

**Hochschule Osnabrück**

University of Applied Sciences

**Fakultät**

**Ingenieurwissenschaften und Informatik**

Schriftliche Ausarbeitung zum Thema:

Auftragsverwaltungssystem

im Rahmen des Moduls

Webanwendungen

des Studiengangs Informatik-Medieninformatik

|  |  |
| --- | --- |
| Autor: | Jan Weimer, Nelson Morais |
| Matr.-Nr.: | 868487, 879551 |
| E-Mail: | [jan.weimer@hs-osnabrueck.de](mailto:jan.weimer@hs-osnabrueck.de),  nelson.morais@hs-osnabrueck.de |
| Dozent: | Björn Plutka |

Abgabedatum: 09.03.2022

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc422394453)

1.1 Vorstellung des Themas 3

[1.2 Vorstellung des Themas 3](#_Toc422394454)

[1.3 Zielgruppe 3](#_Toc422394455)

[2 Darstellung der Grundlagen 4](#_Toc422394457)

[2.1 Technische Grundlagen 4](#_Toc422394458)

[2.2 Konzept 5](#_Toc422394459)

[3 Anwendung 7](#_Toc422394460)

[4 Zusammenfassung und Fazit 11](#_Toc422394461)

# Einleitung – Jan Weimer

## Vorstellung des Themas

Das Thema der Projektarbeit handelt von der Realisierung eines Auftragsverwaltungssystems in Form einer nativen App auf einem mobilen Endgerät. Hier können Teams eines Unternehmens eine Übersicht über vorhandene und neue Aufträge bekommen. Hier besteht die Möglichkeit Aufträge von einem schwarzen Brett zu untersuchen und anzunehmen. Dabei werden die angenommenen Aufträge in der Teamseite gezeigt und verschwinden vom schwarzen Brett. Außerdem soll die App ein sogenanntes Threadboard besitzen, womit andere Unternehmensmitglieder sich miteinander verständigen können. Dazu können die Profildetails jederzeit verändert werden.

## Zielgruppe

Zielgruppe dieser App sind Unternehmen, die von Kunden Aufträge entgegennehmen. Dabei ist es nicht wichtig um was für eine Art von Unternehmen es sich handelt. Optimalerweise wird die App von Firmen genutzt, die eine Arbeitsweise in Teams pflegen wollen.

## Aufbau des Berichts

Zuerst werden alle Grundlagen erklärt, die zum Verständnis der Realisierungsweise des Projekts nötig sind. Danach wird auf das Konzept eingegangen. Was ist der Plan zum Aufbau der Oberfläche? Anschließend werden dann alle Schritte erklärt, die gemacht wurden, um das Ergebnis zum Projektende zu erreichen. Abschließend wird ein Fazit aus dem Projekt gezogen.

# Darstellung der Grundlagen – Jan Weimer

## Technische Grundlagen

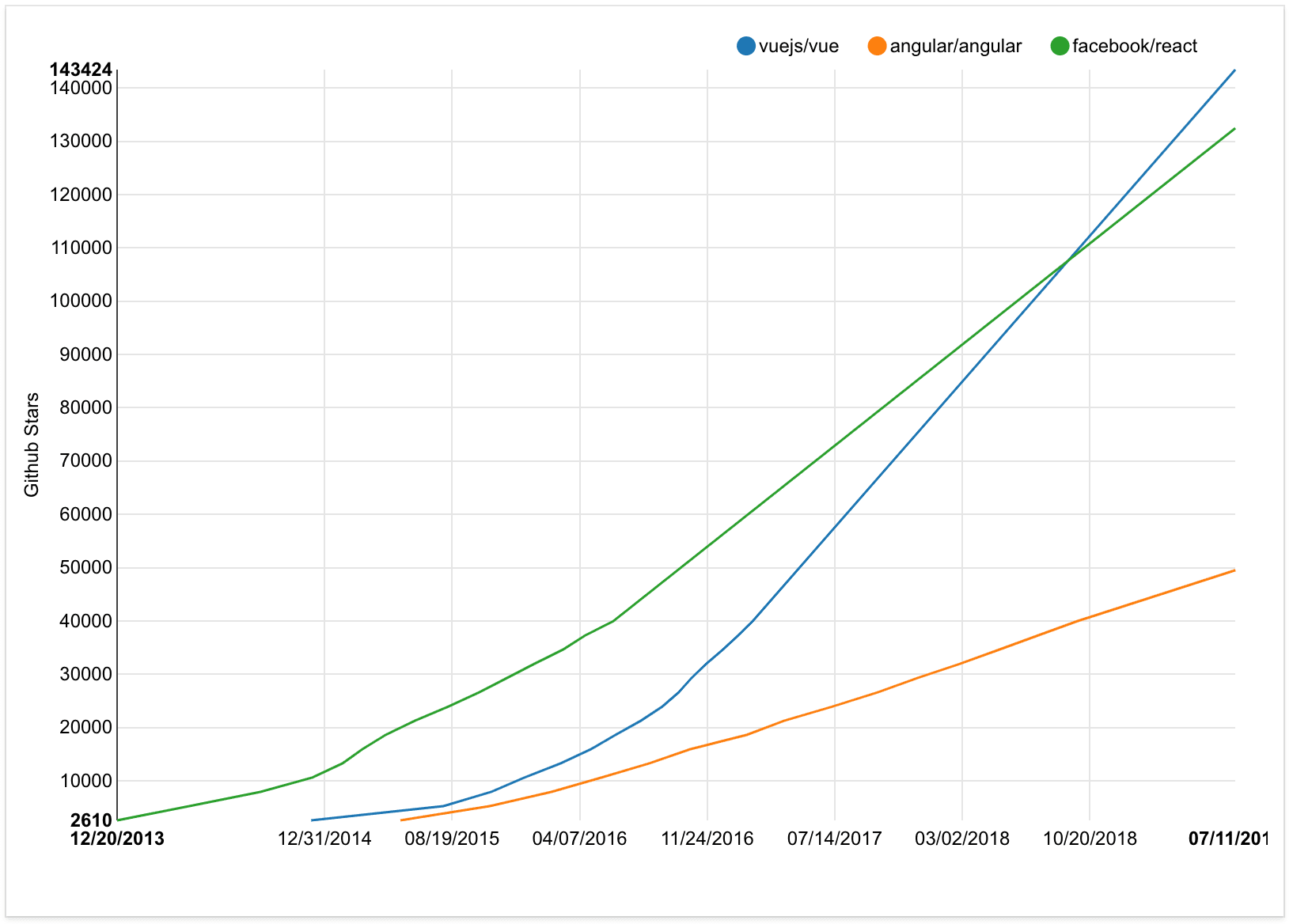
**Vue.js**

****

Abbildung 1: Vue.js Logo

Vue.js ist ein clientseitiges JavaScript-Framework, welches sehr leichtgewichtig ist und komponentenweise arbeitet. Dadurch können bessere Ordnerstrukturen geschaffen werden und ausgelagert werden, welches der Software-Architektur zugutekommt. Der Code wird dadurch lesbarer und besser wartbar. Die Leichtgewichtigkeit resultiert in der winzigen Downloadgröße von ca. 18 KB.

Die Flexibilität von Vue.js stellt jedoch auch Risiken dar. Dadurch kann es passieren, dass in Unternehmensteams viele Entwickler einen anderen Codingstil haben und dadurch andere Probleme bekommen, diese Realisierungsschritte nachzuvollziehen. Demnach sind intern festgesetzte Codingrichtlinien eine gute Möglichkeit das Risikoausmaß zu minimieren. Z.B. kann festgehalten werden das Listen und Tabellen immer in Komponenten ausgelagert werden müssen.

  
Abbildung 2: Popularität zwischen Frameworks über Zeit

**Quasar**

****Abbildung 2: Quasar Logo

Quasar ist ein Framework, welches auf Vue.js basiert. Der Entwickler bekommt dadurch eine Richtlinie, die dabei helfen sollen, eine benutzerfreundliche Oberfläche zu erstellen, die dazu noch responsiv ist. Da die Anwendung am Ende auf mobilen Endgeräten laufen soll, kann hier Capacitor genutzt werden. Der Vorteil liegt darin, dass der Code nur einmal geschrieben werden muss und diese App in egal welcher Art rausgebracht werden kann.

## Konzept

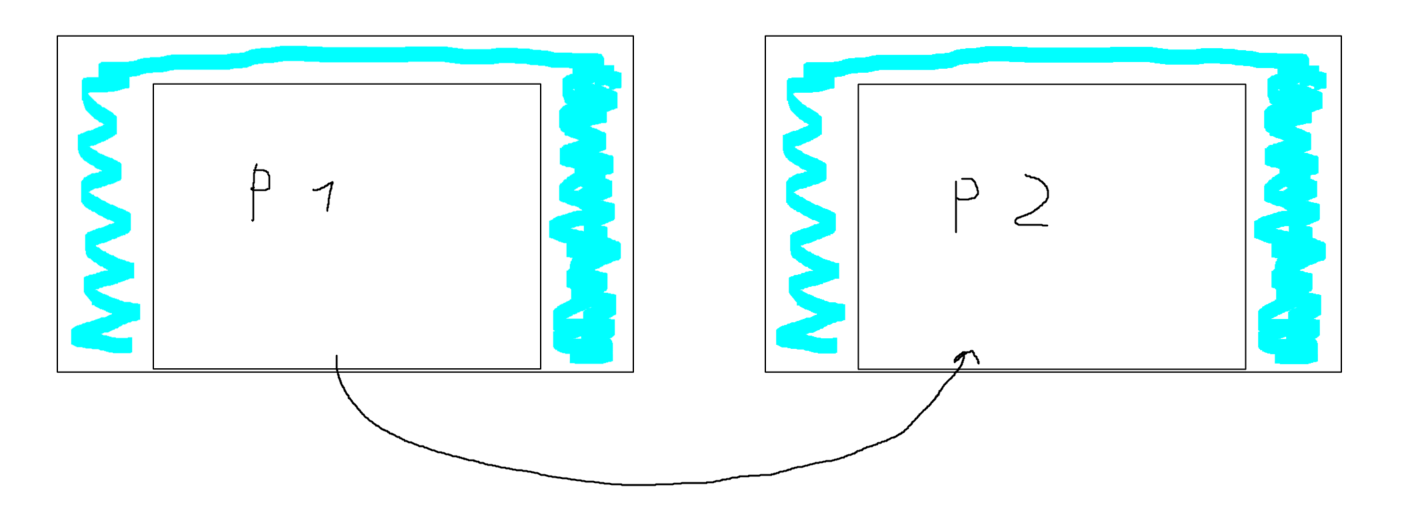
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 3: Layout Builder von Quasar

Mit dem Layout Builder kann das Layout erstellt werden, welches für die Anwendung verwendet werden soll. Hier wurde sich für ein visible Header und ein left- und right-side Drawer entschieden. Der Gedanke darin besteht den left-side Drawer als Leiste zu nutzen, um auf die gewünschten Threads zu kommen. Der right-side Drawer soll die Menüspalte sein, wo man aufs Profil, Team und schwarze Brett gelangen kann.

Innerhalb der App befinden sich die Pages, wo hin- und wegnavigiert werden kann. Möglichst soll vieles als Component realisiert werden, um eine bessere Trennung zu schaffen.

Abbildung 4: Routingbeispiel: hier ist das Feld in blauer Farbe das Layout, welches sich nicht ändert. Lediglich wird die Seite geändert.

# Anwendung

**Layout**

Das Layout wird durch die *MainLayout.vue* realisiert. Diese ist über alle Pages vorhanden. Sie besitzt drei Komponenten, einmal ThreadLink, ThreadForm und EssentialLink.

Ein Bild, das Text enthält.

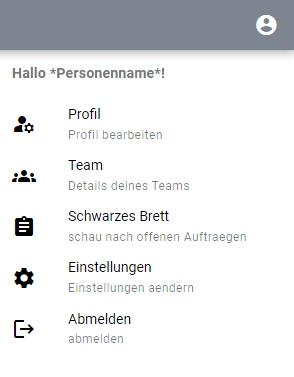
Automatisch generierte BeschreibungAbbildung 5: left-side Drawer

Die linke Leiste zeigt alles was um Threads handelt. Hier sollen mit ToggleButtons nur globale, Team- und eigene Threads jeweils separat angezeigt werden können. Der ThreadLink zeigt den Titel des Threads und das Datum. Der Button „ADD“ ist die ThreadForm-Komponente. Durch das Klicken auf dieser öffnet sich ein Dialog mit dem Inhalt des Formulars zur Erstellung eines Threads.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung Abbildung 6: Formular Thread erstellen

Hier kann man den Titel, die Beschreibung und aussuchen, ob es global oder teamintern sichtbar sein soll.

Abbildung 7: right-side Drawer

Hier soll die Navigation zwischen den Pages stattfinden. Oben wird der Name des Nutzers dargestellt. Darunter sind die EssentialLinks, die Titel, Untertitel, Icon und Link beinhalten. Die Icons werden von material.io benutzt und werden auch von quasar unterstützt.

# Zusammenfassung und Fazit

Am Ende wurde eine App realisiert, worauf das Design auf Tablets angepasst ist. Es ist ein funktionierendes Frontend für ein Threadboard entstanden. Dabei wird unterteilt und gefiltert in global, teamintern und eigene Threads. Außerdem ist die Main-Page das Schwarze Brett mit allen offenen Aufträgen. Dazu kann ein Auftrag angenommen und untersucht werden. Beim Besuchen der ProfilPage, werden die Daten aus der API geladen, um die Werte für Name und Team zu füllen. Darüber hinaus können diese im Frontend verändert, wodurch mit Betätigung des Zuständigen Buttons, die neuen Werte zur API zurückgesendet wird. Um welchen Nutzer es sich handelt wird anfangs in den Properties festgelegt.

Die Arbeit mit Vue.js und dem dazugehörigen Framework Quasar war zufriedenstellend. Der rasante Anstieg der Popularität konnte durch die erstmalige Arbeit im Rahmen dieses Projektes nachvollzogen werden. Die einzelnen Komponenten sind schnell und separat realisierbar.