Personalplanung für den Helpdesk

Individuelle Praktische Arbeit 2019

Projektdaten

Kunde Basler Versicherungen Corporate IT (Helpdesk)

Auftraggeber Matthias Cullmann

Autor Elia Reutlinger

Ort Basel-Stadt

Zeitraum 24.04. – 10.05.2019

Version [**TODO**]

Status In Arbeit [**TODO**]

Beteiligte Personen

Anwender Corporate IT Helpdesk

Prüfung Matthias Cullmann *Verantwortliche Fachkraft*

Beat Sommer *Hauptexperte*

Felix Tobler *Nebenexperte*

Genehmigung Tschan Günter *Validierungsexperte*

Dokumentenmanagement

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Name | Beschreibung |
| 0.1 | 24.04.2019 | Elia Reutlinger | Dokument erstellt, Hauptkapitel notiert, Arbeitsjournal erstellt. |
| 0.1.1 | 25.04.2019 | Elia Reutlinger | Grundlegende Kapitel erweitert. |
| 0.2 | 02.05.2019 | Elia Reutlinger | Detaillierte Aufgabenstellung eingefügt und Analyse geschrieben. Kapitel Organisation und Firmenstandards begonnen. |
| 0.3 | 03.05.2019 | Elia Reutlinger | Kapitel Projektmanagement & Planung beendet, Auswertung der Aufgabenstellung beendet, Initialisierung begonnen und zahlreich verbessert (Details & Grammatik) |

Glossar

[**TODO** (ganz am Ende)]

Inhaltsverzeichnis

[Personalplanung für den Helpdesk 1](#_Toc7782980)

[1 Kurzfassung 5](#_Toc7782981)

[1.1 Ausgangssituation 5](#_Toc7782982)

[1.2 Ziel 5](#_Toc7782983)

[1.3 Umsetzung 5](#_Toc7782984)

[1.4 Ergebnis 5](#_Toc7782985)

[Teil 1: Ablauf und Umfeld 6](#_Toc7782986)

[2 Detaillierte Aufgabenstellung 6](#_Toc7782987)

[2.1 Ausgangslage 6](#_Toc7782988)

[2.2 Detaillierte Aufgabenstellung 6](#_Toc7782989)

[2.3 Tests 8](#_Toc7782990)

[2.4 Mittel und Methoden 9](#_Toc7782991)

[2.5 Vorkenntnisse 9](#_Toc7782992)

[2.6 Vorarbeiten 9](#_Toc7782993)

[2.7 Neue Lerninhalte 10](#_Toc7782994)

[2.8 Arbeiten in den letzten 6 Monaten 10](#_Toc7782995)

[3 Analyse der Aufgabenstellung 11](#_Toc7782996)

[3.1 Kurzfassung Ist/Soll 11](#_Toc7782997)

[3.2 Messbare Ziele 11](#_Toc7782998)

[3.3 Dokumentation 11](#_Toc7782999)

[4 Projektmanagement und Planung 12](#_Toc7783000)

[4.1 Organisation 12](#_Toc7783001)

[4.1.1 Zeitplan & Termine 12](#_Toc7783002)

[4.1.2 Projektaufbauorganisation 13](#_Toc7783003)

[4.1.3 Arbeitsplatz 14](#_Toc7783004)

[4.1.4 Datensicherung & Versionierung 14](#_Toc7783005)

[4.2 Auswertung der Aufgabenstellung 15](#_Toc7783006)

[4.2.1 Wandlung der vorgegebenen Tests 15](#_Toc7783007)

[4.2.2 Mittel und Methoden 15](#_Toc7783008)

[4.2.3 Abhängigkeiten zu Vorarbeiten 16](#_Toc7783009)

[5 Firmenstandards 17](#_Toc7783010)

[5.1 Schriftarten 17](#_Toc7783011)

[5.2 Farben 17](#_Toc7783012)

[5.3 Logos 18](#_Toc7783013)

[5.4 Formularelemente 18](#_Toc7783014)

[6 Arbeitsjournal 19](#_Toc7783015)

[6.1 Tag 1 – 24.04.2019 19](#_Toc7783016)

[6.2 Tag 2 – 25.04.2019 20](#_Toc7783017)

[6.3 Tag 3 – 26.04.2019 21](#_Toc7783018)

[6.4 Tag 4 – 30.04.2019 22](#_Toc7783019)

[6.5 Tag 5 – 02.05.2019 23](#_Toc7783020)

[6.6 Tag 6 – 03.05.2019 24](#_Toc7783021)

[6.7 Tag 7 – 07.05.2019 25](#_Toc7783022)

[6.8 Tag 8 – 08.05.2019 26](#_Toc7783023)

[6.9 Tag 9 – 09.05.2019 27](#_Toc7783024)

[6.10 Tag 10 – 10.05.2019 28](#_Toc7783025)

[7 Abschlussbericht 29](#_Toc7783026)

[7.1 Ist-Soll 29](#_Toc7783027)

[7.2 Schwierigkeiten 29](#_Toc7783028)

[7.3 Fazit zum Projekt 29](#_Toc7783029)

[7.3.1 Entstandene Abweichungen zur Aufgabenstellung 29](#_Toc7783030)

[7.4 Persönliches Fazit 29](#_Toc7783031)

[7.5 Schlussreflexion 29](#_Toc7783032)

[7.5.1 Arbeitsmethodik 29](#_Toc7783033)

[7.5.2 Werkzeuge 29](#_Toc7783034)

[Teil 2: Dokumentation und Umsetzung 30](#_Toc7783035)

[8 Initialisierung 30](#_Toc7783036)

[8.1 Analyse Vue.js 30](#_Toc7783037)

[8.1.1 Konzept 31](#_Toc7783038)

[8.1.2 Vergleich 31](#_Toc7783039)

[8.2 Vertiefte Studie Ist-Zustand & Umgebung 31](#_Toc7783040)

[8.3 Umsetzung Projektmethode & Vorgehensmodell 32](#_Toc7783041)

[8.4 Ziele 33](#_Toc7783042)

[8.4.1 Projekt 33](#_Toc7783043)

[8.4.2 Persönlich 33](#_Toc7783044)

[8.5 Anforderungen 33](#_Toc7783045)

[8.5.1 Funktional 33](#_Toc7783046)

[8.5.2 Nicht-Funktional 34](#_Toc7783047)

[8.6 Informationssicherheit & Datenschutz 34](#_Toc7783048)

[8.7 Konzepte 35](#_Toc7783049)

[8.7.1 Testkonzept 35](#_Toc7783050)

[8.7.2 Sicherheitskonzept 35](#_Toc7783051)

[9 Realisierung 35](#_Toc7783052)

[9.1 Grundlegende Konfiguration 35](#_Toc7783053)

[9.1.1 Mehrsprachigkeit 35](#_Toc7783054)

[9.1.2 Authentifizierung 35](#_Toc7783055)

[9.1.3 Ansichten / Routes 35](#_Toc7783056)

[9.2 Umsetzung Design 35](#_Toc7783057)

[9.3 Umsetzung der Ansichten 35](#_Toc7783058)

[9.3.1 Support 35](#_Toc7783059)

[9.3.2 Dashboard 35](#_Toc7783060)

[9.3.3 Einstellungen 35](#_Toc7783061)

[9.3.4 Plan-Einstellungen 35](#_Toc7783062)

[9.3.5 Dienstplan 35](#_Toc7783063)

[9.4 Testprotokoll 35](#_Toc7783064)

[9.5 Sicherheitsprüfung 36](#_Toc7783065)

[10 Abschluss 36](#_Toc7783066)

[10.1 Auswertung des Kanban-Boards 36](#_Toc7783067)

[10.2 Funktionen & Verbesserungen für die Zukunft 36](#_Toc7783068)

[10.3 Danksagung 36](#_Toc7783069)

[11 Anhang 36](#_Toc7783070)

[11.1 Quellenverzeichnis 36](#_Toc7783071)

[11.2 Abbildungsverzeichnis 36](#_Toc7783072)

# Kurzfassung

## Ausgangssituation

## Ziel

## Umsetzung

## Ergebnis

Teil 1: Ablauf und Umfeld

# Detaillierte Aufgabenstellung

Folgend die Definition der Aufgabenstellung, welche als «detaillierte Aufgabenstellung» auf pkorg.ch erfasst wurde.

## Ausgangslage

Der IT-Helpdesk der Baloise Group besteht aus einem Team von ca. 30 Mitarbeiter und erbringt Dienstleistungen in 5 Sprachen für die Ländergesellschaften der Baloise Group in Belgien, Luxemburg, Deutschland und der Schweiz.

Aktuell werden die Einsätze der Mitarbeiter in einer Excel-Tabelle geplant. Diese Tabelle soll in eine Webanwendung umgewandelt werden, damit die Inhalte mit wenig Aufwand anpassbar sowie jederzeit von überall abrufbar sind. Ebenso sollen Bedienungsfehler und Datenverluste vermieden werden.

Der Helpdesk der Baloise befindet sich, wie auch die Softwareentwicklung Schweiz, in deren Rahmen die IPA durchgeführt wird, am Hauptsitz der Baloise, Aeschengraben 21, 4051 Basel.

## Detaillierte Aufgabenstellung

Grundlegende Informationen

Resultat der Aufgabe soll eine Webanwendung zur Einsatzplanung am Helpdesk sein. Damit ist die Entwicklung eines Front-Ends zu einem bereits bestehenden Back-End (einer Rest-API) gemeint. Dieses Back-End sowie die dazugehörige Datenbank werden bereits vor der IPA geplant und umgesetzt. Die Anwendung soll von mehreren Benutzern genutzt werden können, wobei sich diese in Teams unterteilen und durch Rollen unterschiedliche Berechtigungen haben.

Struktur der Benutzerverwaltung

Im Anschluss an die IPA soll in der Anwendung die Verwaltung von Teams implementiert werden. Für die IPA und zu Demo-Zwecken soll jedoch nur ein vordefiniertes Team «Helpdesk» bestehen. Die Teamverwaltung ist also während der IPA nicht umzusetzen.

Ein Team beinhaltet mindestens einen Teamleiter und ein Mitglied. Teamleiter können weitere Mitglieder hinzufügen, deren Rolle (Teamleiter oder Mitglied) bearbeiten, und Mitglieder entfernen. Teamleiter und Mitglieder haben jeweils nur Zugriff auf Daten, welche das eigene Team betreffen.

Das hinzufügen neuer Mitglieder soll anhand deren interner Benutzerkennung geschehen, worauf während der IPA aber noch nicht zugegriffen werden kann. Deshalb sollen für die IPA und zu Demo-Zwecken einige vordefinierte Mitglieder definiert werden, wobei man diese noch nicht durch das Front-End erweitern kann. Die Funktion für das Hinzufügen neuer Mitglieder ist also während der IPA nicht umzusetzen.

Struktur der Anwendung

In der Anwendung können die Einsätze von Mitgliedern eines Teams von deren Teamleitern geplant werden. Dazu gehören Schichten (Was zu tun ist, z.B. Vor-Ort Support, Telefon-Support, ...) und Tageszeiten (Wann es zu tun ist, z.B. Morgens, Mittags, Abends, …). Ein Einsatz ist die Zuordnung eines Mitglieds zu einer Schicht und einer Tageszeit an einem bestimmten Datum (z.B. Herr Maier, Telefon-Support, Mittags, 12.12.2019).

Diese Anforderungen sowie die der Benutzerverwaltung fordern also mindestens folgende Ansichten (mit geschätzter Implementierungsdauer):

* Eine Tageszeiten-Ansicht, in welcher Teamleiter die verfügbaren Tageszeiten bearbeiten können. (4h)
* Eine Schichten-Ansicht, durch welche Teamleiter die verfügbaren Schichten bearbeiten können (Möglicherweise zusammen mit der Tageszeiten-Ansicht zu implementieren). (4h)
* Eine Einsatz-Ansicht, auf welcher alle Einsätze von Mitgliedern eingesehen und von Teamleitern bearbeitet werden können. (8h)
* Eine Einstellungen-Ansicht, in welcher ein Teamleiter die Rollen der Mitglieder bearbeiten kann und jeder Benutzer die Sprache seiner Oberfläche anpassen kann. (4h)
* Eine einfache Login-Ansicht (Details folgen) für Demo-Zwecke. (2h)

Summe der Aufwände ist 22H, mit Projektkonfiguration und Recherchen ca. 26H.

Authentifizierung

Im Anschluss an die IPA wird die Anwendung in der Baloise intern aufgesetzt. Dabei kommt für die Authentifizierung eine bereits vorhandene (Single Sign-On) Infrastruktur zum Einsatz, weshalb im Front-End bzw. während der IPA keine Login- und Registrierungs-Maske implementiert werden soll. Wie bereits beschrieben soll es für Demo-Zwecke bereits vordefinierte Teams und Benutzer geben, wobei eine möglichst einfache Ansicht für den Login erstellt werden soll. Damit soll man sich durch auswählen eines vordefinierten Benutzers mit dessem Profil anmelden können.

Die im Voraus implementierte Rest-API arbeitet mit JWT (Json Web Token), welche vom Front-End nur noch verwaltet werden müssen.

Weitere Kriterien

Die Anwendung soll Mobile-Fähig sein. Das bedeutet, dass sie auch im Browser eines Smartphones problemlos dargestellt werden kann, und alle Funktionen hier gleich funktionieren. Die Gestaltung der Oberfläche kann abhängig vom Inhalt beliebig sein, wobei die Farben und Logos von den Corporate Design Richtlinien der Baloise (https://webstyleguide.baloise.com/) abgeleitet werden sollen. Da die Mitarbeiter des Helpdesks unterschiedliche Sprachen sprechen soll die Anwendung in 2 Sprachen zur Verfügung stehen: Englisch und Deutsch. Die Sprache soll vom Benutzer selbst angepasst werden können.

Abgrenzungen

Es ist zu beachten, dass für die IPA nur das Front-End umzusetzen ist. Dies bedeutet, dass jegliche Arbeiten bezüglich Datenbank und Back-End bereits im Voraus erledigt sind und keinen fachlichen Bezug zur IPA haben. Dies betrifft also Details zu Attributen in der Datenbank und deren Struktur, sowie weiterer Parameter. Ebenso gehören dazu die Berücksichtigung von Sicherheit- und möglichen Programmier-Standards im Back-End.

Projektmethode

Da die Implementierung als Teil der IPA zeitlich begrenzt ist und nicht viel Raum für Besprechungen und Abklärungen bleibt, entschieden wir uns für die Projektmethode «Kanban». Die Umsetzung dieser wird in Verbindung mit GitHub Issues online abgehalten. Dadurch kann der Projektfortschritt und der anstehende Arbeitsaufwand jederzeit abgerufen werden.

## Tests

Zur Prüfung der korrekten Funktion der Anwendung sollen manuelle Tests definiert und mit einem Protokoll ausgeführt und festgehalten werden. Notwendige Test, welche die vorausgesetzten Funktionen der Anwendung bestätigen, sind folgende:

Einsatz planen

Es ist einem Teamleiter möglich, den Einsatz eines Mitglieds zu planen. Dazu gehört die Auswahl einer Schicht in einer Tageszeit. Die Daten werden anschliessend korrekt verarbeitet und im Back-End gespeichert.

Einsatz bearbeiten

Es ist einem Teamleiter möglich, vorhandene Einsätze zu verändern (die Schicht anzupassen) und zu löschen. Die Daten werden anschliessend korrekt verarbeitet und im Back-End gespeichert.

Einsätze darstellen

Es soll allen Mitgliedern eines Teams möglich sein, die Einsätze von sich selbst und der anderen einzusehen. Dies in einer dafür sinnvoll formatierten Tabelle.

Schichten & Tageszeiten bearbeiten

Es soll einem Teamleiter möglich sein, die Schichten und Tageszeiten zu erweitern, bearbeiten und löschen. Falls es Abhängigkeiten zu geplanten Einsätzen gibt, sollen die betroffenen Einsätze gelöscht werden. Die Daten werden anschliessend korrekt verarbeitet und im Back-End gespeichert.

Rollen der Mitglieder bearbeiten

Es soll einem Teamleiter möglich sein, die Rollen (Teamleiter oder Mitglied) der anderen Mitglieder anzupassen. Die Daten werden anschliessend korrekt verarbeitet und im Back-End gespeichert.

Sprache anpassen

Alle Nutzer der Anwendung sollen die dargestellte Sprache anpassen können. Die Auswahl soll dabei aus Deutsch und Englisch bestehen. Die Daten werden anschliessend korrekt verarbeitet und im Back-End gespeichert.

## Mittel und Methoden

Hardware

* Notebook

Software

* Atom (Code-Editor)
* Git
* GitHub Desktop
* Vue CLI
* XAMPP (Lokale Back-End Umgebung)
* Google Chrome
* Word, Excel, Paint.Net (Grafiken)

Dienste

* Github, Waffle.io (Kanban)
* Netlify (für Demo-Zwecke)

Sprachen & Abhängigkeiten

* HTML, CSS, JavaScript, XML, JSON
* VueJS, Vuetify (Design), i18n (Übersetzungen), Axios (XMLHttpRequests),…

Firmenvorgaben

* Farbpalette der Corporate-Design Richtlinien
* Firmenlogos

Methoden

* Kanban (Waffle.io)

## Vorkenntnisse

Der Lernende hat bereits alle erwähnten Mittel und Methoden mindestens 1 Mal angewendet. VueJS ist jedoch bisher am wenigsten vertieft worden, während die Kenntnisse in allen anderen Punkten sehr gefestigt sind. Der Lernende arbeitet bereits seit mehreren Jahren mit den meisten Mitteln und mit VueJS seit einigen Monaten.

## Vorarbeiten

Rest-API

Während der IPA wird eine im Voraus umgesetzte Rest-API (Representational State Transfer Application Programming Interface) zur Verfügung stehen, welche hauptsächlich vom Lernenden mit PHP und MySQL implementiert wurde. Sessions sind somit Stateless wobei für die Validierung JWT (JSON Web Token) verwendet wird. Damit steht bereits eine Struktur der Daten und möglichen Abfragen fest, an welcher sich die Umsetzung der IPA orientieren muss.

Git-Repository

Um in der Durchführungs-Phase der IPA direkt mit der Entwicklung starten zu können, wird bereits im Voraus ein Git-Repository auf GitHub eingerichtet. Das Repository wird unter der firmeneigenen Organisation «Baloise» erstellt und «pelan-ui» genannt. Das Urheberecht gehört somit der Baloise unter der Verwendung einer MIT-Lizenz.

Kanban-Board

Mit dem Git-Repository wird auch ein Kaban-Board eingerichtet, welches voraussichtlich mit der Integration von Waffle.io zustande kommt. Es werden auch bereits die Anforderungen in Form von GitHub-Issues eingetragen.

## Neue Lerninhalte

Möglicherweise werden während der Entwicklung weitere Abhängigkeiten benötigt, zu welchen sich der Lernende in den entsprechenden Dokumentationen informieren muss. Zu den bekannten Mitteln kann sich der Lernende im Internet auf den zugehörigen Webseiten oder in Forums informieren. Die Umsetzung der Arbeit sollte jedoch nicht an einer unerwartet grossen, neuen Kompetenz scheitern, da es sich hierbei nur um kleinere Abhängigkeiten handeln wird.

## Arbeiten in den letzten 6 Monaten

In den letzten Monaten bestanden die Arbeiten des Lernenden aus hauptsächlich Front-End lastigen Arbeiten im Bereich Web-Development. Dabei wurden folgende Techniken genutzt: HTML, CSS, JavaScript, PHP, SQL, REST, Java, JavaScript-Libraries (JQuery, VueJS, Bootstrap,…).

Ein grosses Projekt war dabei der «Cash-Calculator», welcher den Leistungslohn aller Lernenden der Baloise berechnet. Dies geschieht anhand unterschiedlicher Parameter wie Noten und Verhalten.

Der «Cash-Calculator» steht allen Lernenden und der Personalabteilung zur Verfügung, und wird regelmässig genutzt. Für die Umsetzung wurden hier PHP und JavaScript (JQuery) genutzt, wobei der lernende auch in Kontakt mit der internen Authentifizierungs-Infrastruktur «Medusa» kam.

Darüber hinaus hat der Lernende eine Landing-Page für Start-Ups auf der Baloise.ch Webseite implementiert, über welche interessierte Start-Ups mit Mitarbeitern der Baloise in Kontakt treten können. Diese Aufgabe wurde mit Java im Back-End und JavaScript im Front-End realisiert.

# Analyse der Aufgabenstellung

Folgende Kapitel sollen die Aufgabenstellung aus pkorg.ch analysieren und wichtige Details für die Durchführungsphase erläutern.

## Kurzfassung Ist/Soll

Ist-Zustand

Die Einsätze der Mitarbeiter werden in einer Excel-Tabelle geplant, welche sich auf einem Netzlaufwerk befindet. Die Tabelle bildet die Einsätze aller Mitarbeiter für ein Semester ab, und kann nur mit einem Passwort bearbeitet werden.

Soll-Zustand

Die Einsatzplanung findet in einer Webanwendung statt, welche von allen Mitarbeitern genutzt werden kann. Die Berechtigung zur Bearbeitung der Einsätze ist abhängig von der Benutzergruppe des angemeldeten Benutzers und Tageszeiten sowie Schichten können von Teamleitern mit wenig

## Messbare Ziele

* Die Anwendung kann ohne weitere Anpassungen produktiv eingesetzt werden.
* Der Arbeitsfortschritt wird mit der Projektmethode «Kanban» dokumentiert.
* Änderungen in der Planung, die während der Realisierung auftreten, sind dokumentiert.
* Die Anwendung bietet die Auswahl zwischen 2 Sprachen an.
* Tageszeiten und Schichten lassen sich dynamisch anpassen.
* Teamleiter können die Gruppe eines Benutzers anpassen.
* Teamleiter können Einsätze planen, welche von allen Teammitgliedern eingesehen werden können.

## Dokumentation

Die Dokumentation der Entwicklung findet auf GitHub statt. Die anfallenden Arbeitsschritte werden als Issues notiert und mit Labels geordnet. Dadurch können sowohl Änderungen der Arbeit als auch Fortschritte festgehalten werden, wobei Kanban genutzt wird. Änderungen in der Planung und Probleme werden ebenfalls mit einem bestimmten Label notiert.

# Projektmanagement und Planung

## Organisation

Die Durchführung einer IPA fordert eine gute Planung und Organisation. Dazu gehört neben einem Zeitplan auch eine klare Aufteilung der Rollen, damit bei möglichen Rückfragen die richtige Ansprechperson kontaktiert werden kann.

### Zeitplan & Termine

Der Zeitplan soll in der Planung eine Einschätzung des anfallenden Arbeitsaufwandes abbilden. Dazu wird die Arbeit in die 2 Abschnitte «Entwicklung» und «Dokumentation» aufgeteilt, wobei das tägliche Ergänzen des Arbeitsjournals berücksichtigt werden muss. Da dies nur zwischen 10-20 Minuten dauern soll, wurde es im Zeitplan nicht explizit erwähnt. Nach Abschluss des Projekts soll ein Ist-Soll-Vergleich die Einhaltung des Zeitplans erkennbar machen.

Durchführungsblock

Startblock 3: 24.04.2019 – 10.05.2019

Durchführungs-Ort: Basler Versicherungen AG, Aeschengraben 21, 4002 Basel

Abgabe: 10.05.2019 um 18:00 Uhr

Termine

Vorbesprechung 05.04.2019 um 16:00 Uhr, @ Basler Versicherungen AG

Erster Besuch 25.04.2019 um 16:30 Uhr, @ Basler Versicherungen AG

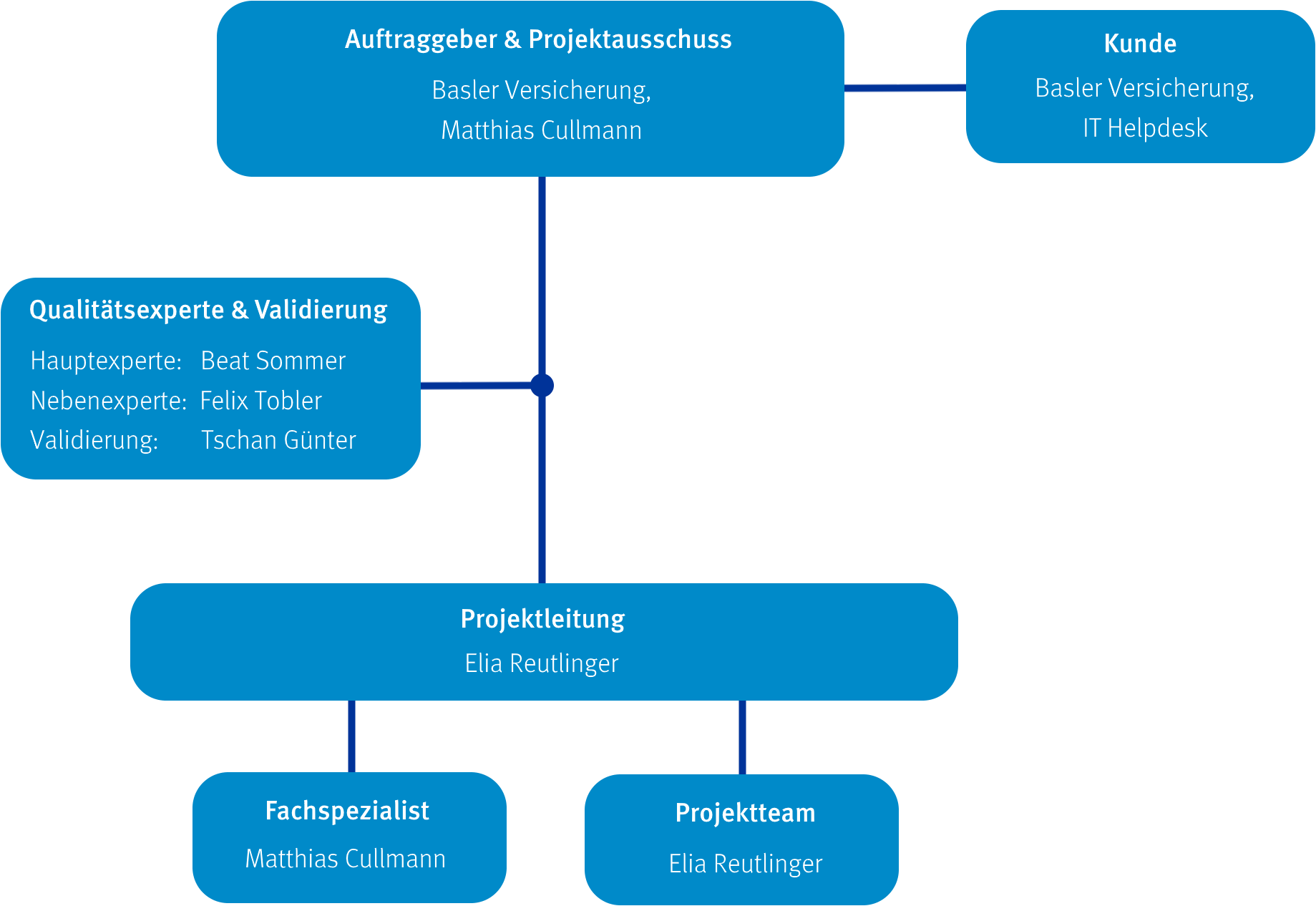
Zweiter Besuch 03.05.2019 um 11:00 Uhr, @ Basler Versicherungen AG

Präsentation & Fachgespräch 21.05.2019 um 15:00 Uhr, @ Basler Versicherungen AG

[**TODO** weitere Termine ergänzen, wenn nötig]

[**TODO**: Zeitplan einfügen (ganz am Ende)]

### Projektaufbauorganisation



Rollen

Auftraggeber Matthias Cullmann

Projektausschuss Matthias Cullmann

Kunde Corporate IT Helpdesk

Qualitätsexperten Beat Sommer, Felix Tobler

Validierungsexperte Tschan Günter

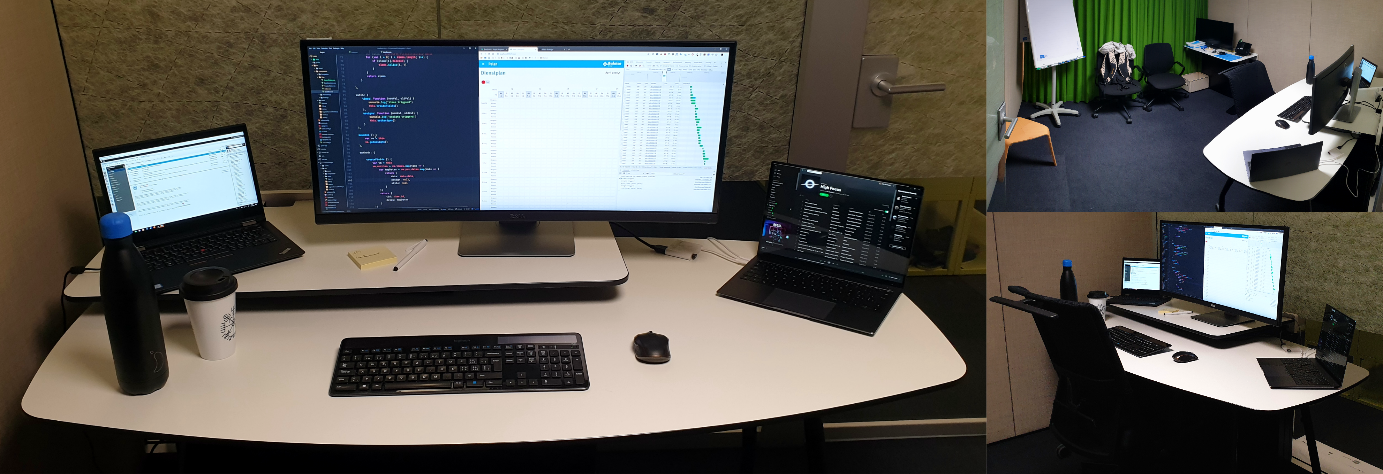
Projektleitung Elia Reutlinger

Fachspezialist Matthias Cullmann

Projektteam Elia Reutlinger

### Arbeitsplatz

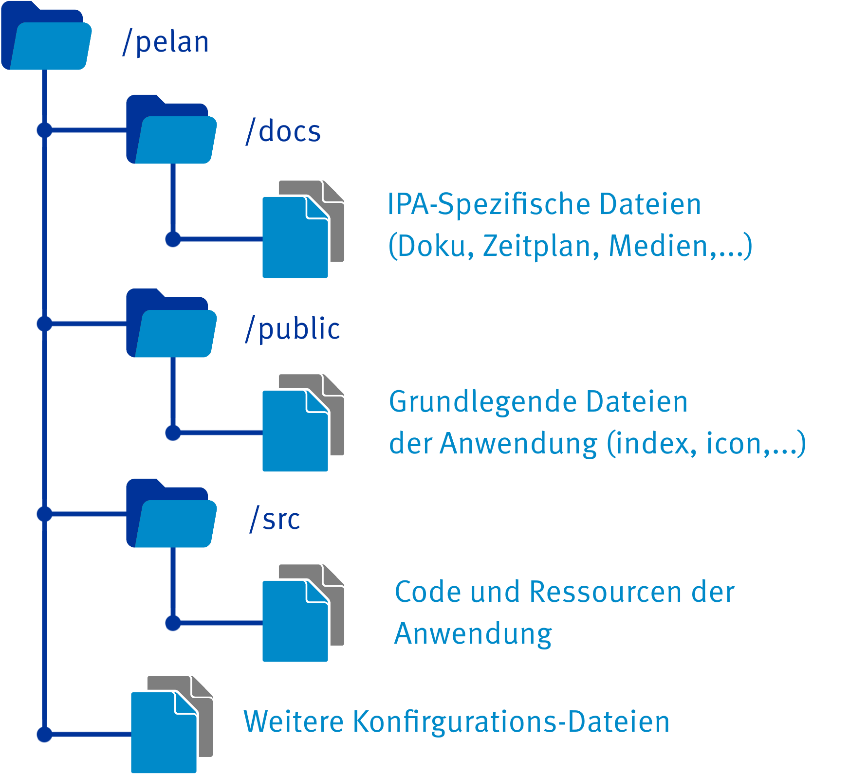
Die Ausführung der IPA findet im Hauptsitz der Basler Versicherungen statt. Im 4. Stock befinden sich neben dem Flex-Office einige Rückzugsräume, wobei einer von diesen für die IPA reserviert ist.



### Datensicherung & Versionierung

Die Datensicherung während der IPA findet vorrangig auf GitHub statt. Dabei wird die Anwendung mit dem IPA-Bericht in einem separaten Verzeichnis hochgeladen, was nach jeder grösseren Änderung mindestens 1 Mal pro Tag geschehen sollte. Mit jedem Hochladen können ein Titel sowie eine Beschreibung des Fortschritts notiert werden, wodurch erkennbar gemacht werden soll, was genau in der Version geändert wurde. Gleichzeitig kann man auf GitHub-Issues verweisen, wodurch im Kanban ersichtlich wird, in welcher Version welcher Arbeitsschritt erledigt wurde.

Als zusätzliche Absicherung bei Ausfällen erstelle ich auf einer externen Festplatte ein tägliches Backup der gesamten Partition meines Notebooks. Dies ist zwar laut Erkenntnissen aus der IPA-Vorbesprechung keine direkte Anforderung der IPA, ermöglicht es mir aber, im Notfall schneller zu meiner Arbeit zurückzukehren.



## Auswertung der Aufgabenstellung

In diesem Kapitel wird die Nutzung der detaillierten Aufgabenstellung als Quelle erläutert. Die Reflektion und Umwandlung der dort bestimmten Kriterien sollen hier erkennbar gemacht werden, da sich die Implementierung der Anwendung ausschliesslich an diesen orientierte.

### Wandlung der vorgegebenen Tests

Zur Prüfung der korrekten Funktion der Anwendung werden manuelle Tests durch ein Konzept definiert und mit einem Protokoll ausgeführt und festgehalten. Notwendige Test, welche die vorausgesetzten Funktionen der Anwendung bestätigen, wurden in der Aufgabenstellung auf pkorg.ch bereits grob beschrieben und müssen im Testkonzept enthalten sein. Dabei werden diese mit einzelnen Testschritten, erwarteten Resultaten und wenn nötig weiteren Details ergänzt. Die vordefinierten Tests bilden gleichzeitig die Use-Cases der Anwendung, nach welchen sich die Implementierung richtete.

### Mittel und Methoden

Zur Entwicklung der Anwendung und Erfassung des IPA-Berichts werden einige Tools benötigt, mit denen diese Aufwände optimal erledigt werden können. Da die Entwicklung losgelöst von internen Abhängigkeiten sein soll wird ein privates Notebook verwendet. Dies führt auch zu einer Absicherung vor möglichen Komplikationen mit der Infrastruktur der Baloise (beispielsweise Proxys, Firewalls und fehlende Berechtigungen).

Der Schwerpunkt während der Entwicklung lag bei unterschiedlichen Programmiersprachen und Frameworks, die ich in der Vergangenheit schon genutzt habe und dabei bereits einige Kenntnisse sammeln konnte.

* HTML & CSS

Zur Strukturierung und Gestaltung der Anwendung. Seit ungefähr 8 Jahren bei den meisten meiner Projekte in Verwendung.

* JavaScript

Für Funktionalitäten und die grundlegende Entwicklung der Anwendung. Bekannt seit Beginn meiner Ausbildung und bereits in zahlreichen Projekten verwendet.

* Vue.js

Webframework (JavaScript) zur Entwicklung clientseitiger Anwendungen. Erstmals im Dezember 2018 angewendet und bis jetzt 2 Projekte damit realisiert.

Bei der Verwendung von Vue.js entstehen in der Regel weitere Abhängigkeiten. Beispielsweise wird für das Design die Komponenten-Bibliothek Vuetify genutzt, durch die ein gängiges «Material Design» zeitnah realisiert werden kann.

Software & Dienste

Damit die Entwicklung lokal und mit Testdaten stattfinden kann, wird eine Instanz der Pelan-API benötigt. Da diese mit PHP und einer Datenbank arbeitet kann sie durch XAMPP betrieben werden. XAMPP ermöglicht unter anderem die Nutzung eines Apache Webservers auf einem lokalen Gerät, wobei zahlreiche Optionen zur Instanziierung einer API mitgeliefert werden.[[1]](#footnote-1)

Atom ist ein von GitHub entwickelter Texteditor, mit dem der Code von Pelan geschrieben wurde. Er bietet die Installation von zahlreichen Plugins an, durch die das Entwickeln vereinfacht werden soll. Weil die Anwendung auf GitHub dokumentiert und publiziert werden soll, wird GitHub Desktop verwendet. GitHub Desktop ist eine grafische Oberfläche zur Nutzung von Git, wodurch Änderungen im Code einfacher ersichtlich sind und schnell hochgeladen werden können. [[2]](#footnote-2) Durch die Integration eines Project-Boards auf GitHub werden die Issues des Pelan-Repositorys mit einem Kanban-Board visualisiert.

Zur Demonstration wird die Anwendung mit dem Online-Dienst Netlify unter einer Subdomain publiziert. Netlify bietet kostenloses Hosting von statischen Webseiten an, wobei ein Deployment-Script genutzt werden kann. Dadurch ist es möglich, einen Produktions-Build der Anwendung zu erstellen, was die endgültige Funktion garantieren soll. Alternativ könnte die Anwendung auch lokal im produktiven Modus demonstriert werden. [[3]](#footnote-3)

### Abhängigkeiten zu Vorarbeiten

Das Back-End zur Anwendung wurde bereits vor Beginn der IPA implementiert. Es besteht aus einer Rest-API, welche unter der Verwendung von JSON-Web-Tokens (JWT[[4]](#footnote-4)) Anfragen verarbeiten kann. Die Authentifizierung findet ausschliesslich im Back-End statt, da dies dem produktiven Einsatz innerhalb der Baloise entspricht. Die Implementierung des Front-Ends muss sich nach den bestehenden Endpunkten der API richten. Während der IPA dürfen keine gravierenden Anpassungen an der API vorgenommen werden, da dies nicht im Rahmen der Aufgabenstellung liegt.

Es wurde vorweg ein Repository unter der Organisation «Baloise» auf GitHub erstellt und mit grundlegenden Anforderungen/Aufgaben (Issues) in Berücksichtigung der detaillierten Aufgabenstellung ergänzt. Ebenfalls in der Aufgabenstellung festgelegt ist die Verwendung von Kanban als Projektmethode. Die Umsetzung findet in Form eines «Project-Boards» auf GitHub statt, wobei die Anforderungen als Issues visualisiert werden. Die Durchführung der IPA und Dokumentierung des Codes orientieren sich an diesem Kanban-Board. Des Weiteren muss mindestens ein tägliches Backup des Codes in dieses Repository geladen werden.

# Firmenstandards

Für den Wiedererkennungswert der Baloise in der Anwendung sollen möglichst viele Firmenstandards übernommen werden. Da die Funktion der Anwendung jedoch im Vordergrund stehen sollte, wird keine vollständige Umsetzung erwartet, was auch den Rahmen einer IPA sprengen würde. Da die Komponenten-Bibliothek «Vuetify» zur Umsetzung verwendet werden soll, sind hier bereits einige Abweichungen von den Firmenstandards eingeplant. Grundlegende Standards, die verwendet werden müssen, sind anschliessend in diesem Kapitel genauer dokumentiert. Sie stammen aus dem Webstyleguide[[5]](#footnote-5) und den Corporate Design Richtlinien[[6]](#footnote-6) der Baloise. Bei Unklarheiten zu Einzelfällen soll das weitere Vorgehen mit dem Fachvorgesetzten besprochen und entschieden werden.

## Schriftarten

Die Baloise nutzt grundsätzlich die Schriftarten der MetaPro-Reihe. Die textuellen Inhalte der Anwendung sollen deshalb ausschliesslich mit diesen angezeigt werden. Dabei stehen 2 Versionen zur Verfügung, welche je nach Zweck (Überschrift, Titel, Text, …) eingesetzt werden können.

## Farben

Der Webstyleguide der Baloise unterscheidet 3 Farbsorten:

Primärfarben

Diese bilden die beiden Blautöne der Baloise, #003399 und #008AC9, sowie ein dunkler Grauton #444444 als Textfarbe. Dieser hat jedoch lediglich den Zweck eine angenehme Lesbarkeit an Bildschirmen zu gewährleisten und könnte entsprechend mit einem gleichwertigen Farbton ersetzt/kombiniert werden.

Sekundärfarben

Für Benachrichtigungen sowie Erfolgs- und Fehlermeldungen wurden Sekundärfarben definiert. Sie bestehen aus einem Grün #2DB200, Rot #FF3366 und Orange #FF9900 (oder #FF8304). Der Einsatz dieser ist kontext-bezogen was bedeutet, dass sie nur für bestimmte Akzente verwendet werden sollten.

Flächenfarben

Hintergründe dienen der inhaltlichen Abgrenzung auf einer Webseite, weshalb die Baloise auch hier einige Farbabstufungen definiert hat. Diese Abstufungen gehen jeweils vom Blauton #99D0E9 und Grauton #7E7E7E aus. Eine genaue Vorgabe zum Einsatz dieser Farben besteht nicht, wobei die Grautöne möglichst vermieden werden sollen.

## Logos

Damit Benutzer der Anwendung die Zuordnung zur Baloise direkt erkennen können, soll das Baloise-Group-Logo über alle Seiten der Anwendung präsent sein. Da voraussichtlich keine Funktionen für Soziale-Medien oder andere Kanäle bestehen sollten, ist dies auch das einzige geplante Logo.



## Formularelemente [[7]](#footnote-7)

Für eine tiefgreifende Umsetzung des Baloise-Webstyleguide müssten sämtliche Formularelemente eigenständig angepasst werden. Da dies jedoch über den Zeitrahmen der IPA hinausgehen könnte und keinen weiteren Einfluss auf die Funktion der Anwendung hat, sollte zur Veranschaulichung mindestens 1 Element übernommen werden. Folgende Elemente könnten dazu verwendet werden, wobei der Webstyleguide der Baloise auch weitere dokumentiert hat.

Textfelder

Ein Textfeld sollte durch eine blaue Umrandung (#008AC9) dargestellt werden. Als Platzhalter kann eine Info zur Zeichenbegrenzung eingefügt werden. Sowohl wenn der Benutzer das Feld anwählt, als auch während einer Eingabe, soll der Hintergrund des Textfeldes in einen abgewandelten Blauton übergehen.



Buttons

Navigations-Knöpfe wie ein «Absenden» Knopf sollten in ihrer Grundform als rechteckige Fläche mit orangem Hintergrund und weissem Text dargestellt werden. Bei Berührung mit dem Cursor des Benutzers sowie beim Anklicken sollte die Hintergrundfarbe in einen Blauton übergehen.



# Arbeitsjournal

## Tag 1 – 24.04.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 07:00 | Dokument erstellt/formatiert, Vue-Projekt auf GitHub initialisiert. | Elia Reutlinger | 02:00:00 | 01:30:00 |
| 08:30 | Konfigurationen vorgenommen (Design, i18n, PWA, Vue, Routing). | Elia Reutlinger | 04:00:00 | 02:30:00 |
| 12:00 | Authentifizierung implementiert + kleinere Optimierungen. | Elia Reutlinger | 02:00:00 | 03:30:00 |
|  | | | 08:00:00 | 07:30:00 |

Tagesthemen

Mit Beginn der IPA stand heute die grundlegende Konfiguration des Vue-Projekts sowie die Implementierung der Authentifizierung mit der API auf dem Plan. Nachdem ich das neue Projekt initialisiert und die Doku erstellt habe, nahm ich einige Einstellungen an Abhängigkeiten vor, damit ich weiterhin problemlos arbeiten kann. Die meisten Einstellungen konnte ich aus meinen alten Projekten übernehmen, wobei ich schneller fertig war als erwartet. Die Authentifizierung dauerte dafür länger, da ich mich etwas länger mit dem Routing nach Berechtigungen auseinandersetzen musste.

Hilfestellungen

Stackoverflow.com, Github.com, vuejs.org, alte Projekte (github.com/erleiuat/minska)

Reflexion +

Ich konnte alle Ziele des Tages erreichen und war sogar etwas schneller fertig als geplant. Deshalb konnte ich noch einige Verbesserungen am Design und der Performance der App vornehmen. Insgesamt bin ich mit der heutigen Arbeit sehr zufrieden, da ich mir durch diese Grundlagen nun ein besseres Bild vom Endprodukt machen kann.

Reflexion –

Ich habe mich im Zeitplan mit den heutigen Aufwänden ein wenig verschätzt, da ich dabei zu sehr von meinen bisherigen Projekten ausgegangen bin. Ich denke aber, dass sich das auf heute beschränken wird.

Erkenntnisse

Ich habe eine bessere Umsetzung fürs Routing nach Berechtigungen gefunden als in meinen bisherigen Projekten. Wahrscheinlich werde ich das irgendwann auch da verbessern.



## Tag 2 – 25.04.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 07:30 | PWA Eingerichtet | Elia Reutlinger | 02:00:00 | 00:30:00 |
| 08:00 | «Plan-Einstellungen» implementiert | Elia Reutlinger | 04:00:00 | 03:30:00 |
| 12:00 | «Einstellungen» implementiert | Elia Reutlinger | 02:00:00 | 02:30:00 |
| 14:30 | Mehrsprachigkeit angepasst | Elia Reutlinger | 00:00:00 | 01:30:00 |
|  | | | 08:00:00 | 08:00:00 |

Tagesthemen

Heute mussten noch die letzten Einstellungen zum PWA (Progressive-Web-App) gemacht werden, wobei ich mit diesen bereits gestern in der «Konfigurations-Phase» angefangen habe. Deshalb benötigte ich hier weniger Zeit, wobei ich meine Schätzung von 2 Stunden auch etwas übertrieben finde. Anschliessend habe ich mit den «Plan-Einstellungen» die erste Seite implementiert, wobei keine erwähnenswerten Probleme entstanden sind. Bei der «Einstellungen»-Seite gab es jedoch am Ende einen Fehler, da die Sprachauswahl zwar gespeichert, aber nicht von der Oberfläche übernommen wurde. Deshalb musste ich dort nochmals nachhaken, was etwas Zeit beansprucht hat.

Hilfestellungen

Stackoverflow.com, kazupon.github.io/vue-i18n

Reflexion +

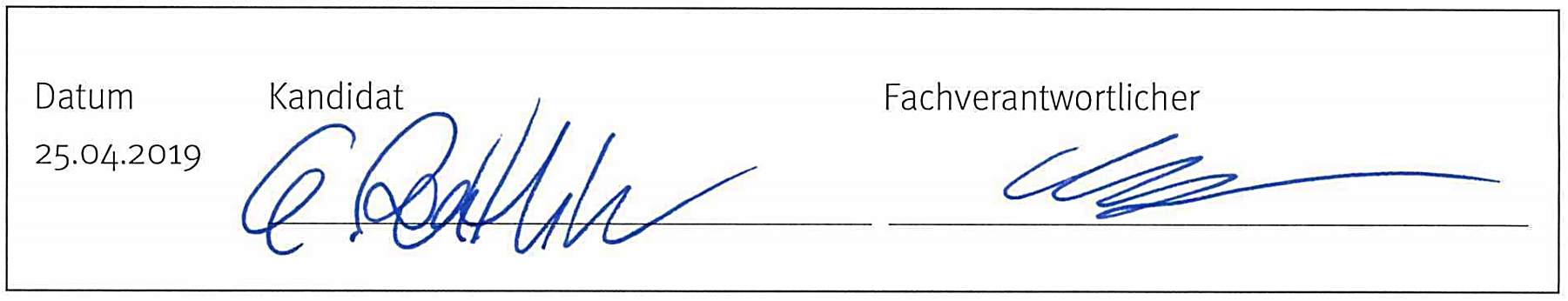
Ich hatte sozusagen Glück, dass ich trotz dem einen Fehler im Zeitplan blieb. Ich konnte ansonsten alles wie geplant implementieren und folgte den Schätzungen zum Zeitaufwand fast genau. Die beiden Seiten finde ich vom Design her ziemlich gelungen, da sie doch noch sehr übersichtlich und «modern/material» wirken, auch wenn viele Daten angezeigt werden. Ausserdem konnte ich das Formular zum Bearbeiten/Erstellen von Schichten nach Baloise-Standards kreieren, wodurch ich dieses Kriterium auch schon erfüllen konnte.

Reflexion –

Wegen dem Fehler habe ich im Internet länger nach einer Lösung gesucht aber keine gefunden. Ich habe die Abhängigkeit entfernt und neu installiert, was auch nicht erfolgreich war. Deshalb habe ich selbst über die Ursache nachgedacht und entdeckt, dass das Problem an der Verteilung der Übersetzungen lag (die Sprache lässt sich nicht in einer Komponente ändern, welche Übersetzungen definiert). Davon steht aber absolut nichts in der Doku des Plugins, was ich sehr schlecht finde.

Erkenntnisse

Obwohl ich das Problem mit der Anpassung der Sprache so noch nie hatte, werde ich es in zukünftigen Projekten direkt berücksichtigen können und damit einiges an Zeit sparen.



## Tag 3 – 26.04.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 07:00 | Doku-Kapitel erweitert/vertieft. | Elia Reutlinger | 00:00 | 01:00 |
| 08:00 | Implementierung «Plan» begonnen | Elia Reutlinger | 04:00 | 03:00 |
| 12:00 | Implementierung «Plan» | Elia Reutlinger | 04:00 | 04:00 |
|  | | | 08:00 | 08:00 |

Tagesthemen

Nach dem 1. Expertenbesuch gestern Nachmittag bin ich dem Rat von Herr Sommer gefolgt und habe deshalb heute als erstes die Kapitel der Doku erweitert. Dabei versuchte ich die Kriterien und alle Punkte, welche die Entwicklung betreffen, direkt zu beachten. Anschliessend konnte ich mit dem Plan beginnen, welcher sozusagen den wichtigsten Teil der ganzen Anwendung darstellt. Ich benötigte einige Versuche, bis ich eine funktionsfähige Struktur der Komponenten fand, bei welcher die Daten «reaktiv» bleiben. Dabei musste ich auch auf das Design achten, da nun unterschiedliche Tabellen zum Einsatz kommen, welche sich aber Design-Eigenschaften teilen müssen. Obwohl ich mit dem Design schon soweit fertig bin, konnte ich noch keine optimale Lösung für die Reaktivität finden, vor allem auf der Ebene der einzelnen Einsätze (unterste Ebene/Komponente).

Hilfestellungen

Stackoverflow.org, vuejs.org / vuex.vuejs.org, alligator.io

Reflexion +

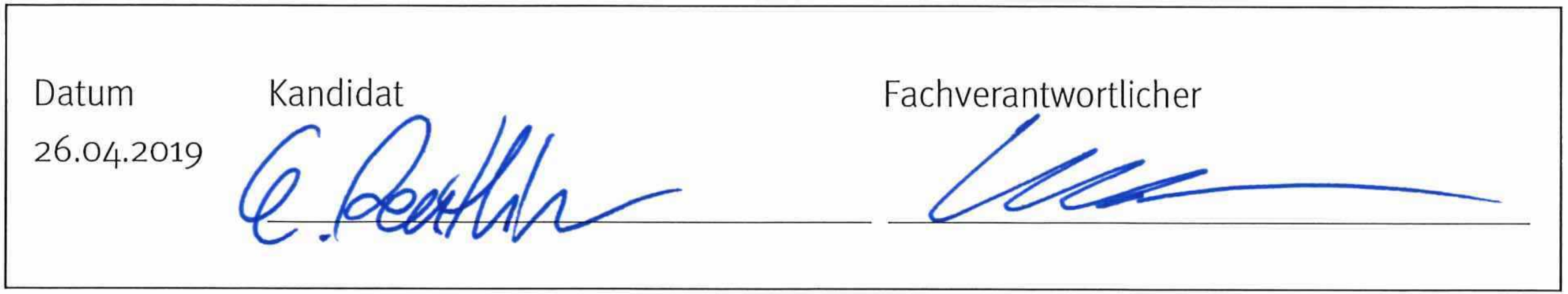
Durch die Vertiefung der Kapitel weiss ich besser, was in der Dokumentations-Phase auf mich zu kommt und kann bereits jetzt einiges notieren, was ich später noch brauche bzw. berücksichtigen sollte.

Reflexion –

Die Reaktivität von Daten zwischen Komponenten ist eine etwas komplexere Thematik als ich es mir vorgestellt habe. Ich musste heute schon ziemlich grübeln und viel recherchieren, um einen Fortschritt zu erzielen, wobei ich aber noch keine endgültige Lösung finden konnte.

Erkenntnisse

Vuex (ein globaler Speicher für Vue.js) löst nicht in jedem Fall eine Aktualisierung in einer Komponente aus. Da der Plan eine sehr hohe Dynamik der Daten fordert (variable Anzahl Benutzer und geplante Einsätze) musste ich (und muss ich noch) mich dazu genauer informieren.



## Tag 4 – 30.04.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 07:00 | Implementierung «Plan» (Vormittag) | Elia Reutlinger | 04:00 | 04:00 |
| 12:00 | Implementierung «Plan» (Nachmittag) | Elia Reutlinger | 00:00 | 03:00 |
| 15:00 | Implementierung «Dashboard» | Elia Reutlinger | 02:00 | 01:00 |
|  | | | *08:00* | 08:00 |

Tagesthemen

Heute sollte laut Zeitplan die Plan-Seite fertig implementiert werden. Dazu musste ich mich zuerst genauer über die Reaktivität von Komponenten informieren, weil ich letzte Woche dort stehen geblieben bin. Ich testete unterschiedliche Möglichkeiten um die Einsätze dynamisch im globalen Speicher abzulegen, was mich einiges an Zeit kostete. Als ich die Lösung fand, hatte ich bereits länger benötigt als im Zeitplan angegeben. Ich musste dann aber noch einige andere Komponenten an die neue Lösung anpassen, was noch mehr Zeit benötigte. Schlussendlich habe ich zwar den Plan fertiggestellt, aber die anderen Tagesziele (Dashboard- & Support-Seite) nicht ganz erreicht. Ich denke aber, dass ich trotzdem pünktlich mit der Entwicklungs-Phase fertig sein werde, da ich für die letzten Schritte wahrscheinlich nicht so viel Zeit benötige wie geplant.

Hilfestellungen

Stackoverflow.org, vuejs.org/v2/guide/reactivity, vuejs.org/v2/guide/computed, vuex.vuejs.org

Reflexion +

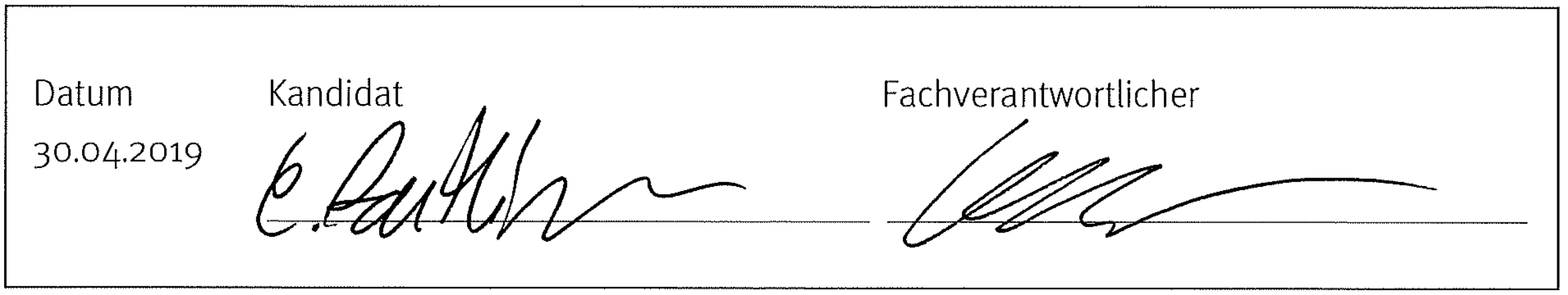
Die schlussendliche Umsetzung des Plans finde ich äusserst gut. Nun kann man die Daten, die der Plan nutzt, überall dynamisch verändern und der Plan aktualisiert sich zuverlässig. Auch wenn es länger gedauert hat denke ich, dass diese Zeit so richtig investiert wurde.

Reflexion –

Mit der neuen Lösung musste ich einige andere Komponenten anpassen, die ich bereits zuvor beendet hatte. Ohne diese Anpassungen hätten die Komponenten aber gewisse Daten nicht richtig verarbeitet, sodass keine Reaktivität entstanden wäre. Die Anpassungen waren also notwendig.

Erkenntnisse

Die Lösung war die Funktion «Object.assign()» mit einer Reinitialisierung des betroffenen Objektes, wodurch sich Daten bzw. Objekte auch nach der Initialisierung der Vue-Instanz als dynamische Objekte zum globalen Speicher hinzufügen lassen. In Verbindung mit einigen «Watchers» sind die einzelnen Benutzer-Zeilen bzw. Einsätze nun auch reaktiv/dynamisch.



## Tag 5 – 02.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligt | H-«Soll» | H-«Ist» |
| 07:30 | Dashboard fertig implementiert | Elia R. | 00:00 | 01:00 |
| 08:30 | Support-Seite implementiert | Elia R. | 00:00 | 01:00 |
| 09:30 | Fehlermeldungen implementiert / verbessert | Elia R. | 02:00 | 01:30 |
| 11:00 | Zahlreiche Verbesserungen | Elia R. | 02:00 | 01:00 |
| 12:30 | Weitere, kleinere Verbesserungen (App fertig) | Elia R. | 00:00 | 00:30 |
| 13:00 | Doku: Aufgabenstellung & Analyse fertig | Elia R. | 02:00 | 01:30 |
| 14:30 | Doku: Planung & Standards begonnen | Elia R. | 02:00 | 01:30 |

Tagesthemen

Heute habe ich das Dashboard und die Support-Seite fertiggestellt. Die Fehlermeldungen sind ebenfalls fertig und wurden noch weiter verbessert. Anschliessend habe ich noch sehr viele Punkte verbessert, wobei die Responsiveness, Performance und Firmenstandards im Fokus lagen. Bei einem kurzen Test erschien noch ein Bug bei der Verwaltung der Tageszeiten, welcher aber ohne viel Aufwand behoben werden konnte. Zum Schluss habe ich den Code nochmals verschönert/formatiert und weitgehend kommentiert. Damit war die Implementierungsphase auch abgeschlossen und ich begann mit den ersten Kapiteln der Doku. Dabei bemerkte ich, dass die Einordnung im Zeitplan nicht viel Sinn ergibt, weshalb ich den Zeitplan leicht angepasst habe. Ich konnte 2 ganze Kapitel beenden und auch schon die Firmenstandards und das Projektmanagement beginnen.

Hilfestellungen

webstyleguide.baloise.com, vuetifyjs.com

Reflexion +

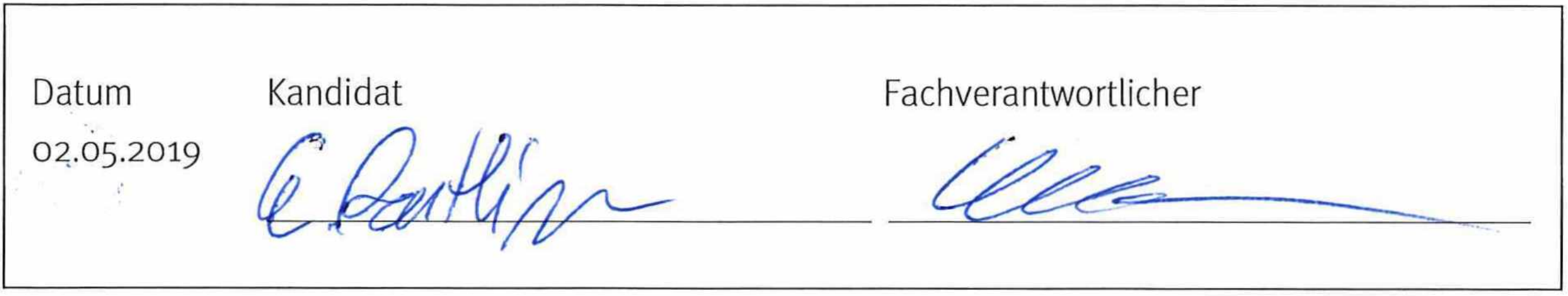
Ich finde es sehr gut, dass ich fast genau nach Plan mit der Implementierung fertig wurde. Obwohl es einiges nachzuholen gab, schaffte ich es, alles zu erledigen und sogar noch viel zu verbessern. Ich konnte auch noch meinen Code «verschönern», indem ich alles formatiert und kommentiert habe. Dabei würde ich sogar sagen, dass ich noch nie einen saubereren Code geschrieben habe.

Reflexion –

Die Bearbeitung der Tageszeiten funktionierte nicht mehr richtig und musste verbessert werden. Des Weiteren schienen die Schriftarten nicht ganz korrekt zu sein, worauf ich die Offiziellen von Intern herunterladen musste. Dabei habe ich auch gleich weitere Standards verbessern müssen.

Erkenntnisse

Ich habe herausgefunden, wie man Schriftarten dynamisch nachladen kann (durch eine Einstellung vom Plugin «Webpack»). Dadurch wurde die Performance der App sehr viel verbessert.



## Tag 6 – 03.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligt | H-«Soll» | H-«Ist» |
| 07:30 | Doku: Projektmanagement & Planung fertiggestellt | Elia R. | 02:00 | 01:30 |
| 09:00 | Doku: Auswertung der Aufgabenstellung | Elia R. | 02:00 | 01:00 |
| 10:00 | Doku: Verbesserungen & Notizen | Elia R. | 00:00 | 01:00 |
| 12:00 | Doku: Kapitel «Initialisierung» begonnen | Elia R. | 02:00 | 4:30 |

Tagesthemen

Heute konnte ich die Kapitel «Projektmanagement und Planung» und die Auswertung fertigstellen. Da dabei keine uneingeplanten Arbeiten entstanden investierte ich noch etwas Zeit in zahlreiche Verbesserungen (bzgl. Detailgrad & Grammatik). Dabei hatten auch die Erkenntnisse Einfluss, welche ich durch die Fragen des heutigen Expertenbesuchs erhalten habe. Am Kapitel «Initialisierung» konnte ich sogar länger arbeiten, da die vorherigen Kapitel nicht so lange dauerten.

Hilfestellungen

IPA-Experten, IPA-Kriterienkatalog, webstyleguide.baloise.com, Wikipedia.org, Baloise.ch

Reflexion +

Ich bin nach wie vor äusserst froh darüber, dass die Implementierung abgeschlossen und ohne grössere Probleme verlaufen ist. Der Expertenbesuch hat ebenfalls einige Fragen geklärt, wodurch ich nun ein ziemlich klares Ziel vor Augen habe. Des Weiteren finde ich es gut, dass ich heute bereits mehr erledigen konnte, als im Zeitplan geplant.

Reflexion –

Das Standardkriterium Nr. 1 aus Teil A sagt, dass die korrekte Anwendung der Projektmanagement-Methode (in unserem Fall das Kanban-Board) im IPA-Bericht ersichtlich sein muss. Ich kann mir jedoch noch nicht konkret vorstellen, wie ich das realisieren sollte, da unser Kanban ausschliesslich auf GitHub.com geführt wird. Möglicherweise werde ich eine Art von Diagramm einsetzen, wobei ich mir hier noch nicht ganz sicher bin.

Erkenntnisse

Nach dem Expertenbesuch weiss ich nun um einiges besser, was ich wie in der Dokumentation beschreiben sollte.

|  |
| --- |
| Datum Kandidat Fachverantwortlicher  03.05.2019  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Tag 7 – 07.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |

Tagesthemen

…

Hilfestellungen

…

Reflexion +

…

Reflexion –

…

Erkenntnisse

…

|  |
| --- |
| Datum Kandidat Fachverantwortlicher  07.05.2019  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Tag 8 – 08.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |

Tagesthemen

…

Hilfestellungen

…

Reflexion +

…

Reflexion –

…

Erkenntnisse

…

|  |
| --- |
| Datum Kandidat Fachverantwortlicher  08.05.2019  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Tag 9 – 09.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |

Tagesthemen

…

Hilfestellungen

…

Reflexion +

…

Reflexion –

…

Erkenntnisse

…

|  |
| --- |
| Datum Kandidat Fachverantwortlicher  09.05.2019  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Tag 10 – 10.05.2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zeit | Tätigkeit | Beteiligte | Aufwand «Soll» | Aufwand «Ist» |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |
| 00:00 |  | Elia Reutlinger | 00:00:00 | 00:00:00 |

Tagesthemen

…

Hilfestellungen

…

Reflexion +

…

Reflexion –

…

Erkenntnisse

…

|  |
| --- |
| Datum Kandidat Fachverantwortlicher  10.05.2019  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Abschlussbericht

## Ist-Soll

## Schwierigkeiten

## Fazit zum Projekt

### Entstandene Abweichungen zur Aufgabenstellung

## Persönliches Fazit

## Schlussreflexion

[**TODO**: Unterschiedlichste Sachen reflektieren **UND** vergleichen. Z.b: …]

### Arbeitsmethodik

### Werkzeuge

Teil 2: Dokumentation und Umsetzung

# Initialisierung

## Analyse Vue.js

Vue.js ist ein JavaScript-Framework zur Realisierung von Single-Page-Webanwendungen. Durch die Nutzung eines Mapping-Routers kann eine Webanwendung jedoch in mehrere Seiten unterteilt werden. Hauptmerkmale von Vue.js sind unter anderem:[[8]](#footnote-8)

* Komponentenbasierte Umsetzung (Wiederverwendbarkeit)
* Viel Flexibilität
* Hohe Performance
* Grosse Community (= gute Dokumentation und reichlich Unterstützung)

Bei korrekter Umsetzung des Konzepts kann mit Vue.js eine professionelle Webanwendung implementiert werden, die mit eigenen Scripts und Komponenten beliebig erweitert werden kann. Deshalb sollte sich Vue.js zur Realisierung einer Einsatzplanung eignen, während für bestimmte Anforderungen noch zusätzliche Abhängigkeiten eingebunden wurden. Zu den wichtigsten gehören dabei folgende:

* Durch «Vuetify» lässt sich ein grundlegendes Design schnell umsetzen, wobei zahlreiche Variablen individuell angepasst werden können. Die mitgelieferten Komponenten entsprechen den Material-Design Standards und lassen sich dynamisch einbinden.
* Für eine mehrsprachige Anwendung kann «Vue-i18n» genutzt werden. Dies ist das gängigste Modul, um Übersetzungen innerhalb von Komponenten anwendbar zu machen.
* Mit «Axios» können Anfragen an einen Server bzw. eine API asynchron durchgeführt werden. Dabei lassen sich die Parameter einer Anfrage beliebig anpassen.

Durch weitere Entwicklungs-Abhängigkeiten wurde die Implementierung einfacher gemacht, wodurch ich mich stärker auf die wichtigen Punkte konzentrieren konnte. Schlussendlich war es also möglich, durch Vue.js eine Anwendung performant, stabil und schnell umzusetzen.

### Konzept

Vue.js baut auf einem gängigen Konzept der Realisierung von Webanwendungen auf. Dabei kommen Komponenten zum Einsatz, welche aus einer Darstellungs-Struktur (HTML), mehreren Funktionen (JS) und Design-Definitionen (CSS) bestehen. Dadurch sollen die Elemente einer Anwendung modularisiert werden. Komponenten können dann beliebig als View oder in anderen Komponenten verwendet und dynamisch eingebunden werden.

Views stellen die unterschiedlichen Ansichten dar, wobei hier in der Regel nicht die Hauptfunktionen eingebaut werden. Es wird definiert, welche Komponenten in der View dargestellt werden, und wie diese miteinander in Verbindung stehen.

Der Sinn hinter den Komponenten ist die Wiederverwendbarkeit von Elementen, wodurch diese an unterschiedlichen Stellen genutzt werden können und bei Anpassungen nur eine Stelle im Code bearbeitet werden muss.

[**TODO**: Bild Vue.js Konzept]

Mit Vuex entsteht zudem eine Art globaler Speicher (der sogenannte «Store»), auf welchen alle Komponenten zugreifen können. Dabei ist es möglich den Inhalt einer Komponente mit diesem Speicher zu verknüpfen, sodass bei Veränderungen an einer anderen Stelle der Inhalt in der verknüpften Komponente automatisch aktualisiert wird.

### Vergleich

Zur Auswahl standen auch weitere Frameworks wie React oder Angular, mit welchen ich bisher keine oder nur wenig Erfahrung habe. Nach einer kurzen Evaluation der Frameworks stellte sich für mich heraus, dass React und Angular sehr viele zusätzliche Abhängigkeiten liefern, die in dieser Anwendung nicht notwendig sind. Da diese die Performance der Anwendung beeinflussen können, und einen Zeitaufwand für die Konfiguration benötigen, habe ich mich gegen die Verwendung dieser Frameworks entschieden.

## Vertiefte Studie Ist-Zustand & Umgebung

Die Einsatzplanung am Helpdesk wird mit einer komplexen Excel-Tabelle geführt. Darin wird ein gesamtes Semester abgebildet sowie alle Mitarbeiter des Teams, was die Übersicht sehr einschränkt. Abhilfe würde hier beispielsweise die Darstellung von nur einem Monat bieten.

[**TODO**: Bild von Excel]

Die Berechtigungen werden durch einen Passwort-Schutz umgesetzt, wodurch nur bestimmte Personen Änderungen an der Tabelle vornehmen können. Andere können das Dokument lediglich einsehen aber nicht bearbeiten. Damit alle darauf zugreifen können, wurde es auf einem Netzlaufwerk gespeichert. Dabei wird es bei einem Zugriff für andere Mitarbeiter gesperrt, was die Dynamik einschränkt. Gleichzeitig besteht hier ein Mehraufwand durch das Suchen der Datei und die Eingabe des Passwortes, was durch eine Webanwendung nicht mehr notwendig wäre.

Um Änderungen am Plan vorzunehmen, wurden Makros eingebaut. Die Nutzung von Makros birgt jedoch ein Sicherheitsrisiko und kann zu Verständnisproblemen führen. Klar definierte Funktionen zur Bearbeitung mit einer benutzerfreundlichen Darstellung würden dies optimieren. Eine Webanwendung kann bei korrekter Implementierung auch die Sicherheitsrisiken eliminieren.

Die Umgebung, in welche die Anwendung nach der IPA eingebettet werden soll, besteht aus der internen Infrastruktur der Baloise. Dabei ist für die Anwendung selbst ein Server eingeplant, welcher den Code kompiliert und intern veröffentlicht. Der Code wird dazu aus dem GitHub-Repository geladen, auf welchem die IPA stattfindet. Als wichtigste und aktuell einzige Schnittstelle gilt die Pelan-API, welche sich ebenfalls auf einem internen Server befindet und Zugriffe über Medusa erlaubt.

[**TODO**: Bild der Entwicklungsumgebung]

[**TODO**: Schnittstelle Medusa beschreiben]

## Umsetzung Projektmethode & Vorgehensmodell

[**TODO**: Erwähnen warum Kanban gewählt wurde (…weil oft in Baloise undso) und warum nicht etwas anderes gewählt wurde]

Die Projektmethode während der IPA wurde mit der detaillierten Aufgabenstellung auf Kanban festgelegt. Kanban bildet die Arbeitsschritte der Implementierung als einzelne «Tasks/Aufgaben» ab, und ordnet diese unterschiedlichen Stationen zu. Diese Stationen beschreiben den aktuellen Stand des Tasks, wobei ich mich für übliche Stationen der Anwendungsentwicklung entschieden habe:

1. Backlog: Hier werden alle Tasks eingeteilt, die im Projekt benötigt werden und noch nicht begonnen wurden. Sie können durch die Anforderungen der Aufgabenstellung oder während der Umsetzung dieser Anforderungen entstehen.
2. Waiting: Diese «Zwischenstation» enthält alle Tasks, welche gerade nicht bearbeitet werden können, da dazu andere Tasks und/oder Abklärungen gemacht werden müssen.
3. In Progress: Wenn ein Task aktiv ist, wird er in dieser Station eingeteilt. Hier findet die Umsetzung bzw. Implementierung des Tasks statt, wobei dies zusammen mit dem Entwurf geschieht.
4. Testing: Bevor ein Task abgeschlossen werden kann, sollte mit dieser Station die korrekte Umsetzung getestet werden. Bei gewissen Tasks wie Abklärungen kann diese Station übersprungen werden.
5. Done: Alle erledigten Tasks werden hier abgelegt. Sie befinden sich dabei in einem Zustand, in welchem sie problemlos im Betrieb eingesetzt werden können. Dazu kommt eine Gelöst-Markierung. \*

\* Das Kanban wird auf GitHub durch ein «Project-Board» dargestellt. Die einzelnen Tasks sind dabei GitHub-Issues, welche nach Abschluss als «Closed/Gelöst» markiert werden können. Issues können bei Bedarf auch Labels tragen, welche auf die Art des Issues bzw. der Aufgabe schliessen lassen.

[**TODO**: Bild von Kanban auf GitHub]

Kanban bildet an sich bereits eine Art von Vorgehensmodell, bei welchem die Anzahl paralleler Tasks und somit die Durchlaufzeit möglichst klein gehalten wird. Dadurch können Probleme und Engpässe schnell sichtbar gemacht werden. Die Stationen bzw. Arbeitsschritte eines Kanban-Boards entsprechen häufig denen des Wasserfallmodells, wobei für die IPA die Schritte «Anforderung», «Entwurf» und «Wartung» weggelassen wurden. Die Anforderungen wurden bereits vor Beginn der IPA definiert, und der Entwurf wird aus zeitlichen und technischen Gründen in der Implementierung erarbeitet.[[9]](#footnote-9)

## Ziele

### Projekt

Das Endprodukt des Projekts soll dem Helpdesk eine Alternative zur Excel-Tabelle liefern. Dabei sollen die Übersichtlichkeit und Verwaltung sichtlich verbessert worden sein. Weitere Ziele in Berücksichtigung der IPA Kriterien sind:

* Implementierung mit Führung eines Kanban-Boards.
* Dokumentierung und Veröffentlichung des Fortschritts auf GitHub.
* Für die Produktion einsetzbares Endprodukt.
* Wiedererkennbarkeit der Baloise im Endprodukt.
* Mehrsprachige Anwendung mit Erweiterungspotential.
* Berücksichtigung gängiger Sicherheitskriterien.

### Persönlich

Persönliche Ziele, welche ich mir selbst für die IPA gesetzt habe, sind:

* Vertiefung des Fachwissens zu Vue.js.
* Problemlose Abwicklung durch gute Planung (Zeitplan).
* Termingerechter Abschluss des Projekts.
* Umsetzung einer optimalen, produktiv anwendbaren Anwendung.

## Anforderungen

Das Ziel der IPA ist eine funktionsfähige Anwendung. Dazu bestehen funktionale Anforderungen an die Anwendung sowie nicht-funktionale während der Entwicklung, wobei die folgend definierten das Minimum an zu erfüllenden Anforderungen darstellen. Sie wurden durch die detaillierte Aufgabenstellung der IPA erstellt und zusätzlich vertieft, wobei die gesamten Anforderungen ersichtlich werden.

[**TODO**: Use-Case Diagramm über die gesamte Anwendung]

### Funktional

* Grundlegende Voraussetzungen
  + Einhaltung definierter Firmenstandards
  + Benutzerfreundlichkeit über die gesamte Anwendung
  + Unveränderte Verwendbarkeit für den produktiven Betrieb (1:1)
  + Einhaltung von Coding-Standards bzw. Best-Practice
  + Mehrsprachigkeit (mindestens 2 Sprachen)
  + Absicherung gegen gängige Angriffe
* Funktionen
  + Einsätze darstellen
  + Einsätze erstellen, bearbeiten und entfernen
  + Schichten und Tageszeiten erstellen, bearbeiten und löschen
  + Berechtigungen von Teammitgliedern bearbeiten
  + Sprache anpassen

### Nicht-Funktional

* Dokumentation des Fortschritts auf GitHub
* Dokumentierung von möglichen Änderungen und Problemen
* Dokumentation des Projekts als IPA-Bericht
  + Führung eines Arbeitsjournals
  + Testkonzept und Testprotokoll
  + Sicherheitskonzept und Sicherheitsprüfung
* Einsatzfähige Demo-Version mit Testdaten
* Präsentation des Projekts

## Informationssicherheit & Datenschutz

Datenschutz ist bei der Baloise eine Voraussetzung für alle Anwendungen. Man muss strenglichst darauf achten, dass keine Daten nach aussen gelangen, die auf betriebsinterne Abläufe oder Personendaten schliessen lassen könnten.[[10]](#footnote-10)

Da die Anwendung selbst jedoch keine Daten speichert, sondern lediglich mit dem Back-End kommuniziert, müssen hier keine weiteren Vorsichtsmassnahmen getroffen werden. Das Back-End entscheidet auf welche Daten von wem zugegriffen werden kann, und übernimmt den gesamten Authentifizierungs-Prozess. Die Verbindung zwischen Front-End und Back-End benötigt in Berücksichtigung der aktuellen Anforderungen auch keine zusätzliche Massnahme, da beide nur betriebsintern zugänglich sein sollen.

Zu beachten ist jedoch, dass die Entwicklung öffentlich auf GitHub stattfindet. Entsprechend sollten keine möglichen Schlüssel oder Ähnliches, was in Zusammenhang mit internen Abhängigkeiten steht nach aussen gelangen.

Sicherheitsrisiken, welche im Front-End zu Problemen führen könnten wie DOM-Based und Reflected Cross-Site-Scripting müssen unabhängig vom Back-End geprüft werden. Deshalb werden dazu ein Sicherheitskonzept sowie entsprechende Tests/Überprüfungen erstellt.

## Konzepte

### Testkonzept

### Sicherheitskonzept

# Realisierung

## Grundlegende Konfiguration

### Mehrsprachigkeit

### Authentifizierung

### Ansichten / Routes

## Umsetzung Design

[**TODO**: Erwähnen, wo was wie von Firmenstandards kommt und wie Design umgesetzt wurde.]

## Umsetzung der Ansichten

### Support

### Dashboard

### Einstellungen

### Plan-Einstellungen

### Dienstplan

## Testprotokoll

## Sicherheitsprüfung

[**TODO**: Mindestens 3 Injektionsversuche sind abgewehrt & dokumentiert]

# Abschluss

## Auswertung des Kanban-Boards

[**TODO** Issue-Verlauf während der IPA irgendwie grafisch darstellen]

## Funktionen & Verbesserungen für die Zukunft

## Danksagung

[**TODO**: Auch die erwähnen, die korrigiert haben.]

# Anhang

## Quellenverzeichnis

[**TODO**: Erwähnen wie alles korrigiert wurde.]

## Abbildungsverzeichnis

[**TODO**: Bilder/Diagramme jeweils kurz erklären! (Standardkriterium Teil2: #9)]

[**TODO**: Alle Bilder beschriften]

1. Apachefriends.org [↑](#footnote-ref-1)
2. Wikipedia.org – Atom (Texteditor) [↑](#footnote-ref-2)
3. Netlify.com [↑](#footnote-ref-3)
4. Jwt.io [↑](#footnote-ref-4)
5. Webstyleguide.baloise.com [↑](#footnote-ref-5)
6. Brandportal.baloise.com – Corporate Design Manual [↑](#footnote-ref-6)
7. Webstyleguide.baloise.com [↑](#footnote-ref-7)
8. Wikipedia.org – Vue.js [↑](#footnote-ref-8)
9. Wikipedia.org – Kanban (Softwareentwicklung) [↑](#footnote-ref-9)
10. Baloise.ch - Datenschutzbestimmungen [↑](#footnote-ref-10)