

Bearbeitungsbeginn: 01.09.2020

Vorgelegt am: 28.02.2021

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang Online Medien

an der Fakultät Digitale Medien

Referent: Henri Charles Joß

Matrikelnummer: 256193

Konzeption und Implementierung eines Dokumentationssystems

Erstbetreuer: Prof. Dr. Gabriel Rausch

Zweitbetreuer: Max Mittag

Abstract

Am Fraunhofer ISE wird an neuen Technologien und der Weiterentwicklung von Fotovoltaik-Modulen gearbeitet. Hierzu werden wissenschaftliche Versuche und Prozessvariationen durchgeführt und Prototypen hergestellt. Die beteiligten Proben, Prozesse und sonstige verwendete Parameter werden hierbei dokumentiert.

Da das derzeitige Dokumentationssystem einige Schwächen aufweist, widmet sich diese Bachelorarbeit der Entwicklung eines neuen und verbesserten Systems. Dabei beschäftigt sie sich mit der Frage, wie die konkrete Ausgestaltung und Implementierung des Systems aussehen muss, um eine langfristig einheitliche und zugängliche Dokumentation zu gewährleisten.

Es wurde ein adäquates Vorgehensmodell gewählt und nach dessen Prinzipien eine Anforderungsanalyse mit Interviews und Online-Umfrage durchgeführt. Auf der Basis der gewonnenen Erkenntnisse wurde eine flexible und umfangreiche Web-Anwendung mit Datenbankanbindung konzipiert und umgesetzt.

Die anwendungsorientierte Forschung erhält damit die Grundlage für ein neues umfangreiches Dokumentationssystem zur Sicherung guter wissenschaftlicher Praxis.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	V
Quellcodeverzeichnis.....	VI
1. Einleitung.....	1
1.1 Vorstellung der Fraunhofer Organisationseinheit.....	1
1.2 Fraunhofer ISE.....	1
1.3 Vorstellung des Projekts.....	2
1.4 Problemstellung.....	2
1.5 Zielsetzung.....	2
2. Methodik.....	6
2.1 Phasen der Softwareentwicklung.....	6
2.1.1 Anforderungsanalyse.....	6
2.1.2 Designphase.....	7
2.1.3 Implementierung.....	7
2.1.4 Tests und Integration.....	7
2.1.5 Qualitätssicherung.....	7
2.2 Vorgehensmodelle in der Softwareentwicklung.....	8
2.2.1 Wasserfallmodell.....	8
2.2.2 Iterative Entwicklung.....	8
2.2.3 Iterativ Inkrementelle Entwicklung.....	9

2.2.4	Agile Softwareentwicklung	9
2.3	Auswahl für das Projekt	10
3.	Dokumentation	12
3.1	Grundlagen & Theorie zur Dokumentation.....	12
3.2	Istzustand Fraunhofer ISE	12
4.	Anforderungsanalyse	15
4.1	Vorgehen.....	15
4.2	Rahmenbedingungen.....	16
4.3	Mitarbeiterbefragung	18
4.3.1	Vorgehen	18
4.3.2	Durchführungsbeschreibung.....	18
4.3.3	Auswertung.....	19
4.4	Anforderungsbestimmung	20
4.5	Aktivitäten.....	21
4.6	Analyse der zu erfassenden Daten	23
4.7	Entscheidung für die Eigenentwicklung	23
5.	Konzeption.....	25
5.1	Benutzeroberfläche	25
5.2	Grobdesign.....	33
5.2.1	Technologien	33
5.2.2	Architektur.....	35
5.3	Datenstruktur.....	36
5.4	Vergleich zur Anforderungsanalyse	40
6.	Implementierung	43
6.1	Serverseitig	43

6.1.1	Datenbank.....	43
6.1.2	Grandstack.....	44
6.2	Clientseitig.....	51
7.	Schlussteil	57
7.1	Zusammenfassung.....	57
7.2	Ergebnisse	57
7.3	Ausblick.....	58
	Literaturverzeichnis.....	60
	Anhangsverzeichnis	64
	Anhang.....	65
	Eidesstattliche Erklärung.....	87

Abbildungsverzeichnis

Abbildung 1: Entwicklungsphasen	6
Abbildung 2: Aufbau derzeitiges System	13
Abbildung 3: Anforderungsgranularität	16
Abbildung 4: Aktivitätsdiagramm Probe erstellen	22
Abbildung 5: Layout Benutzeroberfläche	26
Abbildung 6: Flussdiagramm Navigation	28
Abbildung 7: Wireframe Listenansicht Experimente	29
Abbildung 8: Screendesign Probe erster Entwurf	30
Abbildung 9: Screendesign Detailsseite Probe	32
Abbildung 10: Listenansicht Probe	33
Abbildung 11: Technologien Übersicht	34
Abbildung 12: Klassendiagramm	35
Abbildung 13: Teilgraph Experiment	37
Abbildung 14: Zuordnungsmatrix	38
Abbildung 15: Teilgraph Planungsschritt	39
Abbildung 16: Vollständige Datenstruktur als Graph	40
Abbildung 17: Value-Propostion-Canvas	42
Abbildung 18: Grundsätzlicher Aufbau des Frontends	52
Abbildung 19: Listenkomponente Aufbau & Ablauf	53
Abbildung 20: Listenkomponente Verlinkungen	54
Abbildung 21: Bearbeitungsansicht am Beispiel einer Probe	55
Abbildung 22: Probendetailsseite implementierte Version	56

Tabellenverzeichnis

Tabelle 1: Prinzipien des Probendokumentationssystems 17

Quellcodeverzeichnis

Quellcode 1: Beispiel einer Cypher Anweisung.....	43
Quellcode 2: inferSchema Funktion zur Generierung des GraphQL Schemas	45
Quellcode 3: Auszug aus der GraphQL Schema Datei (Schema für Projekt)	46
Quellcode 4: makeAugmentedSchema Funktion zur Generierung des GraphQL Endpunktes.....	48
Quellcode 5: Beispiel einer eigens entwickelten Mutation zur Erzeugung eines Projekts	50

1. Einleitung

1.1 Vorstellung der Fraunhofer Organisationseinheit

Die Fraunhofer Gesellschaft ist mit 74 Instituten und rund 28.000 Mitarbeitern die größte Organisation für angewandte Forschung in Deutschland. Dabei stehen die Entwicklung und Forschung von Zukunftstechnologien im Fokus. Das Ziel der daraus resultierenden Erkenntnisse ist dabei die Verwendung in Wirtschaft und Industrie. Die 1949 gegründete Organisation trägt somit einen signifikanten Teil zur deutschen Forschungsarbeit bei und finanziert sich dabei zu 70 Prozent aus eigens erwirtschafteten Aufträgen, welche z. T. aus der Industrie sowie öffentlich finanzierten Projekten stammen. Die restlichen 30 Prozent werden durch Bund und Länder zur finanzunabhängigen Forschung an potentiell zukünftig relevanten Technologien subventioniert. Die einzelnen Institute sind jeweils auf unterschiedliche Technologien spezialisiert und agieren dabei weitestgehend unabhängig vom Hauptsitz in München.¹

1.2 Fraunhofer ISE

Das Fraunhofer Institut für Solare Energiesysteme (ISE) in Freiburg beschäftigt sich mit der Forschung und Entwicklung umweltfreundlicher Energiekonzepte. Mithilfe der Forschungsschwerpunkte Energiegewinnung, Energieeffizienz, Energieverteilung und Energiespeicherung, trägt das Institut einen erheblichen Beitrag zur Energiewende bei.²

In der Abteilung PV des Instituts wird an der Forschung und Weiterentwicklung von Photovoltaik-Modulen und deren Herstellungsprozessen gearbeitet. Dabei werden in den Laboratorien der Abteilung Prototypen hergestellt, Versuche und Prozessvariationen durchgeführt. Die beteiligten Proben, Materialien und Prozesse sowie Messprotokolle und sonstige verwendete Parameter werden hierbei dokumentiert.

¹ Vgl. „Fraunhofer“, o. D.

² Vgl. „Fraunhofer ISE“, o. D.

1.3 Vorstellung des Projekts

Im Rahmen dieser Bachelorarbeit soll ein bereits vorhandenes Dokumentationssystem signifikant erweitert und verbessert werden. Das derzeitige System beschränkt sich im Wesentlichen auf die Dokumentation der Proben und den dabei beteiligten Materialien. Ein großer Teil der anfallenden Daten wird inkonsistent und an unterschiedlichen Orten gespeichert und verwaltet. Das System zur Dokumentation der Proben gestaltet sich in Form einer Exceltabelle, die nur grobe Vorgaben zur Dokumentation macht.

Das neue System soll die Problematiken der derzeitigen Dokumentation beseitigen und langfristig in der Abteilung eingesetzt werden. Dies soll in Zusammenarbeit bzw. durch regen Austausch mit den relevanten Stakeholdern des derzeitigen und zukünftigen Systems der Abteilung PVM geschehen. Diese Arbeit stellt dabei sowohl die Dokumentation des zu erstellenden Systems als auch die Betrachtung der theoretischen Grundlagen der Vorgehensweisen und Methoden dar.

1.4 Problemstellung

Aufgrund der inkonsistenten, verteilten und unvollständigen Dokumentation ist die Bereitstellung und Zugänglichkeit der anfallenden Daten nur eingeschränkt möglich. Die mangelnde Standardisierung erschwert den schnellen und gezielten Zugriff auf akut benötigte Daten. Auswertungen und dessen Verfügbarmachung sind nur eingeschränkt möglich. Um diese Probleme zu beheben und die Arbeitsqualität und Effizienz der Abteilung zu steigern, gestaltet sich die grundsätzliche Fragestellung der Bachelorarbeit wie folgt:

Wie muss die konkrete Ausgestaltung und Implementierung eines Dokumentationssystems aussehen, um den Anforderungen der Mitarbeiter gerecht zu werden und eine langfristig einheitliche und zugängliche Dokumentation zu gewährleisten?

1.5 Zielsetzung

Das Ziel dieser Bachelorarbeit ist die Konzeption und Umsetzung eines Dokumentationssystems das klaren Prinzipien folgt, durch die sich eine

kohärente, qualitative und langfristige Datenhaltung ergibt. Die übergeordnete These der Arbeit gestaltet sich dabei wie folgt:

Hauptthese:

Durch die Vereinheitlichung und Standardisierung eines Dokumentationssystems kann die Qualität und die Effizienz der Dokumentation gesteigert werden.

Neben einer Probendokumentation soll eine Versuchs- und Prozessdokumentation erstellt sowie ggf. notwendige weitere Hilfsstrukturen etabliert werden (bspw. Stammdatenverwaltung: Anlagenlisten, Mitarbeiter, Materialbibliothek etc.). Zudem soll das System die standardisierte Erfassung der Dokumentation gewährleisten bzw. unterstützen.

Aus dieser Hypothese, die die Entwicklung des neuen Systems und dieser Arbeit zur Folge hat, ergeben sich verschiedene untergeordnete Nebenthesen.

Nebenthese 1:

Um eine situativ adäquate Arbeitsweise und eine langfristige Weiterentwicklung und Wartung des Systems zu gewährleisten, muss ein passendes Vorgehensmodell der Softwareentwicklung gewählt werden.

Hierfür sollen die Phasen der Softwareentwicklung und dessen Vorgehensmodelle vorgestellt und miteinander verglichen werden, um anschließend ein geeignetes Modell für diese Arbeit zu wählen.

Nebenthese 2:

Durch die Analyse des vorhandenen Systems und intensiver Auseinandersetzung mit beteiligten Mitarbeitern der Abteilung können klare Anforderungen und Ziele definiert werden.

Es soll untersucht werden, welche Maßnahmen bei der konkreten Ausgestaltung getroffen werden müssen, um eine hohe Akzeptanz und sinnvolle Verwendung des Systems zu erreichen. Mithilfe verschiedener Methoden und Modelle wird eine Anforderungsanalyse mit relevanten Beteiligten durchgeführt.

Nebenthese 3:

Durch intensiven Austausch und Validierung kann in der Konzeption die Benutzerfreundlichkeit und Zufriedenheit der Mitarbeiter mit der Benutzeroberfläche gesteigert werden.

Ausgehend davon soll in einer Konzeptionsphase ein umfangreiches Konzept für die Datenstruktur und Benutzeroberfläche erstellt werden. Hierbei sollen Aufbau, Struktur und das Oberflächendesign in mehreren Feedbackschleifen besprochen und verbessert werden.

Nebenthese 4:

Um die in der Konzeption modellierten Anforderungen technisch umsetzen zu können, müssen geeignete Technologien gewählt und eine strukturell logische und verständliche Implementierung erarbeitet werden.

Das System soll in Form einer webbasierten Anwendung mit Datenbankbindung umgesetzt werden. Die Schnittstelle zum Backend (API), die dort verwendeten Datenstrukturen und das Datenbankkonzept soll dabei in enger Zusammenarbeit mit Mitarbeitern einer Nachbarabteilung erarbeitet werden. Der erstellte Quellcode soll verständlich und erweiterbar aufgebaut sein.

Nebenthese 5:

Zur Überprüfung, ob eine Implementierung einsatzfähig ist und um spätere Überarbeitungen bzw. Verbesserungen zu erleichtern, müssen umfangreiche Tests vorgenommen werden.

Nachdem das System mithilfe der definierten Strukturen programmatisch umgesetzt worden ist, soll die Implementierung mit Tests auf unterschiedlichen Ebenen validiert werden. Dazu zählen unter anderem System, Unit und Akzeptanztests, die sowohl die technische als auch die Qualität der Benutzerfreundlichkeit überprüfen sollen. Wenn diese Phase erfolgreich abgeschlossen ist, kann die Software mithilfe einer Benutzerschulung in der Abteilung eingeführt werden.

Nebenthese 6:

Das Verständnis eines Softwareprodukts ergibt sich aus der Dokumentation des Erstellungsprozesses, der Struktur und der Implementierung der Software.

Die entstandene Software und der Prozess der Entstehung sollen nachvollziehbar dokumentiert werden. Die Dokumentation für Anforderungen, Aufbau und Architektur der Software soll es später Dritten ermöglichen, schnelles Verständnis für potenzielle Weiterentwicklungen oder Wartungen zu entwickeln.

2. Methodik

Softwareentwicklung lässt sich in verschiedene Phasen gliedern. Um eine für das Projekt adäquate Methodik zu wählen, folgt ein kurzer Überblick über die typischen Phasen und den verschiedenen Herangehensweisen bzw. Vorgehensmodellen.

2.1 Phasen der Softwareentwicklung

Betrachtet man das Vorgehen in unterschiedlichen Softwareprojekten, so stellt sich heraus, dass es bestimmte Prozesse gibt »die typischerweise in jedem Software-Projekt vorkommen, dies sind: Anforderungsanalyse, Design, Implementierung sowie Test und Integration. All diese Prozesse werden von Qualitätssicherungsmaßnahmen begleitet.«³

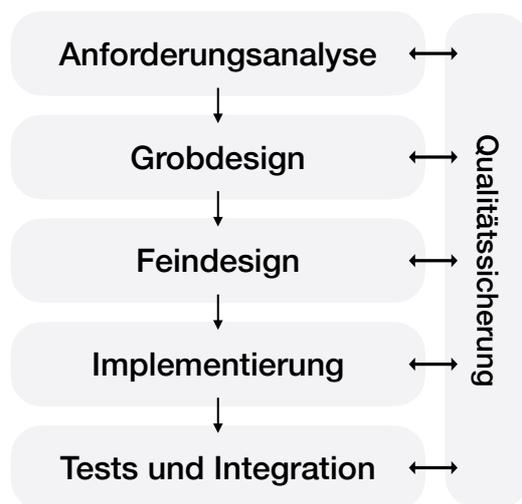


Abbildung 1: Entwicklungsphasen

Quelle: In Anlehnung an Kleuker, 2018, S.24

2.1.1 Anforderungsanalyse

Die erste Phase in der Entwicklung eines Softwareprodukts ist immer die Anforderungsanalyse. In dieser Phase soll geklärt werden welche Anforderungen, Erwartungen und Wünsche der Kunde an das finale Produkt hat. Es sollen Rahmenbedingungen und konkrete messbare Ziele definiert

³ Kleuker, 2018, S. 23

werden.⁴ Da dies maßgeblich für die endgültige Zufriedenheit des Kunden ist, muss ein enger Dialog mit diesem stattfinden. Nicht immer kann ein Kunde seine Ergebniswünsche konkret äußern, umso wichtiger ist es also durch gezielte Fragestellungen und konkrete Vorschläge dies ausfindig zu machen. Das Resultat dieser Phase sind die sogenannten Anwenderforderungen.⁵

2.1.2 Designphase

In dieser Phase unterscheidet man zwischen den beiden Teilen des Grob- und des Feindesigns. Im Grobdesign werden die Anforderungen in ein verwertbares Modell überführt mithilfe dessen die grobe Struktur, Softwarearchitektur und Datenbankdesign bestimmt werden kann.⁶ Anschließend wird im Feindesign das Design der konkreten Implementierung (Softwaredesign) sowie das Design der Benutzeroberfläche (Interfacedesign) geplant.

2.1.3 Implementierung

Sobald Struktur und Modelle festgelegt sind können diese mit verschiedenen Programmiersprachen und Technologien umgesetzt werden. Das Ziel dieser Phase ist es eine lauffähige Software bereitzustellen.⁷

2.1.4 Tests und Integration

In der letzten Phase müssen alle entstandenen Teile der Software zu einem fertigen Softwareprodukt gebündelt bzw. integriert werden. Diese Software muss vor der finalen Auslieferung umfangreich getestet werden, dabei unterscheidet man zwischen System, Unit und User bzw. Akzeptanztests.⁸

2.1.5 Qualitätssicherung

Begleitend zu allen Phasen spielt die Qualitätssicherung eine wichtige Rolle. »Zentrale Aufgabe der Qualitätssicherung ist es, für jedes Teilprodukt im gesamten Entwicklungsprozess sicher zu stellen, dass vor dem Projekt

⁴ Vgl. Brandt-Pook & Kollmeier 2015

^{5, 6} Vgl. Kleuker, 2018

⁷ Vgl. Alshamrani & Bahattab, 2015

⁸ Vgl. Jorgensen, 2018

definierte Qualitätskriterien erfüllt sind. Dabei dürfen Folgearbeiten erst beginnen, wenn die benötigten Produkte eine geforderte Qualität erreicht haben. «⁹

2.2 Vorgehensmodelle in der Softwareentwicklung

Auch wenn die in Abschnitt 2.1 vorgestellten Phasen nicht zwangsläufig in jedem Vorgehensmodell genauso auftreten müssen, so gibt es doch Ähnlichkeiten in der grundlegenden Struktur.¹⁰ Im Folgenden ein kurzer Überblick über die bekanntesten Modelle und deren Eigenschaften.

2.2.1 Wasserfallmodell

Das Wasserfallmodell zeichnet sich durch ein sequenzielles Vorgehen und Abarbeiten der einzelnen Phasen aus. Dabei müssen einzelne Phasen abgeschlossen sein, um in die nächste Phase zu springen. Beendete Phasen können durch einen Meilenstein gekennzeichnet werden, bei dem Phasenergebnisse validiert werden. Anschließende Phasen knüpfen jeweils an die Ergebnisse der vorherigen Phase an.¹¹ Ursprünglich war ein Zurückspringen in vorherige Phasen dabei nicht vorgesehen, was zeitweise zu mangelnder Flexibilität führen konnte, wenn im Nachhinein ein Fehler in einer vorherigen Phase entdeckt wurde.

Das Hauptproblem an dieser Vorgehensweise ist, dass theoretisch alle Anforderungen bereits nach Abschluss der ersten Phase vorliegen müssten. Tatsächlich ist dies aber »[...] praktisch unmöglich«¹² denn »durch die intensivere Beschäftigung mit dem Projekt werden neue Überlegungen beim Kunden angestoßen, der erkennt, wie die resultierende Software seine Probleme besser lösen könnte.«¹³

2.2.2 Iterative Entwicklung

Um Problematiken wie im Wasserfallmodell zu vermeiden wurden neue Konzepte wie z. B. die iterative Entwicklung ins Leben gerufen. Sie beschreibt

⁹ Kleuker, 2018, S.25

¹⁰ Vgl. Berg et al., 2014

¹¹ Vgl. Kleuker, 2018

^{12, 13} Kleuker, 2018, S. 26

die mehrfache Wiederholung aller Phasen in mehreren Iterationen, um frühzeitig auf erkannte Probleme oder etwaige Änderungen eingehen zu können. Dies hat den großen Vorteil, dass der Kunde frühzeitig mit dem Produkt konfrontiert wird und somit schnell kommuniziert werden kann, welche Teile des Produkts den Anforderungen des Kunden entsprechen und welche nicht.¹⁴ Anschließend kann das Produkt mit jeder Iteration weiter verfeinert werden, bis schlussendlich alle Anforderungen des Kunden erfüllt sind.

2.2.3 Iterativ Inkrementelle Entwicklung

Diese Entwicklungsweise ist eng verwandt mit der iterativen Entwicklung, mit dem Unterschied, dass bei jedem Durchlauf der Phasen auch weitere Funktionalität hinzugefügt wird. Beispielsweise wird in einer ersten Iteration nur die Basisfunktionalität umgesetzt und anschließend in weiteren Iterationen der Funktionsumfang des Produkts erweitert.¹⁵

2.2.4 Agile Softwareentwicklung

Die Agile Softwareentwicklung versteht sich als eine Menge an Leitsätzen Prinzipien und Werten, wobei sich die daraus entstandenen Methoden heutzutage weitestgehend in der Durchführung von Softwareprojekten durchgesetzt haben.

Folgende vier Grundprinzipien wurden dabei definiert:

- »Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- Funktionierende Software mehr als umfassende Dokumentation
- Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
- Reagieren auf Veränderung mehr als das Befolgen eines Plans«¹⁶

Die Leitfäden wurden dabei im sogenannten Agilen Manifest festgehalten, welches 12 Prinzipien bzw. Verhaltensweisen festlegt nach denen gehandelt werden soll.

¹⁴ Vgl. Hoffmann, 2013

¹⁵ Vgl. Kleuker, 2018

¹⁶ Beck et.al., 2001

Zu den bekanntesten Ausprägungen der agilen Softwareentwicklung zählen heutzutage Scrum und Kanban.

2.3 Auswahl für das Projekt

Bei der Auswahl eines geeigneten Vorgehensmodells muss berücksichtigt werden, dass ein sehr begrenzter Zeitrahmen für das Projekt im Sinne dieser Bachelorarbeit zur Verfügung steht. Da das Projekt in seinem Umfang theoretisch deutlich den Rahmen sprengen könnte, muss in jedem Fall gewährleistet sein, dass sich das Projekt am Ende in einem konsistenten Zustand befindet. D.h. unabhängig vom endgültigen Umfang des entstandenen Systems muss ein Funktionieren sichergestellt werden und die Möglichkeit einer nahtlosen Weiterentwicklung bestehen.

Würde man das Wasserfallmodell nutzen würde zum einen die große Gefahr bestehen Probleme, und damit verbundene Zeitverzögerungen, erst zu spät zu erkennen und zum anderen, aufgrund des potentiell großen Umfangs des Systems, die Anforderungsanalyse zu zeitintensiv ausfallen.

Mithilfe der Iterativ Inkrementellen Entwicklung lassen sich diese Risiken vermeiden, indem in mehreren Iterationen Funktionalität entwickelt und zum System hinzugefügt wird. Somit kann die erste Iteration vor allem der Klärung der grundsätzlichen Anforderungen und dem frühzeitigen Erkennen von Problemen technischer oder organisatorischer Art dienen. Es kann sich unabhängig vom Gesamtfortschritt auf den sauberen Abschluss einer Iteration konzentriert werden, anstatt aufgrund von Zeitdruck Qualitätseinbußen in Kauf nehmen zu müssen.

Um den Aufwand für Organisation und Planung gering zu halten und nicht durch die Vorgaben einer Methode beschränkt zu werden, soll in dieser Arbeit eine Mischform aus folgenden Methoden verwendet werden. Das Iterativ inkrementelle Vorgehen aus den oben genannten Gründen kombiniert mit einigen Prinzipien der agilen Softwareentwicklung, um den besonderen Umständen und Herausforderungen dieses Projekts gerecht zu werden.

Die Orientierung an den Prinzipien der agilen Softwareentwicklung erleichtert zudem die Zusammenarbeit mit den Mitarbeitern der Abteilung, indem Termine und Absprachen flexibel und unbürokratisch vorgenommen werden können. Zusätzlich sollen einige Vorgehensweisen aus Ausprägungen wie Scrum verwendet werden, wie z. B. den Einsatz von User-Storys, um die funktionalen Anforderungen des Systems zu definieren.

3. Dokumentation

3.1 Grundlagen & Theorie zur Dokumentation

Der grundsätzliche »Sinn und Zweck der Dokumentation [...] ist das gezielte Wiederfinden und Nutzbarmachen von Dokumenten und Information«¹⁷. Dokumentation umfasst nicht nur die reine Speicherung von Wissen, vielmehr ist »Weder das Sammeln noch das Ordnen von Dokumenten oder Informationen [...] für sich allein genommen schon Dokumentation«¹⁸. Der ausschlaggebende Teil hingegen ist die sogenannte Güte einer Dokumentation, denn sie gibt an »ob die für eine Suchfrage relevanten - und nur die relevanten - Dokumente wiedergefunden werden«¹⁹.

Für Dokumentation spielt also die Verarbeitung und Verfügbarmachung der Information eine größere Rolle als die eigentliche Speicherung. In diesem Kontext soll das zu entwickelnde System zusätzlich zur Überarbeitung der Erfassung von relevanten Daten auch deren zielgerichtete Ausgabe ermöglichen. Besonders für diese zielgerechte Ausgabe ist es von Nöten einheitliche Vorgaben für die Ausgestaltung der Dokumentation zu machen. Nur dies ermöglicht die spezifische Suche mithilfe geeigneter Schlagwörter und das gezielte Filtern nach bestimmten Kriterien.

3.2 Istzustand Fraunhofer ISE

Um den Bedarf der Verbesserung und Erweiterung des vorhandenen Systems besser verstehen zu können, im Folgenden ein kurzer Überblick über den Istzustand des vorhandenen Systems und dessen Problematiken.

Das derzeitige System ist im Wesentlichen nur für die Dokumentation der Proben konzipiert und gestaltet sich in Form einer Exceltabelle, die auf einem internen Server des Fraunhofer ISE verwaltet wird. Zur Dokumentation von Versuchen/Experimenten wurde das Probendokumentationssystem aufgrund der hohen Ähnlichkeit gespiegelt. Beide Systeme sind nicht miteinander

^{17, 18, 19} Gaus, 2005, S.11

verknüpft. Das Versuchsdokumentationssystem ist nicht etabliert. Das System wird derzeit hauptsächlich nach der Arbeit im Labor am Arbeitsplatz der Mitarbeiter verwendet, da im Labor selbst lediglich an einem Computer der Zugriff darauf möglich ist.

Die Tabelle der Probendokumentation gibt folgende Angaben standardisiert erfasst vor: ID, Datum, Projekt, Mitarbeiterkürzel und Probenbezeichnung. Die Probenbezeichnung der Proben bzw. Versuche ist dabei durch eine Kombination aus Datum, Projekt-Nr., Kurzzeichen des Mitarbeiters und einer ID einheitlich umgesetzt. Für die frei wählbare ID wird allerdings hauptsächlich eine fortlaufende Nummerierung gewählt, anstatt eine konkret verständliche Bezeichnung der Probe oder des Versuchs.

Die weiteren Spalten sind Probentyp, Probenherstellung und anschließend mehrere Spalten für beteiligte Materialien. Die Vorgehensweise bei der Erfassung der Herstellung von Proben oder die Durchführung von Versuchen ist nicht standardisiert. Es bestehen keine direkten Verweise oder Beziehungen zwischen Projekten, Versuchen oder Proben, sondern lediglich indirekt händisch eingetragene Nummern bzw. ID's als Verweis auf die Zugehörigkeit. Die jeweilig verwendeten Prozesse und deren Anlagen sowie eingestellte Programme oder Parameter werden dabei inkonsistent dokumentiert. Bei der Herstellung von Proben verwendete Materialien werden ebenfalls händisch und dadurch nicht einheitlich eingetragen.

ID	Datum	Projekt	Kürzel	Probenbezeichnung	Probentyp	Herstellung	Material	Material2	...
00001	28.02.21	999	Hjo	280221_999_Hjo_00001	Modul	Hier wird die Herstellung beschrieben	ZelleA + Eigenschaften	ZelleB + Eigenschaften	
...									
...									

Abbildung 2: Aufbau derzeitiges System

Quelle: Eigene Darstellung

In einer Organisationsanweisung werden zwar Vorgaben zur einheitlichen Dokumentation zwecks Prozessen, Parametern und Materialien definiert, das derzeitige System ist allerdings nicht darauf ausgelegt diese auch einzuhalten. Dies liegt an der grundsätzlichen Verwendungsweise von Microsoft Excel, dass lediglich die Erfassung mithilfe einfacher Textfelder zulässt.

Durch diese unvollständige Systemgestaltung und Nutzereinbindung ist bspw. die gezielte Meta-Auswertung über mehrere hergestellte Proben nicht möglich (bspw. durch uneinheitliche Materialbezeichnungen). Außerdem besteht eine mangelnde Schriftlichkeit und somit resultierende personengebundene Datenhaltung, die sich als flüchtig erweist.

4. Anforderungsanalyse

4.1 Vorgehen

In der agilen Softwareentwicklung ist die Anforderungsanalyse bzw. das Requirements Engineering Teil einer jeden Iteration. Somit können Anforderungen durch den konstanten Erkenntnisgewinn stetig geändert, verbessert oder verworfen werden. Um mögliche Einschränkungen und Rahmenbedingungen frühzeitig zu erkennen und die grundsätzlich gewünschte Basisfunktionalität sichtbar zu machen, ist die Anforderungsanalyse allerdings besonders stark in der ersten Iteration ausgeprägt.²⁰ Eine grundsätzliche Gefahr dieses Vorgehens ist die ständige Erweiterung des Gewünschten, welche mit einem gesetztem Budget (Zeit, Aufwand) kollidieren kann.

Im ersten Schritt des Requirements Engineering müssen zunächst die Stakeholder, also Personen, Systeme oder Organisationen die Einfluss auf das Projekt haben oder vom Projekt beeinflusst werden, ermittelt werden.²¹ Da im Falle des Dokumentationssystems alle Mitarbeiter der Abteilung, die aktiv das derzeitige System verwenden oder das neue System verwenden werden, als Stakeholder einzustufen sind, sollten möglichst die Mitarbeiter in die Anforderungsanalyse mit einbezogen werden. Hierzu wird eine Befragung als Methode gewählt. Die Befragung der Mitarbeiter geschieht dabei in einem zweiseitigen Verfahren. Zum einen werden alle Mitarbeiter gebeten an einer Online-Umfrage teilzunehmen. Zum anderen werden Interviews mit ausgewählten Mitarbeitern geführt, die eine besondere Rolle für das Dokumentationssystem spielen, da sie bspw. besonders umfangreich mit dem derzeitigen System agieren, langjährige Erfahrungen haben oder bereits an der Entwicklung des aktuellen Systems beteiligt waren.

²⁰ Vgl. Partsch, 2010

²¹ Vgl. International Requirements Engineering Board (IREB), 2017

Die aus diesen Befragungen gewonnen Erkenntnisse werden anschließend in Anforderungen auf unterschiedlichen Granularitätsleveln (Siehe Abb. 3) umgewandelt.

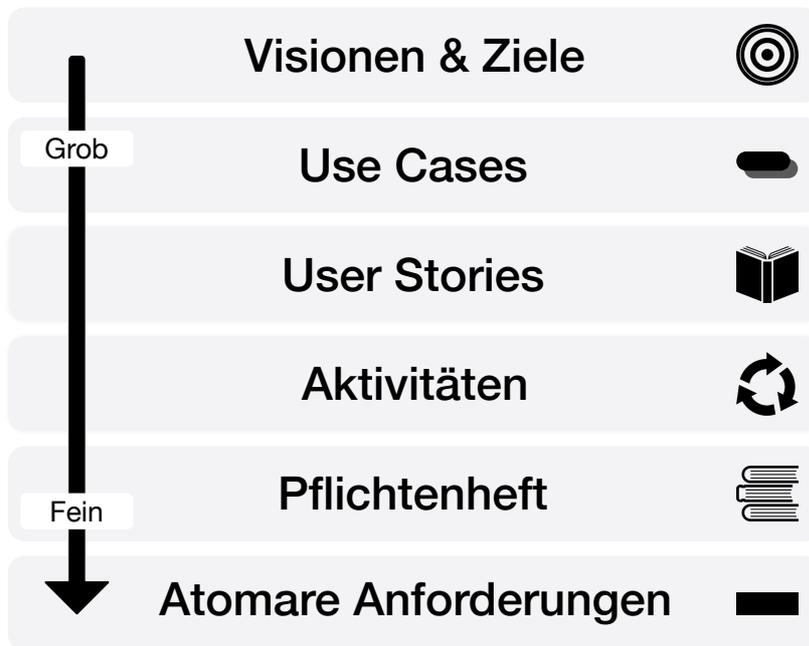


Abbildung 3: Anforderungsgranularität
Quelle: In Anlehnung an Bergsmann, 2018

4.2 Rahmenbedingungen

Bevor die Anforderungen bestimmt werden können müssen allerdings zuerst die Rahmenbedingungen des Projekts genauer betrachtet werden. In einer Organisationsanweisung der Abteilung PVM wurden Richtlinien zur Dokumentation festgehalten. Diese Anweisung dient der Einhaltung guter wissenschaftlicher Praktiken und der Verbesserung der Arbeitsqualität. Sie gibt vor, dass jedwede Dokumentation auf eine Art und Weise zu erfolgen hat, dass ein Dritter hergestellte Proben oder durchgeführte Versuche nachvollziehen und ohne weitere Information erneut produzieren oder durchführen könnte. Weiterführend definiert sie bestimmte Attribute für Materialien und Prozesse, die bei jedem Eintrag vorhanden sein müssen. Außerdem beschreibt sie, dass Proben mit einer eindeutigen und vorgegebenen Kennzeichnung bezeichnet werden müssen, um

Verwechslungen zu vermeiden und sie eindeutig Projekten oder Personen zuordnen zu können.

Zusätzlich wurde eine Menge an Prinzipien definiert, die für das Dokumentationssystem generell gelten müssen. (Siehe Tab. 1)

1	Verbindlichkeit & Willkürfreiheit	Die Probendokumentation ist für alle Proben verbindlich. Sie ist gemäß der anderen genannten Ziele durchzuführen.
2	Vollständigkeit	Die Dokumentation ist stets vollständig durchzuführen. Erfasst [sic] werden müssen mindestens: <ul style="list-style-type: none"> - Probenbezeichnung - alle zur Herstellung / Herkunft notwendigen Daten - alle an der Probe durchgeführten Versuche - alle zur ordnungsgemäßen Entsorgung der Probe notwendigen Informationen
3	Universalität	Die Dokumentation einer Probe muß [sic] unabhängig von ihrer Herkunft, Gestalt, Art und Verwendung möglich sein.
4	Richtigkeit	Die Dokumentation muß [sic] frei von Fehlern sein. Jegliche aus der Dokumentation folgenden Zuordnungen müssen von einer Richtigkeit ausgehen können. Die Struktur der Dokumentation und damit verbundener Prozesse muß [sic] die Fehlerfreiheit sicherstellen.
5	Einfachheit	Die Probendokumentation darf den bei der Einhaltung der Grundsätze wissenschaftlichen Arbeitens üblichen Aufwand nicht überschreiten.
6	Nachvollziehbarkeit	Die Dokumentation muß [sic] so gestaltet sein, daß [sic] sie eine langfristige Nachvollziehbarkeit der Umstände ihrer Entstehung, ihrer Verwendung und ihrer Entsorgung sowie ihrer Zugehörigkeit zu Herstellern, Projekten, Bearbeitern und Versuchen ermöglicht.
7	Standardisierung	Die Dokumentation muß [sic] für alle Proben von gleicher Gestalt sein.
8	Klarheit & Übersichtlichkeit	Die Dokumentation muß so gestaltet sein, daß [sic] Mißverständnisse [sic] und der Informationsverlust durch Unklarheiten vermieden werden.
9	Ordnung	Die Dokumentation muß [sic] systematisch und unabhängig von individuellen Ordnungskriterien erfolgen. Hierzu ist ein zentrales System zu verwenden
10	Zeit	Die Dokumentation muß [sic] der Herstellung, Verwendung und Entsorgung der Probe zeitlich zugeordnet erfolgen und dauerhaft verfügbar sein.

Tabelle 1: Prinzipien des Probendokumentationssystems

Quelle: Fraunhofer ISE Abt. PVM (2012)²²

²² Quelle aus dem Intranet (nicht öffentlich zugänglich) des Fraunhofer ISE Abt. PVM

4.3 Mitarbeiterbefragung

4.3.1 Vorgehen

Zur systematischen Befragung der Mitarbeiter in Interviews und Online-Umfrage wird ein Fragenkatalog (siehe Anhang 1) verwendet. Der erstellte Fragenkatalog ist dabei in drei Kapitel unterteilt. Im ersten allgemeinen Teil wird neben den freiwilligen Angaben, wie Name und Rolle am Institut, nach der Häufigkeit der Dokumentation gefragt. Anschließend teilt sich der Katalog in zwei Teile je nachdem ob die befragte Person bereits das vorhandene Dokumentationssystem verwendet. Abhängig davon wird die befragte Person entweder zur Verwendung des derzeitigen Systems befragt oder zur Begründung der Nicht-Verwendung. Im letzten Teil wird nach den konkreten Wünschen, Vorstellungen und Erwartungen an das neue System gefragt. Die Fragen gestalten sich dabei als eine Mischung aus qualitativen und quantitativen Forschungsmethoden unter Verwendung mehrerer Fragetypen (Offen, geschlossen etc.).

4.3.2 Durchführungsbeschreibung

Die Befragung in Form von Interviews fand dabei persönlich am Institut statt und wurde durch den Teamleiter begleitet. Die Ergebnisse wurden vorerst in den Fragebögen selbst festgehalten und anschließend in einer Tabelle zusammengefasst. Für die Online-Umfrage wurde das vom Fraunhofer ISE intern etablierte Tool LimeSurvey verwendet. Die Online-Umfrage wurde intern verbreitet und anschließend wurden die Ergebnisse ebenfalls in einer Tabelle zusammengefasst. Da nur relativ wenig Personen an der Umfrage teilgenommen haben und die Ergebnisse sehr individuell sind, wurde auf eine statistische Auswertung verzichtet. Zum besseren Verständnis der Anforderungen wurde mit bestimmten Personen persönliche Rücksprache gehalten. Wie in Abschnitt 2.2.4 dargestellt, beruht dies auf den Prinzipien der agilen Softwareentwicklung. Die zusammengefassten Ergebnisse und Rohdaten liegen dieser Arbeit in digitaler Form bei.

4.3.3 Auswertung

Die Auswertung der Befragung nach der Verwendung des derzeitigen Systems ergab, dass trotz klar ersichtlicher Mängel oder fehlender Funktionalität, grundsätzlich doch eine gewisse Zufriedenheit herrscht. Diese ist vor allem auf die Einfachheit und Übersichtlichkeit der Dokumentationstabellen zurückzuführen. Durch die umfangreichen Sortier- und Filterfunktionen von Microsoft Excel lassen sich angelegte Einträge, trotz fehlender Einheitlichkeit, zudem schnell wiederfinden. Auf Basis dessen muss diese unkomplizierte Verwendung im neuen System möglichst beibehalten werden. Um eine Verschlechterung der Bereitstellung durch z. B. fehlende Suchfunktionalität zu vermeiden, soll die Möglichkeit bestehen, bestimmte Datensätze in ein gängiges Format zu exportieren (bspw. *.csv).

Negativ bewertet wurde die Verteilung der Dokumentation auf unterschiedliche Orte und Listen sowie die mangelnde Standardisierung der Erfassung. Besonders das Fehlen von klaren Vorgaben oder Auswahlmöglichkeiten sorgt für einen schwankenden Detaillierungsgrad und zur individuellen Ausprägung der Dokumentation. Des Weiteren wurde die lückenhafte Prozessdokumentation und die fehlende Nachvollziehbarkeit des zeitlichen Ablaufs von Prozessschritten bemängelt. In Folge dessen soll es daher im neuen System möglich sein, ein oder mehr Prozesse einem Planungs-, Herstellungs- oder Durchführungsschritt mit festgelegter Reihenfolge zuzuordnen.

Bei der Befragung nach dem Vorgehen bzw. dem Prozess der Dokumentation konnte eine deutliche Diskrepanz in der Verwendung des derzeitigen Systems zur geplanten Verwendung festgestellt werden. Speziell die angewandten Prozesse bei der Planung, Herstellung oder Durchführung von Proben oder Versuchen zeigten dabei eine sehr individuelle Charakteristik auf. Durch diese enormen Unterschiede in der Prozessgestaltung ließ sich früh ein großes Maß an Flexibilität als eine der Hauptanforderungen erkennen. Um dieser Flexibilität gerecht zu werden, soll es möglich sein verschiedene Vorlagen für

Versuche, Proben und Prozesse zu erstellen, die den jeweiligen Anforderungen entsprechen.

Eine der am häufigsten genannten Anforderungen des neuen Systems war die Zusammenführung der Dokumentationen (Versuche, Prozesse, Materialien ...) in einem zentralen System. Unterstützend wurde dabei nach einer strengeren Kopplung bzw. Zugehörigkeit der Datensätze verlangt. Bspw. soll die Beteiligung einer Probe an einem Versuch oder die verwendeten Materialien bei der Herstellung einer Probe klar ersichtlich sein. Zudem wurde sich für die Auswahl von bestimmten Daten wie z. B. Materialien, Proben- und Versuchstypen oder Anlagen eine klare Standardisierung gewünscht.

4.4 Anforderungsbestimmung

Auf der obersten Ebene der Anforderungsanalyse befinden sich die Visionen und Ziele. Visionen sind in diesem Fall vage formulierte Erwartungen oder Vorstellungen, wobei Ziele eine genauere und vor allem messbare Formulierung des zu erreichenden darstellen.²³

Die Visionen gestalten sich dabei wie folgt:

Das System soll verhindern, dass Proben verloren gehen, nicht identifizierbar sind oder es zu Fehlern durch Verwechslungen kommt – Festlegung der Probenmarkierung.

Zudem soll die Arbeitsqualität erhöht werden und die langfristige Sicherung und den Austausch von Wissen gewährleisten – Dokumentation als solche.

Durch die Dokumentation der Versuche, Proben, Prozesse und Materialien sollen nachgelagerte Arbeiten (z. B. das Schreiben von Berichten) erleichtert werden – Nutzbarmachung.

Außerdem soll das System zur Standardisierung von Prozessen beitragen, mithilfe dessen die Verteilung der Durchführung auf verschiedene Menschen

²³ Vgl. Bergsmann, 2018

ermöglicht werden kann und somit der Verlust von Informationen reduziert wird – Organisationsverbesserung.

Anhand der gewonnenen Erkenntnisse aus Online-Umfrage und den Interviews, sowie der Funktionalität des derzeitigen Systems ergibt sich die Basisfunktionalität. Diese ist in mehreren Iterationen überarbeitet in Form eines Use-Case Diagramms modelliert (siehe Anhang 2). Um genauer auf die einzelnen Anwendungsfälle einzugehen und deren Akzeptanzkriterien zu beschreiben, sind diese in Form von User-Stories beschrieben (siehe Anhang 3). Einige der hier beschriebenen Anforderungen wurden in späteren Iterationen wieder verworfen und sind dementsprechend gekennzeichnet. Bspw. die anfangs oft gewünschte und geplante Anbindung eines Content-Management Systems zur Ablage relevanter Dateien musste aufgrund der somit zwangsläufig verbundenen redundanten Datenspeicherung wieder verworfen werden.

4.5 Aktivitäten

Um der geforderten Flexibilität gerecht zu werden, sind die Hauptaktivitäten innerhalb des Systems als Aktivitätsdiagramme festgehalten (siehe Anhang 4). Diese modellieren den prinzipiell gewünschten Ablauf der Dokumentation (Soll-Szenario). Aufgrund unterschiedlicher Arbeitsweisen oder spontaner Änderungen in früheren Phasen muss allerdings auch ein abweichender Ablauf der Dokumentation gewährleistet sein. Bspw. muss das Hinzufügen eines Durchführungsschritts, der nicht in der Planung berücksichtigt ist, unkompliziert möglich sein (Anpassungsmöglichkeiten für Ist-Szenario).

Der grundsätzliche Ablauf der Erstellung eines Versuchs oder einer Probe ist dabei sehr ähnlich. Im ersten Schritt wird die Planung erstellt, auf Grundlage dessen die Probe hergestellt oder der Versuch durchgeführt wird.

Hierbei müssen vorerst allgemeine Angaben wie Name, zugehöriges Projekt, beteiligte Verantwortliche etc. getroffen werden. Im Anschluss werden in beliebiger Reihenfolge die geplanten Prozesse und Materialien bzw. Proben

definiert. Bei der Planung eines Versuchs werden die Prozesse zusätzlich in einzelne Planungsschritte mit eigener Beschreibung und Status unterteilt.

Das Erstellen oder Bearbeiten eines Prozesses ist in mehrere Bereiche eingeteilt. Diese Bereiche unterteilen sich dabei in allgemeine Angaben, Set-up- und Qualitäts-Parameter. Die genauen Inputs innerhalb der Bereiche werden durch die gewählte Vorlage definiert. Set-up-Parameter sind hier Größen die einen Prozess beschreiben (bspw. Temperaturrampen) und Qualitäts-Parameter sind Ergebnisgrößen (bspw. erzielte Effizienz des Solarmoduls). Die Einrichtung der Verwendung von „Vorlagen“ ist das Ergebnis der Anforderungsanalyse: Benutzern werden standardisierte Eingabemasken bei verringerter Komplexität angeboten. Sie stellen ein wesentliches Feature dar.

Sind die beteiligten Prozesse sowie die Materialien bzw. Proben ausgewählt, so können diese über eine Zuordnungsmatrix in Verbindung gebracht werden. Diese Beteiligung wurde in einer späteren Iteration als vorerst optional definiert.

Danach wird die Durchführung des Plans äquivalent zur Erstellung des Plans dokumentiert, sodass sich eine klare Trennung in Soll- und Istwerte ergibt. Werden ein bereits durchgeführter Versuch oder eine bereits hergestellte Probe dokumentiert, ist es ebenfalls möglich, die Planung zu überspringen.

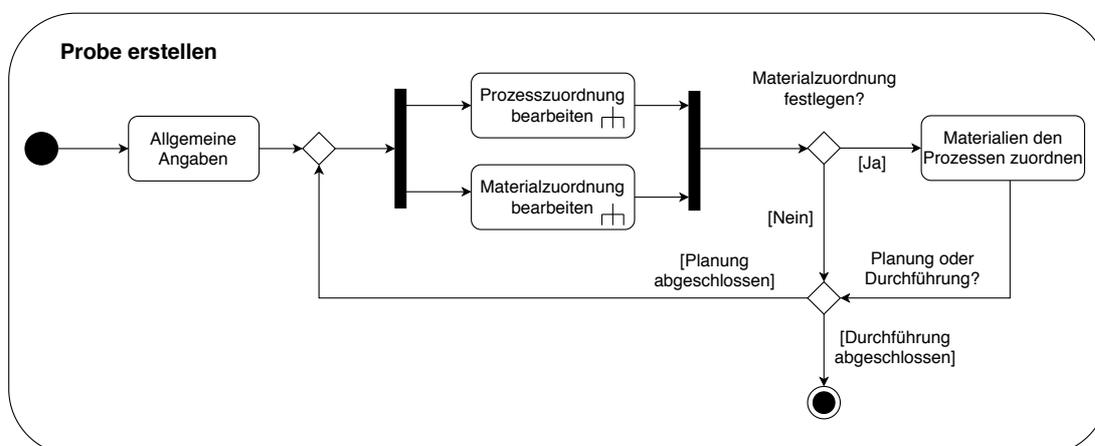


Abbildung 4: Aktivitätsdiagramm Probe erstellen
Quelle: Eigene Darstellung

Zur abschließenden zusammenfassenden Beschreibung sind die Erkenntnisse aus Befragung und den daraus modellierten Modellen in einem Pflichtenheft als Anforderungen formuliert (siehe Anhang 5). Zusätzlich sind die Anforderungen in atomarer Form in einem Anforderungskatalog festgehalten (siehe Anhang 6). Die einzelnen Anforderungen sind dabei nach der MoSCoW Methode (Must have, Should have, Could have, Won't have) priorisiert.

4.6 Analyse der zu erfassenden Daten

Bevor die genaue Datenstruktur in der Konzeptionsphase modelliert und bestimmt werden kann, müssen die beteiligten Daten, deren Beziehungen und Mengenverhältnisse bestimmt werden. Diese ergeben sich, wie die Basisfunktionalität, ebenso aus der Analyse des bereits vorhandenen Systems und der Befragung der Mitarbeiter. Die Dokumentation dieser Anforderungen »[...] umfasst mindestens sowohl eine verbale Beschreibung des Auftrages mit einer klaren Zielsetzung als auch eine Liste der relevanten Informationssachverhalte [...]«²⁴. Diese Beschreibung (siehe Anhang 7) lässt sich anschließend weiter in ein Entitäten-Beziehungsmodell (siehe Anhang 8) konkretisieren.

4.7 Entscheidung für die Eigenentwicklung

Der Markt bietet bereits eine Vielzahl an elektronischen Laborbüchern zur Dokumentation und Planung von wissenschaftlichen Arbeiten, die als Alternative zur Eigenentwicklung eines neuen Systems verwendet werden könnten. Eine Übersicht bzw. Zusammenstellung geeigneter Systeme findet sich bspw. in einem Beitrag von Schöning des Fraunhofer IPA, der die Systeme nach verschiedenen Kriterien wie z. B. Kosten, Strukturierung und Technik vergleicht.²⁵ Auffällig hierbei ist, dass die meisten etablierten Laborsysteme für Anwendungsgebiete aus dem Bereich der Biologie, Chemie oder Medizin konzipiert sind. Oftmals bieten diese zwar die Möglichkeit einer domänenunspezifischen Dokumentation, allerdings besteht hierbei die

²⁴ Meier und Kaufmann, 2016, S. 25-26

²⁵ Vgl. Schöning, o. D.

Gefahr, dass durch die verallgemeinerten Funktionalitäten spezielle Anforderungen vernachlässigt werden könnten. Selbst wenn ein System die speziellen Anforderungen erfüllen könnte, so würde sich ein sehr hoher Konfigurationsaufwand ergeben, der evtl. sogar den Aufwand einer Eigenentwicklung übersteigen könnte. Dieser Nachteil wird durch die Bindung und das vorzubringende Vertrauen an einen womöglich unbekanntem Softwarehersteller verstärkt. Aus Datenschutzgründen müsste das System außerdem intern gehostet werden und somit der technische Verwaltungsaufwand in jedem Fall selbst getragen werden. Auch die Kosten, die bei den meisten Systemen aufgrund von Abo Modellen kontinuierlich anfallen würden, übersteigen die einer Eigenentwicklung durch kostengünstige studentische Arbeitskräfte. Nicht zu vernachlässigen wäre außerdem das verloren gegangene Potenzial für künftige Erweiterungen bzw. Umsetzungen individuell geforderter Funktionen.

Aufgrund dieser Nachteile und der hohen individuellen Charakteristik des domänenspezifischen Anforderungsprofils wurde sich für eine Eigenentwicklung des Dokumentationssystems entschieden.

5. Konzeption

5.1 Benutzeroberfläche

Um eine hohe Benutzerfreundlichkeit des Systems zu gewährleisten und den unterschiedlichen Anforderungen der Mitarbeiter gerecht zu werden, wird für die Gestaltung der Oberfläche das Vorgehen und die Prinzipien der nutzerorientierten Gestaltung verwendet. Der nutzerorientierte Gestaltungsprozess definiert die Miteinbeziehung der Nutzer während des gesamten Vorgehens und zielt zudem auf das Verständnis der Aufgaben, die ein Nutzer innerhalb des Systems hat, ab.²⁶ Um diesem Vorgehen gerecht zu werden, findet während des gesamten Gestaltungsprozess eine starke Kommunikation mit einzelnen Mitarbeitern statt. Nach Abschluss bestimmter Phasen (Wireframes, Klick-Prototyp, Screendesigns, implementierte Version) wird mit verschiedenen Methoden, wie z. B. der Cognitive Walkthrough Methode gearbeitet, um die entwickelten Oberflächen zu validieren. Des Weiteren wird in der ISO Norm 13407 das Prinzip der Überarbeitung des Designs in mehreren Iterationen beschrieben, was kompatibel zur angewandten iterativ inkrementellen Entwicklung ist. Außerdem werden als Teil der Hauptaktivitäten die Erstellung von Designprototypen und die Evaluation des Designs mithilfe dieser beschrieben.²⁷ Auf Grundlage dieser Prinzipien und um den strukturellen Aufbau der Benutzeroberfläche zu erarbeiten, wurde vorerst ein Wireframe Modell inklusive Klick Funktionalität erstellt. Diese Wireframes bilden alle notwendigen Bedienungs- und Darstellungselemente der Anwendung in einer sehr rudimentären Form ab. Mit dieser Darstellung können im Wesentlichen die Strukturierung und die zu durchlaufenden Prozesse bzw. Arbeitsweisen validiert werden. Nutzer können in diesem frühen Prototyp Betriebsdurchläufe simulieren und so Anforderungen an die Abläufe überprüfen.

Das grundsätzliche Layout der Oberfläche kann, wie in Abbildung 5 zu sehen ist, in 4 Bereiche eingeteilt werden.

²⁶ Vgl. Karwowski, Ahram, Etinger, Tanković & Taiar, 2020

²⁷ Vgl. Stone et al., 2005

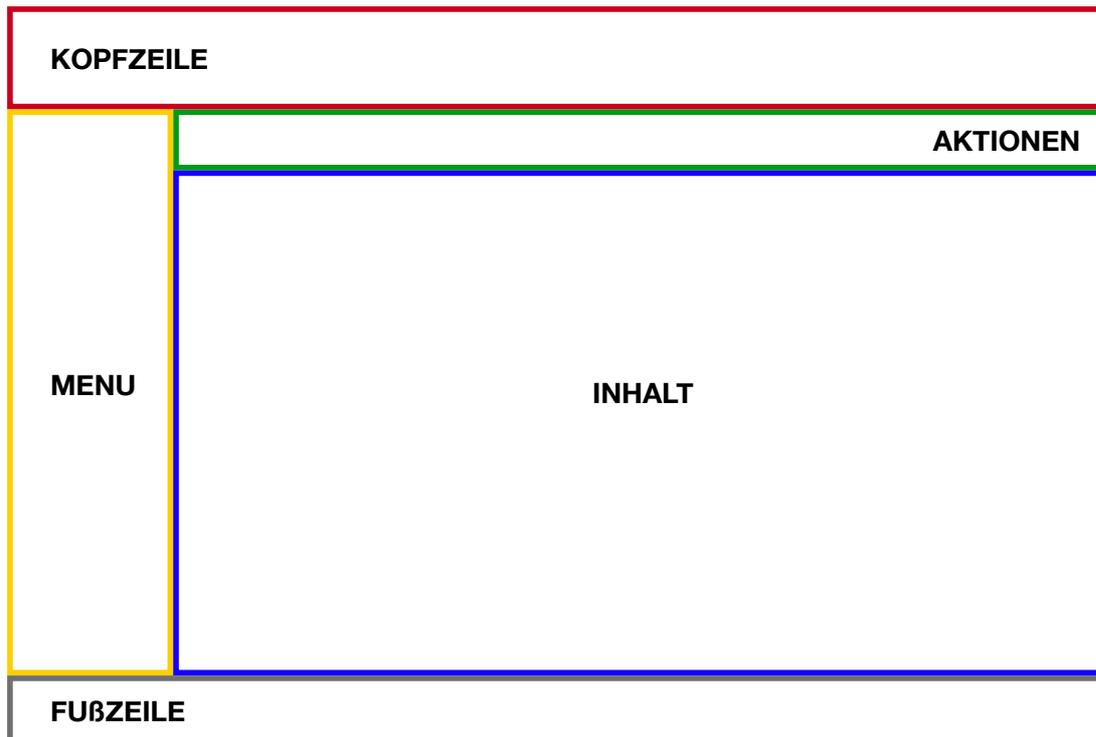


Abbildung 5: Layout Benutzeroberfläche
Quelle: Eigene Darstellung

Diese Einteilung orientiert sich an den grundsätzlichen Gestaltgesetzen, die bspw. von Bühler et al. (2017, S. 30f) thematisiert werden. So werden die einzelnen Bereiche vom Nutzer als inhaltlich und logisch abzugrenzende Gruppen wahrgenommen, indem sie jeweils einen gewissen Abstand zueinander haben (Gesetz der Nähe). Zusätzlich besitzen bestimmte Elemente innerhalb der Bereiche immer das gleiche äußere Erscheinungsbild und verstärken somit den Effekt der Zusammengehörigkeit (Gesetz der Gleichheit). Durch die gleichbleibende Anordnung der Elemente wird dem Nutzer die Bedienung und das Zurechtfinden im System erleichtert (Gesetz der Konstanz)²⁸.

Um die Darstellung mehrerer Ebenen innerhalb der Navigation zu ermöglichen, befindet sich das Menu seitlich und ist somit vertikal und

²⁸ Vgl. Bühler et al., 2017

horizontal skalierbar. Um der Erwartungskonformität, die in der ISO Norm 9241-10 als die Konsistente und der Erfahrung des Nutzers entsprechende Darstellung beschrieben wird²⁹, gerecht zu werden, befindet sich der Aktionsbereich des Systems für jeden Dialog oberhalb des Inhalts. Dieser ermöglicht dem Nutzer die unkomplizierte Interaktion mit dem jeweilig angezeigten Kontext. Bspw. das Erstellen, Bearbeiten oder Löschen eines Eintrags. Ob diese Anordnung auch für die Bedienfelder eines Formulars beibehalten werden soll, da dies evtl. mit der Erfahrung der Nutzer kollidieren könnte, wird im späteren Verlauf genauer untersucht, stellte aber in ersten Usertests kein Problem dar.

Als Folge der Anforderung nach einem einfach verständlichen System, ist die Gestaltung der Navigationsstruktur möglichst wenig verschachtelt und schnell erfassbar aufgebaut. Dies wird erreicht, indem das System grundsätzlich in folgende Ansichten eingeteilt ist: Listenansicht, Detailansicht, Bearbeitungs- und Erstellungsansicht. Zusätzlich gibt es eine Startseite, eine Übersichtsseite und eine Suchseite. Die primäre Navigation bildet alle Eintragstypen und falls diese eine Kategorisierung besitzen, auch deren Kategorien ab. Über die einzelnen Typen (z. B. Probe, Experiment, Material) gelangt man zur Listenansicht, die alle Einträge eines Typs listet. Somit lassen sich die Startseite, die Übersichtsseite und die Listenansicht für alle Einträge von überall aus erreichen.

²⁹ Vgl. Stapelkamp, 2007

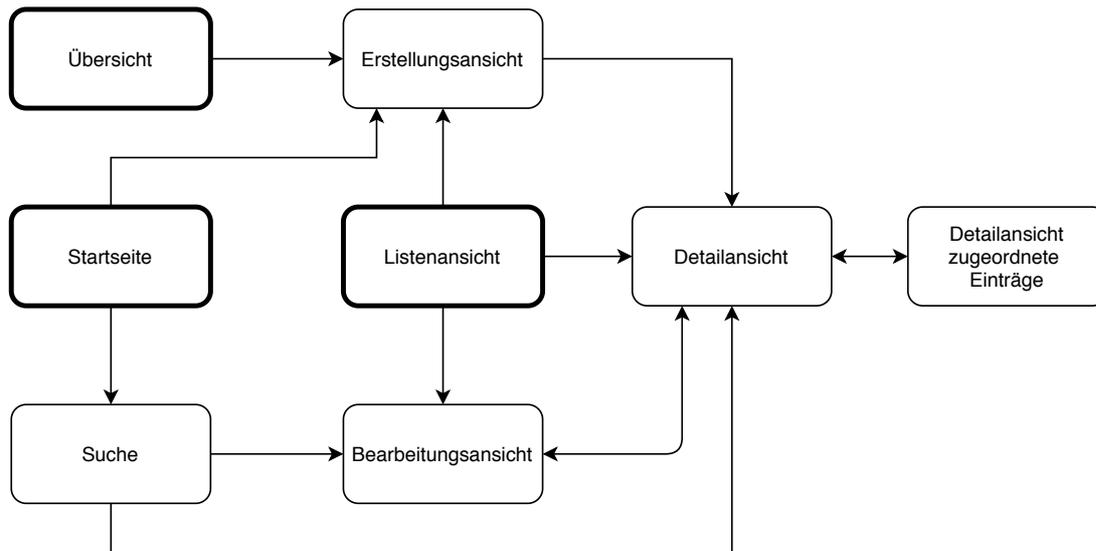


Abbildung 6: Flussdiagramm Navigation

Quelle: Eigene Darstellung

Das als Abbildung 6 eingefügte Flussdiagramm stellt in einer abstrahierten Form die Navigation durch das System dar. Die jeweiligen Ansichten existieren für alle Typen an Einträgen jeweils in leicht abgeänderter Form, je nachdem welche Eigenschaften die Typen besitzen. Die beiden Typen Probe und Experiment sind dabei die komplexesten und besitzen zudem zugeordnete Einträge, die auf jeweilig im Kontext stehende Detailseiten verlinken.



Abbildung 7: Wireframe Listenansicht Experimente
Quelle: Eigene Darstellung

Die Bedienelemente des Wireframes unterscheiden sich lediglich in Ihrer Form und Beschriftung, um die Unterscheidung zwischen Bedien- und reinem Darstellungselement hervorzuheben. Durch diese einheitliche Darstellungsweise soll die Navigation bereits in diesem frühen Stadium der Oberfläche erleichtert werden. Zusätzlich befinden sich bereits vollständige Beschriftungen für die später darzustellenden Informationen in den Wireframes.

Mithilfe der Cognitive Walkthrough Methode wird die Benutzerfreundlichkeit bzw. die Erlernbarkeit des Produkts überprüft. Dabei werden verschiedene Aufgaben und deren Varianten mit einer Gruppe an Versuchspersonen im System durchgeführt. Für jede Aufgabe wird eine Aktionssequenz definiert, mithilfe dessen verschiedene Fragen, wie z. B. "Würde ich als Benutzer wissen, was in diesem Schritt zu tun ist?", diskutiert werden. Die Aufgaben orientieren sich im Falle dieser Arbeit an den Aktivitäten aus der

Anforderungsanalyse, deren Abläufe mit dieser Methode überprüft werden können.³⁰

Die Überprüfung des Konzepts mit der Cognitive Walkthrough Methode ergab, dass der allgemeine Aufbau des Systems die nötige Flexibilität bei der Dokumentation von Proben oder Versuchen ermöglicht bzw. kompatibel mit den unterschiedlichen Arbeitsweisen der Mitarbeiter ist. Eine umstrittene Frage dabei war die Zuordnung der Materialien bzw. Proben direkt zu einem Prozess. Die Zuordnung auf der Detailseite des jeweiligen Prozesses stellte sich als zu kompliziert bzw. zu versteckt heraus. Um dieses Problem zu beheben, wurde die Zuordnung um eine Ebene nach oben direkt zum Eintrag der Probe oder des Experiments verschoben. Um die Beziehung zu den Prozessen wiederherzustellen, wurde sich für eine Matrixdarstellung entschieden. Zusätzlich wurde entschieden, dass die Zuordnung zu Prozessen über die Matrix vorerst optional sein soll, da diese nicht zwangsläufig von Nöten ist.

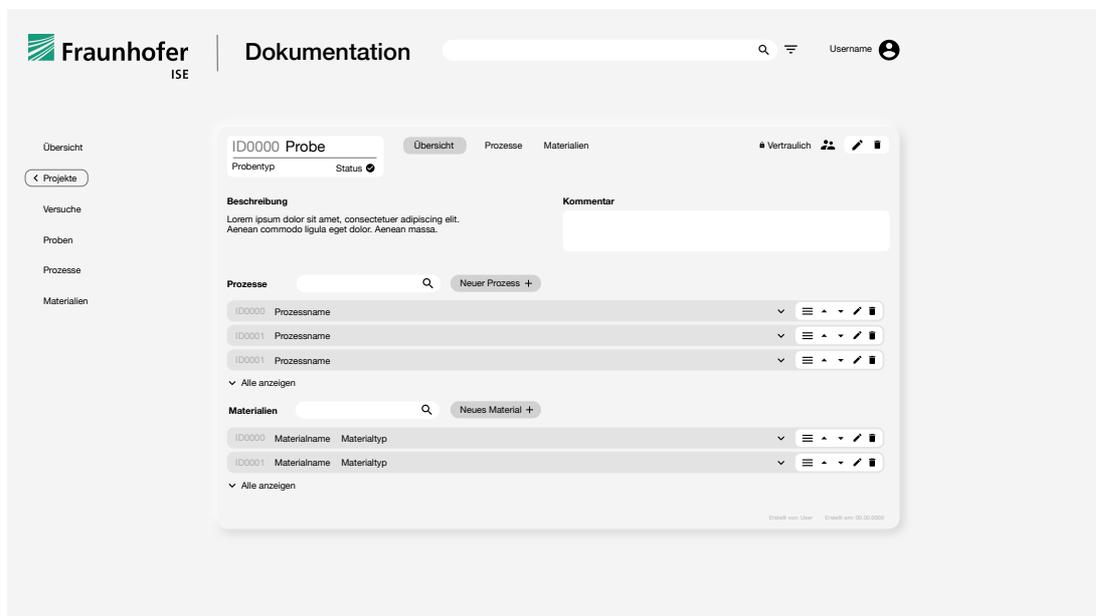


Abbildung 8: Screendesign Probe erster Entwurf
Quelle: Eigene Darstellung

³⁰ Vgl. Wilson, 2013

Ebenfalls iterativ wurde das Corporate Design bzw. die Screendesigns für das System entwickelt. Dieses orientiert sich grundsätzlich an der Assoziation von Dokumentation zu einem Ordner bzw. einer Akte. So stellt die Detailansicht eines Eintrags das Aussehen einer echten Akte nach, indem mehrerer Reiter am oberen Rand sowie ein Label oben links verwendet werden. Diese Darstellung bedient sich der in der Gestaltung von Interfaces vielseitig verwendeten Metapher zur Büroumgebung, dessen Abstrahierung bereits ausreichend ist, um eine intuitive Verwendung des Systems zu verstärken.³¹

Wichtige Allgemeine Informationen werden unter dem Reiter Übersicht der jeweiligen Detailseite des Eintrags dargestellt. Hierzu zählen ID, Name, Verantwortliche, Eintragungsspezifische Informationen und auch eine Vorschau der zugeordneten Einträge. Diese Informationen sind jeweils logisch und grafisch gruppiert, um eine übersichtlichere Darstellung zu erreichen. Eine Liste mit allen zugeordneten Einträgen eines Typs befindet sich unter einem extra Reiter. Da die Darstellung für alle Einträge einheitlich gehalten ist, wird mithilfe einer auf den Corporate Colors basierenden Farbcodierung gearbeitet. Dieses System ordnet jedem Eintrag eine einzigartige Farbe zu. Die Farbcodes stammen aus Fraunhofer ISE intern definierten und kommunizierten Werten. Zusätzlich werden bestimmte Zustände, wie z. B. Planung oder Durchführung mit bestimmten Farben hervorgehoben. Dies ermöglicht dem Benutzer eine leichtere Orientierung im System und verbesserte Abgrenzung zwischen Einträgen oder der Unterscheidung von Planung und Durchführung.

³¹ Vgl. Glaser, Hurtienne & Mohs, 2005

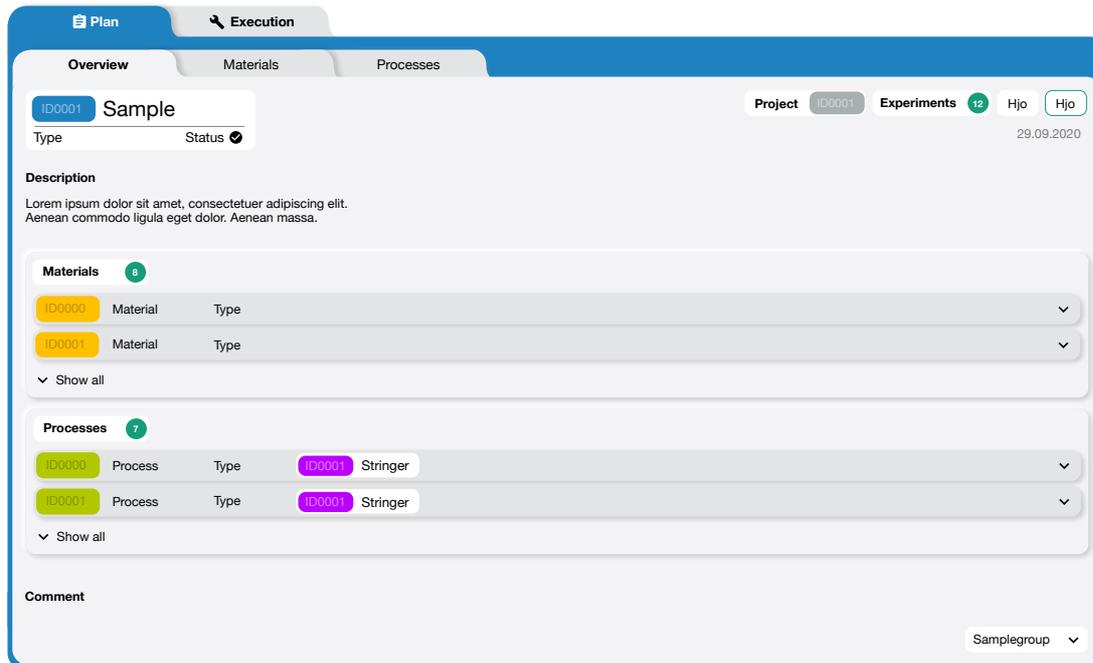


Abbildung 9: Screendesign Detailsseite Probe

Quelle: Eigene Darstellung

Die Listenansicht, die ebenfalls auf den Detailseiten mit ggf. geringerer Anzahl dargestellter Einträge verwendet wird, bildet eine Übersicht über alle vorhandenen Einträge eines Typs ab. Wie bereits auf der Detailseite ist die ID jeweils farblich markiert und dient gleichzeitig als Link zu dieser. Die wichtigsten Informationen eines Eintrags werden direkt innerhalb der Liste in Tabellenform dargestellt. Falls sich unter den Informationen zugeordnete Einträge eines anderen Typs befinden, verlinken diese ebenfalls direkt auf die entsprechenden Detailseiten. Sollten Informationen zu viel Platz in Anspruch nehmen, so werden diese nur teilweise und horizontal scrollbar dargestellt. Am rechten Rand eines jeden Eintrags befinden sich die durchführbaren Aktionen, die zur verbesserten Übersichtlichkeit bis auf den Bearbeiten-Button ausgeblendet sind. Oberhalb der Liste befindet sich eine Schnellsuche, mithilfe dessen Einträge nach bestimmten eintragungsspezifischen Eigenschaften durchsucht werden können.

Frontend Framework React. Im Falle dieses Projekts wird Vue.js anstelle von React verwendet, was aber keinerlei Auswirkungen auf die Funktionalität des Stacks hat. Für die Entwicklung der Benutzeroberfläche wird außerdem die CSS Bibliothek Bootstrap und der CSS Präprozessor SASS verwendet.

Ein zusätzlicher Vorteil der Verwendung der gleichen Technologien ist, dass zukünftige Wartungen oder benötigter Support mithilfe des Know-hows der Nachbarabteilung durchgeführt werden kann.

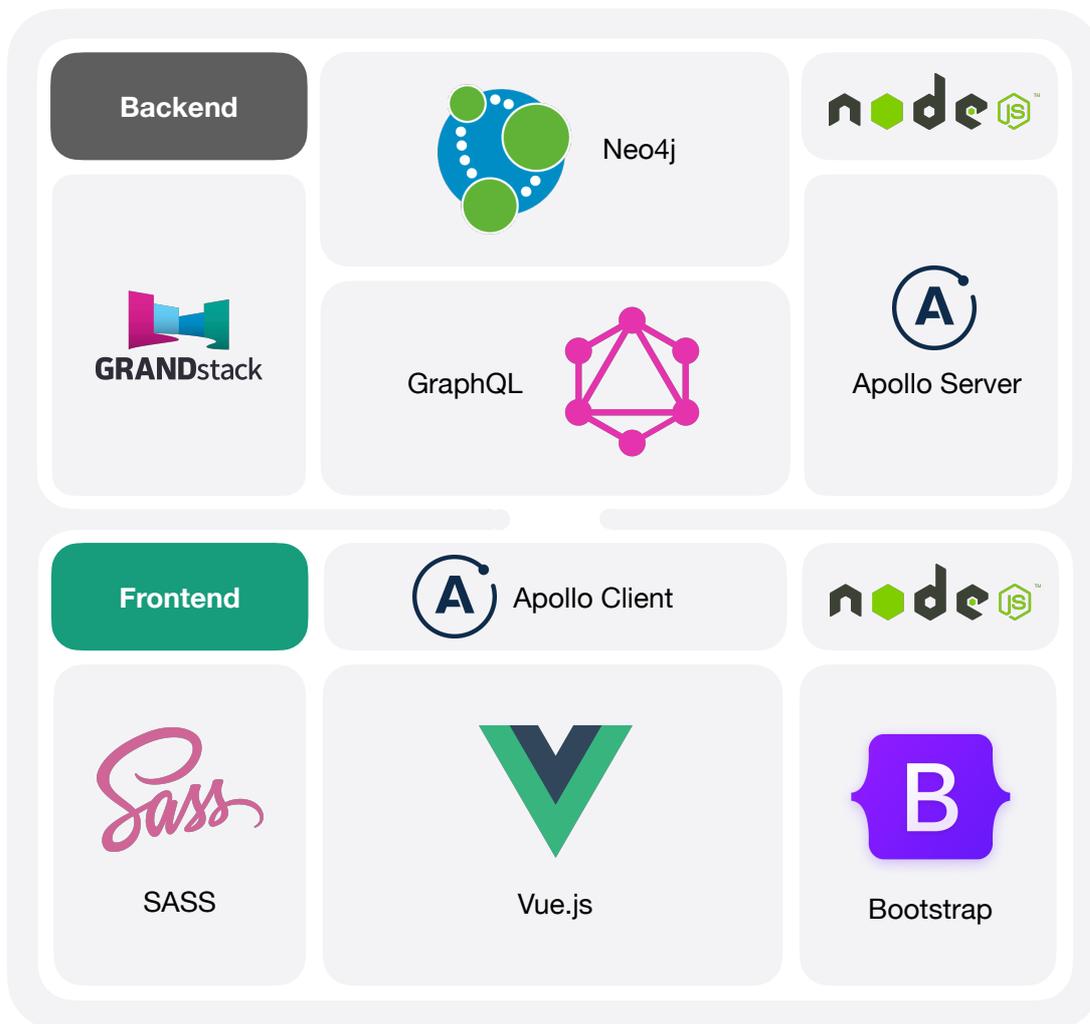


Abbildung 11: Technologien Übersicht

Quellen: <https://neo4j.com>, <https://nodejs.org>, <https://grandstack.io>, <https://graphql.org>, <https://apollographql.com>, <https://sass-lang.com/>, <https://vuejs.org/>, <https://getbootstrap.com/>

5.2.2 Architektur

Für die Bestimmung der allgemeinen Struktur bzw. der Architektur werden verschiedene Diagrammtypen verwendet. Ausgehend von der Datenanalyse, die die verschiedenen Entitäten im System bereits in Beziehung zueinander setzt, wird mithilfe eines Klassendiagramms diese Beziehungsstruktur weiter verfeinert.

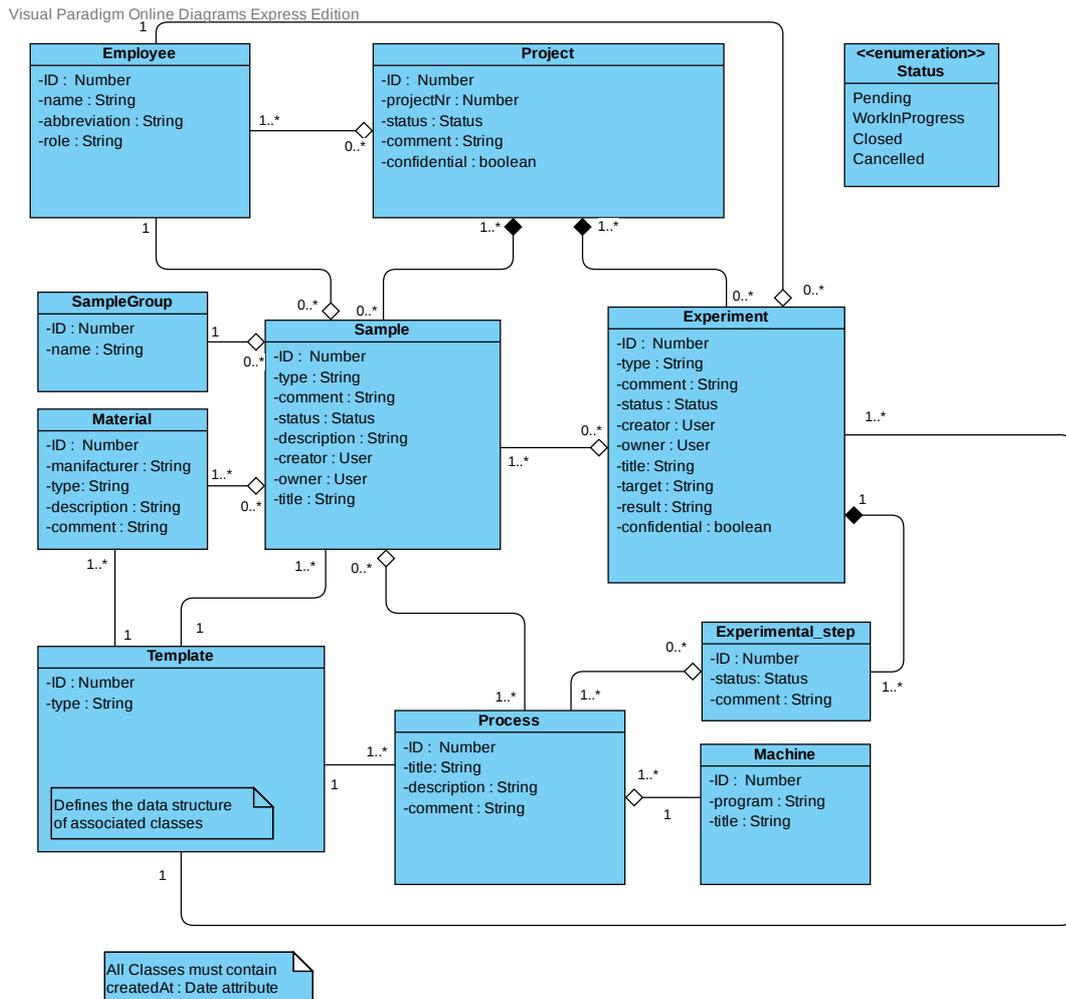


Abbildung 12: Klassendiagramm

Quelle: Eigene Darstellung

In diesem Diagramm sind Proben und Experimente per Komposition Projekten zugeordnet, d. h. diese müssen zwangsläufig immer einem Projekt zugeordnet sein. Das entspricht auch der Arbeitsorganisation am Fraunhofer ISE. Proben können Teil einer Probengruppe sein, mithilfe der sich geringfügig

voneinander abweichende Proben gruppieren lassen (Ordnungskriterium). Des Weiteren lässt sich erkennen, dass Materialien zu Proben und Proben zu Experimenten per Aggregation zugeordnet sind. Diese Beziehung stellt eine optionale Zugehörigkeit dar und definiert die Zusammensetzung einer Probe und die Verwendung dieser innerhalb eines Experiments. Zudem sind durch die gleiche Beziehungsart die Prozesse, die durch Vorlagen definiert werden, an Proben und Experimenten beteiligt. Eine Besonderheit bei der Zuordnung zu Experimenten ist, dass hier die Prozesse noch mal in Experimentschritten logisch gruppiert werden. Equipment, das für die Durchführung von Prozessen notwendig ist, wird dem Prozess direkt zugeordnet.

5.3 Datenstruktur

Ausgehend von der bisherigen Modellierung des Systems und der existierenden Datenstruktur des Chargentools wird eine zu den Anforderungen kompatible Struktur erstellt. Bei der bisherigen Modellierung wurde sich an klassischen Datenstrukturen mit festem Schema und Beziehungen orientiert. Durch den Einsatz einer Graphdatenbank ergibt sich eine neue Ebene an Flexibilität. Aufgrund der fehlenden Einschränkungen bezüglich des Hinzufügens oder Verändern von Datenknoten und Eigenschaften lassen sich diese flexibel und schnell anpassen. Eine genauere Erläuterung zum Thema Graphdatenbanken und speziell Neo4j findet sich in Kapitel 6.1.1.

Da sich der grundsätzlich geplante Aufbau des Systems in weiten Teilen mit der Struktur des Chargentools deckt, wurde die Datenstruktur auf Basis dieser entwickelt.

Aufgrund der hohen Ähnlichkeit im Aufbau von Versuchen und Proben werden diese innerhalb der Datenstruktur unter einem gemeinsamen Knoten namens Experiment verwaltet. Die Unterscheidung findet mithilfe eines Knoten namens ExperimentType statt. Für die Unterteilung in Planung und Durchführung gibt es zur Planung eines Experiments den Knoten PlanningStep, dieser stellt einen geplanten Prozess dar. Wie in Abbildung 13 zu sehen ist, ist das Experiment über den Knoten ProcessPlan mit dem Knoten PlanningStep verbunden. Dieser Zwischenknoten ist für zukünftige Erweiterungen der

Experimenthistorie gedacht. Aufgrund niedrigerer Priorisierung der Anforderung an eine Historie wird dieser allerdings erst in einer späteren Iteration verwendet.

Um die zeitliche Abfolge der Prozesse zu speichern, zeigt ein PlanningStep über eine Beziehung auf den jeweilig nachfolgenden.

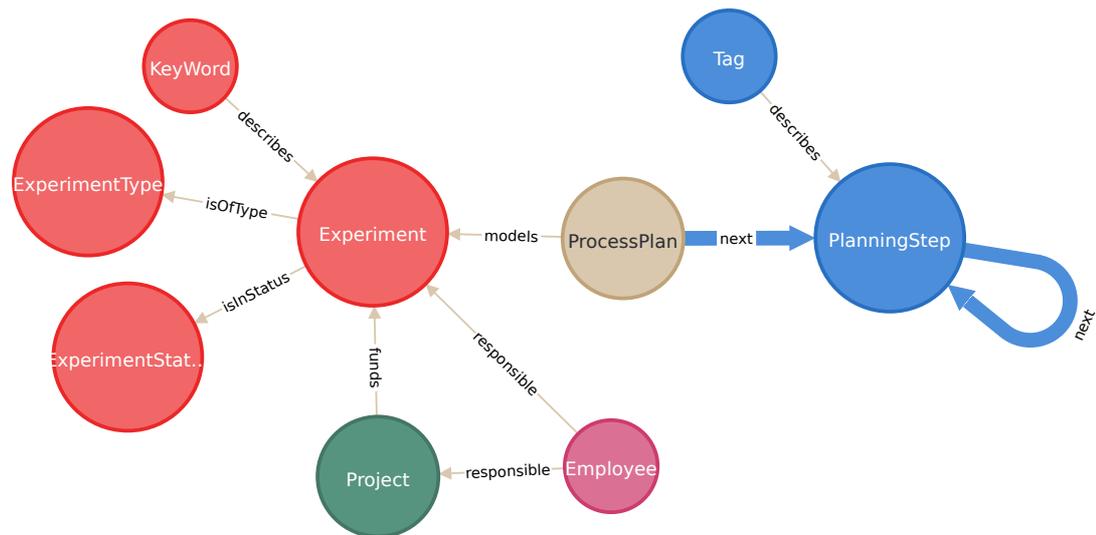


Abbildung 13: Teilgraph Experiment
Quelle: Eigene Darstellung

Einem geplanten Prozess können Materialien und Proben über die PlanningGroup zugeordnet werden. Diese vereint ein oder mehr Materialien oder Proben zu einer logischen Gruppe. In der ersten Version der Implementierung werden diese PlanningGroups vorerst immer nur aus einem Material oder einer Probe bestehen. Eine PlanningGroup kann dabei eine Beziehung zu mehreren PlanningSteps besitzen. Wie in Abbildung 14 von Michael Unger zu sehen ist, lässt sich diese Zuordnung mithilfe einer Matrix visualisieren.

	Planning Group 1	Planning Group 2	Planning Group 3
Initiation Step	X	X	X
Variation Step 1	X		
Variation Step 2		X	
Variation Step 3			X
Finalisation	X	X	X

Abbildung 14: Zuordnungsmatrix

Quelle: Unger, 2018, S.9

Da ein PlanningStep jeweils immer einen Prozess abbildet, werden auch diese durch Prozessvorlagen (Layouts) definiert. Einen wesentlichen Unterschied zum Chargentool (vergleichbares Dokumentationssystem der benachbarten Abteilung) stellen dabei die Qualität- und Set-up-Parameter dar, die durch die Vorlagen definiert werden. Vorlagen sind außerdem durch verwendetes Equipment und eine Kategorisierung spezifiziert. Zusätzlich lassen sich in der Vorlage bereits voreingestellte Set-up-Parameter innerhalb des PlanningSteps überschreiben. Wird bspw. in der Durchführung eines Experiments eine andere Temperatureinstellung als im Set-up-Parameter der Vorlage definiert verwendet, so kann dies ohne Probleme in der Dokumentation des Prozesses abgebildet werden.

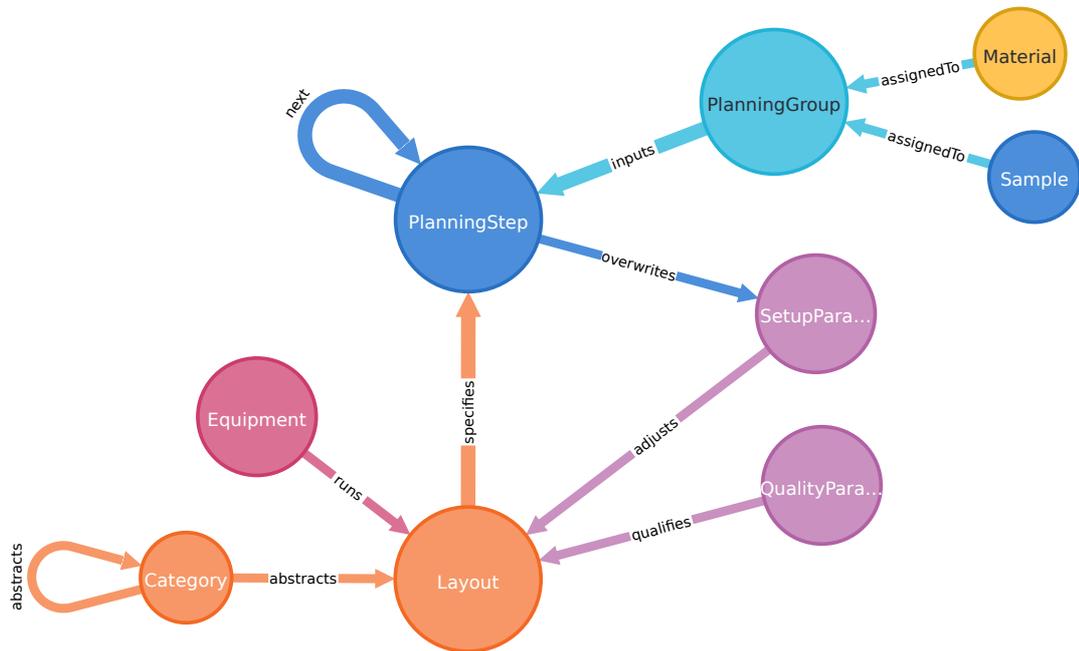


Abbildung 15: Teilgraph Planungsschritt

Quelle: Eigene Darstellung

Um eine Fortschrittskontrolle und die Unterscheidung in Soll- und Istwerte im System zu ermöglichen und die erfolgte Durchführung der geplanten Prozesse zu speichern, wird ein Knoten ExecutionStep in Beziehung zum PlanningStep gesetzt. Dieser ExecutionStep bildet also den ausgeführten Prozess ab und speichert die verwendeten Setupparameter. Wenn nun allerdings mehrere Materialien oder Proben an einem Prozess teilnehmen, lässt sich die Zuordnung dieser zu einzelnen Qualitätsparametern nicht mehr nachvollziehen. Um dieses Problem zu lösen, wird ein Knoten ExecutionUnit als Verbindungsstück zwischen PlanningStep und ExecutionStep verwendet, der die Materialien oder Proben mit den Qualitätsparametern in Verbindung bringt. Eine genauere Darstellung und Erklärung in einem leicht abgewandten Szenario kann in der Arbeit von Michael Unger nachvollzogen werden.

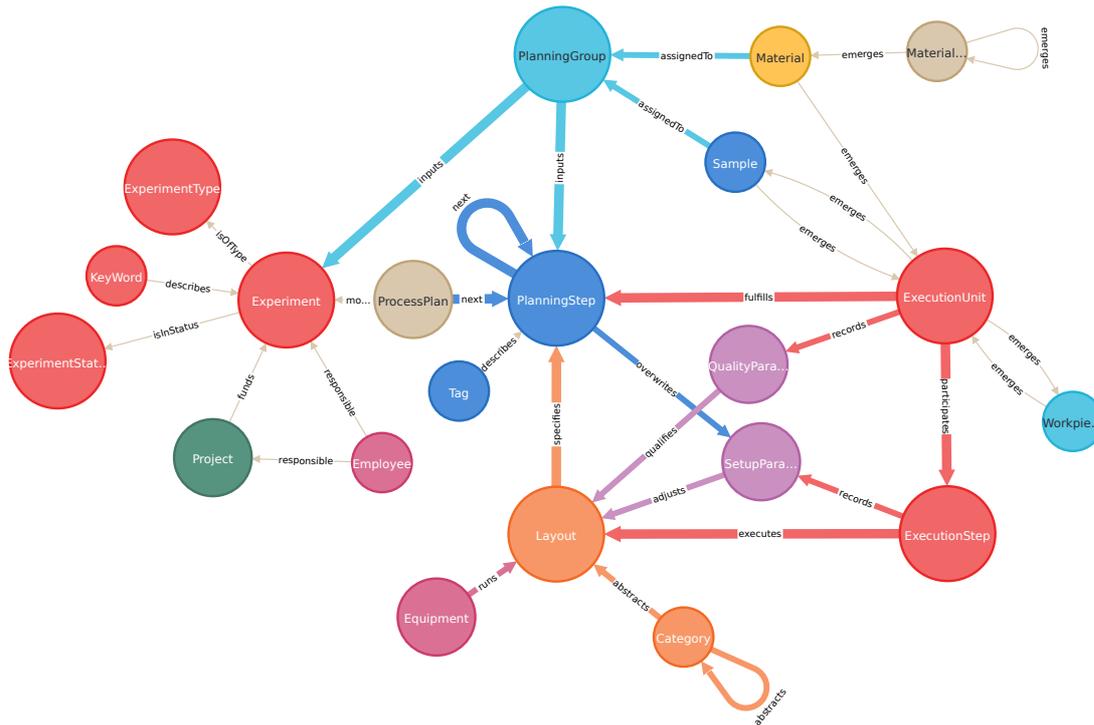


Abbildung 16: Vollständige Datenstruktur als Graph
Quelle: Eigene Darstellung

5.4 Vergleich zur Anforderungsanalyse

Zur Überprüfung, ob durch die vorliegende Konzeption die definierten Anforderungen erfüllt, sowie Schwierigkeiten im existierenden Dokumentationsprozess behoben werden, wird ein Value-Proposition-Canvas verwendet. Bei diesem Modell werden die Merkmale des Wertversprechens in Produkte und Services (Products&Services), den Zugewinn (Gain creators) und die Linderung von Unannehmlichkeiten (Pain relievers) unterteilt. Diese Merkmale werden dem Kundenprofil, das in die Aufgaben (Customer Jobs), dem Streben nach Vorteilen (Gains) und den Hindernissen (Pains) des Kunden unterteilt ist, gegenübergestellt. Wenn anschließend die Vorstellung wie ein Mehrwert für den Kunde erzeugt werden kann, die Hürden und Probleme des Kunden beseitigt, so ist die Konzeption passend.³²

³² Vgl. Osterwalder et al., 2014

Das Value-Proposition-Canvas Modell zeigt, dass mithilfe des neuen Systems einige Umständlichkeiten und Schwierigkeiten des bisherigen Systems behoben werden können. Speziell in Bezug auf Arbeitersparnis und Konsistenz in der Dokumentation kann ein klarer Vorteil generiert werden. Außerdem werden einige neue Funktionalitäten integriert, die ebenfalls zu Erleichterungen im Arbeitsalltag führen sollen. Speziell bisherig abgekapselte Aufgaben der Mitarbeiter werden vom neuen System mit abgedeckt und sorgen somit für eine bessere Vereinheitlichung.

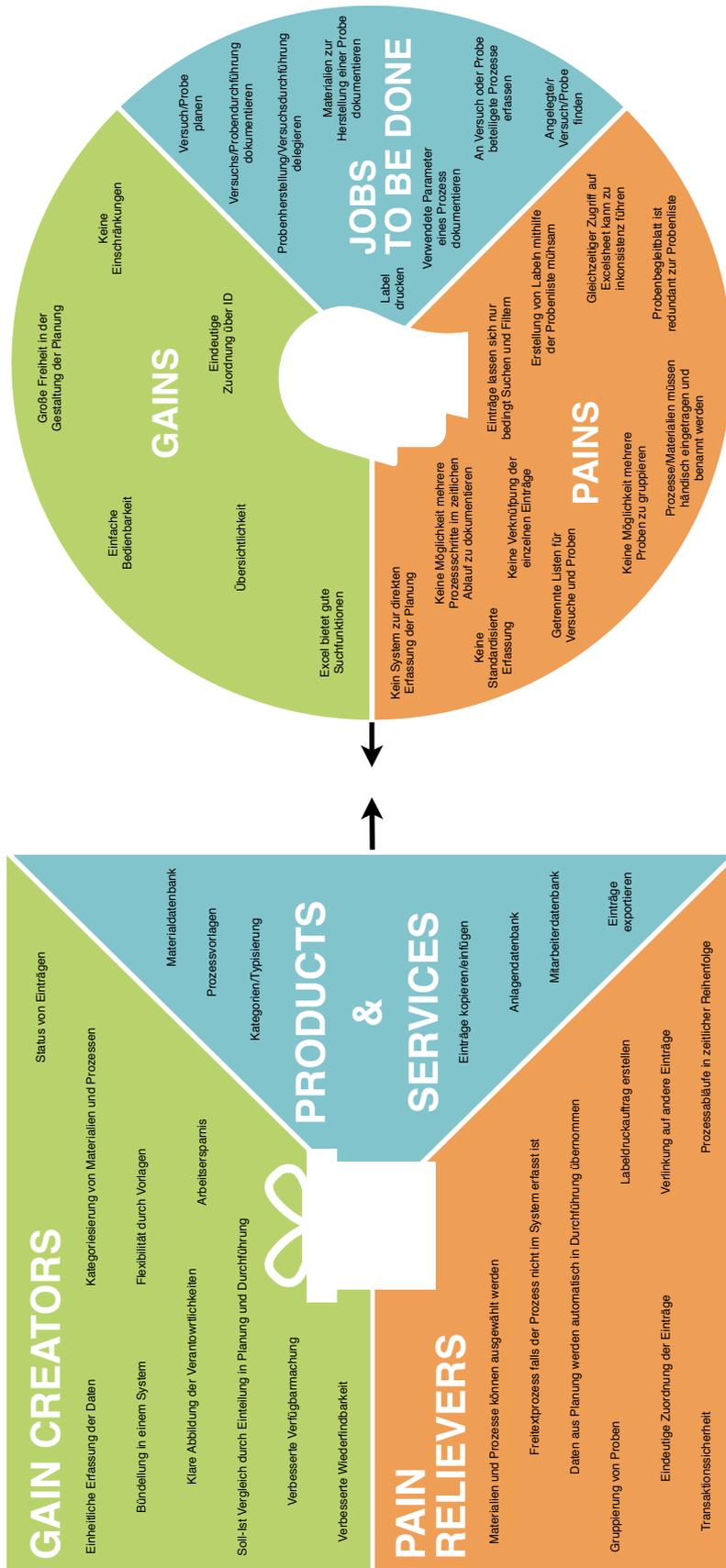


Abbildung 17: Value-Proposition-Canvas
Quelle: Eigene Darstellung

6. Implementierung

6.1 Serverseitig

6.1.1 Datenbank

Zum besseren Verständnis der Implementierung des Backends vorerst eine kurze Einführung in die Funktionsweise von Graphdatenbanken am Beispiel von Neo4j. Fundamental gesehen besteht eine Graphdatenbank aus den Konstrukten Knoten, Beziehung und Eigenschaft. Die Knoten bilden die einzelnen Entitäten ab und werden mit einem Label voneinander unterschieden. Um ihre Charakteristik zu speichern besitzen sie Eigenschaften, die flexibel verwaltet werden können. Beziehungen verbinden Knoten miteinander und können von unterschiedlichem Typ sein. Die Besonderheit hierbei ist, dass Beziehungen eine Richtung aufweisen und ebenfalls Eigenschaften besitzen können. Eigenschaften werden in Form von Schlüssel/Wert-Paaren gespeichert und können unterschiedliche Datentypen annehmen.³³

Neo4j besitzt mit Cypher eine eigene Datenabfrage- und Manipulationssprache mithilfe dessen Datenbanktransaktionen durchgeführt werden können. Cypher ermöglicht es auf eine einfache Art und Weise direkt mit der Datenbank zu interagieren. Cypher Statements lassen sich bspw. direkt in der Kommandozeile der Neo4j Browser Instanz ausführen. Ein einfaches Beispiel eines Cypher Statements sieht wie folgt aus:

```
MATCH (e:Experiment)-[:isOfType]->(eType:ExperimentType{name: "Sample"})
RETURN (e)
```

Quellcode 1: Beispiel einer Cypher Anweisung

Mit der MATCH Anweisung lassen sich Knoten identifizieren, auf die das darauffolgende Muster passt. Innerhalb der runden Klammern wird bestimmt, welchen Typ die Knoten besitzen sollen. Dabei befindet sich links der Doppelpunkte ein frei wählbarer Variablenname und rechts der Labelname

³³ Vgl. Scifo, 2020

des Knoten. Optional kann mit geschweiften Klammern nach bestimmten Eigenschaften gefiltert werden. Im Beispiel werden nur ExperimentType Knoten identifiziert, die für die „name“ Eigenschaft „Sample“ als Wert besitzen. Zwischen den beiden Knoten wird eine Beziehung in ASCII-artiger Darstellung repräsentiert. Hierbei wird nach dem gleichen Schema wie für Knoten ein Variablenname und der Name der Beziehung angegeben. Zudem wird die Richtung der Beziehung angezeigt. In der darauffolgenden Zeile wird mit dem RETURN Befehl bestimmt, welchen Teil des Ergebnisses man erhalten möchte.³⁴ Im Beispiel werden alle Experimentknoten, die eine Beziehung zum Knoten ExperimentType mit dem Wert „Sample“ als „name“ Eigenschaft haben, zurückgeliefert.

6.1.2 Grandstack

Um vom Frontend (Client) aus mit der Datenbank interagieren zu können, benötigt es eine Schnittstelle (API) zum Backend. Diese Schnittstelle definiert, welche Datenbanktransaktionen durchgeführt werden können, wie diese durchgeführt werden und welches Ergebnis an den Client zurückgeliefert werden soll. Da im Falle des Grandstacks diese Schnittstelle in Form eines GraphQL Endpunktes umgesetzt ist, muss vorerst ein GraphQL Schema erarbeitet werden. Dieses Schema bildet die Struktur der Datenbank ab und definiert welche Transaktionen möglich sind.

Da die Datenstruktur dieses Systems auf dem bereits existierenden Chargentool aufbaut, bietet sich eine Funktionalität namens inferSchema des Grandstacks an. Mit dieser Funktion lässt sich aus einer bereits existierenden Neo4j Datenbank automatisch eine GraphQL Schemadatei erzeugen.

³⁴ Vgl. „Getting Started with Cypher - Developer Guides“, o. D.

```
/*
 * Generate a GraphQL schema file out of an existing neo4j instance and
 * save it in a schema.graphql file into the current directory
 * WARNING: If the schema.graphql file already exists it will be
 * overwritten!
 */
inferSchema(driver, schemaInferenceOptions).then((result) => {
  fs.writeFile('schema.graphql', result.typeDefs, (err) => {
    if (err) throw err;
    console.log('Updated schema.graphql');
    process.exit(0);
  });
});
```

Quellcode 2: inferSchema Funktion zur Generierung des GraphQL Schemas

Typischerweise lässt man diese Schemadatei einmal zu Beginn generieren und passt sie anschließend auf die eigenen Bedürfnisse an.³⁵ Dabei gilt es zu beachten, dass alle Änderungen innerhalb der Schemadatei auch in der Datenbank durchgeführt werden müssen. Wird beispielsweise ein neues Attribut für einen Typ definiert, so muss dieses Attribut auch in der Datenbank erzeugt werden, um die Kompatibilität zu gewährleisten.

Im Folgenden ein Auszug aus der generierten und bereits angepassten Schemadatei:

³⁵ Vgl. „Inferring GraphQL Schema From An Existing Database | GRANDstack“, o. D.

```

type Project {
  projectID: ID!
  _id: Int
  projectNr: String!
  created: DateTime
  lastseen: DateTime
  name: String!
  type: String
  comment: String
  confidentiality: Boolean!
  fundsExperiments: [Experiment] @relation(name: "funds", direction: OUT)
  projectManager: Employee
  @cypher(
    statement: """
      match(this)<-[res:responsible{type:'project'}]-(emp:Employee) return
emp
      """
  )
  deputyProjectManager: Employee
  @cypher(
    statement: """
      match(this)<-[res:responsible{type:'deputyProject'}]-(emp:Employee)
return emp
      """
  )
  responsibleEmployees: [Em_Pr_responsible]
}

```

Quellcode 3: Auszug aus der GraphQL Schema Datei (Schema für Projekt)

Anzumerken ist hierbei die Verwendung einer extra Eigenschaft für die ID eines Typs. Die standardmäßig verwendete Kennung von Neo4j ist die Eigenschaft „_id“. Da diese allerdings einige Nachteile mit sich bringt, wie z. B. dass gelöschte Kennungen theoretisch wiederverwendet werden können, empfiehlt es sich, eine eigens definierte Eigenschaft zur Kennung zu verwenden. Die APOC Bibliothek bietet hierfür eine Funktion zur Erstellung einer einzigartigen und eindeutigen Kennung an. Im Falle des Auszugs ist diese in der Eigenschaft „projectID“ gespeichert.

In dem Auszug, der das Schema der Projekte darstellt, lassen sich neben den üblichen GraphQL Typ und Attributdefinitionen auch sogenannte Direktiven verwenden. Mithilfe der Direktive @relation lassen sich über eine Beziehung

zugeordnete Einträge als Attribut definieren. Dabei wird die Beziehung mit ihrem Namen und ihrer Richtung deklariert. Wenn man nun eine Abfrage nach dem Attribut (im Auszug: fundsExperiments) tätigt, so erhält man ein Array mit allen zugeordneten Einträgen (im Auszug: Experimente).³⁶

Eine weitere sehr nützliche Möglichkeit, dem Schema eigene Logik zu geben, ist die `@cypher` Direktive. Diese ermöglicht es eigene Cypher Statements direkt im Schema zu definieren. Bspw. können so die Attribute für die verantwortlichen Mitarbeiter eines Projekts abgefragt werden. Da die Beziehung für Verantwortlichkeiten (im Auszug: responsible) für alle Einträge verwendet wird, wird mithilfe eines Attributs (im Auszug: type) an der Beziehung zwischen den Einträgen unterschieden. Somit kann durch die Deklaration des Beziehungstyps „project“ auf verantwortliche Projektleiter verwiesen werden. Durch diese individuell hinzufügbare Logik lassen sich alle möglichen Arten von Abfragen realisieren. Zudem lassen sich so z. B. Abfragen für Einträge filtern und sortieren. Wichtig hierbei anzumerken ist, dass für die Verwendung der `@cypher` Direktive das APOC Bibliotheks Plugin für Neo4j benötigt wird.

Eine weitere Besonderheit des Grandstack ist die Funktion „makeAugmentedSchema“, mit ihr lässt sich aus der GraphQL Schemadatei ein vollumfänglicher API Endpunkt generieren. Hierbei werden automatisch die Abfragen (queries) und Mutationen (mutations) inklusive der internen Abhandlung (resolver) dieser generiert. Außerdem werden Felder für das Sortieren, Filtern und für die spätere Seitennummerierung angelegt.

Der Funktion kann optional ein Konfigurationsobjekt mitgegeben werden, mithilfe dessen einzelne Typen von der automatischen Generierung ausgeschlossen werden können oder die Erstellung von Abfragen oder Mutationen ganz deaktiviert werden kann.³⁷

³⁶ Vgl. Lyon, 2020

³⁷ Vgl. „GraphQL Schema Generation And Augmentation | GRANDstack“, o. D.

```
/*
 * Create an executable GraphQL schema object from GraphQL type definitions
 * including autogenerated queries and mutations.
 * Optionally a config object can be included to specify which types to
include
 * in generated queries and/or mutations. Read more in the docs:
 * https://grandstack.io/docs/neo4j-graphql-js-
api.html#makeaugmentedschemaoptions-graphqlschema
 */

const schema = makeAugmentedSchema({
  typeDefs,
  config: {
    query: true,
    mutation: {
      exclude: ['Experiment'],
    },
  },
});
```

Quellcode 4: makeAugmentedSchema Funktion zur Generierung des GraphQL Endpunktes

Für alle in der Schemadatei definierten Typen wird dabei eine Abfrage sowie Operationen für das Erzeugen, Aktualisieren und Löschen eines Eintrags erzeugt. Zusätzlich werden Mutationen für das Erzeugen und Löschen von Beziehungen erstellt.³⁸

In der Praxis zeigt sich, dass die automatisch generierten Abfragen und Mutationen zwar eine Arbeitersparnis darstellen, allerdings nur für rudimentäre Datenbanktransaktionen ausreichend sind. So sind sie z. B. geeignet, um alle Einträge eines Typs abzufragen oder einen einfachen Typ zu erzeugen. Soll bspw. aber ein Eintrag mit Beziehungen erstellt werden, so müsste man erst den Eintrag und anschließend die Beziehung erzeugen. Da Datenbanktransaktionen in jedem Fall atomar ausgeführt werden sollten, d. h. ganz oder gar nicht, stellt dies allerdings keine valide Option dar.

Für diesen Fall gibt es die Möglichkeit, eigene Mutationen zu definieren. Dies kann entweder direkt in der Schemadatei mit der @cypher Direktive oder Programmtisch in einem Resolver Objekt der makeAugmentedSchema

³⁸ Vgl. „GraphQL Schema Generation And Augmentation | GRANDstack“, o. D.

Funktion mitgegeben werden. Letzteres ist speziell dafür geeignet, Logik zu definieren, die über den Funktionsumfang von Cypher hinaus geht, bspw. um Daten aus einem anderen System miteinbeziehen oder eigene Validierungsrichtlinien aufzustellen.³⁹

Im Folgenden ein Beispiel für eine eigens definierte Mutation, die das Erzeugen von Projekten mit zugehörigen Abhängigkeiten ermöglicht:

³⁹ Vgl. Lyon, 2020

```
type Mutation {
  CreateProject(
    projectNr: String!
    created: String
    name: String!
    type: String
    comment: String
    confidentiality: Boolean!
    projectManager: ID!
    deputyProjectManager: ID!
  ): Project
  @cypher(
    statement: """
CREATE (p:Project)
SET p += {
  projectID: apoc.create.uuid(),
  projectNr: $projectNr,
  created: datetime($created),
  name: $name,
  type: $type,
  comment: $comment,
  confidentiality: $confidentiality
}
with p
MATCH (emp:Employee{empID: $projectManager})
with p, emp
CREATE (emp)-[res:responsible{type:'project'}]->(p)
with p
MATCH (emp:Employee{empID: $deputyProjectManager})
with p, emp
CREATE (emp)-[res:responsible{type:'deputyProject'}]->(p)
return p
"""
  )
}
```

Quellcode 5: Beispiel einer eigens entwickelten Mutation zur Erzeugung eines Projekts

In diesem Beispiel werden vorerst alle Attribute des Projekts gespeichert und anschließend Beziehungen zu den verantwortlichen Mitarbeitern erzeugt. Dies ermöglicht es, ein Projekt mit nur einer Datenbanktransaktion zu erstellen und somit die vollständige Durchführung oder Nicht-Durchführung der Transaktion

zu gewährleisten (Atomarität)⁴⁰. Auf gleiche Art und Weise sind für alle anderen Typen, die Beziehungen besitzen, jeweils Create, Update und Delete-Mutationen definiert.

6.2 Clientseitig

Wie bereits in Abschnitt 5.2.1 beschrieben, wurde das Frontend mithilfe des JavaScript Framework Vue.js entwickelt. Da dies ein komponentenbasiertes Framework ist, im Folgenden ein Überblick über die allgemeine Struktur und die Zusammensetzung der Komponenten innerhalb der Anwendung.

Die Komponente `App.vue` bildet den Einstiegspunkt der Anwendung und beinhaltet die Komponenten, die den grundsätzlichen Aufbau der Oberfläche definieren. Dieser Aufbau lässt sich mit einer Baumstruktur vergleichen, bei der die `App.vue` Komponente die Wurzel darstellt und alle weiteren Komponenten davon abgeleitet sind. Am oberen Rand der Benutzeroberfläche befindet sich die `TopBar` Komponente, die das Logo und eine Suchleiste für die Detailsuche beinhaltet. Am linken Rand befindet sich die `NavBar` Komponente, die die Navigation abbildet. Sie beinhaltet eine `NavItem` Komponente, die benötigt wird, um mehrere Ebenen innerhalb der Navigationsstruktur dynamisch darzustellen. Dies wird erreicht, indem die `NavItem` Komponente sich selbst beinhaltet und somit rekursiv aufruft, solange der derzeitige Navigationspunkt weitere Unterpunkte besitzt. Mittig befindet sich der Inhaltsbereich, welcher mithilfe des Vue-Routers die jeweilig aktuell ausgewählte Seite darstellt. Am unteren Rand befindet sich die `FooterBar` Komponente zur Darstellung der Fußleiste. Die Anordnung der einzelnen Komponenten geschieht dabei mithilfe des Bootstrap-Grids.

⁴⁰ Vgl. Heuer, Saake, Sattler, Grunert & Meyer, 2020

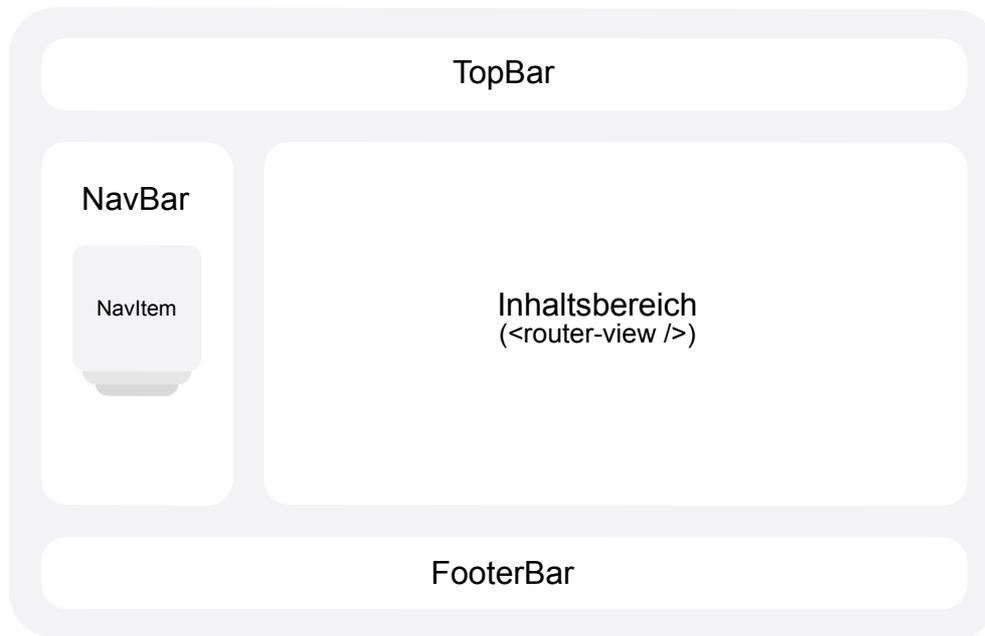


Abbildung 18: Grundsätzlicher Aufbau des Frontends
Quelle: Eigene Darstellung

Für alle Eintragstypen gibt es die Listenansicht, auf die man über den jeweiligen Navigationspunkt gelangt. Die Angabe, welcher Navigationspunkt auf welche Komponente verlinkt wird dabei über die Datei `index.js` im Verzeichnis „`src/router`“ verwaltet. Jeder Typ besitzt eine `List<typbezeichnung>` Komponente, dessen Aufbau jeweils relativ ähnlich und im „`src/views`“ Verzeichnis gespeichert ist. Innerhalb der Komponenten werden die jeweilig eintragungsspezifischen Daten, wie z. B. welche Eigenschaften in der Liste angezeigt werden sollen oder nach welchen Eigenschaften gesucht werden kann, in der `data` Funktion angegeben. Bei der Erzeugung, also sobald eine der Komponenten angezeigt werden muss, wird mithilfe der Apollo `SmartQuery` Funktionalität eine Anfrage an den Server geschickt, der die jeweiligen Entitäten des Typs zurückliefert. Diese Daten werden anschließend automatisch in einem Array aus Objekten ebenfalls in der `data` Funktion gespeichert.

Zur Darstellung der Tabelle bzw. Liste der Entitäten werden die Komponenten `TableTitle`, `TableSearch`, `TableItems` und `ListPagination` verwendet. Damit die Einträge auf mehrere Seiten verteilt, sortiert und gefiltert sind, werden die

Daten durch verschiedene Funktionen geschickt und anschließend an die ListPagination Komponente weitergegeben. Diese speichert die auf der aktuellen Seite befindlichen Einträge in einer Variable namens pageOfItems, die anschließend an die TableItems Komponente zur Darstellung übergeben wird.

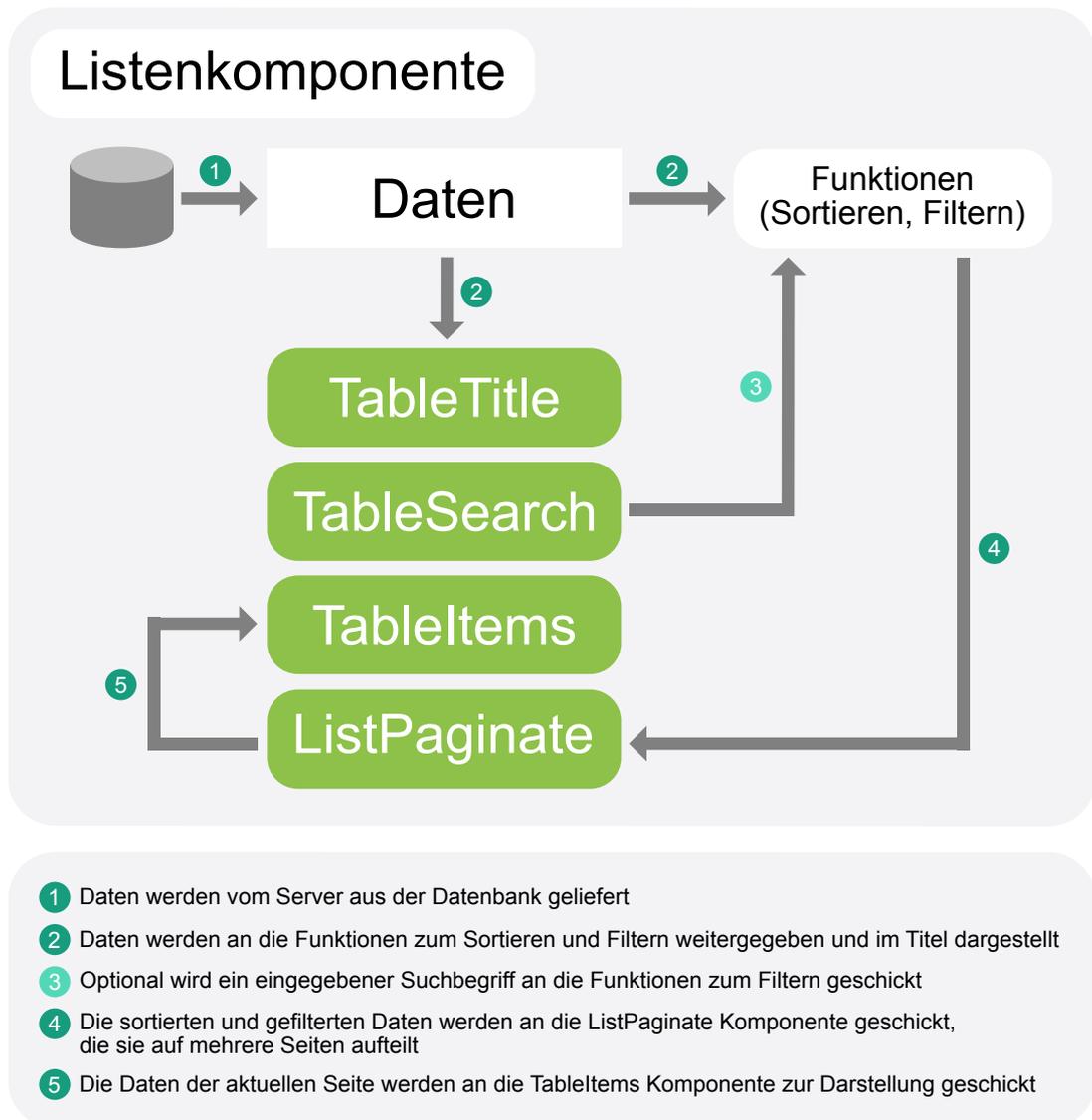


Abbildung 19: Listenkomponente Aufbau & Ablauf

Quelle: Eigene Darstellung

Das Sortieren und Filtern der Einträge geschieht in diesem Fall clientseitig, was den Vorteil hat, sehr schnell Ergebnisse zu liefern und somit ein

reaktionsschnelles System suggeriert. Die Skalierung sollte in diesem Umfeld vorerst unproblematisch sein.

Um auf die Detail- und Bearbeitungsseite der jeweiligen Einträge zu gelangen, werden der TableItems Komponente zwei Links mitgegeben, die mithilfe des Vue-Routers die ID des jeweiligen Eintrags beinhalten. Über diese ID können innerhalb der Komponenten der Detail- und Bearbeitungsseite ähnlich wie bei der Listenkomponente zu Beginn die Daten vom Server abgefragt werden und somit dynamisch der jeweilige Eintrag dargestellt werden.

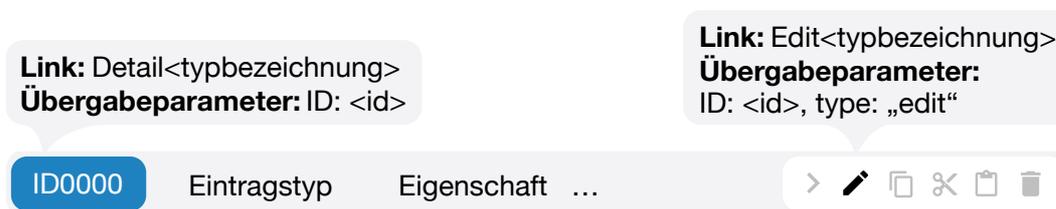


Abbildung 20: Listenkomponente Verlinkungen
Quelle: Eigene Darstellung

Die Bearbeitungsseiten bzw. Komponenten gestalten sich für die Basisdaten wieder relativ ähnlich und werden im Verzeichnis „src/components/baseData/<typbezeichnung>/<Typbezeichnung>Form“ verwaltet. Da sich das Erstellen und das Bearbeiten eines Eintrags auf Ebene der Oberfläche nur darin unterscheidet, ob bereits Eigenschaften existieren oder nicht, können die Bearbeitungskomponenten ebenfalls für das Erstellen von Einträgen verwendet werden.

Für diese Unterscheidung wird dem Link, der zu der jeweiligen Seite führt, zusätzlich ein Parameter namens „type“ mitgegeben, der für die Erstellung den Wert „add“ und für die Bearbeitung den Wert „edit“ besitzt. Mithilfe dieses Parameters können für den Fall, dass es sich um das Bearbeiten eines Eintrags handelt, die Eingabefelder des Formulars mit den Eigenschaften des Eintrags befüllt werden. Außerdem werden beim Bearbeiten eines Eintrags

neu hinzugefügte oder abgeänderte Eingaben farblich hervorgehoben, indem die Beschriftung des Eingabefelds blau markiert wird.

Edit Sample

Submit ✓ Cancel ✕ Reset changes ↺

General Materials Processes

Sample-Nr. * 000001 Name * Probenherstellung für Vallis Sloaris

Status * Done Project * 000001 Confidentiality

Sample Manager * Henri Joß Dep. Sample Manager * Max Mittag

Description 350 Watt Testmodule

Comment Dies ist ein neu hinzugefügter Kommentar

Abbildung 21: Bearbeitungsansicht am Beispiel einer Probe
Quelle: Eigene Darstellung

Zum Speichern oder Abbrechen neu erstellter oder bearbeiteter Einträge wird die Komponente FormControls verwendet. Sollte es beim Abschicken des Formulars zu Validierungsfehlern kommen, bspw. wenn die Projektnummer bereits vergeben ist, so wird das über die betroffene Beschriftung farblich rot markiert dargestellt.

Für die komplexeren Typen Experiment und Probe besteht die Erstellungs- bzw. Bearbeitungsansicht aus einer Create<Typbezeichnung> Komponente, die drei weitere Komponenten beinhaltet. Diese drei Komponenten bestehen aus einem allgemeinen Formular (ähnlich wie bei den Basisdaten), einer Komponente zur Zuordnung der Materialien bzw. Proben und einer Komponente zur Zuordnung und Erstellung von Prozessen.

Nach erfolgreicher Erstellung oder Bearbeitung eines Eintrags gelangt man auf die Detailsseite des Eintrags. Hier werden neben den allgemeinen

Informationen über den Eintrag auch alle Abhängigkeiten bzw. Zugehörigkeiten mithilfe der TableItems Komponente dargestellt.

The screenshot shows the ELISE web application interface. At the top left is the ELISE logo. A search bar and a filter button are at the top right. The main content area is titled 'Sample' and features a navigation sidebar on the left. The sidebar includes links for Home, Overview, Samples, Experiments, Processes, Base-Data, Projects, Materials, Employees, Equipment, and Layouts. The main content area has tabs for 'General', 'Materials', and 'Processes'. The 'General' tab is active, showing the sample ID '000001', name 'Minimodul', and status 'done'. It also displays a description '350 Watt Testmodule' and a comment 'Handle with care!'. Below this, there are two tables: 'Materials' with 3 items and 'Processes' with 6 items. The Materials table lists Silicone 1, Müller Weissglas 3.2, and EVA. The Processes table lists EL-Messung, Lamination, and Stringen Plus.

ID	Name	Description	Manufacturer	Timestamp
MATERIAL	Silicone 1	Glued between glas and reflector	Manufacturer 1	17.2.2021
MATERIAL	Müller Weissglas 3.2	200 x 200 x 3 mm		28.1.2021
MATERIAL	EVA			28.1.2021

ID	Name	Description	Timestamp
PROCESS	EL-Messung	Eine EL-Messung am Modul unter STC Bedingungen	21.2.2021
PROCESS	Lamination	Lamination Standard 60 Zeller	21.2.2021
PROCESS	Stringen Plus	Stringen mit für Halbzellen PLUS Inspektion	21.2.2021

Abbildung 22: Probedetailsseite implementierte Version
Quelle: Eigene Darstellung

Der gesamte Quellcode (Client- und serverseitig) liegt dieser Arbeit in digitaler Form bei. Beide Teile wurden mit der Versionsverwaltungssoftware git erstellt und beinhalteten eine grobe Anleitung zur Installation und Einrichtung in der „README.md“ Datei. Zusätzlich befindet sich im Server Verzeichnis ein Speicherauszug der Datenbank, der mit Testdaten befüllt ist.

7. Schlussteil

7.1 Zusammenfassung

Die grundlegende Intention dieser Arbeit war es, ein neues und innovatives Dokumentationssystem für die Abteilung PVM des Fraunhofer ISE zu schaffen. Um dies zu erreichen, musste im ersten Schritt ein auf die besonderen Bedingungen dieser Arbeit angepasstes Vorgehensmodell gewählt werden.

Anschließend wurde eine umfangreiche Anforderungsanalyse erstellt, um frühzeitig klare Ziele der Dokumentation und dessen praktischer Durchführung der Abteilung zu erarbeiten. Diese Anforderungen konnten im weiteren Verlauf der Entwicklung durch stetige Kommunikation und Validierung der einzelnen Ergebnisse weiter verfeinert und angepasst werden.

Auf Basis der gewonnenen Erkenntnisse wurde zusammen mit ausgewählten Mitarbeitern eine technische und gestalterische Konzeption des Systems entworfen. Dieser Entwurf wurde nach den Prinzipien der nutzerorientierten Gestaltung in mehreren Iterationen validiert und weiter verbessert.

Ausgehend von der konzeptionellen Arbeit konnte eine technische Umsetzung geplant und durchgeführt werden. Dabei mussten die passenden Technologien gewählt und eine Implementierung auf Client- und Serverseite erstellt werden.

Das dabei entstandene Softwareprodukt bildet die Grundlage für ein langfristig einsetzbares und erweiterbares System.

7.2 Ergebnisse

Mit der iterativ inkrementellen Entwicklungsweise zusammen mit den Prinzipien der agilen Softwareentwicklung wurde eine flexible Herangehensweise gewählt, die durch die mehrfache Wiederholung der einzelnen Phasen frühzeitig etwaige Probleme erkennen ließ und mithilfe der regen Kommunikation und Interaktion schnell zu Lösungsansätzen führte. Durch diese Wahl konnte sichergestellt werden, dass sich das System in

einem konsistenten Zustand befindet und die nahtlose Weiterentwicklung und Wartung problemlos möglich sind.

Aus der Anforderungsanalyse ging hervor, dass die Flexibilität, Einfachheit und Einheitlichkeit im Dokumentationsprozess die wichtigsten Anforderungen darstellen. Um diesen gerecht zu werden und gleichzeitig die definierten Prinzipien guter Dokumentation einhalten zu können, wurde ein wenig restriktives, aber gezielt unterstützendes System konzipiert. Dabei stellt die Verwendung von Vorlagen für die unterschiedlichen Einträge ein wesentliches Ergebnis für die konsistente und qualitativ hochwertige Dokumentation dar.

Indem die Nutzer eine wichtige Rolle in der Konzeption des Oberflächendesigns gespielt haben, konnte eine hohe Passgenauigkeit der allgemeinen und individuellen Anforderungen in Design und Struktur erreicht werden. Die Überprüfung der Prozessabläufe innerhalb des Systems mithilfe der Cognitive Walkthrough Methode zeigte auf, dass die vorliegende Konzeption den notwendigen Aufbau und die nötige Flexibilität für eine hohe Nutzerakzeptanz bzw. Zufriedenheit bietet.

Auf Basis einer existierenden Datenstruktur wurde ein Modell entwickelt, das das konzipierte System in technischer Form ermöglicht. Eine wesentliche Erweiterung stellte dabei die Einführung von parametrisierten Vorlagen in der Datenstruktur dar. Zudem wurden eigens entwickelte Abfragen und Manipulationen zur Interaktion mit der Datenbank erarbeitet, die die Schnittstelle zum Frontend darstellen. Für das Frontend wurde eine hierarchische Architektur aus Komponenten entwickelt, die durch logische Aufteilung der Implementierung eine hohe Verständlichkeit und die Möglichkeit für Erweiterungen schafft.

7.3 Ausblick

Obwohl sich das System in einem lauffähigen und konsistenten Zustand befindet, konnten innerhalb des Zeitrahmens der Bachelorarbeit nicht die notwendigen Tests zur Überprüfung der Einsatzfähigkeit erbracht werden. Da die Entwicklung des Systems allerdings nahtlos fortgeführt wird, können diese

zeitnah durchgeführt und eine erste Version in der Abteilung eingeführt werden.

Die Überprüfung, inwiefern eine vereinheitlichte und standardisierte Dokumentation die Effektivität und Effizienz im Arbeitsalltag steigern kann, kann erst nach Etablierung und einer gewissen Verwendungszeit überprüft werden und bietet somit Potenzial für zukünftige Arbeiten.

Der derzeitige Stand des Systems bildet eine erste Version ab und beinhaltet die grundlegend notwendigen Funktionen für den Dokumentationsprozess. Anforderungen und Ideen für zukünftige Erweiterungen sind bereits in dieser Arbeit enthalten und können im Laufe der Weiterentwicklung implementiert werden.

Literaturverzeichnis

- Alshamrani, A. & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1), 106–111. Abgerufen von <https://sijal-technology.com/images/company/86b122d4358357d834a87ce618a55de0.pdf>
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development. Abgerufen am 15. September 2020, von <https://agilemanifesto.org/>
- Berg, B., Knott, P. & Sandhaus, G. (2014). *Hybride Softwareentwicklung: Das Beste aus klassischen und agilen Methoden in einem Modell vereint (Xpert.press) (German Edition)* (2014. Aufl.). <https://doi.org/10.1007/978-3-642-55064-5>
- Bergsmann, J. (2018). *Requirements Engineering für die agile Softwareentwicklung: Methoden, Techniken und Strategien*. Abgerufen von <https://learning.oreilly.com/library/view/requirements-engineering-fur/9781492067702/?ar>
- Brandt-Pook, H. & Kollmeier, R. (2020). *Softwareentwicklung kompakt und verständlich: Wie Softwaresysteme entstehen (German Edition)* (3., verb. Aufl. 2020 Aufl.). <https://doi.org/10.1007/978-3-658-10876-2>
- Bühler, P., Schlaich, P. & Sinner, D. (2017). *Visuelle Kommunikation: Wahrnehmung - Perspektive - Gestaltung (Bibliothek der Mediengestaltung)* (1. Aufl. 2017 Aufl.). https://doi.org/10.1007/978-3-662-53770-1_3
- Fraunhofer. (o. D.). Abgerufen am 20. Oktober 2020, von <https://www.fraunhofer.de/>

Fraunhofer ISE. (o. D.). Abgerufen am 20. Oktober 2020, von

<https://www.ise.fraunhofer.de/>

Fraunhofer ISE Abt. PVM (2012). Anforderungen und Prinzipien der Probendokumentation [Organisationsanweisung]

Gaus, W. (2005). *Dokumentations- und Ordnungslehre: Theorie und Praxis des Information Retrieval (eXamen.press) (German Edition)* (5., überarb. Aufl. 2005 Aufl.). <https://doi.org/10.1007/3-540-27518-5>

Getting Started with Cypher - Developer Guides. (o. D.). Abgerufen am 20. Januar 2021, von <https://neo4j.com/developer/cypher/intro-cypher/>

Glaser, C., Hurtienne, J. & Mohs, C. (2005). Intuitiv und effizient –Thesen zur Intuitivität von User Interfaces. In: Hassenzahl, M. & Peissner, M. (Hrsg.), Tagungsband UP05. Stuttgart: Fraunhofer Verlag. (S. 146-153). Abgerufen von <https://new-dl.gi.de/handle/20.500.12116/5843>

GraphQL Schema Generation And Augmentation I GRANDstack. (o. D.). Abgerufen am 20. Januar 2021, von

<https://grandstack.io/docs/graphql-schema-generation-augmentation/#pagination>

Heuer, A., Saake, G., Sattler, K., Grunert, H. & Meyer, H. (2020).

Datenbanken: Kompaktkurs (German Edition) (1. Aufl.). Abgerufen von <https://learning.oreilly.com/library/view/datenbanken-kompaktkurs/9783958457843/>

Hoffmann, D. W. (2008). *Software-Qualität (eXamen.press) (German Edition)* (1. Aufl.). <https://doi.org/10.1007/978-3-642-35700-8>

Inferring GraphQL Schema From An Existing Database I GRANDstack. (o. D.). Abgerufen am 19. Januar 2021, von

<https://grandstack.io/docs/infer-graphql-schema-database/>

International Requirements Engineering Board (IREB). (2017). A Glossary of Requirements Engineering Terminology. Abgerufen am August 2020, von https://www.ireb.org/content/downloads/2-cpre-glossary-2-0/ireb_cpre_glossary_en_2.0.pdf

Jorgensen, P. C. (2018). *Software Testing: A Craftsman's Approach, Fourth Edition* (4. Aufl.). Abgerufen von <https://learning.oreilly.com/library/view/software-testing-4th/9781466560680/>

Karwowski, W., Ahram, T., Etinger, D., Tanković, N. & Taiar, R. (2020). Human Systems Engineering and Design III: Proceedings of the 3rd International Conference on Human Systems Engineering and Design (IHSED2020): Future ... Intelligent Systems and Computing, Band 1269) (1st ed. 2021 Aufl.). <https://doi.org/10.1007/978-3-030-58282-1>

Kleuker, S. (2018). *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten (German Edition)* (4. Aufl. 2018 Aufl.). <https://doi.org/10.1007/978-3-8348-9843-2>

Lyon, W. L. (2020). *Fullstack GraphQL Applications with GRANDstack Essential Excerpts*. Abgerufen von <https://neo4j.com/fullstack-graphql-applications-with-grandstack/>

Meier, A. & Kaufmann, M. (2016). *SQL- & NoSQL-Datenbanken (eXamen.press) (German Edition)* (8., überarb. u. erw. Aufl. 2016 Aufl.). <https://doi.org/10.1007/978-3-662-47664-2>

Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A. & Papadakos, T. (2014). *Value Proposition Design: How to Create Products and Services Customers Want (The Strategyzer series)* (1. Aufl.). Wiley. Abgerufen von <https://learning.oreilly.com/library/view/-/9781118968062/?ar>

-
- Partsch, H. (2010). *Requirements-Engineering systematisch*.
<https://doi.org/10.1007/978-3-642-05358-0>
- Schöning, S. (o. D.). Noch eine unverbindliche Liste elektronischer Laborbücher | Lost in Life Sciences. Abgerufen am 23. Oktober 2020, von <https://websites.fraunhofer.de/lost-in-life-sciences/?p=52>
- Scifo, E. (2020). *Hands-On Graph Analytics with Neo4j*. Abgerufen von <https://learning.oreilly.com/library/view/hands-on-graph-analytics/9781839212611/>
- Stapelkamp, T. (2007). *Screen- und Interfacedesign: Gestaltung und Usability für Hard- und Software (X.media.press) (German Edition) (2007. Aufl.)*. <https://doi.org/10.1007/978-3-540-32950-3>
- Stone, D., Jarrett, C., Woodroffe, M. & Minocha, S. (2005). *User Interface Design and Evaluation (Interactive Technologies)*. Burlington, USA: Morgan Kaufmann.
- Unger, M. (2018). *Analysing and Tracking Data Flows of a R&D Production* (Bachelorarbeit, Informatik). Freiburg, Deutschland: Albert-Ludwigs-Universität.
- Wilson, C. (2013). *User Interface Inspection Methods: A User-Centered Design Method* (1. Aufl.). Burlington, USA: Morgan Kaufmann.

Anhangsverzeichnis

Anhang 1: Fragenkatalog zum Dokumentationssystem	65
Anhang 2: Use-Case Diagramm.....	69
Anhang 3: User Storys	70
Anhang 4: Aktivitätsdiagramme.....	76
Anhang 5: Pflichtenheft	79
Anhang 6: Anforderungskatalog	80
Anhang 7: Datenanalyse verbale Beschreibung	82
Anhang 8: Entitäten-Beziehungsmodell	86

Anhang

Anhang 1: Fragenkatalog zum Dokumentationssystem

Quelle: Eigene Darstellung

Fragebogen zum Dokumentationssystem in der Abt. PVM

In dieser Umfrage geht es um das Dokumentationssystem für Versuche, Prozesse und Proben sowie sonstige anfallende Daten in den Laboratorien der Abteilung PVM.

Im Rahmen meiner Bachelorarbeit beschäftige ich mich mit der Fragestellung wie das momentane System erweitert bzw. verbessert werden kann.

Mithilfe der gewonnen Erkenntnisse soll ein neues System konzipiert und umgesetzt werden.

Bitte verfasse deine Antworten möglichst präzise und verständlich.

Solltest du Fragen/Anmerkungen zu einer Frage haben, wende dich gerne direkt an mich per Mail: henri.joss@ise.fraunhofer.de

Vielen Dank für Deine Teilnahme!

Name: [Klicken Sie hier, um Text einzugeben.](#) (Angabe freiwillig)

Rolle am Institut: [Klicken Sie hier, um Text einzugeben.](#) (Angabe freiwillig)

Datum: [Klicken Sie hier, um ein Datum einzugeben.](#)

1. Bei der Durchführung von Versuchen, Erstellung von Proben, Verwendung von Materialien etc. verwende ich das derzeitige Dokumentationssystem (Probenliste/Versuchsliste/Probenbegleitblatt...)

Ja Nein Teilweise

Wenn Ja/Teilweise: Pro Monat dokumentiere ich durchschnittlich

[Klicken Sie hier, um Text einzugeben.](#) (Versuche/Proben/Prozesse/Materialien)

Wenn Nein: Bitte zu Frage 6 springen.

2. Für folgende Dinge verwende ich das derzeitige System:

Probenbauplanung Probenherstellung Versuchsplanung

Versuchsdurchführung Prozessdokumentation

Materialdokumentation

Sonstiges:

3. Für folgende Dinge verwende ich das derzeitige System nicht:

4. Folgende Dinge finde ich am derzeitigen System gut:

5. Folgende Dinge stören mich am derzeitigen System:

6. Das derzeitige System ist in meinem Arbeitsalltag von hoher Signifikanz:

Trifft zu Trifft eher zu Neutral Trifft eher nicht zu Trifft nicht zu

7. Das derzeitige Dokumentationssystem halte ich insgesamt für:

Sehr gut Gut Befriedigend Schlecht Sehr schlecht

8. *(Nur wenn Frage 1 mit nein beantwortet wurde)*

Aus folgenden Gründen verwende ich das derzeitige System nicht:

9. *(Nur wenn Frage 1 mit nein beantwortet wurde)*

Ich könnte mir vorstellen in Zukunft ein System zur Dokumentation zu verwenden:

Trifft zu Trifft eher zu Neutral Trifft eher nicht zu Trifft nicht zu

10. Ein typisches Szenario und dessen grundsätzlicher Ablauf für die Dokumentation sieht aus wie folgt:

(Bitte beschreiben Sie wo die anfallenden Daten entstehen und wann, wo und wie diese dokumentiert werden)

11. In folgenden Situationen/Szenarien brauche ich Zugriff auf die dokumentierten Daten:

Im Labor Im Büro Unterwegs

Sonstiges:

12. Nach folgenden Kriterien suche ich um Daten wiederzufinden:

ID Name Typ Ersteller Beteiligte Proben

Beteiligte Materialien

Sonstiges:

13. Auf folgende Art und Weise würde ich gerne nach Daten suchen können:

14. Folgende Basisfunktionalitäten muss das neue System enthalten:

(Hiermit sind Funktionen gemeint die im neuen System zwangsläufig vorhanden sein müssen)

15. Folgende Funktionalitäten würde ich mir zusätzlich im neuen System wünschen:

(Bitte bewerte die einzelnen Punkte auf einer Skala von 1-10 nach Ihrer Relevanz, wobei 1 unwichtig und 10 sehr wichtig)

16. Folgende Funktionalitäten müssen oder sollten explizit nicht im neuen System enthalten sein:

17. Folgende Daten müssen im neuen System zwangsläufig erfasst werden können

(Hiermit sind Daten wie z.B. Probenart, Versuchsziel, Versuchsergebnis gemeint)

18. Für folgende Daten wäre eine Erfassung im System wünschenswert:

(Bitte wie in Frage 10 nach Relevanz bewerten)

19. Folgende Daten sollten in einem separaten System verwaltet werden:

20. Für die einheitliche Benennung der zu dokumentierenden Daten könnte ich mir folgendes System vorstellen:

(Bisher: YYMMDD_Projekt_Kzz_Proben-ID)

- Das bisherige System Ausschließlich Nummern (ID's)
 Ausschließlich Bezeichnung Mischform aus Nummern und Bezeichnung

Sonstiges:

21. Mit folgenden anderen Systemen/Schnittstellen sollte das System verknüpft sein

- Labeldrucker TLPV Schnittstelle zu EL Schnittstelle zu IV
 Einbindung des Messdatenordners aus dem VLAN

Sonstiges:

22. Durch ein neues System erhoffe ich mir eine Steigerung der Arbeitsqualität/-effizienz:

- Trifft zu Trifft eher zu Neutral Trifft eher nicht zu Trifft nicht zu

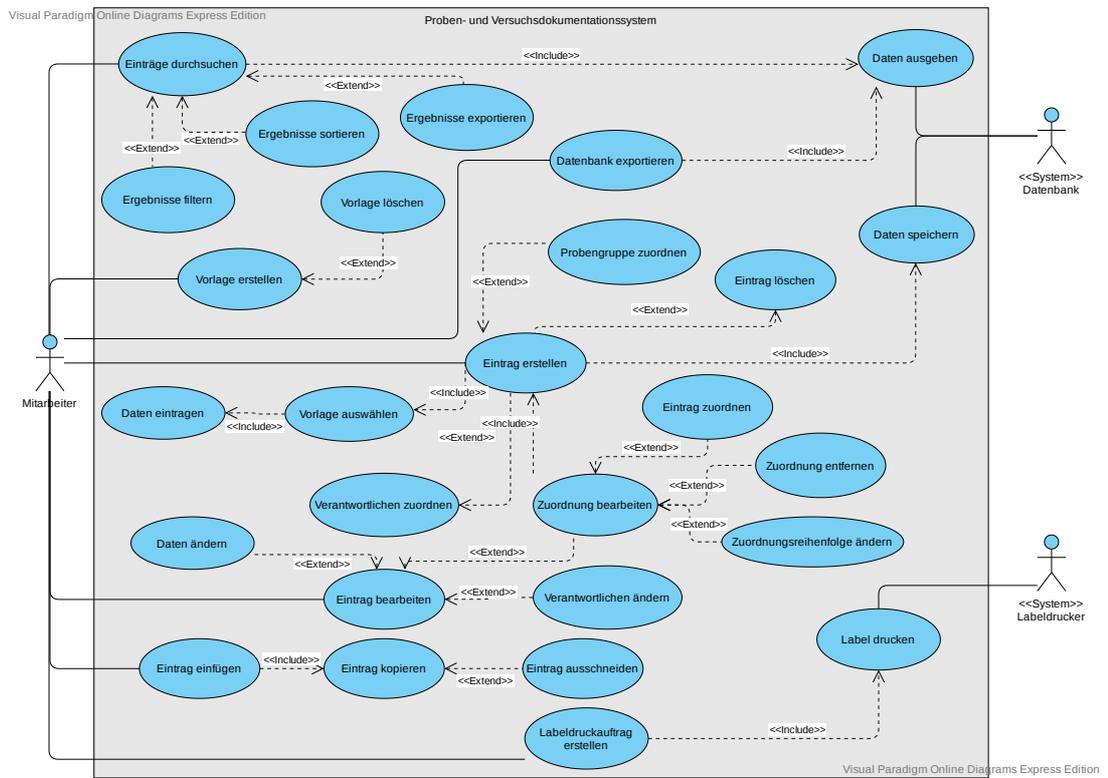
Zusätzliche Kommentare/Anmerkungen:

Bitte sende das ausgefüllte Formular wieder an folgende E-Mail Adresse zurück:

henri.joss@ise.fraunhofer.de

Anhang 2: Use-Case Diagramm

Quelle: Eigene Darstellung



Anhang 3: User Storys

Quelle: Eigene Darstellung

Suche&Filter	Priorität 10	15.09.2020
<p>User Story: Ich als wissenschaftliche Hilfskraft möchte nach bestimmten Kriterien im Dokumentationssystem nach Einträgen suchen und filtern können, um schnell an relevante Informationen zu gelangen.</p>		
<p>Akzeptanzkriterium: Angenommen ein bestimmter Vorgang soll nachvollzogen werden können, wenn mithilfe eines Schlagworts und bestimmter Kriterien nach diesem gesucht wird, dann sollen relevante Ergebnisse übersichtlich und schnell angezeigt werden.</p>		
<p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Indexierung der Datenbankeinträge für schnelle Suche • Sinnvolle Such- und Filterkriterien 		

Zugehörigkeit	Priorität 10	15.09.2020
<p>User Story: Ich als Teamleiter möchte die Möglichkeit haben Prozesse und Materialien zu Proben, Proben und Prozesse zu Versuchen und Versuche oder Proben zu Projekten zuzuordnen, damit die Beziehungen bzw. die Zugehörigkeiten eindeutig ersichtlich sind.</p>		
<p>Akzeptanzkriterium: Angenommen Versuche werden durchgeführt, wenn Proben an diesen beteiligt sind, dann müssen diese den Versuchen zuordenbar sein.</p>		
<p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Folgende Hierarchien gelten dabei: Projekt -> Versuch -> Prozess/Probe Projekt -> Probe -> Material/Prozess Projekt -> Prozess -> Anlage 		

Kopieren&Einfügen	Priorität 9	25.09.2020
User Story:		
Ich als Praktikant möchte Einträge und Referenzen auf Einträge kopieren, ausschneiden und einfügen können, um eine Arbeitersparnis bei neu anzulegenden Einträgen, die bereits ähnlich existieren, zu haben.		
Akzeptanzkriterium:		
Angenommen ein Eintrag soll angelegt werden, wenn dieser mithilfe eines existierenden Eintrags erstellt werden soll, dann muss sich der existierende Eintrag kopieren lassen.		
Rahmenbedingungen:		
<ul style="list-style-type: none"> • Es muss klar ersichtlich sein ob ein Eintrag kopiert wird oder ob lediglich die Zuordnung, also die Referenz darauf kopiert wird. 		

Vorlagen	Priorität 9	25.09.2020
User Story:		
Ich als Abteilungsleiter möchte die Möglichkeit haben Vorlagen für Versuche, Proben, Prozesse und Materialien zu erstellen, mithilfe dessen sich Inputs/Outputs definieren lassen aus denen sich Einträge erstellen lassen.		
Akzeptanzkriterium:		
Angenommen es existiert ein besonderer Prozess, der sich nicht mit der Standardvorlage abbilden lässt, wenn dieser dokumentiert werden soll, dann muss sich eine neue Vorlage anlegen und daraus ein Eintrag erstellen lassen.		
Rahmenbedingungen:		
<ul style="list-style-type: none"> • Vorlagen sind kategorisiert (Bsp. Messung/Elektronisch/EL) 		

Status	Priorität 10	25.09.2020
<p>User Story: Ich als Mitarbeiter möchte den aktuellen Status einer Probe oder eines Versuchs sehen und bearbeiten können, um zu wissen ob sich bspw. beteiligte Proben noch in Benutzung finden oder nicht.</p>		
<p>Akzeptanzkriterium: Angenommen ein Versuch befindet sich in Bearbeitung, wenn dieser abgeschlossen wurde ,dann muss der Status dementsprechend aktualisiert werden und ersichtlich sein.</p>		
<p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Es gibt eine vordefinierte Auswahl an Status aus den gewählt werden kann (Geplant, In Bearbeitung, Abgeschlossen, Verworfen) 		

Labeldrucken	Priorität 8	15.09.2020
<p>User Story: Ich als wissenschaftlicher Mitarbeiter möchte den Druckauftrag für das Probenlabel automatisch aus einer im System vorhandenen Probe erstellen lassen, um Prozesse zu automatisieren und Zeit zu sparen.</p>		
<p>Akzeptanzkriterium: Angenommen eine Probe soll mit einem Label versehen werden, wenn dafür ein Druckauftrag erstellt wird, dann soll dies automatisiert aus den Informationen des Systems geschehen.</p>		
<p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Labeldrucker muss mit dem System verbunden sein 		

Versuchsdarstellung	Priorität 8	15.09.2020
<p>User Story:</p> <p>Ich als Versuchsleiter möchte einzelne Schritte, verwendete Parameter und beteiligte Proben eines Versuchs in einer grafischen Übersicht dargestellt bekommen, damit man einen besseren Überblick über den Ablauf und die Durchführung bekommt.</p>		
<p>Akzeptanzkriterium:</p> <p>Angenommen ein Versuch wird durchgeführt, wenn dieser in mehreren Schritten mit jeweilig unterschiedlichen Parametern und verwendeten Proben stattfindet, dann muss eine übersichtliche grafische Darstellung möglich sein.</p>		

Materialbibliothek	Priorität 10	15.09.2020
<p>User Story:</p> <p>Ich als Laborleiter möchte bei der Dokumentation einer Probe aus einer Liste an Materialien auswählen können, um Zeit zu sparen und die einheitliche Benennung der Materialien zu gewährleisten.</p>		
<p>Akzeptanzkriterium:</p> <p>Angenommen eine Probe wird hergestellt, wenn verschiedene beteiligte Materialien zugeordnet werden, dann müssen diese aus einer Liste auswählbar sein.</p>		
<p>Rahmenbedingungen:</p> <ul style="list-style-type: none"> • Materialien werden aus einer Materialdatenbank geladen • Individuelle Bezeichnungen sollen durch vorgegebene Bezeichnungen weitestgehend ausgeschlossen sein • Sollte das Material sich nicht in der Datenbank befinden, muss es möglich sein dieses entweder der Datenbank hinzuzufügen oder als Freitextfeld zu dokumentieren • Materialien sind bestimmten Kategorien zugeordnet 		

Prozessbibliothek	Priorität 9	25.09.2020
<p><i>User Story:</i></p> <p>Ich als Laborleiter möchte, bei der Dokumentation einer Probe oder eines Versuchs, aus einer Liste an Prozessen auswählen können, um Zeit zu sparen und die weitestgehende Standardisierung der Prozesse und verwendeten Parameter zu gewährleisten.</p>		
<p><i>Akzeptanzkriterium:</i></p> <p>Angenommen eine Probe wird hergestellt oder ein Versuch wird durchgeführt, wenn verschiedene Prozesse daran beteiligt sind, dann müssen diese aus einer Bibliothek auswählbar sein.</p>		
<p><i>Rahmenbedingungen:</i></p> <ul style="list-style-type: none"> • Prozesse werden aus einer Prozessdatenbank geladen • Sollte sich der Prozess nicht in der Datenbank befinden, muss es möglich sein diesen entweder der Datenbank hinzuzufügen oder als Freitextfeld zu dokumentieren • Prozesse sind bestimmten Kategorien zugeordnet 		

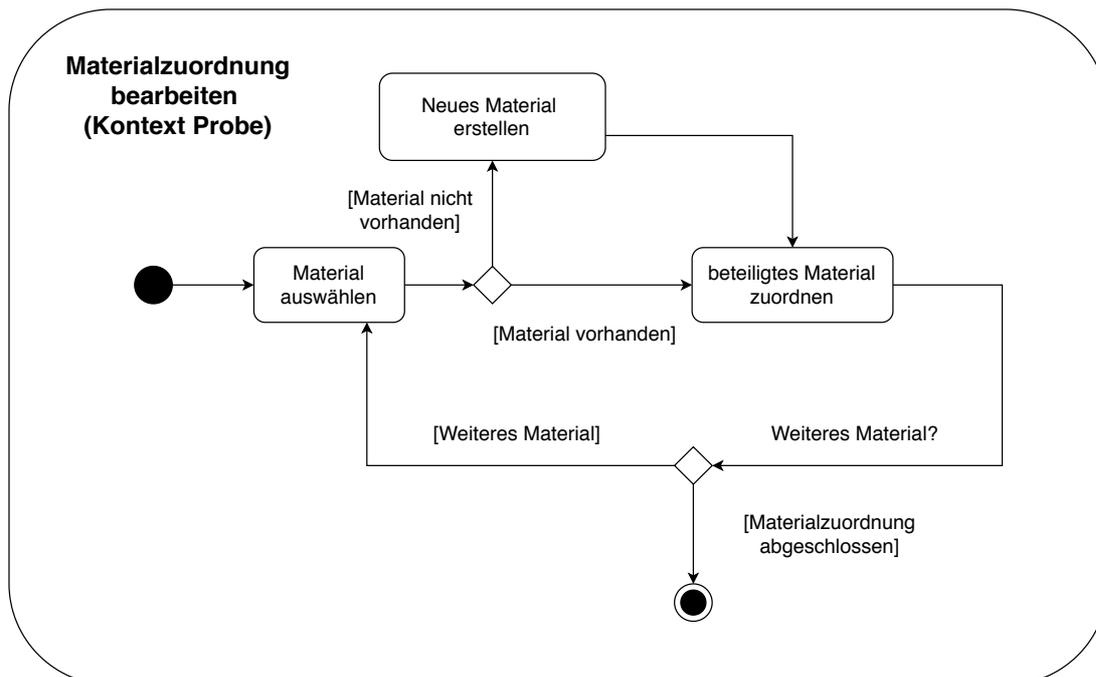
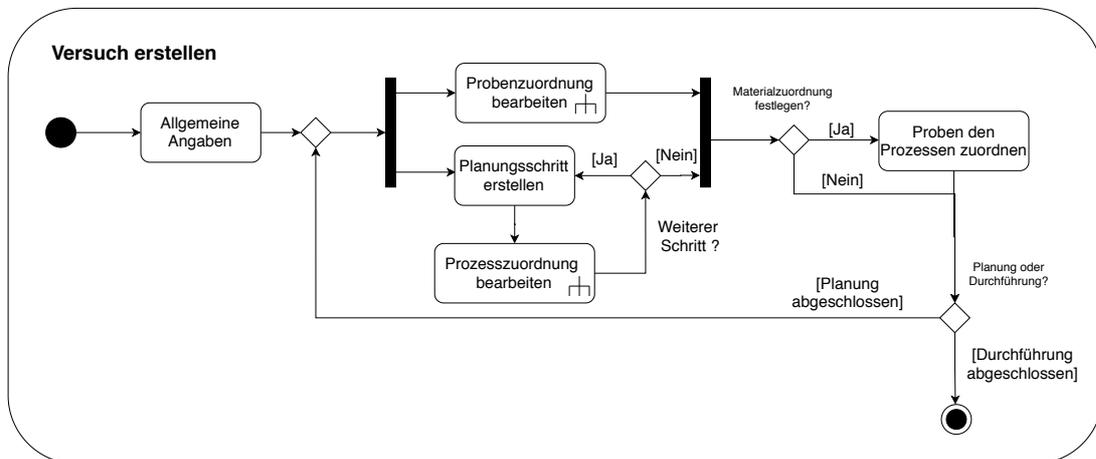
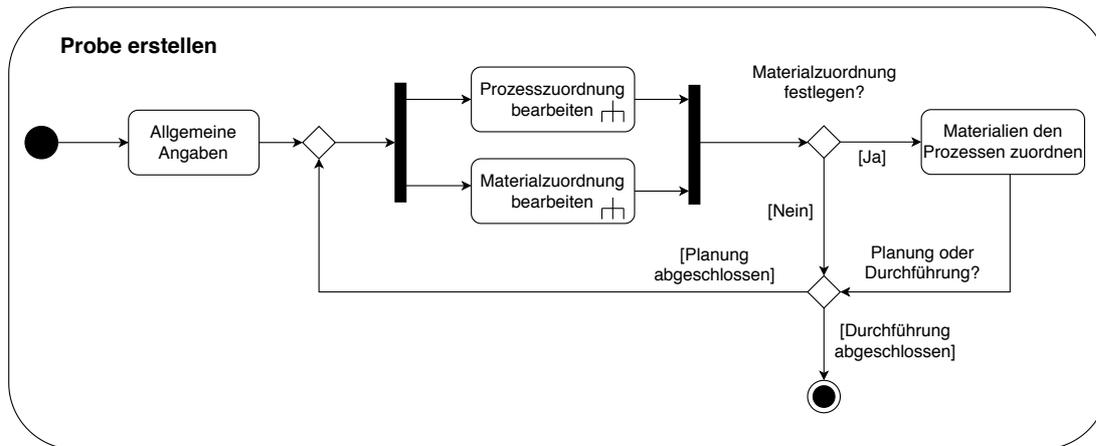
Probenverantwortlichkeit	Priorität 9	15.09.2020
<p><i>User Story:</i></p> <p>Ich als Projektleiter möchte bei Proben zwischen Hersteller und Eigentümer unterscheiden können, da die Verantwortlichkeit einer Probe nach Herstellung weitergegeben werden kann.</p>		
<p><i>Akzeptanzkriterium:</i></p> <p>Angenommen eine Probe wird hergestellt, wenn diese die Probe nach Herstellung einer anderen Person zugeordnet wird, dann muss der Hersteller und Besitzer im System nachvollziehbar sein.</p>		
<p><i>Rahmenbedingungen:</i></p> <ul style="list-style-type: none"> • Hersteller und Besitzer müssen im System als Mitarbeiter angelegt sein • Hersteller ist nach Eintragung final, Besitzer kann sich dynamisch ändern 		

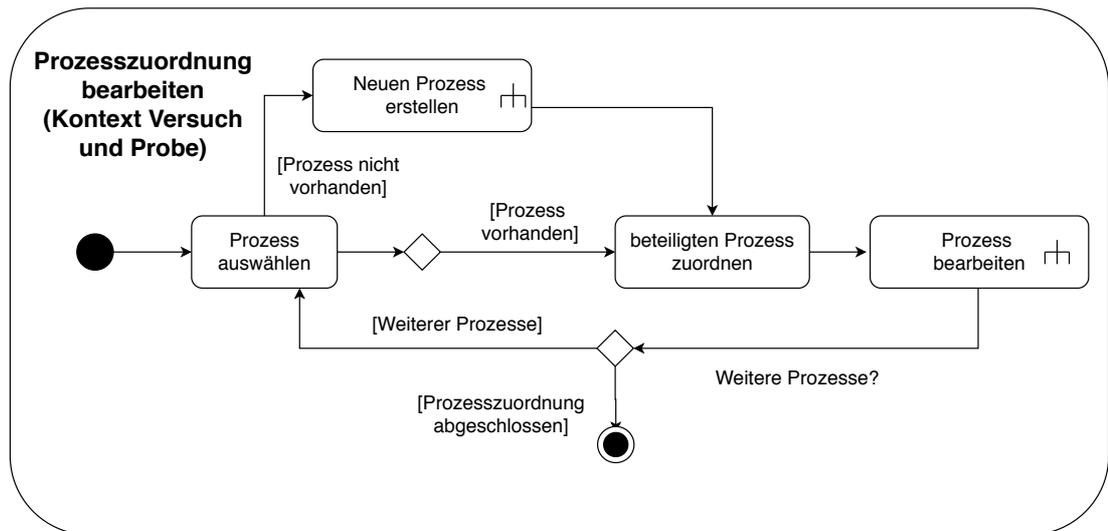
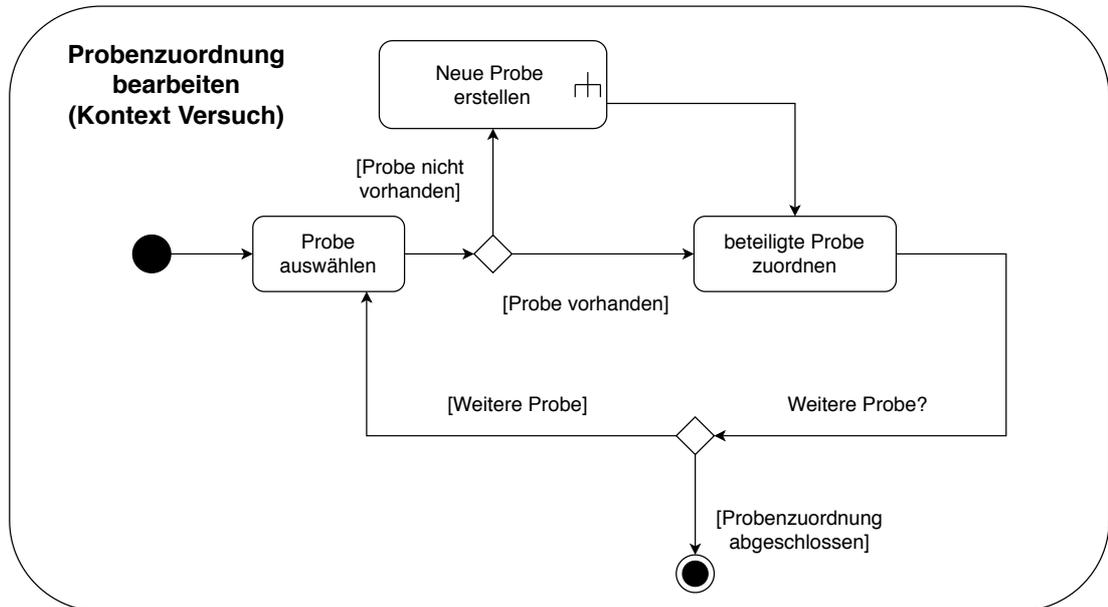
Probengruppen	Priorität 6	15.09.2020
User Story:		
Als Ingenieur möchte ich Probengruppen erstellen können, bei denen die Abweichungen zwischen den einzelnen Proben ersichtlich werden, weil diese Proben sich lediglich in einzelnen Parametern unterscheiden.		
Akzeptanzkriterium:		
Angenommen es existiert eine Gruppe an Proben, dessen einzelne Proben sich nur geringfügig voneinander unterscheiden, wenn diese dokumentiert werden, dann muss es möglich sein nur die Abweichungen eintragen zu können.		
Rahmenbedingungen:		
<ul style="list-style-type: none"> • Proben können einer Probengruppe zugeordnet werden 		
Hinweise:		
<ul style="list-style-type: none"> • Darstellung der Abweichung wird vorerst nicht umgesetzt, sondern nur die Zuordnung einer Probe zu einer Gruppe 		

Anhang	Priorität 9	15.09.2020 VERWORFEN
User Story:		
Ich als wissenschaftlicher Mitarbeiter möchte an einen Versuch oder eine Probe Dateien, wie z. B. Bilder, anhängen können, um zugehörige Daten schnell wiederzufinden und verfügbar zu haben.		
Akzeptanzkriterium		
Angenommen ein Versuch wird geplant, durchgeführt oder ausgewertet, wenn dabei zusätzliche Dokumente anfallen, dann müssen diese als Anhang zum Versuch abgelegt werden können.		
Hinweise:		
<ul style="list-style-type: none"> • Wird vermutlich in Form von Links o.ä. umgesetzt, da CMS impraktikabel ist 		

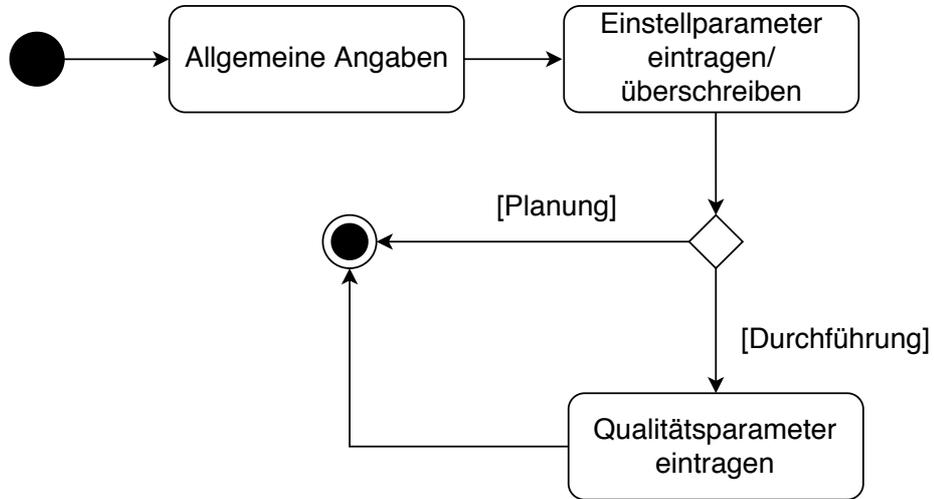
Anhang 4: Aktivitätsdiagramme

Quelle: Eigene Darstellung

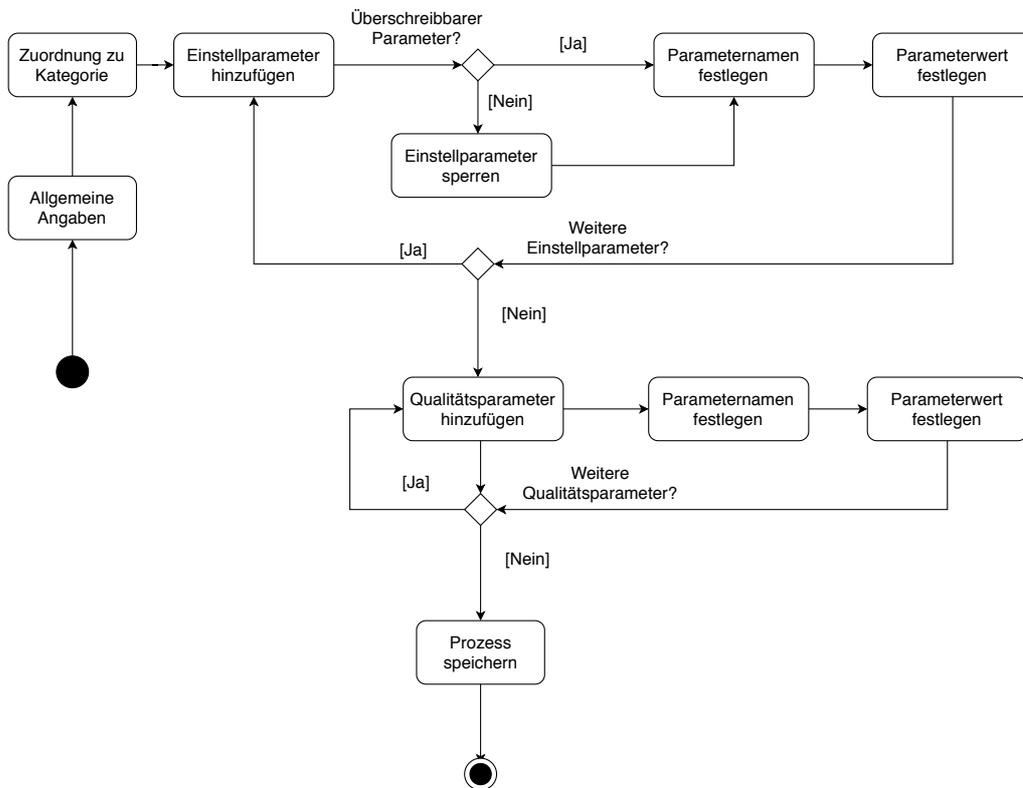




Prozess erstellen/bearbeiten (Kontext Versuch und Probe)



Prozessvorlage erstellen



Anhang 5: Pflichtenheft

Quelle: Eigene Darstellung

Pflichtenheft

1. Benötigt wird ein System, das eine **Probendokumentation** abbildet, die einem fachkundigen Dritten erlaubt Proben erneut herstellen zu können. Die Dokumentation ist eine Anleitung zur Herstellung und hat relevante Beteiligte, Projektzugehörigkeit, Materialien, Anlagen, Prozesse und Herstellungsschritte zu dokumentieren. Die Dokumentation ist mit relevanten Informationen zu ergänzen, ohne deren eine wiederholte Herstellung der Probe nicht möglich ist. Das betrifft unter Anderem zusätzlich verwendete Hilfsmittel, besondere Aufbauten oder Prozessschritte. Das System muss in der Dokumentation verschiedene Rollen von Personen abbilden können. Proben müssen Probengruppen zugeordnet werden können und auf **Vorlagen** basieren können.
2. Verwendete **Materialien** sind unter Verwendung der korrekten Bezeichnung (z.B. Hersteller und Materialkennzeichnung) sowie individuell relevanter technischer Merkmale (bspw. Dicke, Farbe, Chargennummer etc.) zu dokumentieren. Individuelle Bezeichnungen sollen durch vorgegebene Bezeichnungen weitestgehend ausgeschlossen sein. Materialien und Parameter müssen gespeichert und den Proben zugeordnet werden können. Materialien sollen unabhängig von Proben, Versuchen und Prozessen angelegt, bearbeitet und verwaltet werden können (Materialbibliothek). Materialien sollen aus **Materialvorlagen** erstellt werden können, welche typische und zwingend erforderliche Inputs vordefinieren.
3. **Prozesse** sind unter Nennung der verwendeten Anlagen und wichtiger Prozessparameter (bspw. Prozesstemperaturen, -zeiten, -drücke) zu erfassen. Die Erfassung und Speicherung von Prozessparametern soll durch eine Prozessbibliothek unterstützt werden. Prozesse und -parameter sollten weitestgehend standardisiert sein (Rezepte). Prozesse sollen unabhängig von Versuchen und Proben angelegt, bearbeitet und verwaltet werden können (Prozessbibliothek).
4. Prozesse (aus Herstellung und Versuchen) sollen durch vorgegebene Eingaben (bspw. Prozessgrößen) und Ergebnisse (bspw. aus Messungen) gekennzeichnet sein. Diese Eingaben sollen bei Anlage von Proben und Versuchen sowie bei der Ergebnisdokumentation von Versuchen abgefragt werden. (Bsp: Ein Versuch erfordert eine Prüftemperatur von x °C. Diese soll dann bei der Versuchsanlage als Input abgefragt werden. Nach Versuchsdurchführung steht ein Messergebnis vom Typ Y %Leistungsverlust zur Verfügung, welches auch abgefragt werden soll.). **Prozessvorlagen** sollen die notwendigen Inputs/Outputs definieren.
5. Unter Nutzung von Proben werden **Versuche** durchgeführt, welche ebenfalls zu dokumentieren sind. Hierbei sind wichtige Versuchsparameter (bspw. Ziel, Personen, Proben, Anlagen, Materialien etc.), der aktuelle Stand von Versuchsschritten (Status) sowie relevante Prozesse, Hilfsmittel, Personen und Ergebnisse zu dokumentieren. Versuche sollen unabhängig von Materialien, Prozessen und Proben angelegt, bearbeitet und verwaltet werden können (Versuchsbibliothek).
6. Das System muss über eine zugängliche **graphische Benutzeroberfläche** verfügen, die die festgelegten Prozesse (siehe Anhänge) abbildet.
7. Das System muss die **organisatorischen Strukturen** abbilden (Projekte, Organisationseinheiten etc.).
8. Das System muss **Schnittstellen** zu weiteren externen Systemen bereitstellen: Labeldrucker, Prozessanforderungsblätter (Stringen, Laminieren, EL, IV etc.)
9. **Editieren**: Das System muss die zeitlich getrennte Erstellung, Ergänzung, Zuordnung und Veränderung von Einträgen gestatten. Eine Nachverfolgbarkeit (Historie, Log) von Veränderungen ist optional. Proben, Versuche, Prozesse etc. müssen kopiert werden können.
10. Das System muss über eine **Ausgabe** verfügen. Bibliotheken/Datenbanken müssen durchsucht, gefiltert und sortiert werden können. Entsprechende Exports müssen verfügbar sein. Die Historie von Proben (Beteiligung an Versuchen) muss erkennbar sein. Für den Fall, dass eine Suchfunktion nicht im System hinterlegt ist, muss eine Alternative möglich sein (bspw. Gesamtdatenbankexport als Excel). Eine Druckfunktion für ausgewählte Elemente und Teilsysteme ist notwendig.
11. Das System muss dokumentiert, wartbar und modular (Teildokumentationssysteme Probe, Versuch, ...) umgesetzt werden.
12. Die Möglichkeit zur **Hinterlegung von Dateien**, Ergebnisdokumenten etc. soll umgesetzt werden.
13. **Bezeichnungen** für Proben, Versuche, Prozesse, Anlagen etc. sollen weitestgehend standardisiert sein. Für ausgewählte Elemente sind vorgegebene Bezeichnungen zu verwenden (bspw. Proben-ID).
14. **Vorlagen** (Proben, Materialien, Versuche etc.) sollen dynamisch in die GUI eingebunden werden können (Ziel: einfache Erstellung von weiteren Vorlagen).
15. Die Implementierung muss dergestalt erfolgen, dass regelmäßig Feedback zu GUI und System gegeben werden kann (regelmäßige Funktionsprototypen, iterative Entwicklung mit aufsteigendem Implementierungsumfang,...).

Anhang 6: Anforderungskatalog

Quelle: Eigene Darstellung

Nutzer/Rollensystem	
User anlegen, verwalten, löschen	ch
Userrollen und damit verbundene Rechte zuweisen	ch
User Verantwortlichkeiten delegieren	ch
Suche&Filter	
Einträge durchsuchen	mh
Einträge nach bestimmten Kriterien filtern	mh
Einträge nach bestimmten Kriterien sortieren	mh
Projekt	
Projekt erstellen, verwalten, löschen	mh
Versuche und Proben Projekten zuordnen	mh
Versuch/Probe	
Versuch/Probe erstellen, bearbeiten, löschen	mh
Proben Versuchen zuordnen	mh
Materialien Proben zuordnen	mh
Versuche/Proben Prozesse zuordnen	mh
Probengruppen erstellen	sh
Darstellung der Abweichung zwischen einzelnen Proben	ch
Versuchsplanung anlegen	ch
Reminder für Versuchsplan	ch
Versuch/Probenvorlage erstellen, bearbeiten, lösche	ch
Prozess	
Prozess erstellen, bearbeiten, löschen	mh
Prozessvorlage erstellen, bearbeiten, löschen	mh
Material	
Material erstellen, bearbeiten , löschen	mh
Material Prozess zuordnen	mh
Materialvorlagen erstellen, bearbeiten, löschen	sh
Anlagen (Equipment)	
Anlage erstellen, bearbeiten, löschen	mh
Kopieren&Einfügen	
Eintrag kopieren	mh

Eintrag ausschneiden	mh
Eintrag einfügen	mh
Status	
Status von Einträgen festlegen, bearbeiten	mh
Import&Export	
Label drucken	mh
Versuchsplan drucken	ch
Messauftrag exportieren	ch
Import Prozessparameter aus Formular	ch
Projekte von externen Quellen importieren	ch
Dateianhänge	
Relevante Dateien an ein Projekt, Versuch oder Probe anhängen können	ch
Sonstiges	
Probenhistorie verfolgen	sh

Anhang 7: Datenanalyse verbale Beschreibung

Quelle: Eigene Darstellung

Projekt	<i>Auftrag Nr. 1</i>
	<i>Datum 09.11.2020</i>
Ziel: Projekte bilden auf oberster Ebene den Rahmen für Proben und Versuche.	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Ein Projekt besitzt eine eindeutige Projektnummer 2. Proben und Versuche müssen einem Projekt zugeordnet sein 3. Ein Projekt kann beliebig viele Proben und Versuche beinhalten 	

Mitarbeiter	<i>Auftrag Nr. 5</i>
	<i>Datum 09.11.2020</i>
Ziel: Mitarbeiter sind im System in verschiedenen Rollen zur Erfassung der Verantwortlichkeiten und zur Verbesserung der Bereitstellung von Daten erfasst.	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Mitarbeiter besitzen eine spezifische Rolle 2. Mitarbeiter können Projektleiter eines Projekts sein 3. Mitarbeiter können Eigentümer oder Verantwortliche eines Versuchs oder einer Probe sein 4. Mitarbeiter können Anlagenverantwortliche sein 5. Mitarbeiter können Prozessen von Proben oder Versuchen zugewiesen sein 	

ID	<i>Auftrag Nr. 8</i>
	<i>Datum 08.10.2020</i>
Ziel: Jeder Eintrag im System muss eindeutig identifizierbar sein	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Jede ID ist einzigartig 2. Jeder Eintrag besitzt eine ID 	

Status	<i>Auftrag Nr. 4</i>
	<i>Datum 08.10.2020</i>
Ziel: Der aktuelle Status von Projekten, Versuchen oder Proben und Prozessen muss erkenntlich sein	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Die verfügbaren Status sind: Geplant, In Bearbeitung, Abgeschlossen, Verworfen 2. Ein Projekt kann den Status „Abgeschlossen“ erst bekommen, wenn alle zugeordneten Versuche und Proben den Status „Abgeschlossen“ besitzen 3. Ein Versuch kann den Status „Abgeschlossen“ erst bekommen, wenn alle Versuchsschritte den Status Abgeschlossen besitzen 	

Versuch	<i>Auftrag Nr. 2</i>
	<i>Datum 08.10.2020</i>
Ziel: Zur Nachvollziehbarkeit der Planung und Durchführung eines Versuchs muss dieser mit klarer zeitlicher Abfolge und allen Zugehörigkeiten abgebildet sein.	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Ein Versuch besteht aus Versuchsplanung und Durchführung 2. Ein Versuch wird mit ein oder mehr Versuchsschritten durchgeführt 3. Jeder Versuchsschritt enthält einen Status und eine Beschreibung des Schritts sowie ein oder mehr beteiligte Prozesse mit definierter Reihenfolge 4. Ein Versuch besitzt zusätzlich ein oder mehr beteiligte Proben 5. Ein Versuch besitzt ein definiertes Versuchsziel und ein Versuchsergebnis 6. Ein Versuch besitzt einen Versuchseigentümer und Verantwortlichen 	

Probe	<i>Auftrag Nr. 3</i>
	<i>Datum 08.10.2020</i>
Ziel: Die Planung und Herstellung einer Probe muss nachvollziehbar mit allen Beteiligten Zugehörigkeiten dokumentiert werden können	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Es wird in Planung und Erstellung einer Probe unterschieden 2. Eine Probe besteht aus ein oder mehr verwendeten Materialien 3. Eine Probe besitzt ein oder mehr beteiligte Prozesse zur Herstellung 4. Eine Probe kann an ein oder mehr Prozessen beteiligt sein 5. Eine Probe besitzt einen Probeneigentümer und Verantwortlichen 	

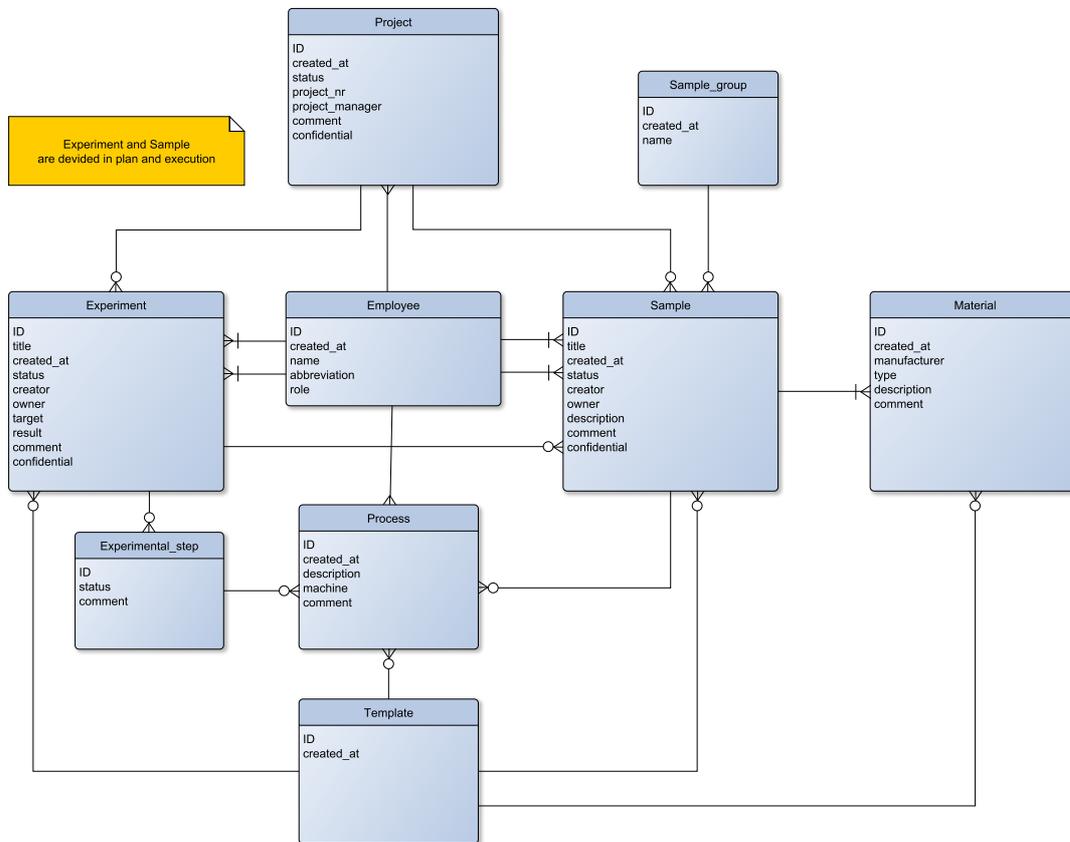
Prozess	<i>Auftrag Nr. 6</i>
	<i>Datum 08.10.2020</i>
Ziel: Prozesse dienen der Erfassung der Herstellung oder Durchführung einer Probe oder eines Versuchs	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Jeder Prozess besitzt eine Beschreibung 2. Jeder Prozess besitzt eine Anlage 3. Prozesse sind kategorisch erfasst (Bsp: Messung/elektrisch/EL-Messung) 4. Proben oder Materialien können Prozessen zugeordnet werden 5. Zu dokumentierende Angaben, Qualitäts- und Set-up-Parameter werden durch Vorlagen definiert 	

Vorlage	<i>Auftrag Nr. 7</i>
	<i>Datum 09.11.2020</i>
Ziel: Vorlagen definieren die Charakteristik von Prozessen	
Informationssachverhalte:	
<ol style="list-style-type: none"> 1. Vorlagen definieren allgemeine Angaben, Einstell- und Qualitätsparameter 2. Jeder Eintrag wird anhand immer genau einer Vorlage erstellt 	

Material	<i>Auftrag Nr. 9</i>
	<i>Datum 09.11.2020</i>
Ziel: Materialien definieren die Zusammensetzung einer Probe	
Informationssachverhalte: <ol style="list-style-type: none">1. Materialien werden zur Herstellung von Proben benötigt2. Zur besseren Strukturierung sind sie in Kategorien eingeteilt3. Materialien besitzen einen Typ und eine Beschreibung	

Anhang 8: Entitäten-Beziehungsmodell

Quelle: Eigene Darstellung



Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unzulässige fremde Hilfe angefertigt habe.

Alle verwendeten Quellen (Literatur, Internet) sind im Literaturverzeichnis vollständig zitiert.

Ort, Datum:

Unterschrift: