**Projekt-Schlussbericht**

**Projekttitel:** **S**chlaues**R**aum**B**uchungs**S**ystem (SRBS)

**Autoren:** Olivier Winkler

Domenico Winkelmann

**Datum:** 07.01.2020

**Inhaltsverzeichnis**

[Teil 1 5](#_Toc29284570)

[1. Aufgabenstellung und Management Summary 6](#_Toc29284571)

[2. Deklaration der Vorkenntnisse 7](#_Toc29284572)

[3. Deklaration der Vorarbeiten und fremden Code 7](#_Toc29284573)

[4. Deklaration der verwendeten Firmenstandards 7](#_Toc29284574)

[5. Zeitplan 8](#_Toc29284575)

[5.1 Phase Vorbereitung 9](#_Toc29284576)

[5.2 Phase Initialisierung 9](#_Toc29284577)

[5.3 Phase Konzept 9](#_Toc29284578)

[5.4 Phase Realisierung 9](#_Toc29284579)

[5.5 Phase Einführung 9](#_Toc29284580)

[6. Arbeitsprotokoll 10](#_Toc29284581)

[6.1 Arbeitsjournal vom Dienstag 29.10.2019 10](#_Toc29284582)

[6.2 Arbeitsjournal vom Dienstag 05.11.2019 11](#_Toc29284583)

[6.3 Arbeitsjournal vom Dienstag 12.11.2019 12](#_Toc29284584)

[6.4 Arbeitsjournal vom Dienstag 19.11.2019 13](#_Toc29284585)

[6.5 Arbeitsjournal vom Dienstag 26.11.2019 14](#_Toc29284586)

[6.6 Arbeitsjournal vom Dienstag 29.10.2019 15](#_Toc29284587)

[6.7 Arbeitsjournal vom Dienstag 05.11.2019 16](#_Toc29284588)

[6.8 Arbeitsjournal vom Dienstag 12.11.2019 17](#_Toc29284589)

[6.9 Arbeitsjournal vom Dienstag 19.11.2019 18](#_Toc29284590)

[6.10 Arbeitsjournal vom Dienstag 26.11.2019 19](#_Toc29284591)

[Teil 2 20](#_Toc29284592)

[7 Situationsanalyse 21](#_Toc29284593)

[7.1 Ausgangslage 21](#_Toc29284594)

[7.2 Stärken 21](#_Toc29284595)

[7.3 Schwächen 21](#_Toc29284596)

[8. Systemziele 21](#_Toc29284597)

[8.1 Rahmenbedingungen 21](#_Toc29284598)

[8.2 Abgrenzung 22](#_Toc29284599)

[9 Lösungsvorschläge 22](#_Toc29284600)

[9.1 Variantenübersicht 22](#_Toc29284601)

[9.2 Beschreibung der Varianten 22](#_Toc29284602)

[9.3 Bewertung der Varianten (Tabelle) 23](#_Toc29284603)

[9.4 Lösungsbeschreibung 23](#_Toc29284604)

[10 Systemarchitektur 25](#_Toc29284605)

[10.1 Systemdesign 25](#_Toc29284606)

[10.2 Schnittstellendefinitionen 26](#_Toc29284607)

[10.3 Sicherheit 26](#_Toc29284608)

[11. Testkonzept und Testspezifikationen 26](#_Toc29284609)

[11.1 Systemtest - Testspezifikations 26](#_Toc29284610)

[11.3 Testprozedur 29](#_Toc29284611)

[12. Einführungskonzept 30](#_Toc29284612)

[12.1 Einführungsplan 30](#_Toc29284613)

[12.2 Migrationsplan 31](#_Toc29284614)

[12.3 Ausbildungsplan 31](#_Toc29284615)

[13. Testprotokoll 32](#_Toc29284616)

[13.1 Testobjekt 32](#_Toc29284617)

[13.2 Testresultate 33](#_Toc29284618)

[13.3 Testauswertung 34](#_Toc29284619)

[13.4 Abnahme 35](#_Toc29284620)

[14. Benutzerdokumentation / Anleitungen 36](#_Toc29284621)

[14.1 Konfigurations-Dokumentation 36](#_Toc29284622)

[14.2 Benutzerhandbuch 41](#_Toc29284623)

[14.3 Supporthandbuch 43](#_Toc29284624)

[15. Projekterfahrung 44](#_Toc29284625)

[15.1 Erfahrung Winkler Olivier 44](#_Toc29284626)

[15.2 Erfahrung Winkelmann Domenico 44](#_Toc29284627)

[Teil 3 45](#_Toc29284628)

[16. Dokumentation Quellcode 46](#_Toc29284629)

[16.1 Code von Frontend 46](#_Toc29284630)

[16.2 Code von Backend 47](#_Toc29284631)

[17. Literaturverzeichnis 50](#_Toc29284632)

[18. Glossar 50](#_Toc29284633)

**Abbildungsverzeichnis**

[Abbildung 1 7](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284634)

[Abbildung 2 7](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284635)

[Abbildung 3 8](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284636)

[Abbildung 4 25](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284637)

[Abbildung 5 30](#_Toc29284638)

[Abbildung 6 31](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284639)

[Abbildung 7 36](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284640)

[Abbildung 8 36](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284641)

[Abbildung 9 36](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284642)

[Abbildung 10 37](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284643)

[Abbildung 11 37](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284644)

[Abbildung 12 38](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284645)

[Abbildung 13 38](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284646)

[Abbildung 14 38](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284647)

[Abbildung 15 39](#_Toc29284648)

[Abbildung 16 39](#_Toc29284649)

[Abbildung 17 40](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284650)

[Abbildung 18 40](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284651)

[Abbildung 19 41](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284652)

[Abbildung 20 41](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284653)

[Abbildung 21 42](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284654)

[Abbildung 22 42](#_Toc29284655)

[Abbildung 23 43](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284656)

[Abbildung 24 43](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284657)

[Abbildung 25 43](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284658)

[Abbildung 26 46](file:///H:\3.%20Lehrjahr%202019%20-%202020\M306\Schluss\Schlussbericht_Olivier_Domenico.docx#_Toc29284659)

# Teil 1

## 1. Aufgabenstellung und Management Summary

Unser Projektteam besteht aus SBB-Lernenden. Wir haben uns bei der Suche nach einer Projektidee gefragt ob es eventuell ein Produkt in der SBB geben soll, welches noch nicht entwickelt ist aber in naher Zukunft in Planung kommen soll. Wir haben dann ein wenig recherchiert und herausgefunden, dass eine neue Raumplanungssoftware in der SBB eingesetzt werden soll.

Als die Idee festgelegt wurde, konnten wir mit unserem Auftraggeber das Projekt besprechen. Die Applikation soll eine Webapplikation sein, die auf diversen Geräten wie kleinen Touchscreens oder Smartphones korrekt angezeigt werden soll. Dazu soll sie der Funktionalität ähnlich wie ihrem Vorgänger sein. Das Problem beim Vorgänger war die Benutzerfreundlichkeit und die Stabilität. Dies sollte mit unserem Produkt verbessert werden. Da diese Applikation für die SBB entwickelt wurde, müssen speziell definierte Standards eingehalten und erfüllt werden. All dies wurde in einem Vorgespräch besprochen.

Das Projekt wird mittels Hermes umgesetzt im Modul 306. Dieses wird von dem Projektteam einmal pro Woche für 3 Stunden besucht. Mit Hermes wird das Projekt in verschiedene Phasen unterteilt, welche so die Produktivität und Quantität erheblich erhöhen. In der Gibb, Zimmer 304 wird da Projekt umgesetzt während dem ersten Semester im dritten Lehrjahr.

Die Firmenstandards werden nach SBB-Art eingehalten. Bei unserem Webprojekt können spezielle Frameworks verwendet werden, die spezielle SBB-Komponente verwenden. Zudem ist das Projekt bezüglich der Technologie eingeschränkt. Der SBB-Standard verwendet nicht alle Programmiersprachen / Frameworks. Die entwickelte Applikation ist in den SBB-Technologien umgesetzt worden. Als Frontend wurde Angular verwendet. Angular ist in der SBB die meist verbreitete Websprache bezüglich Frontendbereich. Im Backend wurde auf Springboot gesetzt, welches auch sehr bekannt intern ist.

Die Applikation wurde in zwei Haupteditoren umgesetzt. Springboot (Java) wurde vor allem in der Entwicklungsumgebung Intellij entwickelt, wobei Angular in Visual Studio Code gecodet wurde. Als Quellcodeverwaltung wurde Gitlab verwendet. Die Gibb hat eigene Server. Dank Gitlab konnte der Gitflow verwendet werden, welches das gleichzeitige Arbeiten an einem gleichen Teil vereinfacht. Für unsere Dokumente wurden die Shares der Gibb verwendet.

Während dem gesamten Projekt könnten Risiken eintreten. Diese wurden in jeder Phase falls nötig überarbeitet. Mit einer solchen Risikoanalyse konnten die Risiken minimiert werden und bei tatsächlichem Eintreten vermindert werden. Das grösste Risiko in unserem Projekt war eine Abwesenheit und somit die wäre die Projektteamkapazität nur noch bei einer Person, was zu Verzögerungen hätte führen können. Zudem hätten diverse Risiken bezüglich der Daten auftreten können. All diese Risiken trafen aber während des Projektes nicht ein.

## 2. Deklaration der Vorkenntnisse

Das Projekt wurde wie schon erwähnt im SBB-Standard entwickelt. Da wir bereits in der Vergangenheit mit diesen Komponenten gearbeitet haben, waren wir an diesen Stellen sehr schnell.

Zudem wurde der Variantenentscheid aus der Konzeptstudie miteinbezogen. Dort wurde die beste Variante ausgewählt, basierend auf den besten Vorkenntnissen. In unserem Projekt sind die Kenntnisse eher in der Webumgebung gefragt. Im Frontend wird Angular verwendet. Angular besteht einerseits aus Typescript, einer Abweichung von Javascript mit sehr ähnlichen Syntax, CSS (SCSS) und HTML. Diese Komponente wurden bereits in den zwei vorherigen Lehrjahren erworben. In diversen Modulen oder speziell im Betrieb wurden die Sprachen angeschaut. Im Backend wird Springboot verwendet, welches mit Java funktioniert. Auch Javakenntnisse sind im Projektteam gut vorhanden. Java wurde in unserem Projektteam seit dem zweiten Lehrjahr verwendet. Jedoch wurde Springboot im Projekt verwendet, welches zwar in Javasyntax geschrieben ist, aber komplexere Komponente mit sich bringt. Bevor mit der Entwicklung in Spring begonnen wurde, hat sich das Entwicklerteam weitergebildet umso die benötigten Kenntnisse zu erlangen.

## 3. Deklaration der Vorarbeiten und fremden Code

Eine Vorgängerversion von unserem Projekt existierte bereits. Jedoch wurde von diesem kein Code übernommen. Die ältere Version war nicht ganz mit dem Standard umgesetzt. Unser Projekt hat sich aber an dem Vorgänger orientiert bezüglich der Funktionen welches die Applikation haben soll.

## 4. Deklaration der verwendeten Firmenstandards

Unsere Applikation wurde für die SBB entwickelt. Die SBB hat verschiedene Standards die steht’s eingehalten werden müssen. Das Entwicklerteam musste sich exakt an diese halten, um die Applikation intern einführen zu können.

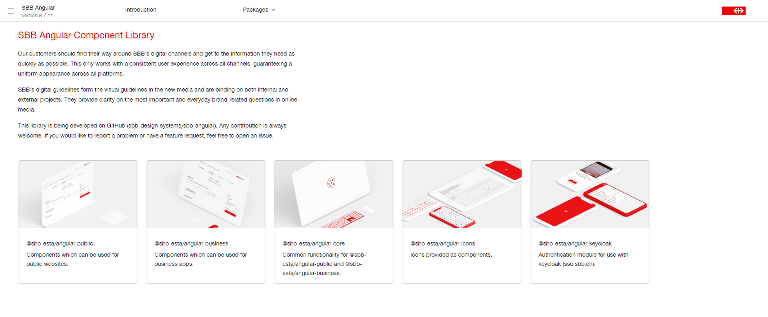
Der Standard schreibt die Technologie vor, die verwendet werden muss. Bei einer Webapplikation müssen spezielle Komponente verwendet werden, wie zum Beispiel das Logo, welches immer an der gleichen Stelle stationiert ist.

Abbildung 1

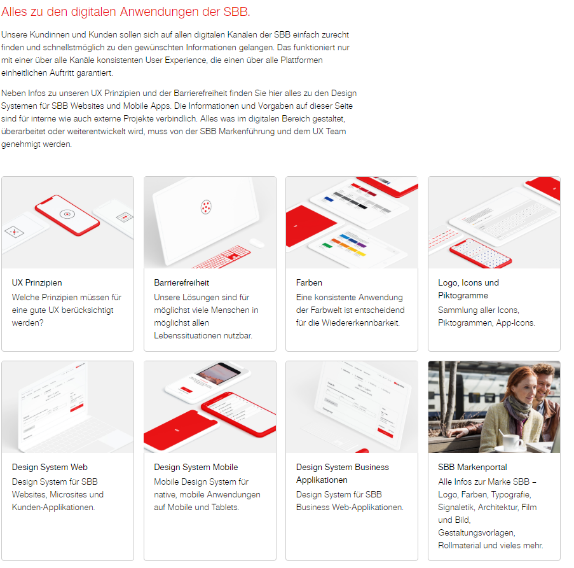


Abbildung 2

## 5. Zeitplan

Der Zeitplan wurde in der Vorbereitungsphase erstellt. Der Zeitplan dient als Grundlage im Zeitmanagement über die ganze Projektzeit. Der Zeitplan ist im Grunde eine Schätzung für die verschiedenen Phasen und wie lange diese dauern. Das Ziel ist, dass das Projekt in der vorgegebenen Zeit umgesetzt wird. In unserem Projektteam hat dies sehr gut funktioniert. Wir waren steht’s in der Zeit und haben auch genug Pufferraum für kleinere Fehler eingeplant.

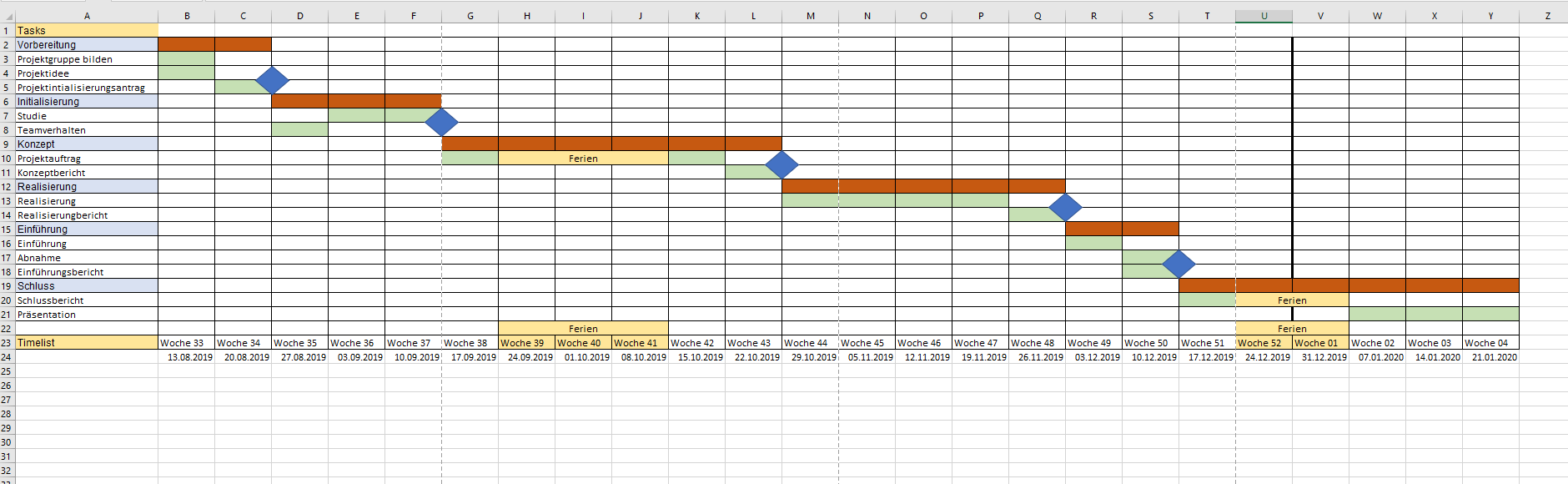


Abbildung 3

### 5.1 Phase Vorbereitung

Die Phase «Vorbereitung» war die erste Phase in unserem Projekt. In dieser wurde ein Projektteam gebildet, sowie eine Idee bestimmt. Zudem wurde ein «Projektinitialisierungsantrag» erstellt. Dieser Antrag dient als Vereinbarung zwischen dem Auftraggeber und dem Projektleiter. Es wird ein kurzer Überblick über das Projekt geboten und wie an dieses herangearbeitet wird. Mit der Bewilligung des Auftraggebers kann in die «Initialisierungsphase» begonnen werden.

### 5.2 Phase Initialisierung

Die «Initialisierungsphase» wurde mit dem OK des Auftraggebers begonnen. In dieser Phase wurde eine Studie über das komplette Projekt erstellt. In dieser Studie wurden verschiedene Punkte wie die Situationslage, Ziele, Lösungsvarianten, Anforderungen etc. aufgelistet. Die Studie dient als Basis für die Projektumsetzung, denn bereits in dieser Phase werden wichtige Dinge komprimiert und so die Projektrealisierung vereinfacht. Zudem wurde das Teamverhalten wie man am besten ein erfolgreiches Team leitet, eingeplant.

### 5.3 Phase Konzept

In der «Konzeptphase» wurde ein Konzeptbericht erstellt. In diesem werden verschiedene Konzepte für Dinge erstellt, die in der Realisierungsphase anstehen werden. In diesem ist zum Beispiel ein «Testkonzept» aufgelistet, indem erklärt wird wie die Applikation mit welchen Kriterien getestet werden soll.

### 5.4 Phase Realisierung

In der Realisierung wird das Projekt umgesetzt basierend auf den vorhandenen Konzepten und Pläne. In dieser Phase wurde unser Projekt umgesetzt. Zudem wurde über diese Umsetzung ein Bericht verfasst, der die Realisierung mitsamt Testing etc. zusammenfasst.

### 5.5 Phase Einführung

Während der Einführungsphase wird das Projekt dem Kunden übergeben. Zu der Einführung wird ein Bericht verfasst, indem beschrieben wird in welchen Schritten die Einführung geschieht und ob eventuell die Benutzer geschult werden müssen etc.

## 6. Arbeitsprotokoll

### 6.1 Arbeitsjournal vom Dienstag 29.10.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkler Olivier, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Anhand des Konzeptes die beste Lösung für unsere Anforderungen finden. | Winkler Olivier | | | 1.5 | | 1.6 | |
| Dokumentieren | Winkler Olivier | | | 0.3 | | 0.2 | |
| Total: |  | | | **3.0** | | **2.9** | |
| Tages Ablauf | |  |  | |  | |
| Heute war der erste Tag in der Realisierungsphase. Zu Beginn des Unterrichts gab uns Herr Beutler eine Präsentation über die Realisierungsphase. Dort hat er uns die verschiedenen Punkte in der Realisierung vorgestellt, die wir beachten sollen, um möglichst effizient diese Phase durchzulaufen.  Danach habe ich mit dem Entwicklungsteam ein Daily gemacht. In diesem wurde der heutige Stand angeschaut und welche Person welche Arbeit als nächstes ausführen wird.  Nach dem Daily konnte ich endlich mit der Realisierung beginnen. Da im Projektteam noch nicht allzu viel Wissen über Spring Boot und Backendentwicklung vorhanden ist, habe ich mich dazu den ganzen Nachmittag informiert. Ich konnte diverse Beispiele finden und habe dies mit dem Team angeschaut. Zum Schluss habe ich noch dieses Tagesjournal geschrieben und angefangen am Realisierungsbericht zu schreiben. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Ich benötigte heute nicht gross Hilfe musste aber bei einigen organisatorischen Dingen auf mein Team zugehen. | | | | | | |
| Reflexion | |  |  | |  | |
| Der heutige Tag war ein guter und effizienter Start in die neue Phase. Wir konnten eine gute Lösung finden und können so das nächste Mal richtig beginnen. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Das nächste Mal werde ich mit dem Implementieren des Backend’s beginnen und mit meinem Team zusammen die Anbindung ans Frontend managen. | | | | | | |

### 6.2 Arbeitsjournal vom Dienstag 05.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkler Olivier, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Implementation Backend | Winkler Olivier | | | 2.5 | | 3 | |
| Dokumentieren | Winkler Olivier | | | 0.3 | | 0.3 | |
| Total: |  | | | **3.0** | | **3.5** | |
| Tages Ablauf | |  |  | |  | |
| Heute war der zweite Tag in der Realisierungsphase. Zu Beginn hielten wir unser Daily und organisierten einige wichtige Dinge. Danach konnte ich mit der Methode, die ich letztes Mal für das Projekt definiert habe, beginnen zu implementieren. Zuerst habe ich das Projekt ordnungsgemäss aufgesetzt. Dabei kam es zu Verzögerungen, da unsere virtuellen Maschinen langsamer liefen als sonst gewohnt.  Nachdem endlich das Projekt korrekt aufgesetzt war, konnte ich mit dem Implementieren beginnen. Heute habe ich einen Controller mitsamt einem Model geschrieben. Zudem habe ich noch die Anbindung an die Datenbank realisiert. Am Ende des Tages hat alles super funktioniert. Zum Schluss habe ich noch dokumentiert. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich nicht gross Hilfe benötigt | | | | | | |
| Reflexion | |  |  | |  | |
| Ich bin heute gut vorangekommen obwohl ich Schwierigkeiten mit der Hardware hatte. Der geschriebene Code bewährt sich jetzt schon bereits. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Das nächste Mal versuche ich das Backend fertig zu implementieren. | | | | | | |

### 6.3 Arbeitsjournal vom Dienstag 12.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkler Olivier, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Implementation Backend | Winkler Olivier | | | 2.5 | | 2.5 | |
| Dokumentieren | Winkler Olivier | | | 0.3 | | 0.3 | |
| Total: |  | | | **3.0** | | **3.0** | |
| Tages Ablauf | |  |  | |  | |
| Heute ist genau die Hälfte der Phase vorbei. Mein Ziel heute war das Backend fertig zu programmieren. Dies ist mir auch gelungen. Zu Beginn des Unterrichts hatten wir unser Daily. Ich konnte die restliche Komponente implementieren und ich habe zum ersten Mal das Backend mit Frontend getestet ob der Datenaustausch funktioniert. Zum Schluss habe ich mich noch den Dokumentationen gewidmet. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt, jedoch habe ich zusammen mit dem Team das Frontend mit dem Backend verbunden. | | | | | | |
| Reflexion | |  |  | |  | |
| Heute war ein erfolgreicher Tag. Ich konnte gut arbeiten und somit das Backend fertig abschliessen | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächste Woche werde ich im Frontend entwickeln, falls man mich dort benötigt. Ansonsten werde ich mich voll dem Realisierungsbericht widmen. | | | | | | |

### 6.4 Arbeitsjournal vom Dienstag 19.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkler Olivier, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Konzept für zukünftiges Deployment | Winkler Olivier | | | 2.0 | | 2.5 | |
| Dokumentieren | Winkler Olivier | | | 0.3 | | 0.2 | |
| Total: |  | | | **3.0** | | **3.4** | |
| Tages Ablauf | |  |  | |  | |
| Heute hat der Endspurt in dieser Phase begonnen. Bereits nächste Woche wird der Bericht abgeben und die Einführungsphase bestritten. Heute habe ich mich dem Konzept gewidmet ob wir noch ein Deployment für unsere Applikation durchführen wollen. Ich habe mich über verschiedene Dienste informiert. Das Hauptziel war aber der Preis. Das Ganze sollte gratis sein. Ich habe bin dann auf Firebase gestossen. Ich habe das noch mit meinem Team besprochen ob wir dies umsetzen, wenn noch Zeit übrigbleibt. Zum Schluss habe ich noch dokumentiert. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt. | | | | | | |
| Reflexion | |  |  | |  | |
| Ich bin heute gut mit der Dokumentation & dem Konzept für das Deployment vorangekommen. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächste Woche ist der letzte Tag in dieser Phase. An diesem Tag wird die Applikation getestet und der Realisierungsbericht fertig gestellt. | | | | | | |

### 6.5 Arbeitsjournal vom Dienstag 26.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkler Olivier, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Testen unserer Applikation | Projektteam SRBS | | | 2.0 | | 2.0 | |
| Dokumentieren | Winkler Olivier | | | 0.3 | | 0.3 | |
| Total: |  | | | **3.0** | | **3.0** | |
| Tages Ablauf | |  |  | |  | |
| Heute war der letzte Tag in dieser Phase. Heute wurde unsere Applikation auf ihre Funktionalität getestet. Zuerst gab es aber eine Präsentation der Lehrperson über die kommende Einführungs- & Schlussphase. Danach habe ich mit dem Enwicklungsteam unsere Applikation getestet und noch kleine Verbesserungen vorgenommen. Zum Schluss habe ich noch den Realisierungsbericht fertiggestellt. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt. | | | | | | |
| Reflexion | |  |  | |  | |
| Heute konnten wir gut die Realisierungsphase abschliessen. Das Testing hat im Großen und Ganzen sauber funktioniert. Zudem sind wir bereits mit dem Realisierungsbericht fertig und konnten diesen abgeben. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächste Woche werden wir mit der Einführungsphase beginnen & eventuell noch kleine Anpassungen an den Applikationen vornehmen. | | | | | | |

### 6.6 Arbeitsjournal vom Dienstag 29.10.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkelmann Domenico, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Frontend aufsetzen & Tagesjournal | Domenico Winkelmann | | | 2 | | 2 | |
| Total: |  | | | **3.0** | | **2.7** | |
| Tagesablauf | |  |  | |  | |
| Heute war der erste Tag in der Realisierungsphase. Als erstes gab uns Herr Beutler eine Präsentation über die Realisierungsphase, in welcher er uns die verschiedenen Punkte in der Realisierung vorstellte, welche wir beachten sollen, um möglichst effizient diese Phase durchzulaufen.  Danach haben wir in unserem Team ein Daily gemacht. In diesem wurde der heutige Stand angeschaut und die Aufgaben unter den Teammitgliedern aufgeteilt.  Als nächstes fing ich an das Frontend aufzusetzen, dazu richtete ich das Git-Repository ein, und erstellte ein neues leeres Projekt und machte den «initial commit». Zum Schluss habe ich noch dieses Tagesjournal geschrieben und angefangen am Realisierungsbericht zu schreiben. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Ich benötigte heute keine Hilfe. | | | | | | |
| Reflexion | |  |  | |  | |
| Der heutige Tag war ein guter und effizienter Start in die neue Phase. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Als nächstes werden wir die Verbindung zwischen Frontend und Backend besprechen. Zudem werde ich anfangen mit der Erstellung des Frontends nach Mockups. | | | | | | |

### 6.7 Arbeitsjournal vom Dienstag 05.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkelmann Domenico, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Implementierung des Frontends | Domenico Winkelmann | | | 2.8 | | 2.8 | |
| Total: |  | | | **3.0** | | **3.0** | |
| Tagesablauf | |  |  | |  | |
| Heute war der zweite Tag in der Realisierungsphase. Als erstes hielten wir wieder unser Daily und organisierten einige Dinge.  Danach fuhr ich fort mit der Implementierung des Frontends, da ich schon letztes Mal alles aufsetzte, konnte ich heute schon beginnen mit dem Erstellen der ersten Views. Dazu erstellte ich als erstes alle Components und fügte noch packages ein, wie dieses von SBB-Angular, von welchem wir dann verschiedene Components und Icons nutzen können. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Ich benötigte heute keine Hilfe. | | | | | | |
| Reflexion | |  |  | |  | |
| Heute bin ich ziemlich gut vorangekommen, hatte anfangs jedoch einige Fehler mit den neuen Components von SBB-Angular, diese konnte ich jedoch nach einiger Zeit beheben. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächstes Mal werde ich die Oberfläche (ohne Zusatzfunktionen wie Sortieren usw.) fertigstellen. | | | | | | |

### 6.8 Arbeitsjournal vom Dienstag 12.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkelmann Domenico, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Daily Scrum Meeting & organisatorische Dinge | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Frontend Implementation | Domenico Winkelmann | | | 2.8 | | 2.8 | |
| Total: |  | | | **3.0** | | **3.0** | |
| Tagesablauf | |  |  | |  | |
| Heute machten wir wieder als Erstes unser Daily-Meeting. Als nächstes arbeitete ich weiter am Frontend. Ich konnte alle Funktionen fertigstellen, welche mit dem Backend kommunizierten. Das heisst, alle Funktionen, wie das Holen, Erstellen und Updaten funktionieren bereits. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt. | | | | | | |
| Reflexion | |  |  | |  | |
| Heute kamen wir sehr gut voran, unter anderem konnten alle Funktionen im auch im Backend implementiert werden. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächstes Mal werde ich noch Funktionen wie Sortieren der Bookings und die Validierung beim Erstellen und Updaten einbauen. | | | | | | |

### 6.9 Arbeitsjournal vom Dienstag 19.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkelmann Domenico, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Implementierung Frontend |  | | | 2.5 | | 2.5 | |
| Total: |  | | | **3.0** | | **3.2** | |
| Tages Ablauf | |  |  | |  | |
| Heute hat der Endspurt in dieser Phase begonnen. Das heisst heute baute ich noch kleinere Funktionen wie Sortieren bei der Booking-Tabelle und Validierung beim Erstellen und Updaten der Bookings. Für dies schrieb ich Mthoden damit die beiden Daten nicht kleiner als das heutige Datum sind usw., zudem baute ich auch eine Regex für die eNummer ein. Auch räumte ich noch den Code noch ein wenig auf und löschte unnötiges raus. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt. | | | | | | |
| Reflexion | |  |  | |  | |
| Ich bin heute gut vorangekommen. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächste Woche wird getestet und noch allfällige Bugs gefixt. | | | | | | |

### 6.10 Arbeitsjournal vom Dienstag 26.11.2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Winkelmann Domenico, INF2017.J***    ***Tätigkeiten*** | ***Beteiligte Personen*** | | | ***Aufwand***  ***geplant (Std)*** | | ***Aufwand***  ***effektiv (Std)*** | |
| Einführung Präsentation | Ganze Klasse mit Lehrperson | | | 0.5 | | 0.5 | |
| Daily Scrum Meeting | Projektteam SRBS | | | 0.2 | | 0.2 | |
| Testen unserer Applikation | Projektteam SRBS | | | 2.0 | | 2.0 | |
| Bugfixing | Winkelmann Domenico | | | 1 | | 1 | |
| Total: |  | | | **3.0** | | **3.7** | |
| Tages Ablauf | |  |  | |  | |
| Heute war der letzte Tag in dieser Phase. Als erstes gab es eine Präsentation der Lehrperson über die kommende Einführungs- & Schlussphase.  Als nächstes testeten wir mit dem Enwicklungsteam unsere Applikation. Dabei tauchten noch einige Bugs auf, welche ich dann fixte. Zudem dokumentierte ich noch ein wenig mehr meinen Code. | | | | | | |
| Hilfestellungen | |  |  | |  | |
| Heute habe ich keine Hilfe benötigt. | | | | | | |
| Reflexion | |  |  | |  | |
| Heute konnten wir erfolgreich die Realisierungsphase abschliessen, was sehr erfreulich war. | | | | | | |
| Nächste Schritte | |  |  | |  | |
| Nächste Woche werden wir mit der Einführungsphase beginnen & eventuell noch kleine Anpassungen an den Applikationen vornehmen. | | | | | | |

# Teil 2

## 7 Situationsanalyse

### 7.1 Ausgangslage

Die Sitzungszimmer in SBB-Gebäuden werden via Outlook gebucht. Vor Ort kann der Raum zusätzlich über einen kleinen Bildschirm besetzt werden. Dieser Bildschirm ist mit einem GUI ausgestattet. Dieses GUI macht zurzeit immer wieder Probleme bei den Benutzern. Die Mitarbeitende haben Mühe mit diesem umzugehen. Das GUI ist nicht benutzerfreundlich gestaltet. Das Entwicklungsteam besteht aus 3. Lehrjahr Lernende in der Informatik Fachrichtung Applikationsentwicklung.

### 7.2 Stärken

* Kenntnisse mit der Programmiersprache *Angular*
* Kenntnisse in dieser Umgebung (Web)
* Wissen von zwei Lehrjahren in Applikationsentwicklung

### 7.3 Schwächen

* Kenntnisse bezüglich Security schlecht vorhanden
* Kenntnisse im Backend / Datenbank schwacher vorhanden als Frontend

## 8. Systemziele

### 8.1 Rahmenbedingungen

Die Vorarbeiten sind erledigt, der Projektinitialisierungsauftrag ist erteilt. Fr 23.08.19 (KW34)

Die Phase «Initialisierung» ist am Dienstag 10.09.2019 abgeschlossen (Phase Initialisierung ist abgeschlossen, die Studie ist abgeschlossen, der Projektauftrag ist erteilt, die Zeitplanung ist aufgegleist, das Dokument Projektführung ist eröffnet.) Diese Phase ist spätestens am

Fr 13.09.19 (KW37) abgegeben.

Die Phase «Konzept» ist am Dienstag 22.10.2019 abgeschlossen & ist spätestens am Fr 25.10. 2019 (KW43) abgegeben.

Die Phase «Realisierung» ist am Dienstag 26.11. 2019 abgeschlossen & ist spätestens am Fr 29.11. 2019 (KW49) abgegeben.

Die Phase «Einführung» ist am Dienstag 10.12.2019 abgeschlossen & ist spätestens am Fr 13.12. 2019 (KW51) abgegeben.

Schlussbericht und Präsentation sind am 07.01.2020 fertiggestellt

Jede Woche haben wir am Dienstagnachmittag 3 Stunden in der Schule für unser Projekt Zeit. Der Unterricht findet im IET Gebäude im Raum IE-203 statt.

### 8.2 Abgrenzung

* Ausfall des Programmes durch Netzwerkprobleme
* Ausfall durch höhere Gewalt (Naturkatastrophe wie erbeben, Überschwemmungen, usw.…)
* Ausfall durch Gewalteinwirkung / Manipulation

## 9 Lösungsvorschläge

### 9.1 Variantenübersicht

**Variante 1**

Das Projekt wird nach SBB-Standards mit Angular (Frontend) und Spring Boot (Backend) realisiert.

Es werden zwei Views erstellt, auf der einen werden alle Buchungen angezeigt und auf der Zweiten kann eine neue Buchung erstellt werden.

**Variante 2**

Das Projekt wird nicht nach SBB-Standard realisiert, sondern mit Angular(Frontend) und NodeJS(Backend).

Es wird eine zusätzliche View erstellt, auf welcher die detaillierte Ansicht einer Buchung angezeigt wird.

**Variante 3**

Das Projekt wird nicht nach SBB-Standard realisiert, sondern mit ReactJS (Frontend) und Spring Boot(Backend).

Es wird nur eine View erstellt, um eine Buchung zu erstellen, wird eine Lightbox erstellt.

### 9.2 Beschreibung der Varianten

**Variante 1**

Die erste Variante wird nach SBB-Standard realisiert. Das Frontend wird mit Angular entwickelt und das Backend mit Spring Boot. Diese Variante könnte theoretisch ohne Probleme bei der SBB eingesetzt werden, sollte dies zustande kommen. Da diese Variante den Standard einhält, schneidet diese am besten in der Tabelle ab, da die benötigten Ressourcen an Hardware & Knowhow vorhanden sind. Die Umsetzung mit diesen beiden Technologien ereignet sich als die beste Möglichkeit, da das Knowhow hier am höchsten ist und so die Realisierungsdauer und Fehlerquote tief gehalten werden kann.

**Variante 2**

Die zweite Variante wird nicht ganz mit SBB-Standard realisiert. Auch hier wird Angular als Frontend verwendet. Als Backend wird aber NodeJS verwendet, welches nicht ganz dem Standard entspricht. Es wäre aber nicht sicher ob diese Architekturen bei der SBB verwendet werden dürften. Da der Standard nicht eingehalten wird, sind auch die verschiedenen Punkte in der untenstehenden Tabelle schlechter bewertet. Bezüglich des Knowhows wäre diese Variante auch im Nachteil. Das Entwicklerteam hat noch kein richtiges Projekt mit NodeJS umgesetzt, jedoch schon einige kleinere Applikationen mit dieser Architektur entwickelt. Dies würde heissen, dass die Vorbereitungszeit länger andauern würde und es eventuell in der Realisierung zu Schwierigkeiten kommen würde.

**Variante 3**

Die dritte Variante variiert stark vom SBB-Standard ab. Das Frontend wird hier nicht mehr in Angular umgesetzt, sondern in ReactJS. Als Backend würde Spring Boot zum Einsatz kommen. Da der SBB-Standard nicht beachtet wird, wird es schwierig dieses Projekt intern korrekt einzusetzen. Zudem sind auch die verschiedenen Punkte in der untenstehenden Tabelle schlechter bewertet wegen der Nichteinhaltung des Standards. Auch das Knowhow wäre nicht im Frontend vorhanden und müsste aufgearbeitet werden und dies könnte Verzögerungen in den Planungsphasen und der Realisierung verursachen. Dies ist ein sehr grosser Nachteil.

### 9.3 Bewertung der Varianten (Tabelle)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Anforderungen** | | **Variante 1** | | **Variante 2** | | **Variante 3** | |
| **Muss** | | | | | | | |
| SBB-Standard | | Ja | | Nein | | Nein | |
| Verfügbarkeit | | Ja | | Ja | | Ja | |
| Sicherheit | | Ja | | Ja | | Ja | |
| Knowhow | | Ja | | Ja | | Nein | |
| **Kann** | **Gewicht** | **Punkte** | **Total** | **Punkte** | **Total** | **Punkte** | **Total** |
| Schutz/Sicherheit | 20 | 4 | 80 | 3 | 60 | 4 | 80 |
| Benutzerfreundlichkeit | 30 | 5 | 150 | 4 | 120 | 5 | 150 |
| Anzahl benötigte Ressourcen | 20 | 4 | 80 | 3 | 60 | 3 | 60 |
| Verwaltung | 30 | 4 | 120 | 3 | 90 | 2 | 60 |
| **Total** | 100 |  | 430 |  | 330 |  | 350 |

**1:** nicht erfüllt **2:** knapp nicht erfüllt **3:** ausreichend **4:** gut **5:** sehr gut

### 9.4 Lösungsbeschreibung

Das Entwicklerteam hat sich für die erste Variante entschieden wegen der höchsten Anzahl von Punkten in der Bewertungstabelle. Mit dieser Umsetzungsvariante ist der SBB-Standard gesichert und kann so vielfältig in der SBB eingesetzt werden. Zudem ist das benötigte Knowhow vorhanden. Diese Variante wird Vorteile bringen und die Projektumsetzung vereinfachen und im Zeitplan halten. Unser Projekt wird mit folgenden Komponenten realisiert:

* Architektur => Angular (Frontend), Spring Boot (Backend)
* SBB-Addons => SBB-Schriften, Komponente für Standard

**Entwicklungsumgebung**

Das Projekt wird auf der Gibbix entwickelt. Die verwendete Integrierte Entwicklungsumgebung ist Intellij und Visual Studio Code. Als Versionsverwaltung dient der Gitlabserver der Gibb.

**Umsetzung**

Das Frontend wird mit Angular 8.0 realisiert. Zu der Frontendentwicklung hat das Entwicklungsteam Mockups erstellt. Das Frontend wird nach diesen Mockups designt, welche nach SBB-Standard aufgebaut sind. Als Backend wird Spring Boot 2.1.8 verwendet. Im Backend werden die verschiedenen Funktionen, welche benötigt werden um eine neue Buchung hinzuzufügen, löschen etc., programmiert. Zudem wird im Backend die Datenbank angebunden. Mit dem Backend sollte das Frontend mit den Werten aus der DB versorgt werden.

## 10 Systemarchitektur

### 10.1 Systemdesign

Während der Umsetzung von unserem Konzept hat sich unsere Systemarchitektur nicht verändert. Zurzeit besteht immer noch ein Front- sowie Backend mitsamt einer Datenbank für die Behausung von Daten.

#### 10.1.1 Struktur des Systemdesigns

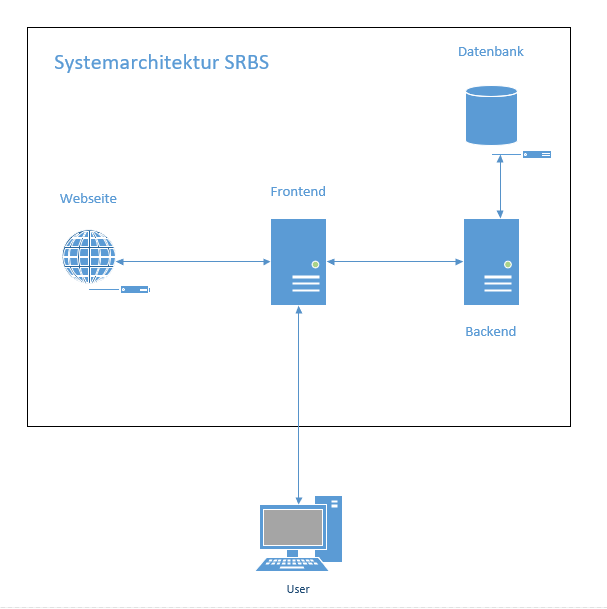


Abbildung 4

#### 10.1.2 Beschreibung der Elemente

**Frontend:**

**Module**: Frontend in Angular (Libraries von SBB werden verwendet)

**Backend & Datenbank:**

**Module**: Backend in Spring Boot, Datenbank in SQL angebunden

### 10.2 Schnittstellendefinitionen

**Externe Schnittstelle**

* **ES1**: Zugriff auf Webseite (Benutzer ruft via URL unsere Webapplikation auf)

**Interne Schnittstelle**

* **IS1**: Frontend enthält benötigte Daten von Backend
* **IS2**: Backend wird von Frontend angesprochen.
* **IS3**: Backend bekommt gespeicherte Buchungen von der Datenbank
* **IS4**: Datenbank bekommt Anfrage für Datentransfer zum Backend
* **IS5**: Webseite wird dem Benutzer angezeigt, dahinter Frontend -> Backend -> Datenbank

### 10.3 Sicherheit

Unsere Applikation verfügt über keine schützenswerten Daten. Da dieser Fall eintrifft hat unsere Applikation keine speziellen Anforderungen an Informationssicherheit & Datenschutz. Unsere Applikation ist im internen Firmennetz verfügbar und enthält Informationen über die Buchung von XY, Zeit der Buchung und die Person, die den Raum gebucht hat.

## 11. Testkonzept und Testspezifikationen

### 11.1 Systemtest - Testspezifikations

#### 11.1.1 Kritikalität der Funktionseinheit

Für jeden Testfall ist eine Kritikalität definiert. Falls einer der Testfälle nicht erfolgreich durchgeführt werden kann, wird anhand von der Kritikalität entschieden ob dies zulässig ist oder das Produkt vor der Abgabe noch verbessert werden muss.

#### 11.1.2 Testanforderungen

Die Tests werden während der normalen Bedingungen getestet. Zudem wird geschaut was passiert, wenn verschiedene Komponente nicht richtig funktionieren z.B. das Backend nicht responsfähig ist.

#### 11.1.3 Testverfahren

Die Tests werden in drei Abschnitten aufgebaut. Zuerst wird ein Test die «Vorbereitung» durchlaufen. Gefolgt von der «Durchführung» und der «Auswertung».

#### 11.2.4 Testkriterien

**Abdeckungsgrad:**

Die Testfälle sind darauf abgestimmt alle Anforderungen auf ihre Funktionalität zu testen.

**Checklisten**:  
Die dazugehörigen Checklisten befinden sich direkt in den Testfällen falls benötigt.

**Ende-Kriterien**:  
Der Testfall gilt als abgenommen, wenn alle Kriterien erfolgreich bestanden worden sind.

#### 11.2.5 Testfälle

|  |  |
| --- | --- |
| **Nr.** | **1** |
| Kritikalität | Hoch |
| Anwendungsfall | Backend bekommt benötigte Daten von Datenbank |
| Ausgangssituation | Datenanforderung (Eine Buchung anzeigen) von Backend an Datenbank mittels GET-Request |
| Erwartetes Ergebnis | Die Datenbank gibt alle gespeicherten Daten zurück, welche durch das Backend angefordert wurden. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

|  |  |
| --- | --- |
| **Nr.** | **2** |
| Kritikalität | Hoch |
| Anwendungsfall | Frontend bekommt benötigte Daten von Backend |
| Ausgangssituation | Datenanforderung (Eine Buchung anzeigen) von Frontend an Backend mittels GET-Request via URL |
| Erwartetes Ergebnis | Das Backend gibt die benötigten Daten ans Frontend, welche zuvor vom Backend aus der Datenbank gelesen werden. Das Frontend zeigt anschliessend die gespeicherten Daten dem Benutzer an. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

|  |  |
| --- | --- |
| **Nr.** | **3** |
| Kritikalität | Hoch |
| Anwendungsfall | Frontend (Webseite) wird dem Benutzer korrekt angezeigt via Localhost (127.0.0.1) |
| Ausgangssituation | Der Benutzer ruft das Frontend resp. die Webseite mit einer URL auf und diese wird ihm dann vorgeführt. |
| Erwartetes Ergebnis | Der Benutzer kann die Webseite einsehen und mit dieser interagieren. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

|  |  |
| --- | --- |
| **Nr.** | **4** |
| Kritikalität | Hoch |
| Anwendungsfall | Benutzer erstellt eine neue Buchung |
| Ausgangssituation | Der Benutzer erstellt im GUI eine neue Buchung und diese soll korrekt hinterlegt werden. |
| Erwartetes Ergebnis | Der Benutzer erstellt seine Buchung mit allen benötigten Daten. Diese werden vom Frontend korrekt verarbeitet und ans Backend weitergeleitet. Dieses sendet die Daten an die Datenbank und werden dort hinterlegt. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

|  |  |
| --- | --- |
| **Nr.** | **5** |
| Kritikalität | Hoch |
| Anwendungsfall | Benutzer bearbeitet eine Buchung |
| Ausgangssituation | Der Benutzer editiert eine Buchung im GUI und die geänderten Daten werden gespeichert. |
| Erwartetes Ergebnis | Der Benutzer kann eine Buchung im GUI bearbeiten. Er kann alle Daten bearbeiten und diese werden anschliessend korrekt in der Datenbank gespeichert und somit ist die Buchung mit den Änderungen aktualisiert worden. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

|  |  |
| --- | --- |
| **Nr.** | **6** |
| Kritikalität | Hoch |
| Anwendungsfall | Benutzer löscht eine Buchung |
| Ausgangssituation | Der Benutzer löscht eine Buchung im GUI. |
| Erwartetes Ergebnis | Der Benutzer kann eine Buchung löschen. Diese wird dann auch in der Datenbank entfernt und somit auch nicht mehr angezeigt. |
| Benötigte Tools/Apps | * Frontend * Backend (+ Datenbank) |

### 11.3 Testprozedur

Die Tests sind der Reihe nach von Nummer eins bis sechs abzuarbeiten. Funktioniert der erste Testfall nicht, funktionieren auch die nachfolgenden Testfälle nicht korrekt.

|  |  |
| --- | --- |
| **Nr.** | **Ausgangszustand der Testfälle** |
| 1 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |
| 2 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |
| 3 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |
| 4 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |
| 5 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |
| 6 | Front- & Backend sind gestartet. Chrome ist installiert für optimale Anzeige |

**Vorbereitung**

**Voraussetzungen:**

Für die Tests muss das Frontend sowie das Backend gestartet werden. Ohne Backend kann das Frontend keine Daten anzeigen und das Backend ist ohne Frontend benutzerunfreundlich. Dies ist der einzig benötigte Schritt in der Vorbereitung um die Tests durchzuführen.

**Konfiguration**:

Die Konfiguration sollte so sein, wie sie in diesem Dokument konfiguriert wurde. Bei Abweichungen kann es zu Fehlern und Anzeigebugs kommen, da wichtige Ressourcen fehlen.

**Durchführung**:

|  |  |
| --- | --- |
| **Nr.** | **Interaktion der Testfälle** |
| 1 – 6 | Auf dem Testgerät werden beide Applikationen (Front- & Backend) gestartet. Der Benutzer besucht die Webseite, welche Buchungen zeigt.  Benutzer sieht alle Buchungen. Front- & Backend zeigen alles korrekt an. |

**Nachbearbeitung**:

|  |  |
| --- | --- |
| **Nr.** | **Nummer des Testes** |
| Erwartet Resultat | Beschreibung des Resultates, dass vor dem Test erwartet wurde. |
| Tatsächliches Resultat | Beschreibung des Resultates, dass beim Test herausgekommen ist. |
| Folgen | Beschreibung der Folgen, die entstanden sind bei Abweichungen. |

## 12. Einführungskonzept

### 12.1 Einführungsplan

Unsere Applikation wurde so gebaut, dass sie in verschiedenen Bereichen eingesetzt werden kann. Das eigentliche Ziel der Applikation wäre es produktiv in einer Firma zu laufen.

Bei der Umstellung auf das neue Produkt ist eine Migration erforderlich. Unsere Applikation soll mit den bereits in einer Datenbank vorhanden Daten bestückt werden. Somit kann ein grosser Aufwand vermieden werden. Die Daten sind alle buchbaren Sitzungszimmer in einer beliebigen Firma, in unserem Fall die SBB. Die neue Applikation kann so nur an die Datenbank angehängt werden und schon sind alle Räume wieder verfügbar.

Zudem soll die Applikation sicher während Bürozeiten einwandfrei laufen. Die Mitarbeiter sind von unserer Applikation abhängig, da Räume von mehreren Mitarbeitern pro Tag gebucht werden können.

Um einen möglichst ruhigen Übergang in die neue Applikation erfolgt, wird es in verschiedenen Stufen eingeführt. Zuerst würde eine Art «Pilotenphase» durchgeführt werden. In dieser wird die Applikation an einem Testsystem nochmals ordentlich getestet. Zudem können in dieser Phase noch Änderungen durchgeführt werden, die durch den Kunden noch verlangt werden. In der zweiten Phase werden die Mitarbeiter, die zukünftig das Produkt verwalten werden, geschult. In der letzten Phase wird die Applikation in der Firma für alle freigegeben. Das Produkt läuft somit produktiv in einer Firma.

Während den Einführungsphasen bestehen keine grösseren Risiken. Jedoch könnte es Probleme mit der Datenmigration geben. Zudem kommen Risiken dazu, wenn die Applikation produktiv läuft z.B. Totalausfall was zu einem Chaos bei den Buchungen von Räumen führen könnte.

Pilotenphase

**1**

Testphase

**2**

Einführungsphase

**3**

Abbildung 5

### 12.2 Migrationsplan

Die Datenmigration ist nicht unbedingt nötig bei unserer Applikation. Bei unserer Applikation kann man die Daten selbst erstellen oder diese aus einer bestehenden Datenbank laden. Bei einem grossen Unternehmen wie der SBB macht es Sinn die bereits vorhandenen Daten ins neue System zu migrieren. Die Migration findet nach der Inbetriebnahme des Produktes statt.

### 12.3 Ausbildungsplan

Für die Bedienung von unserem Produkt werden die Stakeholders in zwei Gruppen unterteilt. Auf der einten Seite gibt es die Anwendergruppe und auf der Anderen die Support- / Admingruppe.

**Anwendergruppe**

Für die Anwender ist keine grosse Schulung nötig. Dass durch unser Produkt ausgemusterte System ist von den Bedienbarkeiten nicht allzu anders. Unser Produkt ist selbsterklärend und hat eine gute Validierung, damit der Benutzer nur verwendbare Werte eingeben kann und wird bei Fehlern durch eine Meldung darauf hingewiesen.

**Supportgruppe**

Bei den Supporten oder Administratoren ist jedoch eine Schulung auf das neue System erforderlich.

Für die Schulung wird ein Vormittag vorgesehen. Während der Schulung werden den Personen gezeigt wie das Programm im Hintergrund aufgebaut ist, wie man Support leisten kann bei Einzelfällen oder sogar Teilausfällen und wie das Programm für die Weiterentwicklung vorbereitet ist.

**Aufwand in Stunden**

Anwendergruppe

Supportgruppe

Abbildung 6

3 Stunden

0 Stunden

## 13. Testprotokoll

### 13.1 Testobjekt

|  |  |
| --- | --- |
| **Nr.** | **1** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:00 Uhr |

|  |  |
| --- | --- |
| **Nr.** | **2** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:10 Uhr |

|  |  |
| --- | --- |
| **Nr.** | **3** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:20 Uhr |

|  |  |
| --- | --- |
| **Nr.** | **4** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:30 Uhr |

|  |  |
| --- | --- |
| **Nr.** | **5** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:40 Uhr |

|  |  |
| --- | --- |
| **Nr.** | **6** |
| Testobjekte | Front- & Backend inklusive Datenbank |
| Tester | Winkler Olivier & Winkelmann Domenico |
| Ort | Zimmer IET203 GIBB Bern |
| Datum | 26.11.2019 um 14:50 Uhr |

**Versionen der Testkomponenten**

|  |  |
| --- | --- |
| **Komponente** | **Version** |
| Frontend | Angular 8 |
| Backend | Spring Boot 2.1 |
| Datenbank | Hibernate 5.3.10.Final |

### 13.2 Testresultate

|  |  |
| --- | --- |
| **Nr.** | **1** |
| Erwartet Resultat | Die Datenbank gibt alle gespeicherten Daten zurück, welche durch das Backend angefordert wurden. |
| Tatsächliches Resultat | Das Frontend hat die Daten aus der Datenbank angezeigt. Das heisst, dass die Daten vom Backend gelesen werden konnten und anschliessend erfolgreich ans Frontend weitergeleitet wurden. |
| Folgen | Keine |

|  |  |
| --- | --- |
| **Nr.** | **2** |
| Erwartet Resultat | Das Backend gibt die benötigten Daten ans Frontend, welche zuvor vom Backend aus der Datenbank gelesen werden. Das Frontend zeigt anschliessend die gespeicherten Daten dem Benutzer an. |
| Tatsächliches Resultat | Der Tester konnte alle Daten auf der Startseite einsehen. Mit diesem Resultat lässt sich daraus ziehen, dass alle Daten korrekt verarbeitet wurden. |
| Folgen | Keine |

|  |  |
| --- | --- |
| **Nr.** | **3** |
| Erwartet Resultat | Der Benutzer kann die Webseite einsehen und mit dieser interagieren. |
| Tatsächliches Resultat | Die Webseite wurde korrekt angezeigt. Zu Testzwecken wurde Google Chrome verwendet, da in diesem Browser die Anzeige perfekt garantiert ist. Die Webseite war in allen Punkten voll funktionsfähig. |
| Folgen | Keine |

|  |  |
| --- | --- |
| **Nr.** | **4** |
| Erwartet Resultat | Der Benutzer erstellt seine Buchung mit allen benötigten Daten. Diese werden vom Frontend korrekt verarbeitet und ans Backend weitergeleitet. Dieses sendet die Daten an die Datenbank und werden dort hinterlegt. |
| Tatsächliches Resultat | Auf der Webseite liess sich eine neue Buchung hinzufügen. Diese wurde dann korrekt bei der Übersicht angezeigt, sowie in der Datenbank. |
| Folgen | Keine |

|  |  |
| --- | --- |
| **Nr.** | **5** |
| Erwartet Resultat | Der Benutzer kann eine Buchung im GUI bearbeiten. Er kann alle Daten bearbeiten und diese werden anschliessend korrekt in der Datenbank gespeichert und somit ist die Buchung mit den Änderungen aktualisiert worden. |
| Tatsächliches Resultat | Auf der Übersicht wurde eine spezifische Buchung angeklickt. Somit kommt man in die Bearbeitungsansicht. In dieser View wurden alle Daten korrekt angezeigt und konnten individuell angepasst werden und schlussendlich gespeichert werden. |
| Folgen | Keine |

|  |  |
| --- | --- |
| **Nr.** | **6** |
| Erwartet Resultat | Der Benutzer kann eine Buchung löschen. Diese wird dann auch in der Datenbank entfernt und somit auch nicht mehr angezeigt. |
| Tatsächliches Resultat | Auf der Übersicht wurde bei einer Buchung auf das Papierkorbsymbol geklickt. Nach der Nachfrage ob der Benutzer wirklich die Buchung löschen möchte, wurde die Buchung ordnungsgemäss gelöscht. |
| Folgen | Keine |

### 13.3 Testauswertung

Die Testfälle sind mit der vollen Punktzahl durchgekommen. Es gab während des Testing’s keinerlei Abweichungen von unseren Definitionen. Die Perfektion während der Entwicklungsphase, die durch unser Projektteam vorgenommen wurde, zahlte sich aus, da die Applikation dank diesem Aufwand sorgenfrei läuft.

### 13.4 Abnahme

|  |  |
| --- | --- |
| **Testdatum** | 11.12.2019 |
| **Tester** | Blaser Sabine |
| **Gesamttestresultat** | 🞎 Abgenommen  🞎 Abgenommen mit Nacharbeiten  🞎 Nicht abgenommen |
| **Nacharbeiten** | Zeichenbegrenzung beim Feld, bei dem man den Raum eingeben kann. Ansonsten kann eine unendlich lange Raumbeschreibung eingeben werden.  Beim Datumfeld soll, wenn nichts angegeben wird, eine Fehlermeldung erscheinen.  Fehlermeldungen sollen erscheinen falls eine Buchung nicht gespeichert, resp. bearbeitet werden kann. |
| **Unterschrift Lieferant** |  |
| **Unterschrift Kunde** |  |

## 14. Benutzerdokumentation / Anleitungen

### 14.1 Konfigurations-Dokumentation

In diesem Abschnitt wird beschrieben, wie unsere Applikation auf- und umgesetzt wurde. Zuerst werden die verschiedenen benötigten Komponenten beschrieben und anschliessend die Umsetzung. Unsere Applikation wurde in Angular 8 und Spring Boot 2 geschrieben.

**Installation Angular 8**

Um ein Angularprojekt zu starten, wird dies zuerst installiert. Zudem installieren wird Komponente aus der Public-SBB-Library. Diese verwenden wird um einerseits den Standard einzuhalten und können so vorgefertigte Komponente in unsere Applikation einbauen. Dazu kommen die Icons, die standardmässig auf SBB-Webseiten verwendet werden.



Abbildung 7

Installation:

Installation von ersten verwendeten Package:



Abbildung 8

Installation von zweiten verwendeten Package:



Abbildung 9

Installation Spring Boot 2

Ein Spring Boot Projekt kann mit einem Initializer ganz einfach erstellt werden. Die verschiedenen Properties werden dort angegeben und anhand von diesen wird ein Projekt erstellt. Mit diesem Tool kann sehr schnell ein solches Projekt initialisiert werden.

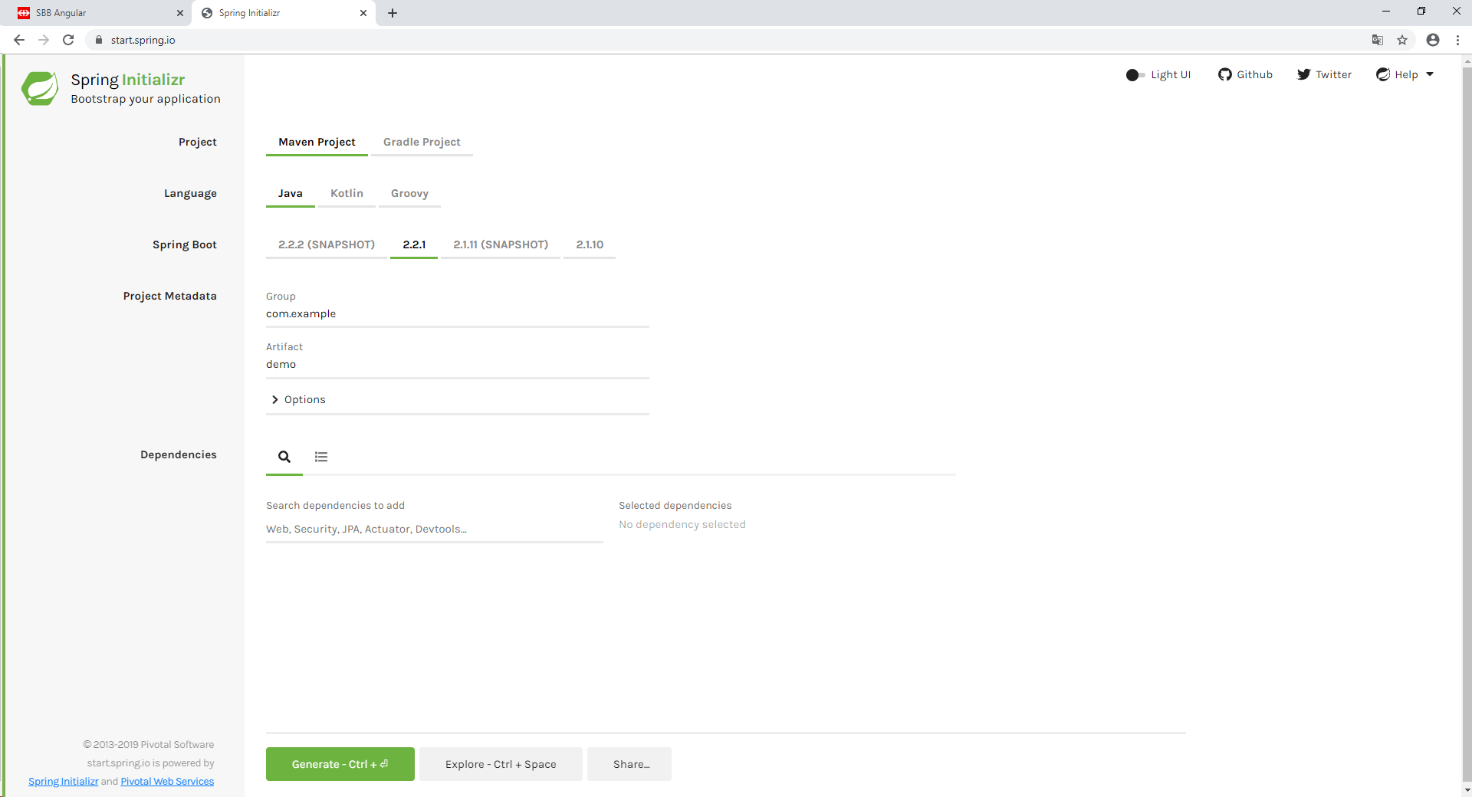


Abbildung 10

**Umsetzung Frontend**

Im Frontend wurden die Komponenten aus den oben installierten Libraries verwendet. Hier werden nun einige aufgelistet.

Kalender um ein Datum und Zeit auszuwählen. Wird für die Start- & Endzeit einer Buchung verwendet.



Abbildung 11

Tabelle um Daten anzuzeigen. Diese wird auf der Startseite verwendet, auf der der Benutzer alle Buchung einsehen kann.



Abbildung 12

Button. Dieser wird mehrfach verwendet. Der Vorteil von diesem Button ist, dass er vorgestylt ist und man Icons in seinen Body anheften kann.

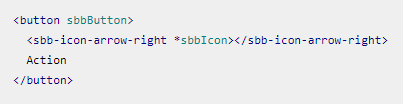


Abbildung 13

Dies ist der Header. Dieser ist ganz oben anzutreffen. Beinhaltet z.B. das SBB-Logo



Abbildung 14

**Umsetzung Backend**

Das Backend ist in Spring Boot 2 geschrieben. Als Datenbank wird Hibernate verwendet. Hibernate ist ein ORM-Framework, dass von Java verwendet wird. Das Backend ist einem MVC aufgeteilt.

Der Controller verarbeitet die kommenden Anfragen vom Frontend. Diese werden direkt an eine Klasse weitergeleitet, die von JPARepository erbt. Diese Klasse gehört in die Spring DATA Abteilung. Mit dieser Klasse kann Hibernate eine Datenbank erstellen und diese verwalten.

GET-Request um alle Buchungen aus der Datenbank zu holen.

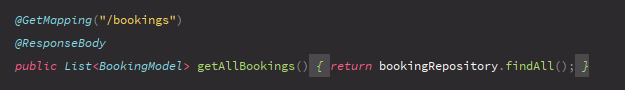


Abbildung 15

GET-Request um eine bestimmte Buchung mittels ID zu finden.

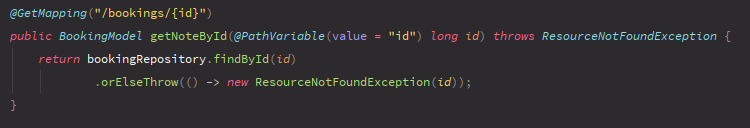


Abbildung 16

1. Methode POST-Request um eine Buchung zu erstellen.
2. Methode PUT-Request um eine Buchung zu aktualisieren.
3. Methode DELETE-Request um eine Buchung zu löschen.

Bei der POST-Methoden werden die Values vom Frontend ans Backend geschickt, welches diese dann richtig verarbeitet und in der Datenbank speichert. Die PUT-Methode funktioniert ähnlich und die DELETE-Methode löscht das Element mit der angegeben ID.

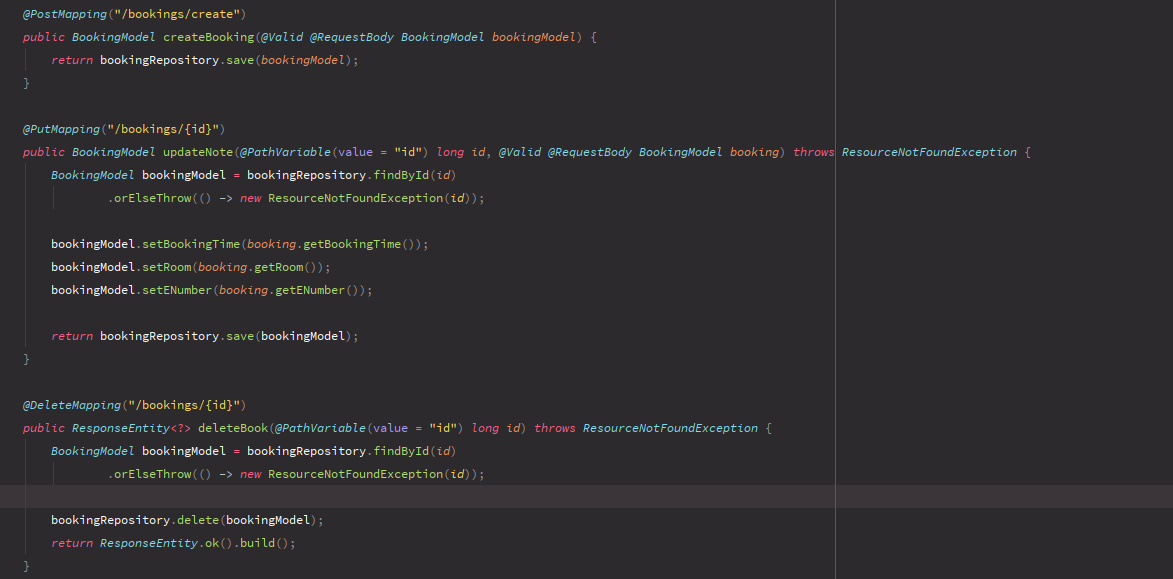


Abbildung 17

Untenstehend ist das Model aufgelistet. Hier sieht man alle Attribute, die eine Buchung hat.

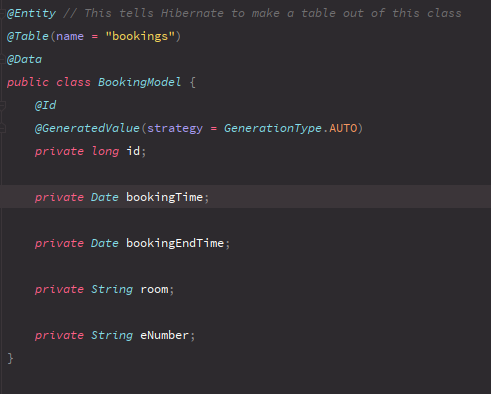


Abbildung 18

### 14.2 Benutzerhandbuch

##### 14.2.1 Systemübersicht

Mit unserer Applikation können Benutzer einen Sitzungsraum buchen. Der Benutzer kann eine Buchung für einen Raum erstellen, bearbeiten und löschen. Die Webseite ist nach SBB-Standard umgesetzt und kann so intern & extern ohne Probleme eingesetzt werden.

#### 14.2.2 Anwenderfunktionalität

**Buchungen ansehen**

Ein Bild, das Screenshot enthält.

Automatisch generierte BeschreibungWenn der Benutzer die Webseite besucht, sieht er alle Buchungen. Bei jeder Buchung ist die jeweilige Anfangszeit der Reservierung sowie das Ende hinterlegt. Zudem wird angezeigt welchen Raum um diese Zeit besetzt ist. Auch sieht man die Benutzernummer, mit der man entnehmen kann, wer die Buchung durchgeführt hat. Neben all diesen Angaben kann man mit dem Stiftsymbol die jeweilige Buchung bearbeiten und mit dem Papierkorb sogar löschen.

Abbildung 19

**Buchung erstellen**

Ein Bild, das Screenshot enthält.

Automatisch generierte BeschreibungWenn der Benutzer eine neue Buchung erfassen möchte, kann er dies indem er auf den roten Button oberhalb der Tabelle klickt, tun. Danach gelingt er in den Erstellungsmodus einer Buchung. Hier kann er die Zeiten, Raumnummer und seine Benutzernummer eingeben. Diese Änderungen werden dann auf der Übersicht angezeigt. Falls der Benutzer die Buchung nicht erstellen möchte, kann er diese verwerfen. Bevor dies aber geschieht wird eine Flashmessage angezeigt, in der der Benutzer gefragt wird, ob er wirklich seine Angaben verlieren möchte.

Abbildung 20

**Buchung bearbeiten**

Wenn der Benutzer auf den Stift klickt, kommt er in den Bearbeitungsmodus. Alle Angaben werden angezeigt und können individuell geändert werden. Die Buchung kann mit dem roten Button gespeichert werden. Falls der Benutzer Änderungen vorgenommen hat, welcher er nicht gespeichert hat, wird eine Flashmessage angezeigt in der gefragt wird ob die Änderungen wirklich, ohne zu speichern verloren gehen sollen.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung**Buchung löschen**

Abbildung 21

Falls der Benutzer eine Buchung löschen möchte, kann er dies tun indem er auf das Papierkorbsymbol klickt. Danach erscheint folgende Flashmessage, die ihn nochmals fragt ob wirklich die Buchung entfernt werden soll.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 22

### 14.3 Supporthandbuch

**Massnahmen bei Benutzerproblemen**

Buchung kann nicht erstellt werden, da das Enddatum kleiner als das Startdatum ist.  
Um dies zu lösen, muss das Enddatum erhöht werden.



Abbildung 23

Buchung kann nicht erstellt werden, da das beide Daten und Zeiten identisch sind.   
Um dies zu lösen, muss die Endzeit oder das Enddatum erhöht werden.

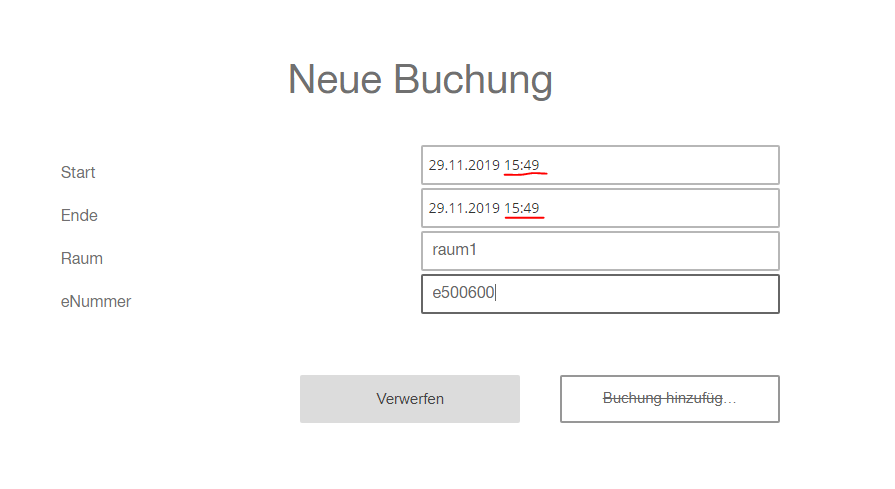


Abbildung 24

**Massnahmen bei technischen Problemen**

Backend -Verbindung schlägt fehl. Der Benutzer kann nun den Support kontaktieren.

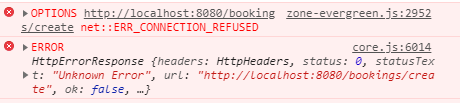


Abbildung 25

## 15. Projekterfahrung

### 15.1 Erfahrung Winkler Olivier

Ich konnte während dem Projekt sehr viel Erfahrungen sammeln. Vor diesem Modul habe ich noch nie mit Hermes praktisch gearbeitet. Für die kommende IPA nächstes Jahr ist dieses Modul eine sehr gute Grundlage. Unser Projektteam besteht aus alten Artgenossen. Wir sind ein eingespieltes Team und haben eine sehr gute Chemie zusammen. Dies ist sicherlich ein Vorteil bei einer grossen Projektarbeit, denn man weiss genau wer welche Aufgabe am besten übernehmen könnte. Für das Projekt haben wir extra etwas genommen, welches wir im Betrieb brauchen und wir beide noch nicht allzu lange darauf gearbeitet haben. Mit Spring Boot habe ich noch nicht so viel gemacht und dies hat mich noch mehr gereizt diese Technologie hier zu verwenden. Die Dokumentation während dem Projekt ging vor allem auf meine Schulter. Im Projektteam wurde aber dies auch so abgemacht. Ich bin zufrieden mit unserem Resultat und um neues zusätzliches Wissen reicher.

### 15.2 Erfahrung Winkelmann Domenico

Ich empfand dieses Projekt als gute Grundlage für die kommende IPA von nächstem Jahr. Unter anderem, da ich noch nie wirklich mit Hermes gearbeitet habe.  
Auch der Faktor dass wir beide in der gleichen Firma arbeiten und in verschiedenen Bereichen Erfahrung haben, konnten wir uns perfekt ergänzen.  
Zudem konnten wir uns in den jeweiligen Bereichen noch vertiefen, unter anderem auch durch die Einbindung unserer Firmenstandards.  
Da das Frontend relativ viel Arbeit ergab, arbeitete Olivier einen grossen Teil an der Dokumentation, welches aber auch so abgemacht wurde.  
Für ein nächstes Mal könnten wir es vielleicht noch ein wenig besser aufteilen, dass alle Projektmitglieder von allem etwa gleich viel machen, grundsätzlich bin ich jedoch zufrieden mit unserem Endprodukt.

# Teil 3

## 16. Dokumentation Quellcode

### 16.1 Code von Frontend

Da der Code aus dem Frontend sehr viel Platz einnehmen würde, ist hier nur die Projektstruktur aufgelistet. Im Literaturverzeichnis ist der Link zu dem Repository hinterlegt.

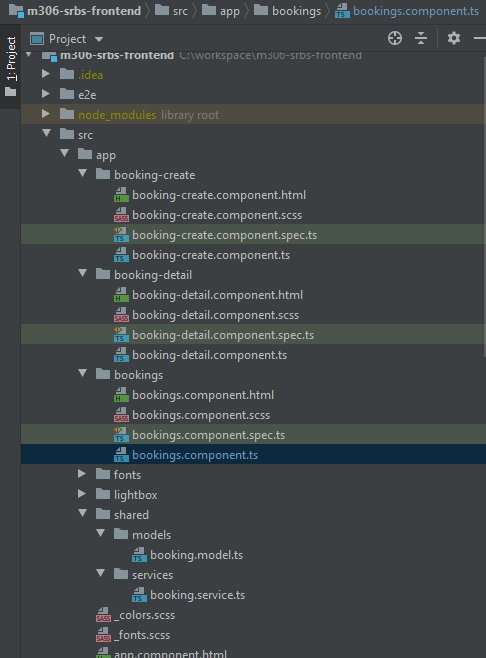


Abbildung 26

### 16.2 Code von Backend

src/main/java/m306backend/M306SrbsBackendApplication.java

*package m306backend*;  
  
*import org.springframework.boot.SpringApplication*;  
*import org.springframework.boot.autoconfigure.SpringBootApplication*;  
  
*/\*  
 Backend of SRBS  
 @author Winkler Olivier, INF5J, 2019  
 \*/  
  
@SpringBootApplication  
public class M306SrbsBackendApplication* {  
  
 *public static void* main(*String*[] *args*) {  
 *SpringApplication*.run(*M306SrbsBackendApplication*.*class*, *args*);  
 }  
}

src/main/resources/application.properties

*spring.datasource.url* = jdbc:mysql://localhost:3306/srbs\_database?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC  
*spring.datasource.username* = root  
*spring.datasource.password* = root  
  
  
*## Hibernate Properties  
# The SQL dialect makes Hibernate generate better SQL for the chosen database  
spring.jpa.properties.hibernate.dialect* = org.hibernate.dialect.MySQL5InnoDBDialect  
  
*# Hibernate ddl auto (create, create-drop, validate, update)  
spring.jpa.hibernate.ddl-auto* = update

src/main/java/m306backend/controller/BookingController.java

*package m306backend.controller*;  
  
*import m306backend.exception.ResourceNotFoundException*;  
*import m306backend.model.BookingModel*;  
*import m306backend.repository.BookingRepository*;  
*import org.springframework.beans.factory.annotation.Autowired*;  
*import org.springframework.http.ResponseEntity*;  
*import org.springframework.web.bind.annotation.*\*;  
  
*import javax.validation.Valid*;  
*import java.util.List*;  
  
*/\*  
 Controllerclass handling requests  
 @author Winkler Olivier, INF5J, 2019  
 \*/  
  
@RestController  
@CrossOrigin  
public class BookingController* {  
  
 *@Autowired  
 private BookingRepository*;  
  
 */\*  
 Returns a list with all bookings found in the hibernate database.  
 @return List  
 \*/  
 @GetMapping*("/bookings")  
 @ResponseBody  
 *public* List<BookingModel> getAllBookings() {  
 *return* bookingRepository.findAll();  
 }  
  
 */\*  
 Returns a single booking found by id  
 @return BookingModel  
 \*/* @GetMapping("/bookings/{id}")  
 *public* BookingModel getNoteById(@PathVariable(value = "id") *long* id) *throws* ResourceNotFoundException {  
 *return* bookingRepository.findById(id)  
 .orElseThrow(() -> *new* ResourceNotFoundException(id));  
 }  
  
 */\*  
 Creates a new booking  
 @return BookingModel  
 \*/* @PostMapping("/bookings/create")  
 *public* BookingModel createBooking(@Valid @RequestBody BookingModel) {  
 *return* bookingRepository.save(bookingModel);  
 }  
  
 */\*  
 Updates a specific booking  
 @return BookingModel  
 \*/* @PutMapping("/bookings/{id}")  
 *public* BookingModel updateNote(@PathVariable(value = "id") *long* id, @Valid @RequestBody BookingModel booking) *throws* ResourceNotFoundException {  
 BookingModel = bookingRepository.findById(id)  
 .orElseThrow(() -> *new* ResourceNotFoundException(id));  
  
 bookingModel.setBookingTime(booking.getBookingTime());  
 bookingModel.setRoom(booking.getRoom());  
 bookingModel.setENumber(booking.getENumber());  
  
 *return* bookingRepository.save(bookingModel);  
 }  
  
 */\*  
 Deletes a booking with id  
 @return ResponseEntitiy  
 \*/* @DeleteMapping("/bookings/{id}")  
 *public* ResponseEntity<?> deleteBook(@PathVariable(value = "id") *long* id) *throws* ResourceNotFoundException {  
 BookingModel = bookingRepository.findById(id)  
 .orElseThrow(() -> *new* ResourceNotFoundException(id));  
  
 bookingRepository.delete(bookingModel);  
 *return* ResponseEntity.ok().build();  
 }  
}

src/main/java/m306backend/exception/ResourceNotFoundException.java

*package m306backend.exception*;  
  
*import org.springframework.http.HttpStatus*;  
*import org.springframework.web.bind.annotation.ResponseStatus*;  
  
*/\*  
 Exception  
 @author Winkler Olivier, INF5J, 2019  
 \*/  
  
@ResponseStatus*(value = *HttpStatus*.NOT\_FOUND)  
*public class ResourceNotFoundException extends RuntimeException* {  
  
 */\*  
 Throws an self-made exception  
 \*/  
 public* ResourceNotFoundException(*Long fieldValue*) {  
 *super*(*String*.format("Booking not found with Id : '%s'",*fieldValue*));  
 }  
}

src/main/java/m306backend/model/BookingModel.java

*package m306backend.model*;  
  
*import lombok.Data*;  
  
*import javax.persistence.*\*;  
*import java.util.Date*;  
  
*/\*  
 Model of a booking  
 @author Winkler Olivier, INF5J, 2019  
 \*/  
  
@Entity // This tells Hibernate to make a table out of this class  
@Table*(name = "bookings")  
*@Data  
public class BookingModel* {  
 *@Id  
 @GeneratedValue*(strategy = *GenerationType*.AUTO)  
 *private long* id;  
  
 *private* Date bookingTime;  
  
 *private* Date bookingEndTime;  
  
 *private* String room;  
  
 *private* String eNumber;  
}

src/main/java/m306backend/repository/BookingRepository.java

*package m306backend.repository*;  
  
*import m306backend.model.BookingModel*;  
*import org.springframework.data.jpa.repository.JpaRepository*;  
*import org.springframework.stereotype.Repository*;  
  
*/\*  
 Class for Spring's own connection to hibernate  
 @author Winkler Olivier, INF5J, 2019  
 \*/  
  
@Repository  
public interface BookingRepository extends JpaRepository*<*BookingModel*, *Long*> {  
  
}

## 17. Literaturverzeichnis

Dokumentation Spring Boot

<https://docs.spring.io/spring-boot/docs/current/reference/html/>

Dokumentation Angular

<https://angular.io/docs>

SBB-Angularkomponenten-Library

<https://sbb-angular.app.sbb.ch/latest/>

SBB-Richtlinien

[https://digital.sbb.ch/de#](https://digital.sbb.ch/de)

Projekt auf Gitlab (Frontend)

<https://gitlab.iet-gibb.ch/owi104556/m306-srbs-frontend>

Projekt auf Gitlab (Backend)

<https://gitlab.iet-gibb.ch/owi104556/m306-srbs-backend>

Firebase

<https://firebase.google.com/docs>

## 18. Glossar

|  |  |
| --- | --- |
| Fremdwort | Bedeutung |
| API | Application Programming Interface 🡪 Kommunikationsstelle zwischen Front- & Backend |
| ES | Externe Schnittstelle |
| IS | Interne Schnittstelle |
| Firebase | Entwicklungsplattform |
| GUI | Grafische Benutzerfläche einer Applikation |
| Frontend | Präsentationsebene einer Webseite |
| Backend | Datenverarbeitung einer Webseite |
| Kritikalität | Wichtigkeit |