



Projektauftrag (PA) Care Connect	2
Projektbezeichnung und Entwicklerteam	2
Ausgangssituation	2
Projektbeschreibung	2
Zielgruppen	2
Funktionale Anforderungen, Anwendungsfälle	3
Anwendungsfall Übersicht	5
Iceberglis	6
Domänenmodell	9
Arbeitsstruktur & Grober Projektplan	10
Rollenverteilung	10
Horizontale Verantwortlichkeiten	10
Grober Projektplan	14
Work Breakdown Structure (WBS)	15
Meilensteinbeschreibung	17
Projektabgrenzungen	17
Schichtendiagramm	18
Lieferkomponenten	19
Software - mit setup-tool	19
Artefakte & Projektdokumentation	19
Abgrenzung des Lieferumfangs	19
Nichtfunktionale Anforderungen	19
Informationswesen	23
Besonderheiten	23

Projektauftrag (PA) Care Connect

Projektbezeichnung und Entwicklerteam

Id	Matrikelnummer	Name	Rolle	Stellvertretende Rolle
canger	12120520	Anger, Chris	Teamleiter Daten Ingenieur	
sschuch	12222156	Schuch, Sebastian	Build/Release Manager Sicherheits Ingenieur	
rfoster	12222173	Foster, Ryan	Usability Ingenieur Technischer Architekt	Kommunikations leiter
jschort	12225164	Schort, Jonathan	Dokumentationsbeauftragter Usability Ingenieur	Testleiter
pnirnberger	12034156	Nirnberger, Philipp	Kommunikationsleiter Wiki-Beauftragter	Teamleiter
noguamalam	12225111	Oguamalam, Noah	Testleiter Requirement Ingenieur	

Ausgangssituation

In Krankenhäusern wird oft veraltete Software verwendet. Diese Software kann entweder nicht alle benötigten Funktionen haben oder schwer zu bedienen sein. Manchmal müssen auch mehrere verschiedene Systeme verwendet werden. Wenn diese Systeme nicht mehr gewartet werden, können Kompatibilitätsprobleme oder andere Fehler auftreten, die sich auf andere Systeme ausbreiten können.

Projektbeschreibung

Ein umfassendes Krankenhaus-Managementsystem (KHMS) ist erforderlich, das eine breite Palette von Anwendungsfällen abdeckt und gleichzeitig benutzerfreundlich und übersichtlich gestaltet ist. Durch eine intuitive Benutzeroberfläche soll das System den Arbeitsalltag des Krankenhauspersonals erleichtern und gleichzeitig die Patientenzufriedenheit steigern. Für das Personal ermöglicht das KHMS eine effiziente Verwaltung der Patientendaten sowie eine Optimierung des Aufnahmeprozesses. Patienten profitieren von einer vereinfachten Terminvereinbarung und einem einfachen Zugang zu ihren Gesundheitsdaten, was insgesamt zu einer verbesserten Versorgung und einem reibungsloseren Ablauf im Krankenhaus führt.

Zielgruppen

- System-Administratoren können Ärzte auf verschiedene Ambulatorien zuweisen. Zusätzlich können Öffnungszeiten von Ambulatorien gesetzt und generelle medizinische Information eingefügt werden.

- Sekretäre können für Patienten deren Profile, Termine und Aufenthalte erstellen und (nur die Patientendaten selbst und Aufenthalte) bearbeiten. Lösch-Berechtigung haben Sekretäre hierbei nur auf Termine, auf welche sie, wie auch Patienten-Aufenthalte, auch Einblick haben.
- Ärzte nutzen ein spezialisiertes Werkzeug, um umfassende Informationen über ihre Patienten zu verwalten. Dies umfasst nicht nur grundlegende Daten wie Untersuchungen, Diagnosen, Behandlungen und verordnete Medikationen, sondern bietet auch die Möglichkeit zur Kommunikation mit den Patienten.
- Patienten können verfügbare Termine annehmen und ihre eigenen Daten verwalten bzw auslesen. Zu den Daten zählen Behandlungen, Termine und verschriebene Medizin und Dauer des Krankenhausaufenthalts.

Rollen im System:

	Rechte			
Rolle	erstellen	editieren	löschen	hat Einblick auf
System-Administrator	System-Administratoren, Sekretäre, Ärzte, Medikamente, Stationen, Ambulanzen	System-Administratoren, Sekretäre, Ärzte, Medikamente, Stationen, Ambulanzen	System-Administratoren, Sekretäre, Ärzte, Medikamente, Stationen, Ambulanzen	alles exklusive Patientenbezogene daten
Sekretär	Patient, Termin, Aufenthalte	Patientendaten, Aufenthalte	Termine	Termine, Aufenthalt
Arzt	Behandlungen, Chat	Behandlungen	Behandlungen	Behandlungen, Patient, Aufenthalt
Patient	Termine, Chat	Termine, Patient	Termine	Termine, Patient, Behandlungen, Aufenthalt

Funktionale Anforderungen, Anwendungsfälle

Die Kunden-Priorität ist spezifiziert als "N" für niedrig, "M" für mittel und "H" für hoch.

Id	Feature	Beschreibung	Kunden-Priorität	Aufwand
1	Digitales Patientenprofil	Es soll möglich sein, Patientenprofile anzulegen und zu bearbeiten.	H	2
2	Digitales Ärzteprofil	Es soll möglich sein, Ärzte-Profile anzulegen, zu bearbeiten und zu löschen.	H	3

3	Digitales Sekretär profil	Es soll möglich sein Sekretärprofile anzulegen, zu bearbeiten und zu löschen.	H	3
4	Anmelde system für Benutzer	Nutzer (Patienten / Ärzte) können sich im System mit E-Mail und Passwort einloggen und die für Sie freigegebenen Aktionen durchführen.	H	5
5	Digitale Terminv ereinbar ung	Es soll möglich sein, über das Programm einen ambulanten Termin zu buchen.	M	5
6	Telemedi zin mit Arzt	Bei Beschwerden kann ein Patient mithilfe des Programmes mit seinem Arzt kommunizieren, um eine Ferndiagnose zu bekommen.	M	8
7.1	Patiente nhistorie auslesen	Patienten können ihre eigene Behandlung aus dem Krankenhaus abrufen.	M	2
7.2	Suche nach Behandl ungen/P atienten daten	Mit verschiedenen Kriterien wird es möglich sein, nach gewissen Behandlungen zu filtern.	M	6
8	Behandl ungen dokume ntieren	Ein Arzt kann für Patienten Behandlungsdaten anlegen.	H	4
9	Digitale Statione n/Ambulanzen	Sind digitale Abbildungen von den Stationen/Ambulanzen des Krankenhaus, welche verwendet werden, um Behandlungsorte und stationierte Ärzte zu definieren.	H	3
10	Auslastu ng von Ambulanzen	Es ist möglich, die derzeitigen Auslastungen von Ambulanzen anzuzeigen.	M	2
11	Schnittst elle zu anderen Systeme n	Es ist möglich, die Daten der Software zu exportieren, um diese in anderen Programmen verwenden zu können.	M	5
12	Administ ratoren	Administratoren können Stationen, Ambulanzen und Administratoren selbst bearbeiten, und neue Ärzte, Stationen, Ambulanzen, Medikamente, Allergien und Administratoren hinzufügen (und löschen).	H	4

13	Setup	Ein initiales Setup beim ersten Start der Anwendung, welche es einem erlaubt Administratoren, Stationen, Ambulanzen, Medikamente und Allergien anzulegen.	N	3
----	-------	---	---	---

Anwendungsfall Übersicht

Die Features aus dem Projektvorschlag (PV) wurden zu folgenden Anwendungsfall-Paketen zusammengefasst. Aufgrund der Anforderungsvielfalt dieses Projektauftrags ist eine Priorisierung (**Hoch**, **Mittel**, **Niedrig**) der einzelnen Pakete zweckmäßig.

1. Digitales Patientenprofil (H): Unbedingt erforderlich, kritisch
2. Digitales Ärzteprofil (H): Unbedingt erforderlich, kritisch
3. Digitales Sekretärprofil (H): Unbedingt erforderlich, kritisch
4. Anmelde-system (H): Unbedingt erforderlich, kritisch
5. Digitale Terminvereinbarung (M): nur Grundfunktionalität notwendig (erstellen/beitreten/abmelden)
6. Telemedizin (M): nur Grundfunktionalität notwendig
7. Patientendaten auslesen (M): nur Grundfunktionalität notwendig
8. Behandlungen dokumentieren (H): Unbedingt erforderlich, kritisch
9. Digitale Stationen/Ambulanzen (H): Unbedingt erforderlich, kritisch
10. Auslastung von Ambulanzen (M): nur Grundfunktionalität notwendig
11. Schnittstelle (M): nur Grundfunktionalität notwendig
12. Administratoren (H): Unbedingt erforderlich, kritisch
13. Setup (N): Erweiternde Features, können auch nach dem Projekt implementiert werden

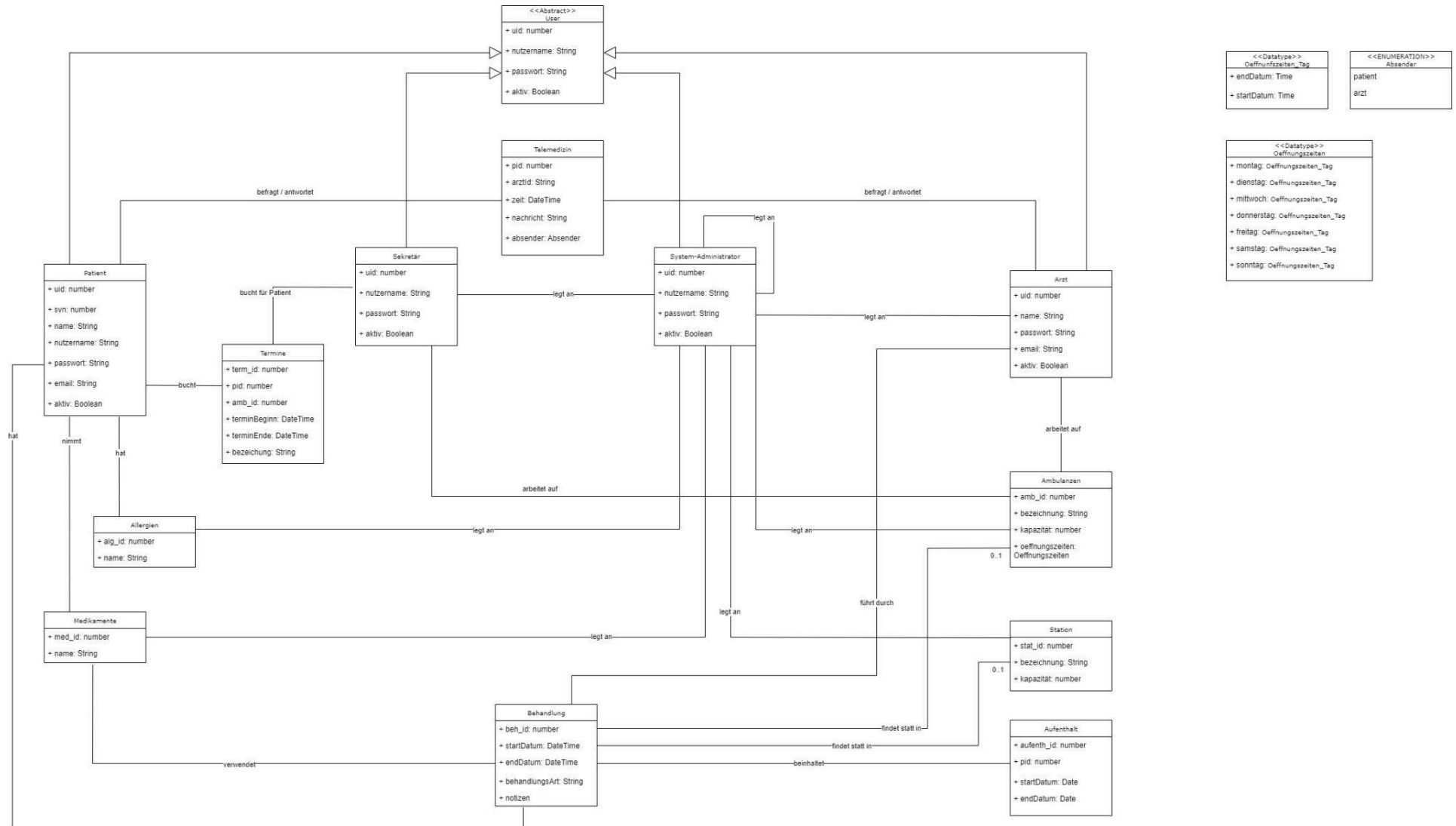
Iceberglist

Id	Feature, Akteur	Anwendungsfälle	Kundenpriorität	Aufwand	Version	Zuständiger
1.1	Digitales Patientenprofil, Sekretär	Patient Account anlegen	H	3	1	rfoster
2.1	Digitales Ärzteprofil, System-Administrator	Arzt Account anlegen	H	3	1	rfoster
3.1	Digitales Sekretärprofil, System-Administrator	Sekretär Account anlegen	H	3	1	noguamalam
9.1	Digitale Stationen/Ambulanzen, System-Administrator	Ambulanz anlegen	H	4	1	sschuch
9.2	Digitale Stationen/Ambulanzen, System-Administrator	Station anlegen	H	4	1	pnirnberger
12.1	Administratoren, System-Administrator	Medikament anlegen	H	2	1	noguamalam
12.2	Administratoren, System-Administrator	Allergie anlegen	H	2	1	pnirnberger
8.1	Behandlungen dokumentieren, Arzt	Behandlung anlegen	H	5	1	jschort
8.2	Behandlungen dokumentieren, Arzt	Behandlung bearbeiten	H	5	1	jschort
4.1	Anmeldesystem, User	Anmeldung an dem System	H	5	1	canger
5.1	Terminvereinbarung, Patient	Termin buchen	M	5	2	rfoster
5.2	Terminvereinbarung, Patient	Termin absagen/löschen	M	4	2	jschort
9.3	Digitale Stationen/Ambulanzen, Arzt/Sekretär	Ambulanz Termine Anzeigen	H	4	2	pnirnberger

5.3	Terminvereinbarung, Sekretär	Termin für Patient buchen	M	5	2	noguamalam
5.4	Terminvereinbarung, Sekretär	Termin für Patient absagen/löschen	M	3	2	jschort
6.1	Telemedizin, Patient	Kommunikation mit Arzt	M	8	2	sschuch
6.2	Telemedizin, Arzt	Kommunikation mit Patient	M	8	2	canger
8.3	Behandlungen dokumentieren, Sekretär	Aufenthalte dokumentieren	H	3	3	sschuch
1.2	Digitales Patientenprofil, Patient/Sekretär	eigene Patientendaten bearbeiten	H	2	3	pnirnberger
2.1	Digitales Ärzteprofil, Arzt/System-Administrator	Ärzte-Daten bearbeiten	H	2	3	rfoster
3.2	Digitales Sekretärprofil, Sekretär/System-Administrator	Sekretär-Daten bearbeiten	H	2	3	noguamalam
9.3	Digitale Stationen/Ambulanzen, System-Administrator	Stations-Daten bearbeiten	H	2	3	canger
9.4	Digitale Stationen/Ambulanzen, System-Administrator	Ambulanz-Daten bearbeiten (auch inaktiv setzen)	H	2	3	canger
12.3	Administratoren, System-Administrator	Administrations-Daten bearbeiten: - user auf inaktiv setzen (kann nicht einloggen) - anlegen neuer administratoren	N	2	3	jschort
8.4	Behandlungen dokumentieren, Sekretär	Aufenthalte bearbeiten	H	3	3	sschuch

13.1	Registrierung, System-Administrator	Systemadministrator Account anlegen	H	2	4	noguamalam
7.1	Patientendaten auslesen, Patient/Arzt	Patientenhistorie auslesen	M	2	4	sschuch
7.2	Patientendaten auslesen, Patient/Arzt	Suche nach Behandlungen/Patientendaten	M	6	4	rfoster
10.1	Auslastung von Ambulanzen, Patient/Sekretär	Auslastung von Ambulanzen anzeigen	N	2	4	jschort
11	Schnittstelle, System-Administrator	Schnittstelle zu anderen Systemen	M	8	4	canger & pnirberger

Domänenmodell



Arbeitsstruktur & Grober Projektplan

Das Projektteam besteht aus 6 Entwicklern, die Expertenrollen (und damit verbundene Horizontale Verantwortlichkeiten) wurden bereits im Projektvorschlag definiert. Es wurde entschieden, SCRUM zu verwenden.

Rollenverteilung

Matrikelnummer	Name	Rolle	Stellvertretende Rolle
12120520	Anger, Chris	Teamleiter Daten Ingenieur	
12222156	Schuch, Sebastian	Build/Release Manager Security Ingenieur	
12222173	Foster, Ryan	Usability Ingenieur Technischer Architekt	Kommunikationsleiter
12225164	Schort, Jonathan	Dokumentationsbeauftragte r Usability Ingenieur	Testleiter
12034156	Nirnberger, Philipp	Kommunikationsleiter, Wiki-Beauftragter	Teamleiter
12225111	Oguamalam, Noah	Testleiter Requirement Ingenieur	

Horizontale Verantwortlichkeiten

Technischer Architekt: rfoster

Technische Architekten kümmern sich um die technische Infrastruktur eines Projekts, wie die Ordnerstruktur und die abhängigen JAR-Bibliotheken der Software.

Die spezifischen Verantwortlichkeiten eines technischen Architekten in diesem Projekt umfassen:

- Das Projekt Objekt Model (Maven pom.xml) verwalten.
- Das Management von Abhängigkeiten, insbesondere Versionen.
- Vertraut mit der Entwicklungsumgebung IntelliJ Idea und Maven haben.
- Die Erstellung von Codingguidelines (Checkstyle).
- Die Verwaltung aller Verbesserungstickets im Tracker.

Teamleiter: canger

Als Teamleiter ist es wichtig, gute Kenntnisse im Projektmanagement zu haben. Der Teamleiter ist dafür verantwortlich, dass alle Projektmanagement-Dokumente wie der Projektplan aktuell sind. Dazu gehört die laufende Dokumentation, das Erstellen von Arbeitsaufteilungen und das Verfolgen des Fortschritts.

Zusammengefasst sind die Hauptaufgaben:

- Projektmanagement-Dokumente aktuell halten
- Fortschritt verfolgen und kontrollieren
- Aufgaben verteilen und überwachen
- Mit Auftraggebern kommunizieren
- Meetings organisieren

Dokumentationsbeauftragter: jschort, Usability Engineer

- Verfügbarkeit der Dokumentation
 - JavaDocs in Klassen und Interfaces
- Sicherstellen, dass alle Klassen, Variablen und Methoden in English dokumentiert werden
- Erstellung von Dokumentationsrichtlinien (Format- und Formatierungsrichtlinien, Spezifikation der Code Conventions, ...)
- Überprüfung der Einhaltung von Dokumentationsrichtlinien
- Überprüfung der Vollständigkeit von Dokumenten

Testleiter: noguamalam

Die Testinfrastruktur umfasst verschiedene Komponenten, die für das Testen der Software entscheidend sind. Dazu gehören Testbibliotheken, Testdaten (zweite Datenbank), und Test-Suiten, die mit dem Spring Framework integriert sind, um Testdaten zu verwalten.

Es ist wichtig sicherzustellen, dass der Testcode klar vom Produktionscode getrennt ist. Dazu gehören die Erstellung eines Testplans und die Festlegung der Testvorgehensweise, einschließlich der Planung von Testläufen.

Im Falle von Fehlern ist es wichtig, die Auswirkungen zu verstehen, Fehlermeldungen angemessen weiterzugeben und entsprechend auf Fehler zu reagieren (Exception Handling).

Zu den Aufgaben gehören:

- Verwaltung aller Problemtickets im Tracker.
- Überwachung von Integrations- und Systemtests in enger Zusammenarbeit mit den technischen Architekten.
- Regelmäßige Überprüfung aller Unittests, um die Qualität der Software sicherzustellen.

Build/Release Manager: sschuch

Development Environment Infrastruktur:

- Scripts zum initialisieren, testen oder verteilen (deploy)
- Build, Deploy Targets von Maven; "Projekt kann jederzeit ausgecheckt und kompiliert werden"
- Gitlab Pipeline verwalten
- Gitlab Pipeline überprüfen, ob das Projekt nicht failed.

Daten Ingenieur: canger

Daten Ingenieure sind verantwortlich für die Verwaltung und Optimierung der Datenbank. Zu ihren Aufgaben gehören die Gewährleistung der Datenqualität, die Entwicklung und Implementierung von Datenmodellen. Spezifische horizontale Verantwortlichkeiten für dieses Projekt sind:

- Datenmodellierung
- Datenbank Optimierung
- Sicherstellung der Datenqualität und Datenintegrität
- Zusammenarbeit mit dem Technischen Architekten bei der Implementation des Datenaustausch zwischen den verschiedenen Schichten
- Regelmäßige Überprüfung der Dateninfrastruktur auf Sicherheitslücken und Schwachstellen

Sicherheits-Ingenieur: sschuch

Sicherheits-Ingenieure sind für die Sicherheit der Software- und Systemarchitektur verantwortlich und stellen sicher, dass alle Sicherheitsanforderungen erfüllt sind. Sie entwickeln Sicherheitsrichtlinien und -verfahren und reagieren auf Sicherheitsvorfälle. Spezifische horizontale Verantwortlichkeiten für dieses Projekt sind:

- Entwicklung von Sicherheitsrichtlinien und -verfahren
- Implementierung von Sicherheitsmaßnahmen für Datenverkehr und Datenbanken
- Überwachung von Sicherheitsbedrohungen und -vorfällen
- Sicherstellung der Einhaltung von Datenschutz- und Compliance-Anforderungen
- Integration von Sicherheitslösungen in dem Entwicklungsprozess
- Regelmäßige Überprüfung und Aktualisierung der Sicherheitsmaßnahmen

Usability Ingenieur: rfoster, jschort

Usability Ingenieure sind für die Benutzerfreundlichkeit und das Benutzererlebnis der Software verantwortlich. Sie erstellen Benutzerschnittstellen und sorgen dafür, dass die Software intuitiv und einfach zu bedienen ist. Spezifische horizontale Verantwortlichkeiten für dieses Projekt sind:

- Entwicklung von Benutzerschnittstellen und Interaktionsdesigns
- Optimierung der Benutzererfahrung durch Feedbackschleifen
- Gewährleistung der Barrierefreiheit und Zugänglichkeit der Software Integration von Benutzerfeedback in den Entwicklungsprozess
- Erstellung von Styleguides und Designrichtlinien
- Zusammenarbeit mit dem Entwicklungsteam, um eine konsistente Benutzererfahrung sicherzustellen
- Regelmäßige Überprüfung und Analyse der Benutzerdaten zur kontinuierlichen Verbesserung der Benutzererfahrung

Requirement Ingenieure: noguamalam

Requirement Ingenieure sind für die Erfassung, Analyse und Verwaltung der Anforderungen an die Software verantwortlich. Spezifische horizontale Verantwortlichkeiten für dieses Projekt sind:

- Erfassung und Dokumentation von Anforderungen
- Analyse und Priorisierung von Anforderungen
- Erstellung von Anforderungsspezifikationen und Use Cases
- Verwaltung von Änderungsanforderungen und Anforderungstraceability
- Sicherstellung der Konsistenz und Vollständigkeit der Anforderungen
- Zusammenarbeit mit dem Entwicklungsteam, um sicherzustellen, dass die Anforderungen ordnungsgemäß umgesetzt werden
- Regelmäßige Überprüfung und Aktualisierung der Anforderungen im Laufe des Projekts

Kommunikationsleiter: pnirnberger

Der Kommunikationsleiter hat folgende horizontale Verantwortlichkeiten:

- Entwicklung und Umsetzung einer umfassenden Kommunikationsstrategie.
- Interne Kommunikation:
 - Klare und effektive Kommunikation innerhalb der Projektgruppe sicherstellen.
- Externe Kommunikation:
 - Repräsentation des Projekts gegenüber externen Partnern (Tutoren).
- Krisenkommunikation:
 - Entwicklung von Krisenkommunikationsplänen und -strategien.
 - Schnelle Reaktion und angemessene Kommunikation im Falle von Krisensituationen.

Wiki-Beauftragter: pnirnberger

Der Wiki-Beauftragter hat folgende horizontale Verantwortlichkeiten:

- Aufbau und Pflege des internen Wikis als zentrale Wissensdatenbank.
- Sicherstellung der Vollständigkeit, Genauigkeit und Aktualität der Wiki-Inhalte.
- Unterstützung bei der Strukturierung und Organisation von Informationen im Wiki.
- Ansprechpartner für Fragen und Anliegen rund um das interne Wiki.

Grober Projektplan

Artefakte, die aus dem Projektauftrag entstehen und Werkzeug-unterstützt oder als eigenständige Artefakte weitergeführt werden, wurden bereits in der Work Breakdown Structure eingeplant. Die Verteilung basiert auf den Rollen und Horizontalen Verantwortungen.

Work Breakdown Structure (WBS)

Die WBS wurde auf Meilenstein-Ebene entworfen und es wurden die ersten Arbeitspakete aufgrund der Horizontalen Verantwortungen berücksichtigt. Die technischen Arbeitspakete der Iceberglist werden in der WBS in diesem Projekt nur auf Meilensteinebene berücksichtigt - wie wir technisch nach SCRUM vorgehen. Das heißt, es werden hier nur "nicht-technische" Arbeitspakete (Dokumentation, Konfiguration, etc.) eingetragen.

Nr.	Arbeitspakete	Anfang	Ende	Personentage	Verantwortlich
MS.0	Kick-Off	19.04.2024	19.04.2024	6	
1.1	Anforderungsanalyse	27.04.2024	02.05.2024	30	
1.1.1	Feature-Liste verfeinern, Ausgangssituation & Projektbeschreibung (vom PV) verfeinern	27.04.2024	28.04.2024	12	
1.1.2	Zielgruppen verfeinern, Aktoren Hierarchie erstellen, Projektplan erstellen	30.04.2024	01.05.2024	6	
MS.1	Projektdefinition, Projektauftrag	02.05.2024	02.05.2024	6	
MR-1	Management Review 1	02.05.2024	02.05.2024	6	
1.2	Anforderungs-Analyse fertigstellen	03.05.2024	03.05.2024	6	
1.2.1	Projektplan verfeinern	03.05.2024	03.05.2024	6	
2.1	Entwurf und Design	04.05.2024	04.05.2024	6	
2.1.1	Domänenmodell abschliessen	04.05.2024	04.05.2024	6	
2.1.2	Komponentendiagramm aktualisieren	04.05.2024	04.05.2024	6	
3.0	Implementierung Sprint 1	04.05.2024	12.05.2024	30	
3.0.1	DB Schema & Testdaten für Arzt, Patient, Sekretär, Ambulanz, ...	04.05.2024	12.05.2024	16	pnirnberger

3.0.2	Persistenz-Schicht, DAOs, Model Klassen	04.05.2024	12.05.2024	16	canger
3.0.3	Service Schicht & erstes GUI Panel	04.05.2024	12.05.2024	16	rfoster
3.0.4	Anmeldesystem für Nutzer	04.05.2024	12.05.2024	16	canger
3.0.6	GUI für Behandlungen	04.05.2024	12.05.2024	16	jschort
3.0.7	Maven Konfiguration (pom.xml)	04.05.2024	02.05.2024	16	rfoster
3.0.8	Tests der Persistenz-Schicht (DAOs), Test-Suites	04.05.2024	12.05.2024	16	noguamalam
MS.3	33% der Icebergliste implementiert, CRUD-Funktionalität	12.05.2024	12.05.2024	6	
IR-1	Internes Review 1 (Anforderungs Review)	13.05.2024	13.05.2024	6	
2.2	Entwurf und Design fertigstellen	14.05.2024	18.05.2024	21	
3.1	Implementierung Sprint 2 bis Version 2, siehe Iceberglis	14.05.2024	26.05.2024	60	
MS.4	57% der Icebergliste implementiert, Terminbuchung / Telemedizin testen	26.05.2024	26.05.2024	6	
MR-2	Management Review 2 (Design Review)	27.05.2024	27.05.2024	6	
3.2	Implementierung Sprint 3 bis Version 3, siehe Iceberglis	28.05.2024	09.06.2024	60	
MS.5	83% der User Stories implementiert, Service-Schicht Tests	09.06.2024	09.06.2024	6	
IR-2	Internes Review (Code Inspektion), Beta Version Demo	10.06.2024	10.06.2024	6	
3.3	Implementierung Sprint 4 bis Version 4, siehe Iceberglis	11.06.2024	23.06.2024	55	
MS.6	100% der User Stories implementiert, Akzeptanztests	23.06.2024	23.06.2024	6	
MR-3	Management Review 3: Projektabnahme	24.06.2024	24.06.2024	6	

Meilensteinbeschreibung

1. Projektauftrag

2. Systemtest

Zu diesem Zeitpunkt soll ein erster Prototyp der Applikation vorhanden sein, d.h. alle Schichten der Architektur sollen miteinander verbunden werden und ein erster Systemtest, mit der GUI und echter Datenbankverbindung, durchgeführt werden.

Dieser Meilenstein könnte auch als sog. "pre-release" betrachtet werden, d.h. die Software läuft, aber es gibt keine wirkliche Funktionalität außer getestete CRUD Methoden.

3. Version 1 (33% der Userstories)

Diese Version erfüllt alle Grundfunktionen, wie das Anlegen von Patienten und Ärzten, Sekretärin, das Anlegen von Medikamenten und Allergien sowie das Anlegen von Stationen und Ambulanzen. Damit sind die meisten grundlegenden CRUD Methoden abgedeckt.

4. Version 2 (56% der Userstories)

Diese Version erweitert das Programm um eine Chat-Funktion zwischen Arzt und Patient und um die Möglichkeit, einem Patienten einen Termin auf einer bestimmten Ambulanz zu buchen, entweder durch den Patienten selbst oder durch das Krankenhauspersonal.

5. Version 3 (83% der Userstories)

In dieser Version werden Krankenhausaufenthalte hinzugefügt, welche mehrere Behandlungen umfassen können, sowie die Möglichkeit, Patienten-, Arzt, Stations- und Ambulanz-Daten zu bearbeiten oder inaktiv zu setzen. Aufenthalte können ebenfalls bearbeitet werden.

6. Version 4 (100% der Userstories)

In Version 4 wird die Möglichkeit gegeben, dass man Administratoren Accounts anlegt und dass man die Patientenhistorie ausliest. Des Weiteren kann man in den Patientendaten nach bestimmten Behandlungen suchen und die Auslastungen der Ambulanzen werden automatisch berechnet. Als letztes Feature wird die Schnittstelle implementiert, sodass das System später einfach erweitert und durch andere Systeme ergänzt werden kann

Projektabgrenzungen

Bei dem Projekt muss eine sinnvolle Erweiterbarkeit gegeben sein, in diesem Teil wird aber vor allem auf die stabile Umsetzung der oben angeführten Anforderungen geachtet.

Folgende Features werden von bestehenden Werkzeugen unterstützt und sollen nicht von Care-Connect abgelöst werden:

- Verrechnungs/Buchhaltungssoftware
- Anbindung an die Elektronische Gesundheitsakte (ELGA)
- Keine Anbindung von medizinischen Geräten (Corpuls, etc)

Folgende Funktionen oder Arbeitspakete liegen nicht im Umfang von Care-Connect Projekt und werden auch nicht vom Entwicklerteam verarbeitet:

- Konvertierung von bestehenden Datenbeständen
- Produktivstellung der Care-Connect Software auf einem institutsinternen Webserver.
- Die Schnittstellen zu externen Systemen funktionieren über eine API, dies muss von den zuständigen Systemadministratoren/Entwicklern übernommen werden.

Schichtendiagramm

Unser Projekt wird in vier Schichten unterteilt.

Frontend:

- **Angular**
Angular ermöglicht eine Entwicklung durch Komponentenarchitektur und bietet Funktionen wie Datenbindung, Routing, Formularsteuerung und HTTP-Client-Support. Mit Angular können Entwickler eine klare Trennung von Logik und Darstellung beibehalten, um robuste und skalierbare Webanwendungen zu erstellen.

Backend:

- **REST (Java / Spring Boot)**
ermöglicht die Erstellung von Endpoints, die Ressourcen über standardisierte HTTP-Methoden wie GET, POST, PUT und DELETE verwalten.
- **Service-Schicht (Java/ Spring Boot)**
ist die Schnittstelle zwischen der Rest-Schicht und der Persistence-Schicht. Hier werden die Geschäftslogik und die Anwendungslogik der Software implementiert. Sie ist verantwortlich für die Verarbeitung von Anfragen aus dem Frontend, das Aufrufen von Datenbankoperationen und die Bereitstellung von Ergebnissen zurück an das Frontend. Zusätzlich werden User-Eingaben hier validiert.
- **Persistence-Schicht (Java / Spring Boot / JDBC / H2)**
ist für die dauerhafte Speicherung und Verwaltung von Daten verantwortlich. Es wird eine H2-Datenbank verwendet, um Informationen langfristig zu speichern. Die Persistence-Schicht bietet Funktionen zum Speichern, Abrufen, Aktualisieren und Löschen von Daten gemäß den Anforderungen der Anwendung.

Verwendete Technologien

- Java Spring Boot
- Lifecycle Management with Maven
- Spring-beans to annotate domain objects for persiting
- Persistence Layer utilizing Data Access Object Pattern in combination with JDBC Templates of the Spring framework and Dependency Injection
- H2 database
- Angular
- Unit Testing with JUnit
- Logging via Log4J Framework

- Full inline documentation (Javadoc)
- GIT (Version Control)

Lieferkomponenten

Während der Projektlaufzeit und vor allem beim Abschluss des Projektes wird dem Kunden die erstellte Software sowie dazugehörige Dokumentation übergeben. In diesem Abschnitt werden die Softwarekomponenten und Dokumentation beschrieben und eine Abgrenzung des Lieferumfangs durchgeführt.

Software - mit setup-tool

Nach Projektabschluss werden dem Kunden folgende Kernkomponenten der Software in produktionsfähigen Zustand übergeben:

- Angular Website-Dateien
- Compiled Backend Jar-Datei
- Benutzerhandbuch in Form einer Onlinehilfe (Wiki)

Artefakte & Projektdokumentation (auf Anfrage)

- Domänenmodell
- Funktionale Anforderungen
 - Anwendungsfallbeschreibung
 - Anwendungsfalldiagramme, Pakete & Aktorenhierarchie
- Testplan, Funktionale Testfälle
- ER Diagramm
- Issues in Gitlab

Abgrenzung des Lieferumfangs

Nicht im Lieferumfang des End-Produkts (MS.5/MR-3) enthalten, sind:

- Projektstrukturplan (GANTT)
- Iceberglist, Burn-Down Charts
- Artefakte des laufenden Projektmanagement (Besprechungsprotokolle, Stundenlisten, Risikoanalyse)
- UI Skizzen

Nichtfunktionale Anforderungen

Sicherheit:

- Datenbank:

Sensible Datenfelder werden in der Datenbank mittels Jasypt verschlüsselt. Der Privatekey von Jasypt wird mittels einer Environment Variable während der Runtime geladen und befindet sich dadurch nicht im Sourcecode unseres Projekts. Als sensible Datenfelder gelten z.B. Art einer Behandlung, somit

kann im Fall eines Datenlecks kein Patient mit seiner Behandlungshistorie nachvollzogen werden.

- **Login/Sessions:**
Bei einem Login sendet der User seine Login Daten per https verschlüsselt zum Backend. Im Backend wird das Passwort gehashed, gesalted und gepeppered. Danach wird der Hash in der Datenbank abgeglichen. Wenn die Hashes übereinstimmen, wird ein JWT Token erstellt, welcher als Payload Informationen wie die User ID und Rolle beinhaltet, um beim Rest Abfragen eine korrekte Accesscontrol sicherzustellen. Jeder Token hat auch ein kurzes Ablaufdatum von ca. 15 Minuten. Es wird auch ein Limit geben, wie oft ein Login Versuch durchgeführt werden kann, um Bruteforcing Versuche zu verschlechtern.
- **Least Privileges:**
Durch verschiedene Rollen, die es in unserem System gibt, ist es auch möglich, jeder Rolle nur so wenig Access wie möglich zu geben und somit den Schaden im Falle eines gehackten Accounts zu verringern.
- **Frontend/Backend:**
Kommunikation vom Frontend zum Backend muss ausschließlich über HTTPS passieren.
- **Logging:**
Während des Programmablaufs werden konstant Logs erstellt, welche keine sensiblen Daten enthalten. Diese Logs können durch Audits auf potentielle Attacken hinweisen, um diese frühzeitig abzuwehren.

Barrierefreiheit:

Barrierefreiheit ist eine grundlegende nicht-funktionale Anforderung für jede Software, insbesondere aber für eine Krankenhaussoftware, die von einer breiten Palette von Benutzern verwendet wird, darunter auch Patienten. Die Gewährleistung der Barrierefreiheit bedeutet, dass die Software für alle Benutzer zugänglich ist, unabhängig von ihren körperlichen oder kognitiven Fähigkeiten. In einer Krankenhausumgebung ist es von entscheidender Bedeutung, dass sowohl das medizinische Personal als auch die Patienten, die möglicherweise mit der Software interagieren müssen, in der Lage sind, sie ohne Hindernisse zu nutzen. Hierfür wird WCAG Single-A als Barrierefreiheit Stufe gewählt.

Leistung:

Die Software muss auch unter Last effizient arbeiten, um eine schnelle Reaktion auf Benutzeranfragen sicherzustellen. Lange Ladezeiten oder Verzögerungen können in einer medizinischen Umgebung kritisch sein. Daher sollte jede Seite des Programms innerhalb von 3 Sekunden geladen sein.

Zuverlässigkeit und Verfügbarkeit:

Krankenhaussoftware muss hochverfügbar sein, da Ausfallzeiten zu schwerwiegenden Konsequenzen führen können. Die Software sollte robust sein und Mechanismen zur Fehlererkennung enthalten, um Ausfälle zu minimieren. Als minimale Uptime werden 99,9% definiert.

Risikoabschätzung

Nr.	Typ	Priorität	Eintrittswahrscheinlichkeit	Folgerisiken	Verantwortliche	Name & Beschreibung	Gegenmaßnahmen
1	immer	hoch	mittel	2, 3	canger	Ausfall eines Projektmitglieds: Ein Projektmitglied kann, zB bedingt durch eine Krankheit, längere Zeit nicht am Projekt mitarbeiten bzw scheidet komplett aus dem Projektteam aus.	Sammeln der bereits geleisteten und noch ausständigen Arbeit; Besprechung und Aufteilung der Aufgaben innerhalb der Gruppe (zusätzlich zum Jour-Fixe, kann auch elektronisch stattfinden). Zuteilung der Rolle zu einem anderen Gruppenmitglied.
2	entwicklung	hoch	niedrig	3	jschort	Rechtzeitige Fertigstellung gefährdet: Die rechtzeitige Fertigstellung des Projekts ist gefährdet. Mögliche Ursachen könnten zu hohe Anforderungen, zu geringe zeitliche Ressourcen, Ausfall von Team-Mitgliedern, etc sein.	Reduktion von Features anhand ihrer Priorisierung (abhängig von aktuellem Fortschritt)
3	entwicklung	hoch	niedrig	2	sschuch	Sicherheit ist gefährdet: Teile der sensiblen Daten sind nicht ausreichend sicher aufbewahrt.	Ermittlung der bereits bestehenden Sicherheitsmaßnahmen. Schwachstellenanalyse des bestehenden Systems. Bessere Encryption verwenden, ...
4	planung, entwicklung	hoch	mittel	2	pnirnberger, rfoster	Kommunikationsprobleme: Missverständnisse oder ineffektive Kommunikation zwischen Teammitgliedern können den Fortschritt des Projekts behindern.	Verwendung von klaren Kommunikationskanälen und -protokollen. Regelmäßige Meetings zur Überprüfung des Projektstatus und zur Klärung von Fragen.
5	planung, entwicklung	hoch	niedrig	2	rfoster	Notwendige Libraries nicht verfügbar: Während des Projektverlaufs stellt sich heraus, dass für ein bestimmtes Feature notwendige	Sofern der Aufwand vertretbar ist, können die Funktionalitäten selbst entwickelt, bzw andere Libraries adaptiert werden; Ist dies nicht

						Libraries nicht (frei) verfügbar sind.	möglich, muss das Feature so weit wie notwendig reduziert werden.
6	planung, entwicklung	mittel	mittel	2	noguamalam	Unzureichende Design-Entscheidungen: Während der Implementierung stellt sich eine Design-Entscheidung als unzureichend heraus.	Design-Entscheidungen werden so früh wie möglich mit allen Projektmitgliedern besprochen, die die Entscheidung bezüglich Ihrer Rolle beurteilen; Modulares Design, damit die Auswirkungen falscher Design-Entscheidungen möglichst klein gehalten werden

Informationswesen

Die Informationsstruktur für das Projekt wird folgendermaßen aussehen:

- Wöchentliche Treffen intern (Jour-Fixe)
- Wöchentliche Treffen mit dem Tutor (Tutortreffen)
- Insgesamt 5 Reviews (2 IRs, 3 MRs) zu den Meilensteinen 1,3,4,5 und 6
- Elektronische Kommunikation synchron mit Discord und Google Docs. Die Mailingliste ist mit niedrigerer Priorität zu verwenden.
- Kommunikation mit den Auftraggebern und Tutor per Mailingliste

Besonderheiten

Es kommen 4 Schichten zum Einsatz, eine Angular Applikation, welche im Browser läuft, und eine Restschicht, eine Serviceschicht inklusive Validierer und eine Persistenzschicht im Backend. Das Frontend und Backend kommunizieren mittels HTTPS miteinander. Um die Trennung zwischen dem Relationenmodell und dem objektorientierten Modell zu schaffen, werden O/R Mapper verwendet.