|  |  |
| --- | --- |
| Aufgabengebiet: |  |
| Autor: | Steven Ringger |
| Verantwortliche Fachkraft: | Gian-Luca Daffré |
| Berufsbildner: | Robert Kolb |
| Hauptexperte: | André Lichtsteiner |
| Nebenexperte: | Roger Bühler |

Individuelle Praktische Arbeit

2020

Inhalt

[1. Umfeld und Ablauf 3](#_Toc34729658)

[1.1 Ausgangslage 3](#_Toc34729659)

[1.2 Detaillierte Aufgabenstellung 3](#_Toc34729660)

[1.3 Projektaufbauorganisation 5](#_Toc34729661)

[1.4 Mittel und Methode 5](#_Toc34729662)

[1.5 Vorkenntnisse 5](#_Toc34729663)

[1.6 Vorarbeiten 5](#_Toc34729664)

[1.7 Neue Lerninhalte 5](#_Toc34729665)

[1.8 Arbeitsplatz 6](#_Toc34729666)

[1.9 Sicherung der Daten 6](#_Toc34729667)

[1.10 Zeitplan 6](#_Toc34729668)

[1.10.1 Planung 7](#_Toc34729669)

[1.10.2 Planung & Realität 8](#_Toc34729670)

[1.11 Sitzungsprotokolle 9](#_Toc34729671)

[1.11.1 Erster Expertenbesuch 9](#_Toc34729672)

[1.11.2 Zweiter Expertenbesuch 9](#_Toc34729673)

[1.12 Arbeitsjournale 10](#_Toc34729674)

[2. Projekt 16](#_Toc34729675)

[2.1 Kurzfassung 16](#_Toc34729676)

[2.1.1 Kurze Ausgangssituation 16](#_Toc34729677)

[2.1.2 Umsetzung 16](#_Toc34729678)

[2.1.3 Ergebnis 16](#_Toc34729679)

[2.2 Informieren 16](#_Toc34729680)

[2.2.1 Abgrenzung und Deklaration gemachter Vorarbeiten 16](#_Toc34729681)

[2.2.2 Analyse und Verfeinerung der Arbeit 16](#_Toc34729682)

[2.2.3 Projektmanagementmethodik 17](#_Toc34729683)

[2.2.4 Use Case Diagramm 17](#_Toc34729684)

[2.2.5 Definition Arbeitspakete 18](#_Toc34729685)

[2.3 Planen 25](#_Toc34729686)

[2.3.1 Use Case Spezifikationen 25](#_Toc34729687)

[2.3.2 Mockups / Wireframes 28](#_Toc34729688)

[2.3.3 Sequenzdiagramm 33](#_Toc34729689)

[2.3.4 Systemkontextdiagramm 34](#_Toc34729690)

[2.3.5 Planung der benötigten Komponenten 35](#_Toc34729691)

[2.3.6 Testkonzept 36](#_Toc34729692)

[2.3.7 Test Cases – Blackbox-Testing 37](#_Toc34729693)

[2.3.8 Design- & Architekturmöglichkeiten 39](#_Toc34729694)

[2.3.9 Gegenüberstellung der Implementierungsvarianten 40](#_Toc34729695)

[2.4 Entscheiden 40](#_Toc34729696)

[2.4.1 Nutzwertanalyse 40](#_Toc34729697)

[2.4.2 Entscheidungsmatrix 40](#_Toc34729698)

[2.4.3 Erklärung der gefällten Entscheidungen 41](#_Toc34729699)

[2.5 Realisieren 41](#_Toc34729700)

[2.5.1 Aufbau der Grundstruktur 41](#_Toc34729701)

[2.5.2 Einzelnen Komponenten 42](#_Toc34729702)

[2.5.3 Storybook Einträge 43](#_Toc34729703)

[2.6 Kontrollieren 45](#_Toc34729704)

[2.6.1 Erster Testdurchlauf 45](#_Toc34729705)

[2.6.2 Fehlerbehebung und zweiter Testdurchlauf 45](#_Toc34729706)

[2.7 Auswerten 45](#_Toc34729707)

[2.7.1 Persönliches Fazit 45](#_Toc34729708)

[2.8 Glossar 46](#_Toc34729709)

[2.9 Quellenverzeichnis 46](#_Toc34729710)

[2.10 Tabellenverzeichnis 47](#_Toc34729711)

[2.11 Abbildungsverzeichnis 48](#_Toc34729712)

[3. Anhänge 48](#_Toc34729713)

# Umfeld und Ablauf

Die Punkte 1.1, 1.2, 1.4, 1.5, 1.6 und 1.7 wurden von PkOrg übernommen.

## Ausgangslage

Diverse unserer Kunden sind im Fachgebiet des Gesundheitswesens tätig. Integraler Bestandteil dessen sind u.a die Bereiche Gesundheitsförderung, Beratung, Prävention sowie klassische und alternative Therapieformen. Obwohl der Markt vereinzelt Produkte im Sinne der Optimierung resp. Entlastung verschiedenster Arbeitsschritte dieser Domäne anbietet, scheint die Nachfrage nach spezifischen Lösungen stets hoch zu sein. Im Rahmen der aufkommenden praktischen Arbeiten unserer Lehrlinge haben wir uns diversen pendenten Kundenanfragen gewidmet, diese analysiert und konnten als Ergebnis dessen eine Gewisse Schnittmenge feststellen. Diese sollte als Inhalt für die Abschlussarbeiten fungieren und die Anforderungen der ersten Iteration definieren. Die Aufgaben der Lernenden sind modular klar aufgeteilt und haben keine Abhängigkeiten. Diese Module werden zu einem späteren Zeitpunkt zusammengeführt. Dies ist aber nicht Teil der Aufgabe.

## Detaillierte Aufgabenstellung

Abgrenzung des Projektumfangs und Definition der Anforderungen:

Hauptaufgabe liegt darin eine initiale Analyse des Klienten bei der Erstkonsultation zu erstellen. Dafür sind entsprechende GUI’s zu programmieren. Dem Anhang beigefügt ist ein skizziertes MockUp. Dieses dient als Grundlage für die Definition. Das statisch hinterlegte Bild eines Körpers aus mehreren Ansichten bietet dem Therapeuten die Möglichkeit die analysierten Probleme gewissen Lokalitäten zuzuordnen.   
Ein GUI muss erstellt werden in dem eine Liste mit den Anamnesen des Klients angezeigt wird. Danach kann eine Anamnese in einem weiteren GUI erfasst werden. Dieses GUI für die Anamnese ist zu erstellen mit einem Abbild eines menschlichen Körper von vorne und von hinten (sh. Abbild). Das Abbild beinhaltet mehrere Punkte am Körper die ausgewählt werden können. Zu einem Punkt können entsprechende Informationen erfasst werden. Somit ergibt sich zu einer Anamnese mehre mögliche Punkte. (1 zu N Beziehung zwischen Anamnese und Punkte).

Bei der Erfassung des Therapieplan werden Vorschläge für mögliche Therapien vorgeschlagen. Es kann entweder einen Vorschlag übernommen werden oder ein neuer Therapieplan erstellt werden.

Allgemeine Informationen

-Für die Visualisierung der Komponenten ist ein Storybook zu erstellen.   
-Die ganze Kommunikation mit dem Backend basiert auf RESTful API. Dabei steht der Backend als Mockup bereits vor Beginn der Arbeit zu Verfügung.   
-Auf eine Berechtigungsmatrix wird verzichtet, da aktuell nur ein Dataowner vorhanden ist. Dies basierend auf die Anforderungen des Kunden. Funktionen und Pages müssen also nicht zwingend eingeschränkt werden.   
-Auf Mehrsprachigkeit wird verzichtet.

User Stories (Arbeitspakete)

AP1 - Design   
Als Kunde erwarte ich, dass für die Visualisierung der Komponenten ein Storybook erstellt wird. Das Storybook beinhaltet die in den Formularen genutzten Komponenten. Damit möchte ich frühzeitig sehen wie die verschiedenen Komponenten aussehen und wie sie sich verhalten. Während der IPA ist steht der Kunde nicht zur Verfügung. Das Vorgehen ist jedoch wichtiger Bestandteil der Aufgabe.

AP2 - UML   
Als Kunde erwarte ich ein UseCase Diagram. Die jeweiligen Spezifikationen sind in fully dressed Spezifikationen abzufassen. (Larman).

AP3 - Body   
Als Therapeut möchte ich ein Bild des menschlichen Körper von vorne und von hinten sehen. Dabei sind auf dem Bild mehrere Punkte zu sehen, bei welchem etwaige Probleme vorhanden sind oder entsprechend neu erfasst werden können. Bei welchen Punkt Probleme vorhanden sind, ist auf dem Bild ersichtlich.

AP4 - Erfassen Anamnese   
Als Therapeut möchte ich eine Anamnese für einen Klient erfassen. Inhaltlich ist der Level des Schmerzes zwischen 1-5 abzubilden. Des weiteren sind subjektive und objektive Wahrnehmung, die Einschätzung anzuzeigen. Bei der Anamnese möchte ich mehrere (Körper-)Punkte gleichzeitig erfassen können.

AP5 - Vorschläge Therapie   
Als Therapeut möchte ich zur Anamnese 3 Vorschläge von möglichen Therapien für den ausgewählten Punkt vorgeschlagen bekommen. Hierfür ist ein Request zum Backend zu machen. Der Backend liefert die 3 erfolgreichsten Therapien. Es kann eine Therapie aus dieser Liste gewählt und abgespeichert werden.

AP5 -Erfassen eigene Therapie ohne Vorschlag   
Als Kunde möchte ich, wenn mich die Therapie-Vorschläge nicht ansprechen, eine neue Therapie erfassen können.

AP6 - Validierung Eingaben   
Als Therapeut möchte ich dass alle Benutzereingaben sinnvoll validiert werden. Für Datum sind Datepicker zu nutzen.

AP7 - Kommunikation Backend   
Als Kunde möchte ich, dass die Kommunikation mit dem Backend über RESTful Services gemacht wird. Für das Mocking sind die bereits bestehenden Mockup-Komponenten zu nutzen.

AP8 - Komponententests   
Als Kunde möchte ich dass es für dieses GUI Komponententests gemacht werden und der Inhalt des GUI’s validiert wird.

AP9 - Systemtest:   
Als Kunde möchte ich dass es für dieses GUI Systemtests mit cypress gemacht werden und diese korrekt durchlaufen..

Erläuterungen zu den gewählten individuellen Beurteilungskriterien:

I1 KATALOG-KRITERIUM 190

Für das GUI wird das Framework React Bootstrap eingesetzt. Das ganze GUI muss komponentenbasiert und die Gestaltgesetze ersichtlich sein. (Prägnanz, Ähnlichkeit, Nähe, Geschlossenheit, Gute Fortsetzung und Gemeinsames Schicksal). Die Applikation muss auf einem IPAD korrekt dargestellt werden, da der Kunde potentiell damit arbeitet.

I2 KATALOG-KRITERIUM 232

Die Komponenten sind funktionsspezifisch zu implementieren. Sie sollen auch wiederverwendbar sein. Kommentare sind sinnvoll einzusetzen. Das heisst nicht aus Prinzip kommentieren, was bereits am Naming erkennbar ist.

I3 FREIES KRITERIUM   
Strukturierung der Komponenten nach Atomic Design Patterns. Es ist ein Storybook für die Visualisierung der GUI-Komponenten zu erstellen. Dieses beinhaltet die diversen Feldtypen, Tabelle und Buttons sowie ein einfaches Formular

I4 KATALOG KRITERIUM 235 (modifiziert)   
Seitens Kandidaten wird ein UseCase Diagram erwartet. Die jeweiligen Spezifikationen sind in fully dressed Spezifikationen abzufassen. (Larman)

I5 KATALOG KRITERIUM 194   
Für das Formularhandling ist die Library Formik einzusetzen und mit der Library Yup die Validierungen vorzunehmen. Die Fehlermeldungen sind entsprechend zu formulieren und dürfen keine Rechtschreibefehler beinhalten. Ein klares Konzept zu den Fehlermeldungen ist erkennbar.

I6 GUI Design (Fokus: Benutzerfreundlichkeit über die ganze Applikation)   
keine Anmerkungen

## Projektaufbauorganisation

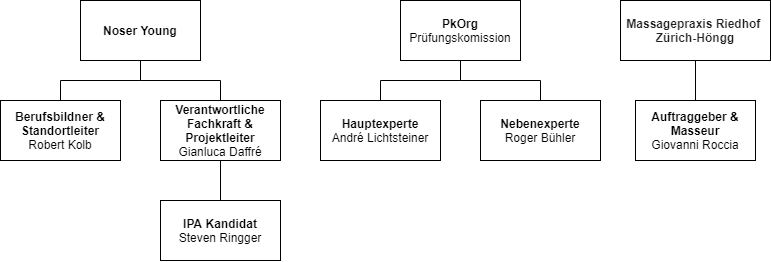


Abbildung 1: Organigramm

## Mittel und Methode

React.JS, React-Bootstrap, JavaScript, CSS, HTML5, Yarn Package Manager, Postman/SoapUI, Visual Studio Code, Cypress.io, Jest.io, StoryBook.js

## Vorkenntnisse

Über 8 Monate Erfahrung mit dem in der praktischen Arbeit angewandten Techhnologie-Stack. Der Kandidat war im Front-End eines Projektes tätig. Für die Umsetzung müssen keine neuen Kenntnisse gesammelt werden

## Vorarbeiten

Kandidat analysierte pendente Kundenanfragen, führte initiales Meeting und erhob die Anforderungen.

Zusammen mit der verantwortlichen Fachkraft wurde der Projektrahmen der praktischen Arbeit definiert.

## Neue Lerninhalte

Keine

## Arbeitsplatz



Abbildung 2: Arbeitsplatz

Vom Betrieb wurden uns die benötigten Mittel zur erfolgreichen Durchführung der IPA zur Verfügung gestellt. Der Arbeitsplatz befand sich währenddem gesamten Zeitraum aufgeräumt, entsprechend den Erwartungen des Betriebs.

Bereitgestellte Hardware:

* Bildschirm: 2 x Samsung S24C650 24 Zoll
* PC: HP Pavilion 500-330nz
* Tastatur: HP PR1101U
* Maus: Logitech M-U0007

## Sicherung der Daten

Damit keine Daten verloren gehen, werden auf mehreren Datenträgern täglich Back-ups erstellt. Der PA-Bericht wird auf dem firmeninternen Fileserver abgelegt und noch zusätzlich auf einem USB Stick gespeichert. Die gesamte Codebase wird mit Git versioniert und das Repository befindet sich auf BitBucket.

## Zeitplan

Der realisierte Zeitplan basiert auf der Projektmanagementmethodik IPERKA welche für diese Arbeit gewählt wurde. Für eine bessere Übersicht wurde jedes Inkrement sowie alle dazu gehörigen Arbeitspakete mit einer eigenen Farbe hinterlegt. Um mögliche Überzeit festhalten zu können, wurde am Ende jedes Tages jeweils ein Zeitraum von 2 Stunden freigehalten.

Zusätzlich zu den sechs IPERKA Inkrementen wurde noch ein weiteres hinzugefügt. Unter „Diverses“ werden allgemeine Arbeiten sowie Expertenbesuche aufgezeigt.

### Planung

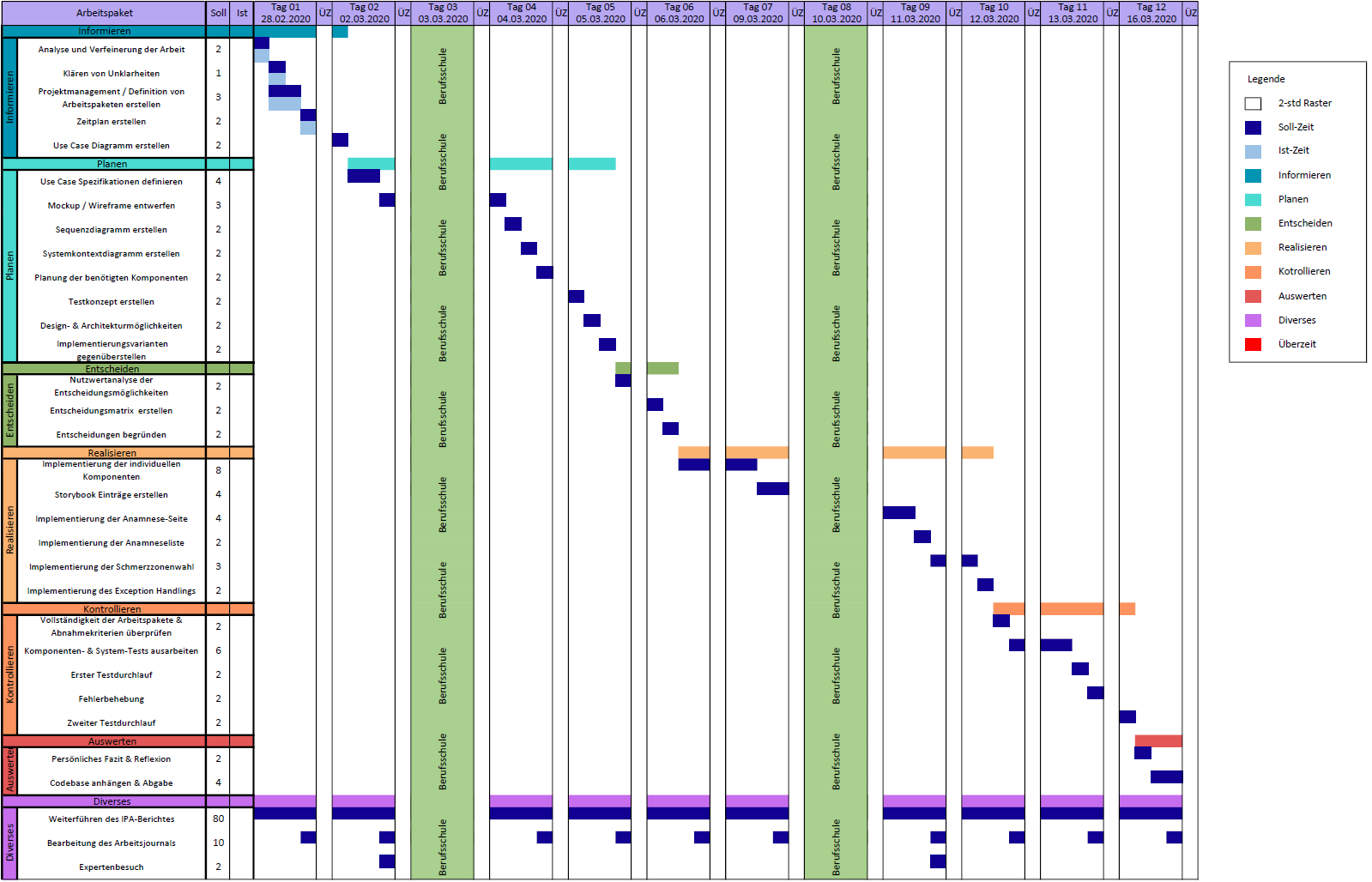


Abbildung 3: Zeitplan "Soll"

### Planung & Realität

## Sitzungsprotokolle

### Erster Expertenbesuch

|  |  |  |
| --- | --- | --- |
| Datum: 02.03.2019 | Ort: Herostrasse 12, Altstetten | |
| Uhrzeit | 14:30 bis 15:30 Uhr | |
| Teilnehmende Personen | * André Lichtsteiner * Gianlucs Daffré * Steven Ringger | (Hauptexperte)  (Verantwortliche Fachkraft)  (IPA-Kandidat) |
| Inhalt | Einzelheiten zum Ablauf der IPA besprochen  Bestehender Zeitplan angeschaut (Identifikationen für einzelne Arbeitspakete)  Wünsche bezüglich Arbeitsjournal geäussert (Tätigkeiten auch in Arbeitsjournal mit «Soll»- und «Ist»-Zeit)  Termin für die IPA-Präsentation abgemacht | |

Tabelle 1: Protokoll - Erster Expertenbesuch

### Zweiter Expertenbesuch

|  |  |  |
| --- | --- | --- |
| Datum: 11.03.2019 | Ort: Herostrasse 12, Altstetten | |
| Uhrzeit | 16:00 bis 17:00 Uhr | |
| Teilnehmende Personen | * André Lichtsteiner * Gianlucs Daffré * Steven Ringger | (Hauptexperte)  (Verantwortliche Fachkraft)  (IPA-Kandidat) |
| Inhalt |  | |

Tabelle 2: Protokoll - Zweiter Expertenbesuch

## Arbeitsjournale

In den Arbeitsjournalen werden zur Erläuterung der erreichten Ziele und der Erwähnung von ungeplanten Arbeiten sowie eventuell anfallender Überzeit. Dabei werden Erfolge und Misserfolge aufgeführt und bei den Arbeitspaketen die angewandte Vorgehensweise kritisch begutachtet.

|  |  |
| --- | --- |
| Tag 1 – 28.02.2020 | |
| Erreichte Ziele | * Erstellung der Dokumentationsstruktur * Texte von PkOrg übernommen * Sicherung der Daten erstellt * Projektmanagementmethode verfasst * Die Arbeitspakete definiert * Zeitplan erstellt |
| Ausgeführte Arbeiten | Der Hauptfokus des ersten Tages lag auf dem PA-Bericht. Ich habe die Texte von PkOrg übernehmen und diese formatieren können. Die Arbeitspakete wurden mithilfe der Aufgabenstellung und des Kriterienkatalogs erstellt und danach im neu erstellten Zeitplan eingetragen.  Die Projektmanagementmethode und die Sicherung der Daten wurden in kleineren Abschnitten beschrieben. |
| Aufgetretene Probleme | Kleinere Unklarheiten zum Inhalt des ersten Kapitels der Dokumentation. |
| Reflexion | Der Einstieg am ersten Tag der PA viel einfacher als ich am Anfang dachte. Was mich allerdings überrascht hat, war wie lange es gedauert hatte um die einzelnen Arbeitspakete zu definieren und zu beschreiben. Dafür ging hingegen das Erstellen des Zeitplans weniger lange als gedacht. Auch das Formatieren der Texte von PkOrg brauchte auch nicht ganz so viel Zeit. Die Unklarheiten konnte ich dank der FArbeit beseitigen. |
| Kritische Würdigung | Durch den heutigen Tag habe ich einen besseren Überblick über das Projekt erhalten. Die kleineren Unsicherheiten haben mir auch gezeigt, dass die FArbeit und vor allem der Kriterienkatalog ein ständiger Begleiter sein werden, ich werde ihn in Zukunft stärker beachten. |
| Wissensbeschaffung | FArbeit und Kriterienkatalog zurate gezogen für die Arbeitspakete. |
| Beanspruchte Hilfestellung | Keine |
| Überzeit | Keine |
| Vergleich mit dem Zeitplan | Aktuell gut im Zeitplan |

Tabelle 3: Arbeitsjournal - Tag 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 2 – 02.03.2020 |  | | Soll | ist |
| Tätigkeiten | | * Use Case Diagramm erstellen * Use Case Spezifikationen definieren * Expertenbesuch * Protokoll / Arbeitsjournal schreiben * Mockup / Wireframe entwerfen | 2h  4h  1h  1h  1h | 1h  5h  1h  1h |
| Ausgeführte Arbeiten | | An diesem Tag lag mein Hauptaugenmerk auf dem Definieren und Spezifizieren der Use Cases. Ich konnte alle mir erdenklichen Use Cases erfassen und ein Diagramm dafür erstellen. Neben dem war der erste Expertenbesuch. Dieser wurde festgehalten und protokolliert. | | |
| Aufgetretene Probleme | | Das Zeitmanagement war heute ein wenig daneben. | | |
| Reflexion | | Der zweite Tag verlief gut. Ich kam gut voran und war während dem Grossteil des Tages im Zeitplan. Jedoch ging das Protokollieren nach dem Expertenbesuch in der Planung unter und somit ergab sich das erste negative Delta. Ich denke jedoch, dies wird kein zu grosses Problem werden und ich kann es schnell wieder aufholen.  Der Expertenbesuch war gut und jetzt weiss ich auch, wer schlussendlich meine IPA korrigieren wird. Ich konnte einiges mitnehmen von diesem Gespräch vor allem kleinere Dinge, die gewisse Aufgaben erleichtern werden.  Auch beim Definieren und spezifizieren der Use Cases hatte ich keine sonderliche Mühe und konnte diese in einem angemessenen Zeitrahmen fertigstellen. | | |
| Kritische Würdigung | | Der heutige Tag hat mir gezeigt, dass man immer an alle möglichen Aufgaben denken muss in einer Zeitplanung. Denn ich hatte eine Stunde für den Expertenbesuch ein, jedoch vergass ich, dass noch ein Protokoll geschrieben werden sollte. Dies führte zu einer leichten Verschiebung im Zeitplan. | | |
| Wissensbeschaffung | | Keine | | |
| Beanspruchte Hilfestellung | | Keine | | |
| Überzeit | | Keine | | |
| Vergleich mit dem Zeitplan | | 1 Stunde zu langsam | | |

Tabelle : Arbeitsjournal - Tag 2

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 3 – 04.03.2020 | | | |
| Tätigkeiten | * P2: Mockup / Wireframe entwerfen * P3: Sequenzdiagramm erstellen * P4: Systemkontextdiagramm erstellen * P5: Planung der benötigten Komponenten | 3h  2h  2h  2h | 2h  2h  2h  2h |
| Ausgeführte Arbeiten | Heute konnte ich die Mockups (P2), das Sequenzdiagramm (P3) und das Systemkontextdiagramm (P4) erstellen. Da ich die Mockups nun erstellt hatte, konnte ich anschliessend die Planung von den benötigten Komponenten umsetzen (P5). Ich habe alle Schritte festgehalten und beschrieben. | | |
| Aufgetretene Probleme | Unklarheiten beim Erstellen des Sequenzdiagramm | | |
| Reflexion | An diesem dritten Tag kam ich sehr gut voran. Ich war sehr motiviert, um den am Vortag entstandenen Rückstand wieder aufzuholen. Zum Glück waren für diesen Tag mehrheitlich Mockups und Diagramme geplant. Ich konnte schon am Morgen früh meinen Rückstand wieder gut machen, da ich eigentlich drei Stunden für die Mockups eingeplant hatte, jedoch weniger Zeit benötigte, um diese fertigzustellen. Auch das Erstellen des Sequenzdiagrammes verlief gut, obwohl ich kurz im UML 2.5 Buch nachschauen musste, wie man gewisse Dinge korrekt darstellt bei dieser Art von Diagramm. Auch das Systemkontextdiagramm konnte ich ohne Probleme Umsetzen und beschreiben.  Das Erstellen der Planung für die Komponenten stellte sich jedoch als schwieriger heraus als zu Beginn gedacht. Ich konnte mich ganz gut auf die Mockups verlassen, um zu sehen welche Komponenten ich benötige für die eindeutigen Dinge. Jedoch musste ich mir auch noch zusätzlich Überlegen, welche Komponenten benötigt werden welche man nicht direkt sehen kann oder nicht irgendwo spezifisch erwähnt wurden. Zum Beispiel habe ich einen Komponenten erfasst der dafür sorgt das bei einem Renderfehler nicht einfach nur eine weisse Seite dargestellt wird. Ich wusste jedoch nur das es eine solche Möglichkeit gibt da ich einen ähnlichen Komponenten in einem früheren Projekt schon einmal gebraucht hatte. | | |
| Kritische Würdigung | Heute hatte ich keine wirklichen Probleme, jedoch bereitete mir das Ausarbeiten der einzelnen Komponenten ein wenig mühe. So wie konnte ich mich nicht mehr genau daran erinnern, wie ich genau gewisse Dinge bei einem Sequenzdiagramm aufzeigen muss. Dies konnte ich jedoch in einem Buch nachlesen. | | |
| Wissensbeschaffung | UML 2.5 Kapitel 11 «Sequenzdiagramm» | | |
| Beanspruchte Hilfestellung | Keine | | |
| Überzeit | Keine | | |
| Vergleich mit dem Zeitplan | Rückstand wieder aufgearbeitet. | | |

Tabelle : Arbeitsjournal - Tag 3

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 4 – 05.03.2020 | | | |
| Tätigkeiten | * P6: Testkonzept erstellen * P7: Design- & Architekturmöglichkeiten * P8: Implementierungsvarianten gegenüberstellen * E1: Nutzwertanalyse der Entscheidungsmöglichkeiten | 2h  2h  2h  2h | 2h  2h  2h  2h |
| Ausgeführte Arbeiten | An diesem Tag konnte ich das Testkonzept(P6) erstellen. Auch habe ich die dazugehörigen Blackbox-Tests verfasst welche sich von den Use Cases ableiten leisen. Mit dem Abschliessen von P7 und P8 habe ich nun endlich die Planungsphase hinter mir. Das letzte Arbeitspaket welches ich heute fertigstellen konnte war E1 bei welcher es sich um die Erstellung einer Nutzwertanalyse handelte. | | |
| Aufgetretene Probleme | Beim Erstellen der Blackbox-Tests habe ich gemerkt das ich mir selber das Leben erschwert habe in dem ich die Use Cases in kleinere Einheiten unterteil habe. | | |
| Reflexion | Heute Kam ich gut voran und habe nun den Planenteil abgeschlossen. Als Nächstes kommt die Entscheidungsphase, welche schneller vorbeigehen wird. Ich liege gut im Zeitplan und habe für die einzelnen Aufgaben heute ziemlich genau so lange gebraucht, wie ich es zu Beginn beim Erstellen des Zeitplans geplant hatte.  Beim Erstellen des Testkonzeptes hatte ich nicht so viel Mühe, jedoch bei den Test Cases stellte ich mir selber ein wenig das eigene Bein. Denn ich hatte die Use Cases ziemlich aufgeteilt damit man besser definieren konnte was in einem Ausnahmefall passiert. Dadurch konnte ich mich für die Abläufe der Test Cases nicht mehr direkt daran orientieren und musste für einige Fälle mehrere Use Cases zusammenfügen.  Danach kamen in der Planungsphase nur noch das Auflisten der Architektur- und Designmöglichkeiten(P7) sowie das Gegenüberstellen von Implementierungsvarianten(P8). Diese verliefen ohne Probleme. Genauso wie das erste Arbeitspaket der Entscheidungsphase. Hier konnte ich auch noch das Prinzip hinter Atomic Design sowie die grundlegenden Prinzipien erklären. | | |
| Kritische Würdigung | Dieses eine kleine Problem, das ich heute hatte, führte zu einem grösseren Aufwand als ich bei der initialen Planung dachte. Dies hatte jedoch glücklicherweise keinen Einfluss auf den Zeitplan. Ich musste einfach schneller arbeiten um doch noch im Zeitrahmen zu liegen. | | |
| Wissensbeschaffung | Atomic Design by Brad Frost | | |
| Beanspruchte Hilfestellung | Keine | | |
| Überzeit | Keine | | |
| Vergleich mit dem Zeitplan | Ich liege gut im Zeitplan | | |

Tabelle : Arbeitsjournal - Tag 4

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 5 – 06.03.2020 | | | |
| Tätigkeiten | * E2: Entscheidungsmatrix erstellen * E3: Entscheidungen begründen * R1: Implementierung der Individuellen Komponenten | 2h  2h  4h | 0h  2h  6h |
| Ausgeführte Arbeiten | Am heutigen Tag konnte ich die Entscheidungsphase mit E2 und E3 abschliessen und mich nun der Realisierung widmen. Der Start der Realisierungsphase verlief ohne Probleme. Und dank einer fehl Überlegung beim Erstellen des Zeitplans standen mir heute zwei Stunden mehr zur Verfügung. | | |
| Aufgetretene Probleme | Heute entstand ein kleines Problem da ich beim Erstellen des Zeitplans und der Arbeitspakete nicht weit genug überlegte das eine Entscheidungsmatrix und eine Nutzwertanalyse fast das Gleiche sind. | | |
| Reflexion | Dieser fünfte Tag verlief sehr gut und ich konnte einiges erledigen. Ich bin froh darüber das nun ein grossteil der ganzen Planung und Theorie nun abgeschlossen ist. Nun geht es ans Realisieren der Applikation. Leider oder vielleicht auch zum Glück hatte ich beim Erstellen der Arbeitspakete und des Zeitplans einen Überlegungsfehler gemacht, wodurch ich nun im Zeitplan zwei Stunden voraus bin. Es war kein grosser Fehler jedoch Schrieb ich auf, dass ich eine Nutzwertanalyse und eine Entscheidungsmatrix machen werde ohne, dass mir dabei in den Sinn kam, dass diese fast dasselbe sind. Ich empfinde, als ziemlich unnötig zweimal das fast gleiche Diagramm zu erstellen, wobei das eine weniger aussagekräftig ist als das andere.Deshalb habe ich mich dazu entschieden die Entscheidungsmatrix wegzulassen und nur eine Nutzwertanalyse zu erstellen. Durch diese Entscheidung habe ich nun mehr Zeit in der Realisierungsphase, welche ich wahrscheinlich auch gut gebrauchen werden kann. Ich konnte durch diesen Fehler auch noch etwas lernen und zwar das man sich auch schon zu Beginn ein wenig mehr Zeit lassen sollte um zumindest kurz alles ein weites mal auf seine Plausibilität zu prüfen.  In der Realisierung kam ich schon sehr gut voran. Ich konnte schon einen Teil der Komponenten implementieren und stehe dadurch immer noch gut im Zeitplan. Ich bin auch optimistisch, dass ich in den verbleibenden zwei Stunden welch ich für diesen Task R1 eingeplant hatte, alles beenden kann wenn ich am Montag ausgeruht und mit neuer Energie zur Arbeit komme. | | |
| Kritische Würdigung | Ich hätte diesen Fehler sicher vermeiden können, jedoch bin ich auch nicht gerad unfroh darüber mehr Zeit zur Verfügung zu haben. Aber ich weiss jetzt für die Zukunft das eine Nutzwertanalyse und eine Entscheidungsmatrix so ziemlich dasselbe sind. | | |
| Wissensbeschaffung | React-Bootstrap Dokumentation | | |
| Beanspruchte Hilfestellung | Keine | | |
| Überzeit | Keine | | |
| Vergleich mit dem Zeitplan | Zwei Stunden voraus | | |

Tabelle : Arbeitsjournal - Tag 5

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 6 – 09.03.2020 | | | |
| Tätigkeiten | * R1: Implementierung der Individuellen Komponenten * R2: Storybook Einträge erstellt * R3: Implementierung der Anamnese-Seite | 4h  4h  0h | 2h  4h  2h |
| Ausgeführte Arbeiten | Am heutigen Tag habe ich die Implementierung der einzelnen Kompoenten(R1) abschliessen können. Danach machte ich mich daran die Storybook Einträge zu den erstellten Komponenten zu erstellen. Die beinhaltete auch noch einige kleinere Anpassungen damit die Komponenten mit dem Storybook funktionierten. Danach begann ich noch mit der Implementierung der Anamnese-Seite(R3). | | |
| Aufgetretene Probleme | Ich hatte keine grösseren Probleme am heutigen Tag. Allerdings musste ich ein paar der erstellten Komponenten überarbeiten damit sie auch mit dem Storybook zusammenspielen. | | |
| Reflexion | Heute konnte ich die Implementierung der einzelnen Komponenten abschliessen. Diese Aufgabe beschäftigte mich allerdings nur für die ersten zwei Stunden des Tages anstelle bis zum Mittag wie es eigentlich geplant war, aber da ich immer noch zwei Stunden voraus bin konnte ich einige Dinge zusätzlich machen. Die Implementierung verlief soweit ohne Probleme und ich konnte mich danach daran machen die einzelnen Komponeten im Storybook zu erfassen.  Dies war wieder einmal sehr zeitaufwändig und das Dokumentieren unterschätzt man noch gerne einmal, jedoch konnte ich den Zeitplan einhalten. Bei dieser Aufgabe musste ich auch noch einige kleinere Dinge an den erstellten Komponeten anpassen damit sie gut mit dem Storybook zusammenspielen.  Die letzte Aufgabe die ich noch hatte war mit der Implementation der Anamnese-Seite zu beginnen. Ich kam schon ein gutes Stück voran, allerdings besteht bei diesem Arbeitspaket(R3) eine gute Chance das ich dafür länger brauchen werde als ich eigentlich geplant hatte. Aber das werden wir am nächsten Tag sehen. | | |
| Kritische Würdigung | Dieser Tag verlief erstaunlich gut. Ich hätte sicher noch weiterkommen können als ich jetzt bin so dass ich meinen Vorsprung nicht zu schnell verliere. Auch hätte ich bei der Implementierung der einzelnen Komponenten schon bis zu einem gewissen Teil an das Storybook denken können. Dann hätte ich heute weniger anpassen müssen. | | |
| Wissensbeschaffung | React-Bootstrap Dokumentation – Error Boundary  NPM-Package – React-Image-Mapper | | |
| Beanspruchte Hilfestellung | Keine | | |
| Überzeit | Keine | | |
| Vergleich mit dem Zeitplan | Zwei Stunden voraus | | |

Tabelle : Arbeitsjournal - Tag 6

|  |  |  |  |
| --- | --- | --- | --- |
| Tag 7 – 11.03.2020 | | | |
| Tätigkeiten | * R3: Implementierung der Anamnese-Seite * R4: Implementierung der Anamnese-Liste * R5: Implementierung der Schmerzzonenwahl * D3: Zweiter Expertenbesuch * D1 + D2: Arbeitsjournal und Sitzungsprotokoll | 4h  2h  1h  1h | 2h  2h  2h |
| Ausgeführte Arbeiten | An diesem Tag habe ich zuerst die Implementierung der Anamnese-Seite abgeschlossen(R3). Das nächste Arbeitspaket war das Implementieren der Anamnese-Liste. Vor dem zweiten Expertenbesuch begann ich noch mit der Implementierung der Schmerzzonenwahl(R5). | | |
| Aufgetretene Probleme |  | | |
| Reflexion |  | | |
| Kritische Würdigung |  | | |
| Wissensbeschaffung |  | | |
| Beanspruchte Hilfestellung |  | | |
| Überzeit |  | | |
| Vergleich mit dem Zeitplan |  | | |

# Projekt

## Kurzfassung

Die nachfolgende Kurzfassung dient dazu, Aufschluss zugeben über die wesentlichen Bestandteile der vorliegenden Arbeit. Dies beinhaltet die Ausgangssituation, die Umsetzung sowie das endgültige Ergebnis. Detailliertere Informationen können Sie dem Kapitel 1.1 ff entnehmen.

### Kurze Ausgangssituation

Die Applikation die während der PA realisiert wird dient zur Vereinfachung des Kundenmanagements und Anamnese-Prozesses innerhalb der Massagepraxis Riedhof Zürich-Höngg. Bislang werden die Kunden und die dazugehörigen Anamnesen in physischer Form verwaltet. Die Geschäftsleitung sieht nun vor dies zu optimieren in dem die gesamte Verwaltung über eine Web-Applikation geschehen soll, welche während den jeweiligen Sitzungen von den Therapeuten bequem über das iPad abgerufen werden können.

### Umsetzung

### Ergebnis

## Informieren

### Abgrenzung und Deklaration gemachter Vorarbeiten

Während der Vorarbeiten wurde mit dem Kunden seine Vision und Anforderungen besprochen. Die erhobenen Anforderungen enthielten Funktional- und Nicht-Funktionale-Anforderungen. Nebenbei wurden auch noch domänenspezifische Anforderungen aufgenommen, welche auch zum besseren Verständnis der Zieldomäne dient. Der bisherige Arbeitsprozess wurde angeschaut und mögliche Verbesserungsmöglichkeiten besprochen, die zur Optimierung von eben diesem führen können. Das Ziel von diesem Initialenmeeting bestand darin, sich einen Überblick zu verschaffen und Verständnislücken sowie Unklarheiten zu schliessen. Für die weitere Analyse wurde ein existierendes Dokument entgegengenommen, um dieses in einem zusätzlichen Prozess zu analysieren, welches eine zentrale Rolle im momentanen Arbeitsprozess spielt.

Eine sehr schlichte Grundstruktur besteht bereits. Dies beinhaltet eine Navigationsleiste so wie im package.json hinzugefügten Dependencies, welche dafür benötigt wurden.

### Analyse und Verfeinerung der Arbeit

Durch die Analyse wird das Verständnis der Domäne gefördert, da sie grundsätzlich implementationsunabhängig ist. Ein Teil der wurde bereits während der Vorarbeit durchgeführt. In Rahmen dieser Arbeit wird diese jedoch noch verfeinert und mit einem Use Case Diagramm belegt.

In der Planungsphase werden die detaillierten Spezifikationen des erstellen Use Case Diagramm als Fully Dressed Use Cases erfasst. Des Weiteren werden noch implementationsabhängige Diagramme erstellt. Dies beinhaltet zum einen ein Sequenzdiagramm, für den grundlegenden Ablauf der Applikation und ein Systemkontexdiagramm um das Umfeld der Applikation genauer aufzuzeigen.

### Projektmanagementmethodik

Für diese Arbeit erachte ich IPERKA als ideal, denn die Anforderungen wurden zu Beginn klar definiert und da ich während diesem Projekt alleine arbeiten werde.

IPERKA unterbindet eine voreilige Umsetzung durch das Fördern einer methodischen Vorgehensweise und einer genauen Planung sowie Entscheidungsphase. Mit jedem Inkrement setzt man sich intensiv mit dem Produkt auseinander. Es sieht eine kontinuierliche Verbesserung vor anhand von mehreren Einzelschritten und bei der Realisierung hat man eine genaue Vorstellung vom Endprodukt

### Use Case Diagramm

Das Use Case Diagramm dient dazu, funktionale Anforderungen die an die Applikation gestellt werden visuell darzustellen. Diese werden während der Analyse, meistens anhand einer kurzen Beschreibung erhoben. Während der Planung / Designs werden diese genauer Spezifiziert.

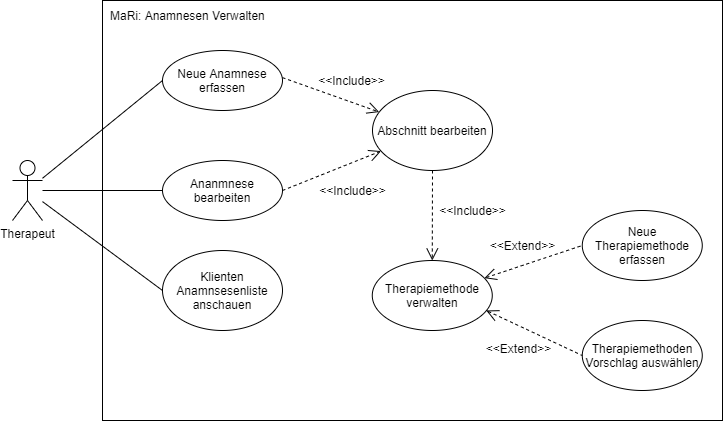


Abbildung 4: Use Case Diagramm

### Definition Arbeitspakete

|  |  |
| --- | --- |
| 1.1 Phase: Informieren [Analyse und Verfeinerung der Arbeit] | |
| Beschreibung | Dieses Arbeitspaket dient dem Grundaufbau der Dokumentation. Besonderen Fokus wird dabei auf die Ausführung des ersten Kapitels gelegt. Alle Anforderungen werden wie definiert von PkOrg übernommen und verfeinert. Auch werden die Methodiken und Diagramme, die während der Analyse verwendet werden, werden aufgezeigt und geplant. |
| Zu erledigende Aufgabe | * Grundstruktur der Dokumentation * Sicherung der Daten / Arbeitsplatz Gestaltung * Verfassen des ersten Teils der Kurzfassung * Deklaration der Vorarbeiten * Vertieftere Analyse und Verfeinerung der Arbeit |
| Soll-Zeit | 2 Stunden |

Tabelle 9: Arbeitspaket - Analyse und Verfeinerung der Arbeit

|  |  |
| --- | --- |
| 1.2 Phase: Informieren [Klären von Unklarheiten] | |
| Beschreibung | Dieses Arbeitspaket dient der Klärung von Unklarheiten in Bezug auf Aufgabenstellung, Vorgaben oder anderen Problematiken (z. B. Bewertungskriterien). Auch wird der Nachweis der Wissenbeschaffung vorbereitet. |
| Zu erledigende Aufgabe | * Klären von Fragen die bei Arbeitspaket 1.1 aufkamen * Genauers durchlesen der Bewertungskriterien * Aufbereitung der Sitzungsprotokolle und Arbeitsjournale bzgl. Wissensbeschaffung |
| Soll-Zeit | 1 Stunde |

Tabelle 10: Arbeitspaket - Klären von Unklarheiten

|  |  |
| --- | --- |
| 1.3 Phase: Informieren [Projektmanagement / Definition von Arbeitspaketen erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient der Wahl der Projektmanagementmethodik sowie der Definition der jeweiligen Arbeitspakete. |
| Zu erledigende Aufgabe | * Wahl der Projektmanagementmethodik * Erläuterung der getroffenen Entscheidung * Definition, Ausarbeitung und Zuordnung von Arbeitspaketen |
| Soll-Zeit | 3 Stunden |

Tabelle 11: Arbeitspaket - Projektmanagement / Definition von Arbeitspaketen erstellen

|  |  |
| --- | --- |
| 1.4 Phase: Informieren [Zeitplan erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient zur Erstellung des Zeitplanes sowie der Einteilung der einzelnen Arbeitspakete. Die erwarteten Eigenschaften werden von den Bewertungskriterien übernommen. |
| Zu erledigende Aufgabe | * Aufbau des Zeitplanes definieren * Zuweisung der Arbeitspakete |
| Soll-Zeit | 2 Stunden |

Tabelle 12: Arbeitspaket - Zeitplan erstellen

|  |  |
| --- | --- |
| 1.5 Phase: Informieren [Use Case Diagramm erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient der Erstellung eines Use Case Diagrammes. Dies ist ein weiterer Teil der Analyse. Das Diagramm muss dem UML 2.5 Standard entsprechen. |
| Zu erledigende Aufgabe | * Definition der Use Cases * Definition der Akteure * Visualisierung des Use Case Diagrammes |
| Soll-Zeit | 2 Stunden |

Tabelle 13: Arbeitspaket - Use Case Diagramm erstellen

|  |  |
| --- | --- |
| 2.1 Phase: Planen [Use Case Spezifikationen definieren] | |
| Beschreibung | Dieses Arbeitspaket dient der Spezifizierung der Use Cases die in Arbeitspaket definiert wurden. Diese müssen als «Fully Dressed Use Cases» nach Larman erfasst werden |
| Zu erledigende Aufgabe | * Fully Dressed Use Cases erfassen |
| Soll-Zeit | 4 Stunden |

Tabelle 14: Arbeitspaket - Use Case Spezifikationen definieren

|  |  |
| --- | --- |
| 2.2 Phase: Planen [Mockup / Wireframe entwerfen] | |
| Beschreibung | Dieses Arbeitspaket dient zum entwerfen und Designen der einzelnen Ansichten der Applikation mit Hilfe des Tools Figma. Dabei werden die grundlegenden Designprinzipien berücksichtigt. |
| Zu erledigende Aufgabe | * Mockups für alles Ansichten werden erstellt |
| Soll-Zeit | 3 Stunden |

Tabelle 15: Arbeitspaket - Mockup / Wireframe entwerfen

|  |  |
| --- | --- |
| 2.3 Phase: Planen [Sequenzdiagramm erstellen] | |
| Beschreibung | Diese Arbeitspaket dient der Erstellung eines Sequenzdiagrammes des Grundlegendenablaufes einer Anamnese, nach dem UML 2.5 Standard. |
| Zu erledigende Aufgabe | * Sequenzdiagrammes erstellt |
| Soll-Zeit | 2 Stunden |

Tabelle 16: Arbeitspaket - Sequenzdiagramm erstellen

|  |  |
| --- | --- |
| 2.4 Phase: Planen [Systemkontextdiagramm erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient dem Zweck den Systemkontext zu erfassen und in einem Diagramm darzustellen. |
| Zu erledigende Aufgabe | * Systemkontext definieren * Systemkontextdiagramm visualisieren |
| Soll-Zeit | 2 Stunden |

Tabelle 17: Arbeitspaket - Systemkontextdiagramm erstellen

|  |  |
| --- | --- |
| 2.5 Phase: Planen [Planung der benötigten Komponenten] | |
| Beschreibung | Dieses Arbeitspaket dient der Planung und Ausarbeitung der für die Applikation benötigten Komponenten. Diese werden aufgelistet und genau spezifiziert. |
| Zu erledigende Aufgabe | * Definition der zu verwendenden Komponenten * Komponenten auflisten und genau spezifizieren |
| Soll-Zeit | 2 Stunden |

Tabelle 18: Arbeitspaket - Planung der benötigten Komponenten

|  |  |
| --- | --- |
| 5.2 Phase: Kontrollieren [Testkonzept erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient dazu das Testkonzept zu erstellen. |
| Zu erledigende Aufgabe | * Testkonzept erstellen * Rahmenbedingungen festlegen * Vorbedingungen und Methodik definieren * Testfälle erstellen * Testprotokoll vorbereiten |
| Soll-Zeit | 2 Stunden |

Tabelle 19: Arbeitspaket - Testkonzept erstellen

|  |  |
| --- | --- |
| 2.6 Phase: Planen [Design- & Architekturmöglichkeiten] | |
| Beschreibung | Dieses Arbeitspaket dient dazu verschiedene Architektur und Design Möglichkeiten aufzuzeigen und diese zu evaluieren. |
| Zu erledigende Aufgabe | * Auflisten der verschiedenen Möglichkeiten * Vor- und Nach-Teile auflisten |
| Soll-Zeit | 2 Stunden |

Tabelle 20: Arbeitspaket - Design- & Architekturmöglichkeiten

|  |  |
| --- | --- |
| 2.7 Phase: Planen [Implementierungsvarianten gegenüberstellen] | |
| Beschreibung | Dieses Arbeitspaket dient dem Aufzeigen und dem herleiten von gewissen Lösungsansätzen zu einer aufkommenden Problematik. |
| Zu erledigende Aufgabe | * Aufzeigen einer aufkommenden Problematik * Erläuterung gewisser Lösungsansätzen |
| Soll-Zeit | 2 Stunden |

Tabelle 21: Arbeitspaket - Implementierungsvarianten gegenüberstellen

|  |  |
| --- | --- |
| 3.1 Phase: Entscheiden [Nutzwertanalyse der Entscheidungsmöglichkeiten] | |
| Beschreibung | Dieses Arbeitspaket dient der Erstellung einer Nutzwertanalyse für die verschiedenen Varianten. |
| Zu erledigende Aufgabe | * Erstellen der Nutzwertanalyse |
| Soll-Zeit | 2 Stunden |

Tabelle 22: Arbeitspaket - Nutzwertanalyse der Entscheidungsmöglichkeiten

|  |  |
| --- | --- |
| 3.2 Phase: Entscheiden [Entscheidungsmatrix erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient dem Erstellen einer Entscheidungsmatrix für verschiedenen Lösungsansätze. |
| Zu erledigende Aufgabe | * Entscheidungsmatrix erstellen |
| Soll-Zeit | 2 Stunden |

Tabelle 23: Arbeitspaket - Entscheidungsmatrix erstellen

|  |  |
| --- | --- |
| 3.3 Phase: Entscheiden [Entscheidungen begründen] | |
| Beschreibung | Dieses Arbeitspaket dient dazu die in den Arbeitspaketen 3.1 und 3.2 getroffenen Entscheidungen zu begründen. |
| Zu erledigende Aufgabe | * Erklärung der gefällten Entscheidung |
| Soll-Zeit | 2 Stunden |

Tabelle 24: Arbeitspaket - Entscheidungen begründen

|  |  |
| --- | --- |
| 4.1 Phase: Realisieren [Implementierung der individuellen Komponenten] | |
| Beschreibung | Dieses Arbeitspaket dient der Implementierung der in Arbeitspaket 2.5 definierten Komponenten. Mit entsprechenden Kommentaren und typisierten Parametern versehen. |
| Zu erledigende Aufgabe | * Komponenten implementieren * typisierten Parameter |
| Soll-Zeit | 8 Stunden |

Tabelle 25: Arbeitspaket - Implementierung der individuellen Komponenten

|  |  |
| --- | --- |
| 4.2 Phase: Realisieren [Storybook Einträge erstellen] | |
| Beschreibung | Dieses Arbeitspaket dient dazu die Einträge im Storybook zu erstellen. Zur einfacheren Übersicht für eine spätere Wiederverwendung. |
| Zu erledigende Aufgabe | * Alle Komponenten ausführlich im Storybook dokumentieren |
| Soll-Zeit | 4 Stunden |

Tabelle 26: Arbeitspaket - Storybook Einträge erstellen

|  |  |
| --- | --- |
| 4.3 Phase: Realisieren [Implementierung der Anamnese-Seite] | |
| Beschreibung | Dieses Arbeitspaket dient der Implementation der Anamnese-Seite. Wobei die benötigten Komponenten eingebunden und die entsprechenden Daten von der REST API geladen werden müssen |
| Zu erledigende Aufgabe | * Einbindung der einzelnen Komponenten * API-Calls implementiert |
| Soll-Zeit | 4 Stunden |

Tabelle 27: Arbeitspaket - Implementierung der Anamnese-Seite

|  |  |
| --- | --- |
| 4.4 Phase: Realisieren [Implementierung der Anamneseliste] | |
| Beschreibung | Dieses Arbeitspaket dient Der Implementation der Anamneselist welche die bestehenden Anamnesen im Profil eines Kunden anzeigt. Es kann eine bestehende Anamnese bearbeitet oder eine Neue erstellt werden. |
| Zu erledigende Aufgabe | * Tabelle wird mit den korrekten Anamnesen gefüllt * Aktion zum Erstellen einer neuen Anamnese |
| Soll-Zeit | 2 Stunden |

Tabelle 28: Arbeitspaket - implementierung der Anamneseliste

|  |  |
| --- | --- |
| 4.5 Phase: Realisieren [Implementierung der Schmerzzonenwahl] | |
| Beschreibung | Dieses Arbeitspaket dient der Implementierung des Bildes von einem menschlichen Körper der vordefinierte Punkte beinhaltet, welche ausgewählt werden können. |
| Zu erledigende Aufgabe | * Punkte auf dem Bild korrekt anzeigen * Richtige Daten laden wenn ein Punkt ausgewählt wird |
| Soll-Zeit | 3 Stunden |

Tabelle 29: Arbeitspaket - Implementierung der Schmerzzonenwahl

|  |  |
| --- | --- |
| 4.6 Phase: Realisieren [Implementierung des Exception Handlings] | |
| Beschreibung | Dieses Arbeitspaket dient zur Implementation des Exception Handlings. Es werden Systemfehler korrekt abgefangen und der Nutzer sieht keine Seite mit Fehlermeldungen. |
| Zu erledigende Aufgabe | * Exception Handling erstellen |
| Soll-Zeit | 2 Stunden |

Tabelle 30: Arbeitspaket - Implementierung des Exception Handlings

|  |  |
| --- | --- |
| 5.1 Phase: Kontrollieren [Vollständigkeit der Arbeitspakete & Abnahmekriterien überprüfen] | |
| Beschreibung | Dieses Arbeitspaket dient dazu, dass erstellte Produkt mit dem in der Aufgabenstellung definierten Ergebnis und den Bewertungskriterien seitens PkOrg zu vergleichen. Gegebenenfalls werden noch Änderungen vorgenommen. |
| Zu erledigende Aufgabe | * Applikation wird mit Aufgabenstellung und definierten Anforderungen abgeglichen. * Überprüfung der Bewertungskriterien. Vergleich mit Dokumentation der PA. |
| Soll-Zeit | 2 Stunden |

Tabelle 31: Arbeitspaket - Vollständigkeit der Arbeitspakete & Abnahmekriterien überprüfen

|  |  |
| --- | --- |
| 5.3 Phase: Kontrollieren [Komponenten- & System-Tests ausarbeiten] | |
| Beschreibung | Dieses Arbeitspaket dient der Definition und Ausarbeitung der Testfälle. Die Applikation wird mittels System- und Komponenten-Tests getestet. |
| Zu erledigende Aufgabe | * Erstellung der Komponententests * Erstellung der Systemtests |
| Soll-Zeit | 6 Stunden |

Tabelle 32: Arbeitspaket - Komponenten- & System-Tests ausarbeiten

|  |  |
| --- | --- |
| 5.4 Phase: Kontrollieren [Erster Testdurchlauf] | |
| Beschreibung | Dieses Arbeitspaket dient der Durchführung der ersten Runde an Tests. Die dabei entstehenden Resultate werden dokumentiert und mit den erwarteten Resultaten abgeglichen. |
| Zu erledigende Aufgabe | * Durchführung der Testfälle * Dokumentieren und abgleichen der Resultate |
| Soll-Zeit | 2 Stunden |

Tabelle 33: Arbeitspaket - Erster Testdurchlauf

|  |  |
| --- | --- |
| 5.5 Phase: Kontrollieren [Fehlerbehebung] | |
| Beschreibung | Dieses Arbeitspaket dient der Verbesserung der Applikation. Es werden die im Arbeitspaket 5.4 aufgekommenen Fehler behoben. Fehlerhafter Code wird ggf. verbessert und/oder ergänzt. |
| Zu erledigende Aufgabe | * Aufgekommene Fehler beheben * Delta dokumentieren |
| Soll-Zeit | 4 Stunden |

Tabelle 34: Arbeitspaket - Fehlerbehebung

|  |  |
| --- | --- |
| 5.6 Phase: Kontrollieren [Zweiter Testdurchlauf] | |
| Beschreibung | Dieses Arbeitspaket dient dem erneuten Ausführen der fehlgeschlagenen Testfälle nach den Verbesserungen. Die Resultate werden abgeglichen und dokumentiert. |
| Zu erledigende Aufgabe | • Durchführung der Testfälle  • Dokumentieren und abgleichen der Resultate |
| Soll-Zeit | 2 Stunden |

Tabelle 35: Arbeitspaket - Zweiter Testdurchlauf

|  |  |
| --- | --- |
| 6.1 Phase: Auswerten [Persönliches Fazit & Reflexion] | |
| Beschreibung | Dieses Arbeitspaket dient dem Verfassen eines persönlichen Fazits resp. einer persönlichen Reflexion. Die gemachten Erfahrungen und die Arbeit werden reflektiert, analysiert und dokumentiert. |
| Zu erledigende Aufgabe | * Verfassung einer persönlichen Reflexion * Analyse der Vorgehensweise * Verbesserungsmöglichkeiten aufzeigen |
| Soll-Zeit | 2 Stunden |

Tabelle 36: Arbeitspaket - Persönliches Fazit & Reflexion

|  |  |
| --- | --- |
| 6.2 Phase: Auswerten [Codebase anhängen & Abgabe] | |
| Beschreibung | Dieses Arbeitspaket dient der Erstellung des Anhangs. Die Codebase muss kopiert, formatiert und angehängt werden. Am Ende wird alles noch einmal überprüft und dann auf PkOrg abgegeben. |
| Zu erledigende Aufgabe | * Aufbereiten der Codebase und Anhängen * Finale Überprüfung * Abgabe auf PkOrg |
| Soll-Zeit | 4 Stunden |

Tabelle 37: Arbeitspaket - Codebase anhängen & Abgabe

|  |  |
| --- | --- |
| 7.1 Phase: Diverses [Weiterführen des IPA-Berichtes] | |
| Beschreibung | Dieses Arbeitspaket dient der Weiterarbeit an dem PA-Bericht. Dies ist eine zweckmässige Tätigkeit, welche deshalb die ganze Arbeit abdeckt. |
| Zu erledigende Aufgabe | * Weiterführung des PA-Berichts |
| Soll-Zeit | 80 Stunden |

Tabelle 38: Arbeitspaket - Weiterführen des IPA-Berichtes

|  |  |
| --- | --- |
| 7.2 Phase: Diverses [Bearbeitung des Arbeitsjournals] | |
| Beschreibung | Dieses Arbeitspaket dient der Erstellung des täglich zuführenden Arbeitsjournales. Dies ist eine zweckmässige Tätigkeit und deckt deshalb die ganze Arbeit ab. |
| Zu erledigende Aufgabe | * Tägliches verfassen des Arbeitsjournals |
| Soll-Zeit | 10 Stunden |

Tabelle 39: Arbeitspaket - Bearbeitung des Arbeitsjournals

|  |  |
| --- | --- |
| 7.3 Phase: Diverses [Expertenbesuch] | |
| Beschreibung | Dieses Arbeitspaket dient der Vorbereitung und Durchführung des Expertenbesuchs. |
| Zu erledigende Aufgabe | * Vorbereiten von Fragen und Unklarheiten * Ausdrucken des Zeitplans * Protokollieren des Gespräches und Verbesserungsvorschläge des HEX |
| Soll-Zeit | 2 Stunden |

Tabelle 40: Arbeitspaket - Expertenbesuch

## Planen

### Use Case Spezifikationen

|  |  |
| --- | --- |
| Use Case ID: 1 | Use Case: Klienten Anamnesenliste anschauen |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Kunde befindet sich auf der Klienten-Übersicht-Seite |
| Kurzbeschrieb | Auf der Klienten-Übersichts-Seite wählt der Therapeut den entsprechenden Klienten aus. Im Profil der Klienten werden die erfassten Anamnesen angezeigt. |
| Hauptablauf | 1. Therapeut klickt auf Klient 2. Profil wird geladen |
| Nachbedingung | Bereits erfasste Anamnesen werden dargestellt |
| Alternative Ablaufschritte | - |

Tabelle 41: Use Case - Klienten Anamneseliste anschauen

|  |  |
| --- | --- |
| Use Case ID: 2 | Use Case: Neue Anamnese erfassen |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Anamneseabschnitt im Klienten Profil wird dargestellt |
| Kurzbeschrieb | Therapeut erfasst einen neue Anamnese für einen Klienten |
| Hauptablauf | 1. Klickt Button „Neue Anamnese Erfassen“ 2. Aktuelles Datum bereits eingefüllt 3. Klickt auf den gewünschten Abschnitt 4. Erfasst Daten (siehe Use Case Nr. 4) 5. Drückt „Speichern“-Button |
| Nachbedingung | Daten werden an das Backend gesendet und gespeichert |
| Alternative Ablaufschritte | 4.2 Seite verlassen ohne zu speichern  4.3 Validation inkorrekt  4.4 Speichern fehlgeschlagen |

Tabelle 42: Use Case - Neue Anamnese erfassen

|  |  |
| --- | --- |
| Use Case ID: 3 | Use Case: Anamnnese bearbeiten |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Anamneseabschnitt im Klienten Profil wird dargestellt |
| Kurzbeschrieb | Therapeut wählt aus der Liste eine bereits existierende Anamnese. Nimmt die benötigten Änderungen vor. Therapeut speichert die Anamnese. |
| Hauptablauf | 1. Wählt die zu bearbeitende Anamnese aus 2. Klickt auf den Gewünschten Abschnitt 3. Bearbeitet Daten (siehe Use Case Nr. 4) 4. Drückt den „Speichern“-Button |
| Nachbedingung | Daten werden an das Backend gesendet und gespeichert |
| Alternative Ablaufschritte | 4.2 Seite verlassen ohne zu speichern  4.3 Validation inkorrekt  4.4 Speichern fehlgeschlagen |

Tabelle 43: Use Case - Anamnese bearbeiten

|  |  |
| --- | --- |
| Use Case ID: 4 | Use Case: Abschnitt bearbeiten |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Abschnitt wurde ausgewählt |
| Kurzbeschrieb | In dem Abschnitt werden die Felder ausgefüllt, validiert und zwischen gespeichert |
| Hauptablauf | 1. Schmerzlevel auf der Skala wird ausgewählt 2. Felder werden mit Text befüllt, wenn nötig 3. Eingaben werden validiert 4. Abschnitt wird zwischen gespeichert |
| Nachbedingung |  |
| Alternative Ablaufschritte | * 1. Abschnittwechsel ohne zwischen speichern   2. Seite verlassen ohne zu speichern   3. Validation inkorrekt   4. Speichern fehlgeschlagen |

Tabelle 44: Use Case - Abschnitt bearbeiten

|  |  |
| --- | --- |
| Use Case ID: 4.1 | Alternative Flow: Abschnittwechsel ohne zwischen speichern |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Änderungen wurden am Abschnitt vorgenommen ohne zu speichern |
| Kurzbeschrieb | Therapeut klickt auf einen anderen Abschnitt ohne vorher zu speichern. |
| Hauptablauf | 1. Therapeut klickt auf anderen Abschnitt 2. Dialog erscheint ob er wirklich wechseln will    1. Ja: Änderungen werden verworfen    2. Nein: bleibt auf dem gleichen Abschnitt |
| Nachbedingung | - |
| Alternative Ablaufschritte | - |

Tabelle 45: Alternative Flow - Abschnittwechsel ohne zwischen speichern

|  |  |
| --- | --- |
| Use Case ID: 4.2 | Alternative Flow: Seite verlassen ohne zu speichern |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Änderungen an der Anamnese vorgenommen ohne zu Speichern |
| Kurzbeschrieb | Benutzer will von der Seite weg navigieren ohne zu speichern |
| Hauptablauf | 1. Therapeut verlässt die aktuelle Seite 2. Dialog erscheint ob er wirklich wechseln will    1. Ja: Änderungen werden verworfen    2. Nein: bleibt auf dem gleichen Abschnitt |
| Nachbedingung | - |
| Alternative Ablaufschritte | - |

Tabelle 46: Alternative Flow - Seite verlassen ohne zu speichern

|  |  |
| --- | --- |
| Use Case ID: 4.3 | Alternative Flow: Validation inkorrekt |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Daten wurden in ein Eingabefeld eingegeben |
| Kurzbeschrieb | Die Validationsfehlermeldung wird rot unter dem Eingabefeld angezeigt. |
| Hauptablauf | 1. Validationsfehlermeldung wird unter Eingabefeld Angezeigt 2. Daten werden nicht abgesendet |
| Nachbedingung | - |
| Alternative Ablaufschritte | - |

Tabelle 47: Alternative Flow - Validation Inkorrekt

|  |  |
| --- | --- |
| Use Case ID: 4.4 | Alternative Flow: Speichern fehlgeschlagen |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Änderungen an der Anamnese vorgenommen und auf speichern geklickt |
| Kurzbeschrieb | Fehler beim Abspeichern der Daten im Backend |
| Hauptablauf | 1. Daten werden an Backend gesendet 2. Fehler tritt auf 3. Therapeut bekommt Mitteilung dass das Speichern fehlgeschlagen ist |
| Nachbedingung | - |
| Alternative Ablaufschritte | - |

Tabelle 48: Alternative Flow - Speichern fehlgeschlagen

|  |  |
| --- | --- |
| Use Case ID: 5 | Use Case: Therapiemethode verwalten |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Therapeut ist am Abschnitt bearbeiten |
| Kurzbeschrieb | Therapeut fügt Therapiemethode zu Abschnitt hinzu. |
| Hauptablauf | 1. „Therapiemethode hinzufügen“-Button wird gedrückt 2. Formular und Vorschläge erscheinen 3. Erfassen (siehe Use Case 6) 4. Vorschlag auswählen (siehe Use Case 7) 5. „Therapiemethode Speichern“-Button drücken 6. Therapiemethode wird oberhalb von Button angezeigt |
| Nachbedingung | Therapiemethode wurde zwischen gespeichert |
| Alternative Ablaufschritte | - |

Tabelle 49: Use Case - Therapiemethode verwalten

|  |  |
| --- | --- |
| Use Case ID: 6 | Use Case: Neue Therapiemethode erfassen |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Formular und Vorschläge werden Angezeigt |
| Kurzbeschrieb | Formular mit 2 Feldern wird ausgefüllt und Validiert. Beim Speichern wird im Backend eine neue Therapiemethode erstellt. |
| Hauptablauf | 1. Titel ausfüllen 2. Beschrieb ausfüllen 3. Validation der Eingaben 4. Therapiemethode Speicher |
| Nachbedingung | Therapiemethode wurde hinzugefügt |
| Alternative Ablaufschritte | * 1. Validation inkorrekt   2. Speichern fehlgeschlagen |

Tabelle 50: Use Case - Neue Therapiemethode erfassen

|  |  |
| --- | --- |
| Use Case ID: 7 | Use Case: Therapiemethoden Vorschlag auswählen |
| Akteur (Primär) | Therapeut |
| Akteur (Sekundär) | - |
| Vorbedingungen | Formular und Vorschläge werden Angezeigt |
| Kurzbeschrieb | Ein Vorschlag wird ausgewählt und man sieht welcher ausgewählt wurde. |
| Hauptablauf | 1. Therapiemethode wird angeklickt 2. Therapiemethode ist gehighlighted |
| Nachbedingung | Therapiemethode ist ausgewählt |
| Alternative Ablaufschritte | - |

Tabelle 51: Use Case - Therapiemethoden Vorschlag auswählen

### Mockups / Wireframes

Die Mockups wurden mithilfe des Onlinetools «Figma» realisiert. Es wurden für jede Hauptansicht jeweils eine Desktop- sowie eine Tablet-Version erstellt. Diese Mockups finden Sie auf den folgenden Seiten, links befindet sich jeweils die Desktop-Version und daneben die Tablet-Version.

Ich habe Mockups erstellt und keine Wireframes, da Mockups mehr mit dem Design zu tun haben und die Aufgabe der Wireframes einschliesst. Denn Wireframes dienen in erster Linie zur Positionierung der einzelnen Elemente und um zu definieren, wie genau eine Seite aufgebaut ist. Dies geschieht bei Mockups auch nur erweitert man diese noch in dem man die Elemente so Gestalt, wie man diese umzusetzen gedenkt. Dies bietet den Entwicklern und auch dem Kunden ein genaueres Bild, wie das Endergebnis aussehen wird.

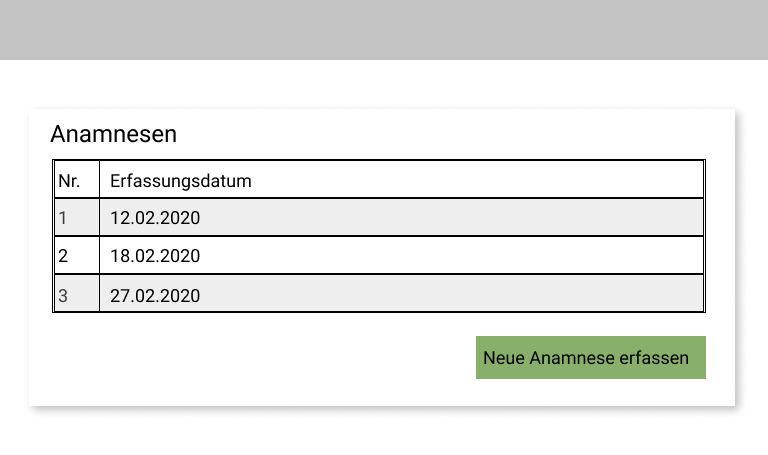
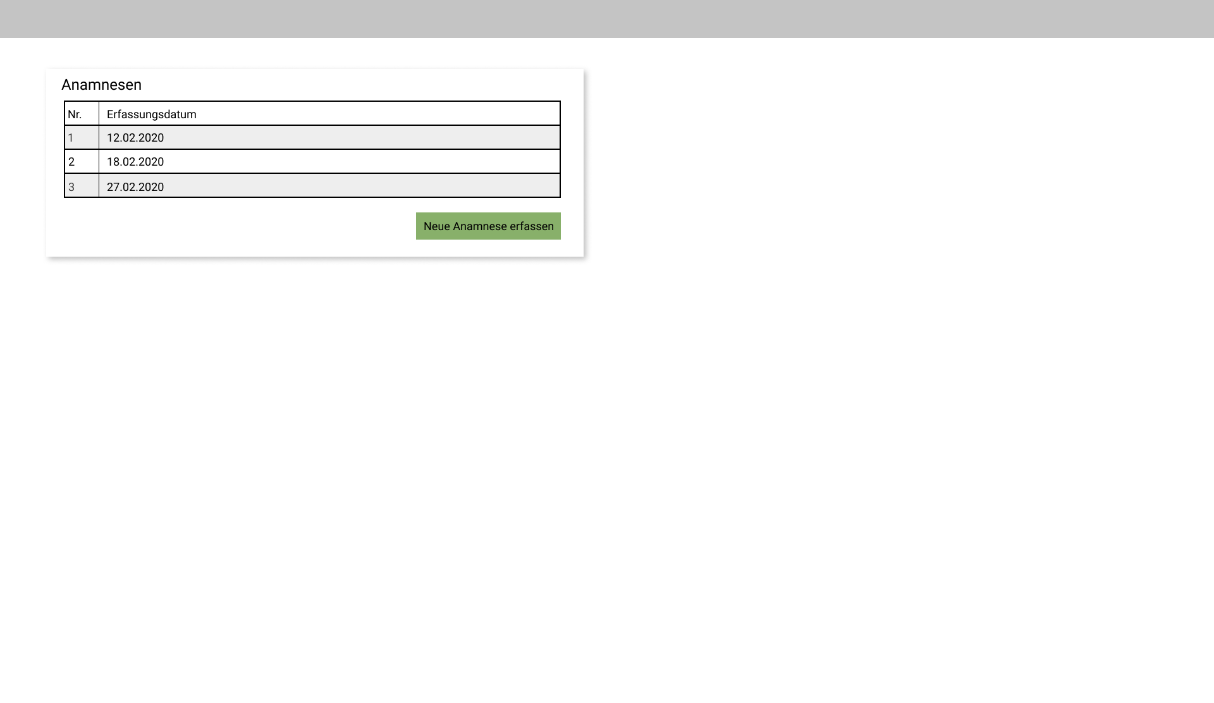


Abbildung : Mockup - Liste der Anamnesen im Profil des Klienten (Desktop)

Abbildung 6: Mockup - Liste der Anamnesen im Profil des Klienten (Tablet)

Bei diesen Abbildungen sieht man die Liste der Anamnesen welche für einen Klienten bereits erfasst wurden. Diese befinden sich in einem eigenen Abschnitt welcher im Profil des Klienten angezeigt wird.

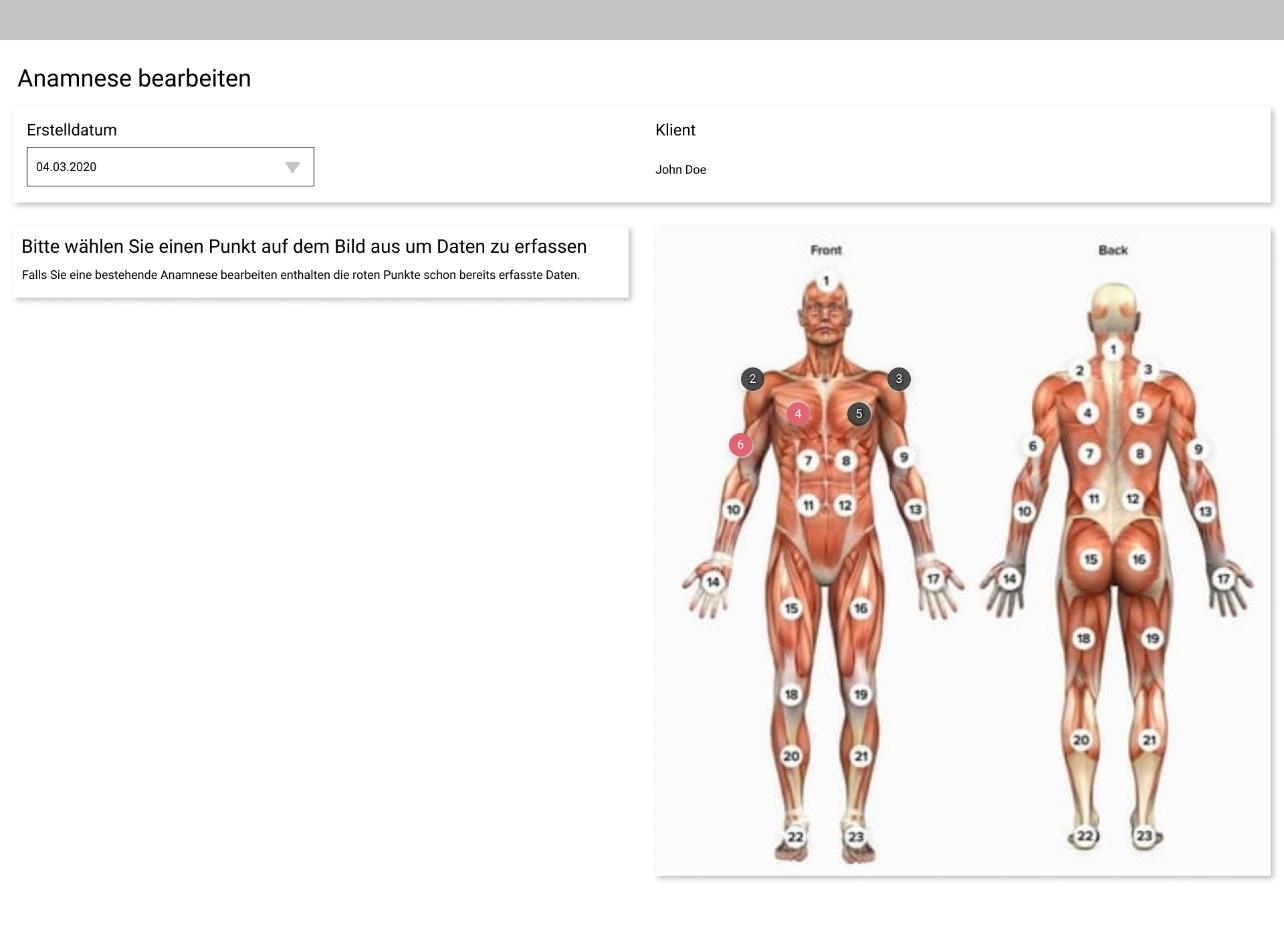
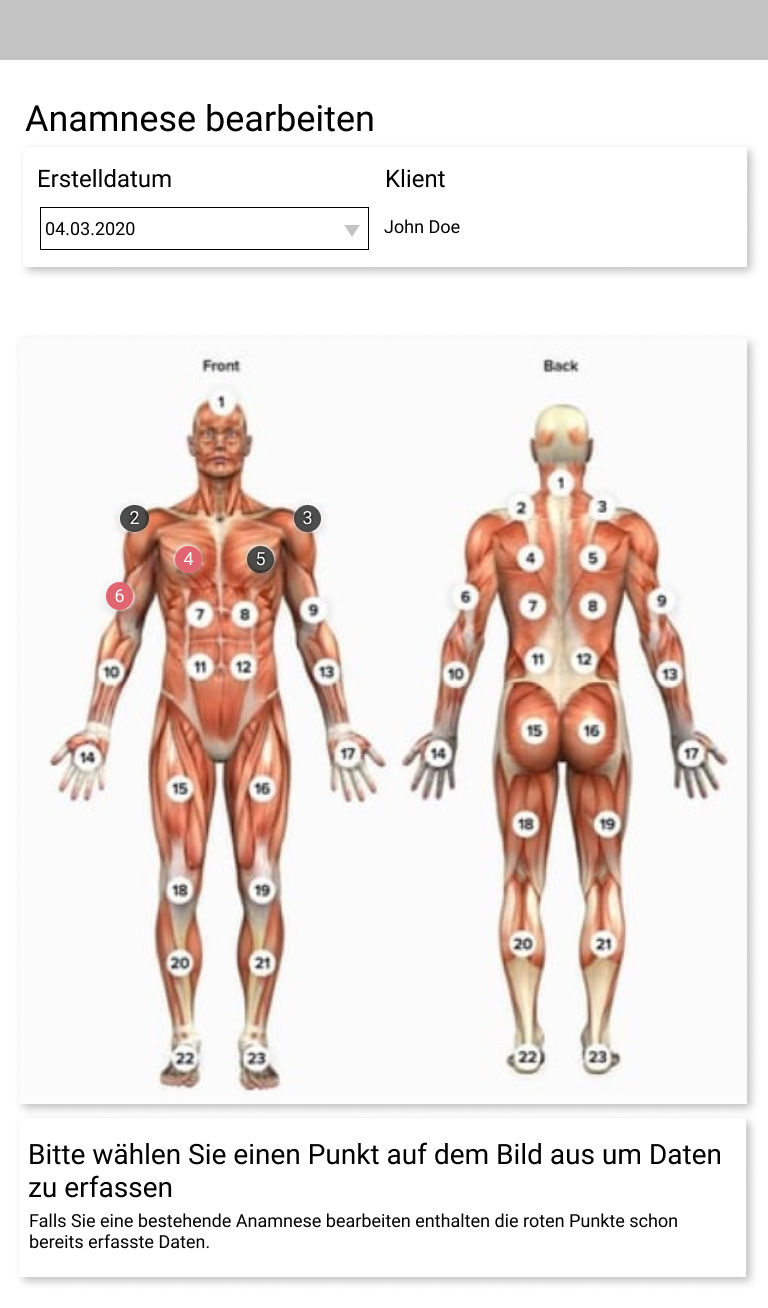
Hier sieht man die grundlegende Ansicht der Anamnese-Maske. Diese hat einzelne Abschnitte welche optisch von einander getrennt sind, da sie nicht direkt zusammenspielen. Auch die Punkt am Körper die bereits erfasste Daten enthalten, sofern man eine existierende Anamnese bearbeitet, werden optisch hervorgehoben für einen schneller überblick. Zum hervorheben wurde eine etwas markantere Farbe gewählt, welche einem besser ins auffällt.

Abbildung : Mockup - Anamnese bearbeiten / erstellen Maske ohne Daten (Desktop)

Abbildung : Mockup - Anamnese bearbeiten / erstellen Maske ohne Daten (Tablet)

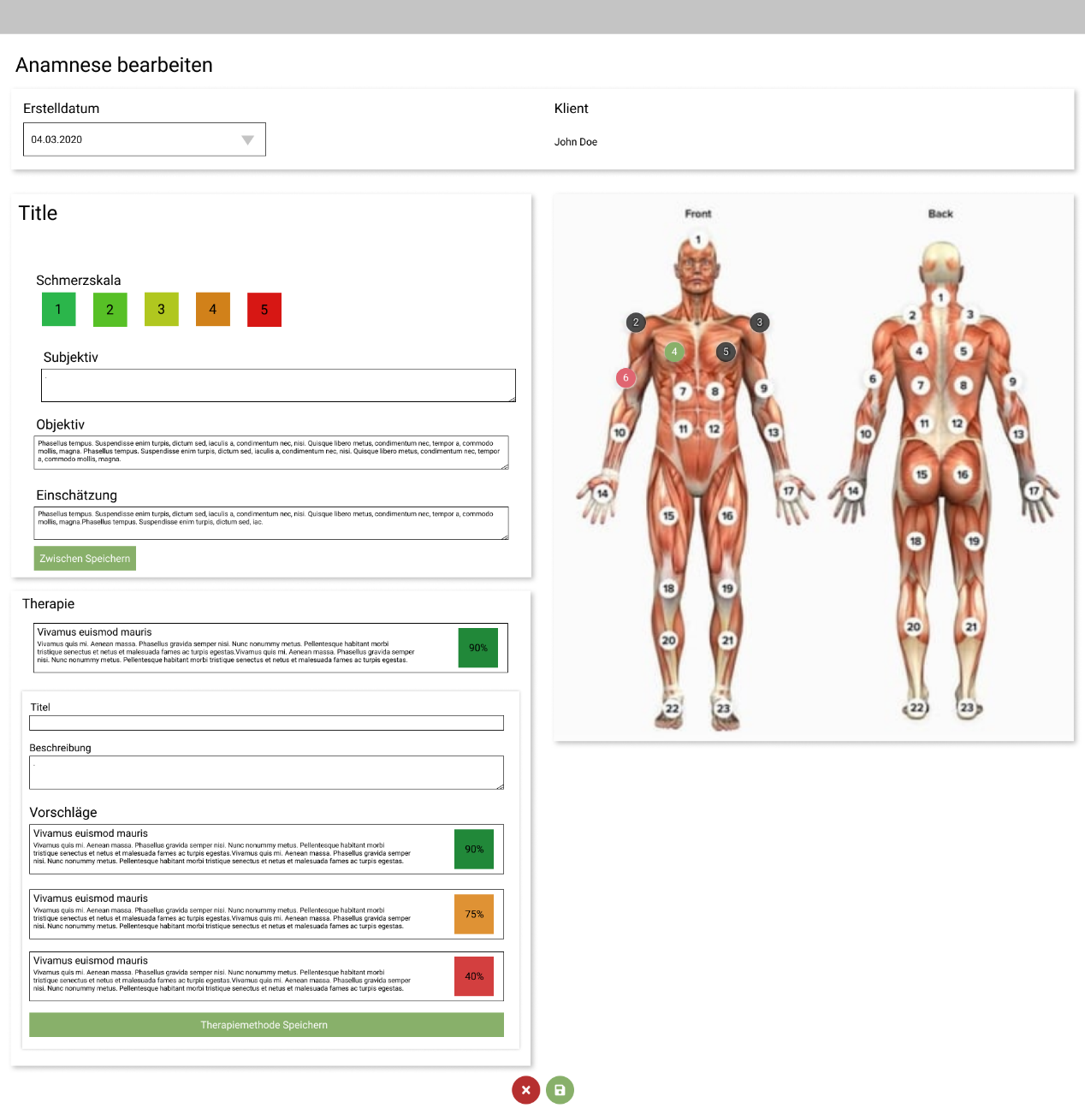
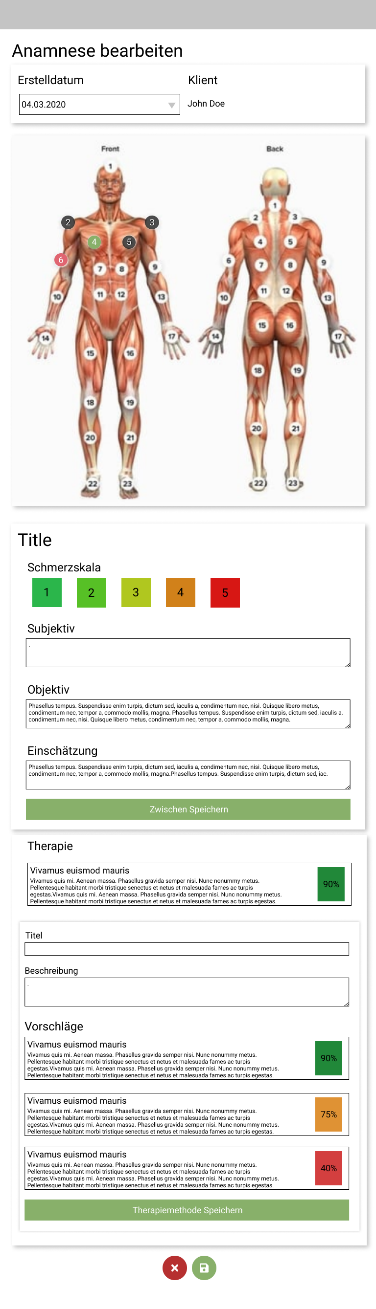


Abbildung : Mockup - Anamnese bearbeiten / erstellen Maske mit Daten (Tablet)

Abbildung : Mockup - Anamnese bearbeiten / erstellen Maske mit Daten (Desktop)

Die letzten Abbildungen zeigen die Ansicht einer Anamnese mit einem ausgewählten Abschnitt. Die einzelnen Sektionen sind abgetrennt, wenn sie zusammengehören. Auch die Überschriften der Felder befinden sich oberhalb der eingebe Felder und sind nicht zu weit entfernt so das eine klare Zugehörigkeit ersichtlich ist.

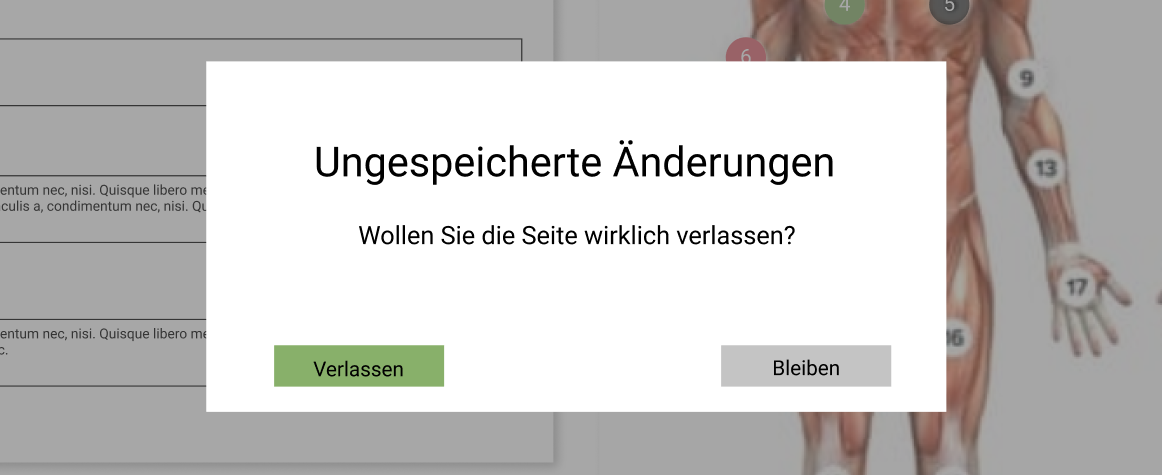


Abbildung : Mockup - Verlassen der Seite bei ungespeicherten Änderungen

Wenn der Benutzer wegnavigieren will, ohne zu speichern, erscheint eine Meldung, welche ihn fragt, ob er dies wirklich tun will.

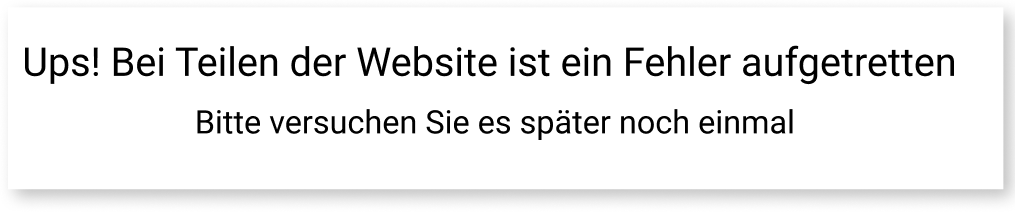


Abbildung : Mockup - Alternativer Komponent bei Render-Fehler

Wenn ein Komponent einen Render-Fehler aufweist und somit die ganze Seite blockieren würde, wird eine alternative angezeigt welche den Benutzer darauf aufmerksam mach das ein Problem vorliegt.

### Sequenzdiagramm

Ein Sequenzdiagramm zeigt zeitlich geordnete Objektinteraktionen auf. Es zeigt die an der Sequenz beteiligten Objekte auf, sowie die Reihenfolge der Nachrichten, die zwischen diesen ausgetauscht werden. Durch das Darstellen der einzelnen Prozesse oder Objekte als vertikale Linien (Lebenslinien) und den gesendeten Nachrichten als horizontale Pfeile, wird die grafische Darstellung des Ablaufes wesentlich vereinfacht.

In dem nachfolgenden Sequenzdiagramm wird der Ablauf des Bearbeitens einer Anamnese abgebildet:

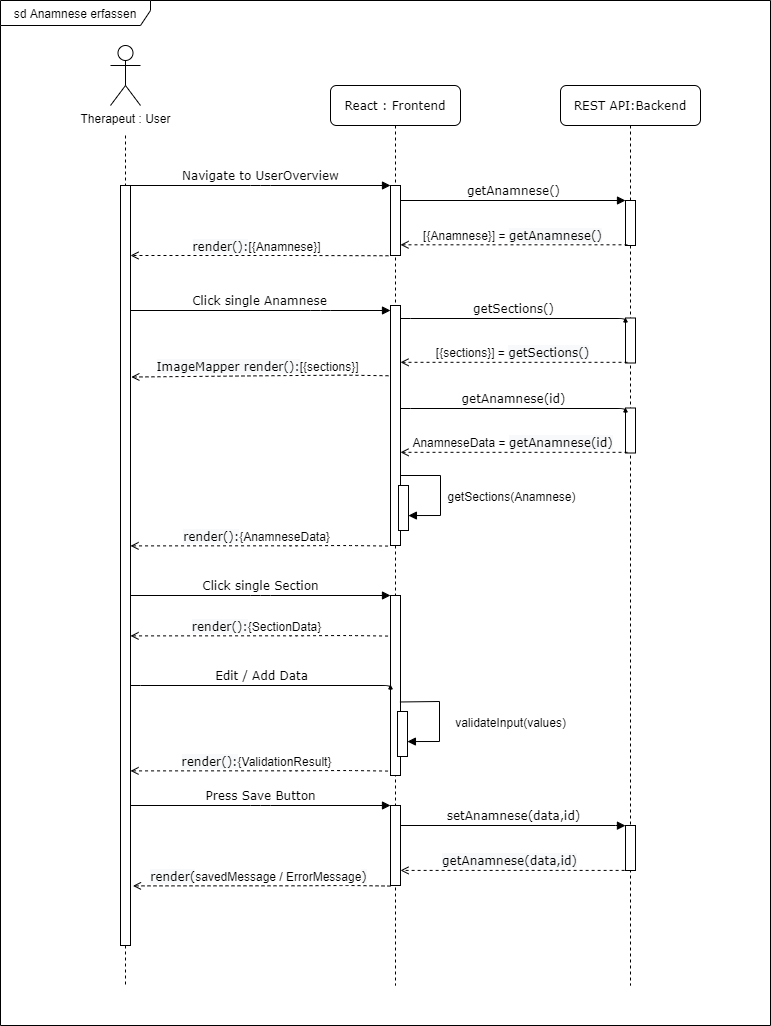


Abbildung 13: Sequenzdiagramm - bearbeiten einer Anamnese

### Systemkontextdiagramm

Das Systemkontextdiagramm dient der Darstellung des Systemkontexts. Dieser zeigt auf welche materiellen und immateriellen Aspekte eine Beziehung zu dem System haben. Dies hilft dabei, eine Übersicht zu erstellen was berücksichtigt werden muss und was nicht.

Im Systemkontextdiagramm fallen alle nicht zu berücksichtigenden Aspekte in die irrelevante Umgebung und werden nicht direkt dargestellt. Alle relevanten Aspekte befinden sich entweder im Systemkontext oder im SUD.

Die Aspekte eines Systems werden in den Anforderungen spezifiziert, welche man in drei verschiedene Arten kategorisieren kann:

**Funktional**

Funktionale Anforderungen sind in der Essenz Use Cases. Dies sind spezifische Funktionen, die genau nach den definierten Spezifikationen implementiert werden müssen. Eine Liste der Use Cases findet man unter dem Kapitel 2.3.1.

Nicht-Funktional

Diese Art der Anforderungen, dienen dazu einen gewissen Qualitätsstandard vorzugeben. Im Gegensatz aber zu den funktionalen Anforderungen besteht keine Vorgabe wie oder mit was diese Anforderung erfüllt werden kann.

Beispiele für nicht-funktionale Anforderungen:

* Die Sicherheitsstandards müssen dem neusten Stand entsprechend sein.
* Die Applikation sollte über das Internet erreichbar sein.
* Die Applikation muss durchgehend verfügbar sein.

**Domänenspezifisch**

Domänenspezifische Anforderungen sind Anforderungen, die speziell für diese Domäne gelten. In dieser Arbeit ist die Domäne die Massagepraxis und mögliche domänenspezifische Anforderungen können folgende sein:

* Anbindung zu vorhandenen Schnittstellen von Krankenkassen
* Anbindung der Applikation an die Schnittstelle zu Treatwell für die einfachere Terminverwaltung

Nachfolgend das oberflächliche Systemkontextdiagramm (die detaillierteren Informationen zum SUD findet man im «Planen»-Abschnitt):

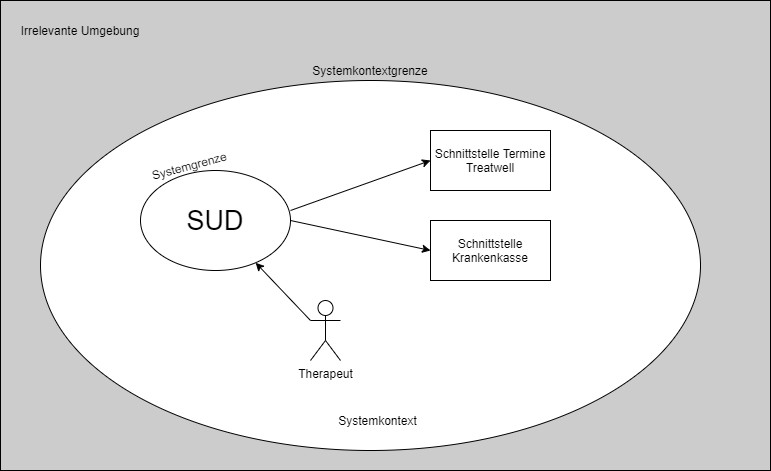


Abbildung 14: Systemkontextdiagramm

### Planung der benötigten Komponenten

Nachfolgend werden alle Komponenten aufgelistet welche durch die in Kapitel 2.3.2 erstellten Mockups ersichtlich wurden. Den einzelnen Komponenten werden jeweils einen Namen und eine Beschreibung gegeben. Zudem werden sie jetzt schon einer Struktur im Atomic Design zu geordnet.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Atomic |
| Text Button | Button der einfachen Text anzeigt. Kann über die Props entweder links, rechts oder zentriert angezeigt werden. Beinhaltet ein onClick-Event. | Atom |
| Icon Button | Runder Button der anstelle von Text ein Icon enthält. Beinhaltet ein onClick-Event. | Atom |
| Custom Table | Generiert eine Tabelle in dem man die Gewünschten Header und Daten mitgibt. Bietet die Option das eine Zeile angeklickt werden kann. | Atom |
| Error Boundary | Umschliesst einen anderen Komponenten, wenn dieser ein Renderfehler aufwirft kann eine Alternative angezeigt werden. | Atom |
| Input Field | Einfaches Input Field Komponent dem man ein Label mitgeben kann und das Validiert wird. | Atom |
| Radio Button | Custom gestylter Radio Button dem man ein Label, einen Namen und ein value mitgibt. | Atom |
| Section | Ein Abschnitt der Applikation kann durch diesen Komponenten visuell abgegrenzt werden. | Atom |
| PopUp alert | PopUp-Meldung für wenn z.B. Daten im Backend gespeichert wurden. | Atom |
| Success Rate Display | Kleiner Komponent der bei einer Therapiemethode die Erfolgsrate in der entsprechenden Farbe anzeigt. | Atom |
| Text Area | Text Area Eingabefeld das mit einem Label erstellt werden kann und Validiert wird. | Atom |
| Alt Component | Ein Alternativer Komponent der angezeigt werden kann falls ein anderer Komponent einen Renderfehler aufweist | Atom |
| Toast | Ein kleines Benachrichtigungselement welches am Rande angezeigt werden kann. Z.B. «Anamnese wurde gespeichert» | Atom |
| Date Picker | Ein einfacher Date-Picker um ein Datum zu wählen. | Molecule |
| Button Group | Die Button Group besteht entweder aus einem Icon Button sofern man sich nicht im Editier-Modus befindet ansonsten aus zwei Buttons. Einer zum Speichern und einer um abzubrechen. | Molecule |
| Image Mapper | Der Image Mapper ist dafür verantwortlich, dass das Bild des Körpers mit den korrekten Punkten versehen wird. | Molecule |
| Modal | Ein einfaches Modales Element, welches den Benutzer warnt falls er die Seite verlassen will ohne zu speichern. | Molecule |
| Pain Scale | Dieser Komponenten besteht aus mehreren Radio Buttons die speziell dargestellt werden. Man kann eine Individuelle Anzahl an Radio Buttons mitgeben sowie einen Farbverlauf erstellen lassen. | Molecule |
| Therapy Method | Ein Komponenten der eine Einzelne Therapiemethode darstellt und mit dem Success Rate Display Komponenten verknüpft. | Molecule |
| Anamnese Form | Das Formular welches zum Erstellen und bearbeiten einer Anamnese gebraucht wird. | Organism |
| Therapy Method Selecter | Lässt den Benutzer aus mehreren Therapiemethoden ein einzelnes Auswählen. Ist dafür verantwortlich die therapiemethoden-Vorschläge vom backend zu laden. | Organism |
| Anamnese | Seite die alle Komponenten zusammenträgt welche zum Erfassen oder Bearbeiten einer Anamnese benötigt werden. Lädt die allgemeinen Daten und ist für das Speichern der neuen Daten verantwortlich. | Page |
| Customer | Mock-Seite eines Klienten um die Tabelle mit den Anamnesen darstellen zu können. | Page |

Tabelle : Benötigte Komponenten

### Testkonzept

Die Blackbox-Test werden von den in Schritt 2.3.1 erstellten Use Cases abgeleitet und während der Planungsphase definiert.

**Umfeld**

* Computer: HP Pavilion 500-330nz – Windows 10 Education
* Code Editor: Visual Studio Code – 1.42.1
* Framework: React - ^16.12.0
* Building-Tool: Yarn - v1.21.1
* REST-API Mock: SoapUI - 5.5.0
* Cypress: ^4.0.1
* Jest: 25.1.0

**Vorbedingungen für alle Tests**

* Gemocktes Backend ist gestartet über SoapUI
* Alle Node-Modules sind installiert
* Die Applikation läuft

**Methoden**

Die einzelnen Ansichten des Frontends, die Abläufe der Use Cases und die einzelnen Komponenten getestet. Die Ansichten des Frontends und die Abläufe der Use Cases werden mithilfe von System-/GUI- und Blackbox-Test durchgeführt.

* System-/GUI-Tests werden mit Hilfe von Cypress durchgeführt
* Komponenten-Tests werden mit Hilfe von Jest erstellt
* Blackbox-Test werden von Hand durchgeführt

### Test Cases – Blackbox-Testing

|  |  |
| --- | --- |
| Test Case ID: 1 | Klienten Anamnesenliste anschauen |
| Use Case ID | 1 |
| Vorbedingung | - |
| Ablauf | 1. Im Browser wird die URL: „<http://localhost:3000/customer/1>“ eingegeben |
| Erwartetes Ergebnis | Liste mit Anamnesen wird Angezeigt |

Tabelle : Test Case 1 - Klienten Anamneseliste anschauen

|  |  |
| --- | --- |
| Test Case ID: 2 | Neue Anamnese erfassen |
| Use Case ID | 2 |
| Vorbedingung | Liste der Anamnesen wird im Klienten-Profil angezeigt |
| Ablauf | 1. Klick auf Button „Neue Anamnese Erfassen“ 2. Auf dem Körperbild Punkt 1 auswählen 3. Schmerzlevel 4 auf der Skala auswählen 4. In Felder Objektiv, Subjektiv und Einschätzung jeweils „Test“ eingeben 5. Zwischen Speichern drücken 6. Auf Button „Therapiemethode Hinzufügen“ klicken 7. Therapievorschlag 1 anklicken 8. Button „Therapiemethode Speichern“ klicken 9. Speichern Knopf drücken |
| Erwartetes Ergebnis | Benachrichtigung wird angezeigt das die Anamnese erfolgreich gespeichert wurde. |

Tabelle : Test Case 2 - Neue Anamnese erfassen

|  |  |
| --- | --- |
| Test Case ID: 3 | Anamnese bearbeiten |
| Use Case ID | 3 |
| Vorbedingung | Liste der Anamnesen wird im Klienten-Profil angezeigt |
| Ablauf | 1. Klick auf existierende Anamnese 1 in 2. Auf dem Körperbild Punkt 1 auswählen 3. Schmerzlevel 4 auf der Skala auswählen 4. In Felder Objektiv, Subjektiv und Einschätzung jeweils „Test“ eingeben 5. Zwischen Speichern drücken 6. Auf Button „Therapiemethode Hinzufügen“ klicken 7. Therapievorschlag 1 anklicken 8. Button „Therapiemethode Speichern“ klicken 9. Speichern Knopf drücken |
| Erwartetes Ergebnis | Benachrichtigung wird angezeigt das die Anamnese erfolgreich gespeichert wurde. |

Tabelle : Test Case 3 - Anamnese bearbeiten

|  |  |
| --- | --- |
| Test Case ID: 4 | Validation inkorrekt |
| Use Case ID | 4.3 |
| Vorbedingung | Abschnitt von Anamnese wurde ausgewählt und ist bearbeitbar |
| Ablauf | 1. Text in Feld eingeben der mehr als 1000 Zeichen enthält. |
| Erwartetes Ergebnis | Fehlermeldung wird rot unter dem Eingabefeld angezeigt. Daten können nicht gespeichert werden. |

Tabelle 56: Test Case 4 - Validation inkorrekt

|  |  |
| --- | --- |
| Test Case ID: 5 | Abschnittwechsel ohne zwischen speichern |
| Use Case ID | 4.1 |
| Vorbedingung | Änderungen wurden am Abschnitt vorgenommen ohne zu speichern |
| Ablauf | 1. Benutzer klickt auf anderen Abschnitt |
| Erwartetes Ergebnis | Dialog erscheint ob er wirklich die wechseln will ohne zu speichern |

Tabelle : Test Case 5 - Abschnittwechsel ohne zwischen speichern

|  |  |
| --- | --- |
| Test Case ID: 6 | Seite verlassen ohne zu speichern |
| Use Case ID | 4.2 |
| Vorbedingung | Änderungen wurden an der Ananmense vorgenommen ohne zu speichern |
| Ablauf | 1. Benutzer klickt um die Seite zu verlassen |
| Erwartetes Ergebnis | Dialog erscheint ob er wirklich die wechseln will ohne zu speichern |

Tabelle : Test Case 6 - Seite verlassen ohne zu speichern

|  |  |
| --- | --- |
| Test Case ID: 8 | Neue Therapiemethode erfassen |
| Use Case ID | 6 |
| Vorbedingung | Anamnese bearbeiten oder erstellen. Abschnitt wurde gewählt und Daten wurden im Oberenteil zwischengespeichert. |
| Ablauf | 1. Im Titelfeld wird „Test“ eigegeben 2. Bei der Beschreibung wird „Test“ eingegeben 3. Es wird auf „Therapiemethode Hinzufügen“ geklickt |
| Erwartetes Ergebnis | Neu erfasste Therapiemethode wird oberhalb des Buttons dargestellt |

Tabelle : Test Case 7 - Neue Therapiemethode erfassen

### Design- & Architekturmöglichkeiten

In diesem Kapitel werde ich kurz auf das Thema Atomic Design eingehen. Dieses Pattern wurde in der Aufgabenstellung verlangt, um die Übersicht über die Codebase nicht zu behalten. Denn React beschäftigt sich in erster Linie mit dem Viewlayer und es werden keine spezifischen Architekturpatterns vorgeschrieben. Auch scheiden sich die Geister darüber, ob man einen Service Layer verwenden soll oder nicht.

Kommen wir nun zum Atomic Design Pattern. Dieses schreibt vor, die Komponenten in Atome, Moleküle, Organismen und Seiten aufzuteilen. Wobei ein Atom die kleinste Art von Komponenten darstellt, wie ein Atom eben, und eine Seite die grösste Art eines Komponenten ist.

Ein Atom ist zum Beispiel ein Button, ein Eingabefeld oder ein Label. Diese sind sehr klein, implementieren nicht viel Logik und werden selten alleine gebraucht. Sie sind so klein, dass man sie nicht noch weiter Aufteilen kann da sie oft nur aus einem HTML-Tag bestehen.

Die nächst grösser Einheit die Moleküle sind ein wenig komplexer aufgebaut. Sie bestehen meistens aus mehreren Atomen. Als Beispiel könnte man ein Komponenten erstellen, welcher ein Eingabefeld mit einem Label verbindet oder ein Löschen- und Speicher-Butten beinhaltet.

Aus diesen Molekülen kann man nun Organismen zusammensetzen. Ein Organismus definiert sich in erster Line dadurch, dass er einen konkreten Teil einer Seite abbildet. Es ist die Verbindung von mehreren Molekülen und Atomen um einen bestimmten Komponenten zu erhalten. Als Beispiel könnte man ein Formular nennen, welches aus Eingabefeldern(Molekülen) und einem Button(Atom) besteht.

Nachfolgend noch eine visuelle Darstellung des Atomic Design Patterns:

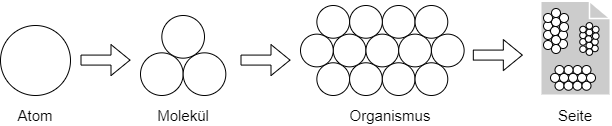


Abbildung : Atomic Design Pattern

### Gegenüberstellung der Implementierungsvarianten

**Class vs functional**

React bietet zwei grundlegende Arten wie man einen Komponenten implementieren kann. Die eine Art ist klassenbasiert und die andere funktionenbasiert.

Beim klassenbasierten Ansatz wird, wie der Name bereits erwähnt, eine Klasse erstellt welche einen eigenen State besitz und Zugriff auf Lifecycle-Methoden hat. Neben dem enthält eine Klasse auch eine Rendermethode, welche dazu dient den entstandenen Komponenten darzustellen.

Im Gegensatz dazu steht der funktionale Ansatz. Bei diesem ist ein Komponent nicht mehr als eine Funktion und der Return-Wert von eben dieser ist, was auf dem Bildschirm schlussendlich angezeigt wird.

**Selber schreiben vs NPM Package vs inspirieren lassen von existierendem Komponenten**

Es gibt sehr viele öffentlich zur Verfügung stehende NPM-Packages und eines dieser entspricht zumindest zum Teil genau dem, was man in seinem Projekt braucht. Jedoch wurde dieses zuletzt vor einem Jahr aktualisiert oder man will nun etwas Spezifisches ändern, geht das nicht so einfach. Nun hat man mehrere Möglichkeiten dieses Problem anzugehen, entweder man schreibt alles selber so, dass man nicht auf dritte angewiesen ist, man benutzt das Package trotzdem oder mal lässt sich vom schon bestehenden Komponenten inspirieren und übernimmt die Dinge, die man weiter brauchen möchte.

Jeder dieser Methoden bietet gewisse Vorteil aber auch Nachteile. Es kommt ganz auf die Umstände und den Verwendungszweck an, für was man sich entscheidet.

## Entscheiden

### Nutzwertanalyse

Bei einer Nutzwertanalyse geht es, darum mehrere Produkte gegenüberzustellen. Dies wird mithilfe einer Tabelle bewerkstelligt. In dieser Tabelle werden die einzelnen Bewertungskriterien aufgelistet welche jeweils eine Gewichtung erhalten. Durch diese Gewichtung wird die Analyse aussagekräftiger als eine Entscheidungsmatrix, denn nicht jedes Kriterium muss einen gleich hohen Stellenwert haben.

In der nachfolgenden Nutzwertanalyse wurde eine Gewichtung gesetzt von 1-5 und die Bewertungspunkte wurden mit 1-10 bewertet.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Möglichkeiten | | | |
|  |  | **Class** | | **Functional** | |
| Kriterium | Gewichtung | Beurt. | Wert | Beurt. | Wert |
| Syntax | 1 | 6 | 6 | 8 | 8 |
| State | 4 | 7 | 28 | 9 | 36 |
| Testing | 2 | 10 | 20 | 6 | 12 |
| Performance | 4 | 8 | 32 | 10 | 40 |
|  |  |  | 86 |  | 96 |

Tabelle : Nutzwertanalyse

### Entscheidungsmatrix

Während der Erstellung des Zeitplanes dachte ich, es wäre eine gute Idee, auch noch eine Entscheidungsmatrix zu erstellen. Jedoch dachte ich, nicht daran das eine Entscheidungsmatrix ziemlich genau dasselbe ist wie eine Nutzwertanalyse. Nur enthält die Nutzwertanalyse eine Gewichtung, wobei diese bei der Entscheidungsmatrix nicht vorkommt.

### Erklärung der gefällten Entscheidungen

Komponenten sollten nur das Nötigste enthalten und übersichtlich Aufgebaut sein. Daneben sollte auch der State möglichst einfach zu verwalten sein. Ein funktionaler Komponent bietet dies an, denn er besteht normaler weise aus weniger Zeilencode und beinhaltet weniger Methoden als ein klassenbasierter Komponent. Auch bei der Verwaltung des States bietet der funktionale Ansatz, durch die in React 16.8 hinzugefügten Hooks, Vorteile durch das aufteilen des States. Man wird nicht mehr gezwungen nur ein State-Objekt pro Komponent zu haben, sondern kann diese in einfacher verwaltbare Objekte unterteilen. Auch die Performance kann besser sein, da man den Komponeten gezielter steuern kann. Die einzigen zwei wirklichen Nachteil eines funktionalen Komponeten sind, dass dieser nicht so gut und ausführlich getestet werden kann und das man nicht auf alle Lifecycle-Hooks Zugriff hat.

Aufgrund von der Nutzwertanalyse und den oben genannten Dingen entscheide ich mich für die funktionale Implementierung. Dies wurde auch so in der Aufgabenstellung erwähnt.

## Realisieren

### Aufbau der Grundstruktur

Bevor ich beginne, will ich die Grundstruktur schon einmal erstellen, damit ich es später einfacher habe. Dies geschah in dem ich die Ordnerstruktur erstellte und einzelnen Dateien hinzufügte. Jeder Komponentenordner von normalen Komponenten enthält nun zwei Dateien, eine JavaScript-Datei sowie eine CSS-Datei.

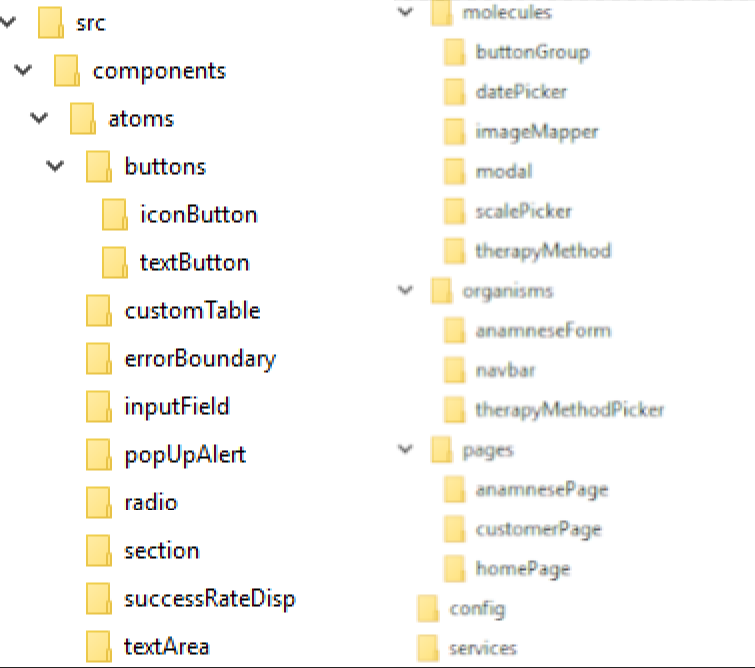
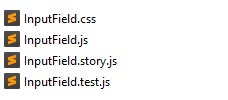
In den späteren Schritten werden noch weiter Dateien hinzukommen. Zum einen eine Datei für die Tests des Komponenten und zum anderen eines für das Storybook.

Abbildung :Dateien in einem Komponentenordner

Abbildung : Ordnerstruktur nach Atomic Design

### Einzelnen Komponenten

**Funktionaler Komponent**

Im Kapitel 2.4.3 habe ich erklärt wieso funktionale Komponenten die bessere Wahl für dieses Projekt ist. Jetzt in der Implementierungsphase werde ich aufzeigen wie ein möglicher funktionaler Komponent aussehen könnte.

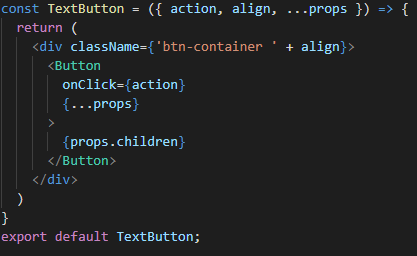


Abbildung : Funktionaler Komponent

Dieser hier abgebildete Komponent besteht aus einer einzelnen Funktion, wie der Name bereits sagt. In diesem Fall werden der Funktion mehrere Parameter mitgegeben, die die separat erwähnt werden dienen dazu, dass sie später in der Funktion einfacher aufgerufen werden können. Denn wird dies nicht gemacht muss man diese über das Props-Objekt aufrufen, dadurch stehen dann einige «props.paramName» im Code, was diesen nicht wirklich leserlicher macht.

**Klassenbasierter Komponent**

Für einige Komponenten musste ich jedoch auch Klassen zurückgreifen. Dies hat den Grund, dass Funktionale Komponeten keine Lifecycle-Hooks besitzen, ich diese jedoch benötigte für gewisse Funktionalitäten.

Abbildung : Klassenbasierter Komponent

Wie man in der Abbildung rechts sehen kann, handelt es sich bei diesem Komponenten um eine Klasse welche Component von React extended. Auch besitzt sie eine Render-Methode sowie einen Konstruktor.

Zwischen dem Konstruktor und der Render-Methode sieht man den Grund wieso ich für diesen Komponenten eine Klasse verwenden musste. Denn die verwendete Lifecycle-Hook «componentDidCatch()» existiert momentan nicht als Hook für funktionale Komponenten.

### Storybook Einträge

Das Storybook hat zwei Hauptaufgaben. Zum einen dient es zur Dokumentierung und Visualisierung der Komponenten für die anderen Entwickler die an der Applikation arbeiten. Zum anderen kann man so mit dem Kunden einzelne Aspekte der Seite isoliert besprechen.

In einem Storybook kann nun also eine Story erfasst werden die genau aufzeigt wie man diesen Button zu implementieren hat und auch welche Optionen man hat. Der eine Teil der Story entsteht in einem File welches z.B. «Button.story.js» heissen kann. Dort können verschiedene Varianten

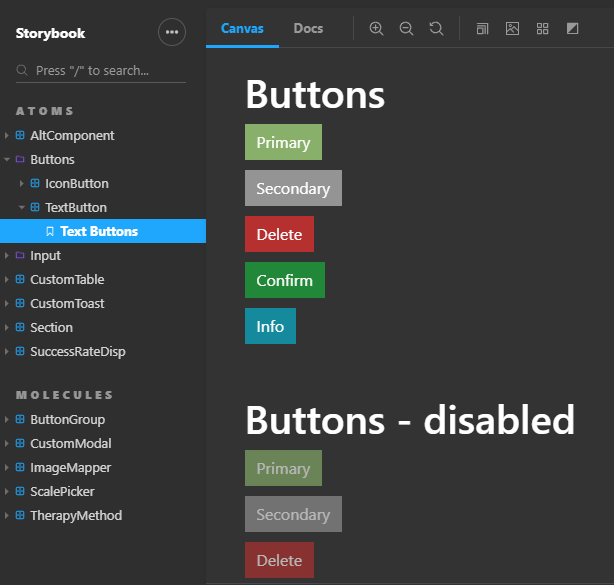
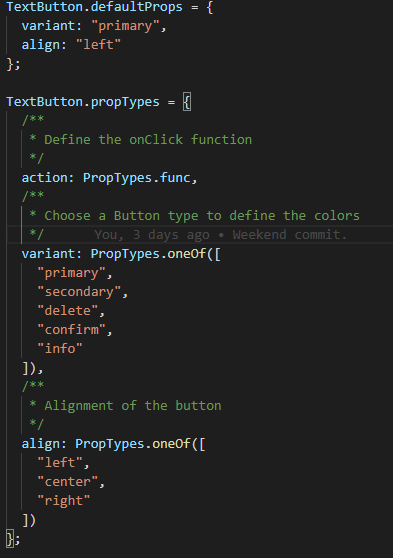
desselben Komponenten implementiert, sowie die Beschriftung im Storybook erstellt werden.

Abbildung : Storybook - Code für Canvasansicht

Abbildung : Storybook - Canvas ansicht

In der Canvasansicht sieht man wie der Komponent aussehen wird. Das Storybook beitet auch die Option den Hintergrund sowie die «Bildschirmgrösse» zu ändern. Diese Ansicht eignet sich dazu dem Kunden einen Einblick zu geben.

Es gibt noch eine weitere Ansichtsmöglichkeit welche sich mehr mit den Props beschäftigt die dem Komponenten mitgegeben werden können. Diese findet man unter «Docs». Die Daten werden direkt im Komponenten erfasst, ohne eine eigene Datei zu benötigen, zusätzlich enthalten sie auch eine kurze Beschreibung. Durch diese und die definierten PropTypes wird dann diese Seite generiert.

Zu jedem Prop kann man seinen Typ festlegen, genauso wie einen Kommentar um zu beschreiben für was dieser genau dient. Zu dem kann ein Default-Wert festgelegt werden.

Abbildung : Storybook - Code für die Doc-Ansicht

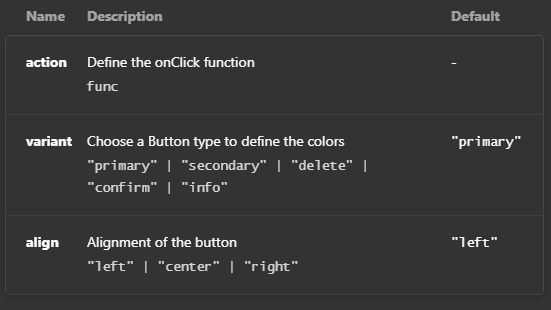


Abbildung : Storybook - Doc-Ansicht

### Exception Handling

### Input Validation

## Kontrollieren

### Erster Testdurchlauf

### Fehlerbehebung und zweiter Testdurchlauf

## Auswerten

### Persönliches Fazit

## Glossar

|  |  |
| --- | --- |
| Abkürzung | Beschreibung |
| SUD | **S**ystem **U**nder **D**esign |
| Prop |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle 61: Glossar

## Quellenverzeichnis

Draw.io für Diagramme, 28.02.2020 – 04.03.2020, https://draw.io

Buch UML 2.5, 04.03.2020, [Orell Füssli](https://www.orellfuessli.ch/shop/home/artikeldetails/ID87876335.html?ProvID=10917737&gclid=Cj0KCQiAwP3yBRCkARIsAABGiPp6ntSEMOy_LJKx8AsWjfb4vRSSbCqbZEd3McE2O5EFn84Sy4VqWAQaAvWREALw_wcB)

Atomic Design by Brad Frost, 05.03.2020, <https://bradfrost.com/blog/post/atomic-web-design/>

React-Bootstrap Dokumentation, 06.03.2020, <https://react-bootstrap.github.io/components/>

React Error Boundary, 09.03.2020, <https://reactjs.org/docs/error-boundaries.html>

Klassenbasierter Image-Mapper, 09.03.2020, <https://github.com/coldiary/react-image-mapper/blob/master/src/ImageMapper.js>

## Tabellenverzeichnis

[Tabelle 1: Protokoll - Erster Expertenbesuch 9](#_Toc34731961)

[Tabelle 2: Protokoll - Zweiter Expertenbesuch 9](#_Toc34731962)

[Tabelle 3: Arbeitsjournal - Tag 1 10](#_Toc34731963)

[Tabelle 4: Arbeitsjournal - Tag 2 11](#_Toc34731964)

[Tabelle 5: Arbeitsjournal - Tag 3 12](#_Toc34731965)

[Tabelle 6: Arbeitsjournal - Tag 4 13](#_Toc34731966)

[Tabelle 7: Arbeitsjournal - Tag 5 14](#_Toc34731967)

[Tabelle 8: Arbeitsjournal - Tag 6 15](#_Toc34731968)

[Tabelle 9: Arbeitspaket - Analyse und Verfeinerung der Arbeit 18](#_Toc34731969)

[Tabelle 10: Arbeitspaket - Klären von Unklarheiten 18](#_Toc34731970)

[Tabelle 11: Arbeitspaket - Projektmanagement / Definition von Arbeitspaketen erstellen 18](#_Toc34731971)

[Tabelle 12: Arbeitspaket - Zeitplan erstellen 18](#_Toc34731972)

[Tabelle 13: Arbeitspaket - Use Case Diagramm erstellen 19](#_Toc34731973)

[Tabelle 14: Arbeitspaket - Use Case Spezifikationen definieren 19](#_Toc34731974)

[Tabelle 15: Arbeitspaket - Mockup / Wireframe entwerfen 19](#_Toc34731975)

[Tabelle 16: Arbeitspaket - Sequenzdiagramm erstellen 19](#_Toc34731976)

[Tabelle 17: Arbeitspaket - Systemkontextdiagramm erstellen 19](#_Toc34731977)

[Tabelle 18: Arbeitspaket - Planung der benötigten Komponenten 19](#_Toc34731978)

[Tabelle 19: Arbeitspaket - Testkonzept erstellen 20](#_Toc34731979)

[Tabelle 20: Arbeitspaket - Design- & Architekturmöglichkeiten 20](#_Toc34731980)

[Tabelle 21: Arbeitspaket - Implementierungsvarianten gegenüberstellen 20](#_Toc34731981)

[Tabelle 22: Arbeitspaket - Nutzwertanalyse der Entscheidungsmöglichkeiten 20](#_Toc34731982)

[Tabelle 23: Arbeitspaket - Entscheidungsmatrix erstellen 20](#_Toc34731983)

[Tabelle 24: Arbeitspaket - Entscheidungen begründen 20](#_Toc34731984)

[Tabelle 25: Arbeitspaket - Implementierung der individuellen Komponenten 21](#_Toc34731985)

[Tabelle 26: Arbeitspaket - Storybook Einträge erstellen 21](#_Toc34731986)

[Tabelle 27: Arbeitspaket - Implementierung der Anamnese-Seite 21](#_Toc34731987)

[Tabelle 28: Arbeitspaket - implementierung der Anamneseliste 21](#_Toc34731988)

[Tabelle 29: Arbeitspaket - Implementierung der Schmerzzonenwahl 21](#_Toc34731989)

[Tabelle 30: Arbeitspaket - Implementierung des Exception Handlings 22](#_Toc34731990)

[Tabelle 31: Arbeitspaket - Vollständigkeit der Arbeitspakete & Abnahmekriterien überprüfen 22](#_Toc34731991)

[Tabelle 32: Arbeitspaket - Komponenten- & System-Tests ausarbeiten 22](#_Toc34731992)

[Tabelle 33: Arbeitspaket - Erster Testdurchlauf 22](#_Toc34731993)

[Tabelle 34: Arbeitspaket - Fehlerbehebung 22](#_Toc34731994)

[Tabelle 35: Arbeitspaket - Zweiter Testdurchlauf 23](#_Toc34731995)

[Tabelle 36: Arbeitspaket - Persönliches Fazit & Reflexion 23](#_Toc34731996)

[Tabelle 37: Arbeitspaket - Codebase anhängen & Abgabe 23](#_Toc34731997)

[Tabelle 38: Arbeitspaket - Weiterführen des IPA-Berichtes 23](#_Toc34731998)

[Tabelle 39: Arbeitspaket - Bearbeitung des Arbeitsjournals 23](#_Toc34731999)

[Tabelle 40: Arbeitspaket - Expertenbesuch 24](#_Toc34732000)

[Tabelle 41: Use Case - Klienten Anamneseliste anschauen 25](#_Toc34732001)

[Tabelle 42: Use Case - Neue Anamnese erfassen 25](#_Toc34732002)

[Tabelle 43: Use Case - Anamnese bearbeiten 25](#_Toc34732003)

[Tabelle 44: Use Case - Abschnitt bearbeiten 26](#_Toc34732004)

[Tabelle 45: Alternative Flow - Abschnittwechsel ohne zwischen speichern 26](#_Toc34732005)

[Tabelle 46: Alternative Flow - Seite verlassen ohne zu speichern 26](#_Toc34732006)

[Tabelle 47: Alternative Flow - Validation Inkorrekt 27](#_Toc34732007)

[Tabelle 48: Alternative Flow - Speichern fehlgeschlagen 27](#_Toc34732008)

[Tabelle 49: Use Case - Therapiemethode verwalten 27](#_Toc34732009)

[Tabelle 50: Use Case - Neue Therapiemethode erfassen 28](#_Toc34732010)

[Tabelle 51: Use Case - Therapiemethoden Vorschlag auswählen 28](#_Toc34732011)

[Tabelle 52: Benötigte Komponenten 36](#_Toc34732012)

[Tabelle 53: Test Case 1 - Klienten Anamneseliste anschauen 37](#_Toc34732013)

[Tabelle 54: Test Case 2 - Neue Anamnese erfassen 37](#_Toc34732014)

[Tabelle 55: Test Case 3 - Anamnese bearbeiten 37](#_Toc34732015)

[Tabelle 56: Test Case 4 - Validation inkorrekt 37](#_Toc34732016)

[Tabelle 57: Test Case 5 - Abschnittwechsel ohne zwischen speichern 38](#_Toc34732017)

[Tabelle 58: Test Case 6 - Seite verlassen ohne zu speichern 38](#_Toc34732018)

[Tabelle 59: Test Case 7 - Neue Therapiemethode erfassen 38](#_Toc34732019)

[Tabelle 60: Nutzwertanalyse 40](#_Toc34732020)

[Tabelle 61: Glossar 46](#_Toc34732021)

## Abbildungsverzeichnis

[Abbildung 1: Organigramm 5](#_Toc34732045)

[Abbildung 2: Arbeitsplatz 6](#_Toc34732046)

[Abbildung 3: Zeitplan "Soll" 7](#_Toc34732047)

[Abbildung 4: Use Case Diagramm 17](#_Toc34732048)

[Abbildung 5: Mockup - Liste der Anamnesen im Profil des Klienten (Desktop) 29](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732049)

[Abbildung 6: Mockup - Liste der Anamnesen im Profil des Klienten (Tablet) 29](#_Toc34732050)

[Abbildung 7: Mockup - Anamnese bearbeiten / erstellen Maske ohne Daten (Desktop) 30](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732051)

[Abbildung 8: Mockup - Anamnese bearbeiten / erstellen Maske ohne Daten (Tablet) 30](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732052)

[Abbildung 9: Mockup - Anamnese bearbeiten / erstellen Maske mit Daten (Tablet) 31](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732053)

[Abbildung 10: Mockup - Anamnese bearbeiten / erstellen Maske mit Daten (Desktop) 31](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732054)

[Abbildung 11: Mockup - Verlassen der Seite bei ungespeicherten Änderungen 32](#_Toc34732055)

[Abbildung 12: Mockup - Alternativer Komponent bei Render-Fehler 32](#_Toc34732056)

[Abbildung 13: Sequenzdiagramm - bearbeiten einer Anamnese 33](#_Toc34732057)

[Abbildung 14: Systemkontextdiagramm 34](#_Toc34732058)

[Abbildung 15: Atomic Design Pattern 39](#_Toc34732059)

[Abbildung 16:Dateien in einem Komponentenordner 41](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732060)

[Abbildung 17: Ordnerstruktur nach Atomic Design 41](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732061)

[Abbildung 18: Funktionaler Komponent 42](#_Toc34732062)

[Abbildung 19: Klassenbasierter Komponent 42](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732063)

[Abbildung 20: Storybook - Code für Canvasansicht 43](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732064)

[Abbildung 21: Storybook - Canvas ansicht 43](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732065)

[Abbildung 22: Storybook - Code für die Doc-Ansicht 44](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732066)

[Abbildung 23: Storybook - Doc-Ansicht 44](file:///D:\Desktop\docs\PA_Bericht_Steven_Ringger.docx#_Toc34732067)

# Anhänge