

## Bachelorarbeit 2019

# Entwicklung eines Business Impact Analyse Systems für die Schweizerische Mobiliar



Student: Mathis Fux

Dozent: Dr. Christiane Jungius

Abgabe: 31.07.2019

[www.hevs.ch](http://www.hevs.ch)

# ZUSAMMENFASSUNG

Das Ziel der vorliegenden Bachelorarbeit ist die Entwicklung eines Prototypen für ein Business Impact Analysen System. In Zusammenarbeit mit der Schweizerischen Mobiliar wird die Bachelorarbeit durchgeführt. Diese verfügt bereits über ein Business Impact Analysen System, welches auf Excel-Dateien basiert. Das Hauptziel dieser Abhandlung besteht in der Entwicklung eines Vorschlages das bisherige System effizienter umzusetzen.

Mithilfe einer intensiven Analyse der bisherigen Lösung werden alle Stärken und Schwächen aufgezeigt. Da die IT-Abteilung der Schweizerischen Mobiliar zukünftig eine Änderung an der Infrastruktur vornimmt und noch keine explizite Auskunft über das neue Software-Umfeld aufzeigen kann, sind bezüglich der verwendeten Technologien keine Einschränkungen vorgegeben. Der Prototyp wird insofern nicht in die Infrastruktur integriert. Infolgedessen wird eine Analyse der potenziellen Technologien in Berücksichtigung der Anforderungen der Schweizerischen Mobiliar aufgestellt. Mittels dieser Analyse werden der Schweizerischen Mobiliar vier unterschiedliche Umsetzungsvarianten präsentiert. Sie entscheiden sich für die Umsetzungsvariante, welche ihren Eigenentwicklungen entspricht. Um das graphische Design des Prototypen bestmöglich umzusetzen, werden Personas für das Verständnis der Arbeitsschritten erstellt. Anhand dieser Analysen werden Mock-Ups für das neue BIA System illustriert, um ein Design vorzuschlagen. Für das Projekt wird eine Priorisierungsliste mit allen Funktionalitäten verfasst. Die verlesene Variante wird durch das Veranschaulichen der Systemarchitektur und dem Zusammenspiel der Technologien erläutert. Daraufhin wird die Entwicklungsphase des Prototypen aufgezeigt. Der Prototyp beinhaltet nur einzelne Teile der Priorisierungsliste.

Der letzte Schritt dieser Bachelorarbeit widmet sich den angewendeten Testen und die Evaluierung der resultierten Arbeit. Die Auswertung des Prototypen und die Analyse der Weiterführung des Systems soll der Schweizerischen Mobiliar die Entscheidung erleichtern, inwiefern das Projekt zukünftig gefördert werden soll.

**Keywords:** Business Impact Analysis, Webservice, Angular, J2EE, Softwareentwicklung

# VORWORT

Die vorliegende Bachelorarbeit beansprucht die Zeitspanne von Mai 2019 bis Ende Juli 2019. Im Rahmen des Moduls Bachelorarbeit ist diese Diplomarbeit in der Fachhochschule HES-SO Valais/Wallis im Studiengang Wirtschaftsinformatik aufgesetzt. Diese Bachelorarbeit hat eine externe Unternehmung vorgeschlagen. Den Projektvorschlag hat die Schweizerische Mobiliar betreut. Frau Dr. Christiane Jungius ist die Betreuerin dieser Bachelorarbeit.

Dieses Projekt weist grosses Potenzial auf, da das Programm von Grund auf neu entwickelt wird. Aufgrund meiner technischen Interessen habe ich mich für diese Arbeit entschieden. Die erlernten Kompetenzen kann ich gut anwenden. Da keine genauere Programmierungsvorgaben vorgegeben sind kann ich anhand meiner eigenen Analyse selber entscheiden, wie das Programm zu realisieren ist. Mein errungenes Wissen der Fachhochschule HES-SO Valais/Wallis, praxisnah in einer Unternehmung anzuwenden, empfinde ich als sehr eine gute Erfahrung.

Das Hauptziel der Arbeit liegt darin, ein bestehendes Programm der Schweizerischen Mobiliar von Grund auf neu aufzubauen, damit die Mitarbeiter effizienter an den unterschiedlichen Prozessen arbeiten können. Daher wurde als erstes eine genaue Analyse der bestehenden Applikation festgelegt, um daraufhin anhand den Wünschen und Entscheidungen der Mobiliar einen nutzbringenden Prototyp zu entwickeln. Dieser Prototyp wird nur einzelne Hauptfunktionalitäten der bisherigen Lösung enthalten. Anfangs stellte sich die Herausforderung, den Projektumfang mit der Mobiliar abzudecken. Die Schweizerische Mobiliar plant im Jahre 2020 alle Eigenentwicklungen auf der Microsoft Azure Cloud bereitzustellen und ihre IT-Infrastruktur in naher Zukunft umzudisponieren. Aufgrund von diesen firmeninternen Gegebenheiten konnte für diese Entwicklung noch keine spezifische Vorgaben definiert werden. Nach Absprache mit der Schweizerischen Mobiliar wurde in der Planungsphase entschieden, einen Prototypen für das bestehende System zu entwickeln und kein funktionsreifes System. Dieser Prototyp soll anschliessend als eine Basis für die Weiterentwicklung des gesamten neuen BIA System dienen.

Nach genauer Analyse und dem Vergleich von bestehender sowie neuer Lösungskonzepte und Technologien bestand die Problematik darin, ein effizientes System mit Angular und J2EE zu entwickeln, dass den Ablauf der Prozesse der Schweizerischen Mobiliar klar vereinfachen wird. Eine weitere Problematik bestand darin, sich in die verwendeten Technologien einzuarbeiten und das Environment so aufzubauen, dass alle Systeme miteinander kommunizieren. Zudem war die Sicherheit der Daten ein äusserst

wichtiger Aspekt, um zu gewährleisten, dass nur berechtigte Personen Zugriff auf die Daten besitzen.

Anhand der Analyse des bestehenden Systems und der Anforderungen der Schweizerischen Mobiliar wurde deutlich festgestellt, dass der Projektumfang nicht für die geplante Zeit ausreicht. Aus diesem Grund wurden die Hauptfunktionalitäten umgesetzt, welche sich der Arbeitgeber wünschte. Der Schwerpunkt dieser Abhandlung besteht darin, durch die Implementierung und die Evaluation des neuen Prototypen eine Entscheidungsgrundlage für die Fortführung des Projektes vorzuführen.

In dieser Bachelorarbeit wird überwiegend die männliche Grundform verwendet, um die Leserfreundlichkeit zu erhöhen. Dabei werden sowohl die Männer als auch die Frauen angesprochen und mit einbezogen.

## DANKSAGUNG

Ich bedanke mich recht herzlich bei all den Personen, welche mich beim Durchführen meiner Bachelorarbeit stets unterstützt haben. Die erhaltene Unterstützung erlaubt mir, einen effizienten Prototyp zu implementieren.

Einen besonderen Dank richte ich an meine Dozentin und Betreuerin Frau Dr. Christiane Jungius. Durch ihre hilfreiche Unterstützung und konstruktive Kritik konnte ich mein Projekt bestmöglich umsetzen.

Zudem bedanke ich mich bei meiner Ansprechperson, Herrn Damian In-Albon. Er bietet mir die Möglichkeit, dass ich für die Schweizerische Mobiliar die Arbeit erstellen kann. Dank den wöchentlichen Feedbacks und der Kooperation mit ihm kann ich das Projekt exakt nach den Wünschen meines Auftraggebers durchführen.

Ein weiterer Dank gebührt meiner Ansprechperson der IT-Abteilung der Schweizerischen Mobiliar Herrn Emanuel Indermühle. Dank dem Einblick in die technische Rubrik der Schweizerischen Mobiliar konnte ich meine Analyse bestmöglich umsetzen und so gute Entscheidungen zu den ausgewählten Technologien treffen.

Zum Schluss bedanke ich mich bei all meinen Freunden und Familienangehörigen, welche mich unterstützt haben und meine Bachelorarbeit korrekturgelesen haben.

# INHALTSVERZEICHNIS

<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>VII</b>
<b>TABELLENVERZEICHNIS.....</b>	<b>X</b>
<b>ABKÜRZUNGSVERZEICHNIS .....</b>	<b>XI</b>
<b>EINLEITUNG .....</b>	<b>1</b>
<b>AKTUELLE SITUATION .....</b>	<b>2</b>
<b>ZIELSETZUNG .....</b>	<b>2</b>
<b>AUFTRAGGEBER .....</b>	<b>3</b>
<b>METHODISCHES VORGEHEN.....</b>	<b>3</b>
<b>1. ALLGEMEINE INFORMATIONEN.....</b>	<b>5</b>
1.1. BUSINESS CONTINUITY MANAGEMENT .....	5
1.2. BUSINESS IMPACT ANALYSIS .....	7
1.3. BIA – BISHERIGES SYSTEM .....	10
<b>2. IST-ANALYSE – BIA SYSTEM.....</b>	<b>17</b>
2.1. USER EXPERIENCE UND USABILITY BIA SYSTEM .....	17
2.2. ANALYSE DER REPORTINGS ERSTELLEN.....	19
2.3. ANALYSE TEAMWORK .....	22
2.4. ANALYSE VERWENDETE TECHNOLOGIEN .....	23
2.5. ANALYSE SICHERHEIT .....	23
2.6. SWOT ANALYSE .....	25
2.7. RISIKOANALYSE.....	26
<b>3. ANFORDERUNGEN DER SCHWEIZERISCHEN MOBILIAR.....</b>	<b>29</b>
3.1. BESTEHENDE SOFTWARE ALS ERSATZ.....	29
3.2. PERSONAS.....	30
3.3. DESIGN NEUES BIA-SYSTEM.....	33
<b>4. ANALYSE POTENZIELLE TECHNOLOGIEN .....</b>	<b>35</b>
4.1. VORGEHENSWEISEN.....	35
4.2. FRONTEND.....	37
4.3. BACKEND.....	39
4.4. VARIANTEN .....	51
<b>5. SYSTEMARCHITEKTUR.....</b>	<b>56</b>
5.1. ZUSAMMENSPIEL LAYER .....	58
5.2. RESTFUL WEBSERVICES.....	59
<b>6. IMPLEMENTATIONSPHASE .....</b>	<b>63</b>
6.1. VORGEHENSWEISE PROJEKTMANAGEMENT .....	63

6.2.	VORGEHENSWEISE IMPLEMENTIERUNG .....	65
6.3.	DATENBANKSCHEMA .....	66
6.4.	HIBERNATE VERBINDUNG MIT DEM DBMS .....	67
6.5.	HTTP-METHODEN .....	68
6.6.	FUNKTIONALITÄTEN .....	74
<b>7.</b>	<b>TESTPHASE .....</b>	<b>90</b>
7.1.	J2EE TESTE .....	90
7.2.	ANGULAR TESTE .....	92
7.3.	CODE COVERAGE .....	94
<b>8.</b>	<b>EVALUATION VOM PROTOTYP .....</b>	<b>95</b>
8.1.	GEGENÜBERSTELLUNG «GESCHÄFTSPROZESS HINZUFÜGEN» .....	95
8.2.	USER EXPERIENCE DES PROTOTYPENS .....	96
8.3.	FORTFÜHRUNG NEUES BIA SYSTEM .....	98
<b>9.</b>	<b>ÜBERGABE .....</b>	<b>102</b>
	<b>SCHLUSSFOLGERUNG .....</b>	<b>103</b>
	<b>LITERATURVERZEICHNIS .....</b>	<b>106</b>
	<b>ANHANG I: SOLL-PLANUNG .....</b>	<b>113</b>
	<b>ANHANG II: IST-PLANUNG .....</b>	<b>114</b>
	<b>ANHANG III: MOCK-UPS .....</b>	<b>115</b>
	<b>ANHANG IV: BESTÄTIGUNG VARIANTE 4 .....</b>	<b>119</b>
	<b>ANHANG V: PRODUCT BACKLOG .....</b>	<b>120</b>
	<b>ANHANG VI: TESTE DER SCHWEIZERISCHEN MOBILIAR .....</b>	<b>123</b>
	<b>ANHANG VII: CODE COVERAGE .....</b>	<b>124</b>
	<b>ANHANG VIII: FORTFÜHRUNG BIA SYSTEM .....</b>	<b>125</b>
	<b>ANHANG IX: RELEASE ROADMAP .....</b>	<b>126</b>
	<b>ANHANG X: IMPLEMENTATIONSITERATION .....</b>	<b>127</b>
	<b>ANHANG XI: SITZUNGSPROTOKOLL 06.05.2019 .....</b>	<b>131</b>
	<b>ANHANG XII: SITZUNGSPROTOKOLL 29.05.2019 .....</b>	<b>133</b>
	<b>ANHANG XIII: PREISLISTE «ALIVE-IT BUSINESS CONTINUITY FRAMEWORK» ....</b>	<b>134</b>
	<b>ANHANG XIV: DEPLOYMENT BLOB STORAGE .....</b>	<b>135</b>
	<b>ANHANG XV: DEPLOYMENT JAVA WEB APP .....</b>	<b>136</b>
	<b>ANHANG XVI: ABNAHME DES PROTOTYPEN .....</b>	<b>137</b>
	<b>SELBSTÄNDIGKEITSERKLÄRUNG .....</b>	<b>138</b>

# ABBILDUNGSVERZEICHNIS

Abbildung 1: Water-Scrum-Fall Modell.....	4
Abbildung 2: Business Continuity Phasen nach ISO Standard 22313 .....	6
Abbildung 3: Beispiel maximal tolerierbare Ausfallzeit .....	9
Abbildung 4: BIA Menu im bisherigen System .....	11
Abbildung 5: BIA Geschäftsprozesse anzeigen im bisherigen System .....	12
Abbildung 6: BIA Geschäftsprozess suchen im bisherigen System.....	12
Abbildung 7: BIA Geschäftsprozess bearbeiten im bisherigen System .....	13
Abbildung 8: BIA IT und Kritikalität im bisherigen System .....	14
Abbildung 9: BIA Ribbon Menu im bisherigen System .....	15
Abbildung 10: BIA Reporting im bisherigen System .....	16
Abbildung 11: BIA Geschäftsprozess anzeigen im bisherigen System.....	18
Abbildung 12: Reporting Gesamtübersicht kompakt .....	20
Abbildung 13: Reporting Übersicht Kritikalität nach Kategorie.....	20
Abbildung 14: Reporting Arbeitsplätze Kritikalitäts-Kategorie.....	21
Abbildung 15: Reporting Lieferantenliste.....	21
Abbildung 16: Reporting Krit. Funktionen Kat .....	22
Abbildung 17: Datenverlust Kroll Ontrack 2010.....	24
Abbildung 18: Risk Matrix Analysis .....	28
Abbildung 19: Mock-Up Dashboard .....	34
Abbildung 20: Drei-Schichten-Architektur.....	36
Abbildung 21: Beispiel Responsive Webdesign .....	38
Abbildung 22: PHP Framework.....	42
Abbildung 23: Node JS Framework.....	44
Abbildung 24: Systemarchitektur .....	56
Abbildung 25: Softwarearchitektur Backend.....	57
Abbildung 26: Zusammenspiel Layer .....	58
Abbildung 27: RESTful zustandslos sein.....	60
Abbildung 28: Beispiel XML vs. JSON.....	61
Abbildung 29: Kanban Dashboard .....	64
Abbildung 30: Vorgehensweise Implementierung BPMN .....	65
Abbildung 31: Datenbankschema BIA System .....	66
Abbildung 32: Implementation persistence.xml .....	67
Abbildung 33: Implementation Entity.....	68
Abbildung 34: Implementierung GET Methode.....	69
Abbildung 35: Implementation POST Methode .....	70

Abbildung 36: Implementation PUT Methode .....	70
Abbildung 37: Implementation DELETE Methode .....	71
Abbildung 38: Implementation HttpClient Angular .....	72
Abbildung 39: Implementation Security Filter .....	73
Abbildung 40: BIA System - Geschäftsprozess hinzufügen.....	74
Abbildung 41: Implementation Reaktive Formular .....	75
Abbildung 42: Implementation individueller Validator .....	75
Abbildung 43: Implementation Geschäftsprozess hinzufügen .....	76
Abbildung 44: BIA System - Assessment Teilnehmer verwalten .....	76
Abbildung 45: BIA System - Geschäftsprozess Übersicht .....	77
Abbildung 46: Implementation Custom Pipe .....	78
Abbildung 47: Implementation Sortierung Tabelle.....	78
Abbildung 48: BIA System - Bestätigungsmeldung Prozess entfernen.....	79
Abbildung 49: BIA System - Geschäftsprozess bearbeiten .....	79
Abbildung 50: BIA System - Auswahl Kritikalitätslevel.....	80
Abbildung 51: Implementation Anzeige der Kritikalität.....	81
Abbildung 52: BIA System - Beispiel eines JSON Web Token .....	82
Abbildung 53: Sequenzdiagramm JWT Generierung .....	83
Abbildung 54: Implementation JWT Generierung.....	84
Abbildung 55: Implementierung Ablaufdatum JWT .....	85
Abbildung 56: BIA System - Navigation Mitarbeiter und Administrator .....	85
Abbildung 57: Implementation Navigation .....	86
Abbildung 58: BIA System - Reportings Dashboard.....	86
Abbildung 59: BIA System - Reporting Geschäftsprozess Kategorie.....	87
Abbildung 60: BIA System - Dashboard.....	88
Abbildung 61: Implementation ChartJS.....	89
Abbildung 62: Implementation HTTP-Method Test.....	90
Abbildung 63: Implementation JWT Test.....	91
Abbildung 64: Implementation Integration Test: Komponente erstellt.....	92
Abbildung 65: Implementation Unit Test Validation reaktives Formular .....	93
Abbildung 66: Code Coverage Teste .....	94
Abbildung 67: BPMN Arbeitsschritte bisherigen System .....	95
Abbildung 68: BPMN Arbeitsschritte neues System .....	95
Abbildung 69: BIA System - Responsive Design.....	97
Abbildung 70: Multiuser-Funktionalität neues BIA-System .....	98
Abbildung 71: Vergleich Alive-IT und Eigenentwicklung.....	100
Abbildung 72: SOLL-Planung.....	113

Abbildung 73: IST-Planung .....	114
Abbildung 74: Mock-Up Login .....	115
Abbildung 75: Mock-Up Dashboard .....	115
Abbildung 76: Mock-Up Prozesse .....	115
Abbildung 77: Mock-Up Prozess hinzufügen.....	116
Abbildung 78: Mock-Up Prozess bearbeiten .....	116
Abbildung 79: Mock-Up IT-Liste .....	116
Abbildung 80: Mock-Up Reporting Dashboard .....	117
Abbildung 81: Mock-Up Reporting Kritikalitäts-Kategorie .....	117
Abbildung 82: Mock-Up Benutzerverwaltung .....	117
Abbildung 83: Mock-Up Workflows .....	118
Abbildung 84: Mock-Up Workflow hinzufügen .....	118
Abbildung 85: Bestätigung Variante 4 .....	119
Abbildung 86: Product Backlog Sicht 1 .....	120
Abbildung 87: Product Backlog Sicht 2 .....	121
Abbildung 88: Product Backlog Sicht 3 .....	122
Abbildung 89: E-Mail Test der Schweizerischen Mobiliar .....	123
Abbildung 90: E-Mail Code Coverage .....	124
Abbildung 91: E-Mail Fortführung BIA System .....	125
Abbildung 92: Release Roadmap.....	126
Abbildung 93: Implementationsiteration Woche 1 .....	127
Abbildung 94: Implementationsiteration Woche 2 .....	127
Abbildung 95: Implementationsiteration Woche 3 .....	128
Abbildung 96: Implementationsiteration Woche 4 .....	128
Abbildung 97: Implementationsiteration Woche 5 .....	129
Abbildung 98: Implementationsiteration Woche 6 .....	130
Abbildung 99: Preisliste «alive-IT Business Continuity Framework» .....	134
Abbildung 100: E-Mail Abnahme des Prototypen .....	137

# TABELLENVERZEICHNIS

Tabelle 1: Muster einer BIA.....	8
Tabelle 2: SWOT Portfolio-Analyse.....	26
Tabelle 3: Risk Analysis Table .....	27
Tabelle 4: Vorstellung von drei Personas.....	31
Tabelle 5: Auswertung PHP Framework .....	49
Tabelle 6: Auswertung NodeJS Framework .....	51
Tabelle 7: Auswertung Varianten .....	54
Tabelle 8: HTTP Status-Codes .....	59
Tabelle 9: Vergleich von XML und JSON.....	62
Tabelle 10: Evaluation der Risikobehhebung .....	96
Tabelle 11: Gegenüberstellung Eigenentwicklung und Alive-IT Produkt.....	99

# ABKÜRZUNGSVERZEICHNIS

<b>IPERKA</b>	Informieren Planen Entscheiden Realisieren Kontrollieren Auswerten
<b>BC</b>	Business Continuity (Geschäftsfortführung)
<b>BCM</b>	Business Continuity Management
<b>BCP</b>	Business Continuity Planning
<b>BIA</b>	Business Impact Analysis
<b>GLB</b>	Geschäftsleitungsbereich
<b>MTA</b>	Maximal tolerierbare Ausfallzeit
<b>XLSM</b>	Excel Spreadsheet XML macro-enabled
<b>XLSX</b>	Excel Spreadsheet XML
<b>VBA</b>	Visual Basic for Applications
<b>UX</b>	User Experience
<b>SWOT</b>	Strengths Weaknesses Opportunities Threats
<b>RAT</b>	Risk Analysis Table
<b>RMA</b>	Risk Matrix Analysis
<b>HR</b>	Human Resources
<b>HTML5</b>	Hypertext Markup Language, Version 5
<b>CSS3</b>	Cascading Style Sheets, Version 3
<b>MVC</b>	Model View Controller
<b>MVT</b>	Model View Template
<b>MVVM</b>	Model View ViewModel
<b>ORM</b>	Object-relational Mapping
<b>CMS</b>	Content Management System
<b>IDE</b>	Integrated Development Environment
<b>IIS</b>	Internet Information Server
<b>J2EE</b>	Java 2 Enterprise Edition
<b>JES</b>	Java Enterprise Service

<b>RWC</b>	Rich Web Client
<b>URI</b>	Uniform Ressource Identifier
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>POM</b>	Project Object Model
<b>REST</b>	Representational State Transfer
<b>BPMN</b>	Business Process Model and Notation

# EINLEITUNG

Eine Versicherungsfirma, wie die Schweizerische Mobiliar, muss den Zugriff auf Ihre Dienstleistungen zu jeder Zeit gewährleisten können. Die schwerwiegendste denkbare Krisensituation wäre ein Totalausfall der Systeme, so dass den Kunden der Zugriff nicht weiter gewährleistet werden kann.<sup>1</sup> Aus solchen Gründen erstellen mehrere Unternehmen einen «Business Continuity» Plan oder zu Deutsch einen «Geschäftsfortführung» Plan. Die Geschäftsprozesse der Mobiliar können durch gravierende Schadenereignisse ausfallen oder behindert werden. Die Ursache für solche drastische Ereignisse bezieht sich auf technisches und vorsätzliches Versagen oder höhere Gewalt. Die unterschiedlichen Ausfälle können erhebliche Auswirkungen für die Schweizerische Mobiliar zur Folge haben. Darunter fallen die finanzielle Auswirkungen, wie zum Beispiel Umsatzverlust oder Verlust von Kunden. Aber auch nicht-finanzielle Auswirkungen können das Image des Unternehmens schädigen. Die Geschäftsfortführung (Business Continuity Planung) befasst sich mit allen notwendigen Vorkehrungen, während oder nach ungewöhnlichen Vorkommnissen, damit die Geschäftstätigkeiten der Mobiliar sichergestellt werden können. Bei der Planung werden alle erforderlichen Ressourcen für die Geschäftstätigkeit betrachtet. Als Ressourcendefinierung wird nicht nur die Technik als zentrale Rolle berücksichtigt, da andere Faktoren auch exorbitant wichtig sind. «Die Geschäftstätigkeit der Mobiliar kann nur mit der notwendigen Technik, geeigneten Räumlichkeiten und dem erforderlichen Personal aufrecht erhaltenen werden», gemäss der Schweizerischen Mobiliar (2016, S. 3). Der Hauptbestandteil eines Business Continuity Plan (BCP) sind die Business Impact Analysen kurz gesagt BIA. Das Akronym «BIA» stellt sich aus den Anfangsbuchstaben des Begriffes **B**usiness **I**mpact **A**nalysis zusammen und wird für dieses System als Applikationsname übernommen (die Schweizerische Mobiliar, 2016, S 2-8).

Um grossen Schaden der Geschäftstätigkeit der Schweizerischen Mobiliar zu vermeiden, ist ein wesentlicher Vorteil frühzeitig über die Risiken nachzudenken. Durch das Bestimmen der Kritikalität für jeden Geschäftsprozess wird dieses Ziel in der Business Impact Analyse erreicht. Diese Analysen müssen jeweils von den verantwortlichen Mitarbeitern regelmässig überprüft werden. Deren Aufgabe besteht darin, zu bestimmen wie zeitkritisch die unterschiedlichen Prozesse für die Aufrechterhaltung des Geschäftsbetriebes sind. Die definierten Zeitvorgaben unterstützen das Management dabei, schnellstmöglich auf Ereignisse zu reagieren und Massnahmen treffen zu können. Für diese diversen Analysen

---

<sup>1</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr Andrist Albert (2018)

benutzt die Schweizerische Mobiliar selbst erstellte Excel-Dateien. Diese Dateien werden ausschliesslich intern in der Schweizerischen Mobiliar verwendet und deshalb gehören die Mitarbeiter zu den einzigen Anwendern des Systems (die Schweizerische Mobiliar, 2016, S. 2-8).

Diese Bachelorarbeit befasst sich mit dem Entwickeln eines Prototypen für das Erstellen der Analysen. Dieser Prototyp soll der Schweizerischen Mobiliar als Basis der Weiterentwicklung dienen. Der Prototyp soll den Mitarbeitern die Möglichkeit bereitstellen, effizienter und gleichlaufend miteinander arbeiten zu können. Der Einsatzbereich dieses Prototypen soll das alte Excel-basierende System in naher Zukunft ersetzen.

## AKTUELLE SITUATION

Das aktuelle Business Impact Analysen System der Schweizerischen Mobiliar besteht aus drei unterschiedlichen Excel-Dateien. Da bereits 92 Prozesse verarbeitet werden, haben die Mitarbeiter den Überblick über die spezifischen Geschäftsprozesse verloren. Aus diesem Grund wünscht sich die Schweizerische Mobiliar ein neu entwickeltes System. Dies soll eine übersichtliche Ansicht, sowie effizientere Arbeitsschritte gewährleisten. Ein äusserst wichtiger Punkt ist die Multiuser-Funktionalität. Die bisherige Lösung erschwert die produktive Zusammenarbeit unter den Mitarbeitern.

## ZIELSETZUNG

Ziel dieser Bachelorarbeit liegt darin, einen produktiveren Ansatz für das Business Impact Analysen Programm zu finden. Dafür werden Analysen und Vergleiche der bestehenden Applikation und der potenziellen Technologien durchgeführt. Das neue System soll den Mitarbeitern die parallele Zusammenarbeit bieten können. Da die IT-Entwicklung des Auftraggebers interne Änderungen an der Infrastruktur vornehmen und bislang noch keine genauere Informationen über die zukünftige Infrastruktur präsentieren können, wird der Prototyp nicht in die Unternehmung integriert. Das Ergebnis dieser Bachelorarbeit soll dem Unternehmen lediglich als eine Basis für die Weiterführung des Projektes dienen. Mithilfe von zeitlichen und finanziellen Analysen soll der Unternehmung zum Schluss aufgezeigt werden, ob sich die Weiterführung des Projektes auszahlt.

# AUFTRAGGEBER

Der Auftraggeber dieser Bachelorarbeit ist «die Schweizerische Mobiliar». Die Schweizerische Mobiliar ist ein schweizweites Versicherungs- und Vorsorgeunternehmen. 1826 wurde das Unternehmen als erste private Versicherung der Schweiz und als Genossenschaft gegründet. Inzwischen sind mehr als zwei Millionen Kundinnen und Kunden bei der Schweizerischen Mobiliar versichert. Das Unternehmen beschäftigt nach dem aktuellen Stand 5374 Mitarbeitende und davon sind 345 Lernende (die Schweizerische Mobiliar, 2018, S. 92, 102-106).

Das Thema dieser Bachelorarbeit wurde vom Abteilungsleiter Business Continuity Management (BCM) der Schweizerischen Mobiliar vorgeschlagen. Täglich wenden die Mitarbeiter der Abteilung BCM das Business Impact Analysen System an und suchen deshalb nach einer effizienteren Lösung.

# METHODISCHES VORGEHEN

Durch eine Projektmanagement Methode wird das Vorgehen dieser Bachelorarbeit strukturiert. Unterschiedliche Projektmanagement-Methoden können verwendet werden. Die Auswahl stützt sich auf klassische oder agile Methoden. Klassische Projektmanagement-Methoden beziehen sich auf mehrere Phasen, welche hintereinander durchgeführt werden. In den ersten Phasen werden Informationen gesammelt, Analysen aufgestellt und anhand unterschiedlicher Analysen werden Entscheidungen über das weitere Vorgehen des Projektes getroffen. IPERKA ist eine sogenannte, klassische Projektmanagement-Methode (Arndt, Hermanns, Prof. Dr. Kuchen, & Poldner, 2009, S. 7-8).

Das gewünschte Programm der Schweizerischen Mobiliar ist ein sehr komplexes System. Das Vordefinieren aller Implementierungsschritte im vornhinein ist sinnlos, da die Schweizerische Mobiliar sich noch nicht über alle gewünschten Funktionalitäten im Klaren ist. In der Entwicklungsphase würde sich eine agile Projektmanagement-Methode besser eignen, da die Schweizerische Mobiliar bei allen ihrer IT-Softwareentwicklungen agile Projektmethoden bevorzugen, wie zum Beispiel SCRUM, SAFe oder Kanban.<sup>2</sup> Bei einer agilen Projektmanagement-Methode wird das Projekt in mehrere Teilstücke, sogenannte Tasks, aufgeteilt und durch mehrere Iterationen implementiert. Diese Projektmethodologie

---

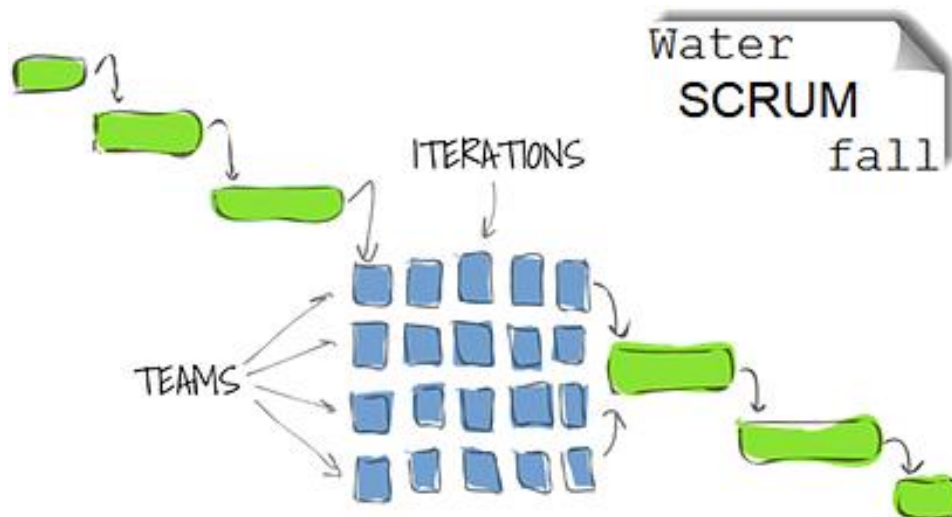
<sup>2</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr In-Albon Damian (2019)

basiert auf einem stetigen Austausch mit dem Kunden und kann so das Programm exakt nach den Wünschen des Kunden entwickeln (Dr. Preussig, 2018, S. 6-10, 13-15).

Aufgrund der benötigten Analysen und Entscheidungen, die für das Projekt der Mobiliar noch erstellt und getroffen werden müssen, wird in dieser Bachelorarbeit eine hybride Projektmanagement Methode angewendet. Die hybride Projektmethodologie ist eine Kombination von verschiedener Vorgehensmodellen (Liebig, 2014, S. 131-132).

In diesem Fall wird für die Informationensammlung, Analyse, Entwurf und Wartung eine klassische Wasserfall Methodologie ausgewählt. Für die Implementierungsphase des Wasserfallmodells, wird eine agile Projektmanagement Methode angewendet, wie in der Abbildung 1 aufgeführt. Eine derart bekannte hybride Methode existiert bereits, namens «Water-Scrum-Fall» (West, 2011, S. 9-11). Damit ein Entwicklungsteam in SCRUM produktiv arbeiten kann, müssen mindestens drei Mitglieder an der Entwicklung beteiligt werden (Schwaber & Sutherland, 2013, S. 6). Aus diesem Grund wird in dieser Bachelorarbeit Kanban als Ersatz von SCRUM verwendet. Kanban ist eine agile Methode, um einen bestehenden Prozess in kleinen Teilschritten zu verbessern. Dabei werden wie in SCRUM, die Prozesse in mehrere Teilprozesse unterteilt und Schritt für Schritt ausgeführt (Dickmann, 2009, S. 11-15). Durch ein wöchentliches Iterations-Meeting mit dem Kunden soll ein stetiges Feedback und Anpassungen für das Projekt wiederlegt werden.

Abbildung 1: Water-Scrum-Fall Modell



Quelle: (Gustainiene, 2017)

# 1. Allgemeine Informationen

Um das bestehende Business Impact Analyse System der Schweizerischen Mobiliar zu verstehen, wird in dieser Abhandlung zuerst das Unternehmen sowie deren Geschäftsprozesse vorgestellt. Darauf folgt eine ausführliche Analyse des komplexen BIA-Systems.

Diese vorliegende Abhandlung wird nicht als vertraulich deklariert und aus diesem Grund werden alle Abbildungen und Tabellen mit fiktiven Daten geführt.

## 1.1. Business Continuity Management

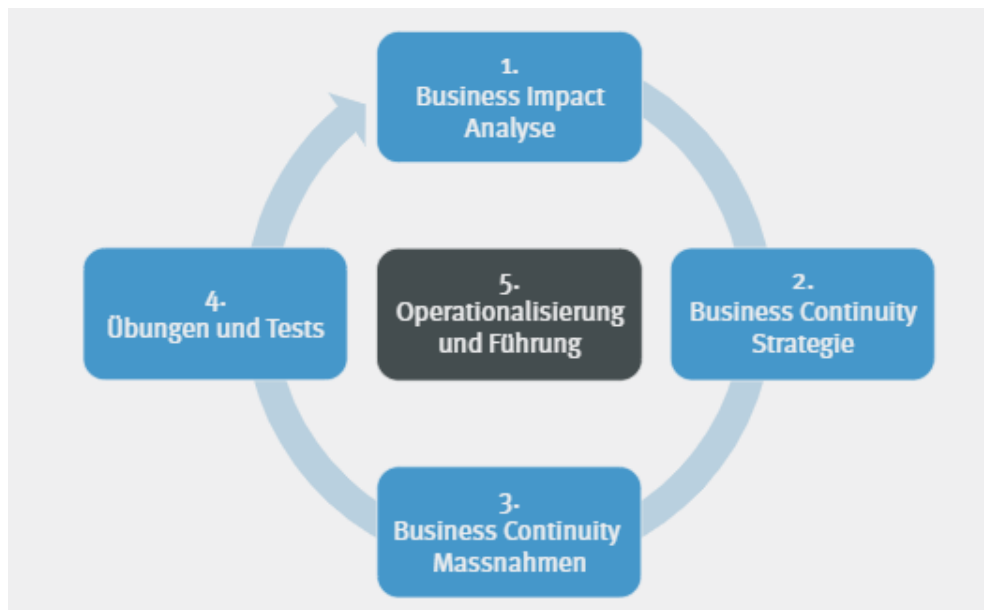
Die Schweizerische Mobiliar verfügt über eine Abteilung, welche für das Riskmanagement der Prozesse verantwortlich ist und sich mit der Business Continuity Prozess Leistung beschäftigt.<sup>3</sup> Frühzeitig über die möglichen Risiken nachzudenken und zu analysieren ist von grossem Vorteil, da bei einer Krisensituation das Management eines Unternehmens gezwungen ist, schnellstmöglich zu reagieren und Veränderungen vorzunehmen. Aufgrund der ständigen Erreichbarkeit der verschiedenen Dienstleistungen der Schweizerischen Mobiliar, muss bei einer Krisensituation rasche Massnahmen getroffen werden können.<sup>4</sup> Diese Analysen werden im Business Continuity Management durchgeführt. Um sich vor internen sowie auch von externen Risiken im Unternehmen zu schützen, werden mehrere Risikopläne in dem Business Continuity Management aufgestellt. Um diese Reaktionspläne stets auf dem Laufenden zu halten und das kontinuierliche Sicherstellen des Managements zu gewährleisten, wird der Vorgang nach fünf Business Continuity Phasen aufgeteilt (PhD Filipović, PhD Krišto, & Podrug, 2017, S. 99-101).

---

<sup>3</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr In-Albon Damian (2019)

<sup>4</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr Andrist Albert (2018)

Abbildung 2: Business Continuity Phasen nach ISO Standard 22313



Quelle: (SVV, 2015, S. 5)

Der Schweizerische Versicherungsverband [SVV] stellt alle fünf Business Continuity Phasen mit deren Bezeichnung in der Abbildung 2 graphisch dar. Die erste Phase des Business Continuity Management befasst sich mit der Business Impact Analyse. In diesen Analysen werden alle wichtige Geschäftsprozesse durchleuchtet. Die Analyse stützt sich dabei auf die zeitkritische Handlungsmassnahmen und die benötigte Ressourcen bei einer Krisensituation. Wichtig bei den Geschäftsprozessen zu berücksichtigen, sind die essenziellen Abhängigkeiten voneinander (SVV, 2015, S. 1-7).

Die jeweiligen Prozessverantwortlichen sind für die ihnen zugeteilten Geschäftsprozesse zuständig. Alle zwei bis drei Jahre werden alle Geschäftsprozesse überprüft und falls notwendig angepasst. Die Schweizerische Mobiliar bezieht sich hierbei auf vier verschiedene Ressourcen: Technik, Infrastruktur, Personal und Lieferanten (die Schweizerische Mobiliar, 2016, S. 6-7).

In der zweiten Phase bezieht sich das Business Continuity Management auf die Business Continuity Strategie. Diese unterschiedlichen Strategien werden basierend von der ersten Phase der Business Impact Analyse erstellt. Die Schweizerische Mobiliar legt in dieser Phase die maximale tolerierbare Ausfallzeit der unterschiedlichen Ressourcen fest. Zudem wird der Umfang der Business Continuity Massnahmen der dritten Phase definiert (die Schweizerische Mobiliar, 2016, S. 4-5).

Die nächste Phase befasst sich mit den Business Continuity Massnahmen. In dieser Phase des Business Continuity Management werden unterschiedliche Umkehrmassnahmen geplant. Diese Vorbereitungen werden angewendet, sobald ein zeitkritischer Geschäftsprozess eintritt, bei dem sofort Massnahmen in Kraft treten müssen. Durch den Einsatz der Massnahme kann der Schaden minimiert oder sogar behoben werden um den Prozess und die Ressourcen wiederherstellen (die Schweizerische Mobiliar, 2016, S. 4-5).

Übungen und Teste sind die vierte Phase des Business Continuity Managements. Die Schweizerische Mobiliar führt für die zeitkritischen Geschäftsprozesse verschiedene Übungen und Teste durch. Mithilfe dieser periodischen Teste kann das Unternehmen sicherstellen, ob die geplante Zeit ausreichend ist, um bei einer Krisensituation die Geschäftsprozesse und Ressourcen wiederherzustellen. Wichtig ist dabei das fortlaufende Protokollieren der Testergebnisse (die Schweizerische Mobiliar, 2016, S. 4-5).

Bei der letzten Phase des Business Continuity Management werden die Operationalisierung und Führung betrachtet. Von grosser Bedeutung ist, dass das Business Continuity Management im Unternehmen verankert und berücksichtigt wird. Für die Ergebnisbewältigung muss eine geeignete Organisation definiert sein, welche für bestimmte Geschäftsprozesse zuständig ist. Für diese Funktionen und Gremien müssen Verantwortungsketten, Rollen und Kompetenzen bestimmt sein (SVV, 2015, S. 7).

## **1.2. Business Impact Analysis**

Die Schweizerische Mobiliar verwendet ein selbstentwickeltes BIA-System. Dieses Business Impact Analysen Programm hilft dem Unternehmen die notwendigen Vorkehrungen und Massnahmen für die Geschäftsprozesse zu planen. Ziel ist, für jeden Geschäftsprozess die Kritikalität zu bestimmen, welches mit dem Erstellen durch Analysen und wiederholenden Überprüfungen erreicht wird. Um die Aufrechterhaltung des Geschäftsbetriebes zu gewährleisten, besteht die Hauptaufgabe des BIA-Systems zu definieren, welche Geschäftsprozesse wie zeitkritisch sind. Dabei muss beachtet werden, dass Geschäftsprozesse die als unkritisch eingeschätzt werden, trotzdem in Betracht gezogen werden müssen. In der Tabelle 1 ist ein Muster einer Beispiel BIA Tabelle mit fiktiven Daten aufgelistet. (die Schweizerische Mobiliar, 2016, S. 6-7).

Tabelle 1: Muster einer BIA

Prozess: Leistungserbringung		1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 W.	2 W.	1 Mt	> 1 Mt
Finanzielle Auswirkungen	Ertragsausfall / Umsatzverlust	1	1	1	1	1	2	2	2	3
	Mehrkosten	1	1	1	1	1	1	1	2	3
	Mehraufwand	1	1	1	1	1	2	3	3	3
Nicht-finanzielle Auswirkungen	Compliance	1	1	1	1	1	1	1	1	3
	Servicequalität / Image	1	1	1	1	2	2	3	4	5
	Betrieb / Operation	1	1	1	1	1	2	2	2	3
<b>Gesamt-Kritikalität</b>		<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

Quelle: eigene Darstellung in Anlehnung an (die Schweizerische Mobiliar, 2016, S. 6)

Die Prozesse werden in finanzielle und nicht-finanzielle Auswirkungen aufgeteilt. Gemäss Bundesamt für Sicherheit in der Informationstechnik (BSI) wird zur Vereinfachung der Darstellung der Kritikalität qualitative Auswirkungsklassen verwendet. Durch diese Auswirkungsklassen ist die Bewertung des Schadens darzustellen (2013, S. 15-20).

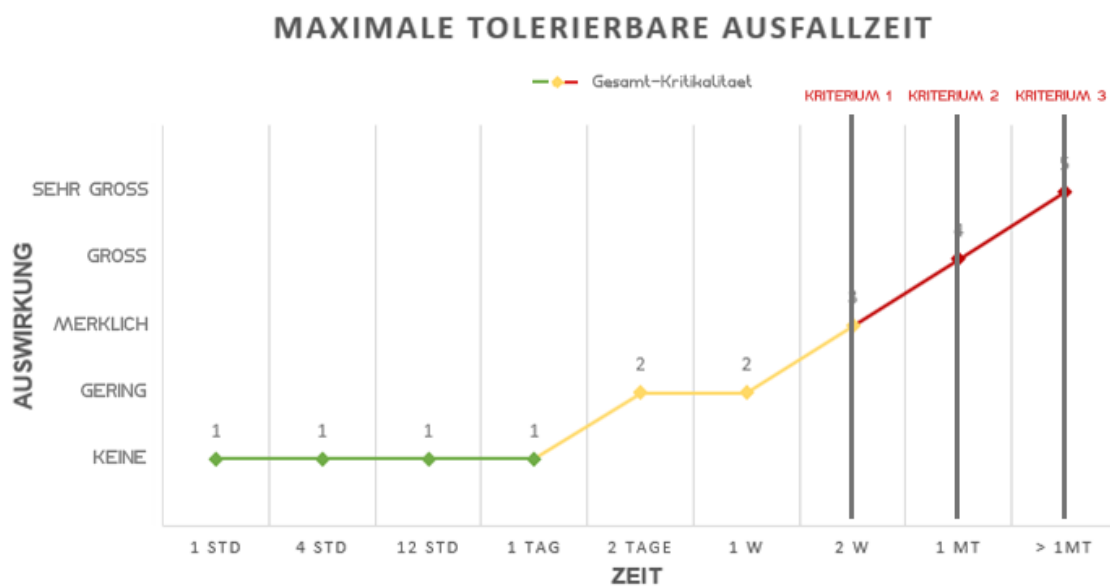
Bei der Schweizerischen Mobiliar wird die Auswirkung eines Totalausfalls mit den unterschiedlichen Geschäftsprozessen von dem jeweiligen Prozessverantwortlichen eingeschätzt. Diese betrachten die Auswirkung anhand von bestimmten Stunden, Tage, Wochen und Monaten (die Schweizerische Mobiliar, 2016, S. 6).

Gemäss der Geschäftsprozessanalyse wird die Gesamt-Kritikalität bestimmt, für die jeweils eine maximal tolerierbare Ausfallzeit (MTA) definiert wird. Damit die Schweizerische Mobiliar keinen hohen Schaden erleidet, wird die maximal tolerierbare Ausfallzeit definiert. Diese bezeichnet die Zeitspanne, in dem sich der Prozess spätestens im Notbetrieb befinden muss (Kötterheinrich & Oevel, 2015, S. 6).

Die folgende Abbildung 3 zeigt in einer graphischen Ansicht die Analyse der Gesamtkritikalität mit den verwendeten Daten des Beispiels der Tabelle 1.

Durch eine Akzeptanzlinie kann die MTA in einem Diagramm visuell abgebildet werden. Daraus lassen sich drei unterschiedliche Kriterien berücksichtigen, bei denen das Unternehmen bereits einen hohen Schaden erwarten muss (Kötterheinrich & Oevel, 2015, S. 6-9).

Abbildung 3: Beispiel maximal tolerierbare Ausfallzeit



(Eigene Darstellung)

Beim ersten Kriterium, welches in der Abbildung 3 ersichtlich ist, wird ein hoher Schaden zu erwarten sein. Wenn die Ausfallzeit des Prozesses bereits ein Monat in Anspruch nimmt, wie in dem Kriterium zwei ersichtlich, muss die Schweizerische Mobiliar bereits mit einem sehr hohen Schaden rechnen. Beim dritten Kriterium ist bereits die höchste Auswirkungsklasse der Schweizerischen Mobiliar «sehr gross» erreicht. Wenn diese Krisensituation erreicht wird, muss das Unternehmen einen sehr hohen finanziellen und/oder nicht-finanziellen Schaden bis hin zum Konkurs erwarten.<sup>5</sup>

<sup>5</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr Andrist Albert (2018)

### 1.3. BIA – bisheriges System

Das Business Impact Analysen System der Mobiliar wird von der Abteilung «Business Continuity Management» verwendet. Das System wurde mit der Tabellenkalkulations Software Excel entwickelt. Als Basis des BIA-Systems werden folgende drei Excel-Dateien verwendet:

#### 1. BIA-Datenaufnahme-Master.xlsm

In dieser Datei werden die Stammdaten des BIA Systems hinterlegt. Gemäss aktuellem Stand pflegt das System 92 Geschäftsprozesse. In dieser Datei befinden sich zusätzlich noch alle Funktionalitäten, um die Geschäftsprozesse zu pflegen und daraus Reportings zu erstellen. Als Dateierendung wird «.xlsm» (Excel Spreadsheet XML macro-enabled) verwendet.

Dieses Dateiformat muss verwendet werden, da innerhalb der Datei mit Makros und Visual Basic for Applications (VBA) gearbeitet wird. Die Autoren Ron Petrusha und Nancy Kotary erklären, dass «In earlier days, a *macro* consisted of a series of keystrokes that was recorded and assigned to a hot key» (2002, S. 42). Hierbei wird dies als eine spezielle Art von einem Unterprogramm bezeichnet, welches heutzutage anspruchsvoller ist. Visual Basic for Applications ist die Programmiersprache, welche in allen Microsoft Office Umgebungen zugrunde liegt (Petrusha & Kotary, 2002, S. 42-47).

#### 2. Reporting.xlsx

In dieser Excel-Datei werden die generierten Analysen von der Datei «BIA-Datenaufnahme-Master.xlsm» angezeigt. Insgesamt werden bis zu 40 Reportings aufgestellt.

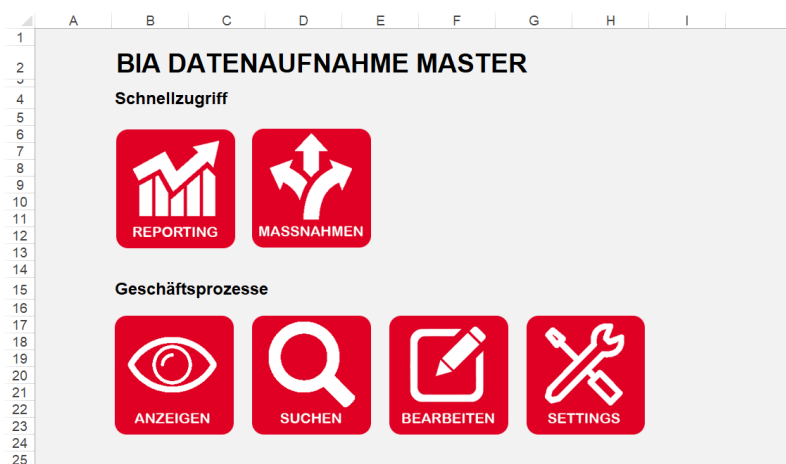
#### 3. Massnahmen.xlsx

Alle existierenden Massnahmen werden mit dem Zusammenhang zu den Geschäftsprozessen in dieser Excel-Datei hinterlegt. Da diese Datei erst seit jüngster Zeit erstellt wurde, sind hier erst sechs Massnahmen hinterlegt. Vor zwei Jahren wurden noch alle Massnahmen der jeweiligen Geschäftsprozesse in ein separates Textdokument hinterlegt. Dazu wurde das Textverarbeitungsprogramm Word von Microsoft verwendet (die Schweizerische Mobiliar, 2017, S. 1-9).

## BIA Menu

Beim Start der «BIA-Datenaufnahme-Master.xlsm» wird automatisch das BIA Menu angezeigt, wie auf der Abbildung 4 ersichtlich. Das BIA Menu besteht aus einem Excelregister, welches als Navigation angewendet wird. Über den Schnellzugriff können die anderen zwei Excel-Dateien geöffnet werden. Die anderen vier Navigationsfunktionen werden für die Ansichten der Geschäftsprozesse verwendet.

Abbildung 4: BIA Menu im bisherigen System



(Eigene Darstellung)

## BIA Geschäftsprozesse anzeigen

Diese Auflistung der Geschäftsprozesse wird in der Datei «BIA-Datenaufnahme-Master.xlsm» über die Navigationsfunktion «Geschäftsprozess anzeigen» erstellt. Alle Geschäftsprozesse erhalten eine Kategoriennummer, welche in dieser Ansicht gruppiert wird. Diese Kategoriennummern sind in der Abbildung 5 ersichtlich. Durch das Betätigen der divergierten Kategorien werden allen diesbezüglichen Prozesse angezeigt. Die Prozesse werden gemäss der ausgewählten Kategorie in dem Geschäftsprozessliste-Register gefiltert angezeigt. Damit alle Geschäftsprozesse wieder in der Liste aufgelistet werden, müssen alle Geschäftsprozesslisten auf dieser Ansicht neu geladen werden. Dies wird durch die blau-markierten Knöpfe erzielt. Beide müssen betätigt werden, um alle Prozesse anzuzeigen.

Abbildung 5: BIA Geschäftsprozesse anzeigen im bisherigen System

	A	B	C	D	E
1					
2		Menu	Geschäftsprozesse neu laden	Alle Geschäftsprozesse anzeigen	
3					
4		01	02	03	04
5		05	06	07	09
6		10	11	12	13
7		14	15	16	17
8		18	19	20	21
9					

(Eigene Darstellung)

## BIA Geschäftsprozesse durchsuchen

In der «BIA-Datenaufnahme-Master.xlsm» ist jeder einzelne Geschäftsprozess als ein eigenes Register hinterlegt. Da das System bereits 92 Geschäftsprozesse verwaltet, stellt das Programm eine Suche zur Verfügung. Die Suche unterstützt den Anwender den Geschäftsprozess schneller zu finden. Gemäss des gesuchten Wortes werden alle Geschäftsprozesse angezeigt, welche diese Zeichenkette enthalten. Dies ist auf der Abbildung 6 durch das Beispielwort «Service» dargestellt. Anschliessend kann durch die Verlinkung direkt auf das Register des Prozesses zugegriffen werden. Damit ein neuer Prozess im Inhaltsverzeichnis angezeigt wird, muss dieser in einem separaten Register angegeben werden.

Abbildung 6: BIA Geschäftsprozess suchen im bisherigen System

	A	B	C	D
1				
2	SUCHEN:	Service	Alle anzeigen	Menu
3				
4		15.a Service Layer	<a href="#">15.a.zuG</a>	
5		17.h Finanzieller Service	<a href="#">17.h.v1</a>	
6		21 Service	<a href="#">21.Service</a>	
7				
8				
9				

(Eigene Darstellung)

## BIA Geschäftsprozesse bearbeiten

Um die Bearbeitung eines Geschäftsprozesses zu vereinfachen, wurde das Formular der Abbildung 7 in der «BIA-Datenaufnahme-Master.xlsm» erstellt. Mit Hilfe eines Auswahlfeldes aller Geschäftsprozesse, kann der gewünschte Prozess ausgewählt werden. Anschliessend werden die Daten als ein Verweis in das Formular eingefügt und können so angepasst werden. Nach diversen Anpassungen muss der Vorgang abgespeichert werden, um die Daten im dazugehörigen Geschäftsprozess zu übernehmen.

Abbildung 7: BIA Geschäftsprozess bearbeiten im bisherigen System

	A	B	C	D	E	F	G	H	I	J	K	
1												
2		<b>GESCHÄFTSPROZESS BEARBEITEN</b>										
3		Prozess:	<input type="text"/>									
4		Prozessbeschreibung:	<input type="text"/>									
5		Risikokoordinator:	<input type="text"/>									
6		GLB/SF:	<input type="text"/>									
7		GLB/SF-Verantwortlicher:	<input type="text"/>									
8		Prozess-Verantwortlicher:	<input type="text"/>									
9		Assessment - Datum:	<input type="text"/>									
10		Assessment - Teilnehmer:	<input type="text"/>		<input type="text"/>		<input type="text"/>		<input type="text"/>		<input type="text"/>	
11		Status:	<input type="text"/>									
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												

(Eigene Darstellung)

## BIA Geschäftsprozesse hinzufügen

Damit ein Geschäftsprozess hinzugefügt werden kann, müssen mehrere Schritte in der Datei «BIA-Datenaufnahme-Master.xlsm» vorgenommen werden. Als erster Arbeitsschritt muss das Musterregister eines Geschäftsprozesses dupliziert werden und der Name angepasst werden. Anschliessend muss der jeweilige Prozess in zwei unterschiedlichen Registern «Geschäftsprozessliste» und «Prozessliste-Planung» separat hinzugefügt werden. Dies sind zwei sehr ähnliche Geschäftsprozesslisten, welche in unterschiedlichen Funktionalitäten als absoluter Bezug verwendet werden.

Beim Einfügen des Prozesses in den unterschiedlichen Registern ist wichtig zu achten, dass der Prozess an der richtigen Stelle eingefügt wird und auch das Format der Tabelle beibehalten wird. Nach diesem Schritt kann der Anwender alle Informationen bezüglich des Geschäftsprozesses in dem dazugehörigen Register ausfüllen. Zu guter Letzt, muss der Name des Geschäftsprozesses noch im Inhaltsverzeichnis hinzugefügt und zusätzlich noch ein Hyperlink erstellt werden.

## BIA IT und Kritikalität

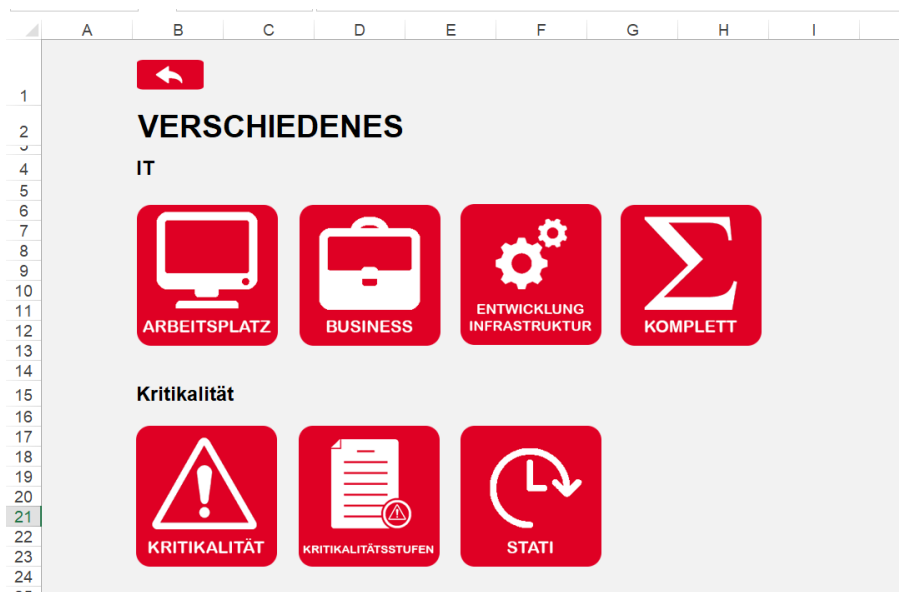
Die Ressourcenplanung spielt in einer Business Impact Analyse eine wichtige Rolle. Von grosser Bedeutung ist die Ermittlung, welche Ressourcen über welchen erforderlichen Zeitrahmen verwendet werden können und welchen Einfluss dies auf die Krisensituationen ausübt (Hiles, 2002, S. 15). Aus diesem Grund fügte die Schweizerische Mobiliar deren IT Ressource zusätzlich in die Datei «BIA-Datenaufnahme-Master.xlsm» ein. Diese Ressourcen werden in folgenden vier Unterkategorien aufgeteilt.

- Arbeitsplatz
- Business
- Entwicklung Infrastruktur
- IT Komplett

Auf dem BIA Menu befindet sich daher die Übersicht der Abbildung 8, damit jeder Anwender direkt die verschiedenen IT Ressourcen ansehen kann. Diese IT-Listen werden halbjährlich durch eine neue Liste der IT-Abteilung ersetzt. Das Ersetzen der aktuellen IT-Listen muss vom Abteilungsleiter durchgeführt werden.<sup>6</sup>

Für die Anpassung der Kritikalität wird lediglich im BIA Menu ein Link zum Register angezeigt. Die tatsächlichen Anpassungen müssen die Mitarbeiter in der jeweiligen Tabelle durchführen.

Abbildung 8: BIA IT und Kritikalität im bisherigen System



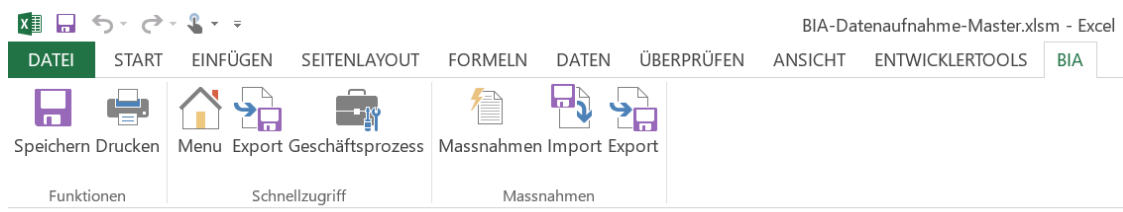
(Eigene Darstellung)

<sup>6</sup> Gemäss Meeting mit Abteilungsleiter BCM Herr In-Albon Damian (06.05.2019)

## BIA Navigation Menu

Um die Arbeitsschritte zu vereinfachen, kann ein selbst erstelltes Excel Menu verwendet werden, sogenanntes Ribbon Menu. Gemäss AGI Creative Team wird dieses Menu als eine Funktionsleiste verwendet, die eingesetzt werden kann, um Excel-Funktionalitäten, wie speichern oder kopieren, auszuführen (2011, S. 20). Die Schweizerische Mobiliar besitzt für das BIA-System ein eigenes Menu und ist deshalb nicht als ein Standard Ribbon Menu im System hinterlegt. Aus diesem Grund muss jeder Mitarbeiter dieses Menu, welches in der nachfolgenden Abbildung 9 dargestellt ist, auf seinem Computer eigenständig importieren. Das Ribbon Menu hilft den Anwendern beim Erstellen einer Massnahme für einen bestimmten Prozess, Drucken mit einer sauberen Ansicht eines Geschäftsprozesses, Schnellzugriff auf die drei Excel-Dateien, Schnellzugriff auf das BIA Menu und beim Importieren oder Exportieren der Excel-Datei oder Massnahmen. Zu jedem Geschäftsprozess kann eine neue Massnahme erstellt oder importiert werden. Der Export der Massnahmen des bezüglichen Geschäftsprozesses dient dem Austausch der Massnahmen mit externen Mitarbeitern.

Abbildung 9: BIA Ribbon Menu im bisherigen System



(Eigene Darstellung)

## BIA Reporting

Alle Analysen und Reportings, welche die Schweizerische Mobiliar in ihren Business Impact Analysen verwenden, werden in der Excel-Datei «Reporting.xlsx» hinterlegt. Da bereits 40 Reportings hinterlegt sind, kann ein Schnellzugriff Menu in dieser Datei benutzt werden. Dieses Menu enthält alle relevanten Reportings als eine Verlinkung, welche am meisten von den Mitarbeitern verwendet werden.<sup>7</sup> Die Abbildung 10 zeigt das Reporting Menu. Zudem besitzt dieses Menu den direkten Zugriff auf die anderen zwei Excel-Dateien.

<sup>7</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr In-Albon Damian (2019)

Abbildung 10: BIA Reporting im bisherigen System



(Eigene Darstellung)

## **2. IST-Analyse – BIA System**

Damit aus dem bestehenden System ein effizienter Prototyp entwickelt werden kann, wird zunächst eine komplexe Analyse über das BIA System durchgeführt. So werden alle Stärken und Schwächen des Systems aufgezeigt und können daher verbessert oder übernommen werden. Als Vorgehensweise werden die unterschiedlichen Prozessabläufe auf deren User Experience (UX), Usability und Fehlerquellen analysiert.

### **2.1. User Experience und Usability BIA System**

Jedes Zusammenspiel zwischen System, Produkt und einem Benutzer kann aus Sicht der Benutzerfreundlichkeit bewertet werden. Mithilfe von User Experience Analysen, werden die Reaktionen der Benutzer betrachtet, wenn sie mit dem System interagieren. Zusätzlich wird die Usability, also die Benutzerfreundlichkeit, eines Systems analysiert d.h. ob ein Benutzer fähig ist, das zur Verfügung gestellte System richtig zu benutzen. Die Anwender eines Systems sollten mit dem Programm so umgehen, wie dies von den Entwicklern vorgesehen ist. Daher ist wichtig, dass das System einwandfrei funktioniert und nicht verwirrend für die Anwender ist (Tullis & Albert, 2013, S. 4-6).

### **Analyse der Navigation-Menus**

Das BIA Menu sowie das Reporting Menu sind schlicht gehalten und übersichtlich. Mithilfe der Schnellzugriff Navigation ermöglicht das System zwischen den verschiedenen Dateien zu wechseln. Dies ist sehr vorteilhaft für die Anwender, da sie die Dateien nicht manuell öffnen müssen. Die Geschäftsprozess-Funktionalitäten und die Navigation der Reportings sind für die Benutzer des Systems sehr zuvorkommend. Die Anwender müssen nicht im Excel nach dem gewünschten Register suchen, sondern können direkt über das Hilfsmenu die Funktion ausführen.

Für das Ribbon Menu müssen sich die Benutzer mit dem System besser auskennen, da dieses Hilfsmenu eigenständig über die Excel Einstellungen importiert werden muss. Die einzelnen Prozessschritte, die das Menu bietet, sind zudem nicht direkt erkennbar und weisen zum Teil Laufzeitfehler beim Gebrauch auf.

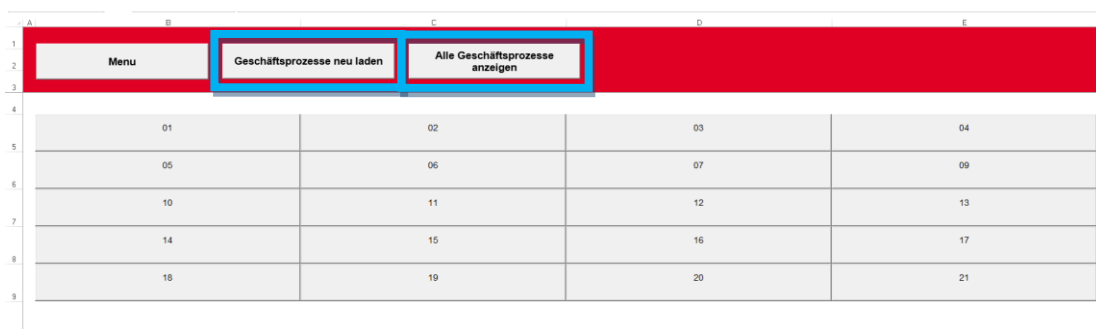
## Analyse des Arbeitsschrittes Geschäftsprozess hinzufügen

Im bisherigen BIA System wurde kein automatischer Workflow für das Hinzufügen eines Geschäftsprozesses implementiert. Die Mitarbeiter müssen viele Schritte vornehmen, um einen neuen Geschäftsprozess in das System zu integrieren. Das Kapitel «1.3 BIA – bisheriges System» erläutert diesen Vorgang. Bei der Durchführung dieser manuellen Arbeitsschritten tauchen viele potenzielle Fehlerquellen auf. Diese möglichen Fehlerquellen beziehen sich auf das Überspringen eines Arbeitsschrittes, das Eintragen des Geschäftsprozesses an der falschen Position, das Format der Tabelle nicht übernehmen oder sogar die Generierung von redundanten Daten. Infolgedessen müssen diese potenziellen Fehlerquellen im Prototypen beseitigt werden.

## Analyse des Arbeitsschrittes Geschäftsprozess anzeigen

Die Anwender können das Register des gewünschten Geschäftsprozesses in der Registerleiste aufrufen oder das BIA Menu verwenden. Um die neusten Prozessdaten zu erhalten, müssen die Anwender die Ansicht immer neu laden. In diesem Arbeitsschritt besteht der wesentliche Nachteil beim Einfügen eines neuen Geschäftsprozesses. Wenn der Benutzer diesen nicht ordnungsgemäss hinzufügt, besteht die Problematik, dass einzelne Prozesse in der Geschäftsprozessübersicht falsch, sinnlos oder sogar überhaupt nicht angezeigt werden.

Abbildung 11: BIA Geschäftsprozess anzeigen im bisherigen System



1	Menu			
2	Geschäftsprozesse neu laden			
3	Alle Geschäftsprozesse anzeigen			
4	01	02	03	04
5	05	06	07	09
6	10	11	12	13
7	14	15	16	17
8	18	19	20	21

(Eigene Darstellung)

Nachdem ein Anwender einen Prozess ausgewählt hat, wird die Geschäftsprozessliste gefiltert nach seiner verlangten Kategorie angezeigt. Die Problematik liegt jedoch darin, dass ein anderer Benutzer diese Ansicht neu laden muss, damit wieder alle Geschäftsprozesse aufgelistet werden. Um einen Geschäftsprozess neu zu laden oder alle Geschäftsprozesse erneut auflisten zu lassen, können die blau-markierten Navigationskomponenten in der Abbildung 11 verwendet werden.

## **Analyse des Arbeitsschrittes Geschäftsprozess suchen**

Die Suche eines bestimmten Geschäftsprozesses ist sehr effizient. Der eingegebene Suchtext inspiziert die Namen aller Geschäftsprozesse, welche diese Zeichenkette enthalten und listet diese auf. Beim Hinzufügen eines neuen Geschäftsprozesses muss der Anwender zusätzlich den Namen im Inhaltsverzeichnis nachführen. Ansonsten wird dieser Geschäftsprozess nicht durchsucht. Erst nach dem Einfügen in die zusätzliche Liste, wird der neue Geschäftsprozess auch auf der Suchliste angezeigt werden. Dies ist eine potenzielle Fehlerquelle, da der Benutzer den Prozess fehlerhaft eintragen oder diesen Arbeitsschritt sogar überspringen kann.

## **Analyse des Arbeitsschrittes Geschäftsprozess bearbeiten**

Über das Auswahlfeld kann der Geschäftsprozess ausgesucht werden, um die Daten zu verändern. Die bestehenden Daten des Geschäftsprozesses werden in die Felder geladen und können nach dem Anpassen gespeichert werden. Da die Schweizerische Mobiliar bereits 92 Geschäftsprozesse im System hinterlegt haben, ist das Auswahlfeld eine ziemlich unpassende Komponente. Die Benutzer haben die Schwierigkeit, den gewünschten Prozess in der Dropdown-Liste zu finden. Zudem besteht das Problem, dass der Benutzer plötzlich einen falschen Geschäftsprozess bearbeitet. Ein weiterer wesentlicher Nachteil liegt darin, dass die Eingabefelder nicht auf deren Korrektheit validiert werden und so die Gefahr von ungültigen Daten im System besteht.

### **2.2. Analyse der Reportings erstellen**

Aus den Daten der Hauptdatei «BIA-Datenaufnahme-Master.xlsm» werden die zahlreichen Reportings erstellt. Diese werden in die Datei «Reprting.xlsx» übernommen, damit die Mitarbeiter diese analysieren können. Das Problem besteht darin, dass einzelne Reportings einen falschen Verweis auf die Stammdaten der Hauptdatei besitzen. Aus diesem Grund werden einzelne Analysen überhaupt nicht generiert oder falsch angezeigt. Im Sitzungsprotokoll vom 06.05.2019, welches im ANHANG XI hinterlegt ist, werden alle relevanten Reportings aufgelistet, welche die Schweizerische Mobiliar ins neues System übernehmen möchten. Diese Reportings werden nachfolgend einzeln analysiert.

## Reporting «Gesamtübersicht kompakt»

Abbildung 12: Reporting Gesamtübersicht kompakt

Übersicht Kritikalität der Geschäftsprozess												
P-Nr.	Prozess-Bezeichnung	GLB	Verantwortlich	1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	> 1 Monat
05.a	Service Product	GLB VOR	M. Muster	1	1	1	1	1	2	3	4	5
06.2a	Fakturierung mon.	GLB MM	P. Fux	1	1	1	1	2	2	2	3	3

(Eigene Darstellung)

Die Abbildung 12 zeigt die Gesamtübersicht Reporting der Schweizerischen Mobiliar. Diese Analyse dient als Übersicht aller existierenden Geschäftsprozessen mit den dazugehörigen Gesamtkritikalitäten. Wie bereits im Kapitel «1.2 Business Impact Analysis» erwähnt, stellt sich die Gesamtkritikalität aus mehreren finanziellen und nicht-finanziellen Auswirkungen zusammen. Diese Auflistung der zeitkritischen Prozessen verwenden die Mitarbeiter der Abteilung BCM am häufigsten.<sup>8</sup> Für jeden Geschäftsprozess wird in dieser Analyse zusätzlich noch die verantwortliche Person und der Geschäftsleitungsbereich (GLB) angegeben.

## Reporting «Übersicht Kritikalität nach Kategorie»

Abbildung 13: Reporting Übersicht Kritikalität nach Kategorie

Übersicht Geschäftsprozess nach Kritikalitäts-Kategorie										Kat. A			Kat. B		Kat. C	
P-Nr.	Prozess-Bezeichnung	Prozess-Verantwortliche	Krit.-Kat.	GLB	1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	> 1 Monat			
<b>Kategorie A - Extrem zeitkritische Prozesse</b>																
2.3	Sicherheit	M. Muster	A	GLB VOR	1	2	2	3	4	4	4	4	5			
10.a	Kommunikation	P. Fux	A	GLB FIN	1	1	2	3	4	4	4	4	4			
<b>Kategorie B - Zeitkritische Prozesse</b>																
5.3	Service Product	P. Furrer	B	GLB VOR	1	1	2	3	3	4	4	4	5			
11.5	Management	M. Muster	B	GLB VER	1	1	1	2	2	3	4	4	5			
<b>Kategorie C - Nicht zeitkritische Prozesse</b>																
03.a	Kommunikation	I. Muster	C	GLB FIN	1	1	1	1	2	2	3	4	5			

(Eigene Darstellung)

Dieses Reporting ist eine ähnliche Analyse wie beim Reporting «Gesamtübersicht kompakt» mit dem Unterschied, dass diese Ansicht noch zusätzlich nach deren Kategorien unterteilt ist. Die Geschäftsprozesse werden in die drei Kategorien A, B und C aufgeteilt, wie dies in der Abbildung 13 durch die blauen Trennlinien angezeigt wird. Prozesse die bereits innerhalb von zwei Tagen die Auswirkungsklasse «gross» erreicht haben, werden in der Kategorie A unterteilt. Geschäftsprozesse die diese Auswirkungsklasse zwischen zwei Tagen und zwei Wochen erreichen, werden in der Kategorie B eingeteilt.

<sup>8</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr In-Albon Damian (2019)

Geschäftsprozesse, die erst nach einem Monat grosse finanziellen oder nicht-finanzielle Auswirkungen vorweisen, werden in der letzten Kategorie C klassifiziert. Diese Analyse unterstützt die Mitarbeiter mit der Evaluierung der Anzahl Geschäftsprozesse in den jeweiligen Kategorien.

## Reporting «Arbeitsplatzanforderungen nach Kritikalitäts-Kategorie»

Abbildung 14: Reporting Arbeitsplätze Kritikalitäts-Kategorie

Arbeitsplatzanforderungen nach Kritikalitäts-Kategorie				Anz. Arbeitsplätze für Minimalbetrieb										Hauptstandort				
P-Nr	Prozess-Bezeichnung	Krit. - Kat.	GLB	AP normal	1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	1 Monat	BG	MB	NY	ZH	
02.a	Service	A	GLB VOR	100	15	25	25	30	40	75	100	100	100	x	x		x	
5.3	Sicherheit	A	GLB VER	24		6	6	7	8	10	20	20	24			x		
Total				123	15	31	31	37	48	85	120	120	124					
10	Kommunikation	B	GLB FIN	13			2	2	3	3	3	7	13	x		x		
11.a	Management	B	GLB VOR	0													x	
Total				13	0	0	2	2	3	3	3	7	13					
5.8	Vertragsdel.	C	GLB FIN	22			5	5	5	7	15	20	22	x				

(Eigene Darstellung)

Anhand des «Arbeitsplatzanforderungen nach Kritikalitäts-Kategorie» Reporting können die Mitarbeiter inspizieren, in welcher bestimmten Zeit welche Menge an Arbeitsplätze zusätzlich angefordert werden müssen. Diese Geschäftsprozessliste wird zudem nach den Kritikalitäts-Kategorien gruppiert und die Menge an Arbeitsplätze summiert. Wie in der Abbildung 14 ersichtlich ist, wird für jeden Geschäftsprozess die verlangten Standorte durch ein «X» markiert.

## Reporting «Lieferantenliste Kat»

Abbildung 15: Reporting Lieferantenliste

Lieferantenliste nach Kritikalitäts-Kategorie																
P-Nr.	Prozess-Bezeichnung	GLB	Krit.-Kat.	Lieferant	Kritikalität	Alternativlieferant	Bemerkung	1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	> 1 Monat
<b>Kategorie A - Extrem zeitkritische Prozesse</b>																
02.a	Service Product	GLB MM	A	BAR Informatik	7. 2 Wochen	Occom	Netzwerk & Support	1	1	1	2	2	3	4	4	5
5.8	Telefonie	GLB IT	A	Swisscom AG	2. 4 Std.	Keine	Telefonie	3	4	5	5	5	5	5	5	5
<b>Kategorie B - Zeitkritische Prozesse</b>																
06.4a	Haustechnik	GLB VOR	B	VER Bern	2. 4 Std.	Keine		3	4	5	5	5	5	5	5	5
08.a	Sicherheit System	GLB IT	B	Schweizerische Bund	2. 4 Std.	Drittanbieter	Security	3	4	5	5	5	5	5	5	5
<b>Kategorie C - Nicht zeitkritische Prozesse</b>																
8.3	Sicherheit Mitarbeiter	GLB MM	C	Securitas	6. 1 Woche	Keine		1	1	1	2	3	4	4	4	5

(Eigene Darstellung)

In dem Reporting «Lieferantenliste Kat» werden alle Geschäftsprozesse mit deren Kritikalität gruppiert und aufgelistet. Zusätzlich wird in dieser Analyse noch der Lieferant des Prozesses angegeben. Falls der Hauptlieferant ausfällt, hat die Schweizerische Mobiliar für die meisten Geschäftsprozesse noch einen Alternativlieferant definiert. Für den Alternativlieferant wird noch die Kritikalität angegeben, wann dieser spätestens liefern müsste, damit keine gravierende finanzielle sowohl auch nicht-finanzielle Auswirkungen

auftreten. In der Abbildung 15 wird auf der linken Seite die Informationen der Lieferanten angegeben und im rechten Teil die Kritikalität über die bestimmte Zeitspanne.

## Reporting «Geschäftsprozesse mit kritischen Funktionen»

Abbildung 16: Reporting Krit. Funktionen Kat

Geschä	prozesse mit kritischen Funktionen	Krit-Kat	GLB	Prozessverantwortlich	Betriebswichtige Funktionen Funktion	Schlüsselperson
P-Nr.	Prozess-Bezeichnung					
<b>Kategorie A - Extrem zeitkritische Prozesse</b>						
2.4	Service Product	A	GLB VOR	Max Muster	Leiter BCM	Helge Furrer
05.d	Sicherheit	A	GLB VOR	Max Fux	Service Manager	Beat Wasmer
<b>Kategorie B - Zeitkritische Prozesse</b>						
5.3	Marketing	B	GLB VOR	Martin Schwery	SAP Verantwortlicher	Roland Martig
11.2	Kommunikation	B	GLB VOR	Michael Brigeler	Product Owner	Max Muster
<b>Kategorie C - Nicht zeitkritische Prozesse</b>						
6	Empfang Nyon	C	GLB AM	Peter Reich	Telefonzentrale	Reto Maier

(Eigene Darstellung)

Die Abbildung 16 zeigt das Reporting «Geschäftsprozesse mit kritischen Funktionen», bei welcher alle Geschäftsprozesse mit kritischen Funktionen aufgelistet werden. Diese Geschäftsprozessliste wird zusätzlich nach deren Kritikalität gruppiert. Für jeden Geschäftsprozess wird der Prozessverantwortliche angegeben sowie deren betriebswichtige Funktionen und Schlüsselpersonen. Die Mitarbeiter brauchen diese Analyse, um die zuständige Person und deren Funktionalität herauszulesen.

## Analyse neue IT Anwendungen importieren

Halbjährlich erhält die Abteilung BCM von der IT Abteilung die neue IT-Liste in Form einer separaten Excel-Datei. Der Nachteil dieses Arbeitsschrittes liegt darin, dass der Abteilungsleiter, diese IT-Liste manuell in das BIA System einfügen muss. Dabei wird die komplette Liste zuerst für das bestimmte Register gefiltert und kopiert, damit diese anschliessend im BIA System die bestehenden Daten überschreiben kann. Die Problematik besteht darin, dass die IT-Liste der IT-Abteilung nicht die gleiche Formatvorlage verwendet wie das BIA System. Daher muss der Abteilungsleiter die Liste in die richtige Formatvorlage umwandeln, damit diese erfolgreich ins BIA System integriert werden kann.

### 2.3. Analyse Teamwork

Da das System eine Excel basierende Lösung ist, kann an diesen Dateien nicht gleichzeitig gearbeitet werden. Trotzdem werden die Dateien von der Abteilung BCM auf der Cloud abgespeichert, um die Dokumente auszutauschen. Jedoch können diese Excel-

Dateien nicht durch die Mitarbeiter auf der Cloud geöffnet werden, da diese durch die Verknüpfungen der Daten erhebliche Fehler aufweisen.<sup>9</sup>

Ein wesentlicher Nachteil an der Zusammenarbeit mit anderen Mitarbeitern liegt darin, dass sich die drei Dokumente immer im gleichen Verzeichnis befinden müssen. Wenn diese drei Excel-Dateien jeweils von einem anderen Verzeichnis geöffnet werden, sind beim Reporting und den Massnahmen einen falschen Bezug den Stammdaten hinterlegt. Diese Beeinträchtigung senkt die Zusammenarbeit mit den Mitarbeitern rapide.

## **2.4. Analyse verwendete Technologien**

Das BIA System wird in drei unterschiedliche Excel-Dateien aufgeteilt. Zwei davon verwenden das Standard-Excelformat und die Hauptdatei das «.xlsm» Format. Durch die Verwendung von Makros muss dieses Format angewendet werden. Eine Gruppierung von mehreren Anweisungen, welche zur gleichen Zeit ausgeführt werden können, werden als Makro zusammengefasst (Sielhorst & Wilhelm, 1992, S. 7).

### **Makros**

Das bestehende BIA System verwendet mehrere Makros, um unterschiedliche Funktionalitäten auszuführen. Zum Erstellen der Reportings wird ein Makro ausgeführt, damit alle relevanten Daten in eine einzelne Tabelle zusammengeführt werden. Aus diesem Register erhält die Datei «Reporting.xlsx» die benötigten Informationen. Das BIA Menu sowie das Ribbon Menu führen mehrere Makros aus, um die Geschäftsprozesse zu filtern oder ein Geschäftsprozess auszudrucken.

## **2.5. Analyse Sicherheit**

Die Daten des BIA System sind vertrauliche Geschäftsdaten, welche nicht an unbefugte Personen übergeben werden sollten. Infolgedessen wird die Sicherheit der bisherigen Lösung analysiert. Zunächst werden die Versionshaltung und Datensicherung der Excel-Dateien betrachtet. Daraufhin wird das System anhand der Berechtigungen begutachtet.

---

<sup>9</sup> Gemäss mündlicher Mitteilung von Leiter Abteilung BCM Herr In-Albon Damian (2019)

## Datensicherung und Versionshaltung

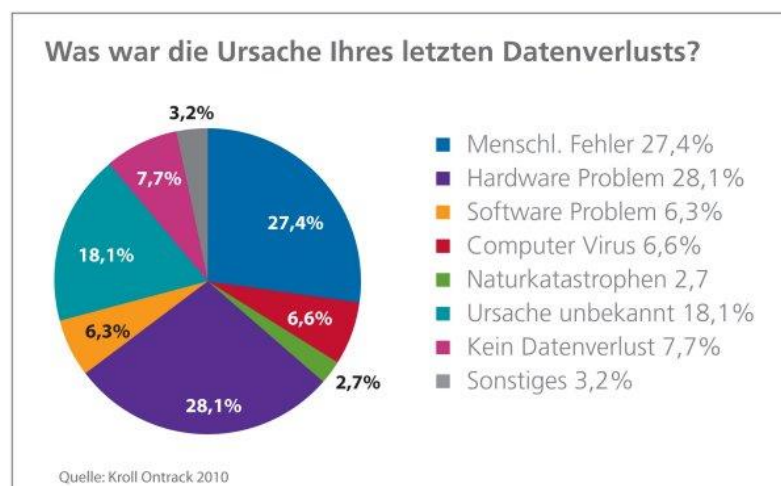
Die Abteilung BCM verfügt in der bisherigen Lösung kein spezifisches Datensicherungskonzept. Wenn ein Mitarbeiter eine Änderung am System vornehmen möchte, muss er beim Abteilungsleiter den USB-Stick verlangen, der das aktuelle System enthält. Nach den Anpassungen der Daten wird dieser USB-Stick an den Abteilungsleiter übergeben. Als ein Backup wird anschliessend jeweils eine Version des Systems im Filehosting-Dienst OneDrive von Microsoft hochgeladen.

Bei dieser Vorgehensweise gewährleisten die Mitarbeiter eine optimale Datensicherung, damit die persönlichen und vertraulichen Daten nicht abhanden kommen. Jedoch erschwert der ständige Austausch des USB-Sticks die produktive Zusammenarbeit der Mitarbeiter. Angenommen ein Mitarbeiter möchte von einem anderen Standort aus arbeiten, muss dieser zuvor die aktuellen Daten sicherstellen.

## Benutzerberechtigung

Das bisherige BIA System verfügt keine Benutzerberechtigung. Jeder der den USB-Stick erhält, ist fähig alle Daten im System zu ändern. Bei komplexeren und komplizierteren technischen Systemen sind ungefähr 60 bis 80 Prozent aller Missgeschicke durch menschliche Fehler zu verantworten (Prof. Dr.-Ing. Johannsen, 1993, S. 387). Wenn nun diesbezüglich das menschliche Versagen als Ursache für den Datenverlust beim Arbeiten an unterschiedlichen Systemen betrachtet, liegt der Grund für den Verlust an den Daten in vielen Fällen am Menschen selber. Um dies zu hinterlegen, hat das Unternehmen Kroll Ontrack 2010 die Studie auf der Abbildung 17 aufgestellt (Kurzlechner, 2010).

Abbildung 17: Datenverlust Kroll Ontrack 2010



Quelle: (ChannelPartner, 2010)

Da jeder Mitarbeiter Zugriff auf alle Funktionalitäten im bisherigen System besitzt und viele Arbeitsschritte grosse menschliche Fehlerquellen aufweisen, sollten im Prototypen diese potenziellen Fehlerquellen beseitigt werden.

Dieses Ziel kann durch Automatisierung von Arbeitsschritten und durch Benutzerberechtigungen erstrebt werden. Damit das System agiler und effizienter wird, müssen die Arbeitsschritte sowie die entwickelten Prozesse automatisiert werden. Aus diesem Grund muss entschieden werden, welche Arbeitsabläufe vorteilhaft automatisiert werden und welche Schritte weiterhin manuell von den Mitarbeitern, welche das benötigte Wissen vorweisen, durchgeführt werden sollen (Abolhassan & Kellermann, 2016, S. 9-10).

## 2.6. SWOT Analyse

Die SWOT Analyse wird als eine Strategiemethode in einem Unternehmen angewendet. Diese spezifische Analyse hilft einem Unternehmen die Ausgangslage zu betrachten, um daraus erfolgreiche Entscheidungen für die Zukunft treffen zu können. Das Akronym SWOT entsteht aus **S**trengths (Stärken), **W**eaknesses (Schwächen), **O**pportunities (Chancen) und **T**hreats (Risiken). Diese Analyse hilft dem Unternehmen die Stärken und Schwächen des bestehenden Systems aufzuzeigen, um daraus strategische Schlussfolgerungen ziehen zu können. Nach der Aufstellung der Stärken, Schwächen, Chancen und Risiken muss festgelegt werden, wie die Stärken eingesetzt werden müssen, um die Chancen effizient zu nutzen und die Risiken bestmöglich zu minimieren. Zudem muss definiert werden wie die Schwächen minimiert werden können, um so die Chancen besser nutzen zu können und die Risiken zu überwinden. In der Tabelle 2 wird eine solche SWOT Portfolio-Analyse bezüglich des bisherigen Systems durchgeführt (Dr. Künzli, 2012, S. 126-129).

Tabelle 2: SWOT Portfolio-Analyse

<b>SWOT</b> Portfolio-Analyse	<b>Chancen</b> - Erfahrene Mitarbeiter mit dem System	<b>Risiken</b> - Hohe Fehlerquelle - Keine technische Sicherheit
<b>Stärken</b> - Filterfunktion von Excel - Übersichtliche Reportings - Grosses Verbesserung Potenzial	Im neuen System die Geschäftsprozesse als Excel Export Datei zur Verfügung stellen.	Durch ein Login die Sicherheit im System gewährleisten
<b>Schwächen</b> - Kein Multiusersystem - Zu komplex - Grosse Know-How nötig - Sehr hoher Wartungsaufwand	Die Applikation benutzerfreundlicher machen	Durch eine saubere Softwarelösung die Fehlerquellen beheben

(Eigene Darstellung)

## 2.7. Risikoanalyse

In dieser Abhandlung wird eine Risikoanalyse durchgeführt, um die potenzielle Problemquellen des bisherigen Systems aufzuzeigen. Diese Fehlerquellen werden folglich im Prototyp berücksichtigt.

Eine Risikoanalyse ist ein bedeutender Bestandteil des Sicherheitskonzeptes. In einem Sicherheitskonzept wird ein Plan mit unterschiedlichen Massnahmen zur Berücksichtigung von analysierten potenziellen Risiken aufgestellt (Pongratz, 1996, S. 229-235). Um eine Risikoanalyse durchzuführen, muss der Bereich definiert werden, welcher analysiert wird. Anschliessend werden alle potenziellen Risiken diesbezüglich notiert. Damit die Analyse auch bewertet werden kann, muss für jede relevante Gefahr die Bedeutung der Auswirkung evaluiert werden. Anhand der Eintrittswahrscheinlichkeit und des Schweregrades der Auswirkung kann das Risiko bewertet werden. Zum Erhalten der Risikoexposition werden diese zwei Kenngrössen multipliziert. Schlussendlich werden die Ergebnisse durch eine visuelle Darstellung verstärkt, damit dies als Entscheidungsvorlage dienen kann (Paulsen, 2003, S. 10-17).

## Risk Analysis Table (RAT)

Gemäss einer «Risk Analysis Table» wird die Risikoerkennung und die Risikobewertung, welche im vorherigen Abschnitt beschrieben werden, visuell dargestellt. Die Eintrittswahrscheinlichkeit und der Schweregrad der Auswirkung werden anhand von Zahlen zwischen eins und zehn repräsentiert. Die «eins» hat eine geringe Auswirkung sowie Eintrittswahrscheinlichkeit und die «zehn» hat eine fatale Auswirkung. Je grösser die Risikoexposition ist, umso wichtiger ist es, dieses Risiko für den Prototypen zu berücksichtigen. Die Tabelle 3 listet 13 potenzielle Probleme auf und evaluiert diese.

Tabelle 3: Risk Analysis Table

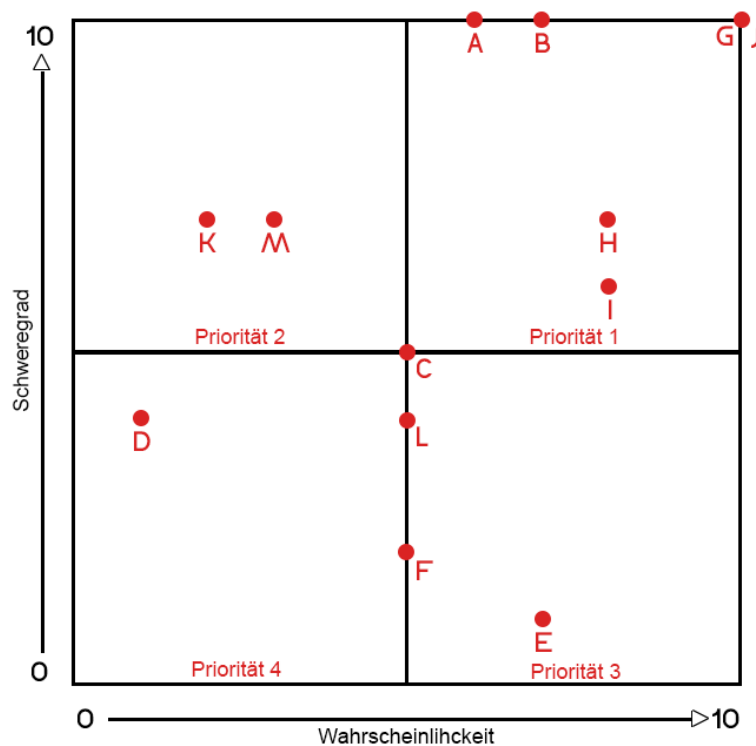
ID	Potenzielles Problem	Eintrittswahrscheinlichkeit	Schweregrad Auswirkung	Risikoexposition
A	Mehrere Mitarbeiter arbeiten zur gleichen Zeit an zwei unterschiedlichen Versionen des bisherigen Systems	6	10	60
B	Die drei Dateien werden nicht vom gleichen Verzeichnis aus gestartet	7	10	70
C	System zeigt einen Referenzfehler an	5	5	25
D	«Geschäftsprozess bearbeiten» weist einen Fehler beim Speichern auf	1	4	4
E	«Geschäftsprozess anzeigen» zeigt nicht jede Gruppierung der Geschäftsprozesse an	7	1	7
F	«Geschäftsprozess suchen» referenziert nicht auf das dazugehörige Register	5	2	10
G	Neuer Geschäftsprozess wird falsch hinzugefügt	10	10	100
H	Redundanzen werden von einem Mitarbeiter erstellt	8	7	56
I	Funktionalitäten des Navigationsmenu (Ribbon Menu) funktionieren nicht einwandfrei	8	6	48
J	Reporting weisen Bezugsfehler auf	10	10	100
K	Massnahmen werden nicht hinzugefügt	2	7	14
L	Massnahmen können nicht importiert werden	5	4	20
M	Massnahmen können nicht exportiert werden	3	7	21

(Eigene Darstellung)

## Risk Matrix Analysis (RMA)

Die Ergebnisse der Risikoanalyse der Tabelle 3 werden durch eine Risiko Matrix Analyse in einem Diagramm dargestellt. Jedes potenzielle Risiko wird durch ein Punkt in das Diagramm der Abbildung 18 eingefügt. Probleme, welche sich in der Priorität eins befinden, sollten im Prototypen verbessert werden. Alle Risiken innerhalb dieser Priorität weisen eine hohe Eintrittswahrscheinlichkeit und auch einen hohen Schweregrad auf. In der Abbildung 18 werden alle potenzielle Risiken der vorherigen RAT-Tabelle visuell dargestellt.

Abbildung 18: Risk Matrix Analysis



(Eigene Darstellung)

### 3. Anforderungen der Schweizerischen Mobiliar

Die Abteilung Business Continuity Management (BCM) der Schweizerischen Mobiliar wünscht sich eine effizientere Lösung, um die Arbeitsschritte an den Prozessen und den Einflussanalysen zu verbessern. Diese Erhöhung der Produktivität soll durch das Automatisieren von gewissen Arbeitsschritten und durch die Zusammenarbeit am System erzielt werden. Die Anforderungen an das neue System liegen darin, dass der Prototyp die produktive Zusammenarbeit der Mitarbeiter stärkt. Dies impliziert, dass die Nachteile des bestehenden Systems behoben werden müssen. Da in der aktuellen Lösung alle Arbeitsschritte sehr komplex aufgebaut sind und hohe Fehlerquellen aufweisen, sollten diese Funktionalitäten im Prototypen angepasst werden. Durch einfachere Arbeitsschritte benötigen die Mitarbeiter kein umfassendes Wissen, um das System zu bedienen und dies erzielt auch eine einfachere Wartung der Geschäftsprozesse. Eine zentrale Anforderung besteht darin, dass das neue System Multiuser fähig sein muss. Durch die Authentifizierung des Benutzers sollte dieses Vorhaben erreicht werden. Des Weiteren wird durch die Authentifizierung auch noch die Berechtigung der Anwender abgedeckt. Ein Administrator soll zudem noch Workflows für externe Benutzer erstellen können, damit denen für einen bestimmten Zeitraum Zugriff auf ausgewählte Geschäftsprozesse gewährleistet werden können. Im aktuellen System exportieren die Mitarbeiter den einzelnen Prozess in eine separate Excel Tabelle und senden diesen dem externen Benutzer zu. Im Anschluss muss schliesslich ein Mitarbeiter den Prozess wieder in das Hauptsystem integrieren.<sup>10</sup>

#### 3.1. Bestehende Software als Ersatz

Die Schweizerische Mobiliar gab bereits eine Bachelorarbeit auf diesem Gebiet in Auftrag. Die Aufgabe der Arbeit bestand darin, potenzielle bestehende Software zu analysieren, welche als Ersatz des BIA-Systems verwendet werden könnte. Anhand der Anforderungen, welche das BIA-System liefern muss, wurden in dieser Bachelorarbeit insgesamt 424 Produkte evaluiert. Jedoch haben nur sechs dieser Anbieter die gewünschten Anforderungen erfüllt (Künzi, 2018, S. 52-55).

Die Schweizerische Mobiliar hat ein Produkt ausführlich getestet und auch wesentliche Inhalte zum Design des BIA-Systems bei dem Produkt beigetragen. Das getestete Produkt «alive-IT Business Continuity Framework» wird von der deutschen Unternehmung «Controllit AG» zur Verfügung gestellt. Das BIA-Modul wurde von der Schweizerischen Mobiliar

<sup>10</sup> Gemäss Meeting mit Abteilungsleiter BCM Herr In-Albon Damian (06.05.2019)

ausführlich analysiert, jedoch wurde dieses nicht als Ersatz des bisherigen Systems übernommen, obwohl das Modul grossmehrheitlich die Anforderungen abdecken würde. Die Problematik lag darin, dass das BIA-Modul nicht einzeln gekauft werden kann, sondern das grosse Gesamtpaket gekauft werden muss. Dieses Gesamtpaket weist viele Einzelmodule auf, welche von der Schweizerischen Mobiliar nicht benötigt werden. Aufgrund dem zu grossen Funktionsumfang des Produkts und der wesentliche Nachteil, dass das BIA-Modul nicht individuell an die Schweizerische Mobiliar angepasst werden konnte, wurde dieses Produkt ausgeschlossen.<sup>11</sup>

In Anbetracht, der 424 analysierten Produkten und aus deren sechs potenzielle Anbieter das erfolgversprechendste Produkt ausgeschlossen wurde, bietet es sich an, der Schweizerischen Mobiliar ein selbstentwickeltes Programm vorzustellen. Diese Software sollte allen Anforderungen des Auftraggebers gerecht werden. Im Kapitel «8.3. Fortführung neues BIA System» wird die Eigenentwicklung mit der Software «alive-IT Business Continuity Framework» gegenübergestellt. Dieser Vergleich unterstützt die Schweizerische Mobiliar bei der Entscheidung, ob die Weiterführung des Endprodukts dieser Bachelorarbeit sinnvoll ist.

Damit der Prototyp optimal für die Benutzer umgesetzt werden kann, werden im folgenden Abschnitt drei Personas vorgestellt. Diese Personas repräsentieren die unterschiedlichen Mitarbeiter die anschliessend am System aktiv arbeiten werden. Die folgende Auflistung unterstützt die Entscheidung der Auswahl der potenziellen Umsetzungstechnologie.

### **3.2. Personas**

Das neue System soll von drei verschiedenen Zielgruppen bedient werden. Darunter zählt ein aktiver Mitarbeiter der Abteilung BCM, ein Administrator, der zugleich auch ein Mitarbeiter der Abteilung BCM ist und zuletzt noch ein externer Mitarbeiter der Abteilung Human Resources (HR). In der Tabelle 4 wird für jede Zielgruppe eine Persona beschrieben.

Personas werden verwendet, um ein realistisches Bild eines Anwenders des Systems zu analysieren. Das Vorstellen der unterschiedlichen Personas verbessert das Verständnis der Anwenderbedürfnisse und stellt so eine einheitliche Sichtweise für alle Beteiligten am Projekt auf (Holt, Winter, & Thomaschewski, 2011, S. 40-41).

---

<sup>11</sup> Gemäss Meeting mit Abteilungsleiter BCM Herr In-Albon Damian (06.05.2019)

Tabelle 4: Vorstellung von drei Personas

<b>Persona: Mitarbeiter</b>	<b>Name: Daniel Marx</b> Alter: 24 Wohnort: Bern, Schweiz Familienstand: ledig Interessen/Hobbys: Tennis, Skifahren
	<b>Technische Fähigkeiten</b> <p>Daniel hat einen Bachelorabschluss in Betriebsökonomie und arbeitet mit seinem Computer am Arbeitsplatz der Schweizerischen Mobiliar im Standort Bern. Er besitzt sehr gute Kenntnisse, in Tabellenkalkulationsprogrammen und Textverarbeitungsprogrammen. Er ist bereit, neue Produkte auszutesten und zu verwenden.</p> <b>Aufgaben am BIA System</b> <p>Daniel ist verantwortlich, dass alle Geschäftsprozesse auf dem neusten Stand sind. Dies impliziert, dass er die Berechtigung besitzt, jeden Prozess bearbeiten zu können. Er kann einen neuen Prozess hinzufügen oder einen existierenden Prozess entfernen. Für jeden Prozess kann er die Massnahmen verwalten und die Kritikalität bestimmen. Mithilfe der Reportings kann Daniel die Geschäftsprozesse überblicken.</p>
<b>Persona: Administrator</b>	<b>Name: Martin Schmidt</b> Alter: 31 Wohnort: Thun, Schweiz Familienstand: verheiratet Interessen/Hobbys: Radfahren, Schwimmen
	<b>Technische Fähigkeiten</b> <p>Martin hat einen Masterabschluss in Business Engineering und arbeitet mit seinem Laptop am Arbeitsplatz der Schweizerischen Mobiliar im Standort Bern. Die Schweizerische Mobiliar stellt Martin den Laptop täglich zur Verfügung, damit er auch extern an dem System mitarbeiten kann. Martin besitzt durchschnittliche Kenntnisse bei der Verwendung von Tabellenkalkulationsprogrammen. Durch sein erlerntes Wissen besitzt er hervorragende Stärken, um die Projektleitung zu übernehmen.</p>

	<p><b>Aufgaben am BIA System</b></p> <p>Martin führt dieselben Aufgaben wie Daniel durch, da ein Administrator gleich wie ein Mitarbeiter geführt wird. Jedoch wird Martin zusätzlich im System als ein Administrator deklariert. Durch diese Berechtigung kann Martin über ein Admin Panel die Anwender eines Systems verwalten. Zudem kann er für externe Mitarbeiter einen Workflow generieren, damit diese für eine bestimmte Zeit für ausgewählte Prozesse Zugriff erhalten. Martin hat die Berechtigung anderen Mitarbeitern die Administratorrechte zu teilen, um Unterstützung zu bitten.</p>
<b>Persona: Externer Mitarbeiter</b>	<p><b>Name: Adeline Beaulieu</b> Alter: 27 Wohnort: Nyon, Schweiz Familienstand: ledig Interessen/Hobbys: Tanzen, Reisen</p>
	<p><b>Technische Fähigkeiten</b></p> <p>Adeline hat einen Bachelorabschluss in Business Administration und arbeitet mit ihrem Computer am Arbeitsplatz der Schweizerischen Mobiliar im Standort Nyon. Sie führt administrative Arbeiten durch und hat gute Kenntnisse für der Verwendung von Tabellenkalkulationsprogrammen und Textverarbeitungsprogrammen.</p> <p><b>Aufgaben am BIA System</b></p> <p>Adeline ist eine externe Mitarbeiterin, welche nicht in der Abteilung BCM tätig ist. Sie bestimmt jedoch für gewisse Prozesse die Schlüsselfunktionen und definiert die benötigten Lieferanten. Aus diesem Grund benötigt sie in einer bestimmten Zeit einen begrenzten Zugriff auf die spezifischen Prozessen, damit sie Anpassungen an dem Prozess und den Massnahmen vornehmen kann. Damit Adeline für eine bestimmte Zeitspanne Zugriff auf die Prozesse hat, erstellt der Administrator einen Workflow für sie. Sie kann anschliessend für eine gewisse Zeit nur die angeforderten Prozesse überarbeiten und ändern. Nach dem zeitlichen Ablauf des Workflows besitzt Adeline kein gültiges Anmeldekonto mehr.</p>

(Eigene Darstellung)

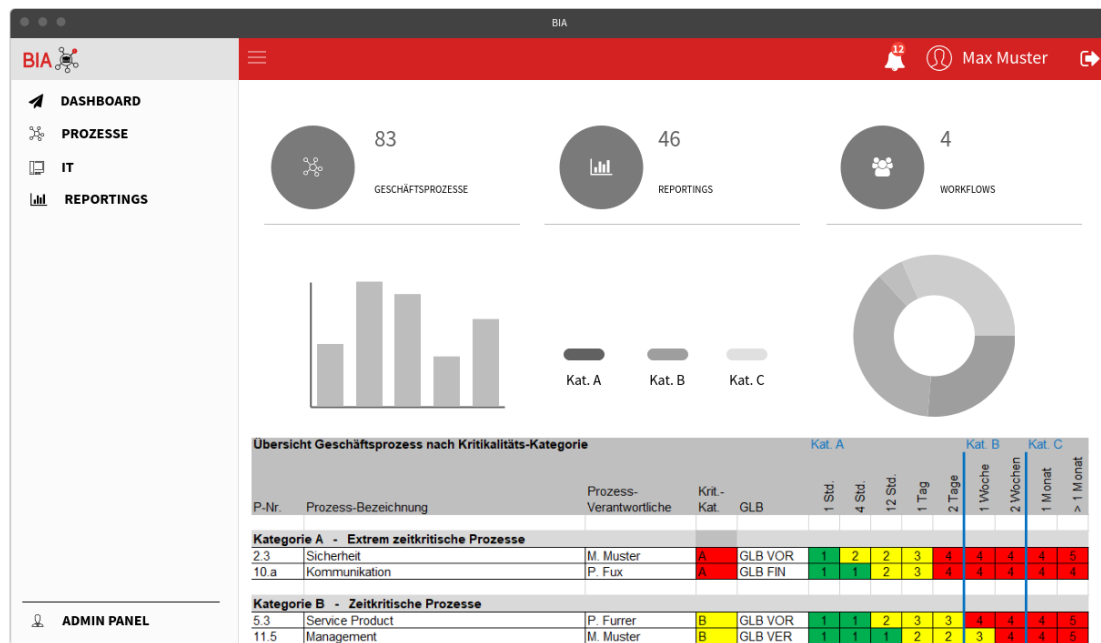
### 3.3. Design neues BIA-System

Unter der Berücksichtigung der Anforderungen der Schweizerischen Mobiliar und den beschriebenen Personas wird ein Design für den Prototypen entworfen. Durch aktive Zusammenarbeit mit dem Kunden konnten für die Hauptfunktionalitäten mehrere Mock-Ups erstellt werden. Die Mock-Ups dienen dazu, die Vorstellungen und Anforderungen des Kunden graphisch darzustellen und eine Vorlage für die Entwickler zu gestalten (Moos, 2016, S. 39-41).

Anhand der Personas lässt sich erkennen, dass keiner der Anwender des Business Impact Analyse Systems in der IT-Branche tätig ist. Die Mitarbeiter besitzen dennoch gute Kenntnisse mit Tabellenkalkulationsprogrammen sowie mit Textverarbeitungsprogrammen und sind bereit, neue Software zu testen und zu verwenden. Anhand der Anforderungen der Schweizerischen Mobiliar, im Kapitel «3. Anforderungen der Schweizerischen Mobiliar» und der IST-Analyse des bisherigen Systems, welches im Kapitel «2.1 User Experience und Usability BIA System» erläutert ist, wird im neuen System viel Wert auf die Benutzerfreundlichkeit festgelegt. Eine saubere graphische Benutzeroberfläche hilft den Mitarbeitern, effizienter die Geschäftsprozesse zu verwalten und die Reportings über alle Prozesse zu analysieren. Das Erstellen von Workflows beschleunigt die Arbeit für den Administrator, da die einzelnen Prozesse nicht wie bislang separat exportiert und anschliessend wieder integriert werden müssen. Die Verwendung der Workflows gewährleistet dauerhaft mit den aktuellsten Daten der Geschäftsprozesse arbeiten zu können, da die Persona Externe Mitarbeiterin als ein Anwender des Systems die Prozesse modifizieren kann. Bei der Umsetzung der Mock-Ups wird auch das Verwenden von Icons berücksichtigt, um die Darstellung zu vereinfachen. Dies wird im bisherigen System über ein Dashboard erzielt, wie in dem Kapitel «1.3 BIA – bisheriges System» aufgezeigt.

Durch diese Evaluation der Personas und der Anforderungen der Schweizerischen Mobiliar, wurden mehrere Mock-Ups erstellt. Jedes Mock-Up wurde mit dem Kunden aktiv überarbeitet, so dass diese auch den Anforderungen entsprechen. Nachfolgend wird eines dieser Mock-Ups aufgezeigt. Im ANHANG III befinden sich alle erarbeiteten Mock-Ups für das neue BIA System.

Abbildung 19: Mock-Up Dashboard



Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

Die Abbildung 19 zeigt das Mock-Up eines Dashboards. Dies ist die angezeigte Hauptseite nachdem sich ein Benutzer im System angemeldet hat. In der roten Navigationsleiste werden alle Nachrichten und der Name eines Benutzers angezeigt. Auf der Seitennavigation hat der Anwender direkten Zugang zu den Prozessen, IT-Liste und zu den Reportings. Auf dem Dashboard wird ein Reporting angezeigt sowie Diagramme mit Informationen über die unterschiedlichen Geschäftsprozesse. Durch dieses Dashboard erhalten die Personas Mitarbeiter und Administrator eine Übersicht aller Prozesse.

## 4. Analyse potenzielle Technologien

In diesem Abschnitt der Bachelorarbeit wird eine Analyse für die potenziellen Technologien aufgestellt, welche für die Umsetzung des Prototypen verwendet werden könnten. Als erster Teil der Analyse, werden drei Punkte berücksichtigt, welche bei der Implementierungsphase und der Entscheidung der Umsetzungstechnologie eine wesentliche Rolle spielen. Infolgedessen wird die Analyse in Frontend und Backend unterteilt und mehrere mögliche Technologien analysiert und evaluiert. Anhand der behandelten Technologien werden mehrere geeignete Varianten zur Entwicklung des neuen Systems aufgestellt. Diese Analysen helfen dem Arbeitgeber bei der Entscheidung der verwendeten Umsetzungstechnologie.

### 4.1. Vorgehensweisen

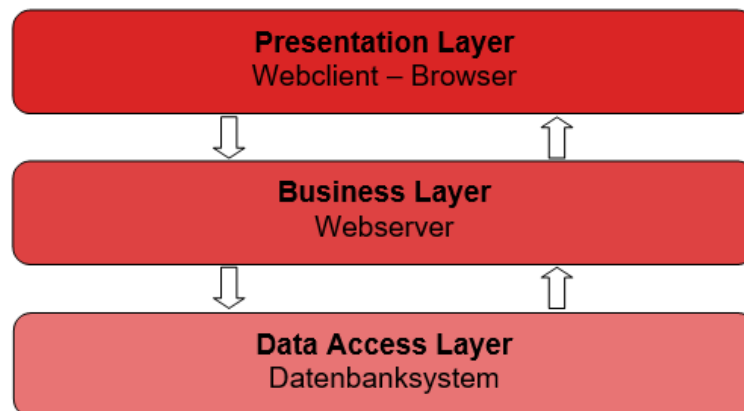
Dieses Unterkapitel befasst sich mit drei wesentlichen Punkten, die Einfluss auf die Entscheidung der unterschiedlichen Technologien nehmen. Zunächst wird die Vorgehensweise von Single-Page Applikationen und Multi-Page Applikationen erläutert. Danach wird eine populäre Softwarearchitektur vorgestellt, welche für Softwareprojekte empfehlenswert sind. Der dritte Punkt definiert den Vorteil der Verwendung eines Frameworks.

#### 1. Single-Page Applikation und Multi-Page Applikation

Bei der Entwicklung einer Webapplikation können zwei unterschiedliche Webanwendungen verwendet werden, der Single-Page Applikation und der Multi-Page Applikation. Bei einer Multi-Page Applikation werden mehrere Seiten entwickelt. Dabei wird jede Seite einzeln vom Server auf den Browser geladen. Im Gegenzug wird bei einer Single-Page Applikation alle Interaktionen nur einmal vom Server auf den Browser geladen. Der bedeutende Vorteil liegt darin, dass alle benötigten Informationen bereits auf der Seite gespeichert werden, ohne jeweils eine neue Anfrage an den Server zu senden. Jede Aktion des Anwenders weist dadurch eine geringere Aktualisierungszeit auf (Mesbah, 2009, S. 10).

## 2. Drei-Schichten-Architektur

Abbildung 20: Drei-Schichten-Architektur



(Eigene Darstellung)

Wie in der Abbildung 20 ersichtlich ist, werden bei der Drei-Schichten-Architektur alle Arbeitsschritte in drei unterschiedliche Schichten (Layer) aufgeteilt. Die erste Schicht ist die Präsentationsebene (Presentation Layer). In dieser Schicht wird die Applikation von den Anwendern über ein Frontend bedient. Dieser Teil der Applikation, kann von den Benutzern über eine grafische Benutzeroberfläche bedient werden. Die Applikationsschicht (Business Layer) ist die zweite Schicht der Architektur. In diesem Layer wird das Backend entwickelt, den Abschnitt der Applikation, welchen die Benutzer nicht betrachten können. Hier werden alle Funktionalitäten des Programmes programmiert. Damit diese Ebene die Daten der Applikation erhält um diese anschliessend im Frontend anzuzeigen, wird die dritte Ebene verwendet. Bei der dritten Schicht, die Datenbankschicht, werden die Daten in einem Datenbanksystem hinterlegt. Der grösste Teil der Funktionalitäten wird in der Applikationsschicht durchgeführt. Hier werden die Daten aus dem Datenbanksystem geladen, bearbeitet und anschliessend dem Anwender mit einem Design im Frontend ausgegeben. Das Hauptziel der Drei-Schichten-Architektur besteht darin, dass das Frontend vom Backend getrennt wird. So kann der Entwickler eine Funktionalität im Frontend in der Präsentationsebene ändern, ohne dass die gleiche Funktionalität noch im Backend in der Applikationsschicht angepasst werden muss (Aarsten, Brugali, & Menga, 1996, S. 1-3).

## 3. Frameworks

Ein Framework bietet die Wiederverwendbarkeit von unterschiedlichen Teilen der Softwareentwicklung an. Die Wiederverwendbarkeit bietet sich sowohl in Implementationsteilen als auch in Software Designs. Die Verwendung von Frameworks ermöglicht dem Entwickler schneller zu einem Ergebnis zu gelangen, die Kosten zu verringern und die Qualität der Software zu verbessern (Fayad & Schmidt, 1997, S. 32-34).

## 4.2. Frontend

Verschiedene Webtechnologien erlauben den Entwicklern innovative und responsive Applikationen zu entwickeln. Durch das responsive Design kann das erstellte Frontend sowohl auf einem Desktop als auch auf einem Mobile Gerät benutzerfreundlich bedient werden. Im nächsten Abschnitt «HTML5 und CSS3» wird das responsive Design detailliert erläutert. Unterschiedliche Web Frameworks und Bibliotheken werden dem Entwickler bereitgestellt, um die Arbeitsschritte effizient zu unterstützen. Der Inhalt einer Webapplikation wird durch die Programmiersprache HTML5 dargestellt. In den meisten modernen Applikationen wird der Inhalt zusätzlich von JavaScript und CSS3 unterstützt. Um dem Inhalt noch eine gewisse Struktur hinzuzufügen und das Verfahren der Navigation zu unterstützen, werden verschiedene Frameworks und Bibliotheken verwendet, wie zum Beispiel jQuery, Angular und Bootstrap (Shahzad, 2017).

### HTML5 und CSS3

HTML5 wird aus dem englischen Namen «**H**ypertext **M**arkup **L**anguage» zusammengestellt. Die fünf steht für die Version der Sprache. HTML5 ist eine Auszeichnungssprache welche im World Wide Web (WWW) verwendet wird. HTML5 bezieht sich nicht auf das Seitenlayout des Dokumentes, sondern auf die angezeigte Struktur. Diese Struktur wird dem Anwender über eine Website angezeigt (Gröning & Schwickert, 1996, S. 8-9).

Das Akronym CSS kommt aus dem englischen Namen «**C**ascading **S**tyle **S**heets». Wobei die Zahl drei für die Versionsnummer ist. Diese stilistische Sprache ist für die visuelle Präsentation der erstellten Dokumente mit HTML5 zuständig (Meyer, 2004, S. 1-7).

Seit der HTML-Version 5 und CSS-Version 3 bietet sich den Entwicklern einer Webseite die Möglichkeit, benutzerfreundliche responsive Webdesigns zu erstellen. Der Hauptvorteil liegt darin, dass die Ansicht der Webseite auf jeder Bildschirmgröße einheitlich aussieht. Die Website wird so gestaltet, dass die Benutzerfreundlichkeit für die Anwender auf jedem Gerät gewährleistet wird. Die nachfolgende Abbildung 21 zeigt ein Beispiel einer Webseite, die das responsive Webdesign anwendet (Gardner, 2011, S. 13-19).

Abbildung 21: Beispiel Responsive Webdesign



Quelle: (Nice, 2017)

## JavaScript und jQuery

JavaScript ist eine Interpretationssprache. Diese Programmiersprache wird bei Webseiten verwendet, um durch unterschiedliche Scripts die Interaktion mit dem Anwender zu gewährleisten. Zudem kann JavaScript eingesetzt werden, um den Webbrowser zu kontrollieren oder die Ansicht und das Design eines Webbrowsers zu ändern (Flanagan, 2006, S. 1-3).

Um Animationen auf einer Webseite einzufügen, kann die Interpretationssprache jQuery verwendet werden. Dies ist eine JavaScript Library, welche die Programmierung der Skripts vereinfachen. Diese frei verfügbare Library ermöglicht der Webseite die Interaktion innerhalb eines HTML Dokumentes (Asokan, 2013, S. 68-70).

## Bootstrap Framework

Bootstrap ist ein frei verwendbares JavaScript Framework, welches dem Entwickler hilft, ein responsives Webdesign zu erstellen. Dieses Framework wurde 2011 von den Mitarbeiter Mark Otto und Jacob Thornton des Unternehmens Twitter veröffentlicht. Durch das Verwenden dieses Produktes werden vordefinierte Designs von Komponenten der Webseite bereits zur Verfügung gestellt. Das individuelle Spaltengitter (in Englisch «Grid system») erleichtert die Entwicklung des responsive Webdesign für alle Geräte (Spurlock, 2013, S. 1-6).

## AngularJS Framework

Dieses Framework wird bei der Realisierung einer Single-Page Applikation verwendet. AngularJS ist ein JavaScript MVC-basierendes Framework. 2009 wurde AngularJS von Google veröffentlicht (Ramos, Valente, & Terra, 2017, S. 1-5). Das Akronym MVC steht für «**M**odel **V**iew **C**ontroller». Dies ist eine ähnliche Struktur wie die Drei-Schichten-Architektur, welche im Kapitel «4.1 Vorgehensweisen» beschrieben wird. Der Programmiercode wird hier klar zwischen dem Datenbanksystem «Model», der Applikationslogik «Controller» und der Ansicht «View» für den Anwender separiert. Der wesentliche Unterschied zur Drei-Schichten-Architektur liegt bei der Kommunikation innerhalb der Schichten. Wenn eine Änderung im Model vorgenommen wird, benachrichtigt diese die Präsentationsebene «View», um die Ansicht zu aktualisieren. Die Applikationsebene «Controller» dient hierbei nicht als Transportmittel wie dies bei der Drei-Schichten-Architektur der Fall wäre (Green & Seshadri, 2013, S. 3).

AngularJS umfasst das HTML Dokument in ein ausdrucksstärkeres Format. Dies wird durch die Verwendung von speziellen Markups durchgeführt, damit die Ansicht automatisch neu geladen wird, anstatt bei einer Änderung diesen Vorgang manuell zu aktualisieren. Die Integration von AngularJS ist benutzerfreundlich, da erst nach Ende des Ladeprozesses die Seite mit der Auswertung beginnt. Ein weiterer Vorteil von AngularJS besteht darin, dass die HTML Webseite mit eigenen Elementen und Attributen erweitert werden kann. Des Weiteren kann auf diese Weise die eigenen wiederverwendbaren Komponenten in einer Library zur Verfügung gestellt werden (Jain, Mangal, & Mehta, 2014, S. 17-18).

### 4.3. Backend

Bei der Entwicklung eines Backends können verschiedene Technologien angewendet werden. In dem nachfolgenden Abschnitt werden mehrere Programmiersprachen mit unterschiedlichen Frameworks vorgestellt, welche für die Entwicklung des Prototypen verwendet werden können. Um dem Arbeitgeber mehrere Varianten vorzustellen, wird für jede Programmiersprache anhand eines Bewertungssystems das bestmögliche Framework empfohlen.

## **Bewertungskriterien**

Für jede Variante werden unterschiedliche Technologien für das Frontend und Backend angewendet. Um die Entscheidung für das effizienteste Framework für das Backend treffen zu können, werden alle analysierten Frameworks anhand von drei unterschiedlichen Kriterien bewertet. Diese Bewertungskriterien berücksichtigen zugleich die Anforderungen der Entwicklung des Prototypen der Schweizerischen Mobiliar. Die Anforderungen der Schweizerischen Mobiliar wird in Kapitel «3. Anforderungen der Schweizerischen Mobiliar» ausführlich erarbeitet. Durch die Bewertung kann für jede Programmiersprache jeweils das passende Framework für die Schweizerische Mobiliar evaluiert werden.

### **1. Module**

Die Schweizerische Mobiliar möchte am Ende der Implementierungsphase über einen Prototyp des neuen Systems verfügen. Dieser Prototyp sollte den Anforderungen der Schweizerischen Mobiliar entsprechen. Um dies zu gewährleisten, sollte das Framework bestehende Module zur Verfügung stellen, welche die Entwicklungsgeschwindigkeit erhöhen. Durch existierende Module, wie die Authentifizierung, wird zusätzlich noch die Sicherheit der Anmeldedaten abgedeckt. Aus diesem Grund müssen die existierende Module analysiert werden, um eine Entscheidung für das Framework treffen zu können. In diesem Kriterium wird das Framework auf deren Funktionalitäten analysiert. Hierbei wird beachtet, ob bereits zur Verfügung gestellte Module existieren. Wichtig ist die Analyse über verfügbare Architekturen, da unterschiedliche Konzepte die Entwicklung strukturieren.

Durch ein dreipunkte System sollen die verschiedenen Frameworks bewertet werden. Der erste Punkt erlangt das Framework, wenn nützliche, verwendbare Funktionalitäten bereits existieren. Als nützliche, verwendbare Funktionalitäten werden Module bezeichnet, welche den Entwickler bei der Implementierung und wesentlichen Arbeitsschritten unterstützen. Ein weiterer Punkt erhält das Framework für die Bereitstellung einer Architektur, wie zum Beispiel MVC. Der letzte Punkt wird für die Leistungsfähigkeit vergeben. Hier wird berücksichtigt, wie leistungstark das Framework ist und zudem kontrolliert, ob alle Anforderungen mit dem angebotenen Framework umgesetzt werden können.

### **2. Zeitumfang**

Jedes Framework wird in diesem Bewertungskriterium analysiert und beachtet, welches den optimalsten Zeitaufwand aufweist. Die Implementationszeit muss zusätzlich das Aufsetzen der Environments sowie die Einarbeitungszeit in das neue Framework berücksichtigen. Die Einarbeitungszeit in die unterschiedlichen Frameworks ist abhängig von

der Programmiersprache und von den vorhandenen Dokumentationen. Wenn eine saubere Dokumentation besteht, kann das Verständnis des Frameworks rasch erlernt werden und erspart kostbare Zeit.

Das Bewertungskriterium bezieht sich in dieser Analyse auf ein dreipunkte System. Der erste Punkt wird dem Framework zugeteilt, wenn das Aufsetzen der Environments weniger als in einer Woche durchlaufen werden kann. Dies kann anhand bestehender Dokumentation analysiert werden. Um den zweiten Punkt zu erhalten, wird die vorhandene Dokumentation betrachtet. Grösstenteils stellen die Herausgeber der Frameworks selbst eine ausführliche Dokumentation zur Verfügung, welche mit in die Analyse einbezogen wird. Der letzte Punkt wird dem Framework zugewiesen, wenn die Einarbeitungszeit in das Framework schätzungsweise nicht länger als eine Woche dauert.

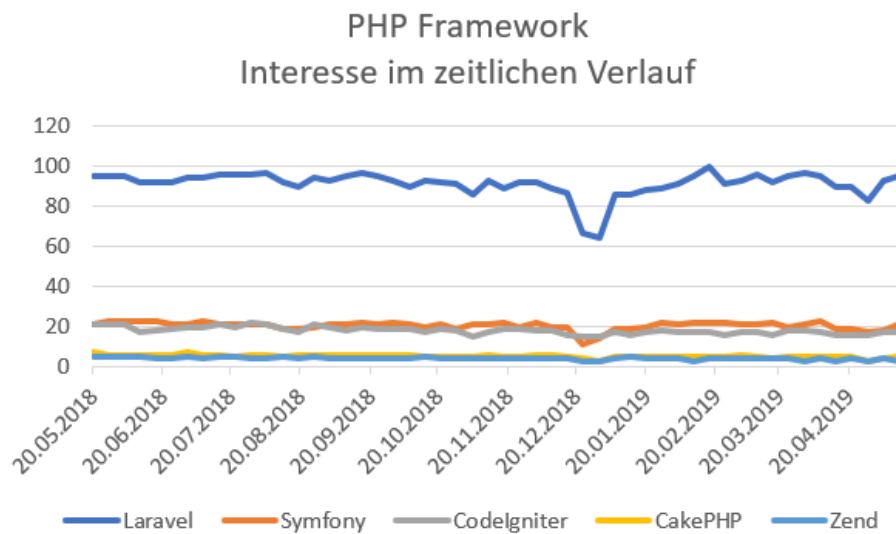
### **3. Modularität**

In diesem Bewertungskriterium wird analysiert, ob eine saubere Lösung für den Prototypen entwickelt werden kann. Dem Anwender des Frameworks soll die Möglichkeit bestehen, den Programmcode zu modulieren. Durch das Aufteilen der einzelnen Programmausschnitte besteht für den Entwickler die Möglichkeit der Wiederverwendbarkeit des Programmcodes. Sauberer Aufbau der Lösung, schnelleres Arbeiten und vermeiden von dupliziertem Code wird hierdurch gewährleistet. Der Hauptvorteil ist die Übersicht, welche dem Anwender angeboten wird.

Um diese Analyse für die einzelnen Frameworks zu bewerten, wird jedem Framework, welches diese Modularität aufweist, noch ein zusätzlicher Punkt zugewiesen.

## PHP Frameworks

Abbildung 22: PHP Framework



Eigene Darstellung und erstellt mit <https://trends.google.com/trends/explore>

Es existieren zahlreiche PHP Frameworks auf dem Markt. Von den populärsten PHP Frameworks ist Laravel das am meistgewählte Framework für alle neuen PHP Projekte, welche Google Trends in der Abbildung 22 aufzeigt.

Diese fünf PHP Frameworks werden nun anhand von unterschiedlichen Anforderungen bewertet und evaluiert. Die Bewertung ermöglicht, das nützlichste Framework für den Prototypen der Schweizerischen Mobiliar auszuwählen.

### Laravel

Laravel kann verwendet werden, um Web Applikationen nach der MVC Architektur zu entwickeln. Laravel ist ein frei verfügbares PHP Web Framework. Laravel ist sehr leistungsstark und einfach anzuwenden. Viele Module werden bereits von Laravel zur Verfügung gestellt, damit diese nicht selbst entwickelt werden müssen. Darunter zählen zum Beispiel die Benutzer Authentifizierung, die Datenbank Migration, die Verwendung von Einheitsteste und vieles mehr. Die zur Verfügung gestellten Module ersparen dem Entwickler viel Zeit und Aufwand (Chen, Ji, Fan, & Zhan, 2017, S. 2).

### Symfony

Symfony ist ein frei verfügbares PHP Framework, welches auf dem MVC Modell basiert. Die einzelnen Module haben seit der neuen Version eine stark verbesserte Leistungsfähigkeit. Das «Object-Relational Mapping» (ORM), oder zu Deutsch

«Objektrelationale Abbildung», wird von diesem Framework verwendet (Porebski, Przystalski, & Nowak, 2011, S. 6-10). ORM Lösungen werden angewendet, um eine Beziehung zwischen den Objekten und der Datenbank aufzubauen, damit die Objekte in die relationale Datenbank gespeichert werden können (Keller, 1997, S. 1-2).

## **CodeIgniter**

Das PHP Framework CodeIgniter ist leistungsstark und ist für weniger komplexere Lösungen entwickelt worden. CodeIgniter ist ein frei verfügbares Framework, welches jedoch nur teilweise auf das MVC Modell basiert. Das Model kann in diesem Framework ausgelassen werden. Bei komplexeren Lösungen ist dieses Framework allerdings nicht zu empfehlen, da die Leistung verschlechtert werden kann. Leistungsminimierung kann durch übermässige Funktionen oder durch zu grossen Zeitaufwand für die Konfiguration auftreten (Porebski et al., 2011, S. 6-10).

## **CakePHP**

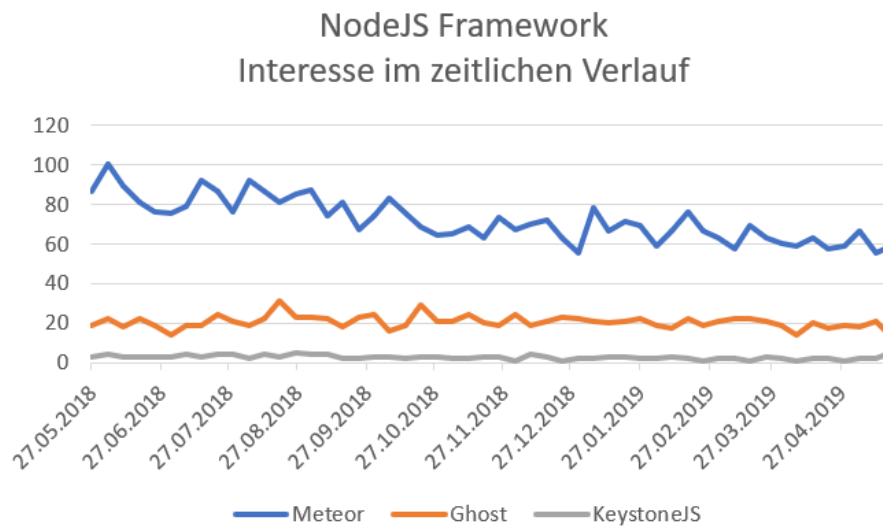
CakePHP ist ein frei verfügbares PHP Framework, welches gesellschaftsorientiert ist. Der Vorteil dieses Framework ist die Benutzerfreundlichkeit des Systems, sowie die Entwicklungsgeschwindigkeit. Für eine rapide Entwicklung des Systems, bietet das Framework viele Module an (Porebski et al., 2011, S. 6-10).

## **Zend**

Das PHP Framework Zend ist frei verfügbar und Komponenten basierend. Die Module können optional verwendet werden. Die MVC Architektur kann angewendet werden, damit das System flexibler ist. Zudem existiert eine gute Dokumentation zu den Funktionalitäten (Porebski et al., 2011, S. 6-10).

## NodeJS Framework

Abbildung 23: Node JS Framework



Eigene Darstellung und erstellt mit <https://trends.google.com/trends/explore>

Es gibt mehrere populäre NodeJS Frameworks. Jedoch gehören nur einzelne dieser Frameworks zu «full-stack» Lösungen. Eine «full-stack» Variante ist gut geeignet, da das BIA System der Schweizerischen Mobiliar eine komplexe Lösung ist. Als «full-stack» Framework sind Frameworks zu verstehen, welche den Entwickler sowohl beim Frontend als auch beim Backend unterstützen (Friedrich, 2017, S. 12). Meteor ist das meistgesuchte Framework, welches von Google Trends in der Abbildung 23 evaluiert wird.

### Meteor

Meteor ist ein Web Framework, welches auf JavaScript basiert. Der Vorteil dieses Framework liegt darin, dass nur eine Programmiersprache für das Frontend und das Backend in dieser «full-stack» Lösung angewendet wird. Dieses Framework kann verwendet werden, um Web oder auch Mobile Applikationen zu entwickeln. Für die Entwickler stehen mehrere Funktionalitäten zur Verfügung, welche als Package in die Software integriert werden kann (Friedrich, 2017, S. 12).

### KeystoneJS

KeystoneJS ist ein «Content Management System» (CMS) Framework. CMS unterstützt den Entwickler mit dem Anzeigen und Erstellen von Inhalten. Mehrere Schritte wie zum Beispiel das Erstellen eines Designs werden zur Verfügung gestellt (Boiko, 2001, S. 8-10). KeystoneJS basiert auf JavaScript und besitzt eine modulare Architektur. Diverse Funktionalitäten stehen zudem den Entwicklern zur Verfügung (Panati, 2016, S. 1-2).

## **Ghost**

Dieses frei verfügbare Framework ist ebenfalls ein CMS Framework. Hauptsächlich wird dieses Framework für Blogging Webseiten verwendet, kann aber sicherlich auch für andere Zwecke genutzt werden. Die Webseite kann durch Einfügen von graphischen Elementen zusammengestellt werden und der Text durch Anklicken bearbeitet werden. Für Entwickler besteht jedoch die Möglichkeit, durch JavaScript noch Änderungen im Backend vorzunehmen. Zudem stehen mehrere Funktionalitäten zur Verfügung (Schürmanns, 2019).

## **ASP .NET MVC**

ASP .NET MVC ist ein Webapplikations-Framework des Unternehmens Microsoft. Dies ist ein frei verfügbares Framework, welches basierend auf der Programmiersprache C# für die Entwicklung des Backends verwendet werden kann. Für die Entwicklung der Applikation kann die integrierte Entwicklungsumgebung (IDE) Visual Studio verwendet werden. Der Vorteil dieser Software liegt darin, dass das Produkt des Anbieters bereits einen lokalen Webserver, Internet Information Server (IIS), integriert hat (Chadwick, Snyder, & Panda, 2012, S. 3-9).

Dieses Framework verfolgt die MVC Architektur und bietet dem Entwickler die volle Kontrolle über alle Webseiten an. ASP .NET MVC arbeitet vielfach mit dem «Model-View-ViewModel» (MVVM) Modell. Der Unterschied zum MVC besteht darin, dass zusätzlich jede View noch ein eigenes Model besitzt. Dieses Model wird als ViewModel beschrieben, in dem alle Daten für die View vorbereitet werden (Steyer & Schwichtenberg, 2014, S. 15-17).

## **J2EE**

J2EE ist die Abkürzung von «Java 2 Enterprise Edition» und basiert auf der Programmiersprache Java. Diese Plattform ist für dynamische und interaktive Webseiten geeignet und ist zugleich sehr leistungsstark. J2EE Applikationen sind unabhängige Hardware-Lösungen, welche nach dem MVC Modell strukturiert werden. Durch die J2EE Plattform werden dem Anwender mehrere Dienste zur Verfügung gestellt. Diese Dienste werden anschliessend von einem Internetbrowser abgerufen. Das System kann sehr effizient angepasst werden, da sich das ganze Programm auf dem Server befindet und auf der Seite des Anwenders nur im Webbrowser aufgeführt wird (Wojciechowski, Sakowicz, Dura, & Napieralski, 2004, S. 342-345).

## Auswertung der PHP Frameworks

Jedes aufgezeigte PHP Framework wird nun gemäss der drei Bewertungskriterien analysiert und evaluiert. Das PHP Framework, welches die grösste Punktezahl aufweist, wird als eine mögliche Variante der Entwicklung des Prototypen empfohlen.

### 1. Module

**Laravel:** Dieses PHP Framework erhält alle drei Punkte. Das Framework verfolgt die MVC Architektur. Zudem sind die Funktionalitäten einfach zu verstehen und leistungsstark. Authentifizierung, das Routing, ein Session Manager, das Caching, IoC Container (Inversion of Control), Datenbank Migration, verwenden von Einheitsteste und zahlreiche verfügbare Komponenten werden als unterschiedliche Module bei diesem PHP Framework angeboten (Chen et al., 2017, S. 2).

**Symfony:** Dieses Framework basiert ebenfalls auf dem MVC Modell. Das System ist sehr leistungsstark und weist eine gute Benutzerfreundlichkeit auf. Als bestehende Funktionalitäten bietet das Framework das Caching, Authentifizierung von Benutzer, Routing, E-Mail und verfügbare Komponenten an (Zaninotto & Potencier, 2007, S. 1-5). Aus diesem Grund erhält dieses Framework alle drei Punkte des Bewertungssystems.

**CodeIgniter:** Das MVC Modell wird nur teilweise angewendet, da das Model optional ist. Diese Lösung wird nur für weniger komplexerer Systeme vorgeschlagen, deshalb ist dieses Framework nicht leistungsstark (Porebski et al., 2011, S. 6-10). Bezüglich den Funktionalitäten bietet CodeIgniter ein Session Management, Datenbank Zugriff, Erstellen von HTML Seiten und Testmodule an. Jedoch weist dieses Framework keine Sicherheit auf (Upton, 2007, S. 16-17). Als Bewertung für dieses Kriterium, erhält dieses Framework null Punkte, da keine der Anforderungen ganz abgedeckt wird.

**CakePHP:** Das Framework CakePHP setzt eine MVC Architektur für die Web Anwendungen ein. Bei der Entwicklung des Frontend stellt CakePHP mehrere Helfermodule zur Verfügung, um den Entwickler bei der Implementierung zu unterstützen. Jede dieser Funktionalität kann individuell bearbeitet und angepasst werden. Weitere Funktionalitäten wie ein Session Manager, das Caching, eingebaute Validation der Daten, Zugriffssteuerungsliste und Sicherheit werden von diesem Framework angeboten. Zudem ist das System leistungsstark und weist eine einfache Verwendung der Funktionalitäten auf (Golding, 2008, S. 1-6). Aus diesem Grund erhält dieses Framework alle drei Punkte des Bewertungssystems.

**Zend:** Das MVC Modell kann in diesem PHP Framework verwendet werden. Das System ist leistungsstark und fokussiert sich auf die Qualität und die Flexibilität des Systems. Jedoch werden nicht viele vordefinierte Module zur Verfügung gestellt (Pope, 2009, S. 2-4). Die Anforderungen an das neue System werden in diesem Punkt nicht erfüllt, da die Funktionalitäten von Grund auf implementiert werden müssen und nicht durch das Zend Framework unterstützt wird. Als Bewertungskriterium erhält dieses Framework deshalb zwei Punkte.

## 2. Zeitumfang

**Laravel:** Das Aufsetzen der Environments von Laravel kann in nur wenigen Schritten durchgeführt werden. Die benötigten Ressourcen sind ein Webserver, um die Webseite zum Laufen zu bringen, die Installation-Datei, welche von der Hauptseite heruntergeladen werden kann und eine Datenbank, um das Projekt zu starten (McCool, 2012, S. 5-6). Zur Unterstützung der Implementierung bietet dieses Framework zahlreiche Dokumentationen und Anleitungen.<sup>12</sup> Dieses PHP Framework erhält alle drei Punkte, da die Einarbeitungszeit in das Framework nicht zu viel Zeit in Anspruch nimmt.

**Symfony:** Nur wenige Massnahmen sind notwendig, um Symfony aufzusetzen. Ein laufender Webserver für die Webseite und eine relationale Datenbank. Symfony kann lediglich auf der Hauptseite heruntergeladen werden. Zudem ist die Einarbeitung in das System nicht zeitaufwendig. Die Präsentationsebene, welche dem Benutzer angezeigt wird, kann in HTML implementiert werden. Der Entwickler benötigt dafür keinerlei Vorkenntnisse des Frameworks (Bowler & Bancer, 2009, S. 7-13). Auf der Hauptseite des Herausgebers ist eine ausführliche Dokumentation aller Funktionalitäten hinterlegt.<sup>13</sup> Aus den folgenden Gründen erhält dieses Framework alle drei Punkte des Bewertungssystems.

**CodeIgniter:** Für das Aufsetzen der Environments wird ein laufender Webserver für die Webseite sowie eine Datenbank benötigt. CodeIgniter kann auf der Hauptseite des Anbieters heruntergeladen werden. Für die Verbesserung des Verständnis der Funktionalitäten werden ausführliche Dokumentationen angeboten. Dadurch kann sich der Entwickler mühelos in das Framework einarbeiten (Upton, 2007, S. 7-22). Dieses Framework erhält alle Punkte des Bewertungssystems.

<sup>12</sup> Quelle: <https://laravel.com/docs/5.8>

<sup>13</sup> Quelle: <https://symfony.com/doc>

**CakePHP:** Für die Environments wird ein Webserver für die Webseite und eine Datenbank benötigt. Das Framework kann auf der Hauptseite des Anbieters heruntergeladen werden. Das Unternehmen Cake Software Foundation (2019) bietet eine umfassende Dokumentation an. Jedoch benötigt dieses Framework eine längere Einarbeitungszeit, da diverse Funktionalitäten bei der Verwendung des Frameworks berücksichtigt werden müssen. Aus diesem Grund erhält das System zwei Punkte.

**Zend:** Das Framework kann auf der Hauptseite heruntergeladen werden. Für das Aufsetzen der Environments wird ein Webserver und eine Datenbank benötigt. Dies impliziert auch die Installation von PHP (Padilla, 2009, S. 1-2). Die Dokumentationen sind nicht extrem ausführlich und deshalb ist die Einarbeitungszeit des Frameworks länger als die Anforderungen entsprechen.<sup>14</sup> Ein Punkt erhält das Framework für das Aufsetzen der Environments.

### 3. Modularität

**Laravel:** Durch die System Architektur und die Struktur des MVC Models, erlaubt es dem Entwickler die Wiederverwendbarkeit des Programmcodes einzusetzen (Chen et al., 2017, S. 2-3). Dieses Framework erhält den Punkt für die Modularität.

**Symfony:** Dieses Framework gewährleistet eine stabile Modularität durch eine klare Strukturierung des Programmcodes. Dadurch können Teilstücke auch wiederverwendet werden (Zaninotto & Potencier, 2007, S. 25-29). Dieses Framework erhält den benötigten Punkt.

**CodeIgniter:** Dieses Framework besitzt eine gute Struktur und hält die Komponenten gut organisiert. Zudem unterstützt das Framework den Entwickler bei der Fehlersuche (Myer, 2008, S. 10). Aus diesem Grund erhält dieses Framework den Punkt.

**CakePHP:** Der Entwickler ist nicht gezwungen, eine bestimmte Struktur anzuwenden. Dies ist ein grosser Nachteil, da bei komplexeren Systemen das System strukturlos wird. So verliert der Entwickler die Möglichkeit der Modularität (Bari & Syam, 2008, S. 7). Deshalb erhält dieses Framework keinen Punkt.

**Zend:** Durch das MVC Modell erhält dieses Framework eine saubere Struktur. Dies erlaubt dem Entwickler die Wiederverwendbarkeit von Codeausschnitten (Padilla, 2009, S. 29). Aus der Sicht des Bewertungssystems erlangt dieses Framework den Punkt.

---

<sup>14</sup> Quelle : <https://docs.zendframework.com/>

## Ergebnis der Auswertung der PHP Frameworks

Tabelle 5: Auswertung PHP Framework

Framework	Module	Zeitungsumfang	Modularität	Bewertung
Laravel	3	3	1	7
Symfony	3	3	1	7
CodeIgniter	0	3	1	4
CakePHP	3	2	0	5
Zend	2	1	1	4

(Eigene Darstellung)

Die Tabelle 5 zeigt die Übersicht der Punkteverteilung des Bewertungssystems. Anhand der Auswertung der Analysen stellt sich heraus, dass die PHP Frameworks Laravel und Symfony am besten für die Implementierung geeignet sind. Beide dieser Frameworks erfüllen alle Anforderungen an den gewünschten Prototypen. Die anderen drei Frameworks erfüllen nur teilweise die Anforderungen für die Entwicklung des neuen Systems.

## Auswertung der NodeJS Frameworks

In diesem Abschnitt werden alle NodeJS Frameworks analysiert und anhand des Bewertungssystems ausgewertet. Das NodeJS Framework, welches die grösste Punktezahl aufweist, wird als eine mögliche Variante der Entwicklung des Prototypen empfohlen.

### 1. Module

**Meteor:** Dieses Framework ist sehr leistungsstark und unterstützt den Entwickler mit mehreren Arbeitsschritten. Durch das nötige Wissen der Webentwicklung gemäss den Programmiersprachen HTML, CSS und JavaScript kann der Entwickler bereits grosse und komplexe Projekte umzusetzen. Dieses Framework stellt viele Funktionalitäten zur Verfügung und wird für Echtzeitapplikationen verwendet. Bei der Anpassung der Daten auf einem System, werden diese direkt auf allen Systemen aktualisiert, ohne dass die Anwender die Seite neu laden müssen (Rust, Schelling, & Schipper, o.D., S. 1-4). Dieses Framework erhält zwei Punkte, da eine Struktur des Codes vorgegeben wird und nicht angepasst werden kann. Das Programm ist andernfalls nicht mehr ausführbar.

**KeystoneJS:** Die «Model View Template» (MVT) Architektur wird in diesem Framework angewendet. Die MVT Architektur ist ähnlich zum MVC Modell, bis auf den minimalen Unterschied, dass KeystoneJS den Controller selber händelt (Panati, 2016, S. XV-XVI). Das System ist enorm leistungsstark und bietet mehrere Funktionalitäten an. Ein Admin Panel wird automatisch erstellt, ein Session Management, eine E-Mail Funktionalität, die Unterstützung des Formular und Datenbankfelder werden von KeystoneJS als

Funktionalitäten angeboten (Panati, 2016, S. 1-2). Dieses System erhält die volle Punktzahl des Bewertungssystems.

**Ghost:** Das NodeJS Framework ist nach dem MVC Modell strukturiert (Terpstra, 2013). Das System ist sehr leistungsstark, da das System von Nicht-Entwicklern verwendet wird und für Entwickler trotzdem die Möglichkeit besteht, den Code individuell anzupassen. Dem Entwickler werden mehrere Funktionalitäten zur Verfügung gestellt, zum Beispiel die Mehrsprachigkeit und das Routing, um beliebige Seitenstrukturen anzuzeigen. Ein Admin Panel wird dem Benutzer zur Verfügung gestellt (Schürmanns, 2019). Aus diesen Gründen erfüllt Ghost alle Anforderungen des Bewertungskriteriums.

## 2. Zeitumfang

**Meteor:** Das Aufsetzen der Environments kann bei diesem Framework rasch durchgeführt werden. Meteor kann von der Hauptseite heruntergeladen und auf dem System installiert werden (Strack, 2015, S. 1-3). Auf der Website von Meteor befindet sich eine detaillierte Dokumentation, welche den Entwickler stets unterstützen kann.<sup>15</sup> Aus diesem Grund erhält dieses Framework alle Punkte des Bewertungssystems.

**KeystoneJS:** Mehrere Vorinstallationen sind für das Aufsetzen der Environments nötig. Nach der Installation von NodeJS und MongoDB kann KeystoneJS installiert werden (Panati, 2016, S. 13-17). Auf der Hauptseite von KeystoneJS werden ausführliche Anleitungen und Dokumentationen zur Verfügung gestellt.<sup>16</sup> Diese Anleitung unterstützt den Entwickler bei der Einarbeitung in das System. Aus diesem Grund erhält das System alle Punkte in diesem Bewertungskriterium.

**Ghost:** Das Aufsetzen der Environments kann beim Ghost Framework durch das Klonen des GitHub Repository vorgenommen werden (Terpstra, 2013). GitHub stellt dem Entwickler eine Plattform zur Verfügung, um darauf frei verfügbare Softwareprojekte zu entwickeln und zu warten. Alle Entwickler, die auf öffentliche Projekte Zugriff haben, können diese Projekte frei verwenden (Russell, 2014, S. 279-281). Da Ghost hauptsächlich für Blogging Webseiten entwickelt wurde, existieren ausführliche Anleitungen, um einen Blog zu erstellen. Für den Entwickler besteht die Möglichkeit, die Dokumentation auf der Hauptseite des Anbieters einzusehen. Jedoch sind die angebotenen Dokumentationen beschränkt. Deshalb

---

<sup>15</sup> Quelle: <https://docs.meteor.com/>

<sup>16</sup> Quelle: <https://keystonejs.com/documentation>

beansprucht die Einarbeitungszeit in das Framework mehr Zeit.<sup>17</sup> Aus diesem Grund erhält das System nur einen Punkt des Bewertungssystems, weil die Environments schnell aufgesetzt werden können.

### 3. Modularität

**Meteor:** Dieses Framework bietet die Aufteilung des Programmcodes an. Die Entwickler der Applikation können mehrere Teilausschnitte wiederverwenden und den Code separieren.<sup>18</sup>

**KeystoneJS:** Durch die verwendete Architektur, besteht die Möglichkeit, den Code gut zu strukturieren, damit Programmteile wiederverwendbar sind (Panati, 2016, S. 1-2).

**Ghost:** Die MVC Architektur ermöglicht dem Entwickler die Implementierung von wiederverwendbarem Code (Terpstra, 2013).

## Ergebnis NodeJS Framework

Tabelle 6: Auswertung NodeJS Framework

Framework	Module	Zeitumfang	Modularität	Bewertung
Meteor	2	3	1	6
KeystoneJS	3	3	1	7
Ghost	3	1	1	5

(Eigene Darstellung)

Die Punkteverteilung des Bewertungssystems wird in obiger Tabelle 6 dargestellt. Anhand der Ergebnisse, welche durch die Analyse gewonnen wird, kann festgelegt werden, dass nur das KeystoneJS Framework die benötigten Anforderungen an den Prototypen erfüllt. Die anderen beiden Frameworks erfüllen die Anforderungen nur teilweise und sind deshalb nicht empfehlenswert für die Entwicklung des neuen Systems der Schweizerischen Mobiliar.

### 4.4. Varianten

In diesem Abschnitt der Bachelorarbeit werden mehrere potenzielle Varianten analysiert, welche angewendet werden können, um den Prototypen zu entwickeln. Jede Variante divergiert sich durch die verschiedenen Technologien, den Zeitumfang für die Einarbeitung der Technologie und die Modularität des Systems. Die verschiedenen Varianten werden jeweils in Frontend und Backend unterteilt.

<sup>17</sup> Quelle: <https://docs.ghost.org/>

<sup>18</sup> Quelle: <https://guide.meteor.com/structure.html#splitting-your-app>

## **Variante 1: PHP Framework Laravel**

### **Frontend**

Bei dieser Variante werden im Frontend mehrere Webtechnologien verwendet. Für das Erstellen des Designs werden HTML5, CSS3, JavaScript und jQuery verwendet. Damit die Daten auf dem Frontend angezeigt werden können, wird die Programmiersprache PHP verwendet.

### **Backend**

Anhand der Ergebnisse der Auswertung der PHP Frameworks, sind die Plattformen Laravel und Symfony am besten für das neue System geeignet. In dieser Variante wird auf Grund der flachen Lernkurve das Framework Laravel verwendet werden (Laaziri, Benmoussa, Khouliji, Larbi, & Yamami, 2018, S. 711).

## **Variante 2: NodeJS Framework KeystoneJS**

### **Frontend**

Für das Frontend einer NodeJS Applikation kann die Template Library «Handlebars.js» verwendet werden. Die «Handlebars» ersetzen den statischen Code im Frontend durch deren eigenen Syntax. Dies hat zur Folge, dass das Template sauber strukturiert und wiederverwendbar ist. Das Design des Frontend basiert zusätzlich noch auf HTML5, CSS3, JavaScript und jQuery (Manricks, 2013, S. 1-4).

### **Backend**

Das NodeJS Framework KeystoneJS wird in dieser Variante für das Backend verwendet. Gemäss den Ergebnissen der Auswertung für das NodeJS Framework, hat KeystoneJS als einzige Plattform alle Anforderungen an den Prototypen erfüllt.

## **Variante 3: C# Framework ASP .NET MVC**

### **Frontend**

Damit serverseitiger Code in dem Frontend angezeigt werden kann, wird in dieser Variante mit der Syntax «Razor» gearbeitet. Razor wird verwendet, um windowsbasierter Webserver mit Webinhalt zu verknüpfen. Als Programmiersprache verwendet diese Syntax C#. Innerhalb der Razor-Templates wird das Design mit HTML5, CSS3, JavaScript und jQuery noch modifiziert (Chadwick, 2011, S. 1-3).

## Backend

Die ASP .NET MVC Plattform basiert auf der Programmiersprache C#. Wie im Kapitel «4.3. Backend» bereits erwähnt, hat diese Variante den Vorteil, dass nicht zusätzlich im Backend noch ein eigener, lokaler Webserver installiert und eingerichtet werden muss.

## Variante 4: Entwicklung nach Mobiliar

Die letzte Variante basiert auf den Technologien, welche die Schweizerische Mobiliar bei Eigenentwicklungen anwendet.<sup>19</sup> Diese Variante wird erarbeitet, um dem Auftraggeber aufzeigen, dass der Prototyp auch in den gleichen Technologien umgesetzt werden kann, wie die Eigenentwicklung der Schweizerischen Mobiliar.

## Frontend

Basierend auf Angular wird eine Rich Web Client (RWC) Applikation entwickelt. RWC ist eine Technologie, welche nicht stetig mit dem Server interagieren muss, damit der Inhalt auf der Webseite angezeigt wird. Dafür wird eine Laufzeit bei dem Client zur Verfügung gestellt, in welcher die Applikation läuft. In Kombination mit Angular kann daraus eine effiziente Single-Page Applikation entwickelt werden (Müller, 2009, S. 20-22). Zudem wird Angular noch jeweils mit HTML5, CSS3 und JavaScript unterstützt.

## Backend

Im Backend werden Java Enterprise Service (JES) verwendet, welche auf J2EE basieren. Auf einem Webserver werden mehrere Webservices bereitgestellt. Webservices repräsentieren Funktionalitäten, welche für die Anwender über Standard-Webtechnologien zugänglich sind (McGovern, Tyagi, Stevens, & Mathew, 2003, S. 3-4). J2EE wird von der Schweizerischen Mobiliar verwendet, da diese Plattform für das Erstellen und Bereitstellen der Webservices einzelne Funktionalitäten zur Verfügung stellen (McGovern et al., 2003, S. 20-21).

---

<sup>19</sup> Gemäss mündlicher Mitteilung von Herr Indermühle Emanuel der Abteilung IT-Architektur und Technologien (10.05.2019)

## Varianten Auswertung

Tabelle 7: Auswertung Varianten

Nr.	Potenzielle Technologien	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5	Woche 6	Woche 7
	<b>Variante 1: PHP Framework Laravel</b>							
1	Einarbeiten Framework							
2	Aufsetzung Environments							
3	Erstellung Datenbank							
4	Implementierung							
	<b>Variante 2: NodeJS Framework KeystoneJS</b>							
1	Einarbeiten Framework							
2	Aufsetzung Environments							
3	Erstellung Datenbank							
4	Implementierung							
	<b>Variante 3: C# Framework ASP .NET MVC</b>							
1	Einarbeiten Framework							
2	Aufsetzung Environments							
3	Erstellung Datenbank							
4	Implementierung							
	<b>Variante 4: Entwicklung nach Mobiliar</b>							
1	Einarbeiten RCW basierend Angular							
2	Einarbeiten JSE basierend J2EE							
3	Aufsetzung Environments							
4	Erstellung Datenbank							
5	Implementierung							

(Eigene Darstellung)

Die Tabelle 7 zeigt den zeitlichen Aufwand der vorgeschlagenen Varianten auf. Die Varianten eins bis drei sind bezüglich der Entwicklungsphase identisch. Bei der Variante vier muss für das Einarbeiten in die verschiedenen Technologien mehr Zeit berücksichtigt werden. Mit der Begründung, dass die Absprache mit der Schweizerischen Mobiliar zusätzlich eingeplant werden muss, damit der Prototyp exakt nach den Eigenentwicklungen aufgebaut wird.

Alle vier Varianten weisen unterschiedliche Vor- und Nachteile auf. Vorteilhaft an den Varianten eins bis drei ist, dass die Einarbeitungszeit in die Technologien erheblich kleiner ist als bei der letzten Variante. Zudem ist zu erwähnen, dass bei allen Varianten der «state of the art», also der neuste Stand der Technologien, angewendet wird. Andererseits wird ein ausschlaggebender Nachteil einbezogen, dass das Ergebnis der Implementationsphase nicht in das System der Schweizerischen Mobiliar übernommen und weitergeführt werden

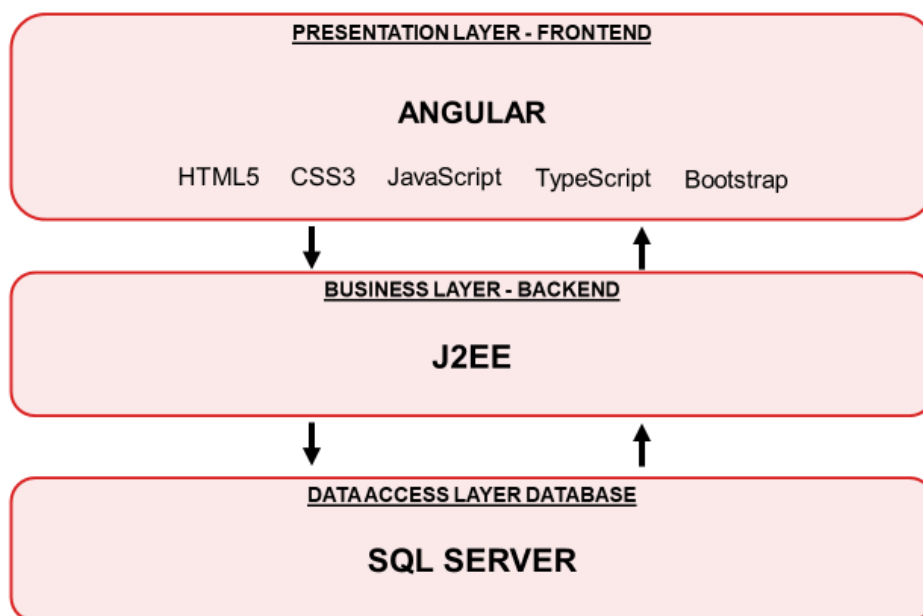
kann. Damit der Arbeitgeber die Applikation in ihrem System einführen kann, muss das Programm nach der Variante vier entwickelt werden.

Zusammen mit dem Auftraggeber wurden alle Varianten evaluiert. Die Entscheidung fiel auf die letzte Variante, wie im ANHANG IV ersichtlich ist. Die Entwicklung nach den Anforderungen der Schweizerischen Mobiliar ist optimal, da diese nach der Bachelorarbeit übernommen und weitergeführt werden können. Die nachfolgenden Kapiteln beschreiben die Entwicklung eines Prototypen basierend auf der ausgewählten Variante. Der Prototyp dient am Ende dieser Bachelorarbeit der Schweizerischen Mobiliar als Basis für die Weiterführung des Projektes.

## 5. Systemarchitektur

Anhand der Auswertung der Varianten im Kapitel «4.4. Varianten» wird der letzte Vorschlag umgesetzt. Bei dieser Variante werden dieselben Technologien eingesetzt, welche die Schweizerische Mobiliar bei Eigenentwicklungen anwenden. Dieser Teil der Abhandlung befasst sich mit der Umsetzung der Systemarchitektur. Das neue BIA System basiert auf der Drei-Schichten-Architektur, welches im Kapitel «4. Analyse potenzielle Technologien» beschrieben wird. Die folgende Abbildung 24 zeigt diese Systemarchitektur auf. In jeder Schicht wird zudem noch die verwendete Technologie dargestellt.

Abbildung 24: Systemarchitektur



(Eigene Darstellung)

### Präsentation Layer – Frontend

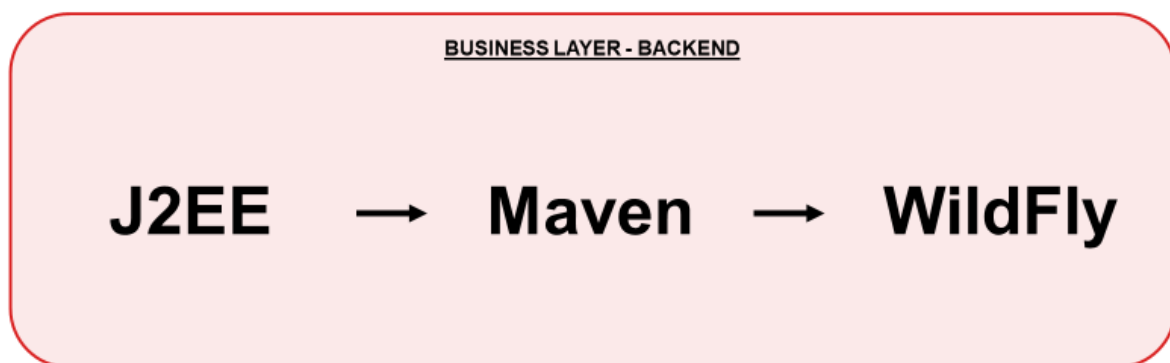
Als Implementierungssprache im Frontend wird Angular Version 7 verwendet. Im Kapitel «4.2 Frontend» wird AngularJS beschrieben. Jedoch wird diese Bachelorarbeit mit der gleichen Version umgesetzt, welche die Schweizerische Mobiliar verwendet. Angular 7 ist eine höhere Version des AngularJS Frameworks. Im Gegensatz zu AngularJS basiert Angular auf TypeScript. Jedoch unterstützen die Internet Browser kein TypeScript. Infolgedessen muss zum Anzeigen auf dem Browser der Code in der Laufzeit in JavaScript umgewandelt werden. Angular besitzt unterschiedliche Tools, welche den Kompilierungsvorgang übernimmt (Deeleman, 2016, S. 14-17). Der Hauptvorteil für die Verwendung von TypeScript liegt darin, dass die Daten unterschiedliche Arten von Inhalten darstellen können, durch sogenannte Anzeigetypen. Angular basiert auf wiederverwendbare

Komponenten, welche die Kontroller von AngularJS ersetzt haben. Mithilfe von HTML5, CSS3 und Bootstrap 4 wird der Inhalt in einem Browser angezeigt (Deeleman, 2016, S. 1-8; 31-34; 39-41).

In Angular wird das Projekt durch die Verwendung von Node JS verwaltet. Node JS ist eine JavaScript Plattform, um serverseitigen Code auszuführen. So kann das erstellte Projekt gestartet aber auch zusätzliche Pakete installiert werden. Um Abhängigkeiten oder neue Pakete zu installieren, wird die offizielle Helferklasse «npm» für Nodeprojekte verwendet. Zum Beispiel musste Bootstrap 4 durch npm in das Angular Projekt installiert werden, damit Bootstrap für das Frontend angewendet werden kann (Hahn, 2016, S. 3-6, 18-25).

## Business Layer – Backend

Abbildung 25: Softwarearchitektur Backend



(Eigene Darstellung)

Im Backend werden JES (Java Enterprise Services) erstellt, welche auf J2EE basieren. Die Abbildung 25 zeigt den Ablauf im Backend auf, um einen Webservice bereitzustellen. Die Schweizerische Mobiliar verwendet RESTful als Programmierschnittstelle, welches im nächsten Kapitel «5.2 RESTful Webservices» detaillierter geschildert wird. Damit für das Frontend Zugriff auf die Webservices gewährleistet wird, muss das J2EE Projekt auf einem Webserver zur Verfügung gestellt werden. Als Webserver wird in dieser Bachelorarbeit WildFly Version 10 verwendet. Die Schweizerische Mobiliar verwendet den jBoss Server Version 7 für ihre Entwicklungsprojekte. Der WildFly Server ist die neuere Version vom jBoss Server, welcher zusätzlich 2013 noch umbenannt wurde (Spolti, 2014, S. 5-8).

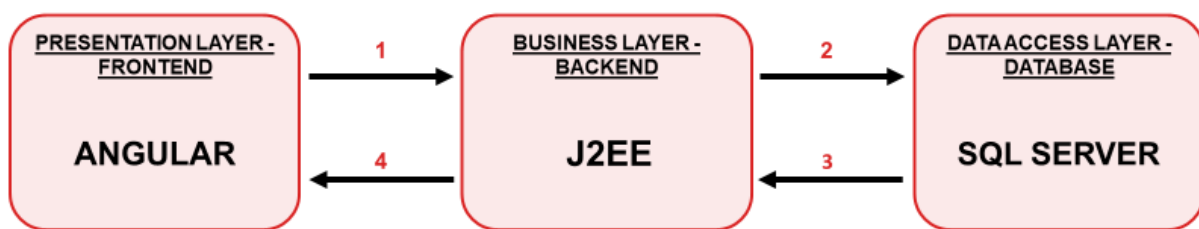
Maven wird verwendet, um das Projekt zu erstellen (im Englischen «build»). Dies ist ein frei verfügbares Tool, welches in J2EE Projekten verwendet werden kann. Maven verfolgt spezifische Standards, die verlangen, dass das Projekt ein Project Object Model (POM) besitzt. POM ist eine XML-basierende Datei, welche in dem Root-Verzeichnis des Projektes hinterlegt wird. In dieser Datei werden alle Metadaten und Abhängigkeiten für das Erstellen des Projektes definiert (Goncalves, 2013, S. 541-543).

## Data Access Layer – Database

Die Datenbank wird mit dem Microsoftprodukt SQL Server 2016 erstellt. Über das Backend kann mit einem authentifiziertem Benutzer auf die Datenbank zugegriffen werden, um so die Daten zu verwalten. Das verwendete Datenbankschema wird im Kapitel «6.3 Datenbankschema» veranschaulicht.

### 5.1. Zusammenspiel Layer

Abbildung 26: Zusammenspiel Layer



(Eigene Darstellung)

Die Abbildung 26 zeigt das Zusammenspiel der Schichten auf. Der erste Pfeil repräsentiert einen Request vom Client zum Server. Dabei ruft der Client eine URL-Adresse auf, welche im Header und im Body der Anfrage Daten mitliefert. Anhand dieses Request führt der Server eine dementsprechende Logik aus und greift auf die benötigten Daten auf die Datenbank zu. Dieser Zugriff wird durch den zweiten Schritt dargestellt. Anschliessend liefert die Datenbankschicht die Daten oder eine Fehlermeldung dem Backend zurück, welches durch die dritte Verbindung dargestellt wird. Zuletzt überträgt der Webservice die Daten oder der Ausnahmezustand dem Client zu.

Jede Antwort, der vom Webservice dem Frontend übergeben wird, enthält zusätzlich einen Status-Code. Dieser HTTP Status-Code informiert den Client über den Zustand des Ablaufprozesses. Wie in der Tabelle 8 ersichtlich ist, erkennt der Client an dem Status-Code 200, dass die Anfrage an den Server erfolgreich war. Wenn sich ein Anwender im entwickelten Prototypen mit falschen Benutzerdaten anmeldet, erhält dieser den Status-

Code 401 «Nicht autorisiert» (Dazer, 2012, S.5). Die Tabelle 8 zeigt die meistverwendeten Status-Codes auf.

Tabelle 8: HTTP Status-Codes

<b>200</b>	OK	<b>400</b>	Bad Request
<b>201</b>	Created	<b>401</b>	Unauthorized
<b>202</b>	Accepted	<b>403</b>	Forbidden
<b>204</b>	No Content	<b>404</b>	Not found
<b>301</b>	Moved Permanently	<b>408</b>	Request Timeout
<b>302</b>	Found	<b>415</b>	Unsupported Media Type
<b>304</b>	Not Modified	<b>500</b>	Internal Server Error
<b>307</b>	Temporary Redirect	<b>501</b>	Not Implemented

Quelle: eigene Darstellung in Anlehnung an (Dazer, 2012, S. 5)

Im nächsten Kapitel «5.2 RESTful Webservices» werden die RESTful Webservices definiert, welche auf dem Backend erstellt werden. Bei diesen Webservices werden die HTTP Status-Codes definiert, welche als Antwort an den Client gesendet werden.

## 5.2. RESTful Webservices

RESTful sind Representational State Transfer (REST) basierende Webservices. REST ist ein Architektur-Stil, der im Jahr 2000 von Roy Fielding an der Universität Kalifornien eingeführt wurde. Die erstellten RESTful Webservices konzentrieren sich auf die Ressourcen eines Systems und auf die Art und Weise, wie diese Ressourcenzustände über eine HTTP-Verbindung angesprochen werden. Alle REST Webservices basieren sich auf die nachfolgenden vier Basisprinzipien (Rodriguez, 2008, S. 1-11).

### Explizit HTTP-Methoden verwenden

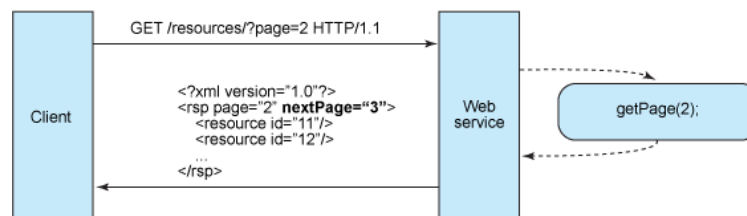
Entwickler, die mit REST arbeiten, werden aufgefordert, die expliziten HTTP-Methoden zu verwenden. Darunter ist das Prinzip der CRUD-Operationen (Create, Read, Update und Delete) zu verstehen, welche bei REST angewendet werden (Rodriguez, 2008, S. 2). Der Autor Rodriguez Alex (2008, S. 1-2) stellt das Prinzip wie folgt dar:

- GET: Um Ressourcen zu erhalten
- POST: Um neue Ressourcen auf dem Server zu speichern
- PUT: Um Ressourcen auf dem Server anzupassen
- DELETE: Um Ressourcen zu löschen

## Zustandslos sein

Zustandslose Serverseitige Komponenten sind für einen Entwickler leichter zu entwickeln und lasten den Server nicht aus. Hierbei wird dem Client die Verantwortung für den Service übergeben. Bei einem Aufruf einer HTTP-Methode überträgt der Client alle nötigen Informationen an den Server, damit der Server eine Antwort aus dem Request generieren kann. Als Beispiel kann Abbildung 27 betrachtet werden. Bei diesem Request wird die Seitenzahl der aktuellen Seite sowie die nächste Seite dem Webservice übergeben. So kann der Server lediglich eine Methode aufrufen, ohne die nächste Seite zu berechnen (Rodriguez, 2008, S. 1-11).

Abbildung 27: RESTful zustandslos sein



Quelle: <http://www.gregbulla.com/TechStuff/Docs/ws-restful-pdf.pdf>

## Verwendung von Verzeichnisstruktur (URIs)

Ein RESTful Webservice wird über die Uniform Ressource Identifier (URI) identifiziert. Wichtig ist dabei acht zu geben, dass die URI sinnvoll benannt wird. Mittels der URI kann eine bestimmte Ressource im System erreicht werden. Die HTTP-Methoden werden anschliessend verwendet, um auf diese adressierbaren Ressourcen zuzugreifen (Dazer, 2012, S. 2-4). Im Kapitel «6.5 HTTP-Methoden» sind Beispiele der verwendeten URI Ressourcen ersichtlich, welche für die Implementierung des Prototypen verwendet werden (Rodriguez, 2008, S. 1-11).

## Transfer von XML und/oder JSON

Bei einem Request auf die unterschiedlichen Ressourcen erhält der Client einen aktuellen Zustand der Ressource, welche die Daten von der Datenbank lädt. Dies kann mit der Darstellung eines Datensatzes einer Datenbank verglichen werden. Serverseitig muss bestimmt werden, welches Format beim Austausch der Ressourcen verwendet werden soll oder erlaubt ist. Als Format kann **eXtensible Markup Language (XML)** und/oder **JavaScript Object Notation (JSON)** verwendet werden (Rodriguez, 2008, S. 1--11).

XML wird gleich wie HTML5 mit sogenannten «Tags» definiert, welche die Daten der Ressource enthalten. Dagegen ist JSON ein textbasierender, offener Standard. JSON hat den Ursprung bei der JavaScript-Sprache. Die Darstellung wird bei JSON durch eine geordnete Liste und als Namen- oder Werten-Paare zusammengestellt (Zunke & D'Souza, 2014, S. 257-259). Die Abbildung 28 zeigt anhand eines kleinen Beispiel wie die gleiche Ressource in den unterschiedlichen Formaten aussehen würde.

Abbildung 28: Beispiel XML vs. JSON

### Beispiel XML

```
<?xml version="1.1"?>
  <message>Welcome to the BIA system</message>
```

### Beispiel JSON

```
{
  "message": "Welcome to the BIA system"
}
```

(Eigene Darstellung)

Auf der folgender Tabelle 9 werden beide Formate in Bezug auf spezifische Aspekte hin unterschieden. Gemäss diesem Vergleich wird das optimalere Format für die Implementierung angewendet.

Tabelle 9: Vergleich von XML und JSON

Bereich	XML	JSON
Branchenübernahme	Wurde zu einem Unternehmensstandard und wird von vielen Frameworks unterstützt	Ist relativ neu auf dem Markt und unterstützt keine Schemas
Anwendbarkeit	Die Daten in einer gewissen Struktur besitzen, um Metadaten definieren zu können	Um lediglich die Daten lückenlos via Internet zu transportieren
Lesbarkeit	Einfach zum lesen	Einfach zum lesen
Lernkurve	Schwierig zum erlernen	Einfach zum erlernen
Orientierung	Dokumentorientiert	Datenorientiert
Metadaten	Anhand den «Tags» grosse Metadaten	Minimale Metadaten
Unterstützte Frameworks	Wird von vielen Frameworks unterstützt	Wird von vielen Frameworks unterstützt
Datentypen	Jeder Datentyp wird unterstützt	Es werden nur klassische Daten gespeichert, Text und Nummern
Einfaches Zuordnen in die objektorientierte Programmierung	Die Struktur von XML erschwert das Zuordnen	Einfache Zuordnung, da JSON datenorientiert ist
Speicherkapazität/Performance	Benötigt Aufgrund der Metadaten grosse Speicherkapazität	Benötigt eine sehr geringe Speicherkapazität
Geschwindigkeit	Langsam	Schnell

Quelle: eigene Darstellung in Anlehnung an (Zunke &amp; D'Souza, 2014, S. 257-259)

Die Verwendung von JSON ist für die Implementierung des Prototypen der Schweizerischen Mobiliar besser geeignet. Der Verwendungszweck von JSON stimmt mit den Anforderungen des neuen Systems überein. Zudem weist JSON eine bessere Performance auf, sowohl auf den Speicherbedarf als auch auf der Flexibilität und der Geschwindigkeit.

## 6. Implementationsphase

Die Implementierungsphase wird in diesem Abschnitt der Bachelorarbeit schrittweise beschrieben. Als erster Schritt wird die Vorgehensweise dokumentiert. Dabei werden alle Vorkehrungen erläutert, die angewendet wurden, bevor die Implementation des Prototypen begann. Anschliessend werden alle entwickelten Funktionalitäten anhand vom Datenbankschema, Code Ausschnitte und Screenshots des Prototypen vermittelt.

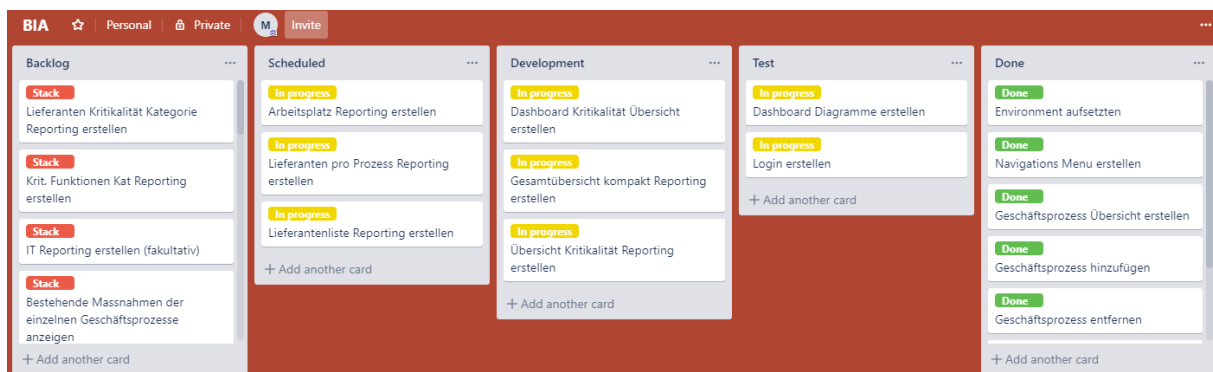
### 6.1. Vorgehensweise Projektmanagement

Mittels der Analyse des bisherigen Systems und den Anforderungen der Schweizerischen Mobiliar an das neue System, wurde ein «Product Backlog» erstellt. Dies ist eine priorisierte Auflistung aller Funktionalitäten und Arbeitsschritten, welche für die Entwicklung des neuen Produktes angewendet wird. Dabei werden der Zeitaufwand und Arbeitsumfang für jeden Task geschätzt. In der Implementationsphase ist der Product Backlog nie vollständig, da dieser jeder Zeit nachgeführt werden kann (Donig & Mühlbauer, 2011, S. 1-3). Während dem Projektverlauf können mehrere Teilfunktionalitäten auftauchen oder bestehende Funktionen sich ändern, welche in dem Product Backlog nachgeführt werden kann. Im ANHANG V befindet sich der detaillierter Product Backlog des neuen BIA Systems

Damit der Product Backlog stets auf dem neusten Stand ist, findet wöchentlich ein Iterations-Meeting zusammen mit dem Abteilungsleiter der Abteilung BCM statt. In den wöchentlichen Meetings werden dem Kunden als erstes die Planung der vergangenen Woche aufgezeigt. Anschliessend wird dem Arbeitgeber das bestehende Programm präsentiert und die geplanten Funktionalitäten der vergangenen Woche aufgezeigt. Nach der Präsentation wird der Product Backlog überarbeitet und die Funktionen für nächste Woche nach den Wünschen des Abteilungsleiters eingeplant. Der ANHANG X zeigt für jede Iterationswoche die definierten Funktionalitäten. Dabei wurde mit dem Kunden zusammen jede Funktionalität in kleinere Aufgaben aufgeteilt und zeitlich eingeplant. Durch das konstruktive Feedback des Abteilungsleiters konnte das Produkt exakt so entwickelt werden, wie sich dies der Auftraggeber vorgestellt hat. Die wöchentlichen Meetings gewährleisten dem Kunden kontinuierliche Integration in der Implementationsphase. Dadurch konnte sein Feedback immer in der nachfolgenden Woche berücksichtigt werden oder im Product Backlog als einen neuen Tasks adaptiert werden.

Neben dem Product Backlog werden parallel noch alle Tasks in einem Kanban Dashboard verwaltet. Das Kanban Dashboard unterstützt den Entwicklungsprozess, um die Tasks immer einem bestimmten Zustand zuordnen zu können. Wie in der Abbildung 29 ersichtlich ist, wird das Dashboard in fünf Spalten aufgeteilt. In der ersten Spalte «Backlog» sind alle Tasks vom Product Backlog hinterlegt. Die zweite Spalte «Scheduled» dient dazu, die nächsten zu implementierenden Tasks zu planen. Die Spalte «Development» wird für die aktuellen Tasks verwendet, welche sich in der Implementationsphase befinden. In der vierten Spalte «Test» werden alle Tasks hinterlegt, bei denen noch Tests durchgeführt werden müssen, damit diese als erledigt deklariert werden können. In der letzten Spalte «Done» wird dann ein Task hinterlegt, wenn die Funktionalität abgeschlossen wurde. Damit ein Task als abgeschlossen bezeichnet werden kann, muss dieser jede Spalte durchlaufen haben, ohne eine davon zu überspringen. Dadurch wird gewährleistet, dass jede Funktionalität geplant, durchgeführt und anschliessend ausführlich getestet wird. Bei den Spalten Scheduled, Development und Test sollten nie mehr als fünf Tasks gleichzeitig hinterlegt sein. Ansonsten besteht die Gefahr, dass Funktionalitäten bei der Entwicklungsphase hängen bleiben, so dass schliesslich keine Funktionalität erfolgreich abgeschlossen wird.

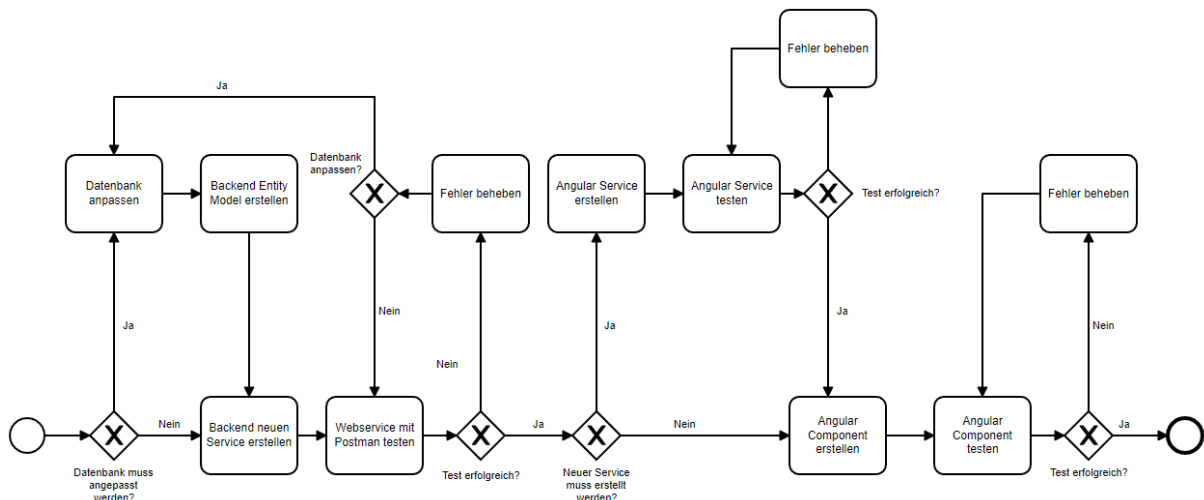
Abbildung 29: Kanban Dashboard



(Eigene Darstellung)

## 6.2. Vorgehensweise Implementierung

Abbildung 30: Vorgehensweise Implementierung BPMN



Eigene Darstellung und erstellt mit: <https://bpmn.io/>

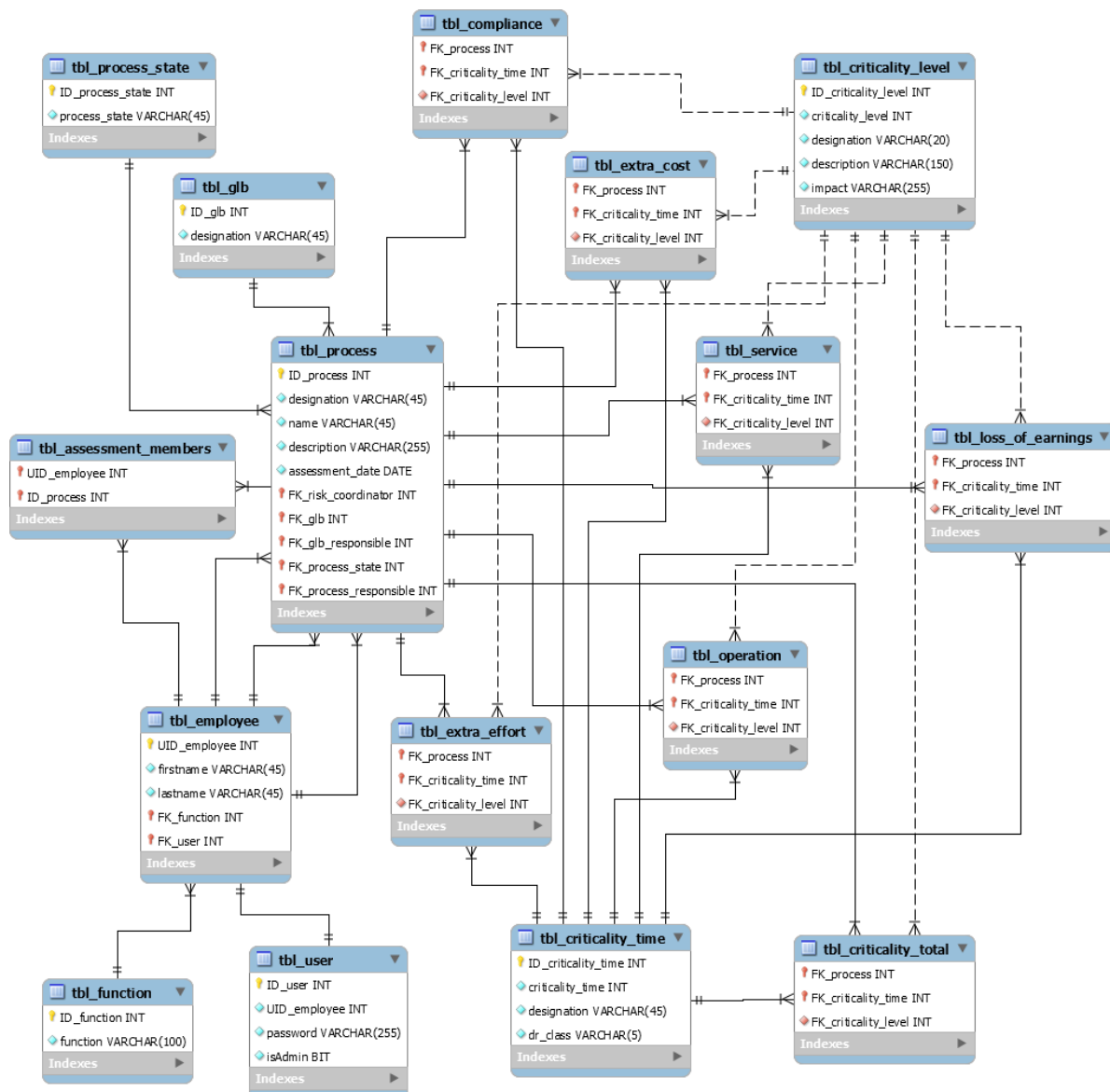
Anhand des **Business Process Model and Notation (BPMN)** Models der Abbildung 30, wird die Vorgehensweise der Implementationsphase graphisch dargestellt. BPMN wird verwendet, um einen Ablauf graphisch so darzustellen, dass dieser sowohl für die Entwickler als auch für die Geschäftsleute nachvollziehbar ist (White, 2004, S. 1-3).

Jeder Task des Product Backlogs wird in gleicher Art und Weise durchgeführt wie die Abbildung 30 aufzeigt. Als erster Schritt wird das Datenbanksystem angepasst, falls dies notwendig ist. Daraufhin wird der Webservice im Backend implementiert und mit der Software Postman<sup>20</sup> getestet. Anhand diesem Test kann festgestellt werden, ob der Webservice wie erwartet funktioniert. Bei fehlerhaften Testiterationen muss dementsprechend das Datenbankschema oder der Webservice angepasst werden. Ansonsten wird in Angular, falls noch nicht vorhanden, ein Service erstellt, welcher auf die Ressourcen des Backend zugreift. Zuletzt wird eine Komponente in Angular erstellt, welche die verlangten Daten in einem Internet Browser anzeigt. Nach der Adaption jeder Funktionalität steht der Test der bereits bestehenden Anwendungen im Zentrum. Bei fehlerhafter Untersuchung müssen passende Massnahmen getroffen werden.

<sup>20</sup> Quelle: <https://www.getpostman.com/>

### 6.3. Datenbankschema

Abbildung 31: Datenbankschema BIA System



(Eigene Darstellung)

Für die Datenbankmanagementschicht wird die Datenbank der Abbildung 31 erstellt. Diese Datenbank wird verwendet, um die Daten des BIA Systems persistent zu speichern. Die Tabellen «tbl\_loss\_of\_earnings», «tbl\_extra\_cost», «tbl\_extra\_effort», «tbl\_compliance», «tbl\_service» und «tbl\_operation» werden verwendet, um die Kritikalität darzustellen. Bei jeder dieser Tabelle werden die Spalten «FK\_process» und «FK\_criticality\_time» als Primärschlüssel der Tabellen definiert. Dies hat zur Folge, dass jeweils nur ein Prozess in einer bestimmten Zeit angelegt werden kann, ohne die Gefahr, Redundanzen zu erstellen. Die Tabelle «tbl\_criticality\_total» wird ausschliesslich für die Reportings verwendet.

## Passwort Hash

In der Tabelle «tbl\_user» wird das Passwort des Anwenders als ein «PBKDF2» Hash gespeichert. Diese Hashfunktion wird verwendet, damit das Passwort sicher gespeichert wird und nicht als Klartext in der Datenbank existiert. Durch die Verwendung des Salt-Mechanismus schützt sich der Hash vor Brute-Force und Rainbow-Tablet Angriffen. Das Ziel des Algorithmus besteht darin, den Vorgang langsam und sicher zu machen, um sich vor Angriffen zu schützen. Aber er sollte dennoch so schnell sein, um nicht die Benutzerfreundlichkeit des Systems zu schwächen (Dharmadasa, 2017).

### 6.4. Hibernate Verbindung mit dem DBMS

Die Verbindung zwischen dem Business Layer und dem Data Access Layer wird in der **Project Object Model (POM)** Datei im Maven Projekt deklariert. Diese Datei enthält alle Abhängigkeiten, um einen Zugriff auf den SQL Server zu erhalten und Hibernate zu verwenden. Hibernate ist ein frei verfügbares ORM-Framework. Die erstellten Objekte in J2EE können durch Anmerkungen in der Datenbank abgebildet werden (Bauer & King, 2005, S. 30-35).

Die folgende Abbildung 32 zeigt die Verbindungsdatei mit dem DBMS. Die sogenannte «persistence.xml» Datei wird verwendet, um die Objekte und die Verbindung mit dem DBMS zu deklarieren. Damit die Webservices Daten mit der Datenbank austauschen können, wird jeweils ein Persistence-Context verwendet. Der Persistence-Context greift auf die Persistence-Unit zu, um den Zugriff mit dem DBMS aufzubauen. Die Benennung der Persistence-Unit (rote Markierung) muss mit der Bezeichnung des Persistence-Context übereinstimmen.

Abbildung 32: Implementation persistence.xml

```
<persistence-unit name="biaservice" transaction-type="RESOURCE_LOCAL">
  <class>dao.Process</class>
  <class>dao.Function</class>
  <class>dao.ProcessState</class>
  <class>dao.GLB</class>
  <class>dao.Employee</class>
  <class>dao.User</class>
  <class>dao.CriticalityLevel</class>
  <class>dao.CriticalityTime</class>
  <class>dao.LossOfEarnings</class>
  <class>dao.ExtraCost</class>
  <class>dao.ExtraEffort</class>
  <class>dao.Compliance</class>
  <class>dao.Service</class>
  <class>dao.Operation</class>
  <class>dao.CriticalityTotal</class>
  <properties>
    <property name="javax.persistence.jdbc.driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
    <property name="javax.persistence.jdbc.url" value="jdbc:sqlserver://localhost:2301;databaseName=MOBI_bia" />
    <property name="javax.persistence.jdbc.user" value="mobi_admin" />
    <property name="javax.persistence.jdbc.password" value="mobi_admin" />
  </properties>
</persistence-unit>
```

(Eigene Darstellung – J2EE)

## Hibernate Objekte

Bei den Verbindungen zwischen zwei oder mehreren Tabellen wird der Abruf-Typ (in Englisch «FetchType») definiert. Bei diesem Typ kann zwischen Lazy oder Eager ausgewählt werden. Der wesentliche Unterschied zwischen diesen Typen besteht darin, dass der Typ Lazy die Daten der Verbindungstabelle nur abrufen, falls diese benötigt werden. Eager ruft die Daten der Verbindungstabelle direkt beim Aufruf der aktuellen Tabelle ab. Bei der Implementierung des Prototypen wird aus Performance Gründen der Typ Lazy ausgewählt. Die Abbildung 33 zeigt ein Beispiel eines Objektes, welches durch Hibernate-Anmerkungen mit der Datenbank verbunden ist. Bei der «OneToMany» Anmerkung wird der Abruf-Typ auf Lazy gesetzt. Wenn der Anwender das Objekt «Process» aufruft, wird diese Verbindung dann genutzt, falls dies nötig ist.

Abbildung 33: Implementation Entity

```
@Entity
@Table(name="tbl_process_state")
public class ProcessState {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="ID_process_state")
    private int id_process_state;

    @Column(name="process_state")
    private String process_state;

    @OneToMany(mappedBy = "process_state", cascade = CascadeType.ALL, fetch=FetchType.LAZY)
    private Set<Process> process_stateConn;

    public ProcessState() {}

    getter
    setter
}
```

(Eigene Darstellung – J2EE)

## 6.5. HTTP-Methoden

Die Bereitstellung der Webservices wird im Backend basierend auf JAX-RS implementiert. Java API for RESTful Web Services (JAX-RS) bietet mehrere Schnittstellen und Anmerkungen für die Entwicklung des BIA System in J2EE. Das Frontend, basierend auf Angular, greift anschliessend über eine HTTP-Verbindung auf die Ressourcen zu, um die Daten zu erhalten. In diesem Abschnitt der Abhandlung wird zuerst gezeigt, wie im Backend unterschiedliche HTTP-Webservices erstellt werden. Im Anschluss zeigt ein Codeausschnitt im Frontend, wie mit dem Server kommuniziert werden kann.

## HTTP-Methoden Anmerkungen

Jede Methode wird durch Anmerkungen (in Englisch «Annotations») definiert. Mithilfe der GET-, POST-, PUT- oder DELETE-Anmerkung wird der Verwendungszweck der Methode festgelegt. Durch die «Path» Anmerkung, kann der spezifische Pfad zur Ressource angegeben werden. Jeder Pfad muss einzigartig sein, damit der Aufruf der Methode klar deklariert ist. Wie im Kapitel «5.2 RESTful Webservices» ausgewertet wird, verwendet der Prototyp das JSON Format. Durch die Anmerkungen «Consumes» und «Produces» wird das Format angegeben, welches die Methode konsumiert und produziert. Als Beispiel, wenn ein Client die Daten im XML Format liefert, wird diese Anfrage vom Server verweigert falls die Anmerkung «Consumes» der Methode angefügt ist. Zuletzt verwenden alle Methoden die eigenentwickelte «Security» Anmerkung. Bei jeder Anfrage des Clients führt diese Anmerkung eine spezifische Methode aus, welche die Zuverlässigkeit des JSON Web Tokens im Header überprüft. Der JSON Web Token wird im Kapitel «6.6 Funktionalitäten» behandelt. Nachfolgend wird jede HTTP-Methode durch ein Codebeispiel detailliert veranschaulicht.

### HTTP-Methode GET

Abbildung 34: Implementierung GET Methode

```
/**
 * get process by the id of a specific process
 * custom security annotation for checking the authorization Bearer JWT
 * @param id_process path parameter with the process id
 * @return Process object
 */
@GET
@Path("/edit/{id}")
@Security
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response getProcessById(@PathParam("id") int id_process) {
    return Response.ok(process_service.getProcessById(id_process)).build();
}
```

(Eigene Darstellung – J2EE)

Die Abbildung 34 zeigt ein Beispiel eines GET-Methoden Aufrufs. In der «Path» Anmerkung wird zusätzlich ein Parameter «{id}» übergeben. Diese Pfadvariable enthält die ID, welche der Client der URL übergibt. Mithilfe der «PathParam» Anmerkung kann diese Variable in einer lokalen Variable gespeichert werden. Die Methode liefert ein «Response» Objekt zurück an den Client. Dieses enthält den HTTP Status-Code 200 durch die Verwendung der Methode «ok()». Zusätzlich wird dem Client der verlangte Prozess als ein JSON Objekt übergeben.

## HTTP-Methode POST

Abbildung 35: Implementation POST Methode

```
/**
 * get the new data to create a new process
 * custom security annotation for checking the authorization Bearer JWT
 * @param data JSONObject containing the new process data
 * @return response ok code 200 with success message
 * @throws ParseException throw exception by errors of the SimpleDateFormat
 */
@POST
@Path("/")
@Security
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response saveNewProcess(JSONObject data) throws ParseException{
    //extract data
    //create new process object
    //call process service to save new process into database
    //call criticality service to save new criticality of the process

    String message = "Der Prozess " + data.get("designation") + " wurde erfolgreich hinzugefügt";
    return Response.ok(gson.toJson(message)).build();
}
```

(Eigene Darstellung – J2EE)

Die POST-Methode der Abbildung 35 besitzt als Übergabeparameter ein JSON Objekt. Die Abbildung zeigt lediglich die Arbeitsschritte als Kommentar auf. Dieses JSON Objekt erhält der Server vom Client, welcher alle Daten des neuen Geschäftsprozesses liefert. Diese Daten werden als erster Schritt extrahiert und in einem neuen Prozess Objekt gespeichert. Daraufhin wird der Speicher durch den Prozess Service persistent in der Datenbank gesichert. Die Kritikalität des Prozesses wird mithilfe des Kritikalität Services abgesichert. Zuletzt liefert der Service eine Nachricht als ein JSON Objekt mit dem Status-Code 200 an den Client.

## HTTP-Methode PUT

Abbildung 36: Implementation PUT Methode

```
/**
 * update existing process with the new data
 * custom security annotation for checking the authorization Bearer JWT
 * @param id_process path parameter with the current process id
 * @param data JSONObject containing the new process data
 * @return response ok code 200 with success message
 * @throws ParseException throw exception by errors of the SimpleDateFormat
 */
@PUT
@Path("/{id}")
@Security
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response updateProcess(@PathParam("id") int id_process, JSONObject data) throws ParseException {
    //extract data
    //get reference of existing process by the id_process
    //change data
    //call process service to save data into database
    //call criticality service to save new criticalities into database

    String message = "Der Prozess " + data.get("designation") + " wurde erfolgreich geändert";
    return Response.ok(gson.toJson(message)).build();
}
```

(Eigene Darstellung – J2EE)

Die Abbildung 36 zeigt eine PUT-Methode, die verwendet wird, um Daten in der Datenbank zu verändern. Die Abbildung zeigt lediglich die Arbeitsschritte als Kommentar auf. Neben dem JSON Objekt erhält die Methode, wie bei der beschriebenen GET-Methode, die Prozess-ID als Pfadvariable. Die Daten des JSON Objektes werden gleich wie mit der POST-Methode behandelt. Jedoch wird kein neues Prozess Objekt generiert, sondern die Referenz des existierenden Objekts verändert.

## HTTP-Methode DELETE

Abbildung 37: Implementation DELETE Methode

```
/**
 * delete existing process
 * custom security annotation for checking the authorization Bearer JWT
 * @param id_process path parameter to delete the process by id
 */
@DELETE
@Path("/{id}")
@Security
@Consumes(MediaType.APPLICATION_JSON)
public void deleteProcess(@PathParam("id") int id_process){
    //remove all foreign keys of the criticalities
    criticality_service.removeCurrentCriticalities(id_process);
    //remove existing process
    process_service.deleteProcessById(id_process);
}
```

(Eigene Darstellung – J2EE)

Wie in der Abbildung 37 ersichtlich ist, besitzt die «Path» Anmerkung den gleichen Wert wie bei der beschriebenen PUT-Methode. Aufgrund der unterschiedlichen HTTP-Methode kann bei diesem Beispiel der gleiche Pfad verwendet werden. Durch die Pfadvariable erhält die Methode die Prozess-ID des Objektes, welches aus der Datenbank entfernt werden soll.

## HTTP-Methode Angular

Abbildung 38: Implementation HttpClient Angular

```
private processBaseUrl = 'http://localhost:8080/biaservice-0.0.1/api/processes';

// header for post and put requests
private httpOptions = {
  headers: new HttpHeaders({
    'Content-Type': 'application/json'
  })
};

constructor(public http: HttpClient) {}

// POST add new process
// process object in body
// response of server (ok message)
addNewProcess(post) {
  return this.http.post(this.processBaseUrl, JSON.parse(JSON.stringify(post)), this.httpOptions);
}

// PUT update existing process
// new process object in body & id of existing process as uri
// response of server (ok message)
updateProcess(id: any, process) {
  return this.http.put(this.processBaseUrl + '/' + id, JSON.parse(JSON.stringify(process)), this.httpOptions);
}

// DELETE remove existing process
// id of process as uri
deleteProcess(id: any) {
  return this.http.delete(this.processBaseUrl + '/' + id);
}

// get all processes
// response: list of processes
getProcesses() {
  return this.http.get<Process[]>(this.processBaseUrl + '/overview');
}
```

(Eigene Darstellung – Angular)

Um Webservices in Angular aufzurufen, wird der HttpClient, wie in der Abbildung 38 ersichtlich ist, verwendet. Eine Anfrage kann durch die definierte URL an den Server gesendet werden. Die POST- und PUT-Methoden liefern ein JSON Objekt im Body mit, welcher die Daten des Prozesses enthalten. Die PUT- und DELETE-Methoden fügen zusätzlich der URL noch die Prozess ID als ein Pfadparameter hinzu. Bei den POST- und PUT-Methoden werden zusätzlich im Header der Inhaltstyp definiert. Zu Grunde der Feststellung im Kapitel «5.2 RESTful Webservices» wird in diesem Projekt das JSON Format verwendet. Durch das eingebundene JSON Web Token Modul «@auth0/angular-jwt» in Angular, wird jeder Anfrage vom HttpClient ein Authorization-Header mit dem JSON Web Token generiert. Der JSON Web Token wird im nächsten Kapitel «6.6 Funktionalitäten» behandelt.

## Security Filter

Abbildung 39: Implementation Security Filter

```
/**
 * security filter which using the Security annotation to check if the token is secure
 * check for containing the Bearer in the token
 * each method which contains the the same annotation can only access the method with a valid
 * authorization header
 * reference for the jwt part: https://antoniogoncalves.org/2016/10/03/securing-jax-rs-endpoints-with-jwt/
 * @param context request context container for the request
 */
@Override
public void filter(ContainerRequestContext context) {
    //get the authorization header
    String authResponse = context.getHeaderString(HttpHeaders.AUTHORIZATION);

    //check if header exists and starts with authorization Bearer
    if(authResponse == null || !authResponse.startsWith(RootResource.BEARER)){
        //header not exists
        LOGGER.log(Level.SEVERE, "Token not contains a Bearer");
        throw new NotAuthorizedException("Sie sind nicht Berechtigt");
    }

    //get the token without the authorization Bearer
    String token = authResponse.substring(RootResource.BEARER.length()).trim();

    try {
        //parse the token with the own secret key to check if the token is valid
        Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token);
        //token is valid
        LOGGER.log(Level.INFO, "Token is valid");
    } catch (Exception e) {
        //token is invalid
        LOGGER.log(Level.SEVERE, "Token is invalid");
        context.abortWith(Response.status(Response.Status.FORBIDDEN).build());
    }
}
```

(Eigene Darstellung – J2EE)

Die Abbildung 39 zeigt den Codeausschnitt, der bei jeder Methode ausgeführt wird, welche die Anmerkung «Security» hinzufügt. Die Methode überprüft den JSON Web Token auf die Integrität. Falls kein Authorization-Header existiert oder der Token nicht mit dem Bearer Schema übereinstimmt, wird der Vorgang unterbrochen. Bei einem gültigen Bearer Schema wird der Token mit dem eigenen Sicherheitsschlüssel auf deren Korrektheit überprüft. Wenn die Überprüfung einen Fehler aufweist, ist der Token ungültig und der Vorgang wird unterbrochen. Der JSON Web Token wird im nächsten Kapitel «6.6 Funktionalitäten» behandelt.

## 6.6. Funktionalitäten

In diesem Teil der Bachelorarbeit werden alle umgesetzten Funktionalitäten des Prototypen beschrieben. Bei jeder Funktionalität wird anhand eines Codeausschnittes und des Ergebnisses der Komponente, die Methodik wiedergegeben.

### Geschäftsprozess hinzufügen

Abbildung 40: BIA System - Geschäftsprozess hinzufügen

The screenshot shows a web browser window with the URL `localhost:4200/process/add`. The application has a red header bar with the BIA logo and the text 'die Mobiliar' and 'Julia Fischer'. A sidebar on the left contains navigation links: DASHBOARD, PROZESS (selected), IT, and REPORTINGS. The main content area is titled '← Geschäftsprozess hinzufügen'. It contains a form with the following fields:

- Bezeichnung: Text input
- Prozessname: Text input
- Prozessbeschreibung: Text area
- Risikokoordinator: Dropdown menu with 'Person auswählen'
- GLB/SF: Dropdown menu with 'GLB auswählen'
- GLB/SF-Verantwortlicher: Dropdown menu with 'Person auswählen'
- Prozess-Verantwortlicher: Dropdown menu with 'Person auswählen'
- Assessment Datum: Date input with format 'tt.mm.jjjj'
- Assessment Teilnehmer: Dropdown menu with 'Person auswählen' and a '+' icon
- Status: Dropdown menu with 'Status auswählen'

A red 'Speichern' button is located at the bottom of the form.

(Eigene Darstellung)

Über das reaktive Formular der Abbildung 40 wird ein neuer Geschäftsprozess in das System persistent gespeichert. Reaktive Formulare besitzen einen flexiblen Ansatz, da das Formular in der Komponente durch Objekte definiert wird. Der Vorgang, um die Daten an das Formular zu übertragen, wird mit dem Paradigma «reactive programming» bezeichnet (Gawryś, 2017). Der Hauptvorteil dieses Formulars liegt darin, dass Unit Tests für die Formulare erstellt werden können. Die erstellten Unit Tests werden im Kapitel «7.2 Angular Tests» dokumentiert. Das reaktive Formular exportiert die Werte der Inputfelder in das JSON Format, so dass dies daraufhin an das Backend übermittelt werden kann.

Abbildung 41: Implementation Reaktive Formular

```
form = new FormGroup( controls: {
  designation: new FormControl( formState: '', validatorOrOpts: [Validators.required, FormValidators.noSpace]),
  name: new FormControl( formState: '', Validators.required),
  description: new FormControl( formState: '', Validators.required),
  riskCoordinator: new FormControl( formState: null, Validators.required),
  glbSF: new FormControl( formState: null, Validators.required),
  glbSFResponsible: new FormControl( formState: null, Validators.required),
  processResponsible: new FormControl( formState: null, Validators.required),
  assessmentDate: new FormControl( formState: '', Validators.required),
  assessmentParticipants: new FormArray( controls: [], Validators.required),
  state: new FormControl( formState: null, Validators.required)
});
```

(Eigene Darstellung – Angular)

Die Abbildung 41 zeigt den Codeausschnitt des reaktiven Formulars. Das Formular setzt sich aus mehreren «FormControl» zusammen. Dies ist ein einzelnes Feld des Formulars (Gawryś, 2017). Jedes Element des Formulars wird durch «Validators.required» als ein Pflicht-Feld deklariert. Das Feld «designation» wird zusätzlich durch einen individuellen Validator «FormValidators.noSpace» adaptiert. Die folgende Abbildung 42 zeigt den Codeausschnitt, welcher das Feld nach einem Leerschlag überprüft. Falls das Element einen Leerschlag enthält, setzt diese Methode die Variable «noSpace» auf Wahr. Infolgedessen zeigt das FormControl «designation» der Abbildung 41 eine Fehlermeldung den Anwendern an.

Abbildung 42: Implementation individueller Validator

```
// check if the designation contains a space
static noSpace(control: AbstractControl): ValidationErrors | null {
  if ((control.value as string).indexOf(' ') >= 0) {
    // input field contains a space
    return {noSpace: true};
  }
  return null;
}
```

(Eigene Darstellung – Angular)

Beim Versuch den neuen Geschäftsprozess in der Datenbank zu speichern, wird zuerst kontrolliert, ob ein Geschäftsprozess mit der gleichen Bezeichnung bereits existiert. Die Abbildung 43 zeigt diese Kontrolle durch die Methode «checkProcessExists(...)». Falls der Prozess bereits in der Datenbank auftaucht, wird der Anwender aufgefordert, eine neue Bezeichnung anzugeben. Ansonsten wird der Prozess persistent gespeichert.

Abbildung 43: Implementation Geschäftsprozess hinzufügen

```

// add new process
// check if the process already exists in the db
this.serviceProcess.checkProcessExists(this.form.value.designation)
  .subscribe(response => {
    if (!response) {
      // process not exists in the db
      // save new process in db
      this.serviceProcess.addNewProcess(this.form.value)
      .subscribe(res => {
        // redirect to the process list and send a success message
        this.router.navigate(['/process'], {
          queryParams: { message: res }
        });
      });
    } else {
      // process already exists in the db
      // set error message for the designation input field
      this.designation.setErrors({
        alreadyExists: true
      });
    }
  });
}

```

(Eigene Darstellung – Angular)

Wie die Abbildung 44 veranschaulicht, kann jeder Geschäftsprozess mehrere Beurteilungs-Teilnehmer besitzen. Mithilfe der Dropdown-Liste kann ein Mitarbeiter ausgewählt werden. Automatisch wird die Mitarbeiter ID in das Form-Array hinzugefügt. Ein Mitarbeiter kann durch das Kreuz wieder entfernt werden. Diese Funktionalität ermöglicht jedem Prozess, eine unbestimmte Anzahl Teilnehmer anzugeben.















Abbildung 44: BIA System - Assessment Teilnehmer verwalten

Assessment Teilnehmer	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input style="width: 100%;" type="text"/> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>U998443 ✕</span> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>U443569 ✕</span> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>U115883 ✕</span> </div> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>U145982 ✕</span> </div> </div>
Status	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Entfällt</span> <span>▼</span> </div> </div> <div style="background-color: #f00; color: white; padding: 5px; text-align: center; margin-top: 5px;">       Speichern     </div>

(Eigene Darstellung)

## Geschäftsprozess Übersicht

Abbildung 45: BIA System - Geschäftsprozess Übersicht

Bezeichnung	Name	Beschreibung	Assessment
03_SI	Sicherheit	Sicherheit im Bereich der Informationstechnologie. Sicherstellen der Einhaltung der Gesetze für den Zugriff auf externe und interne Systeme.	06.12.2019  
04_V	Verkauf	Vorbereitungen auf das Verkaufsgespräch mit den dazugehörigen IT-Anwendungen. Alle Dokumente müssen ausgefüllt werden	03.04.2018  
07_PM	Produktmanagement Service	Service um die Instandhaltung des Produktmanagement. Vergleich der unterschiedlichen Produkten	06.06.2019  
07.1_SM	Servicemanagement	Instandhaltung der Services die für das Produktmanagement verwendet wird.	12.12.2017  
08_KuS	Kunden Service	Allgemeine Kundenanfragen bezüglich der Kundenbetreuung definieren.	23.09.2016  
10_FR	Finanzrechnung	Die Kontrolle über die Finanzen müssen über deren Korrektheit überprüft werden.	04.08.2019  
10_FR_2	Finanzrechnung	Die Finanzrechnung muss stets kontrolliert werden. Zusätzlich ist es notwendig eine Bilanz aufzustellen	04.08.2019  

(Eigene Darstellung)

Die Abbildung 45 zeigt die Übersicht aller existierender Geschäftsprozesse in der Datenbank. Dabei werden lediglich nur die Bezeichnung, Name, Beschreibung und das Beurteilungsdatum jedes Prozesses angegeben. Zusätzlich kann durch das Bearbeitungssymbol (rote Markierung) der Geschäftsprozess bearbeitet werden. Dafür wird in Angular ein Router-Link erstellt, welcher auf die Bearbeitungsseite verlinkt. Die Bearbeitungsseite in der Funktionalität «Geschäftsprozess bearbeiten» beschrieben. Mithilfe des Löschsymbols (blaue Markierung) kann ein bestehender Geschäftsprozess entfernt werden. Ein neuer Geschäftsprozess wird durch das Pluszeichen hinzugefügt. Dabei gelangt der Anwender zur Ansicht der zuvor beschriebenen Funktionalität.

## Geschäftsprozess suchen

Das Suchfeld wird verwendet, um die Prozessliste zu filtern. Die Suchfunktion unterstützt die Mitarbeiter, den gewünschten Prozess schnellstmöglich zu finden. Für die Suchfunktion wird eine eigene Pipe in Angular erstellt. Eine Pipe repräsentiert einen Filter, welcher angewendet wird, um die Daten beim Anzeigen in einer Komponente zu transformieren. Angular bietet mehrere Standard Pipes zur Verfügung, welche angewendet werden können. Jedoch hat der Entwickler die Möglichkeit, eigene Pipes zu entwickeln (Rylan, 2016). Die Abbildung 46 zeigt einen Codeausschnitt einer selbstentwickelten Pipe. Bei jeder Änderung des Suchfeldes wird die Methode «transform(...)» ausgeführt, welche das Prozess Array im Frontend nach dem Suchtext filtert.

Abbildung 46: Implementation Custom Pipe

```
// compare the search input value with the table data
transform(value: any[], actualObject: any, searchText?: string): any[] {
  // return empty object if the table is undefined
  if (!value || !actualObject) {
    return [];
  }

  // return all process if the search field is empty
  if (!searchText) {
    return value;
  }

  // filter process by search text of search input field
  return actualObject.filter(e => e.designation.toUpperCase().includes(searchText.toUpperCase()) ||
    e.name.toUpperCase().includes(searchText.toUpperCase()) ||
    e.description.toUpperCase().includes(searchText.toUpperCase()) ||
    this.datePipe.transform(e.assessment_date, format: 'dd.MM.yyyy').startsWith(searchText) ||
    this.datePipe.transform(e.assessment_date, format: 'dd.MM.yyyy') === searchText ||
    this.datePipe.transform(e.assessment_date, format: 'dd.MM.yyyy').indexOf(searchText) > -1 ||
    this.datePipe.transform(e.assessment_date, format: 'dd.MM.yyyy').endsWith(searchText));
}
```

(Eigene Darstellung – Angular)

## Geschäftsprozess sortieren

Durch Klick auf den Titel der jeweiligen Spalte wird die Geschäftsprozessliste absteigend oder aufsteigend sortiert. Dabei wird eine globale Variable «order» vom Typ Boolean verwendet, welcher den aktuellen Zustand speichert. Die folgende Abbildung 47 zeigt die Sortierungsfunktion. Als Übergabeparameter wird der Schlüsselname des Prozessobjektes übergeben, nach welchem das Array sortiert wird.

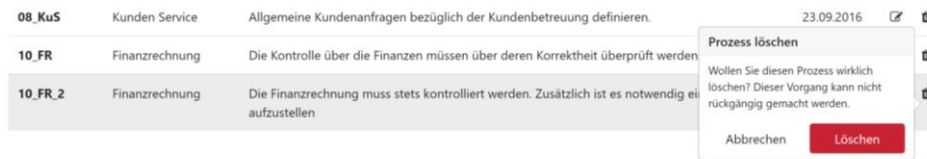
Abbildung 47: Implementation Sortierung Tabelle

```
// sort process list by the key
sortListByKey(key: string) {
  // toggle order by double clicking on column name
  if (this.order) {
    // order asc
    this.processes.sort( compareFn: (a, b) => {
      const x = a[key];
      const y = b[key];
      return ((x < y) ? -1 : ((x > y) ? 1 : 0));
    });
  } else {
    // order desc
    this.processes.sort( compareFn: (a, b) => {
      const x = a[key];
      const y = b[key];
      return ((x > y) ? -1 : ((x < y) ? 1 : 0));
    });
  }
  // toggle variable
  this.order = !this.order;
}
```

(Eigene Darstellung – Angular)

## Geschäftsprozess entfernen

Abbildung 48: BIA System - Bestätigungsmeldung Prozess entfernen

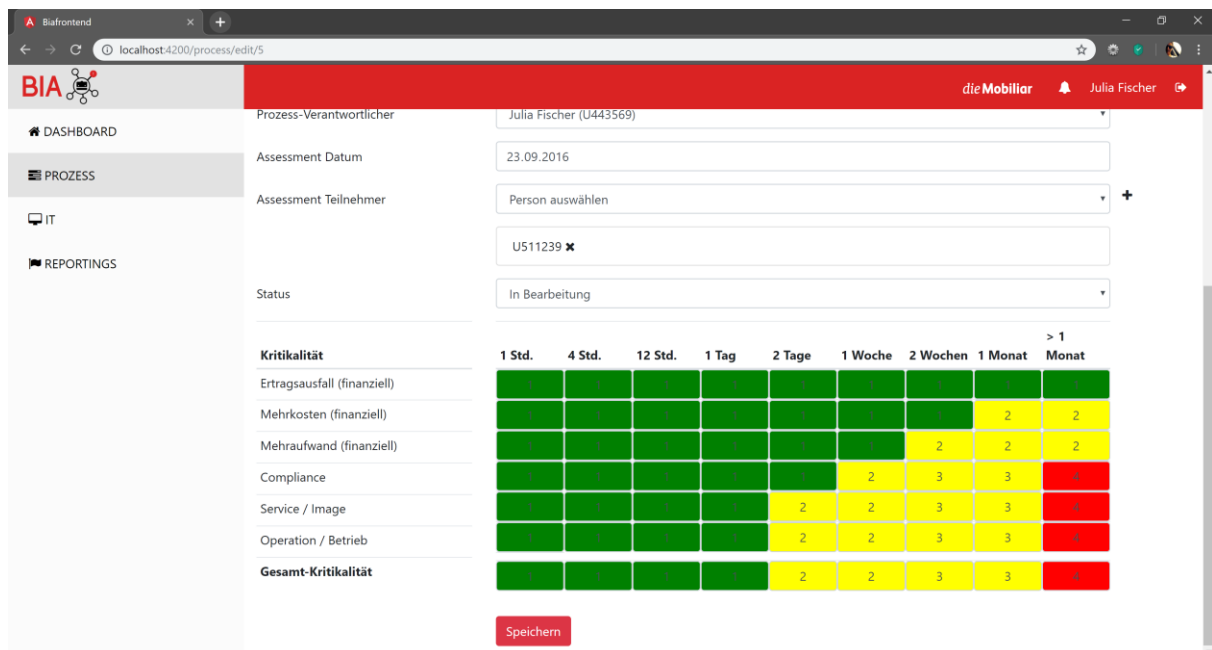


(Eigene Darstellung)

Damit ein Anwender nicht irrtümlich einen Geschäftsprozess entfernen kann, wird dieser durch eine Bestätigungsmeldung darauf hingewiesen, dass dieser Vorgang nicht rückgängig zu machen ist. Beim Drücken auf «Löschen», welches in der Abbildung 48 dargestellt wird, wird der Geschäftsprozess unwiderruflich in der Datenbank entfernt. Diese Bestätigungsmeldung wurde durch die Direktive «angular-confirm-popover» in Angular integriert. Dieses eingebundene Modul basiert auf Bootstrap.

## Geschäftsprozess bearbeiten

Abbildung 49: BIA System - Geschäftsprozess bearbeiten



(Eigene Darstellung)

Wie bereits im Kapitel «4.2 Frontend» beschrieben, liegt der Vorteil von Angular darin, dass die Komponenten wiederverwendet werden können. Aus diesem Grund kann, wie in der Abbildung 49 ersichtlich, dasselbe reaktive Formular verwendet werden wie bei der Funktionalität «Geschäftsprozess hinzufügen». Das reaktive Formular wurde dabei nur einmalig implementiert und mit beiden Komponenten integriert. Bei der Ansicht

«Geschäftsprozess bearbeiten» werden automatisch alle Felder mit den Daten aus der Datenbank gefüllt. In diesem Formular hat der Anwender die Möglichkeit die Bezeichnung des Prozesses anzupassen oder die bestehende Bezeichnung beizubehalten. Falls der Benutzer eine neue Bezeichnung definiert, kontrolliert das System ob dieser noch nicht in der Datenbank hinterlegt ist. So wird die Erstellung von Redundanzen verhindert. Zusätzlich wird in diesem Formular die Kritikalität adaptiert, welche in der nachfolgender Funktionalität beschrieben wird.

## Geschäftsprozess Kritikalität

Das Kritikalitätslevel wird für alle sechs Kategorien und für alle neun Kritikalitätszeiten definiert. Bei den sechs Kategorien werden die drei finanziellen und die drei nicht-finanziellen Auswirkungen bezeichnet, welche im Kapitel «1.2 Business Impact Analysis» beschrieben werden. Dabei wird für jeden Eintrag eine Dropdown-Liste verwendet. Aus dieser Liste kann das Level selektiert werden, wie in der Abbildung 50 ersichtlich ist. Die letzte Spalte «Gesamt-Kritikalität» repräsentiert jeweils den höchsten Wert der Spalte und kann nicht von dem Anwender verändert werden. Das System wurde so implementiert, dass die Kritikalitätszeiten und Kritikalitätslevels dynamisch sind. Aus diesem Grund kann ein Administrator eine neue Kritikalitätszeit definieren, welche automatisch in die Liste integriert wird.

Abbildung 50: BIA System - Auswahl Kritikalitätslevel

1 Std.	4 Std.	12 St
1	1	
1	1	
1	2	
1	3	
1	4	
1	5	
1	1	

(Eigene Darstellung)

Die Abbildung 51 zeigt den Codeausschnitt für die Anzeige der Kritikalitäten auf den Clients. Der Client erhält vom Webservices ein sechsstelliges Objekt, welches eine For-Schleife durchläuft. Für jedes Objekt wird durch eine weitere For-Schleife iteriert, um für jede Kritikalitätszeit das Kritikalitätslevel anzugeben. Dabei wird für jeder Eintrag eine Dropdown-Liste erstellt und das passende Level selektiert. Wenn ein Anwender für einen Eintrag ein neues Kritikalitätslevel definiert, wird die Methode «changeTotal()» ausgeführt. Dabei wird der Komponente mitgeteilt, welche Kategorie an welcher Position ein neues Level besitzt und verändert werden muss. Diese Daten werden jedoch lokal auf dem Client gepflegt und

erst wenn der Anwender das Formular mit «Speichern» abschliesst, werden die Daten in der Datenbank persistent gespeichert.

Abbildung 51: Implementation Anzeige der Kritikalität

```
<!-- insert select for each time and each category with already selected level from the stored process -->
<tr *ngFor="let criticality of criticalities| keyvalue; let i = index ">
  <td *ngFor="let entry of criticality.value" class="cr td">
    <!-- changeTotal to reload list and calculate the highest value of the column for the total criticality row -->
    <select class="editSelection form-control"
      [style.background-color]="notification.value === 1 ? 'green' : (notification.value === 2 || notification.value === 3) ? 'yellow' : 'red'"
      (change)="changeTotal(i, entry.cr_criticality_time, notification.value)"
      #notification>
      <option *ngFor="let level of nbrOfLevels"
        [style.background-color]="level.criticality_level === 1 ? 'green' : (level.criticality_level === 2 || level.criticality_level === 3) ? 'yellow' : 'red'"
        [selected]="level.criticality_level === entry.cr_criticality_level">
        {{level.criticality_level}}
      </option>
    <!-- execute method to initialize the total criticality row -->
    {{changeTotal(i, entry.cr_criticality_time, notification.value)}}
    </select>
  </td>
</tr>
```

(Eigene Darstellung – Angular)

## JSON Web Token

JSON Web Token (JWT) definiert einen offenen Standard RFC7519. JWT wird verwendet, um die Daten sicher zwischen Backend und Frontend auszutauschen. Die Ansprüche (in Englisch «claims») werden im JSON Format übertragen. Durch die digitale Signatur (JWS) können die Daten verifiziert und vertraut werden. Jeder Token besteht aus drei Teilen

**Header:** Beschreibt die Struktur im JSON Format, die für den Token angewendet wird. Zum Beispiel der Algorithmus, der angewendet wird, um die Signatur zu erstellen.

**Payload:** Repräsentiert den Body. Der Inhalt wird im JSON Format übermittelt.

**Signatur:** Beinhaltet den Authentifizierungscode, der auf den Header und Payload angewendet wird.

Alle drei Ausschnitte sind Base64-URL zusammengesetzte Zeichenketten, welche durch einen Punkt getrennt werden (Shingala, 2018, S. 1-4). Die Struktur des Token wird wie folgt aussehen.

xxxxx.yyyyy.zzzzz  
Header.Payload.Signatur

Die Abbildung 52 zeigt ein Beispiel eines verifizierten Token des entwickelten Prototypen. Im Header wird der verwendete Algorithmus «HS256» angegeben. Der Payload enthält den Inhalt, welcher dem Client übermittelt wird. Das Subjekt «sub» ist die Mitarbeiter ID des angemeldeten Benutzers. Zusätzlich werden der Vorname «firstname», der Nachname «lastname» und die Information über den Administrator «isAdmin» übergeben. Das Element «iat» (in Englisch «issued at») definiert den Zeitpunkt, an dem der Token generiert wurde.

Zuletzt wird durch das Element «exp» (in Englisch «expiration time») der Zeitpunkt definiert, an dem der Token ausgelaufen und nicht mehr gültig ist. Die rote Markierung in der Abbildung 52 beinhaltet den Sicherheitscode, welcher im Backend definiert wurde, um den Header und den Payload zu verschlüsseln. Die Signatur kann nur verifiziert werden, wenn das Sicherheitswort bekannt ist.

Abbildung 52: BIA System - Beispiel eines JSON Web Token

### Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI0NDM1Njk  
iLCJpYXQiOiE1NjI4NzI1MDAsImV4cCI6MTU2Mjg  
3NzkwMCwiZmlyc3RuYW11IjoIc2NoZXIiLCJpc0FkbWluIjpmYWx  
zZX0.pug_STPvik7dZan-  
Oh1TLah4xtChHHC15zr53-OdmuQ
```

### Decoded

EDIT THE PAYLOAD AND SECRET

#### HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256"
}
```

#### PAYLOAD: DATA

```
{
  "sub": "443569",
  "iat": 1562872500,
  "exp": 1562877900,
  "firstname": "Julia",
  "lastname": "Fischer",
  "isAdmin": false
}
```

#### VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret base64 encoded
) ☐ secret base64 encoded
```

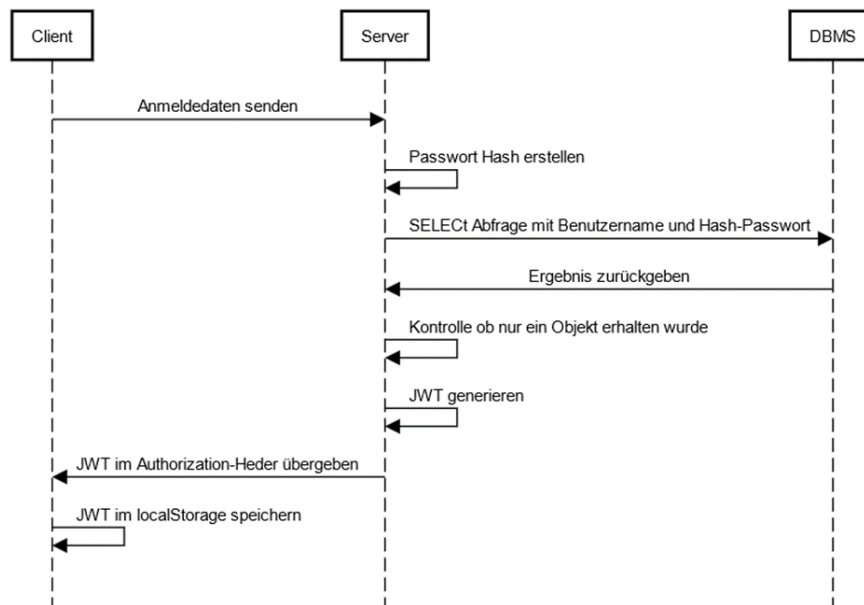
☑ Signature Verified

SHARE JWT

Eigene Darstellung und erstellt mit: <https://jwt.io/>

## AblaufszENARIO JWT

Abbildung 53: Sequenzdiagramm JWT Generierung



Eigene Darstellung und erstellt mit: <https://sequencediagram.org>

Ein Benutzer ist nur dann berechtigt, die Daten des Backends einzusehen, wenn jeder Request einen gültigen JSON Web Token im Header mitliefert. Wie in dem Sequenzdiagramm der Abbildung 53 ersichtlich ist, erhält der Client beim Anmelden in das BIA System den Zugriffstoken. Der Client sendet eine HTTP-POST Anfrage an den Server und übergibt dem Body ein JSON Objekt mit dem Benutzernamen und dem Passwort. Der Webservice auf dem Server erstellt aus dem erhaltenen Passwort einen PBKDF2-Hash. Anschliessend sendet der Server eine Select-Anfrage an die Datenbank, um zu kontrollieren, ob ein Benutzer mit der Mitarbeiter ID und dem Passwort existiert. Das Datenbank Management System sendet das Ergebnis der Anfrage zurück an den Server. Daraufhin wird die Einzigartigkeit des Ergebnisses überprüft und falls diese bestätigt ist, wird ein JWT mit allen benötigten Daten generiert. Der Token wird anschliessend als Antwort auf die Anfrage des Clients im Authorization-Header zurückgeliefert. Angular speichert den erhaltenen Token im lokalen Speicher, damit dieser bei jedem Request an den Server im Header übergeben werden kann.

Durch das beschriebene AblaufszENARIO gewährleistet der Prototyp, dass nur Anwender, die sich erfolgreich angemeldet haben, Zugriff auf die Webservices besitzen. Wenn ein Client versucht, eine Anfrage an den Server zu senden, ohne dabei im Header einen gültigen JSON Web Token zu übergeben, wird die Anfrage vom Server verweigert. Der Authorization-Header basiert auf einem «Bearer»-Schema. Die Abbildung 54 zeigt den Codeausschnitt, der im Backend angewendet wird, um einen JWT zu generieren. Aus Sicherheitsgründen

verlangt die Schweizerische Mobiliar, dass jeder Anwender automatisch nach 90 Minuten abgemeldet wird.<sup>21</sup> Um diese Funktionalität zu gewährleisten, wird der Ablaufzeitpunkt im Token mit der Methode «setExpiration(...)» definiert.

Abbildung 54: Implementation JWT Generierung

```
/**
 * generate JWT token with the data of the logged in user
 * HS256 is used as signature algorithm
 * @param user logged in user object
 * @param firstname first name of the user
 * @param lastname last name of the user
 * @return token as string
 * @throws UnsupportedOperationException
 */
public String generateJWTToken(User user, String firstname, String lastname) throws UnsupportedOperationException{
    //generate token with the data
    //token expire after 90 minutes
    String token = Jwts.builder().setSubject(Integer.toString(user.getEmp_user()))
        .setIssuedAt(new Date()).setExpiration(toDate(LocalDateTime.now().plusMinutes(90)))
        .claim("firstname", firstname)
        .claim("lastname", lastname)
        .claim("isAdmin", user.getIsAdmin())
        .signWith(SignatureAlgorithm.HS256, RootResource.SECRET_KEY.getBytes("UTF-8"))
        .compact();

    return token;
}
```

(Eigene Darstellung – J2EE)

## Benutzerabmelden

Ein Benutzer kann sich über das Logout-Icon abmelden. Dabei wird im Hintergrund von Angular lediglich der Token aus dem lokalen Speicher entfernt. Ein erneutes Anmelden im BIA System gewährleistet dem Mitarbeiter wiederkehrender Zugriff. Bevor der Token ausläuft, wird der Benutzer informiert, dass er automatisch nach fünf Minuten abgemeldet wird und alle Daten speichern muss. Die Abbildung 55 zeigt den Codeausschnitt von Angular, welcher alarmiert, falls die Differenz des Erstellungsdatums und des Ablaufdatums 85 Minuten sprengt. Diese Kontrolle wird in einem Intervall von einer Minute durchgeführt.

<sup>21</sup> Gemäss Iterations-Meeting mit Abteilungsleiter BCM Herr In-Albon Damian (26.06.2019)

Abbildung 55: Implementierung Ablaufdatum JWT

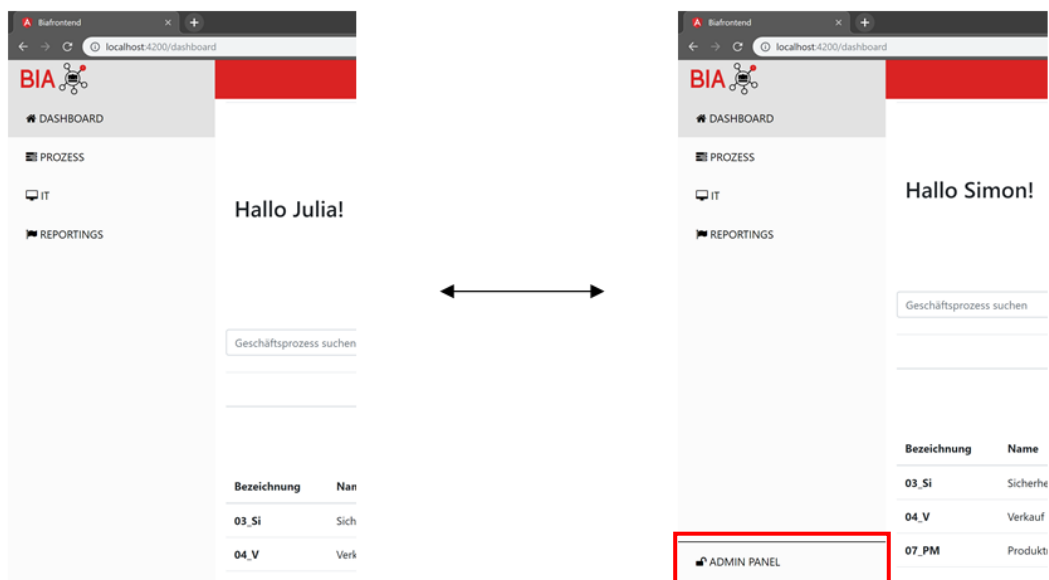
```
ngOnInit() {
  // check each minute the expire date of the token
  this.sub = interval(60000)
    .subscribe((val) => {
      this.expiredAt = this.authService.checkExpired();
      if (this.expiredAt != null) {
        // calculate the difference between the actual datetime and the expire datetime (in minutes)
        this.now = new Date();
        this.diffDateMinutes = Math.round((((this.now - this.expiredAt) % 86400000) % 3600000) / 60000);

        // if the difference is greater than 85 minutes inform the user that he will be logged out automatically soon
        if (this.diffDateMinutes >= 85) {
          // inform user that in 5 minutes the token will be expired, so he can save all changes
          alert(this.alertMessage);
          // unsubscribe message that the message only appears once
          this.sub.unsubscribe();
        }
      }
    });
}
```

(Eigene Darstellung – Angular)

## Navigation

Abbildung 56: BIA System - Navigation Mitarbeiter und Administrator



(Eigene Darstellung)

In der Navigation kann zwischen dem Dashboard, den Geschäftsprozessen, der IT-Liste und dem Reporting gesteuert werden. Wie in der Abbildung 56 ersichtlich ist, besitzt ein Administrator (rechtes Fenster) noch zusätzlich den Navigationspunkt «Admin Panel». Damit ein Anwender ohne Adminrechte nicht über die URL auf das Admin Panel zugreifen kann, werden unterschiedliche Services erstellt. Die Abbildung 57 zeigt die Navigation über die URL. Durch das Element «path» wird die aufgerufene URL deklariert. Mit dem Element «component» wird die Komponente angegeben, welche die Seite anzeigen soll. Mithilfe des Elementes «canActivate» können Klassen angegeben werden, welche das Interface «canActivate» implementiert haben. Die Klasse «AuthService» kontrolliert, ob der Anwender angemeldet ist. Bis auf das Login wird dieser Wächter in jeder Navigation angewendet. So

wird verhindert, dass ein Anwender ohne Berechtigung, Zugriff auf den Prototypen hat. Die Klasse «AdminAuthService» wird erstellt, um zu kontrollieren, ob der angemeldete Benutzer Administratorrechte besitzt. Das Admin Panel wird nur angezeigt, wenn beide «canActivate» Konditionen wahr sind. Wenn ein Service einen falschen Wert liefert, wird die Navigation verweigert und abgebrochen.

Abbildung 57: Implementation Navigation

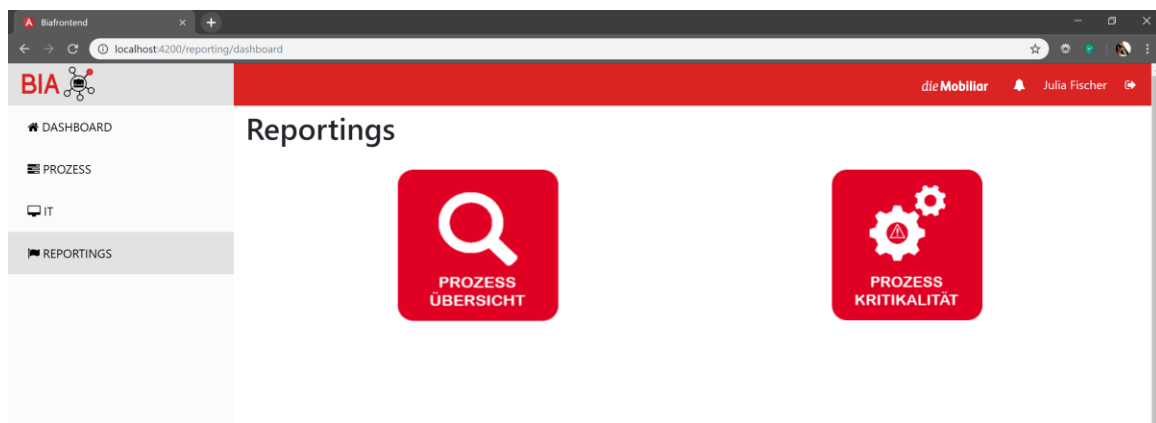
```
RouterModule.forRoot({ routes: [
  {path: '', pathMatch: 'full', redirectTo: 'dashboard'},
  {path: 'dashboard', component: DashboardComponent, canActivate: [AuthService]},
  {path: 'reporting/dashboard', component: ReportingDashboardComponent, canActivate: [AuthService]},
  {path: 'reporting/process/overview', component: ReportingProcessOverviewComponent, canActivate: [AuthService]},
  {path: 'reporting/process/category', component: ProcessCategoryComponent, canActivate: [AuthService]},
  {path: 'process', component: ProcessComponent, canActivate: [AuthService]},
  {path: 'admin', component: AdminComponent, canActivate: [AuthService, AdminAuthService]},
  {path: 'login', component: LoginComponent},
  {path: 'process/add', component: AddProcessComponent, canActivate: [AuthService]},
  {path: 'process/edit/:id', component: EditProcessComponent, canActivate: [AuthService]},
  {path: 'no-access', component: NoAccessComponent},
  {path: '**', component: NotFoundComponent}
]},
```

(Eigene Darstellung – Angular)

## Reportings

Damit der Anwender eine Übersicht über alle Reportings hat, wird ein Dashboard erstellt. Das Dashboard verwendet die gleichen Icons, welche im bisherigen BIA System verwendet werden. Zwei dieser Reporting-Icons sind auf der Abbildung 58 ersichtlich. Der Grund für die Verwendung der gleichen Icons besteht darin, dass die Mitarbeiter vertraut mit den Icons in der bisherigen System sind. Durch den Gebrauch der entsprechenden Icons wird der Wiedererkennungswert für die Mitarbeiter erhöht. Für die Anzeige der Icons wird das «Gridsystem» von Bootstrap verwendet, damit sich die Icons automatisch an die Breite des Fensters anpassen.

Abbildung 58: BIA System - Reportings Dashboard



(Eigene Darstellung)

## Reporting Gesamt-Kritikalität nach Kategorie

Im entwickelten Prototypen wurden zwei Reportings erstellt. Das erste Reporting ist die Standard-Gesamtübersicht der Geschäftsprozesse. Beim zweiten Reporting wird das gleiche verwendet, welches jedoch noch zusätzlich nach der Kategorie gruppiert wird. Die Abbildung 59 repräsentiert das Reporting nach den Kategorien. Bereits im Kapitel «6.3 Datenbankschema» wird erwähnt, dass die Tabelle «tbl\_criticality\_total» nur für Berichtszwecke verwendet wird. Die Daten werden nach dem Bearbeiten eines Geschäftsprozesses in diese Tabelle persistent gespeichert. Diese Tabelle wird anschliessend für die Reporting verwendet, da der Client mit einem Aufruf alle Prozesse und die Gesamt-Kritikalität jedes Prozesses erhalten kann. Angenommen der Prototyp verwendet diese Tabelle nicht, müsste für jeden Prozess noch ein zusätzlicher Zugriff auf die Datenbank erstellt werden, um die Kritikalität der Kategorien zu erhalten. Anschliessend müsste für jeden Geschäftsprozess die Gesamt-Kritikalität berechnet werden. Dieses beschriebene Vorgehen senkt die Performance rapide, da der Aufruf bei mehreren Geschäftsprozessen enorm viel Zeit beansprucht. Infolgedessen wird dem Prototyp diese Tabelle hinzugefügt.

Die Spalte «Krit.-Kat.» enthält die Kategorie des Prozesses. Alle Geschäftsprozesse werden in drei Kategorien unterteilt. Der Kategorie A werden alle Prozesse zugeteilt, die bereits bis zum zweiten Tag schweren Schaden verursachen. Alle Geschäftsprozesse, welche bis zum zweiten Tag ein Kritikalitätslevel von zwei oder drei aufweisen, werden der Kategorie B zugeteilt. Die übrigen Geschäftsprozesse werden der niedrigsten Kategorie C zugeteilt.

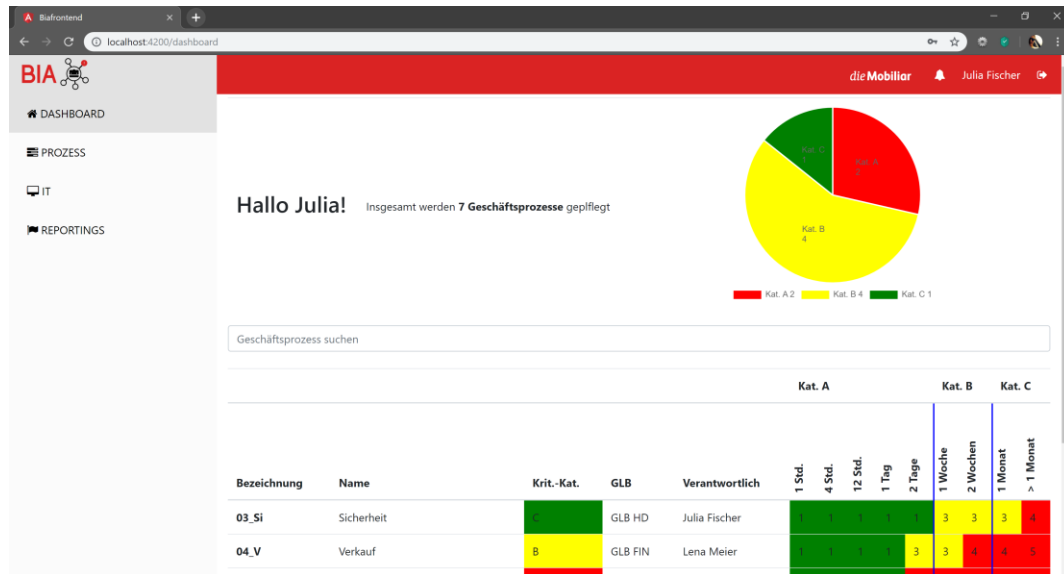
Abbildung 59: BIA System - Reporting Geschäftsprozess Kategorie

Bezeichnung	Name	Krit.-Kat.	GLB	Verantwortlich	Kat. A			Kat. B			Kat. C		
					1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	> 1 Monat
03_Si	Sicherheit	A	GLB HD	Julia Fischer	3	3	3	3	3	3	3	3	3
04_V	Verkauf	B	GLB FIN	Lena Meier	3	3	3	3	3	3	3	3	3
07_PM	Produktmanagement Service	A	GLB MM	Daniel Furrer	3	3	3	3	3	3	3	3	3
07.1_SM	Servicemanagement	B	GLB	Simon Müller	3	3	3	2	2	2	3	3	3
08_KuS	Kunden Service	B	GLB AM	Julia Fischer	3	3	3	2	2	2	3	3	3
10_FR	Finanzrechnung	A	GLB VER	Daniel Furrer	3	3	3	2	4	4	4	4	5

(Eigene Darstellung)

## Dashboard

Abbildung 60: BIA System - Dashboard



(Eigene Darstellung)

Durch ein Dashboards besitzen alle Anwender des Prototypen eine Übersicht sämtlicher Geschäftsprozesse. Wie in der Abbildung 60 ersichtlich, wird das Reporting nach den Kategorien aufgelistet. Das Kreisdiagramm veranschaulicht eine Gruppierung der unterschiedlichen Kategorien. Für diese visuelle Darstellung wurde die sogenannte ChartJS Library in Angular eingebunden. Diese Library ermöglicht die Darstellung, aller Daten in unterschiedlichen Diagrammen. Basierend auf HTML5 wird ChartJS durch ein einzelnes Canvas «Tag» dargestellt. Das Design und die Lieferung der Daten werden im Kontroller an das Canvas übergeben. Die Abbildung 61 zeigt die Konfiguration der Komponente, um das Kreisdiagramm anzuzeigen. Die Variable «pieCartType» kann der verwendete Typ des Diagrammes angegeben werden. Das Kreisdiagramm wird als eine Komponente in dem Dashboard wiederverwendet. Aus diesem Grund sind die Variablen «catA», «catB» und «catC» als Input-Variablen deklariert. Beim Integrieren der Komponente werden die Daten der Kategorien als «Binding»-Variable übergeben.

Abbildung 61: Implementation ChartJS

```
// ng2-charts directive for chart.js
// reference: https://valor-software.com/ng2-charts/
public pieChartOptions: ChartOptions = {
  responsive: true,
  legend: {
    position: 'bottom',
  },
  plugins: {
    datalabels: {
      formatter: (value, ctx) => {
        const label = ctx.chart.data.labels[ctx.dataIndex];
        return label;
      },
    },
  },
};

public pieChartLabels: Label[];
public pieChartData: number[];
public pieChartType: ChartType = 'pie';
public pieChartLegend = true;
public pieChartPlugins = [pluginDataLabels];
public pieChartColors = [
  {
    backgroundColor: ['red', 'yellow', 'green'],
  },
];

ngOnInit() {
  // set the labels and data for the chart
  // data receive from the input fields of the dashboard
  this.pieChartLabels = ['Kat. A\n' + this.catA, 'Kat. B\n' + this.catB, 'Kat. C\n' + this.catC];
  this.pieChartData = [this.catA, this.catB, this.catC];
}
```

Eigene Darstellung (Angular) in Anlehnung an <https://valor-software.com/ng2-charts/>

## 7. Testphase

In diesem Abschnitt der Bachelorarbeit werden die Teste beschrieben, welche für den Prototypen angewendet werden. Die Schweizerische Mobiliar wendet viele Arten von unterschiedlichen Testen an, um sicherzustellen, dass jeder Commit direkt in die Produktion übergeben werden kann. Um das System zu testen werden für Frontend und Backend passende Teste erstellt und durchgeführt. Nachfolgend werden alle angewendeten Teste im J2EE Projekt und in Angular beschrieben.

### 7.1. J2EE Teste

Wie im ANHANG VI erwähnt wird, führt die Schweizerische Mobiliar beim Backend JUnit 5, Service Teste und Performance- und Lastteste mit JMeter durch. Im entwickelten Prototypen werden Service/Integration Teste durchgeführt. Wenn das System grosser Last ausgesetzt wird, sind Performance- und Lastteste sinnvoll. Jedoch arbeiten am Prototypen nie mehr als fünf Mitarbeiter zur selben Zeit.<sup>22</sup> Folglich ist die Implementierung von Performance- und Lastteste überflüssig. Bis anhin kann im entwickelten System kein Unit Test durchgeführt werden, da die Logik innerhalb jeder Methode mit dem Datenbank Management System kommuniziert und davon abhängig sind. Aus diesem Grund werden nachfolgend zwei Beispiele von angewendeten Service/Integration Teste beschrieben.

### HTTP-Methode Ressource Test

Abbildung 62: Implementation HTTP-Method Test

```
/**
 * test the getProcessById method of the ProcessResource class
 * check if the response is not null and the response status is 200
 * check if we really obtain the right Process object
 * @throws Exception
 */
@Test
public void testGetProcessById() throws Exception{
    Response response = process_resource.getProcessById(1);
    assertEquals(200, response.getStatus());
    assertNotNull(response);

    //check if we obtain the right object
    Process process = (Process) response.getEntity();
    assertEquals(process.getId_process(), 1);
}
```

(Eigene Darstellung – J2EE)

---

<sup>22</sup> Gemäss Iterations-Meeting mit Abteilungsleiter BCM Herr In-Albon Damian (26.06.2019)

Die Abbildung 62 zeigt ein Beispiel eines HTTP-Methoden Tests. Eine Methode der Prozessressource wird gemäss einer spezifischen ID eines Prozesses aufgerufen. Die Rückgabe der Methode liefert ein Response Objekt mit einem existierenden Prozess Objekt zurück. Zuerst wird getestet, ob der Status-Code der Antwort 200 (OK) entspricht. Anschliessend wird überprüft, ob das Objekt Null ist. Falls diese zwei Assertionen erfolgreich sind, wird das Objekt in einem Prozess Objekt gespeichert. Daraufhin wird inspiziert, ob die Prozess-ID übereinstimmt.

## Service JWT Test

Abbildung 63: Implementation JWT Test

```
/**
 * test the generateJWTToken method of the AuthResource class
 * check the creation of the token
 * @throws Exception
 */
@Test
public void testGenerateJWTToken() throws Exception{
    //hash pw to compare it afterwards with the db
    String hashedPW = auth_service.encryptPassword("password");
    User validUser = auth_service.checkUserCredentials("443569", hashedPW);

    String token = auth_resource.generateJWTToken(validUser, "Max", "Muster");

    //check if the token is not null
    assertNotNull(token);
    //check if the size is 1
    assertEquals(1, Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token).getHeader().size());
    //check the SignatureAlgorithm
    assertEquals("HS256", Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token).getHeader().getAlgorithm());
    //check the size of the body
    assertEquals(6, Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token).getBody().size());
    //check the subject of the token
    assertEquals(Integer.toString(validUser.getEmp_user()), Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token).getBody().getSubject());
    //check the expiration date
    assertNotNull(Jwts.parser().setSigningKey(RootResource.SECRET_KEY.getBytes("UTF-8")).parseClaimsJws(token).getBody().getExpiration());
}
```

(Eigene Darstellung – J2EE)

Der Test zur Generierung des JSON Web Token wird in der Abbildung 63 dargestellt. Als erstes wird ein PBKDF2 Hash aus dem Passwort erstellt. Durch die Mitarbeiter ID kann daraufhin das Objekt des Benutzer zurückgegeben werden. Für den Benutzer wird ein Token erstellt, bei welchem verschiedene Assertionen durchgeführt werden. Neben der Kontrolle, ob das Objekt Null ist, wird die Grösse des Headers auf die Einzigartigkeit überprüft. Ein weiterer Test besteht in der Übereinstimmung des verwendeten Algorithmus. Zudem wird der Body des Tokens mit der Grösse verglichen. Zuletzt wird das Subjekt des Token mit der Benutzer ID des angemeldeten Mitarbeiter überprüft und kontrolliert ob die Ablaufzeit nicht überschritten wurde.

## Durchführung der Teste

Insgesamt wurden 37 Teste für das Backend implementiert. Diese Teste werden bei jedem Deployment auf dem WildFly Webserver ausgeführt. Erst wenn alle Teste erfolgreich durchlaufen sind, wird die neue Version des Backends auf den Webserver bereitgestellt. So wird verhindert, dass fehlerhafte Ressourcen auf dem Server freigegeben werden. Beim Erstellen einer neuen Ressource müssen jeweils unterschiedliche Teste erstellt werden, um die Methoden zu testen.

## 7.2. Angular Teste

Wie im ANHANG VI erwähnt wird, führt die Schweizerische Mobiliar bei der Verwendung von Angular im Frontend, Unit Test mit Jest oder Karma und End-To-End (E2E) Teste mit Protractor durch. Bei der Installation von Angular wird automatisch Karma in das System integriert. Aus diesem Grund werden im Frontend mit Karma Unit und Integration Teste erstellt, um die Funktionalitäten zu testen. In diesem Prototyp werden keine E2E Teste durchgeführt, da diese einen grossen Zeitaufwand in Anspruch nehmen und bei kleinen Änderungen an der Benutzeroberfläche bereits viele E2E Teste überarbeitet werden müssten. Zudem legt die Schweizerische Mobiliar grossen Wert auf Unit Teste wie im ANHANG VII ersichtlich. Nachfolgend werden Beispiele von unterschiedlichen Testen aufgezeigt.

### Integration Test: Komponente erstellt

Abbildung 64: Implementation Integration Test: Komponente erstellt

```
let component: DashboardComponent;
let fixture: ComponentFixture<DashboardComponent>;
let serviceCriticality: CriticalityService;

beforeEach(async(() => {
  TestBed.configureTestingModule( ... );

  beforeEach(() => {
    fixture = TestBed.createComponent(DashboardComponent);
    component = fixture.componentInstance;
    serviceCriticality = TestBed.get(CriticalityService);
  });

  // check if the right component is created
  it('should create', () => {
    // create a spy on the getCriticalityCategory method
    spyOn(serviceCriticality, 'getCriticalityCategory').and.returnValue(of([{}]));

    // execute ngOnInit()
    fixture.detectChanges();
    expect(component).toBeTruthy();
  });
});
```

(Eigene Darstellung – Angular)

Die Abbildung 64 zeigt ein Beispiel eines Integration Test in Angular, welches kontrolliert, ob die Komponente erstellt wurde. Die Variable «fixture» dient als ein Wrapper für die Komponente. Dieser Variable wird als eine Instanz der Komponente über das TestBed übergeben. Das TestBed wird verwendet, um ein Testmodul zu erstellen, welches nur in den Unit und Integration Teste verwendet wird. Durch den Aufruf von der Methode «detectChanges()» wird die Initialisierung der Komponente ausgeführt. Diese Abfolge wird vor jedem Test durchgeführt. So wird gewährleistet, dass alle Komponenten neu geladen werden, bevor der Test durchgeführt wird. Wenn die Komponente einen Service aufruft, welcher mit dem Backend kommuniziert, muss ein Spion (spyOn) erstellt werden. Diese

Spione können verwendet werden, um den Aufrufe der Methoden zu testen und zu verfolgen. Dabei kann der Rückgabewert der Methode selber bestimmt werden, um die Methode zu kontrollieren.

## Unit Test: Validation reaktives Formular

Zuerst wird der gleiche Test durchgeführt, wie in der Abbildung 64 aufgelistet. Anschliessend werden mehrere Unit Teste für die Validation des reaktiven Formulars absolviert. Die Implementation dieser Teste wird in der Abbildung 65 dargestellt.

Abbildung 65: Implementation Unit Test Validation reaktives Formular

```
let component: LoginComponent;
let fixture: ComponentFixture<LoginComponent>;

.
.
.

// test if the controls are created
it('should create component with controls', () => {
  expect(component.loginForm.contains('username')).toBeTruthy();
  expect(component.loginForm.contains('password')).toBeTruthy();
});

// test if the name control are required
it('should make the username control required', () => {
  const control = component.loginForm.get('username');
  control.setValue('');
  expect(control.valid).toBeFalsy();
});

// test if the name control contains space
it('should no space in the username control', () => {
  const control = component.loginForm.get('username');
  control.setValue(' ');
  expect(control.valid).toBeFalsy();
});

// test if the password control are required
it('should make the password control required', () => {
  const control = component.loginForm.get('password');
  control.setValue('');
  expect(control.valid).toBeFalsy();
});
```

(Eigene Darstellung – Angular)

Der erste Unit Test kontrolliert das Formular, ob die beiden FormControls in der Komponente erstellt wurde. Die beiden Teste «should make the username control required» und «should make the password control required» fügen in das Inputfeld des Formulars einen leeren Wert ein. Mit der Überprüfung der Validation des Formulars, kann kontrolliert werden, ob die Fehlermeldung bei ungültigen Daten erscheint. Der letzte Test «should no space in the username control» fügt einen Leerschlag in das Benutzernamen-Inputfeld ein. Dieser FormControl enthält eine eigene Validation, welche Leerschläge verweigert. So wird kontrolliert, ob das Formular ungültig ist.

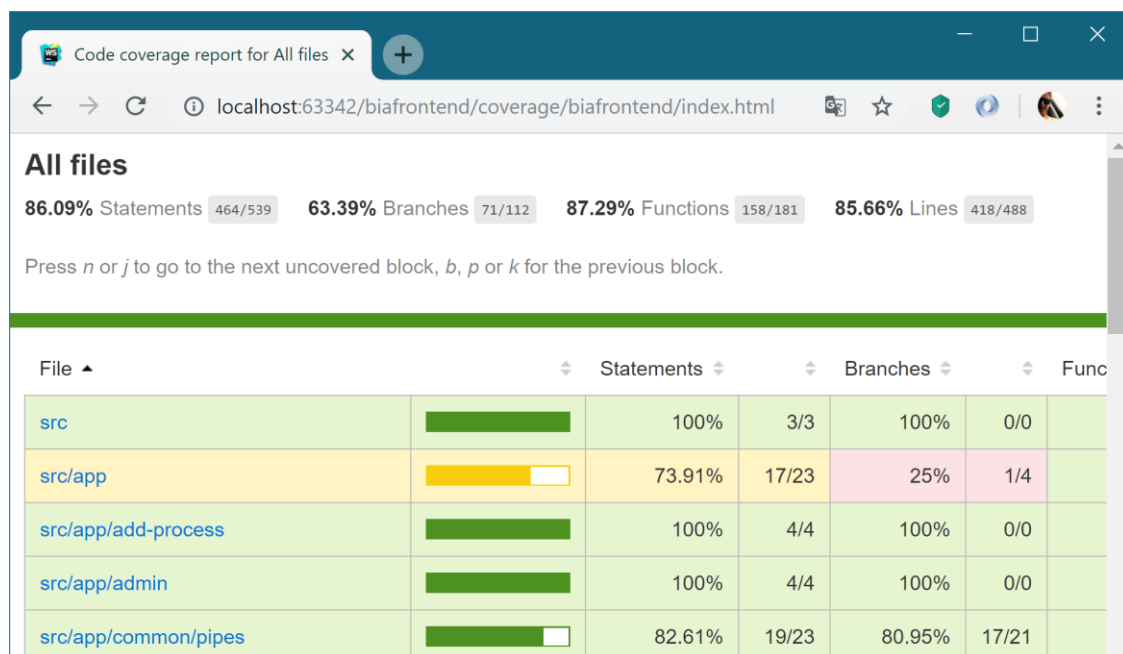
## Durchführung der Tests

Die Tests werden mit Karma durchgeführt. Insgesamt wurden 61 Tests erstellt, welche bei jedem Start des Projektes ausgeführt werden. Nach einer Änderung im Frontend kann der Entwickler alle Tests manuell erneut ausführen, um fehlerhafte Methoden aufzeigen zu lassen.

### 7.3. Code Coverage

Wie im ANHANG VII aufgeführt, sind die Anforderungen der Schweizerischen Mobiliar, dass alle Unit Tests ein Code Coverage von 80% erlangen. In der Abbildung 66 wird das Ergebnis von Karma bezüglich des Code Coverage gezeigt.

Abbildung 66: Code Coverage Tests



(Eigene Darstellung)

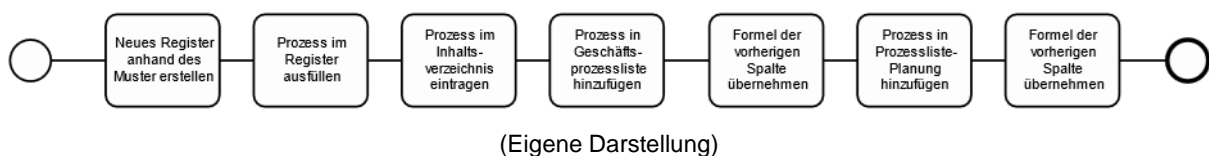
Wie in der Abbildung 66 zu erkennen ist, ist die Prozentanforderung für die Rubrik «Branches» nicht erreicht worden. Für die Rubriken «Statements», «Functions» und «Lines» wurde die Prozentanforderung der Schweizerischen Mobiliar erreicht. Sowohl im Frontend als auch im Backend müssen zusätzliche Tests durchgeführt werden, damit der Prototyp den Anforderungen der Schweizerischen Mobiliar gerecht wird. Erst wenn alle Prozentanforderung erreicht werden, kann der Prototyp in die Produktion überlaufen. Im entwickelten System werden alle Funktionalitäten, die eine POST-, PUT- oder DELETE-Methode vorweisen, bislang noch nicht durch Unit-Tests oder Integration-Tests unterstützt. Damit diese Methoden getestet werden können, müsste eine Testumgebung implementiert werden, welche mit einem Abbild des DBMS nur für Testzwecke verwendet wird.

## 8. Evaluation vom Prototyp

In diesem Kapitel wird die bisherige Lösung dem entwickelten Prototypen gegenübergestellt. Ein Geschäftsprozess hinzuzufügen ist der zeitaufwendigste Arbeitsschritt im bisherigen System. Infolgedessen wird zunächst dieser Arbeitsschritt mit dem entwickelten Prototyp verglichen. Des Weiteren werden die Benutzerfreundlichkeit und die Zusammenarbeit zwischen den Mitarbeitern ausgewertet. Zuletzt wird die Fortführung des Prototypen dem getesteten Produkt der Schweizerischen Mobiliar gegenübergestellt.

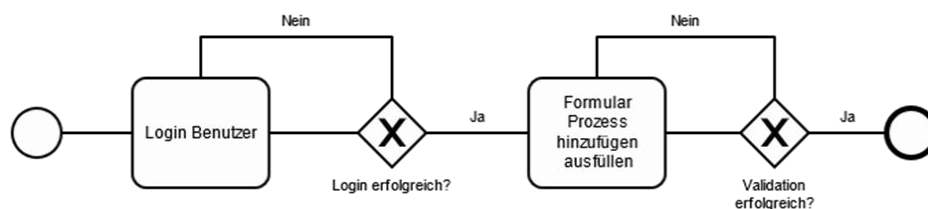
### 8.1. Gegenüberstellung «Geschäftsprozess hinzufügen»

Abbildung 67: BPMN Arbeitsschritte bisherigen System



Im Kapitel «1.3 BIA – bisheriges System» werden detailliert die Arbeitsschritte der Abbildung 67 beschrieben. In der Analyse des Kapitels «2.1 User Experience und Usability BIA System» wird bereits erwähnt, dass dieser Vorgang viele Arbeitsschritte umfasst. Zusätzlich weist dieser Ablauf viele potenzielle Fehlerquellen und Sicherheitslücken auf.

Abbildung 68: BPMN Arbeitsschritte neues System



(Eigene Darstellung)

Die Abbildung 68 zeigt die Arbeitsschritte derselben Funktionalität im Prototypen. Auf den ersten Blick ist erkennbar, dass weniger Arbeitsschritte durchgeführt werden müssen. Mithilfe der Benutzeranmeldung werden die Sicherheitslücken behoben, da nur ein Benutzer einen Geschäftsprozess hinzufügen kann, der auch berechtigt ist. Nach einer erfolgreichen Anmeldung muss der Anwender nur einen Schritt durchführen, um einen Prozess hinzuzufügen. Dieser Prozess wird dann nach erfolgreicher Validation persistent gespeichert. Mit der Validation gewährleistet der Prototyp die Vollständigkeit der Daten. Zudem wird dadurch die Erzeugung von Redundanzen verhindert.

## Evaluation der Risikobehhebung

Im Kapitel «2.7 Risikoanalyse» werden die Risiken der bisherigen Lösung aufgezeigt. Wie die Auswertung der RAT und RMA aufzeigen, befinden sich sechs potenzielle Risiken in der Priorität «eins». Wie bereits in der Auswertung erwähnt, wäre die Behebung dieser Risiken im Prototypen äusserst wichtig. Die Tabelle 10 stellt die Risiken der Priorität «eins» mit der umgesetzten Lösung des Prototypen gegenüber.

Tabelle 10: Evaluation der Risikobehhebung

ID	Potenzielles Problem	Lösung im Prototyp	beseitigt
A	Mehrere Mitarbeiter arbeiten zur gleichen Zeit an zwei unterschiedlichen Versionen des bisherigen Systems	Der Prototyp wird über den Internet Browser verfügbar sein und so arbeiten alle am gleichen System	Ja
B	Die drei Dateien werden nicht von dem gleichen Verzeichnis gestartet	Der Prototyp basiert nicht wie die bisherige Lösung auf Excel-Dateien	Ja
G	Neuer Geschäftsprozess wird falsch hinzugefügt	Durch die Validierung des Formulars ist das Einfügen von ungültigen Daten nicht möglich. Zudem wurde der Arbeitsschritt, um den Prozess im System zu speichern, automatisiert.	Ja
H	Redundanzen werden von einem Mitarbeiter erstellt	Beim Hinzufügen eines neuen Geschäftsprozesses oder beim Bearbeiten eines existierenden Geschäftsprozesses wird immer kontrolliert, ob die Bezeichnung des Prozesses bereits in der Datenbank hinterlegt ist.	Ja
I	Funktionalitäten des Navigationsmenu (Ribbon Menu) funktioniert nicht richtig	Das Ribbon Menu wurde durch eine Seitennavigation ersetzt.	Ja
J	Reporting weisen Bezugsfehler auf	Der Prototyp basiert nicht auf Excel-Dateien und daher wird kein Bezug mehr auf andere Daten erstellt.	Ja

(Eigene Darstellung)

## 8.2. User Experience des Prototypens

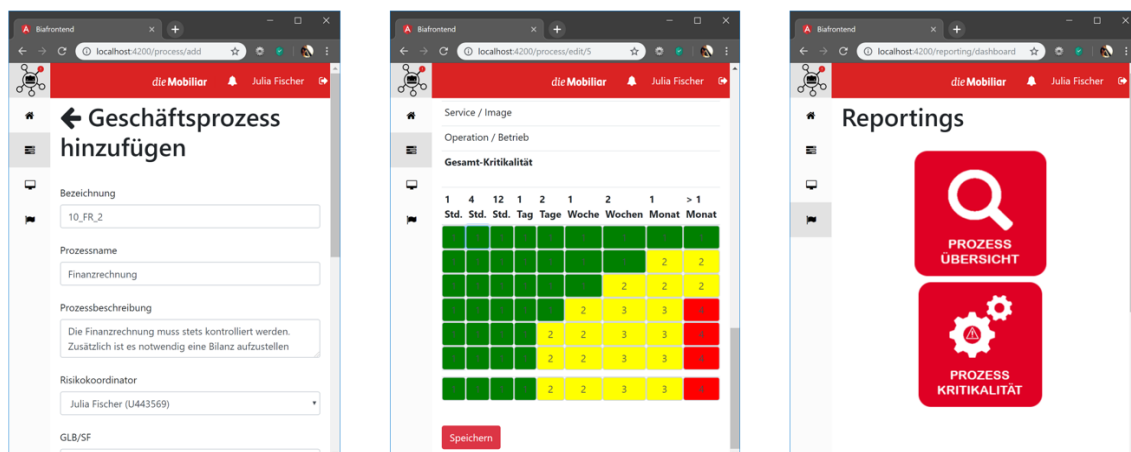
Das Kapitel «2.1 User Experience und Usability BIA System» definiert die Verwendung von User Experience in einem System. UX werden im neuen System durch verschiedene Vorgänge unterstützt. In Angular wird die Library «angular-font-awesome» eingebunden. Font Awesome stellt dem Entwickler Icons zur Verfügung, die durch ein HTML5 «Tag» eingebunden werden können. Zudem wird ein Ladespinner verwendet, welcher immer sichtbar ist, solange der Client auf eine Antwort vom Server wartet. Zusätzlich wird darauf geachtet, dass jede Antwort vom Server nur die nötigen Informationen enthalten. Als Beispiel wird beim Aufruf der Geschäftsprozessübersicht nur lediglich die Mitarbeiter ID des

Prozessverantwortlichem Übergeben, anstatt das ganze Mitarbeiter Objekt. Der Vorteil liegt darin, dass die Ladezeit der Geschäftsprozesse verringert wird, wenn nur eine ID Anstatt ein ganzes Objekt geliefert wird. Die Verringerung der Ladezeit ist hauptsächlich dann ersichtlich, wenn zahlreiche Geschäftsprozesse im System hinterlegt sind. Ausserdem wird der Benutzer über jeden Vorgang informiert. Der Anwender erfährt, wann ein Prozess erfolgreich gespeichert wird oder sobald ein Pflichtfeld ungültig ist. Die Integration von Bootstrap 4 in Angular bietet dem Entwickler die Möglichkeit, ein responsives Design zu gestalten.

## Responsive Design

Die Vorteile der Verwendung eines responsiven Designs werden im Kapitel «4.2 Frontend» beschrieben. Wenn die Ansicht des BIA Systems verkleinert wird, verkleinert sich das Navigationsmenu. Jedes Input-Feld passt sich der Grösse des Fensters an. Dies ermöglicht die Bedienung der Applikation auf kleineren Geräten, wie zum Beispiel einem Tablet. Die Abbildung 69 zeigt drei Ansichten des BIA Systems im responsiven Design.

Abbildung 69: BIA System - Responsive Design



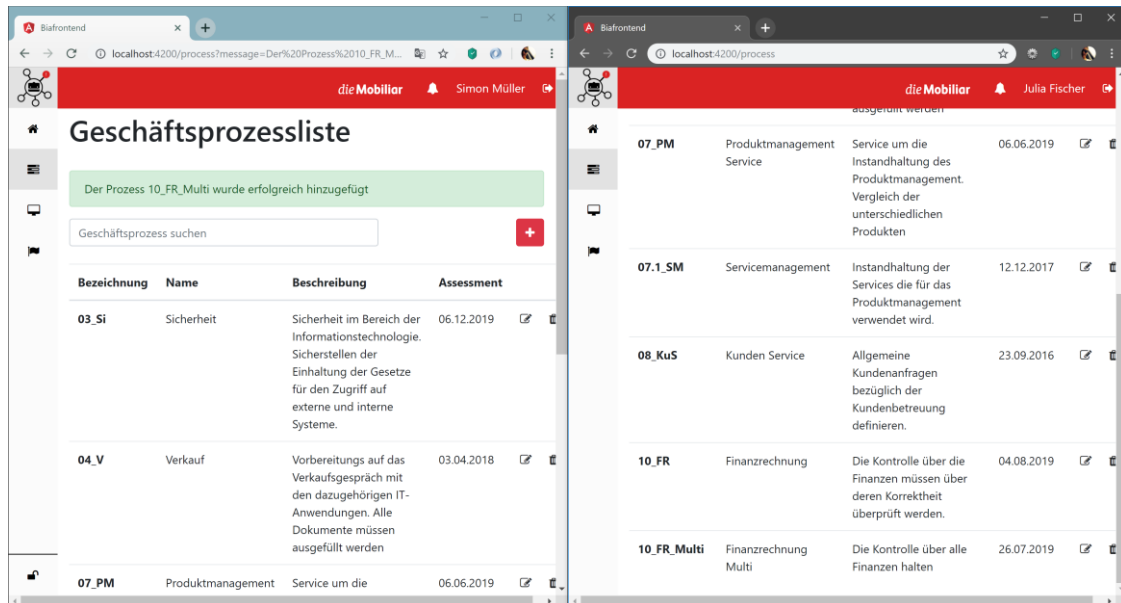
(Eigene Darstellung)

## Multiuser-Funktionalität

Die Anforderungen der Schweizerischen Mobiliar bestanden darin, dass mehrere Mitarbeiter zugleich am Prototypen arbeiten können. Durch die Benutzeranmeldung wird diese Funktionalität gewährleistet. Des Weiteren besteht die Möglichkeit, den Mitarbeitern Benutzerrollen zu verteilen, damit verschiedene Berechtigungen für die Anwendern existieren. Die Abbildung 70 zeigt ein Beispiel der Multiuser-Funktionalität. Dabei ist auf dem linken Fenster ein Administrator angemeldet, welcher einen neuen Geschäftsprozess «10\_FR\_Multi» ins System hinzugefügt hat. Zur selben Zeit ist ein Mitarbeiter auf dem

rechten Fenster angemeldet, welcher in der Übersicht den neuen Geschäftsprozess bearbeiten kann.

Abbildung 70: Multiuser-Funktionalität neues BIA-System



(Eigene Darstellung)

### 8.3. Fortführung neues BIA System

Im Kapitel «3.1 Bestehende Software als Ersatz» wird das Produkt vorgestellt, welches die Schweizerische Mobiliar getestet hat. Wie bereits im Kapitel beschrieben, entspricht der Business Impact Analyse Teil der Software nicht allen Anforderungen der Schweizerischen Mobiliar.

Im «Release Roadmaps» des ANHANG IX wurde die Zeit berechnet, die für die Vervollständigung aller Funktionalitäten des neuen BIA Systems noch benötigt werden. In der Tabelle 11 wird die geplante Implementation mit dem Preis des Produktes «alive-IT Business Continuity Framework» des Unternehmens «Controllit AG» verglichen. Diese Gegenüberstellung soll der Schweizerischen Mobiliar die Entscheidung erleichtern, die Implementierung des Prototypen weiterzuführen. Die Preisliste des Produktes «alive-IT Business Continuity Framework» befindet sich im ANHANG XIII.

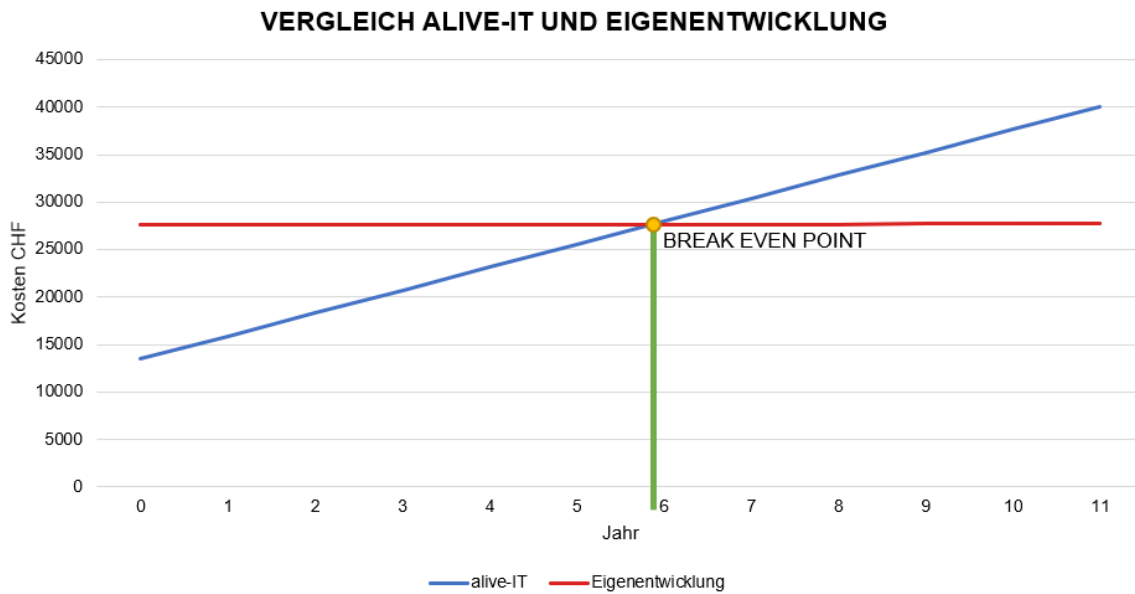
Tabelle 11: Gegenüberstellung Eigenentwicklung und Alive-IT Produkt

Eigenentwicklung BIA System		alive-IT Business Continuity Framework	
Anzahl verbleibenden Wochen	22 Wochen	Lizenz für vier bis 15 gleichlaufende Benutzerkonten (einmalig)	1'950.- CHF
Anzahl Stunden in der Woche	42 Wochen	2. Backup Server bei der Unternehmung (einmalig)	4'000.- CHF
Lohnauszahlung für externer Mitarbeiter (Entwickler)	30.- CHF	instantOLAP (für graphische Berichte / einmalig)	7'500.- CHF
<b>Total</b>	<b>27'720.- CHF</b>	<b>Total</b>	<b>13'450.- CHF</b>
		<b>Jährlicher Wartungspreis (18% des einmaligen Listepreises der Lizenz)</b>	<b>2'421.- CHF</b>

(Eigene Darstellung)

Zu berücksichtigen ist, dass der Kauf der Lizenz des Produktes «alive-IT Business Continuity Framework» deutlich billiger ist als die Eigenentwicklung. Jedoch müssen die jährlichen Wartungskosten mit einbezogen werden. Spätestens nach sechs Jahren wird sich die Eigenentwicklung der Schweizerischen Mobiliar rentieren, da der sogenannte Break Even Point der Abbildung 71 erreicht wird. An diesem Punkt würde das eingekaufte Produkt mehr Kosten aufbringen, als die Eigenentwicklung. Der Vorteil des «alive-IT Business Continuity Framework» liegt darin, dass das Produkt direkt verwendet werden kann. Zudem werden die Wartung und der Support des Programmes von der Unternehmung übernommen. Vorteilhaft an der Weiterführung des neuen BIA Systems ist die unbeschränkte Anzahl der Benutzerkonten. Der wesentliche Hauptvorteil liegt darin, dass das Produkt exakt laut den Wünschen der Schweizerischen Mobiliar entwickelt werden kann. So wird gewährleistet, dass alle Anforderungen des neuen Systems umgesetzt werden können.

Abbildung 71: Vergleich Alive-IT und Eigenentwicklung



Im Sitzungsprotokoll vom 29.05.2019, welches im ANHANG XII hinterlegt ist, wird auf Grund der Anfrage der Schweizerischen Mobiliar die Lösung gezeigt, wie das Bereitstellen des neuen BIA Systems durchgeführt wird. Da die Schweizerische Mobiliar zukünftig plant alle Eigenentwicklungen auf der Microsoft Azure Cloud hochzuladen, wird im nächsten Abschnitt beschrieben, wie dieser Vorgang mit dem neuen BIA System durchgeführt werden kann.

## Deployment auf Azure Cloud

Der Prototyp wurde gemäss den Anforderungen für die Eigenentwicklungen der Schweizerischen Mobiliar durchgeführt. Dadurch basiert das neue BIA System auf den gleichen Technologien, wie die Eigenentwicklungen der Mobiliar. So könnte der Prototyp, gleich wie die bestehenden Entwicklungen der Mobiliar, in Zukunft auf der Azure Cloud bereitgestellt werden. Für das Bereitstellen des Prototypen müssen drei verschiedene Produkte eingerichtet werden.

Für das Frontend, welches auf Angular basiert, wird ein statischer Web Server auf Microsoft Azure Cloud eingerichtet. Dafür würde in Microsoft Azure Cloud ein Azure Storage, mehr spezifisch ein Blob Storage eingerichtet. Dieser Speicher ist eine «Platform as a Service» (PaaS) Variante, auf dem der statische Inhalt hochgeladen werden kann. Zusätzlich kann für einen Blob Storage eine eigene Domain definiert werden (Pietschmann, 2016). Im ANHANG XIII befindet sich eine Anleitung, wie ein Blob Storage auf Microsoft Azure Cloud bereitgestellt werden kann.

Das J2EE-Backend wird durch die Einrichtung eines Web App in Microsoft Azure Cloud gewährleistet. Als Runtime-Stack kann der Wildfly Server ausgewählt werden, welcher im entwickelten Prototypen verwendet wird (Benz, 2016). Die Webservices werden daraufhin auf dem Web App bereitgestellt. Im ANHANG XV wird ein Beispiel aufgezeigt, wie eine Java Web App auf Microsoft Azure Cloud freigestellt werden kann.

Damit das Backend auch auf der Microsoft Azure Cloud mit der SQL-Datenbank kommunizieren kann, wird für das DBMS eine Azure SQL Database eingerichtet. Das Backend muss in der Maven Konfigurationsdatei diesbezügliche Anpassungen vornehmen, damit eine Verbindung aufgebaut wird. Zusätzlich muss die JDBC Verbindung angepasst werden. Als Vergleich kann die Abbildung 32 betrachtet werden, welches die JDBC Verbindung im umgesetzten Prototypen aufzeigt.

## 9. Übergabe

Am Ende der Bachelorarbeit wird der Prototyp an den Abteilungsleiter BCM der Schweizerischen Mobiliar übergeben. Zusätzlich wird das GitHub Repository für das Frontend und das Backend öffentlich gestellt, so dass die Mitarbeiter der IT-Abteilung Zugriff auf den Quellcode erhalten. Der Quellcode wird in beiden Systemen ausführlich kommentiert und für das Backend wird zusätzlich noch ein JavaDoc erstellt. Um eine reibungslose Übergabe zu gewährleisten, wird zudem noch eine Benutzeranleitung und ein technischer Bericht erstellt. Die Benutzeranleitung unterstützt die Mitarbeiter bei der Anwendung des entwickelten Systems. Der technische Bericht wird die IT-Abteilung unterstützen, damit die Fortführung des BIA Systems nicht erschwert wird. Der Product Backlog wird dem Kunden übergeben, da dieser alle Funktionalitäten beinhaltet, um den Prototypen weiterzuführen. Zusätzlich wird das neue Produkt der Abteilung BCM im Standort Bern präsentiert und vorgeführt.

In dem letzten Iterations-Meeting wurde dem Arbeitgeber die neuste Version des BIA Systems aufgezeigt. Des Weiterem wurde die Fortführung des Projektes besprochen und auch die Teste und Code Coverage präsentiert. Wie im ANHANG VIII ersichtlich ist, möchte die Schweizerische Mobiliar den Prototypen als Basis für die Weiterentwicklung verwenden. Zudem wird erwähnt, dass das Resultat dieser Bachelorarbeit als Prototyp dient und erst im Jahr 2020 in Produktion überführt werden soll. Während dieser Phase sollten die geforderten Code Coverage Anforderungen erfüllt werden.

## SCHLUSSFOLGERUNG

In dieser Bachelorarbeit wurde ein erfolgreicher Prototyp für das neue BIA System erstellt. Dieser Prototyp dient als Basis für die Weiterführung des Projektes. Dieses Ziel wurde durch das Analysieren der bisherigen Lösung erreicht. Durch diese Analysen konnten alle Stärken und Schwächen erfasst werden. Das Ergebnis zeigte auf, dass die Zusammenarbeit der Mitarbeiter, aufgrund der verwendeten Technologie, erschwert ist. Zudem verzeichneten alle Arbeitsschritte, die von den Anwendern durchgeführt werden, viele Fehlerquellen auf und beanspruchten viel Zeit. Unter anderem hat die bisherige Lösung viele Sicherheitslücken, welche durch den Prototypen behoben wurden. Da die Schweizerische Mobiliar ihre IT Infrastruktur in naher Zukunft umdisponiert, wurde ein Prototyp entwickelt und kein funktionsreifes System. Die Schweizerische Mobiliar konnten bislang noch keine genauere Auskünfte über die zukünftige Software-Umgebung liefern. Aus diesem Grund konnte diese Bachelorarbeit ohne bestimmte Anforderungen an die Technologie umgesetzt werden.

Infolgedessen wurde eine intensive Analyse für potenzielle Technologien durchgeführt. Diese Technologien wurden aufgrund eines Bewertungssystems ausgewertet. Die Bewertungskriterien der Analyse befassten sich mit den Anforderungen an das neue BIA System. Um die Anforderungen der Schweizerischen Mobiliar zu erfüllen wurden drei Personas vorgestellt. Diese Personas repräsentieren die Zielgruppen, welche an dem BIA System arbeiten. Mittels dieser Analyse konnte der Schweizerischen Mobiliar vier unterschiedliche Varianten vorgeführt werden. Die erste Variante basierte auf einem PHP Framework namens Laravel. Bei der zweiten Variante wurde ein NodeJS Framework namens KeystoneJS empfohlen. Durch eine weitere Variante wurde ein C# Framework basierend auf ASP .NET MVC vorgeschlagen. Als letzte Variante wurde die Entwicklung des Prototypen nach den Anforderungen der Eigenentwicklungen der Mobiliar zur Auswahl gestellt. Diese Varianten divergierten sich zwischen den Technologien und der Art der Umsetzung. Zusammen mit der Mobiliar wurde die letzte Variante für die Umsetzung dieser Bachelorarbeit verlesen. Da der Prototyp nach der Bachelorarbeit übernommen und weitergeführt werden kann, wies diese Variante grosses Potenzial auf.

Nach der Entscheidung bezüglich der verwendeten Variante wurde die Entwicklung nach der Schweizerischen Mobiliar durch das Aufweisen der Systemarchitektur verstärkt. So konnte das Zusammenspiel zwischen Frontend, basierend auf Angular, und dem Backend, basierend auf J2EE drastisch vermittelt werden. Daraufhin wurde das Vorgehen der Implementierung erläutert. Schliesslich wurden die entwickelten Funktionalitäten durch das Datenbankschema, den Codeausschnitten und einer Abbildung des Resultates aufgelistet.

Aufgrund des exorbitanten Projektumfangs, konnten nicht alle Hauptfunktionalitäten in den Prototyp integriert werden. Die Funktionalitäten wurden nach der Prioritätenliste durchgeführt. Die definierten Aufgaben für jede Woche wurden vom Arbeitgeber in jedem wöchentlichen Iterations-Meeting festgelegt. Durch die wöchentlichen Sitzungen wurde die aktive Zusammenarbeit mit dem Kunden gewährleistet. Mithilfe des konstruktiven Feedbacks konnte der Prototyp strikt nach den Wünschen und den Anforderungen der Schweizerischen Mobiliar implementiert werden.

Unter Einsatz von Unit- und Integration-Teste wurde das Frontend und das Backend nachgeprüft. Die Schweizerische Mobiliar verlangt ein Code Coverage von 80% bei dem Bereitstellen auf ihren Systemen. Obwohl dieser Prototyp nicht für die Produktion der Schweizerischen Mobiliar vorgesehen wurde und diese Anforderung deswegen nicht pflichtig war, wurde trotzdem das Code Coverage des Prototypen inspiziert. Von vier Kategorien erreichten drei Einheiten die geforderten Prozentanforderungen. Damit jede Kategorie die geforderten Anforderungen abdecken würden, müssten mehrere Unit- und Integration Teste für die HTTP-Methoden POST, PUT und DELETE durchgeführt werden. Um diese HTTP-Methoden erfolgreich testen zu können, müsste ein Testsystem eingerichtet werden. Dieses Testsystem würde ein exaktes Abbild des Datenbankmanagementsystems beinhalten.

Zum Schluss wurde eine Evaluierung des Prototypen aufgestellt. Dabei wurde der Arbeitsschritt «Geschäftsprozess hinzufügen» des bisherigen Systems mit dem Prototyp verglichen. Diese Gegenüberstellung zeigte eine klare Verbesserung der Sicherheit des Prototypen und der Zeitaufwand für diesen Arbeitsschritt wurde exorbitant verringert. Anschliessend wurden die zu Beginn definierten potenziellen Risiken im Prototypen analysiert. Dabei stellte sich heraus, dass alle schwerwiegende Risiken durch den Prototypen beseitigt wurden. Daraufhin wurde die User Experience und die Multiuser-Funktionalität des Prototypen analysiert. Das Ergebnis dieser Auswertung zeigte eine benutzerfreundliche Oberfläche auf und die Möglichkeit, dass die Mitarbeiter zeitgleich an dem Prototypen arbeiten können.

Die Umdisponierung der Infrastruktur der Schweizerischen Mobiliar beinhaltet unter anderem die Bereitstellung der Entwicklungen auf der Microsoft Azure Cloud. Diesbezüglich wurde zuletzt zwei Anleitungen entworfen, welche das Bereitstellen des Prototypen auf der Microsoft Azure Cloud beschreiben. Um der Schweizerischen Mobiliar die Entscheidung für die Weiterentwicklung des Prototypens zu erleichtern, wurde eine zeitliche und finanzielle Analyse aufgestellt. Diese Analyse vergleicht den Prototypen mit der Software «alive-IT Business Continuity Framework», welche von der Schweizerischen Mobiliar bereits getestet wurde.

Da sich die Schweizerische Mobiliar bislang noch nicht für eine bestimmte Software-Umgebung entschieden hat, ist eine diesbezügliche Klärung notwendig und empfehlenswert. Je nach Entscheid müsste der Prototyp dementsprechend angepasst werden. Bis dahin ist jedoch empfehlenswert, den Prototypen bereits den Mitarbeitern zur Verfügung zu stellen, um konstruktives Feedback zu erhalten.

### **Wie sieht die Zukunft des Prototypen aus?**

Der Prototyp ist noch lange nicht ein funktionsreifes System. Viele Funktionalitäten des Product Backlog müssen noch implementiert werden, damit der Prototyp für die Business Impact Analysen verwendet werden kann. Nach Abschluss dieser Bachelorarbeit wird der Prototyp auf der Microsoft Azure Cloud bereitgestellt, damit das neue System von den Mitarbeitern intensiv getestet und ausgewertet werden kann. Infolgedessen werden mit den involvierten Mitarbeitern noch weiterführende Gespräche nötig sein. Die Schweizerische Mobiliar plant für diesen Prototypen eine Basis zur Weiterentwicklung und anschliessenden Überführung in die Produktion im Jahr 2020.

# LITERATURVERZEICHNIS

- Aarsten, A., Brugali, D., & Menga, G. (1996). *Patterns for Three-Tier Client/Server Applications*. Illinois.
- Abolhassan, F., & Kellermann, J. (2016). *Effizienz durch Automatisierung*. (F. Abolhassan, & J. Kellermann, Hrsg.) Wiesbaden: Springer Gabler.
- AGI Creative Team, AGI Training Team. (2011). *Microsoft Office 2010 Digital Classroom*. Indianapolis: Wiley Publishing, Inc.
- Arndt, C., Hermanns, C., Prof. Dr. Kuchen, H., & Poldner, M. (2009). *Best Practices in der Softwareentwicklung*. Münster.
- Asokan, M. (2013). *Animating the Web with jQuery*. Puthanampatti.
- Bari, A., & Syam, A. (2008). *CakePHP Application Development*. Olton: Packt Publishing.
- Bauer, C., & King, G. (2005). *Hibernate in Action*. Greenwich: Manning Publications Co.
- Benz, B. (2016). *Deploy Java Web Apps to Azure using Eclipse*. o.O.: Microsoft Azure. Abgerufen am 10. 05 2019 von <https://azure.microsoft.com/es-es/blog/deploy-java-web-apps-to-azure-using-eclipse/>
- Boiko, B. (2005). *Understanding Content Management*. o.O.: Hungry Minds Inc. doi:10.1002/bult.221
- Bowler, T., & Bancer, W. (2009). *Symfony 1.3 Web Application Development*. Olton: Packt Publishing.
- Bundesamt für Sicherheit in der Informationstechnik. (2013). *Notfallmanagement mit der Cloud für KMUs*. o.O.
- Cake Software Foundation. (2019). *CakePHP Cookbook Documentation Release 3.7*. o.O. Abgerufen am 23. 05 2019
- Chadwick, J. (2011). *Programming Razor*. Sebastopol: O'Reilly Media, Inc.
- Chadwick, J., Snyder, T., & Panda, H. (2012). *Programming ASP .NET MVC 4*. Sebastopol: O'Reilly Media, Inc.
- ChannelPartner. (28. 07 2010). *ChannelPartner*. Von <https://www.channelpartner.de/a/menschliches-versagen-nimmt-zu,294013> abgerufen

- Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (11. 10 2017). Restful API Architecture Based on Laravel Framework. *IOP Science*, S. 2-3. doi:10.1088/1742-6596/910/1/012016
- Dazer, M. (2012). *RESTful APIs - Eine Übersicht*. Berlin: Technical University Berlin.
- Deeleman, P. (2016). *Learning Angular 2*. Birmingham: Packt Publishing.
- Dharmadasa, K. (2017). *How to store passwords securely with PBKDF2*. o.O.: Medium. Abgerufen am 01. 07 2019 von <https://medium.com/@kasunpdh/how-to-store-passwords-securely-with-pbkdf2-204487f14e84>
- Dickmann, P. (2009). *Schlanker Materialfluss mit Lean Production, Kanban und Innovationen*. (P. Dickmann, Hrsg.) Berlin: Springer-Verlag.
- die Schweizerische Mobiliar. (2016). *Leitfaden Business Continuity*. o.O.
- die Schweizerische Mobiliar. (2017). *Konzept Continuity-Massnahmen*. o.O.
- die Schweizerische Mobiliar. (2018). Geschäftsbericht 2018. In *Geschäftsbericht 2018* (S. 92, 102-106). o.O.: die Schweizerische Mobiliar.
- Donig, J., & Mühlbauer, S. (2011). *Den Überblick über agile Backlogs behalten*. o.O: Sigs datacom.
- Dr. Künzli, B. (2012). SWOT-Analyse: Ein klassisches Instrument der Strategieentwicklung mit viel ungenutztem Potenzial. *Zeitschrift Führung + Organisation ZfO*, 126-129.
- Dr. Preussig, J. (2018). *Agiles Projektmanagement*. Freiburg: Haufe-Lexware GmbH & Co.
- Fayad, M. E., & Schmidt, D. C. (1997). *Object-Oriented Application Frameworks*. o.O.: Communications of the ACM.
- Flanagan, D. (2006). *JavaScript The Definitive Guide*. Sebastopol: O'Reilly Media, Inc.
- Friedrich, J. (2017). *Evaluation von Hybrid Mobile App Frameworks im Kontext von medizinischen Applikationen*. Deutschland: Universität Ulm.
- Gardner, B. S. (2011). Responsive Web Design: Enriching the User Experience. In B. S. Gardner, *Sigma inside the digital ecosystem* (S. 13-19). o.O.: Noblis, Inc.
- Gawryś, Ł. (2017). *Angular: Reactive Forms im Überblick*. Abgerufen am 24. 06 2019 von <https://medium.com/@lukaszalptha/angular-reactive-forms-im-%C3%BCberblick-45e670bf5fb0>

- Golding, D. (2008). *Beginning CakePHP*. o.O.
- Goncalves, A. (2013). *Beginning Java EE 7*. o.O.: Apress.
- Green, B., & Seshadri, S. (2013). *AngularJS*. Sebastopol: O'Reilly Media, Inc.
- Gröning, W., & Schwickert, A. C. (1996). *HTML und Java im World Wide Web*. (A. Dr. Schwickert, Hrsg.) o.O.: Universität Mainz.
- Gustainiene, A. (02. 09 2017). Von LinkedIn: <https://www.linkedin.com/pulse/water-scrum-fall-process-ausra-gustainiene> abgerufen
- Hahn, E. M. (2016). *Express in Action Writing, building, and testing Node.js applications*. Shelter Island: Manning Publications Co.
- Hiles, A. (2002). *Enterprise Risk Assessment and Business Impact Analysis*. Brookfield: Rothstein Associates Inc.
- Holt, E.-M., Winter, D., & Thomaschewski, J. (2011). Personas als Werkzeug in modernen Softwareprojekten. In H. Brau, A. Lehmann, K. Petrovic, & M. Schroeder (Hrsg.), *Tagungsband UP11* (S. 40-44). Stuttgart: German UPA e.V.
- Jain, N., Mangal, P., & Mehta, D. (12. 12 2014). AngularJS: A Modern MVC Framework in JavaScript. *Journal of Global Research in Computer Science*, S. 17-18. Abgerufen am 20. 05 2019
- Keller, W. (1997). *Mapping Objects to Tables A Pattern Language*. Lochham.
- Kötterheinrich, M., & Oevel, G. (2015). *Business Continuity Management (BCM) an Hochschulen*. o.O.
- Künzi, R. (2018). *Business Continuity Management Evaluation von Softwareprodukten*. Bern.
- Kurzlechner, W. (2010). *Unsicherheitsfaktor Mensch*. o.O.: Tech Channel. Abgerufen am 22. 05 2019 von <https://www.tecchannel.de/a/unsicherheitsfaktor-mensch,2030839>
- Laaziri, M., Benmoussa, K., Khouilji, S., Larbi, K. M., & Yamami, A. E. (30. 08 2018). A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering (IJECE)*, S. 711. doi:10.11591/ijece.v9i1.pp704-712
- Liebig, H. C. (2014). Praktische Umsetzung eines internationalen Web-Relaunchs mit Hilfe einer hybriden Projektmethodik. In M. Engstler, E. Hanser, M. Mikusz, & G.

- Herzwurm, *Projektmanagement und Vorgehensmodelle 2014* (S. 131-132). Stuttgart: Gesellschaft für Informatik e.V.
- Manricks, G. (2013). *Instant Handlebars.js*. Birmingham: Packt Publishing.
- McCool, S. (2012). *Laravel Starter*. Birmingham: Packt Publishing.
- McGovern, J., Tyagi, S., Stevens, M. E., & Mathew, S. (2003). *Java Web Services Architecture*. o.O.
- Mesbah, A. (2009). *Analysis and Testing of Ajax-based Single-page Web Applications*. o.O.
- Meyer, E. A. (2004). *Cascading Style Sheets The Definitive Guide*. Sebastopol: O'Reilly Media, Inc.
- Moos, J. (2016). *Mock-Ups für Visualisierungen kooperationsrelevanter medizinischer Daten*. Siegen: Universität Siegen.
- Müller, F. (2009). *Professionelle Rich-Client-Lösungen mit Flex und Java*. München: Addison-Wesley Verlag.
- Myer, T. (2008). *Professional CodeIgniter*. Indianapolis: Wiley Publishing, Inc.
- Nice, B. (18. 02 2017). *Medium*. Von <https://medium.com/level-up-web/best-practices-of-responsive-web-design-6da8578f65c4> abgerufen
- Padilla, A. (2009). *Beginning Zend Framework*. o.O.
- Panati, M. (2016). *Beginning KeystoneJS*. o.O.
- Paulsen, C. (2003). *Risikoanalyse von biometrischen Systemen*. Hamburg: Universität Hamburg. Abgerufen am 11. 05 2019 von [http://agn-www.informatik.uni-hamburg.de/papers/doc/diparb\\_christian\\_paulsen.pdf](http://agn-www.informatik.uni-hamburg.de/papers/doc/diparb_christian_paulsen.pdf)
- Petrusha, R., & Kotary, N. (2002). *Writing Excel Macros with VBA Second Edition*. USA: O'Reilly Media, Inc.
- PhD Filipović, D., PhD Krišto, M., & Podrug, N. (2017). *Impact of crisis situations on development of business continuity management in croatia*. o.O.: Original scientific papaer UDC. doi:10.30924/mjcmi/2018.23.1.99

- Pietschmann, C. (2016). *Static Website Hosting in Azure Storage*. o.O.: Build Azure. Abgerufen am 10. 05 2019 von <https://buildazure.com/static-website-hosting-in-azure-storage/>
- Pongratz, M. (1996). Verfahren zur Risikoanalyse in der Informatik-Revision. In K. Prof.Dr. Bauknecht, D. Prof.Dr. Karagiannis, & S. Dr. Teufel, *Sicherheit in Informationssystemen SIS '96* (S. 229-235). Wien: Hochschulverlag AG.
- Pope, K. (2009). *Zend Framework 1.8 Web Application Development*. Olton: Packt Publishing.
- Porebski, B., Przystalski, K., & Nowak, L. (2011). *Building PHP Applications with Symfony, CakePHP, and Zend Framework*. o.O.: John Wiley and Sons.
- Prof. Dr.-Ing. Johannsen, G. (1993). *Mensch-Maschine-Systeme*. Berlin: Springer-Verlag.
- Ramos, M., Valente, M. T., & Terra, R. (2017). *AngularJS Performance: A Survey Study*. o.O. doi:10.1109/MS.2017.265100610
- Rodriguez, A. (2008). *RESTful Web services: The basics*. o.O.: IBM Corporation.
- Russell, M. A. (2014). *Mining the Social Web*. Sebastopol: O'Reilly Media, Inc.
- Rust, S., Schelling, J., & Schipper, D. (o.D.). *Building Real-Time Web Applications with Meteor*. o.O. Abgerufen am 19. 05 2019 von [https://staas.home.xs4all.nl/t/swtr/documents/wt2015\\_meteor.pdf](https://staas.home.xs4all.nl/t/swtr/documents/wt2015_meteor.pdf)
- Rylan, C. (2016). *Introduction to Angular Pipes*. o.O.: o.V.
- Schürmanns, S. (2019). *Ghost - Node-CMS für Blogs und Websites*. o.O.: CMSstash. Abgerufen am 23. 05 2019 von <https://cmsstash.de/javascript/ghost>
- Schwaber, K., & Sutherland, J. (2013). *Der Scrum Guide*. o.O.: Scrum.Org and ScrumInc.
- Schweizerischer Versicherungsverband SVV. (2015). *Business Continuity Management (BCM) für Versicherungsunternehmen in der Schweiz - Mindeststandards und Empfehlungen*. Zürich: Schweizerischer Versicherungsverband SVV.
- Shahzad, F. (2017). Modern and Responsive Mobile-enabled Web Applications. In E. Shakshuki, & A. Yasar, *Procedia Computer Science* (S. 410-415). Houston: Elsevier B.V.

- Shingala, K. (2018). *JSON Web Token (JWT) based client authentication in Message Queuing Telemetry Transport (MQTT)*. o.O.: Cornell University.
- Sielhorst, G., & Wilhelm, M. (1992). *Makroprogrammierung mit Excel 4.0, 2. aktualisierte Auflage*. Wiesbaden: Friedr. Vieweg & Sohn Verlagsgesellschaft mbH.
- Spolti, F. C. (2014). *WildFly: New Features*. Birmingham: Packt Publishing Ltd.
- Spurlock, J. (2013). *Bootstrap Responsive Web Development*. Sebastopol: O'Reilly Media, Inc.
- Steyer, M., & Schwichtenberg, H. (2014). *Moderne Webanwendungen mit ASP .NET MVC & JavaScript 2. Auflage*. Heidelberg: O'Reilly Verlag GmbH & Co. KG.
- Strack, I. (2015). *Getting Started with MEteor.js JavaScript Framework Second Edition*. o.O.: Packt Publishing.
- Terpstra, E. (2013). *Ghost - Just a Blogging Platform*. o.O.: modernweb. Abgerufen am 23. 05 2019 von <https://modernweb.com/ghost-just-a-blogging-platform/>
- Tullis, T., & Albert, B. (2013). *Measuring the User Experience Collecting, Analyzing, and Presenting Usability Metrics*. USA: Elservice Inc.
- Upton, D. (2007). *CodeIgniter for Rapid PHP Application Development*. Olton: Packt Publishing.
- West, D. (2011). *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*. o.O: Forrester Research, Inc.
- White, S. A. (2004). *Introduction to BPMN*. o.O.: BPTrends.
- Wojciechowski, J., Sakowicz, B., Dura, K., & Napieralski, A. (3 2004). *MVC Model, Struts Framework and File upload Issues in Web Applications Based on J2EE Platform*. Poland. doi:10.1109/TCSET.2004.1365980
- Zaninotto, F., & Potencier, F. (2007). *The Definitive Guide to symfony*. o.O.: Sensio SA.
- Zunke, S., & D'Souza, V. (2014). *JSON vs XML: A Comparative Performance Analysis of Data Exchange Formats*. o.O.: International Journal of Computer Science and Network.

# ANHANGSVERZEICHNIS

ANHANG I: SOLL-PLANUNG .....	113
ANHANG II: IST-PLANUNG.....	114
ANHANG III: MOCK-UPS .....	115
ANHANG IV: BESTÄTIGUNG VARIANTE 4 .....	119
ANHANG V: PRODUCT BACKLOG .....	120
ANHANG VI: TESTE DER SCHWEIZERISCHEN MOBILIAR .....	123
ANHANG VII: CODE COVERAGE .....	124
ANHANG VIII: FORTFÜHRUNG BIA SYSTEM.....	125
ANHANG IX: RELEASE ROADMAP .....	126
ANHANG X: IMPLEMENTATIONSITERATION .....	127
ANHANG XI: SITZUNGSPROTOKOLL 06.05.2019 .....	131
ANHANG XII: SITZUNGSPROTOKOLL 29.05.2019 .....	133
ANHANG XIII: PREISLISTE «ALIVE-IT BUSINESS CONTINUITY FRAMEWORK» .....	134
ANHANG XIV: DEPLOYMENT BLOB STORAGE.....	135
ANHANG XV: DEPLOYMENT JAVA WEB APP .....	136
ANHANG XVI: ABNAHME DES PROTOTYPEN.....	137

# ANHANG I: SOLL-PLANUNG

Abbildung 72: SOLL-Planung

Nr.	Aufgaben	geplante Zeit	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5	Woche 6	Woche 7	Woche 8	Woche 9	Woche 10	Woche 11	Woche 12	Woche 13
<b>1</b>	<b>Informieren</b>	<b>24h</b>													
2	Erstes Meeting mit Betreuer und Unternehmen	2h													
3	Recherche Projektmanagement Methoden	4h													
4	Recherche BCM und BIA	6h													
5	IST-Analyse bisheriges System	6h													
6	IST-Analyse potenzielle Technologien	6h													
<b>7</b>	<b>Planen</b>	<b>27h</b>													
8	Planung - Soll	2h													
9	Product Backlog erstellen	2h													
10	Logo erstellen	2h													
11	Mockups erstellen	8h													
12	Deployment mit der schweizerischen Mobiliar planen	5h													
13	Datenbank Modell planen	8h													
<b>14</b>	<b>Entscheidung</b>	<b>3h</b>													
15	Entscheidung Mockups	1h													
16	Entscheidung verwendete Technologie	1h													
17	Entscheidung Projektmanagement Methode	1h													
<b>18</b>	<b>Realisieren</b>	<b>200h</b>													
19	Realisierung Agile Kanban Methode. Mit einer Iterationszeit von einer Woche. Abarbeiten des Product Backlogs nach Priorität. Nach jeder Woche Test der Funktionen durchführen. Nach jeder Woche ein Skype-Meeting mit Auftragsgeber (Product Owner) um ein Feedback und gewünschte Anpassungen zu erhalten	200h													
<b>20</b>	<b>Kontrollieren</b>	<b>42h</b>													
21	Black-Box-Test & White-Box-Test durchführen	8h													
22	Testphase Funktionalitäten	12h													
23	Fehlerliste erstellen	2h													
24	Kapitel Zusammenfassung verfassen	4h													
25	Abbildungsverzeichnis kontrollieren	4h													
26	Tabellenverzeichnis kontrollieren	4h													
27	Literaturverzeichnis kontrollieren	8h													
<b>28</b>	<b>Auswerten</b>	<b>18h</b>													
29	Schlussfolgerung verfassen	2h													
30	Lektorat	6h													
31	Unterlagen zusammenstellen	2h													
32	Projekt abschliessen, Abgabe Unterlagen	8h													
<b>33</b>	<b>Dokumentation &amp; Präsentation</b>	<b>57h</b>													
34	Dokumentation vorbereiten	1h													
35	Dokumentation schreiben	50h													
36	Dokumentation abschliessen	2h													
37	Präsentation erstellen	4h													
<b>38</b>	<b>Reserve</b>	<b>16h</b>													
39	Reserve	16h													
	<b>Total</b>	<b>387h</b>	<b>24h</b>	<b>24h</b>	<b>24h</b>	<b>25h</b>	<b>25h</b>	<b>25h</b>	<b>25h</b>	<b>25h</b>	<b>30h</b>	<b>40h</b>	<b>40h</b>	<b>40h</b>	<b>40h</b>

(Eigene Darstellung)

## ANHANG II: IST-PLANUNG

In der Realisierungsphase der Abbildung 73 werden die Einarbeitungswochen in die Technologien durch einen roten Rahmen markiert.

Abbildung 73: IST-Planung

Nr.	Aufgaben	geplante Zeit	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5	Woche 6	Woche 7	Woche 8	Woche 9	Woche 10	Woche 11	Woche 12	Woche 13
<b>1</b>	<b>Informieren</b>	<b>25h</b>													
2	Erstes Meeting mit Betreuer und Unternehmen	2h													
3	Recherche Projektmanagement Methoden	3h													
4	Recherche BCM und BIA	4h													
5	IST-Analyse bisheriges System	8h													
6	IST-Analyse potenzielle Technologien	8h													
<b>7</b>	<b>Planen</b>	<b>17h</b>													
8	Planung - Soll	1h													
9	Product Backlog erstellen	2h													
10	Logo erstellen	1h													
11	Mockups erstellen	5h													
12	Deployment mit der schweizerischen Mobiliar planen	4h													
13	Datenbank Modell planen	4h													
<b>14</b>	<b>Entscheidung</b>	<b>2h</b>													
15	Entscheidung Mockups	0.5h													
16	Entscheidung verwendete Technologie	0.5h													
17	Entscheidung Projektmanagement Methode	1h													
<b>18</b>	<b>Realisieren</b>	<b>186h</b>													
19	Realisierung Agile Kanban Methode. Mit einer Iterationszeit von einer Woche. Abarbeiten des Product Backlogs nach Priorität. Nach jeder Woche Test der Funktionen durchführen. Nach jeder Woche ein Skype-Meeting mit Auftragsgeber (Product Owner) um ein Feedback und gewünschte Anpassungen zu erhalten	186h													
<b>20</b>	<b>Kontrollieren</b>	<b>28h</b>													
21	Manuelle Test durchführen	4h													
22	Testphase Funktionalitäten	13h													
23	Fehlerliste erstellen	1h													
24	Kapitel Zusammenfassung verfassen	4h													
25	Abbildungsverzeichnis kontrollieren	2h													
26	Tabellenverzeichnis kontrollieren	1h													
27	Literaturverzeichnis kontrollieren	3h													
<b>28</b>	<b>Auswerten</b>	<b>13h</b>													
29	Schlussfolgerung verfassen	3h													
30	Lektorat	6h													
31	Unterlagen zusammenstellen	2h													
32	Projekt abschliessen, Abgabe Unterlagen	2h													
<b>33</b>	<b>Dokumentation &amp; Präsentation</b>	<b>159h</b>													
34	Dokumentation vorbereiten	1h													
35	Dokumentation schreiben	146h													
36	Dokumentation abschliessen	4h													
37	Präsentation erstellen	8h													
<b>38</b>	<b>Reserve</b>	<b>16h</b>													
39	Reserve	16h													
<b>Total</b>		<b>446h</b>	<b>27h</b>	<b>35h</b>	<b>27h</b>	<b>26h</b>	<b>30h</b>	<b>26h</b>	<b>28h</b>	<b>17h</b>	<b>63h</b>	<b>53h</b>	<b>44h</b>	<b>40h</b>	<b>30h</b>

(Eigene Darstellung)

## ANHANG III: MOCK-UPS

Abbildung 74: Mock-Up Login

Login

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

Abbildung 75: Mock-Up Dashboard

Dashboard

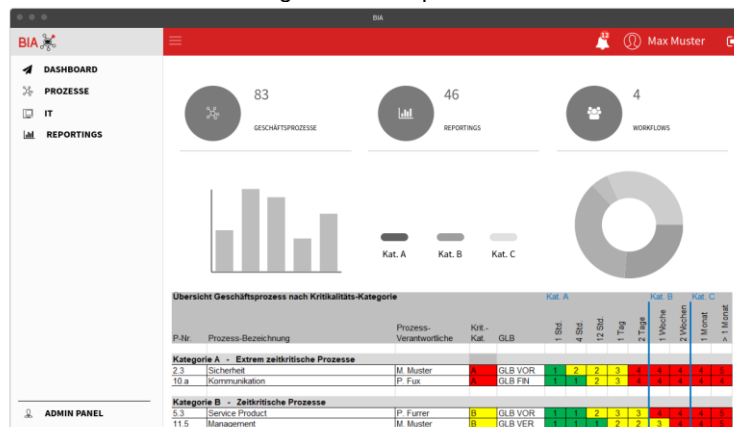
Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

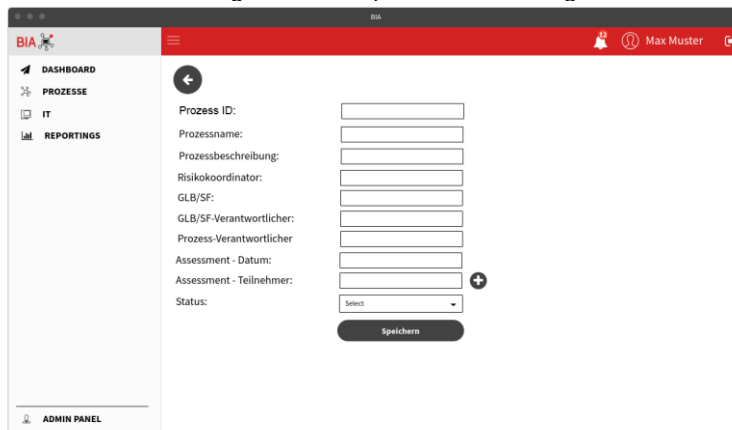
Abbildung 76: Mock-Up Prozesse

Prozess  
Übersicht

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

Prozess  
hinzufügen

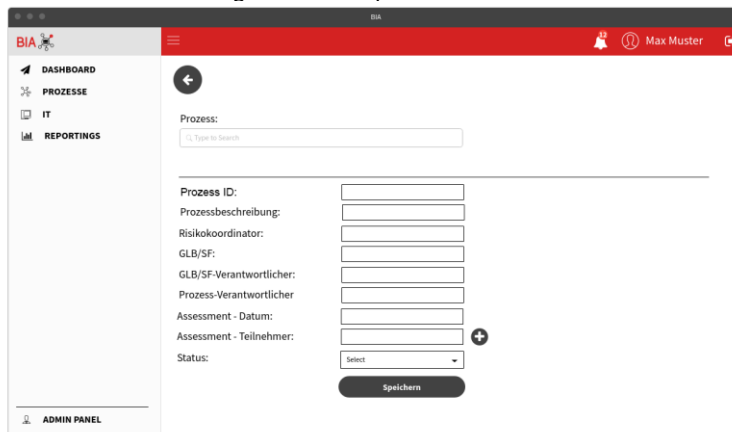
Abbildung 77: Mock-Up Prozess hinzufügen



The mock-up shows a web interface for adding a new process. On the left is a sidebar with a logo and navigation links: DASHBOARD, PROZESSE, IT, and REPORTINGS. The main area has a red header with a back arrow, a user profile 'Max Muster', and a search bar. Below the header, there's a form with the following fields: Prozess ID, Prozessname, Prozessbeschreibung, Risikokoordinator, GLB/SF, GLB/SF-Verantwortlicher, Prozess-Verantwortlicher, Assessment - Datum, Assessment - Teilnehmer, and Status (a dropdown menu). A '+ ' icon is next to the 'Assessment - Teilnehmer' field. At the bottom right of the form is a 'Speichern' (Save) button. The footer contains an 'ADMIN PANEL' link.

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>Prozess  
bearbeiten

Abbildung 78: Mock-Up Prozess bearbeiten

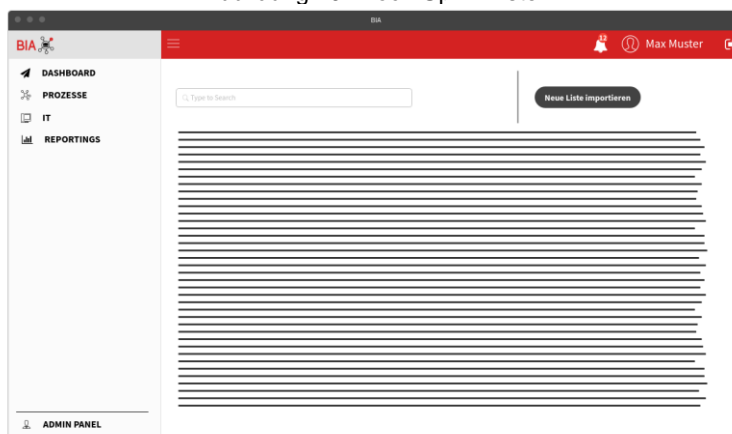


The mock-up shows a web interface for editing an existing process. The layout is identical to the previous one, but the main area features a search bar labeled 'Prozess:' with a placeholder 'Type to Search'. Below the search bar, the same form fields are present: Prozess ID, Prozessbeschreibung, Risikokoordinator, GLB/SF, GLB/SF-Verantwortlicher, Prozess-Verantwortlicher, Assessment - Datum, Assessment - Teilnehmer, and Status (a dropdown menu). A '+ ' icon is next to the 'Assessment - Teilnehmer' field. At the bottom right of the form is a 'Speichern' (Save) button. The footer contains an 'ADMIN PANEL' link.

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

IT-Liste

Abbildung 79: Mock-Up IT-Liste

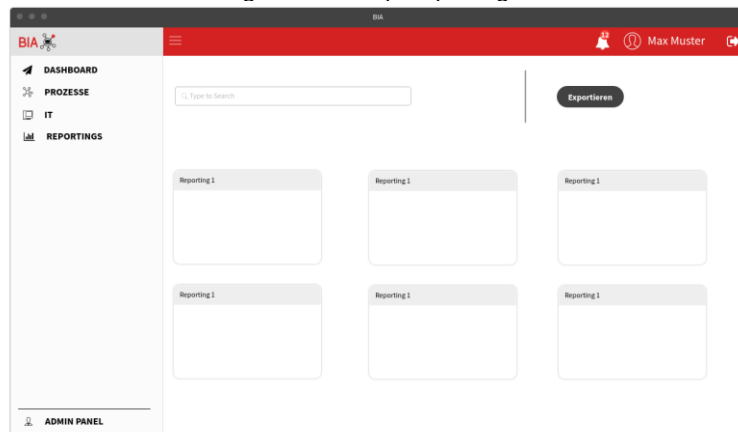


The mock-up shows a web interface for an IT list. The layout is consistent with the previous ones. The main area has a search bar with a placeholder 'Type to Search'. To the right of the search bar is a button labeled 'Neue Liste Importieren'. Below these elements is a large table with many rows, each containing several columns of data, representing a list of IT items. The footer contains an 'ADMIN PANEL' link.

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

## Reporting Dashboard

Abbildung 80: Mock-Up Reporting Dashboard



Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

## Reporting Kritikalitäts-Kategorie

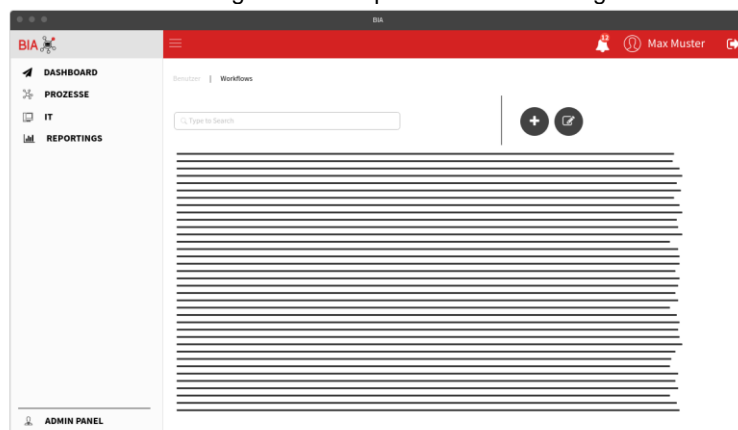
Abbildung 81: Mock-Up Reporting Kritikalitäts-Kategorie

P-ID	Prozess-Bezeichnung	Prozess-Verantwortliche	Krit.-Kat.	GLB	1 Std.	4 Std.	12 Std.	1 Tag	2 Tage	1 Woche	2 Wochen	1 Monat	3 Monate
<b>Kategorie A - Extrem zeitskritische Prozesse</b>													
2.3	Sicherheit	M. Muster	A	GLB VOR	1	2	3	4	5	6	7	8	9
10.9	Kommunikation	P. Fux	A	GLB FRI	1	2	3	4	5	6	7	8	9
<b>Kategorie B - Zeitskritische Prozesse</b>													
5.3	Service Product	P. Furrer	B	GLB VOR	1	2	3	4	5	6	7	8	9
11.5	Management	M. Muster	B	GLB VER	1	2	3	4	5	6	7	8	9
2.3	Sicherheit	M. Muster	A	GLB VOR	1	2	3	4	5	6	7	8	9
10.9	Kommunikation	P. Fux	A	GLB FRI	1	2	3	4	5	6	7	8	9
<b>Kategorie B - Zeitskritische Prozesse</b>													
5.3	Service Product	P. Furrer	B	GLB VOR	1	2	3	4	5	6	7	8	9
11.5	Management	M. Muster	B	GLB VER	1	2	3	4	5	6	7	8	9
2.3	Sicherheit	M. Muster	A	GLB VOR	1	2	3	4	5	6	7	8	9
10.9	Kommunikation	P. Fux	A	GLB FRI	1	2	3	4	5	6	7	8	9
<b>Kategorie B - Zeitskritische Prozesse</b>													
5.3	Service Product	P. Furrer	B	GLB VOR	1	2	3	4	5	6	7	8	9
11.5	Management	M. Muster	B	GLB VER	1	2	3	4	5	6	7	8	9
5.3	Service Product	P. Furrer	B	GLB VOR	1	2	3	4	5	6	7	8	9
11.5	Management	M. Muster	B	GLB VER	1	2	3	4	5	6	7	8	9

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

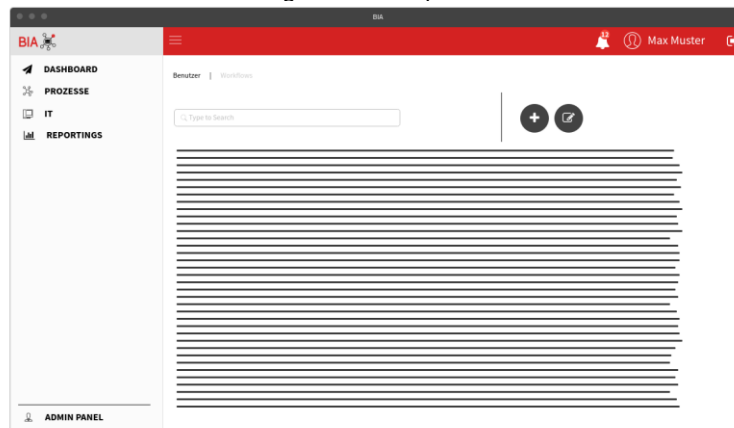
## Benutzer-verwaltung

Abbildung 82: Mock-Up Benutzerverwaltung



Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

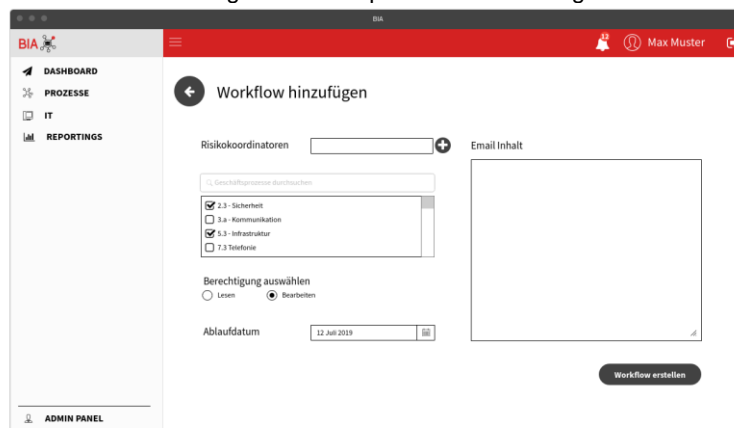
Abbildung 83: Mock-Up Workflows



Workflows

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

Abbildung 84: Mock-Up Workflow hinzufügen



Workflow  
hinzufügen

Eigene Darstellung und erstellt mit <https://www.mockflow.com/>

## ANHANG IV: BESTÄTIGUNG VARIANTE 4

Abbildung 85: Bestätigung Variante 4

**In-Albon Damian** <damian.in-albon@mobi.ch>  
An: Mathis Fux <mathis.fux@gmail.com>

29. Mai 2019 um 12:28

Sehr geehrter Herr Fux

Besten Dank für das konstruktive Meeting und das Sitzungsprotokoll. Gerne bestätige ich Ihnen, dass ich von der Umsetzung der Variante 4 Kenntnis genommen habe und mit der Einschränkung der Funktionalitäten einverstanden bin.

Weiterhin viel Erfolg!

Beste Grüsse

Damian In-Albon

---

**die Mobiliar**

Schweizerische Mobiliar Versicherungsgesellschaft AG

Bundesgasse 35, 3001 Bern

(Eigene Darstellung)

# ANHANG V: PRODUCT BACKLOG

Abbildung 86: Product Backlog Sicht 1

US Nr.	Thema	Als ein	will ich	User Stories	so dass ich	Akzeptanz Kriterien	Priorität	Status	Story Points	Woche	US akzeptiert	MoScow
1	Application	Developer	alle Environments aufsetzen		anschliessend mit der Arbeit beginnen kann	Environment funktionieren; Wissen wie dies zu verwenden ist	1	●	55	1	04.06.2019	Must
3	Application	Benutzer	im Navigations Menü mein Funktion wählen		schneller am System arbeiten kann	Über den richtigen Menü Punkt auf die richtige Seite gelangen	10	●	8	4	25.06.2019	Must
4	Application	Benutzer	eine Übersicht aller Geschäftsprozesse haben		diese im Überblick behalten kann	Liste aller bestehender Geschäftsprozesse	15	●	13	4	25.06.2019	Must
5	Application	Benutzer	einen neuen Geschäftsprozess hinzufügen können mit den Informationsdaten		das System auf dem neusten Stand halten kann	Neuer Geschäftsprozess wurde in der DB gespeichert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	20	●	21	4	25.06.2019	Must
6	Application	Benutzer	einen bestehenden Geschäftsprozess entfernen können		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde aus der DB entfernt; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	25	●	8	4	25.06.2019	Must
7	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Kritikalitäten anzupassen		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	30	●	34	5	02.07.2019	Must
2	Application	Benutzer	anmelden können auf dem System		Zugriff auf alle Funktionalitäten habe	Erfolgreich eingeloggt als valider Benutzer	32	●	21	5	02.07.2019	Must
36	Application	Admin	bei der Anmeldung ein zusätzlichen Punkt in der Navigation haben		auf das Admin Panel zugreifen kann	Navigationspunkt wird nur bei den Administratoren angezeigt;	33	●	8	5	02.07.2019	Must
37	Application	Benutzer	mich abmelden können		erfolgreich meinen Laptop ausschalten kann	Erfolgreich ausgeloggt;	34	●	1	5	02.07.2019	Must
10	Application	Benutzer	eine Übersicht aller Reportings haben		diese im Überblick behalten kann	Anmeldungen werden gelöscht;	45	●	2	6	09.07.2019	Must
11	Application	Benutzer	die Gesamtübersicht kompakt Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung aller Reportings	50	●	13	6	09.07.2019	Must
12	Application	Benutzer	die Übersicht Kritikalität Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	55	●	13	6	09.07.2019	Must
19	Application	Benutzer	ein Dashboard mit der Kritikalität Übersicht haben		diese immer im Blick halten kann	Liste der Geschäftsprozesse inkl. Kritikalität	56	●	2	6	09.07.2019	Could
20	Application	Benutzer	ein Dashboard mit Diagramme haben		eine Analyse der Prozesse habe	Analyse Diagramm anhand der Übersicht der Kritikalität	57	●	13	6	09.07.2019	Could
13	Application	Benutzer	die Arbeitsplatz Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	60	■	13			Should
15	Application	Benutzer	die Lieferantenliste Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	70	■	13			Should
17	Application	Benutzer	die Krit. Funktionen Kat Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	80	■	13			Should
38	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Prozessabhängigkeiten anzupassen		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	85	■	8			Must
39	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Spitzenzeiten anzupassen		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	90	■	8			Must
40	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Standorte anzupassen		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	95	■	21			Must
41	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Schlüsselfunktionen anzupassen		das System auf dem neusten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	96	■	13			Must









(Eigene Darstellung)

Abbildung 87: Product Backlog Sicht 2

US Nr.	Thema	Als ein	User Stories	so dass ich	Akzeptanz Kriterien	Priorität	Status	Story Points	Woche	US akzeptiert	MosCow
42	Application	Benutzer	einen bestehenden Geschäftsprozess bearbeiten können um die Lieferanten anzupassen	das System auf dem neuesten Stand halten kann	Bestehender Geschäftsprozess wurde in der DB aktualisiert; Übersichtsliste der Geschäftsprozesse wurde aktualisiert	97	■	13			Must
38	Application	Benutzer	Die Kritikalität Reportings nach der Zeit filtern können	nach dem Level des Prozesses in einer bestimmten Zeit die Kritikalität Übersicht habe	Filtern nach den Zeiten	98	■	8			Must
21	Application	Benutzer	bestehende Massnahmen eines Geschäftsprozesses angezeigt haben	diese beim einem Notfall abrufen kann	Anzeige der dazugehörigen Massnahmen; Zurück Knopf um wieder alle Geschäftsprozesse anzuzeigen	100	■	21			Must
22	Application	Benutzer	bestehende Massnahmen eines Geschäftsprozesses bearbeiten können	das System auf dem neuesten Stand halten kann	Bestehende Massnahme in der DB aktualisieren; Ansicht der Massnahme aktualisieren	105	■	13			Must
23	Application	Benutzer	bestehende Massnahmen von einem Geschäftsprozesse entfernen kann	das System auf dem neuesten Stand halten kann	Bestehende Massnahme in der DB entfernen; Ansicht der Massnahme aktualisieren	110	■	5			Must
24	Application	Benutzer	eine neue Massnahme einem Geschäftsprozess hinzufügen	das System auf dem neuesten Stand halten kann	Neue Massnahme wird in der DB hinzugefügt; Ansicht der Massnahme aktualisieren	115	■	13			Must
25	Application	Admin	ein Admin Panel haben	das System verwalten kann	Zugriff hat nur ein Admin; Zusätzlich wird in der Navigation ein Menu Punkt hinzugefügt	120	■	2			Should
26	Application	Admin	andere Benutzer als Admin definieren können	einzelne administrative Aufgaben anderen Mitarbeiter übergeben kann	Auflistung aller Benutzer; Benutzer können als Admin definiert werden; Neuer Admin hat Zugriff auf Admin Panel	125	■	8			Should
27	Application	Admin	eine Übersicht aller Benutzer im Admin Panel angezeigt haben	die Benutzer verwalten kann	Auflistung aller Benutzer	130	■	5			Could
28	Application	Admin	einen neuer Benutzer im Admin Panel hinzufügen können	die Benutzer verwalten kann	Neuer Benutzer wird in der DB hinzugefügt; Auflistung aller Benutzer wird aktualisiert	135	■	8			Could
29	Application	Admin	einen bestehenden Benutzer im Admin Panel entfernen können	die Benutzer verwalten kann	Bestehender Benutzer wird in der DB entfernt; Auflistung aller Benutzer wird aktualisiert	140	■	3			Could
30	Application	Admin	im Admin Panel einen Workflow erstellen können	externen Mitarbeiter für eine gewisse Zeit auf spezifische Prozesse Zugriff gewährleisten kann	Neuer Workflow wird in der DB hinzugefügt; Externer Mitarbeiter hat nur solange Zugriff wie auch angegeben; Auflistung aller Workflows wird aktualisiert	145	■	34			Must
31	Application	Admin	im Admin Panel eine History der Workflows angezeigt haben	die externen Mitarbeiter im Überblick habe	Auflistung aller Workflows	150	■	8			Should
32	Application	Admin	im Admin Panel einen Workflow bearbeiten können	den Zugriff bearbeiten kann	Bestehender Workflow wird in der DB aktualisiert; Auflistung aller Workflows wird aktualisiert	155	■	21			Should

(Eigene Darstellung)

Abbildung 88: Product Backlog Sicht 3

US Nr.	Thema	Als ein	will ich	User Stories	so dass ich	Akzeptanz Kriterien	Priorität	Status	Story Points	Woche	US akzeptiert	MosCow
33	Application	Admin	im Admin Panel einen Workflow entfernen können		den Zugriff verweigern kann	Bestehender Workflow wird in der DB entfernt; Externer Mitarbeiter hat keinen Zugriff mehr; Auflistung aller Workflows wird aktualisiert	160		8			Should
34	Application	Admin	eine Benachrichtigung erhalten, falls ein externer Mitarbeiter den Workflow beendet hat		eine Erinnerung erhalten und den Zugriff überprüfen kann	Externer Mitarbeiter beendet seinen Workflow; Externer Mitarbeiter hat keinen Zugriff mehr auf das System; Benachrichtigung erscheint bei allen Admins	165		13			Could
35	Application	Benutzer	Export der Reportings auf externe Excel-Datei		individuelle Reports zusammenstellen zu können	Export der ausgewählten Reportings in Excel-Datei	170		13			Could
14	Application	Benutzer	die Lieferanten pro Prozess Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	200		13			Won't
16	Application	Benutzer	die Lieferanten Kritikalität Kategorie Reporting Ansicht haben		diese analysieren kann	Visuelle Darstellung dieser Analyse	205		13			Won't
8	Application	Benutzer	eine Übersicht aller IT Anwendungen haben		die Ressourcen kontrollieren kann	Liste aller bestehender IT Anwendungen	206		5			Should
9	Application	Benutzer	eine separate Excel-Datei als neue IT-Liste importieren können		nicht jeden neuen IT Prozess manuell eintragen muss	IT-Liste wird durch die neue Liste in der DB aktualisiert; Übersichtsliste der IT Anwendungen wurde aktualisiert	210		89			Should
18	Application	Benutzer	die IT Reporting Ansicht haben (Fakultativ)		diese analysieren kann	Visuelle Darstellung dieser Analyse	215		8			Could

(Eigene Darstellung)

# ANHANG VI: TESTE DER SCHWEIZERISCHEN MOBILIAR

Abbildung 89: E-Mail Test der Schweizerischen Mobiliar

**Indermühle Emanuel** <emanuel.indermuehle@mobi.ch>  
An: Mathis Fux <mathis.fux@gmail.com>  
Cc: In-Albon Damian <damian.in-albon@mobi.ch>

5. Juli 2019 um 15:44

Hallo Mathis

Wir verwenden sehr viele Arten von Tests um sicherzustellen, dass jeder Commit direkt in Produktion gehen kann:

Hier ist was dazu dokumentiert:

<https://cwiki.mobicorp.ch/confluence/display/ST/Testing+im+Agile+Release+Train+%28ART%29+und+in+den+Agilen+Teams>

Im Frontend:

- Unit Tests mit Jest oder Karam
- E2E Tests mit Protractor

Im Backend:

- Unit Tests mit JUnit 5
- Service Tests ...
- Performance- und Lasttests mit JMeter

Zwischen Services

- CDC Pact Tests
  - <https://martinfowler.com/articles/consumerDrivenContracts.html>
  - <https://docs.pact.io/>
- Solution Integration Tests mit LeanFT

Cheers Emanuel

(Eigene Darstellung)

## ANHANG VII: CODE COVERAGE

Abbildung 90: E-Mail Code Coverage

**Indermühle Emanuel** <emanuel.indermuehle@mobi.ch>  
An: Mathis Fux <mathis.fux@gmail.com>

6. Juli 2019 um 13:39

Hallo Mathis

Jede Art von Test hat ihren Zweck, und wenn du den nicht brauchst ist es verlorene Mühe die entsprechenden Tests zu implementieren:

- Unit Tests: Diese sind wohl immer sinnvoll. Sie helfen dir sicherzustellen zukünftige Änderungen an deinem Code vorzunehmen, ohne angst zu haben dabei was kaputt zu machen. Sie erlaube auch Refactorings schneller und mit mehr Vertrauen durchzuführen. Wir haben auf der JAP die Anforderung von 80% Test-Coverage für diese Tests.
- Alle anderen Tests kannst du genauso gut Manuell durchführen, je nach dem in welchem Modus du deployest. Wenn dein Service mit einer CI/CD Pipeline released wird. Das heisst ja dann, dass jeder commit automatisch deployed wird. Dann brauchst du sicher auch Service Tests und E2E Frontend Tests.
- Performance und Last-Tests machst du nur wenn du weisst, dass dein Service grosser Last ausgesetzt sein wird.
- CDC Tests machst du wenn dein Service mit einem anderen Service zusammenarbeitest wobei min. einer davon mit einer CI/CD Pipeline released wird und dadurch angewiesen ist, dass die Schnittstellen automatisch getestet werden.
- SOLIT (Solution Integration Tests) brauchst wenn du Fachprozess abbildest, welche über mehrere Applikationen hinweg funktionieren müssen, und in welcher Services mit einer CI/CD Pipeline released werden.

LG Emanuel

(Eigene Darstellung)

## ANHANG VIII: FORTFÜHRUNG BIA SYSTEM

Abbildung 91: E-Mail Fortführung BIA System

**In-Albon Damian** <damian.in-albon@mobi.ch>  
An: "mathis.fux@gmail.com" <mathis.fux@gmail.com>

18. Juli 2019 um 10:55

Sehr geehrter Herr Fux

Gemäss Telefonat vom 18. Juli 2019 bestätige ich Ihnen, dass Sie das erarbeiteten BIA-System (Prototyp) getestet haben, jedoch nicht die geforderten 80% erreichen.

Der Prototyp dient der Mobiliar als Basis für die Weiterentwicklung und die anschliessende Überführung in die Produktion im Jahr 2020. Während dieser Phase werden erneut Tests durchgeführt werden, die jedoch die geforderten 80% erreichen müssen.

Ich hoffe, ich konnte Ihnen mit diesen Angaben dienen.

Freundliche Grüsse

Damian In-Albon

---

**die Mobiliar**

Schweizerische Mobiliar Versicherungsgesellschaft AG

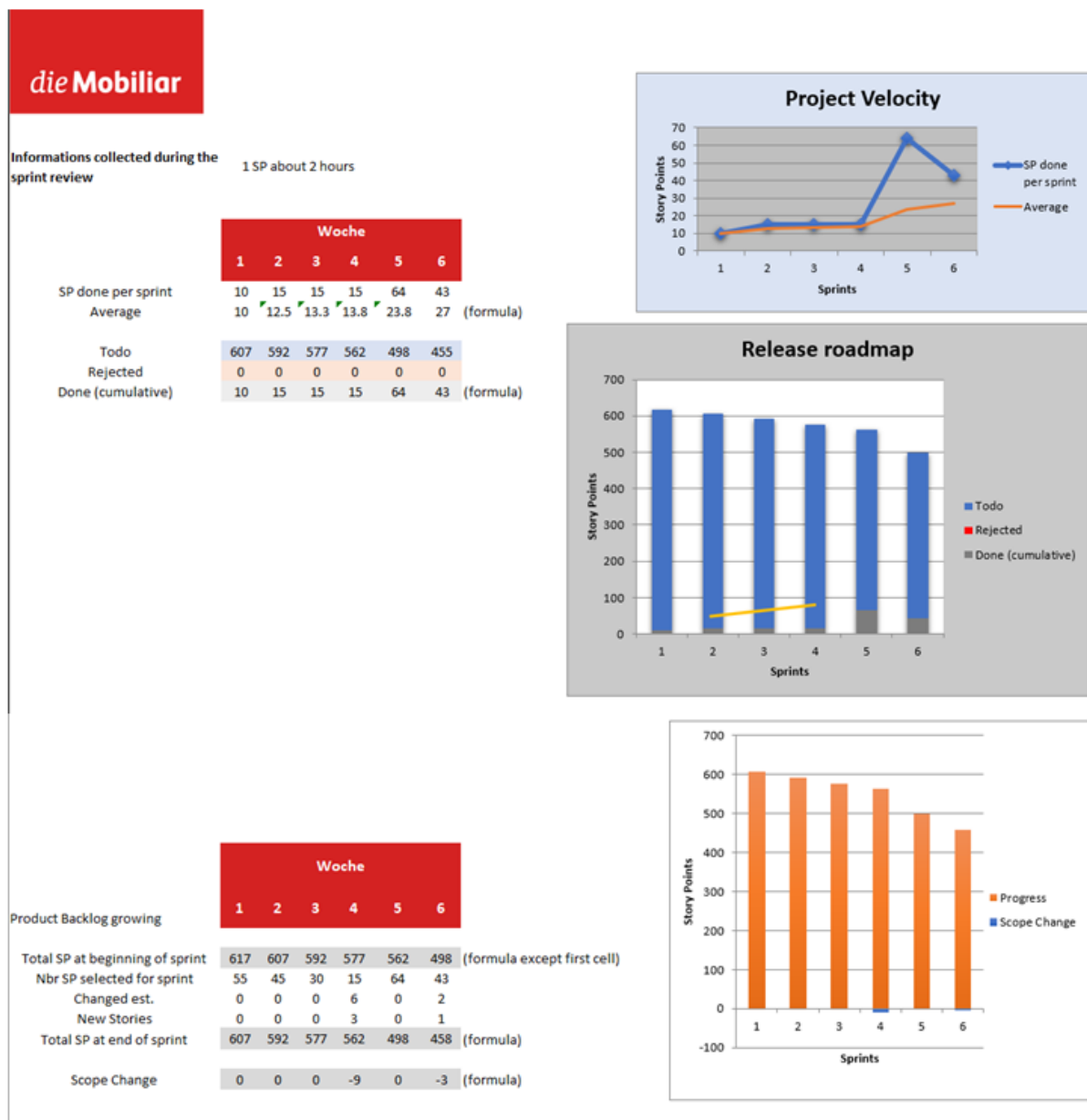
Bundesgasse 35, 3001 Bern

(Eigene Darstellung)

## ANHANG IX: RELEASE ROADMAP

In der Abbildung 92 kann der aktuelle durchschnittliche Wert an umgesetzten Punkten betrachtet werden. Dieser Wert beträgt 27. Da jedoch in den letzten Iterationswochen mehr gearbeitet wurde, wurde gemäss dem Abteilungsleiter der Wert auf 21 gesetzt. Dies bedeutet, dass in der Woche 21 Punkte umgesetzt werden können. Insgesamt sind nun noch 458 Punkte zur Vervollständigung des Systems nötig. Diese Differenz impliziert, dass noch 22 Wochen benötigt werden, um das gesamte BIA System zu entwickeln.

Abbildung 92: Release Roadmap



(Eigene Darstellung)

# ANHANG X: IMPLEMENTATIONSITERATION

Abbildung 93: Implementationsiteration Woche 1



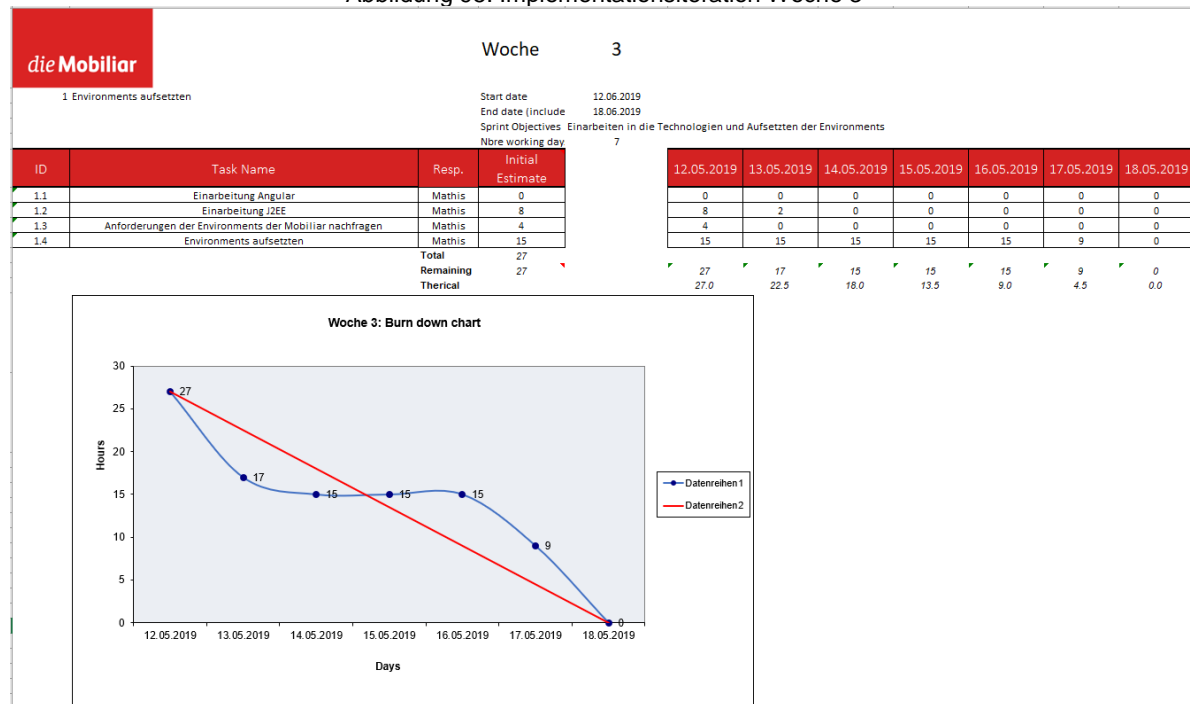
(Eigene Darstellung)

Abbildung 94: Implementationsiteration Woche 2



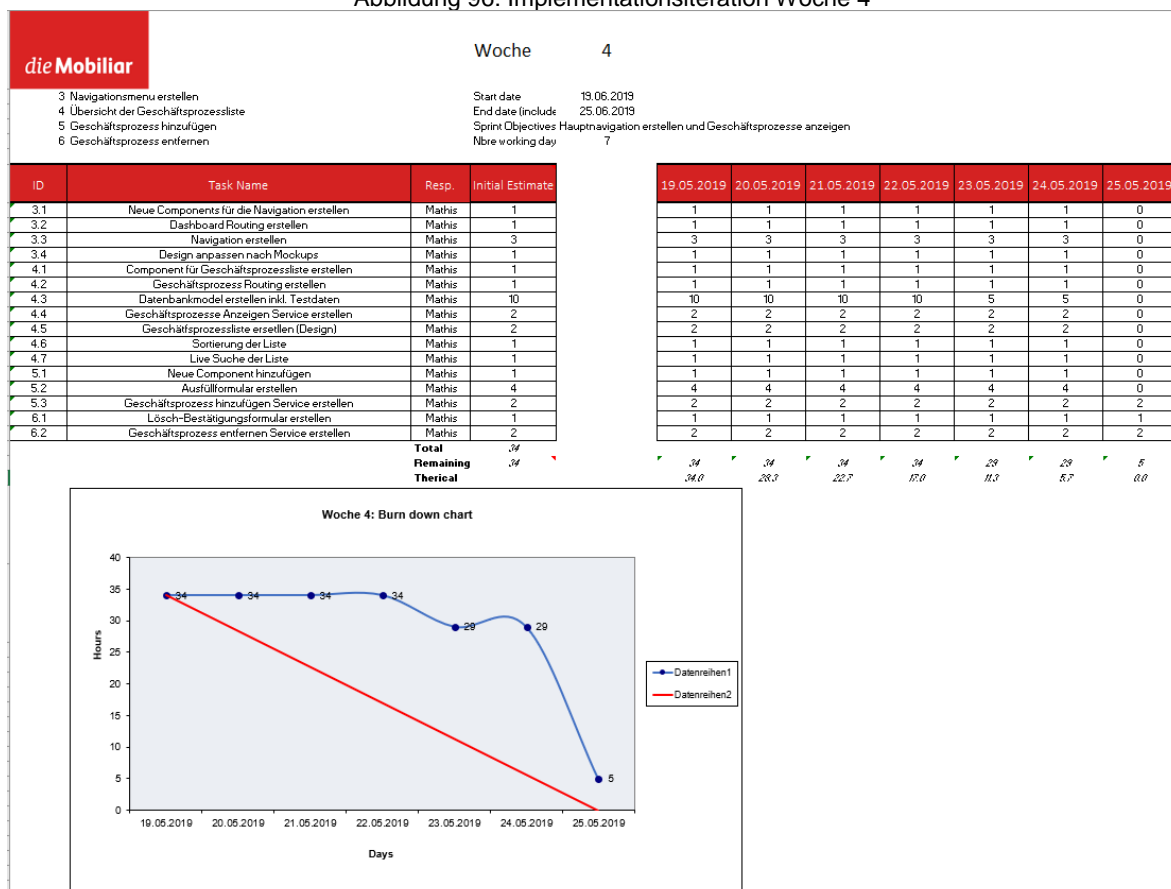
(Eigene Darstellung)

Abbildung 95: Implementationsiteration Woche 3



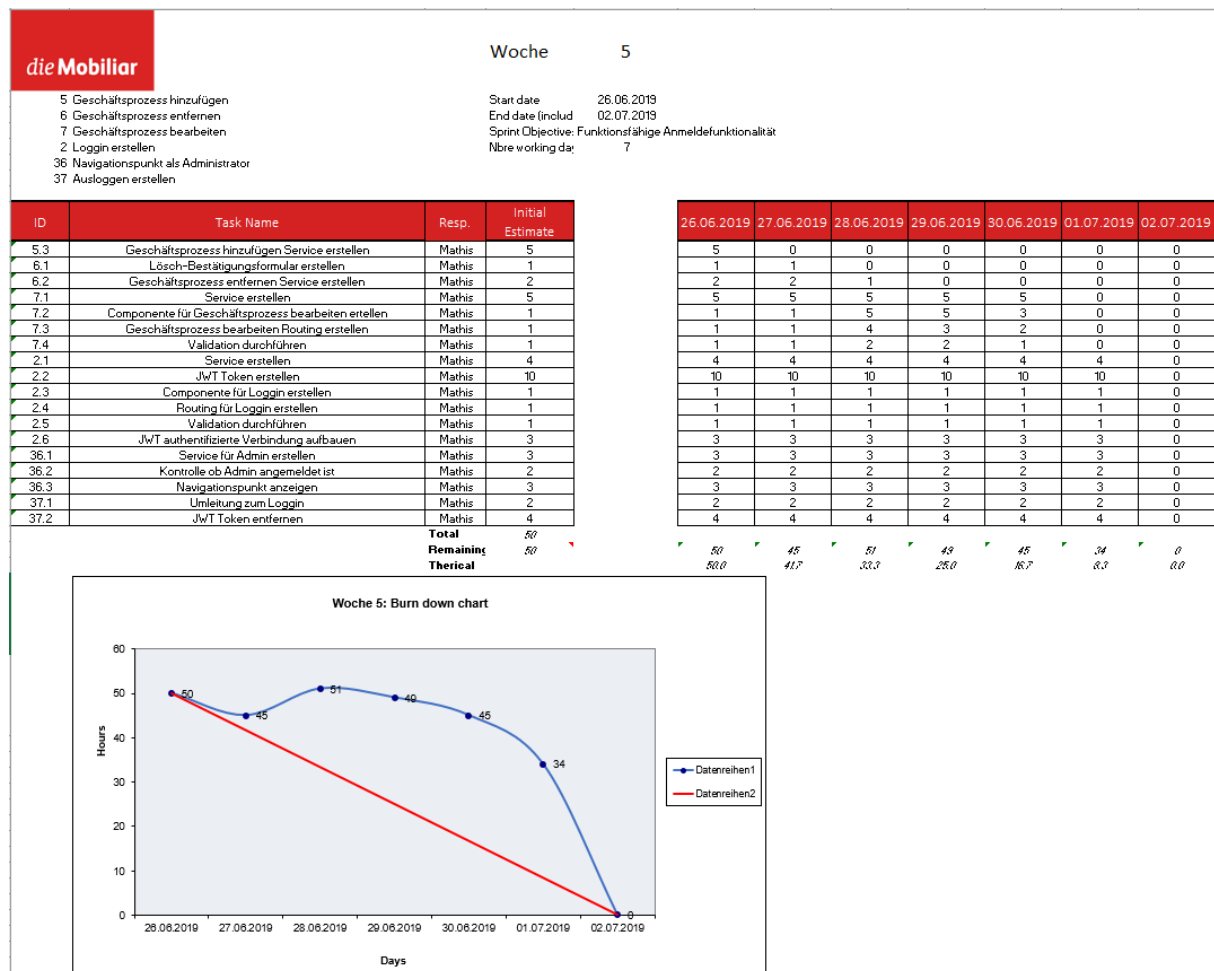
(Eigene Darstellung)

Abbildung 96: Implementationsiteration Woche 4



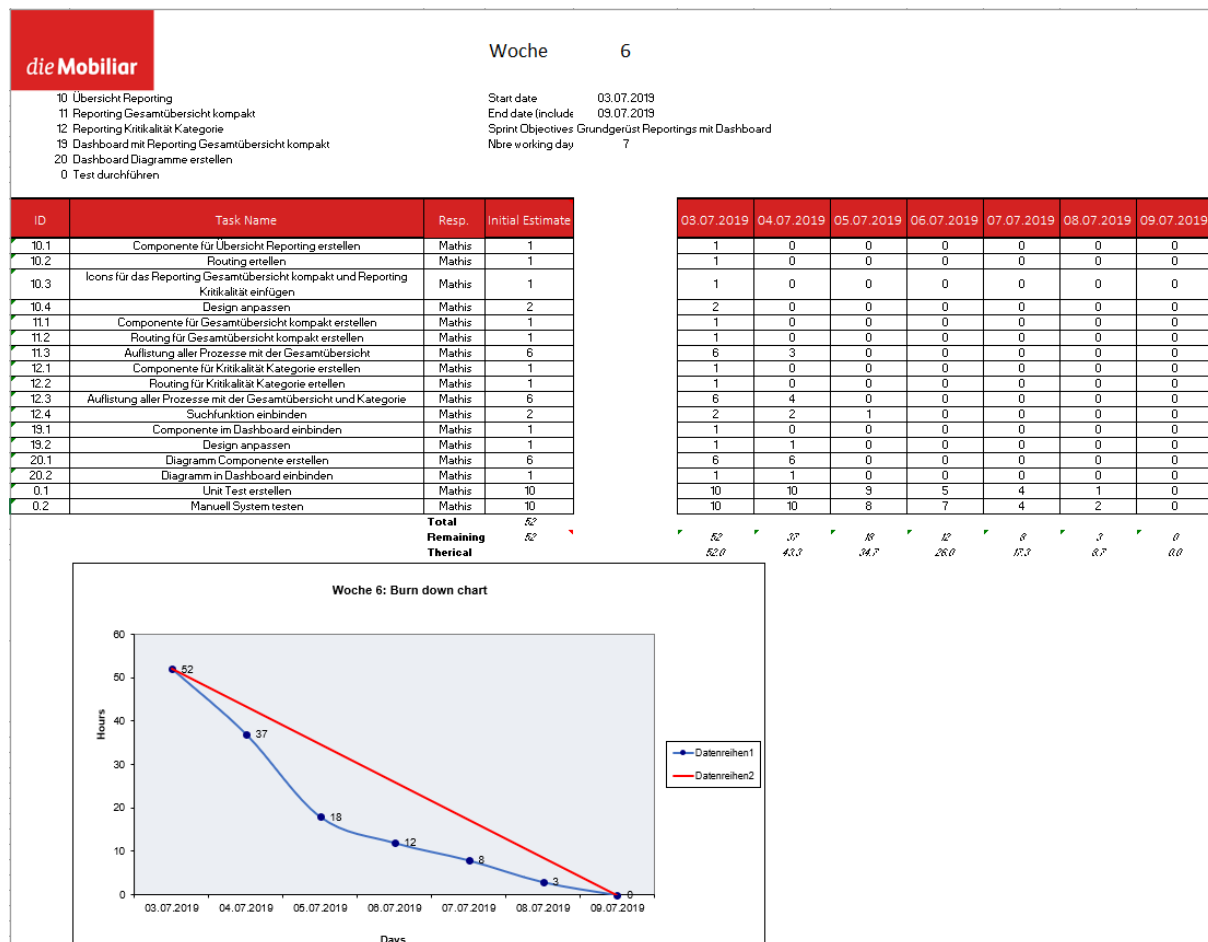
(Eigene Darstellung)

Abbildung 97: Implementationsiteration Woche 5



(Eigene Darstellung)

Abbildung 98: Implementationsiteration Woche 6



(Eigene Darstellung)

## ANHANG XI: SITZUNGSPROTOKOLL 06.05.2019

### Allgemeine Informationen

Datum:	06.05.2019
Besprechungszeit:	60min
Besprechungsart:	Brig-Glis
Teilnehmer:	Herr Damian In-Albon Mathis Fux

### Traktanden

- Ist-Analyse: Aktuelle Funktionen
- Pros / Cons des aktuellen Programmes
- Verwendete Projektmanagement Methode besprechen
- Gewünschte Funktionalitäten
- Funktionalitäten Product Backlog bestimmen (zu realisieren/fakultativ)
- Besprechung weiteres Vorgehen

### Zusammenfassung

#### Ist-Analyse: Aktuelle Funktionen

- Drei Excel-Dateien
- BIA (Hauptdatei mit allen Prozessen)
- Reporting (Analysen der Prozesse)
- Massnahmen (Massnahmen der Prozesse)
- Dieses Jahr werden die Prozesse bearbeitet

#### Pros / Cons des aktuellen Programmes

##### Pros

- Reporting Funktion ist sehr von Vorteil, um Analysen daraus zu erstellen
- Filterfunktion in Excel, damit genau, dass angezeigt wird was gewünscht ist

##### Cons

- Kein Multiusersystem
- Wartung der Daten ist mühsam
- Arbeitsschritte sind zu komplex
- Hohe Fehlerquelle
- Benötigt grosses Know-How, um das System zu bedienen
- Es existieren keine Workflows für externe Mitarbeiter
- Excel-Dateien sind überfüllt und daher sehr langsam

#### Verwendete Projektmanagement Methode besprechen

- In allen Projekten wird SCRUM, Kanban oder Safe als Projektmethodologie verwendet
- Wäre daher von Vorteil, wenn dies im Projekt übernommen werden kann

### **Gewünschte Funktionalitäten**

- Einfachere Arbeitsschritte
- Multiusersystem
- Workflow generieren können
  - o E-Mail an externer Mitarbeiter senden, welcher nur die ausgewählten Prozesse für eine bestimmte Zeit bearbeiten kann
- Reporting
  - o Gesamtübersicht kompakt
  - o Übersicht Kritikalität
  - o Arbeitsplatz
  - o Lieferanten
    - Lieferanten pro Prozess
    - Lieferantenliste
    - Lieferanten Kritikalität Kategorie
  - o Krit. Funktionen Kat
  - o (IT Fakultativ)

### **Funktionalitäten Product Backlog bestimmen (zu realisieren/fakultativ)**

- Zu realisieren sind folgende Funktionen nach Prioritätenliste in einem agilen System
  1. Geschäftsprozesse erstellen/bearbeiten/löschen
  2. (IT-Liste hinzufügen muss aber noch abgeklärt werden)
  3. Reportings inkl. Kritikalitäten
  4. Massnahmen
  5. Login erstellen
  6. Workflow erstellen
  7. Mitteilung an die unterschiedlichen Benutzer senden

### **Besprechung weiteres Vorgehen**

- Eine Woche wird noch für die Analysen verwendet
- Anschliessend wird das Design erstellt
- Und voraussichtlich Montag dem 27.05.2019 wird eine neue Sitzung geplant

### **Nächste Besprechung**

- Ist-Analyse wird für das bestehende Programm erstellt
- Ist-Analyse für potenzielle Programme erstellt
- Kontaktaufnahme mit Herr Grütter
- Product Backlog erstellt

## ANHANG XII: SITZUNGSPROTOKOLL 29.05.2019

### Allgemeine Informationen

Datum:	29.05.2019
Besprechungszeit:	30min
Besprechungsart:	Per Skype
Teilnehmer:	Herr Damian In-Albon Mathis Fux

### Traktanden

- Information bzgl. Präsentation Bachelorarbeit
- Entscheidung Mockups
- Entscheidung über potenzieller Technologie
- Entwicklungsvariante Azure Cloud besprechen

### Zusammenfassung

#### Information bzgl. Präsentation Bachelorarbeit

- Für die Schweizerische Mobiliar wird die gleiche Abschlusspräsentation im Standort Bern gehalten

#### Entscheidung Mockups

- Folgende Mockups sind angenommen:
  - o Login
  - o Dashboard
  - o Prozess anzeigen
  - o IT
  - o Reporting / Reporting Ansicht
  - o Benutzer verwalten
  - o Workflows anzeigen
- Folgende Mockups müssen noch Anpassungen vorgenommen werden:
  - o Prozess hinzufügen (Prozess ID Feld hinzufügen)
  - o Prozess bearbeiten (Prozess ID Feld hinzufügen)
- Folgende Mockups muss noch erstellt werden:
  - o Workflow hinzufügen

#### Entscheidung über potenzieller Technologie

- Entscheidung fällt auf Variante 4 - Entwicklung nach Mobiliar
- Ist effizienter für das Unternehmen, da die Applikation anschliessend übernommen und weitergeführt werden kann

#### Entwicklungsvariante Azure Cloud besprechen

- Frontend wird Angular verwendet
  - o Daraufhin als ein statischer Web Server auf Azure Cloud einrichten
- Backend wird J2EE Web Services verwendet
  - o Dies wird als Web App in Azure Cloud eingerichtet inkl. Webserver als WildFly

### Weiteres Vorgehen

- Jeden Mittwoch um 08:30 wird ein Meeting mit dem Arbeitsgeber durchgeführt, um den aktuellen Stand zu präsentieren.

# ANHANG XIII: PREISLISTE «ALIVE-IT BUSINESS CONTINUITY FRAMEWORK»

Abbildung 99: Preisliste «alive-IT Business Continuity Framework»



## Preisliste 2017

### Lizenzkauf Software [alive-IT] (einmalig)

Produkt Nr.	Name	CC-User Range*	Preis pro CC-User*
a0003	alive-IT 0003	01-03	2.500,00 €
a0015	alive-IT 0015	04-15	1.950,00 €
a0025	alive-IT 0025	16-25	1.850,00 €
a0040	alive-IT 0040	26-40	1.750,00 €
a0055	alive-IT 0055	41-55	1.700,00 €
a0070	alive-IT 0070	56-70	1.650,00 €
a0085	alive-IT 0085	71-85	1.600,00 €
a9999	alive-IT Company Server	>85	1.550,00 €
a1999	alive-IT Second Server**	---	4.000,00 €
aOLAP	instantOLAP***	---	7.500,00 €

#### \*) Concurrent Users

Die Lizenzen für [alive-IT] basieren auf einem Concurrent User-Modell. Sie bezahlen nur für die Anzahl von Benutzern, die das System zur gleichen Zeit nutzen, unabhängig von der Anzahl der Benutzer, die als User definiert sind.

#### \*\*) Hochverfügbarkeit

Beinhaltet eine zweite Server-Lizenz für die Backup-/ DR-Lokation. Kann für Backup-Zwecke mehrfach verwendet werden.

#### \*\*\*) instantOLAP

Beinhaltet eine Server-Lizenz für die Erstellung von grafischen Berichten.

### Wartung

Die Wartung und Pflege von [alive-IT] umfasst den Hotline-Support Mo - Fr (ausgenommen bundesweite deutsche Feiertage) 9 - 17 Uhr, Lieferung aller Patches, sowie die kostenlose Überlassung von Releases und neuen Versionen.

Jährlich findet ein Kundentreffen zum Erfahrungsaustausch statt, zu dem Sie als Kunde eingeladen sind.

Im Kaufpreis ist eine 12-monatige Gewährleistung (ohne Updates, nur Bug-Fixes) enthalten. Wir empfehlen den Abschluss eines Wartungsvertrages.

Die Wartungsgebühr für eine Lizenz beträgt pro Jahr 18% des einmaligen Listenpreises der Lizenz. Wartungsgebühren werden zum Beginn des Wartungszeitraumes fällig und gelten jeweils für 12 Monate ab Abschluss.

Bei Lizenzenerweiterungen erhöht sich die Wartungsgebühr entsprechend mit Beginn des nächsten Kalenderjahres.

Alle Preise verstehen sich zzgl. MWST.

Quelle: Erhalten von der Schweizerischen Mobiliar

## ANHANG XIV: DEPLOYMENT BLOB STORAGE

Pietschmann, C. (2016). *Static Website Hosting in Azure Storage*. o.O.: Build Azure. Abgerufen am 10. 05 2019 von <https://buildazure.com/static-website-hosting-in-azure-storage/>

Die nachfolgende Anleitung dient dazu einen statischen Webserver auf Microsoft Azure Cloud einzurichten. Dies kann verwendet werden um das Frontend, welches auf Angular basiert, in der Azure Cloud bereitzustellen. Der zertifizierte Azure Lösung Architekt Pietschmann Chris beschreibt das Bereitstellen eines statischen Webserver in den folgenden vier Schritten.

1. Der erste Schritt besteht darin einen Azure Storage Account einzurichten. Dazu kann das Azure Portal verwendet werden, welches die Arbeit erleichtert.
2. Nachdem der Azure Storage Account eingerichtet wurde, kann über das Azure Portal einen neuen Blob Container erstellt werden. Damit Zugriff auf den Container gewährleistet wird, muss die Zugriffsart entweder auf Blob oder Container gestellt werden.
3. Durch die Installation des Azure Storage Explorer auf dem lokalen Computer, kann dieser verwendet werden, um den statischen Inhalt bereitzustellen.
4. Über einen Browser kann nun auf die statische Webseite, die auf Azure Storage gehostet wird, zugegriffen werden. Die URL setzt sich aus der URL des Storage Account, dem Namen des Containers und dem Namen der Datei oder des Blobs. Ein Beispiel einer URL sieht wie folgt aus:

*<https://staticwebsitehosting.blob.core.windows.net/content/index.html>*

Bei der Verwendung eines statischen Webserver muss beachtet werden, dass der Name des Containers in der URL erwähnt werden muss. Dies kann auch bei der Verwendung einer eigenen Domain nicht umgangen werden. Zudem muss beachtet werden, dass der statische Webserver von Azure Cloud kein Standarddokument besitzt. Deshalb wird der Anwender bei nur der Eingabe des Domainnamens nicht automatisch auf die Webseite weitergeleitet. Eine weitere Einschränkung besteht derzeit noch darin, dass bei der Verwendung einer eigenen Domain, keine SSL-Verschlüsselung aufgebaut werden kann.

## ANHANG XV: DEPLOYMENT JAVA WEB APP

Benz, B. (2016). *Deploy Java Web Apps to Azure using Eclipse*. o.O.: Microsoft Azure. Abgerufen am 10. 05 2019 von <https://azure.microsoft.com/es-es/blog/deploy-java-web-apps-to-azure-using-eclipse/>

Für die Bereitstellung des Backends, welches auf J2EE basiert, wird in Microsoft Azure Cloud eine Web App eingerichtet. Die folgende Anleitung dient dazu, ein J2EE-Projekt auf einem Web App auf der Azure Cloud freizugeben. Der Cloud Entwickler Benz Brian beschreibt das Freigeben eines Eclipse Projektes auf die Azure Cloud in den folgenden fünf Schritten.

1. Einrichtung eines Web App auf der Microsoft Azure Cloud. Dieser Vorgang kann über das Azure Portal durchgeführt werden.
2. Installation des Azure Toolkit for Eclipse über den Eclipse Marketplace. Danach können die Azure Dienste in Eclipse verwendet werden kann.
3. Rechtsklick auf das Projekt und Im Kontextmenu «Azure» selektieren, danach kann durch der Navigationspunkt «Publish as Azure Web App» das Projekt freigegeben werden.
4. Daraufhin müssen folgende Informationen angegeben werden: Name des App Services, einen Speicherort, die Preisstufe und eine Instanz-Grösse.
5. Nach den Angaben kann das Projekt auf der Azure Cloud bereitgestellt werden.

Diese Anleitung basiert auf die Verwendung von Eclipse als Entwicklungsumgebung. Eine andere Entwicklungsumgebung kann verwendet werden. Jedoch muss bei diesen jeweils auch ein Plugin installiert werden, um die Azure Dienste verwenden zu können.

## ANHANG XVI: ABNAHME DES PROTOTYPEN

Abbildung 100: E-Mail Abnahme des Prototypen

---

**In-Albon Damian** <damian.in-albon@mobi.ch>  
An: "mathis.fux@gmail.com" <mathis.fux@gmail.com>

23. Juli 2019 um 15:28

Sehr geehrter Herr Fux

Herzlichen Dank für den Stick mit Ihrer Arbeit. Der Prototyp entspricht voll und ganz unseren Erwartungen und hilft uns für die Weiterentwicklung des Tools sehr.

Wir danken Ihnen für die sehr angenehme Zusammenarbeit und freuen uns, mit Ihnen im nächsten Jahr die Software fertig zu stellen und in den Betrieb zu übergeben.

Freundliche Grüsse  
Damian In-Albon

(Eigene Darstellung)

## SELBSTÄNDIGKEITSERKLÄRUNG

«Ich bestätige hiermit, dass ich die vorliegende Bachelorarbeit alleine und nur mit den angegebenen Hilfsmitteln realisiert habe und ausschliesslich die erwähnten Quellen benutzt habe. Ohne Einverständnis des Studiengangsleiters und des für die Bachelorarbeit verantwortlichen Dozierenden sowie des Forschungspartners, mit dem ich zusammengearbeitet habe, werde ich diesen Bericht an niemanden verteilen, ausser an die Personen, die mir die wichtigsten Informationen für die Verfassung dieses Berichts geliefert haben und die ich nachstehend aufzähle: Dr. Christiane Jungius, Damian In-Albon, Emanuel Indermühle».

Mathis Fux

Glis, den 31. Juli 2019