FH Vorarlberg
University of Applied Sciences

TBD

Bachelorarbeit II
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, 20. Mai 2021

Widmung

TODO

„TODO“
TODO

Kurzreferat

TODO

TODO

Abstract

TODO

TODO

Geschlechtergerechte Sprache

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

Abkürzungsverzeichnis

MPA Multi-page Application

PWA Progressive Web Application

SPA Single-page Application

UI User Interface

1 Einleitung

Diese Bachelorarbeit verfolgt das Ziel, einen Einblick in die Single-page Application (SPA) Frameworks *Angular*[1] und *Vaadin*[2] zu geben und deren Gemeinsamkeiten, Unterschiede sowie Vor- und Nachteile zu beleuchten.

Um ein grundlegendes Verständnis über die Thematik von Single-page Applications zu erlangen, wird zu Beginn der Arbeit auf das Konzept einer SPA eingegangen und die zugrundeliegende Herangehensweise mit der einer klassischen Multi-page Application (MPA) beziehungsweise eines Rich Clients verglichen. Im weiteren Verlauf werden die unterschiedlichen Ansätze von Angular und Vaadin genauer betrachtet und eine tatsächliche Umsetzung der zuvor erläuterten Technologien mittels zweier Demo-Applikationen getestet. Am Ende dieser Arbeit wir darauf eingegangen, ob sich - anhand unterschiedlicher Kriterien und Anwendungsfälle - eine Empfehlung für eines der beiden SPA Frameworks aussprechen lässt.

1.1 Motivation

In den letzten Jahren lässt sich beobachten, dass Webapplikationen, Apps und Anwendungen allgemein verstärkt mittels des SPA-Ansatzes umgesetzt werden und somit auf einen Thin Client - im Gegensatz zu klassischeren Rich Clients und Multi-page Applications - setzen. Für die Umsetzung einer solchen Applikation stehen eine Vielzahl von Frameworks zur Verfügung, die darüber hinaus weitere Features bieten und Entwickler:innen bei der Umsetzung unterstützen.

[1] Angular (https://angular.io)
[2] Vaadin (https://vaadin.com)

Die richtige Wahl des Frameworks, der jeweiligen Technologien und der im Hintergrund agierenden Strukturen spielen eine wesentliche Rolle bei der Planung und Umsetzung eines neuen Projektes. Welches Framework sich besser eignet, lässt sich oftmals nicht auf den ersten (oder sogar zweiten) Blick feststellen. Diese Arbeit befasst sich daher genauer mit dem Konzept von Single-page Applications und vergleicht zwei darauf aufbauende Frameworks, die mit deutlich unterschiedlichen Technologie-Stacks arbeiten und zu vergleichbaren Lösungen führen.

1.2 Problemstellung

Die in Abschnitt 1.1 auf Seite 10 angesprochene Vielzahl an SPA-Frameworks bietet grundlegend den Vorteil, dass eine große Auswahlmöglichkeit und eine gewisse Konkurrenz untereinander zu einem hohen Qualitätsstandard führt. Zudem wird dadurch sichergestellt, dass es für jedes Projekt - unabhängig von den jeweiligen Anforderungen und etwaigen Eigenheiten - eine Möglichkeit gibt, dieses mit einem der verfügbaren Frameworks umzusetzen. Auf der anderen Seite führt die stetig wachsende Anzahl an Möglichkeiten jedoch dazu, dass sich meist nur schwer beurteilen lässt, welches Framework sich für die Umsetzung einer Applikation bestmöglich eignet.

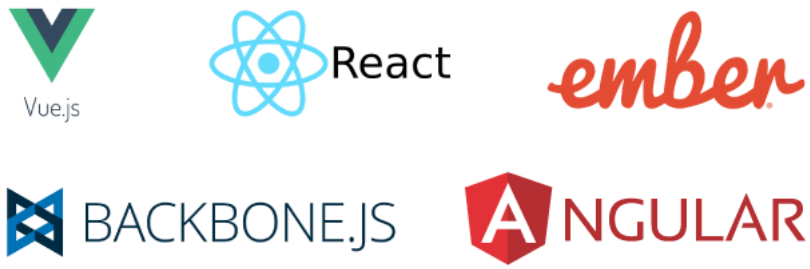

Abbildung 1.1: Liste möglicher JavaScript-Frameworks zur Umsetzung von Single-page Applications (Quelle: A. 2020)

Um eine geeignete Wahl eines Frameworks treffen zu können, sollten vorab Kriterien und Anforderungen definiert werden, die schlussendlich erfüllt werden

müssen. Neben den projektspezifischen Eigenheiten, die in den meisten Fällen von einer Vielzahl der Frameworks abgedeckt werden können, stellt sich die Auswahl der zugrundeliegenden Technologien als eine der größten Herausforderungen dar. Diese Entscheidung muss gut überlegt und abgewogen werden, da diese im weiteren Verlauf weitreichende Folgen bei der Umsetzung einer (Web-) Applikation zur Folge hat und ein Umstieg nach begonnener Entwicklung nur unter großem Aufwand umsetzen lässt.

Die in Abbildung 1.1 auf Seite 11 dargestellten Frameworks zeigen eine Auswahl an Frameworks auf, die auf *JavaScript* aufbauen beziehungsweise basieren und somit primär im Frontend - dem Browser - eingesetzt werden können. Single-page Applications lassen sich jedoch nicht nur Frontend-seitig entwickeln (bei denen ein Großteil der Logik auf einem externen Server läuft), sondern können ebenfalls mittels auf *Java* basierenden Frameworks umgesetzt werden, die sowohl die Logik als auch das User Interface (UI) kombinieren, zu denen unter anderem Vaadin gehört.

Da die unterschiedlichen Ansätze, sowohl hinter Vaadin als auch Angular, gewisse Vor- und Nachteile sowie Tücken mit sich bringen, fällt die Wahl auf eines der beiden SPA-fähigen Frameworks auf den ersten Blick nicht leicht. Hinzu kommt die Frage, welches der Frameworks weiterführende Funktionalitäten bietet, um mit geringem Aufwand beispielsweise eine Progressive Web Application (PWA) umzusetzen oder anwendungsspezifische Daten lokal sowie extern persistieren zu können.

1.3 Zielsetzung

Die in den Abschnitten 1.1 und 1.2 angeführten Punkte haben aufgezeigt, dass die große Anzahl an Frameworks, mit denen Single-page Applications umgesetzt werden können, zwar sehr positiv einzuschätzen ist, die damit verbundenen Probleme bei der Auswahl des richtigen Frameworks werden dadurch jedoch verstärkt. Aufgrund der unterschiedlichen zugrundeliegenden Technologien und einhergehenden Herangehensweisen ist eine bedachte Wahl wichtig.

Diese Arbeit verfolgt daher das Ziel, das JavaScript Framework *Angular* dem auf Java basierenden Framework *Vaadin* gegenüberzustellen und zu vergleichen. Das Ziel ist es, mittels Literatur belegter Vergleiche einen Allgemeinen Überblick über Single-page Applications zu geben, diese klassischen Ansätzen gegenüberzustellen und zwei Demo-Applikationen zu entwickeln. Diese Webapplikationen werden dann herangezogen, um anhand von vorab definierten Kriterien feststellen zu können, ob und in wie weit Empfehlungen für eines der beiden Frameworks ausgesprochen werden kann.

Um den Fokus dieser Arbeit genau zu definieren und einzuschränken, wird bei der Planung, Umsetzung sowie abschließenden Beurteilung der Applikationen anhand der Kriterien auf folgende Punkte beschränkt:

- Möglichkeit zur einfachen Umsetzung einer Progressive Web Application (PWA)
- Möglichkeit der Wiederverwendbarkeit von Komponenten, gegebenenfalls mittels *Web Components*
- Möglichkeit Daten lokal sowie extern (serverseitig) zu persistieren

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 vorgeschlagene Architektur beleuchtet und auf das zukünftig eingesetzte CQRS-Pattern eingegangen. Am Ende des Kapitels werden die gängige Schichtenarchitektur und CQRS gegenübergestellt und Vor- sowie Nachteile aufgezeigt.

Literatur

A., Sviatoslav (Jan. 2020). *The Best JS Frameworks for Front End*. URL: `https://rubygarage.org/blog/best-javascript-frameworks-for-front-end` (besucht am 08. 02. 2021).

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 20. Mai 2021 Dominic Luidold