

Sperrvermerk

Auf Wunsch der Firma Fusonic GmbH, im Auftrag der Viterma Handels GmbH, ist die vorliegende Arbeit bis zum [DATUM] für die öffentliche Nutzung zu sperren.

Veröffentlichung, Vervielfältigung und Einsichtnahme sind ohne ausdrückliche Genehmigung der oben genannten Firma und dem Verfasser nicht gestattet. Der Titel der Arbeit sowie das Kurzreferat/Abstract dürfen jedoch veröffentlicht werden.

Dornbirn, am [Datum]

Unterschrift des Verfassers (Dominic Luidold)

Firmenstempel

[Titel der Arbeit]

[Untertitel der Arbeit]

Bachelorarbeit I
zur Erlangung des akademischen Grades

Bachelor of Science in Engineering (BSc)

Fachhochschule Vorarlberg
Informatik – Software and Information Engineering

Betreut von
Prof. (FH) Dipl. Inform. Thomas Feilhauer

Vorgelegt von
Dominic Luidold
Dornbirn, [Datum]

Widmung

Trotz des Sperrvermerks, an der diese Arbeit gebunden ist, möchte ich es mir nicht nehmen lassen, den Personen einen Dank auszusprechen, die mich bei der Umsetzung und Realisierung meiner Bachelorarbeit I unterstützt haben.

Zu Beginn möchte ich meinen Betreuern Thomas Feilhauer (Fachhochschule Vorarlberg) und Michael Zangerle (Fusonic GmbH) danken, die jederzeit ein offenes Ohr für Fragen, Anliegen und Unklarheiten hatten. Durch ihre fachliche Kompetenz, ihre langjährige Erfahrung sowie ihr Know-how konnte ich vieles lernen, das ich in mein weiteres Berufsleben mitnehmen kann.

Als nächstes möchte ich meiner gesamten Familie danken, denen ich während des Schreibens dieser Arbeit wahrscheinlich mehr als nur einmal auf die Nerven gegangen bin. Nur durch die Tatkräftige Unterstützung mit Snacks, Süßigkeiten und dem besten Zimmerservice der Welt konnte ich so ungestört an der Bachelorarbeit schreiben. Ohne ihr Korrekturlesen wären zudem mehr Tippfehler vorhanden als mir lieb sind und ich auf meine Tastatur schieben kann.

Zu guter Letzt möchte ich meine Arbeit all jenen widmen, die für die Gleichbehandlung und Gleichberechtigung aller kämpfen, sich für eine Sache einsetzen, die größer als sie selbst ist und für die sie teilweise sogar ihr Leben aufs Spiel setzen. Auch 2020 braucht es weltweit immer noch laute Stimmen - Stimmen, die sich nicht unterkriegen lassen.

„The time is always right to do what is right.“

Dr. Martin Luther King Jr.

Kurzreferat

[Deutscher Titel Ihrer Arbeit]

[Text des Kurzreferats]

Abstract

[English Title of your thesis]

[text of the abstract]

Vorwort

Der Verfasser der vorliegenden Arbeit bekennt sich zu einer geschlechtergerechten Sprachverwendung.

Um die Lesbarkeit zu gewährleisten und zugunsten der Textökonomie werden die verwendeten Personen beziehungsweise Personengruppen fix männlich oder weiblich zugeordnet. Zum Beispiel wird immer „die Entwicklerin“ und „der Benutzer“ verwendet. Es wurde besonders darauf geachtet, stereotype Rollenbeschreibungen zu vermeiden. Die insgesamt eventuell dadurch hervorgerufene Irritation bei den Lesenden ist gewünscht und soll dazu beitragen, eine Bewusstheit für die bestehende, Frauen diskriminierende Sprachgewohnheit (generelle Verwendung der männlichen Begriffe für beide Geschlechter) zu wecken beziehungsweise zu stärken.

Inhaltsverzeichnis

Abbildungsverzeichnis	10
Quellcodeverzeichnis	11
Tabellenverzeichnis	12
Abkürzungsverzeichnis	13
1 Einleitung	14
1.1 Arbeitgeber Fusonic GmbH	14
1.2 Nutzung & Umfeld des „Better Life System“	15
1.3 Problemstellung	16
1.4 Anforderungen	18
1.5 Zielsetzung	18
2 Stand der Technik	19
2.1 Aufbau des Backends	19
2.2 Standard Symfony Architektur	20
Literaturverzeichnis	21
[evtl. Anhang]	22
Eidesstattliche Erklärung	23

Abbildungsverzeichnis

1.1	Teilausschnitt der BLS-Ordnerstruktur	17
-----	---	----

Quellcodeverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

BLS Better Life System

CI Continuous Integration

CD Continuous Deployment

CQRS Command Query Responsibility Segregation

CRUD Create, Read, Update, Delete

DTO Data Transfer Object

IDE Integrierte Entwicklungsumgebung

LTS Long Term Support

1 Einleitung

Diese Bachelorarbeit verfolgt das Ziel, einen Einblick in die Implementierung und Erweiterung des bereits bestehenden Backend-Systems des „Better Life System“ (kurz „BLS“) der Viterma Handels GmbH mit dem sogenannten „CQRS“-Pattern zu geben.

Das Better Life System - welches von Fusonic GmbH entwickelt wird und mittels einer Weboberfläche und gängigen Webbrowsern bedient werden kann - ermöglicht es Endkunden, mithilfe einer Handelsvertreterin der Firma Viterma (beziehungsweise mit einer Vertreterin eines ihrer Franchise-Partner) ein Badezimmer anhand der jeweiligen Bedürfnisse auszusuchen und zu konfigurieren, um sich schlussendlich ein entsprechendes Angebot dafür erstellen zu lassen.

1.1 Arbeitgeber Fusonic GmbH

Die Fusonic GmbH hat ihren Firmensitz in Götzis, Vorarlberg, und besteht aus einem Team von aktuell mehr als 25 Angestellten, Softwareentwicklerinnen und Projektleiterinnen. Diese sind aufgeteilt in diverse kleinere Teams, die intern unter anderem „Duck-Team“ beziehungsweise „Parrot-Team“ genannt werden. Die Teams arbeiten dabei an jeweils eigenständigen Projekten und setzen diverse Technologien ein. Zu den verwendeten Technologien gehören unter anderem C# mit .NET, PHP mit Symfony und JavaScript/TypeScript mit Angular. (vgl. Fusonic GmbH 2020, "Übersicht aller Technologien") Während es regelmäßigen Austausch unter den Teams gibt, besteht jedes aus Frontend- sowie Backend-Entwicklern, da die meisten Projekte - aufgrund der zugrundeliegenden Anwendungsfälle - aus einem Web-Frontend sowie serverseitigen Backend bestehen.

1.2 Nutzung & Umfeld des „Better Life System“

Die Hauptaufgabe des Better Life System liegt darin, als computergestütztes und plattformunabhängiges Tool, Vertreterinnen rund um die Viterma Handels GmbH bei der Konfiguration und Zusammenstellung eines Badezimmers - zusammen mit dem Endkunden (meist Haus- beziehungsweise Wohnungsbesitzer oder Hotels) - zu unterstützen und diesen Vorgang zu erleichtern. Das BLS wird des Weiteren dazu verwendet, aktuelle und abgeschlossene Angebote zu verwalten, Produkte, Produktinformationen und dazugehörige Preisaufschläge zu pflegen und Kundendaten abzulegen. Zudem dient es als Kontrollinstanz für in Auftrag gegebene Sanierungen, um sicherzustellen, dass alle benötigten Materialien und Teile in entsprechenden Mengen vorhanden und kompatibel sind.

Die Nutzung des BLS findet zu großen Teilen auf mobilen Rechnern beziehungsweise Laptops statt, die während der Beratung und Betreuung von Kunden zum Einsatz kommen. Da nicht immer gewährleistet werden kann, dass eine aufrechte Internetverbindung besteht, ist die Offline-Fähigkeit bei gleichbleibender Nutzung im Webbrowser ein wichtiger Bestandteil des Web-Frontends.

Aufgrund der Anforderungen der Viterma Handels GmbH, dass das Better Life System offline ebenso wie online verwendet werden kann, basiert das Frontend auf dem TypeScript Web-Framework Angular¹, welches solch eine Funktionalität ohne eine Installation oder zusätzliche Voraussetzungen am Endgerät unterstützt. Im Gegensatz zu weniger komplex gehaltenen Frontends - die primär Daten darstellen, die vom Backend eingehen - ist dieses beim BLS für einen Großteil der Ablauflogik, Berechnung von Preisen und Generierung von diversen PDFs zuständig. Das Frontend wird, da der Fokus dieser Arbeit auf der Entwicklung im Backend der Anwendung liegt, in weiterer Folge jedoch nicht genauer beleuchtet und als gegeben angenommen. Es kann davon ausgegangen werden, dass eingehende Anfragen an das Backend aus Benutzereingaben beziehungsweise zeitlich gesteuerten Events resultieren.

¹Angular (<https://angular.io>)

Das Backend, welches alle API-Anfragen bearbeitet und als Schnittstelle zur Datenbank dient, ist in der Programmiersprache PHP geschrieben und verwendet das Framework Symfony² sowie diverse „Bundles“, die darauf aufbauen. Ein detaillierterer Überblick über den Aufbau, bisher angewandte Konzepte sowie technische Begebenheiten werden in Kapitel 2 auf Seite 19 behandelt.

Während die Nutzung des Better Life System auf allen Plattformen - die einen modernen Webbrowser anbieten - möglich ist, findet die Entwicklung primär auf der Linux-Distribution Ubuntu statt. Der Grund für diese Betriebssystemwahl ist damit zu erklären, dass sowohl die lokale Entwicklung als auch die Continuous Integration Umgebung (genutzt für automatisierte Tests) und das Continuous Deployment mittels Docker³ Containern stattfindet. Für die Entwicklung im Backend-Bereich der Anwendung wird die IDE PhpStorm verwendet, während im Frontend auf Visual Studio Code gesetzt wird.

1.3 Problemstellung

Die Entwicklung des BLS hat im Frühjahr 2017 begonnen und die Anwendung ist seit dem stetig weiterentwickelt worden. Über die Zeit haben sich entsprechend sowohl die Anforderungen und Wünsche des Kunden als auch die technischen Begebenheiten, Best Practices und Möglichkeiten verändert.

Die bisher bei der Umsetzung des Projekts verwendete Struktur sowie der Aufbau wurde stark an die empfohlene Herangehensweise von Symfony angelehnt und die Code-Basis entsprechend darauf ausgelegt (siehe dazu Symfony 2020, „The Symfony Framework Best Practices“). Durch wachsende Anforderungen und entsprechend benötigten Programmcode, der diese erfüllt, hat sich jedoch ein komplexes System ergeben, dessen *Controller*, *Entities*, *Manager (Services)* und *Data Transfer Objects (DTOs)* auf viele Verzeichnisse und Unterverzeichnisse verteilt sind. Die Abbildung 1.1 auf Seite 17 zeigt einen Ausschnitt von Ordnern die im Projekt vorkommen und jeweils alle Klassen ihrer Art beinhalten (im Verzeichnis „Controller“ befinden sich, um ein Beispiel zu nennen, alle Controller des Projekts).

²Symfony (<https://symfony.com>)

³Docker (<https://docker.com>)

📁 Command	📁 Import
📁 Controller	📁 Manager
📁 DataFixtures/ORM	📁 Migrations
📁 Doctrine/Type	📁 Model
📁 Dto	📁 Repository
📁 Entity	📁 Security
📁 Event	📁 Task
📁 EventListener	📁 Utility
📁 Exception	📁 Validation

Abbildung 1.1: Teilausschnitt der BLS-Ordnerstruktur

Durch die bisher gewählte Herangehensweise und den entsprechenden Aufbau hat sich jedoch ergeben, dass eine logische Gruppierung bzw. Kapselung von zusammengehörenden Klassen, Controllern, Managern etc. eines API Endpunkts nur schwer möglich ist. Das hat zur Folge, dass das System an manchen Stellen sehr komplex aufgebaut ist und externe Personen eine gewisse Zeit brauchen, um sich mit internen Abläufen vertraut machen zu können. Neben der Verständlichkeit ist dadurch in weiterer Folge auch die Wartbarkeit und einfache Erweiterbarkeit im Falle von neuen Funktionalitäten nicht im vollen Ausmaß gegeben. Das Testen einzelner Bestandteile mittels sogenannter *Unit Tests* findet im Moment nur an wenigen Stellen statt. Primär wird beim Better Life System auf *Funktionale Tests* gesetzt, die im Problemfall jedoch nur grundsätzlich darauf hindeuten können, wo ein Fehler aufgetreten ist.

Da sich die Anforderungen an das Backend auch in Zukunft noch weiterentwickeln können und eine Skalierung der Infrastruktur beziehungsweise Abspaltung einzelner Teile (in sogenannte Microservices) nicht ausgeschlossen werden kann, bedarf es Möglichkeiten, dies möglichst effektiv und ohne großen, zusätzlichen Aufwand umsetzen zu können. Die von Symfony empfohlene Herangehensweise kann dies dabei nur bedingt unterstützen, was auch in diesem Fall einen limitierenden Faktor darstellt.

1.4 Anforderungen

Die in Abschnitt 1.3 angeführten Umstände und den daraus resultierenden Herangehensweisen, die nicht mehr in vollem Maße zufriedenstellend sind, führen zu neuen Anforderungen an das Better Life System. Diese sollen entsprechend umgesetzt werden, um ein zukunftsicheres Backend für die gesamte Anwendung garantieren zu können und die Entwicklung neuer sowie bestehender Funktionalitäten zu erleichtern und zu vereinheitlichen.

Zum Start des Berufspraktikums Anfang Juli 2020 stand bereits fest, dass in weiterer Folge das CQRS-Pattern (was für *Command Query Responsibility Segregation* steht) zum Einsatz kommen wird. Andere Projekte der Fusonic GmbH, die primär auf C# und .NET basieren, haben bereits gute Erfahrungen damit gemacht und gehen deshalb davon aus, dass der Einsatz Patterns die Qualität der bestehenden Code-Basis verbessern wird.

Die Anforderungen, die im Zuge des Berufspraktikums und folglich dieser Bachelorarbeit zu erfüllen sind, bestehen darin, das aktuell bestehende System, neben dem „Tagesgeschäft“ (sprich Bugfixes, Feature Requests etc.), laufend umzustellen und neue Funktionalitäten entsprechend mit der neuen Struktur und dem Pattern umzusetzen.

1.5 Zielsetzung

Die in den Abschnitten 1.3 und 1.4 angeführten Punkte ergeben folgende Ziele, die im Verlauf des Berufspraktikums - so weit wie möglich - umzusetzen sind.

Als Ziele einzustufen sind:

- Vorbereitung der bestehenden Infrastruktur auf CQRS in Zusammenarbeit mit Mitarbeitern der Fusonic GmbH
- Umstellung der bestehenden API Endpunkte auf CQRS bei Beibehaltung von möglichst viel Programmlogik
- Umsetzung neuer Funktionalitäten mittels CQRS
- Weiterhin Abwickeln des Tagesgeschäftes

2 Stand der Technik

Das folgende Kapitel gibt einen Überblick über den aktuellen Stand des Backends des Better Life System, dessen technischem Aufbau und dem zugrundeliegenden Konzept. Anschließend wird die von Symfony empfohlene Architektur beleuchtet und auf das zukünftig eingesetzte CQRS-Pattern eingegangen. Am Ende des Kapitels werden die Symfony Standard-Architektur und CQRS gegenübergestellt und Vor- sowie Nachteile aufgezeigt.

2.1 Aufbau des Backends

Das Backend des BLS basiert auf der Programmiersprache PHP und setzt hierbei auf die zum aktuellen Zeitpunkt (Stand August 2020) neueste Version *PHP 7.4*. Zur Verwaltung benötigter Abhängigkeiten kommt das Paketverwaltungssystem Composer¹ zum Einsatz, welches alle benötigten Symfony Bundles und anderweitige Packages sowohl im Produktivsystem als auch während der Entwicklung zur lokalen Installation sowie Nutzung zur Verfügung stellt.

Als Grundgerüst für das Backend dient das PHP-Framework Symfony in der LTS-Version *4.4*, wobei zum Zeitpunkt des Verfassens dieser Arbeit Vorbereitungen für einen Umstieg auf Version *5.x* getroffen werden. Das Backend dient als Schnittstelle zwischen dem Angular-Frontend und einem Datenspeicher, wobei hierfür eine relationale Datenbank auf Basis von MySQL eingesetzt wird.

Der grundsätzliche Aufbau des Symfony Backends entspricht aktuell der empfohlenen Vorgehensweise laut Symfony Dokumentation und wird im folgenden Abschnitt 2.2 auf Seite 19 erläutert.

¹Composer (<https://getcomposer.org>)

2.2 Standard Symfony Architektur

Literatur

Fusonic GmbH (2020). *Web, Mobile, Desktop-Anwendungen*. Fusonic, Vorarlberg. URL: <https://www.fusonic.net/de/leistungen/> (besucht am 04.08.2020).

Symfony (2020). *The Symfony Framework Best Practices (Symfony Best Practices)*. Library Catalog: symfony.com. URL: https://symfony.com/doc/current/best_practices.html (besucht am 06.08.2020).

[evtl. Anhang]

Formatvorlage für den Fließtext.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit I selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dornbirn, am [Datum]

Dominic Luidold