H), Login(H), MVP-Umsetzung(N), Home-Ansicht und Navigation(N),*
- ▶ **Velkov Viktor:**
 - ▶ *Testen(H), Mediplan(N), Diagnose-Klasse(N)*
- ▶ **Sellathurai Janahan:**
 - ▶ *Patient Stammdaten anzeigen(H), Scrum(N), Datenbank-Logik(N), Login(N)*
- ▶ **Nippel Alain:**
 - ▶ *Home-Ansicht und Navigation(H), MVP-umsetzung(H), Patient Stammdaten anzeigen (N), Report(N), Medikation(N)*

H = Hauptverantwortlich/Hauptentwickler

N = Nebenverantwortlicher/Mitentwickler

Berner Fachhochschule | Haute école spécialisée bernoise | Bern University of Applied Sciences

Product Backlog

ID	Story Name	Description	Priority	Effort Plan Original	Effort Plan Updated	Effort Actual	Status
1	Main-Gerüst	Projekt nach Vaadin erstellen. Alle notwendigen Views und Klassen erstellen.	high	42	57	59	done
2	Medikamenten-Verwaltung	Der Facharzt kann die Medizinverordnung eines Patienten im Medikamenten-View strukturiert sehen. Der Facharzt kann neue Medizinverordnungen einem Patienten zuordnen und abfragen.	high	97	111	167	done
3	Patienten-Verwaltung	Der Facharzt kann die Patientenstammdaten und die Patientenberichte auf dem Patienten-View strukturiert sehen.	high	20	18	25	done
4	Termin-Verwaltung	Der Facharzt kann seine Patiententermine auf dem Termin-View strukturiert sehen. Die Termine werden den Patienten zugeordnet.	medium	4	4	4	done
5	Authentisierung	Um unsere Applikation verwenden zu können, muss sich der Facharzt auf der Login-View authentifizieren. Damit stellen wir sicher, dass die Patientendaten von aussen nicht für Drittpersonen sichtbar sind.	medium	5	6	17	done
6	Datenbank	Um die Patientendaten persistent zu machen, wird eine relationale Datenbank verwendet.	high				cancelled
7	Admin-Funktion	Der Arzt kann Einstellungen vornehmen, um die Benutzeroberfläche anzupassen oder Passwort zu verändern	low				cancelled

Berner Fachhochschule | Haute école spécialisée bernoise | Bern University of Applied Sciences

Product backlog aktualisieren

Sprint Backlog 1

ID	Sprint	Name	Description	Components	Owner	Reviewer	Priority	Effort Plan Original	Effort Plan Updated	Effort Actual	Status
1.1	1	Navigation	Navigationskonzept erarbeiten + Im Vaadin Elemente für das GUI überprüfen	View	Alle	Alle	high	12	8	9	done
1.2	1	Grundgerüst	Alle Klassen gemäss Klassendiagramm erstellen (nur Grundgerüst)	Model / View / Presenter	Alle	Alle	high	6	3	3	done
1.3	1	GUI loginView	GUI für loginView erstellen und Navigation implementieren	View	selvs1	sellj1	medium	4	12	18	done
1.4	1	GUI mainView	GUI für mainView erstellen und Navigation implementieren	View	Gaupa1	selvs1	high	4	1	3	done
1.5	1	GUI patientView	GUI für patientView erstellen und Navigation implementieren	View	sellj1	velkv1	high	4	6	6	done
1.6	1	GUI medicationView	GUI für medicationView erstellen und Navigation implementieren	View	milem2	Gaupa1	medium	4	6	6	done
1.7	1	GUI reportView	GUI für reportView erstellen und Navigation implementieren	View	velkv1	nippa1	medium	4	3	3	done
1.8	1	GUI appointmentView	GUI für appointmentView erstellen und Navigation implementieren	View	nippa1	milem2	medium	4	3	3	done
1.9	1	patientView	patientView Logik nach MVP implementieren inkl. trivialer Getter-Methoden für Patientenaufistung.	Model / View	sellj1	nippa1	high	12	15	17	done

Sprint Backlog 2

ID	Sprint	Name	Description	Components	Owner	Reviewer	Priority	Effort Plan Original	Effort Plan Updated	Effort Actual	Status
2.2	2	Datenabnk: ER Modell	Klassendiagramm aktualisieren, ER-Diagramm überprüfen.	Database	Gaupa1	velkv1	high	2	2	3	done
2.21	2	Medikation: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	Gaupa1	milem2	high	8	14	14	done
3.1	2	Dummydaten	Datenbank mit Patienten und Doktoren befüllen	Model / Database	sellj1	selvs1	high	1	7	10	done
2.3	2	Mediplan: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	velkv1	gaupa1	high	8	9	9	cancelled
2.4	2	Prescription: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	nippa1	milem2	medium	8	9	10	cancelled
3.2	2	Patient: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	sellj1	selvs1	high	8	11	11	done
2.6	2	Report: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	selvs1	sellj1	medium	5	6	3	done
2.7	2	Diagnose: Persistierung/View	Korrektur Design; JPA Annotation, Verknüpfung mit dem Presenter	Model / View / Presenter	selvs1	sellj1	high	8	10	14	done
2.8	2	Grundgerüst	Grundgerüst nach MVP entwickeln. Fehler beheben	Model / View / Presenter	nippa1, milem2	nippa1, milem2	high	28	28	28	done
2.9	2	IDE Setup	IDE für JPA vorbereiten	IDE	selvs1	sellj1	high	4	6	12	done
5.1	2	Login: Persistierung/View	Doctor Klasse Annotation und als Vorlage zur Verfügung stellen	Model / Presenter / Database	selvs1	sellj1	medium	5	4	17	done

Berner Fachhochschule | Haute école spécialisée bernoise | Bern University of Applied Sciences

Sprint Backlog 3

ID	Sprint	Name	Description	Components	Owner	Reviewer	Priority	Effort Plan Original	Effort Plan Updated	Effort Actual	Status
2.11	3	UI Korrektur	IDs durch Datum oder PatNamen ersetzen	View	nippa1, milem2	gaupa1, selvs1	medium	4	8	4	done
2.12	3	Baustellen fertigstellen	Datenbank	Model / Presenter / Database	selvs1	sellj1	high	8	4	8	done
2.13	3	JPA	modellieren Zusatzverbindungen Service Klasse (DBServiceQuery + FK)	Model / Presenter / Database	selvs1, gaupa1	sellj1	high	4	4	9	done
2.14	3	MVP	Struktur nach MVP Auslagerung der ServiceKlassen in Presenter.	Model / View / Presenter	nippa1	milem2	high	4	8	32	done
2.15	3	Code Inspection	Gesamtprojekt scannen und allfällige Fehler beheben und testen mit junit	Model / View / Presenter	gaupa1	velkv1	medium	9	9	20	done