



REQUIREMENTS

MHC-PMS Team YELLOW

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

Versionsverzeichnis

VERSION	BESCHREIBUNG	NAME	DATUM
0.1	Initiales Dokument	Nadine Siegfried	01.04.2019
0.2	Einführung	Nadine Siegfried	03.04.2019
0.3	Systemanforderungen	Nadine Siegfried, Yaron Walter, Lucien Heuzeveldt, Simon Oppliger, Michelle Lüscher	03.04.2019
0.4	Nutzeranforderungen	Nadine Siegfried, Yaron Walter, Lucien Heuzeveldt, Simon Oppliger, Michelle Lüscher	05.04.2019
0.5	Systemarchitektur	Lucien Heuzeveldt	05.04.2019
0.6	Weiterentwicklung	Yaron Walter	05.04.2019
0.7	System Modell	Simon Oppliger	06.04.2019
0.8	Testing	Michelle Lüscher	07.04.2019

Inhalt

Versionsverzeichnis.....	0
Vorwort.....	4
Einführung.....	4
Seminar-Finder.....	4
Wiki.....	4
Forum.....	4
Umsysteme.....	4
Glossar.....	5
Nutzeranforderungen – Übersicht.....	5
Seminar-Finder: Use Cases.....	5
Wiki: Use Cases.....	6
Forum: Use Cases.....	6
Allgemein: Use Cases.....	7
Nutzeranforderungen – Detaillierte Ausarbeitung.....	8
Use Case 1 – Bestehende Seminare anzeigen.....	8
Use Case 2 – Seminar hinzufügen.....	9
Use Case 3 – Detailansicht zu Seminar.....	10
Use Case 4 – Seminare filtern.....	11
Systemarchitektur.....	12
Frontend.....	12
Backend.....	12
Datenbank.....	12
Systemanforderungen.....	13
Funktionale Anforderungen.....	13
Nicht-funktionale Anforderungen.....	14
System Modell.....	15
Weiterentwicklung des Systems.....	16
Perspektiven.....	16
Seminar-Finder.....	16
Wiki.....	16
Forum.....	16
Allgemein.....	17
Änderungen am Technologiestack.....	17

Testing.....	18
Anhang	18
Abbildungsverzeichnis	18
Tabellenverzeichnis.....	18

Vorwort

In diesem Dokument werden die fachlichen Anforderungen an die Applikation (Patientenmanagementsystem: Sozialphobie - Fachbereich Angehörige) spezifiziert und erklärt. Weiter wird auch auf die erforderlichen technologischen Anforderungen eingegangen.

Da die Zielgruppen dieses Dokuments verschiedene Personen in verschiedenen Rollen sind, wurde darauf geachtet, eine möglichst natürliche Sprache zu verwenden. Zielgruppe der Applikation sind Angehörige von Patienten, während das Dokument selber in erster Linie an die Entwickler, bzw. die Auftraggeber seitens regionaler Gesundheitsbehörde gerichtet ist.

Das Dokument wurde in der Version 1 am 8. April 2019 fertiggestellt und durch den Auftraggeber abgenommen. Anpassungen an der abgeschlossenen Fassung sind mit den Verantwortlichen abzusprechen und müssen in diesem Kapitel aufgeführt sein.

Einführung

In unserer Applikation gibt es folgende Hauptfunktionen:

Seminar-Finder

Die Benutzer finden hier eine Übersicht über die anstehenden Seminare. Es kann nach Kategorien und Schlagworten durchsucht werden. Wird ein Seminar ausgewählt, werden weitere Informationen zum Anlass angezeigt, und ein Link zur offiziellen Seite wird dem Benutzer zur Verfügung gestellt.

Die Seminare werden durch die Experten gepflegt.

Wiki

Hier befinden sich eine Beschreibung, häufig gestellte Fragen und andere Informationen zur sozialen Phobie. Die Benutzer können das Wiki durchsuchen. Das Wiki wird von Experten unterhalten und aktualisiert.

Forum

Das Forum besteht aus verschiedenen Kategorien, welche durch die Moderatoren definiert werden. Die angemeldeten Benutzer können hier in einer Kategorie einen Beitrag erfassen, das kann eine Frage oder ein sonstiges Thema sein. Andere Benutzer können dann auf den Beitrag antworten. Wenn der Autor des Beitrags einen Kommentar sieht, welcher für ihn die Lösung des Problems darstellt, kann er diesen markieren. Die Beiträge/Kommentare von Experten werden speziell gekennzeichnet.

Wenn jemand keinen Account hat oder nicht angemeldet ist, kann er dennoch auf das Forum zugreifen, er hat aber keine Möglichkeit, selbst Beiträge zu erfassen oder Kommentare zu schreiben.

Umsysteme

Im Rahmen dieses Projektes ist keine spezielle Anbindung an andere Umsysteme angedacht. Aufgrund der Kartenfunktion im Seminar-Finder wird eine Anbindung an einen externen Kartendienst notwendig werden.

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
Technologiestack	In der Applikation verwendete Software
Black-Box-Testing	Die Tests werden ohne genaue Kenntnisse des Codes entwickelt und durchgeführt
Debugging	Fehlerbereinigung

Tabelle 1: Glossar

Nutzeranforderungen – Übersicht

Seminar-Finder: Use Cases

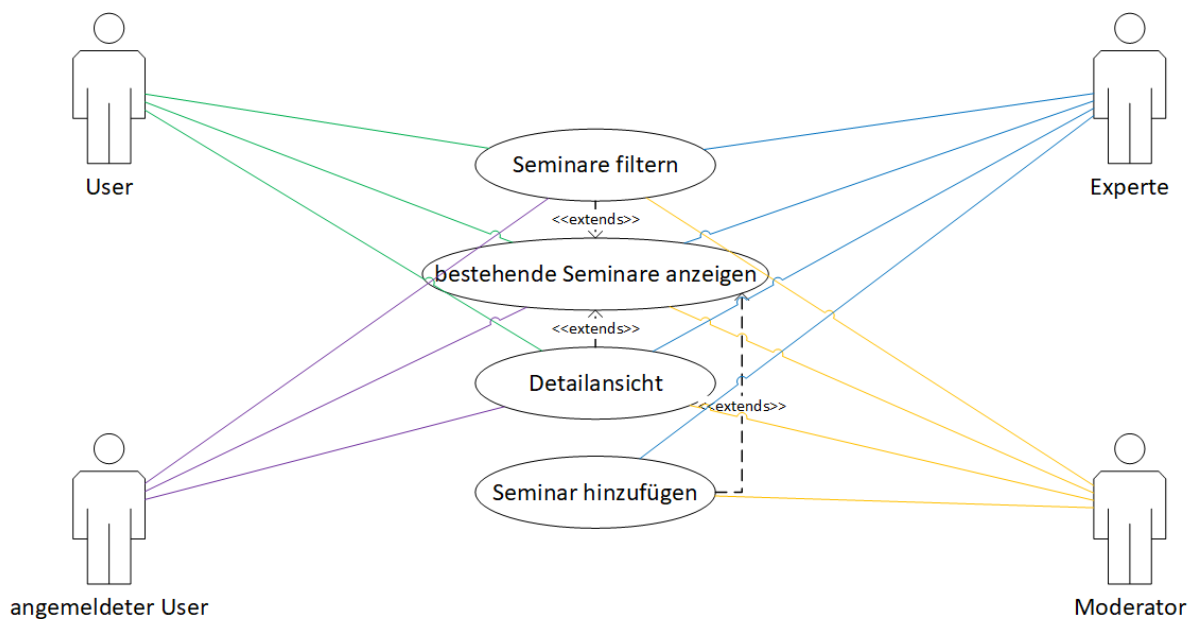


Abbildung 1: Seminar-Finder (Use-Case-Diagramm)

1. Bestehende Seminare anzeigen: Der Benutzer kann sich eine Liste bevorstehender Seminare ansehen. Die Standorte werden auf einer Karte angezeigt.
2. Seminar hinzufügen: Berechtigte Benutzer können ein neu angekündetes Seminar erfassen.
3. Detailansicht: Er kann sich detaillierte Informationen zu den Seminaren holen.
4. Seminare filtern: Die Ansicht zu den Seminaren kann nach gewissen Kriterien eingeschränkt werden.

Wiki: Use Cases

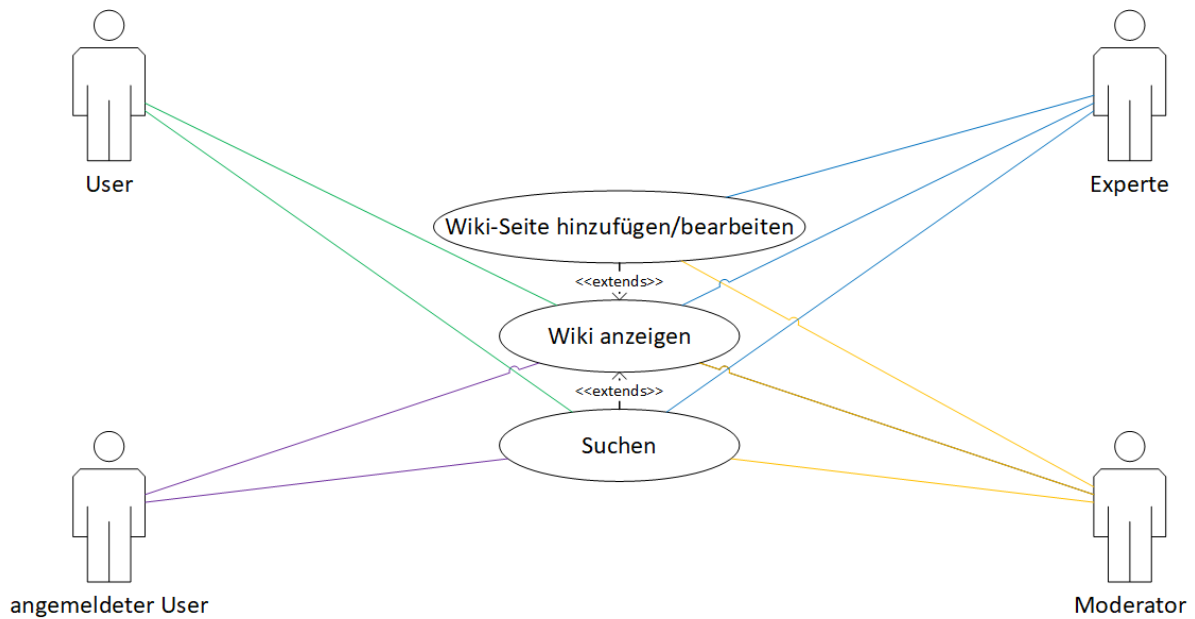


Abbildung 2: Wiki (Use-Case-Diagramm)

5. Wiki anzeigen: Im Wiki können Informationen zur sozialen Phobie gefunden werden.
6. Wiki-Seite hinzufügen / bearbeiten: Moderatoren und Experten können neue Informationen im Wiki hinzufügen und bei Neuerungen auch Bearbeiten.
7. Suchen: Die Artikel im Wiki sind durchsuchbar und können anhand des Titels eingegrenzt werden. Dadurch findet sich der Benutzer schneller zurecht.

Forum: Use Cases

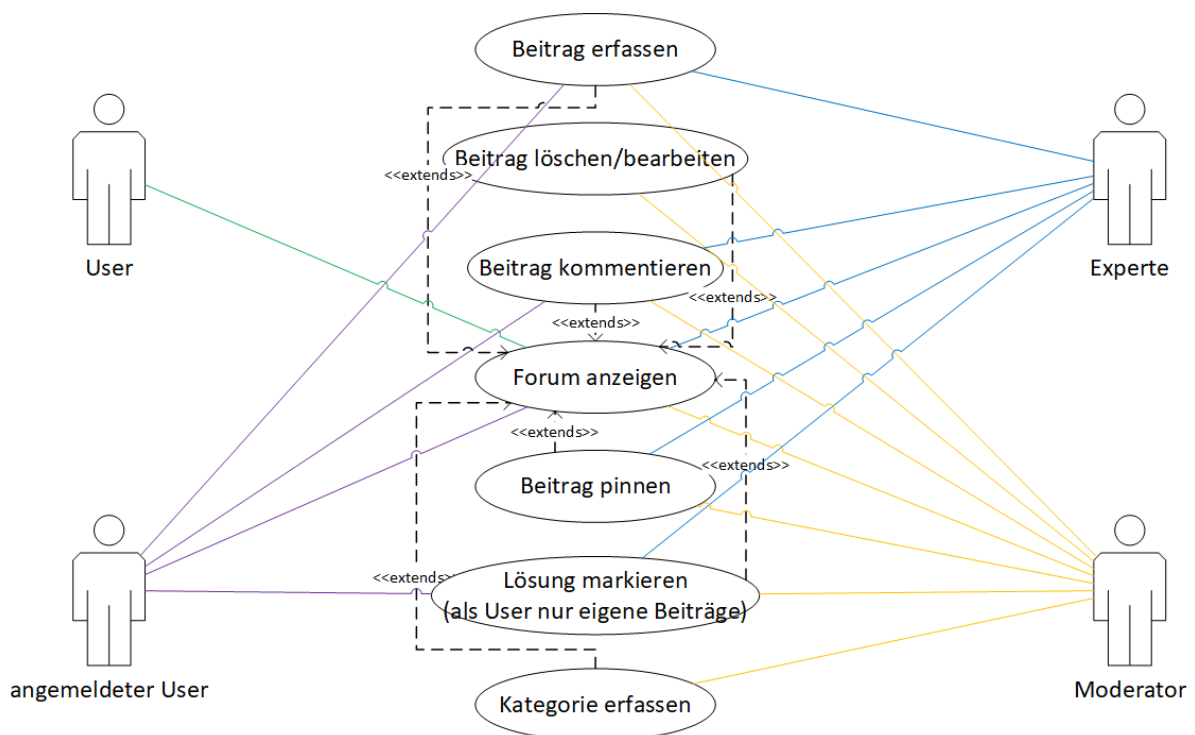


Abbildung 3: Forum (Use-Case-Diagramm)

8. Beitrag erfassen: Der Benutzer kann Fragen zur sozialen Phobie stellen oder Diskussionen zu gewissen Themen starten.
9. Beitrag löschen / bearbeiten: Moderatoren können in Ausnahmefällen aus rechtlichen oder anderen Gründen Beiträge anpassen.
10. Beitrag kommentieren: Jeder angemeldete Benutzer kann bei einer Frage beitragen.
11. Forum anzeigen: Alle Benutzer können die Kategorien im Forum ansehen, in eine Kategorie absteigen, die Beiträge der entsprechenden Kategorie anzeigen lassen und Beiträge, bzw. auch dazugehörige Kommentare lesen.
12. Beitrag pinnen: Moderatoren und Experten können bestimmte Einträge als besonders relevant an den obersten Rand einer Kategorie anheften.
13. Lösung markieren: Der Fragesteller kann aus einer Auswahl von Antworten die beste Antwort markieren.
14. Kategorie erfassen: Moderatoren können Kategorien erstellen.

Allgemein: Use Cases

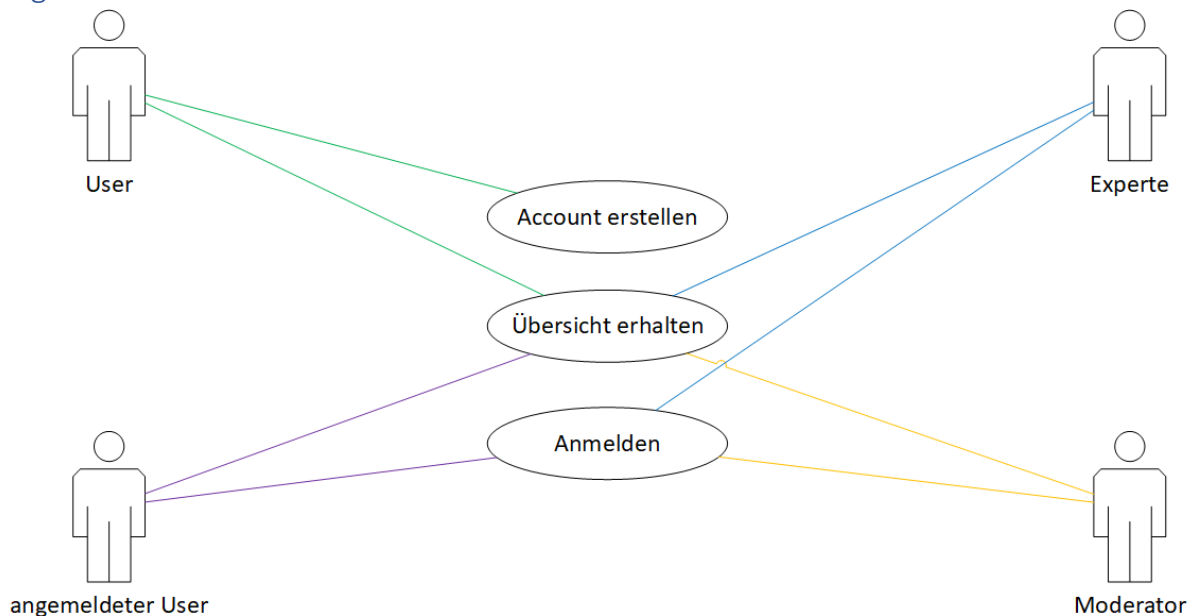


Abbildung 4: Allgemein (Use-Case-Diagramm)

15. Account erstellen: Benutzer können sich einen Account erstellen, der dann für den Zugriff auf das Forum oder das Erstellen von Seminaren genutzt wird.
16. Übersicht erhalten: Alle Benutzer haben eine Startseite zur Verfügung die ihnen einen Überblick über das Tool und seine Funktionen gibt sowie ein Menüband, dass den Wechsel zwischen den Funktionen ermöglicht.
17. Anmelden: Benutzer können sich auf ihren Account anmelden.
18. Die Benutzer haben Rollen (normaler User, Experte, Moderator) und es gibt auch unangemeldete Benutzer. Die Funktionen stehen nur den im Use Case Diagramm mit den Funktionen verknüpften Benutzergruppen zur Verfügung.

Nutzeranforderungen – Detaillierte Ausarbeitung

Gem. der Aufgabenstellung im Modul (Durchführungsrahmen des Projekts) sollte bloss eine beschränkte Zahl an Use Cases im Detail ausgearbeitet werden, weswegen sich hier nur ein Teil aller in den Use Case Diagrammen ersichtlichen Anwendungsfälle in ausgearbeiteter Form vorfinden. Es handelt sich dabei um Bestandteile des Seminar-Finders.

Use Case 1 – Bestehende Seminare anzeigen

Nr. und Name:	1 – Bestehende Seminare anzeigen
Szenario:	Seminare
Kurzbeschreibung:	Ein Benutzer kann sich die Seminarseite anzeigen
Beteiligt Akteure:	Nutzer der Plattform (öffentlich)
Auslöser / Vorbedingung:	Befindet sich auf der Seminarseite.
Ergebnisse / Nachbedingung:	Der Benutzer hat einen Überblick über die kommenden Veranstaltungen.

Ablauf:

Nr.	Wer	Was
1	System	Zeigt Seminare in Tabelle mit folgenden Details an: <ul style="list-style-type: none"> • Titel • Ort • Datum / Zeit • Kategorie Eine grafische Übersicht steht zur Verfügung.

Ausnahmen, Varianten:

Nr.	Wer	Was
1	System	Vergangene Seminare werden nicht mehr angezeigt.

Tabelle 2: Bestehende Seminare anzeigen (Use Case 1)

Use Case 2 – Seminar hinzufügen

Nr. und Name:	2 – Seminar hinzufügen
Szenario:	Seminare
Kurzbeschreibung:	Ein berechtigter Benutzer kann ein neues Seminar auf der Seminarseite eintragen.
Beteiligt Akteure:	<ul style="list-style-type: none"> • Experte / Arzt • Moderator
Auslöser / Vorbedingung:	<ul style="list-style-type: none"> • Eingeloggt • Seminar-Übersicht offen
Ergebnisse / Nachbedingung:	Ein neues Seminar wurde erstellt und ist für andere Benutzer ersichtlich.

Ablauf:

Nr.	Wer	Was
1	Benutzer	Knopf auf der Seminarseite für die Erstellung eines neuen Seminars wird gedrückt.
2	System	Seminarmaske wird angezeigt.
3	Benutzer	Füllt die benötigten Informationen ab: <ul style="list-style-type: none"> • Titel • Datum / Zeit • Kategorie • Kurze Beschreibung • Ort • Adresse • Externer Link
4	Benutzer	Speichert seine Eingaben.
5	System	Eingaben werden vom System validiert.
6.1	System	Wenn Eingaben nicht korrekt (z.B. ungültige Uhrzeit), wird eine Fehlermeldung angezeigt. Eingabemaske mit Inhalten wird beibehalten. (Zurück zu Schritt 3)
6.2	System	Wenn die Eingaben korrekt sind, werden die Eingaben in einem neuen Seminar gespeichert.
7	System	Die Anzeige über Seminare wird wieder geöffnet. (Use Case 1)

Ausnahmen, Varianten:

Nr.	Wer	Was
1	Benutzer mit zu wenig Rechten	Sollte den Button für ein neues Seminar nicht angezeigt bekommen.

Tabelle 3: Seminar hinzufügen (Use Case 2)

Use Case 3 – Detailansicht zu Seminar

Nr. und Name:	3 – Detailansicht zu Seminar
Szenario:	Seminare
Kurzbeschreibung:	Ein Benutzer kann auf der Seminarseite Details zu einem Seminar einsehen.
Beteiligt Akteure:	Anonymer Benutzer
Auslöser / Vorbedingung:	Seminar-Übersicht offen
Ergebnisse / Nachbedingung:	Dem Benutzer sind die Details bekannt.

Ablauf:

Nr.	Wer	Was
1	Benutzer	Wählt aus, mehr Details zu einem Seminar zu sehen.
2	System	Zeigt die ursprünglich eingegebenen Informationen an: <ul style="list-style-type: none"> • Titel • Datum / Zeit • Kategorie • Kurze Beschreibung • Ort • Adresse • Externer Link
3	Benutzer	Kann auf externen Link klicken und wird dabei auf die entsprechende Seite weitergeleitet.

Ausnahmen, Varianten:

Nr.	Wer	Was

Tabelle 4: Detailansicht zu Seminar (Use Case 3)

Use Case 4 – Seminare filtern

Nr. und Name:	4 – Seminare filtern
Szenario:	Seminare
Kurzbeschreibung:	Ein Benutzer kann auf der Seminarübersicht die angezeigten Seminare eingrenzen.
Beteiligt Akteure:	Anonymer Benutzer
Auslöser / Vorbedingung:	Seminar-Übersicht offen
Ergebnisse / Nachbedingung:	Die Übersicht ist eingeschränkt nach den Kriterien des Benutzers.

Ablauf:

Nr.	Wer	Was
1	System	Zeigt Filterungsmöglichkeiten auf der Seminarübersicht an.
2	Benutzer	Entscheidet sich für Filterungskriterien: <ul style="list-style-type: none"> • Kategorie • Datum (Von, Bis) • Ort
3	System	Zeigt die Seminarübersicht (Use Case 1) mit den Seminaren an, auf welche die Kriterien zutreffen.

Ausnahmen, Varianten:

Nr.	Wer	Was
1	Benutzer	Entsprechen dem gesetzten Filter keine Seminare, wird ein Info-Text angezeigt.
2	Benutzer	Vergangene Seminare können mit dem Filter nicht angezeigt werden. (Von-Datum ist mindestens „Heute“.)

Tabelle 5: Seminare filtern (Use Case 4)

Systemarchitektur

Die Applikation soll als Webapplikation umgesetzt werden. Wir unterscheiden zwischen Frontend (Client), Backend (Server) und Datenbank (Datenspeicherung).

Frontend

Die Applikation ist via Webbrowser erreichbar. Sowohl PCs wie auch Smartphones werden unterstützt. Damit Darstellung und Bedienung auf Smartphones benutzerfreundlich sind, wird ein Responsive Design implementiert. Es wird ein aktueller und moderner Webbrowser vorausgesetzt, damit das Responsive Design richtig dargestellt wird.

Backend

Die Applikation wird von der Regionalen Gesundheitsbehörde in ihrem Datacenter oder in der Cloud gehostet und betrieben. Gemäss ihren Anforderungen muss sie redundant gehostet werden können.

Die Hauptfunktionen (siehe [Kapitel Einführung](#)) sollen möglichst unabhängig voneinander sein, damit sie getrennt entwickelt und gewartet werden können und um die Komplexität zu reduzieren.

Als Technologiestack wird Java 11 und das Webframework [Vaadin](#) eingesetzt.

Datenbank

Alle Applikationsdaten werden in einer Datenbank gespeichert, das Backend soll nur die Programmlogik abbilden.

Systemanforderungen

Die hier beschriebenen Anforderungen beziehen sich auf die vorher beschriebenen Use Cases. Dabei ist zu beachten, dass aufgrund der speziellen Aufgabenstellung – nur ein Teil der Use Cases ist vollständig ausdefiniert – auch nur diese Anforderungen komplett formuliert werden konnten. Mögliche Lücken bei anderen Modulen der Applikation sind damit zu erklären.

Funktionale Anforderungen

Nr.	Name	Beschreibung	Use Case-Nr.
FA1	Login	Der Benutzer kann sich über eine Login-Maske in die Applikation einloggen (Username, Passwort)	17
FA2	Account erstellen	Ein Benutzer kann sich selbstständig einen neuen Account anlegen.	15
FA3	Startseite	Beim Aufrufen der Applikation erhält der Benutzer eine Übersicht über die Funktionalitäten.	16
FA4	Menu	Die Applikation ist einfach zu navigieren, der Benutzer kann von jeder Hauptfunktion zu einer anderen navigieren	16
FA5	Seminare anzeigen (Tabelle)	Die aktuellen Seminare werden in einer Tabelle dargestellt. Dabei sind die folgenden Informationen auf einen Blick ersichtlich: Titel, Ort, Datum, Uhrzeit, Kategorie.	1
FA6	Seminare anzeigen (Karte)	Die Seminare werden gemäss ihrer Adresse auf einer Karte dargestellt.	1
FA7	Seminare filtern	Die Seminare können nach Kategorie, Datum (von, bis) und Ort gefiltert werden. Die Anzeige der Seminare wird dadurch verändert, sowohl auf der Karte als auch die Tabelle.	4
FA8	Seminar hinzufügen	Einem Experten oder Moderator steht die Möglichkeit offen, ein neues Seminar hinzuzufügen. Dabei muss er Titel, Datum, Uhrzeit, Kategorie, Ort, Adresse, Externer Link sowie optional eine kurze Beschreibung angeben.	2
FA9	Seminar Detailansicht	Durch Klicken auf ein Seminar (in der Tabelle oder auf der Karte) werden die Details zu diesem angezeigt.	3
FA10	Externer Link	Der Externe Link in einem Seminar ist klickbar und ruft die gewünschte Webseite auf.	3
FA11	Wiki	Das Wiki besteht aus verschiedenen Unterseiten zu verschiedenen Themen	5
FA12	Wiki-Menu	Die verschiedenen Themen werden in einer Baum-Struktur angezeigt	5
FA13	Wiki Eintrag erfassen	Experten oder Moderatoren können einen neuen Eintrag erfassen. Dabei wählen sie die Kategorie aus, setzen einen Titel und stellen den Text zur Verfügung.	6
FA14	Wiki Eintrag bearbeiten	Experten und Moderatoren können bestehende Einträge bearbeiten.	6
FA15	Wiki durchsuchen	Die Benutzer haben die Möglichkeit, das Wiki nach den Überschriften zur durchsuchen.	7
FA16	Forum	Das Forum besteht aus verschiedenen Kategorien, welche die verschiedenen Beiträge enthalten. Die Beiträge werden danach sortiert, dass die zuletzt kommentierten Beiträge zuoberst sind.	11
FA17	Forum Beitrag	Hier wird der ursprüngliche Beitrag angezeigt, sowie die Kommentare dazu.	11

FA18	Beitrag erfassen	Der Benutzer kann in einer Kategorie einen Beitrag erfassen (Titel, Inhalt)	8
FA19	Beiträge bearbeiten	Die Moderatoren können bestehende Beiträge bearbeiten oder löschen.	9
FA20	Kommentar schreiben	Die Benutzer können zu einem bestehenden Beitrag einen Kommentar abgeben.	10
FA21	Beitrag pinnen	Die Experten oder Moderatoren können Beiträge pinnen, sodass diese zuoberst in der Kategorie dargestellt werden.	12
FA22	Lösung markieren	Der Autor des Beitrages kann eine Lösung markieren. Diese wird dann sowohl farblich als auch von der Position hervorgehoben.	13
FA23	Kategorie erstellen	Ein Moderator kann eine neue Kategorie erstellen, oder eine bestehende umbenennen. Wenn eine Kategorie keine Beiträge enthält kann er diese löschen.	14
FA24	Webapplikation	Die Applikation ist über moderne und aktuelle Webbrowser zugänglich.	Frontend
FA25	Responsive Design	Ein Responsive Design ist implementiert, damit die Applikation auch auf mobilen Geräten benutzerfreundlich bedient werden kann.	Frontend
FA26	Redundanz	Die Applikation muss redundant betrieben werden können	Backend
FA27	Speicherung Applikationsdaten	Die Applikationsdaten sind in einer Datenbank abgespeichert.	Datenbank

Tabelle 6: Funktionale Anforderungen

Nicht-funktionale Anforderungen

Nr.	Name	Beschreibung	Use Case-Nr.
NFA1	Benutzerfreundlichkeit	Die Handhabung der Applikation sollte möglichst Benutzerfreundlich sein, also mit möglichst wenig Interaktion des Benutzers stattfinden.	allgemein
NFA2	Berechtigungskonzept	Die Benutzer sehen nur die Teile der Applikation, zu welchen sie die Berechtigungen haben.	allgemein
NFA3	Input-Validierung	Die Eingaben der verschiedenen Benutzer werden validiert, sodass keine Inkonsistenzen auftreten	2
NFA4	Bestätigung	Beim Speichern, Bearbeiten oder Löschen von Inhalten soll eine Bestätigungsmeldung angezeigt werden, welche der Benutzer bestätigen muss	allgemein
NFA5	Benachrichtigung	Im Falle eines Fehlers (Validierung oder Server-Fehler) erhält der Benutzer eine entsprechende Meldung.	allgemein
NFA6	Zugänglichkeit	Die Applikation ist von überall aus zugänglich	allgemein
NFA7	Geschwindigkeit	Die Applikation soll flüssig bedienbar sein	allgemein
NFA8	Responsive	Die Applikation lässt sich auf jedem Gerät ansehen	allgemein
NFA9	Datenschutz	Die Daten der Benutzer sind nicht einsehbar	allgemein
NFA10	Auslieferung als Quellcode	Der Quellcode der Applikation ist ein Lieferobjekt.	Backend

Tabelle 7: Nicht-Funktionale Anforderungen

System Modell

Die Webapplikation ist in drei Module aufgeteilt, die die Hauptfunktionen widerspiegeln:

- Wiki
- Forum
- Seminar-Finder

Übersicht der Web-Applikation:

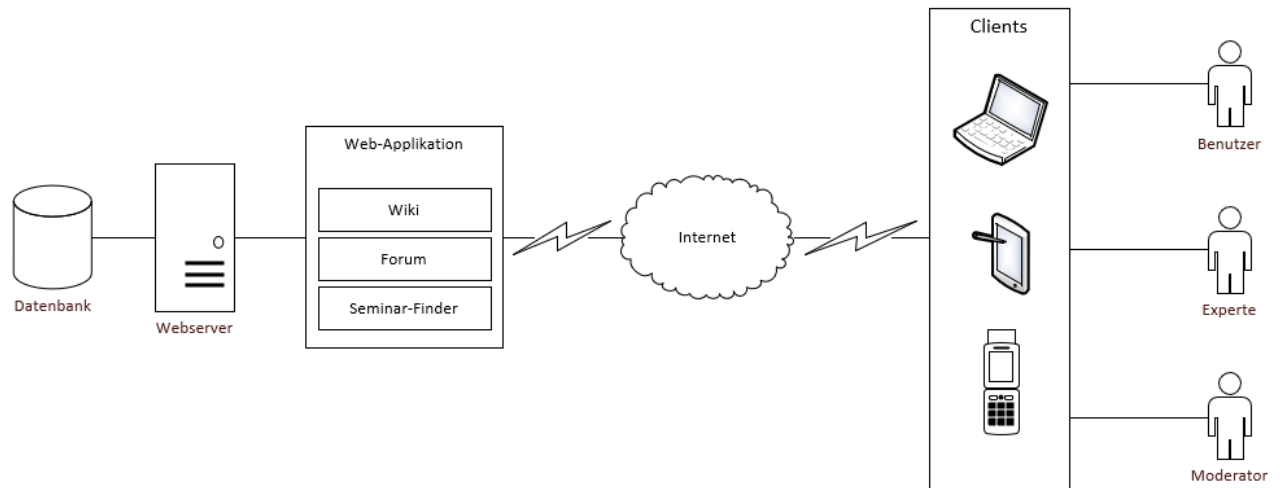


Abbildung 5: Model

Weiterentwicklung des Systems

Perspektiven

Es gibt diverse Richtungen, in die sich die Applikation aufgrund erster Kundenfeedbacks und der Aufnahme auf dem Markt weiterentwickeln könnte. Es ist wichtig, vor einem nächsten Projekt eine erneute Evaluation der Bedürfnisse durchzuführen.

Die verschiedenen zusätzlichen Wünsche sind unter den im Projekt aufgeführten Komponenten gegliedert. Zusätzliche Bedürfnisse ausserhalb dieser Komponenten mögen zwar vorhanden sein, doch haben diese auf die Entscheidungen des Projekts keinen Einfluss. Eine Erweiterbarkeit sollte aufgrund des gewählten Aufbaus grundsätzlich gegeben sein, müsste jedoch in einem späteren Projekt aufgegriffen werden.

Seminar-Finder

Die Seminar-Funktion könnte um die Funktionalität erweitert werden, andere Benutzer aktiv an ein Seminar einzuladen. Dabei kann es sich um eines handeln, dass man als Experte selber erstellt hat oder auch als einfacher Benutzer z.B. zusammen mit der Familie besuchen möchte.

Weiter könnte das Bedürfnis nach einer Benachrichtigungs-Funktion bestehen, um über neue Seminare an einem bestimmten Ort und/oder Kategorie informiert zu werden und auf dem Laufenden zu bleiben.

Neben der heute bestehenden Kurzbeschreibung inkl. Link könnte allenfalls auch das Bedürfnis bestehen, Inhalte direkt auf der Seminarseite einzupflegen.

Bei der Entwicklung zu berücksichtigen:

Solange das Seminar ein eigenständiges Element des Datenmodells bleibt, sollten diese Funktionen umsetzbar sein. Eine Ergänzung des Objekts um eigene Inhalte wäre die grösste Anpassung, wobei hier «nur» ein Handling leerer Felder berücksichtigt werden muss, wobei die Komponenten für Inhalte aus dem Forum bezogen werden können.

Wiki

Für Experten könnte ein Diskussions-Feature zu einzelnen Beiträgen interessant sein, um Unklarheiten zur Editierung zu klären. So ist es möglich, Streitigkeiten unkompliziert im Tool zu klären.

Auch der Wunsch nach einer ergänzten Suchfunktion könnte aufkommen, was vor allem den Einbezug von Inhalten betreffen könnte. Zurzeit wird bloss die Suche nach Titeln umgesetzt.

Bei der Entwicklung zu berücksichtigen:

Die Diskussions-Funktion könnte mit Modulen aus dem Forum implementiert werden. So wäre eine Diskussion als Forums-Beitrag definiert mit entsprechenden Kommentaren. Speziell wären die eingeschränkten Berechtigungen für allgemeine Nutzer. Die Such-Funktion muss im Daten- oder Applikationsmodell nicht berücksichtigt werden.

Forum

Benutzer könnten bei der Verwendung das Bedürfnis verspüren, über neue Einträge unter ihren Fragen informiert zu werden. Dies würde die Einrichtung einer Benachrichtigungsfunktion notwendig machen.

Zudem wäre es für Benutzer vielleicht interessant, nachschauen zu können, welche Einträge sie bereits verfasst haben. Dies könnte z.B. mit einer Übersicht auf der eigenen Profilseite geschehen.

Bei der Entwicklung zu berücksichtigen:

Die Benachrichtigungsfunktion würde je nach Entwicklung entweder über App oder Mail eingerichtet werden. Dies muss nicht im Voraus behandelt werden, da bei Interesse an einer Benachrichtigung nachträglich nach einer Mail-Adresse gefragt werden kann. Beim Erstellen eines Eintrags ist der Autor zwingend zu speichern, so dass später für das System nachvollziehbar ist, welche Einträge zum Benutzer gehören.

Allgemein

Benutzer könnten sich eine Funktion zum Zurücksetzen ihrer Passwörter via Mail wünschen.

Auch könnte eine individuelle Sicht auf die Startseite gewünscht werden, wo persönlich bevorzugte Themen angezeigt werden.

Bei der Entwicklung zu berücksichtigen:

Es könnte Sinn machen bei der Registrierung die E-Mail-Adresse der Benutzer bereits zu speichern, damit auch bereits registrierte Benutzer ohne grosse Umstände nach Einführung einer solchen Funktion ihr Passwort zurücksetzen können. Eine persönliche Sicht auf der Startseite muss nicht berücksichtigt werden, da es sich nur um eine andere Darstellung bereits bestehender Daten handelt.

Änderungen am Technologiestack

Das als Basis für die Applikation verwendete Framework Vaadin wie auch die Programmiersprache Java und die entsprechende Laufzeitumgebung sind stetigen Veränderungen durch Weiterentwicklung unterworfen. Nicht alle Updates müssen zwingend sofort eingespielt werden, doch macht es Sinn, hier möglichst vorausschauende Entscheidungen zu treffen. Folgende Massnahmen werden getroffen:

- Es werden keine Java-Klassen verwendet, welche veraltet (deprecated) sind.
- Bei der Produktivsetzung werden aktuell unterstützte Softwareversionen verwendet.
- Bei längerem Betrieb wird bei Wartungen eine Aktualisierung der Komponenten vorgenommen.

Allenfalls könnte im Laufe der Zeit auch das Bedürfnis nach einer Mobile-App aufkommen. Eine mögliche Realisierung wäre eine native Web-App, was mit der derzeitigen technischen Umsetzung einfach zu realisieren ist.

Testing

Damit die Applikation schlussendlich auch so funktioniert, wie sie es soll, wird sie getestet. Dies wird einerseits bereits während der Entwicklung der Software mit sogenannten Unit Tests sichergestellt. Hier können erste funktionale Fehler bereits gefunden und eliminiert werden. Diese Unit Tests werden in unregelmässigen Abständen wiederholt, damit sichergestellt werden kann, dass durch das Hinzufügen von neuen Features nicht irgendwelche alten Dinge verändert oder zerstört wurden.

Dazu gehört auch, dass eventuelle Schnittstellen zwischen dem Frontend und dem Backend getestet werden.

Wenn der Code in einem Zustand ist, in den man ihn releasen könnte, wird die ganze Applikation via Black-Box-Testing getestet. Hierbei wird für jede Anforderung einen Testfall geschrieben, wo die auszuführende Tätigkeit und das erwartete Ergebnis dokumentiert werden. Mit diesem Testfall muss nun eine Person, die den Code nicht kennt, die auszuführende Tätigkeit ausführen und notieren, ob das erwartete Ergebnis eingetreten ist. Der Test gilt als bestanden, wenn das erwartete Ergebnis eingetroffen ist. Aus Ausgangslage für das Black-Box-Testing dienen die Anforderungen, welche weiter oben definiert wurden.

Beispiel für FA1:

Testfall	T01 – erfolgreiches Login
Betrifft Anforderung	FA1 - Login
Beschreibung	Der Benutzer kann sich über eine Login-Maske in die Applikation einloggen (Username, Passwort)
Testvoraussetzung	Applikation wurde in einem den Anforderungen gerechten Browser gestartet
Testschritte	Existierender Username und Passwort eingeben, auf "Login" klicken
Erwartetes Ergebnis	Der User wird eingeloggt und wird auf die Startseite weitergeleitet

Anhang

Abbildungsverzeichnis

Abbildung 1: Seminar-Finder (Use-Case-Diagramm)	5
Abbildung 2: Wiki (Use-Case-Diagramm)	6
Abbildung 3: Forum (Use-Case-Diagramm)	6
Abbildung 4: Allgemein (Use-Case-Diagramm)	7
Abbildung 5: Model	15

Tabellenverzeichnis

Tabelle 1: Glossar	5
Tabelle 2: Bestehende Seminare anzeigen (Use Case 1)	8
Tabelle 3: Seminar hinzufügen (Use Case 2)	9
Tabelle 4: Detailansicht zu Seminar (Use Case 3)	10
Tabelle 5: Seminare filtern (Use Case 4)	11
Tabelle 6: Funktionale Anforderungen	14
Tabelle 7: Nicht-Funktionale Anforderungen	14