

# EazyMenu

## DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:

Benjamin Besic

Bozidar Spasenovic

David Ignjatovic

Betreuer:

Prof. DDI Clemens EISSERER

Projektpartner:

Oberösterreichische Versicherung AG, Linz

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

Benjamin Besic & Bozidar Spasenovic & David Ignjatovic

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an alle Geschlechter.

# Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!* Suspendisse vel felis.

Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ausgangssituation . . . . .	1
1.2	Ist-Zustand . . . . .	1
1.3	Problemstellung . . . . .	1
1.4	Aufgabenstellung . . . . .	1
1.5	Ziel(e) . . . . .	2
1.5.1	Zielgruppe . . . . .	2
<b>2</b>	<b>Planung</b>	<b>3</b>
2.1	Use Cases . . . . .	3
2.2	UI entwickeln . . . . .	3
<b>3</b>	<b>State of Art</b>	<b>4</b>
<b>4</b>	<b>Verwendete Technologien</b>	<b>5</b>
4.1	IntelliJ IDEA . . . . .	5
4.2	Git . . . . .	5
4.2.1	Versionskontrolle . . . . .	5
4.2.2	Git Funktionsweise . . . . .	6
4.2.3	Git Befehle . . . . .	7
	Clone . . . . .	7
	Commit . . . . .	7
	Push . . . . .	7
	Pull . . . . .	7
	Branch . . . . .	8
4.2.4	GitHub . . . . .	8
4.3	Java . . . . .	9
4.4	Java EE . . . . .	9

4.5	Maven . . . . .	9
4.5.1	Java EE vs. Quarkus . . . . .	9
4.5.2	JPA . . . . .	9
4.5.3	Hibernate . . . . .	9
4.5.4	Panache . . . . .	9
4.6	JBoss . . . . .	9
4.7	Cypress . . . . .	9
4.8	Keycloak . . . . .	9
4.9	Oracle Datenbank . . . . .	9
4.10	Vue.js . . . . .	9
4.10.1	Vue Funktionsweise . . . . .	10
	Model-View-Viewmodel (MVVM) Pattern . . . . .	10
	Vue Instance . . . . .	11
	Lifecycle Diagram . . . . .	11
	Vue Components . . . . .	12
4.11	Angular vs. Vue . . . . .	13
4.12	HTML . . . . .	13
4.13	CSS . . . . .	13
4.14	Javascript . . . . .	13
4.15	Json Web Token (JWT) . . . . .	13
4.16	Progressive Web App(PWA) . . . . .	13
4.17	Google Charts . . . . .	13
	Deepest . . . . .	13
<b>5</b>	<b>Implementierung</b>	<b>14</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>16</b>
	<b>Literaturverzeichnis</b>	<b>VI</b>
	<b>Abbildungsverzeichnis</b>	<b>VII</b>
	<b>Tabellenverzeichnis</b>	<b>VIII</b>
	<b>Anhang</b>	<b>IX</b>

# **1 Einleitung**

## **1.1 Ausgangssituation**

Die Oberösterreichische Versicherung ist der Marktführer im Versicherungsbereich in Oberösterreich und beschäftigt in ihrer Zentrale in Linz über 500 Mitarbeiter. Das Unternehmen besitzt eine Kantine, wo täglich 3 Hauptspeisen serviert werden. Dazu noch eine Vorspeise und eine Nachspeise.

## **1.2 Ist-Zustand**

Die derzeitige Bestellmöglichkeit funktioniert über eine simple Datenbankanwendung mit IBM Notes. Dieses Programm ist auf jedem Rechner installiert und jeder Mitarbeiter ist mit seinen Daten bereits eingeloggt. Auf einem simplen Interface kann man zwischen den heutigen und zukünftigen Mahlzeiten wählen und die Uhrzeit, wann man diese konsumiert. Nach erfolgreicher Bestellung hat man eine kleine Übersicht über die vergangenen Bestellungen.

## **1.3 Problemstellung**

Das derzeitige Programm ist sehr veraltet und nicht besonders benutzerfreundlich. Außerdem läuft das Programm lokal auf jedem Rechner und eine mobile Bestellung ist ausgeschlossen. Außerdem ist zu erwähnen ist, dass der Benutzer nur eine beschränkte Möglichkeit hat sein Bestellverhalten zu visualisieren bzw. zu analysieren.

## **1.4 Aufgabenstellung**

Die Aufgabenstellung der Firma ist eine Webanwendung zu entwickeln, die den Vorgänger ablöst und ein modernes und benutzerfreundliches Interface hat. Zudem soll eine Bestellung über das Smartphone möglich sein. Zusätzlich soll ein Empfehlungssystem entwickelt werden, dass einem ermöglicht Mahlzeiten zu bestellen, die auf einen

abgestimmt sind. Außerdem soll man eine Einsicht über seine Bestellhistorie haben mit diversen Statistikelementen.

## 1.5 Ziel(e)

- Erleichterung des Bestellprozesses für den Benutzer
- Flexible Bestellmöglichkeiten
- Der Benutzer hat mehr Verständnis über sein Bestellverhalten

### 1.5.1 Zielgruppe

Mitarbeiter der OÖ Versicherung AG



## **2 Planung**

### **2.1 Use Cases**

### **2.2 UI entwickeln**

## 3 State of Art

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Citing [1] properly.

Was ist eine Globally Unique Identifier (GUID)? Eine GUID kollidiert nicht gerne.

Kabellose Technologien sind in abgelegenen Gebieten wichtig [2].

# 4 Verwendete Technologien

## 4.1 IntelliJ IDEA

IntelliJ IDEA ist eine der führenden Entwicklungsumgebungen für die Programmiersprache Java. Sie wurde vom Unternehmen JetBrains im Jahre 2000 entwickelt.

Außerdem bietet sie ebenfalls Entwicklungsmöglichkeiten für Kotlin, Groovy, Scala und auch Android. Sie ist immer auf dem neusten Entwicklungsstand, wird laufend mit Updates versorgt und unterstützt die derzeit gängigen Programmierertools wie Docker, Kubernetes, Maven, Datenbank-Tools, Git, Jakarta EE und viele weitere. Es gibt eine kostenpflichtige Ultimate Version und eine Community Version, die kostenfrei zur Verfügung gestellt wird.

IntelliJ zeichnet auch die Anzahl an Erweiterungen mittels Plugins aus. Die Umgebung besitzt auch eine sehr intuitive Intelligenz, die es dem Entwickler sehr einfach macht damit zu programmieren. [3]

Wir haben uns dafür entschieden, da wir damit viel Erfahrung hatten und die oben genannten Punkte unterstützten unsere Entscheidung enorm.

## 4.2 Git

Git ist ein Versionskontrollsystem (oft abgekürzt durch VCS) für Entwickler. Es ist ein Open-Source System, das im Jahre 2005 von Linus Torvald entwickelt wurde. Laut einer Stack Overflow-Umfrage von Entwicklern nutzen über 87 % der Entwickler Git. [4] Zu aller erst muss man den Begriff Versionskontrolle erklären, um Git zu verstehen.

### 4.2.1 Versionskontrolle

Diese dient dazu, um den originalen Quellcode effizient mit mehreren Personen editieren bzw. entwickeln zu können.

Die Entwickler arbeiten mit Verzweigungen und Zusammenführungen. Jeder Entwickler kann Änderungen sicher durchführen, ohne seine Kollegen dabei zu behindern. Diese

Änderungen können dann, sobald sie funktionsfähig sind, wieder in den Hauptquellcode eingebunden werden. Alle Änderungen sind zurückzuverfolgen und bei Bedarf kann man sie dann wieder zurücksetzen. [4]

### 4.2.2 Git Funktionsweise

Jeder Entwickler hat seine eigene Version des Projekts (Working Directory), die er frei bearbeiten kann. Diese bekommt man durch einen Klon des Projekts (Clone).

Diese Änderungen kann man aufteilen und in Paketen bereitstellen, nach dem man diese durch Commits trennt. Einen Commit kann man benennen.

Diese Commits kann man dann online veröffentlichen durch einen Push. Ein Push ist nur möglich, wenn man die aktuellste Version des Projekts auf seinen Rechner gezogen hat (Pull). Einem Push kann man einen bestimmten Zweig (Branch) zuordnen. Diese

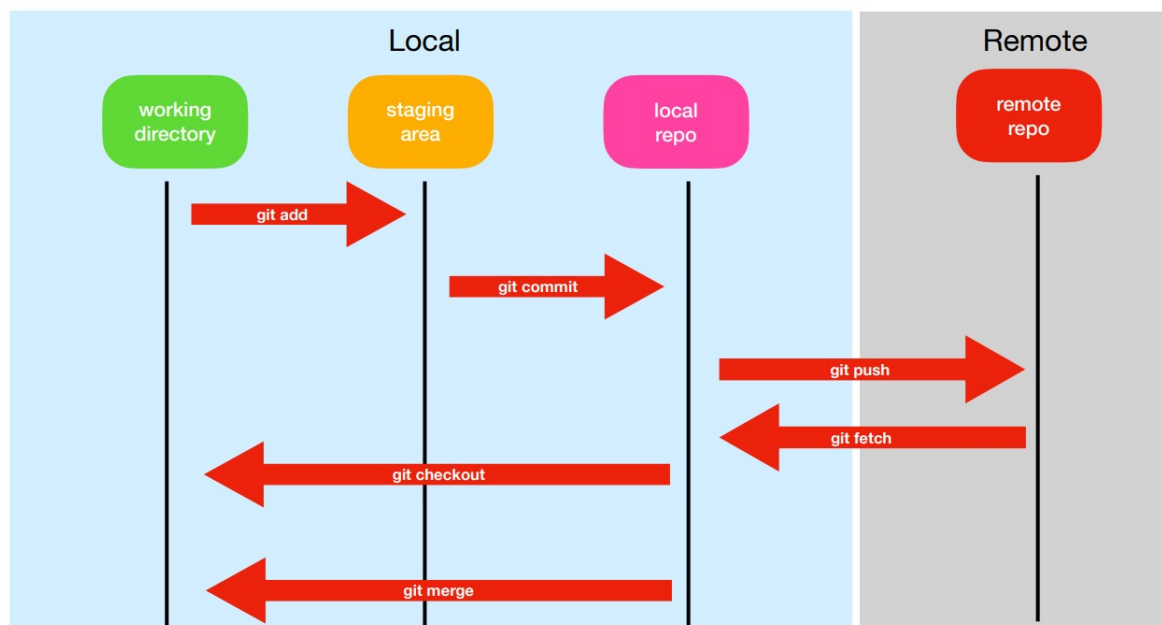


Abbildung 1: Darstellung des Git Workflows

Zweige dienen dazu, um das Projekt in verschiedene unabhängige Teile zu trennen, falls man zum Beispiel an einer Demo Version weiterschreiben möchte.

Nach einem erfolgreichen Push sind die Änderungen online auf dem Repository zu finden. Ein Repository ist der "Ordner", wo alle Dateien online gespeichert zu finden sind.

Ein Vorteil, den Git ebenfalls bietet ist das jeder Commit eine Version des Projekts ist, die man bis zum Zeitpunkt vom Commit herunterladen oder klonen kann. Jedes

Projekt kann privat oder öffentlich gemacht werden, sodass auch Personen, die keine Entwickler sind, darauf zugreifen können. [5]

### 4.2.3 Git Befehle

#### Clone

Es initialisiert ein Git Repository auf dem Rechner und ladet die zugehörigen Dateien runter. Wenn man es nicht spezifisch angibt, klonet es den Master Branch. Der Master Branch ist der Hauptzweig, eines jeden Git Projekts. Innerhalb des erstellten Ordners können alle weiteren Git Befehle ausgeführt werden. [6]

#### Commit

Ein Commit beschreibt Änderungen, die man im Projekt gemacht hat. Jeder Commit hat eine Bezeichnung, mit der der Entwickler die Änderungen, die er gemacht hat beschreiben kann. Zu jedem Commit gehören auch die Dateien die dabei geändert bzw. hinzugefügt wurden.

Es speichert den Zustand des gesamten Projekts bis zu dem Zeitpunkt und kann danach jederzeit abgerufen oder rückgängig gemacht werden. Diese Änderungen bleiben aber zunächst nur lokal auf dem Rechner. [6]

#### Push

Ein Push dient dazu, um die lokalen Änderungen (Commits) zu veröffentlichen. Es kopiert den aktuellen, lokalen Stand und speichert diesen auf das vom Internet erreichbare Repository.

Einem Push kann ein Zweig (Branch) zugeordnet werden um die Änderungen zuzuordnen. Ansonsten wird der Master Branch genommen. [6]

#### Pull

Der Pull Befehl kopiert die Inhalte vom öffentlichen Repository und fasst diese mit den lokalen Zustand auf dem rechner zusammen (merge). Es dient dazu die aktuelle Version des Projekts auf den Rechner herunterzuladen.

Falls es Konflikte zwischen der derzeitigen und neusten Version gibt, werden die Änderungen zusammengeführt. [6]

## Branch

Die Branch Befehle dienen dazu um eine neue Abzweigung des Projekts (Branch) zu erstellen.

Branches erstellt man, wenn man an einer neuen Version des Projekts arbeitet und diese vom Hauptteil trennen will. Meistens werden dadurch neue Funktionen programmiert, die später wieder in den Master Branch eingebunden werden.

Der Vorteil daran ist, dass man an neuen Funktionalitäten experimentieren kann, ohne den Hauptentwicklungsstand zu beeinflussen. Branches können jederzeit gelöscht oder wieder ins Hauptprojekt integriert werden. Es kann simultan am Master Branch und an Nebenzweigen gearbeitet werden durch trennen mit dem Push Befehl. [5]

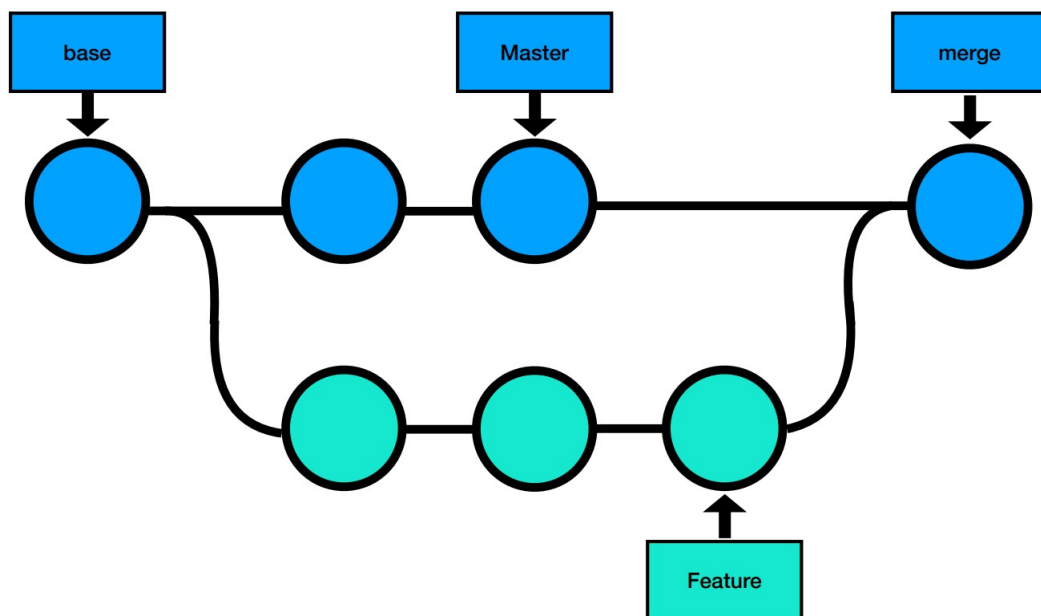


Abbildung 2: Darstellung von Branches in einem Repository

### 4.2.4 GitHub

GitHub ist ein gewinnorientiertes Unternehmen, dass einen auf Cloud basierten Git Repository Hosting-Service anbietet. Es wurde im Februar 2008 gestartet und von Chris Wanstrath, PJ Hyett, Scott Chacon und Tom Preston-Werner entwickelt. Es ist das beliebteste Tool um Softwareprojekte zu verwalten und wird von über 73 Millionen Entwicklern und über 4 Millionen Organisationen benutzt. Dazu ist es das größte und am meisten fortgeschrittene Entwicklungssystem, das es gibt. [7]

Es vereinfacht die Nutzung von Git für Teams und auch Einzelpersonen. Jeder kann sich einen GitHub Account erstellen und direkt loslegen und seine Arbeiten auf Reposi-

tories veröffentlichen. Es ist nicht nur zwingend für Code-basierte Projekte verwendbar sondern auch Websites erstellen und das Schreiben von Büchern ist möglich.

Was GitHub ausmacht ist die Benutzerfreundlichkeit und die Integration von Git. Außerdem bietet Github viele andere Funktionen wie zum Beispiel ein Projekt Board an, was es erleichtert innerhalb eines Teams, Probleme besser lösen zu können.

Es gibt ebenfalls bezahlte Pläne, die es vor allem Organisationen und Unternehmen leichter macht Unternehmensprojekte zu verwalten durch zusätzliche Funktionen. [4]

## **4.3 Java**

## **4.4 Java EE**

## **4.5 Maven**

### **4.5.1 Java EE vs. Quarkus**

### **4.5.2 JPA**

### **4.5.3 Hibernate**

### **4.5.4 Panache**

## **4.6 JBoss**

## **4.7 Cypress**

## **4.8 Keycloak**

## **4.9 Oracle Datenbank**

## **4.10 Vue.js**

Vue.js ist ein JavaScript-Webframework, das zum Erstellen von Single-Page-Webanwendungen dient. Es wurde von einem kleinen Team im Jahre 2014 entwickelt mit dem ursprünglichen Autor Evan You.

Vue ist relativ neu und die große Stärke von Vue ist die einfache Lernkurve, die Viel-

seitigkeit und die Leichtgewichtigkeit. Man benötigt Kenntnisse in JavaScript, HTML, CSS und schon kann man loslegen mit deren ausführlich dokumentierten Guide[8]. [9] [10]

### 4.10.1 Vue Funktionsweise

#### Model-View-Viewmodel (MVVM) Pattern

Vue.js benutzt das MVVM Pattern. Das Pattern trennt die Darstellung von der Logik der Benutzer-UI's. Dazu ist ein Datenbindungsmechanismus vorausgesetzt. Dadurch können sich Entwickler und Interfacedesigner trennen und ihre Aufgaben im Projekt aufteilen.

Dieses Pattern wurde 2005 von John Gossman veröffentlicht. [11]

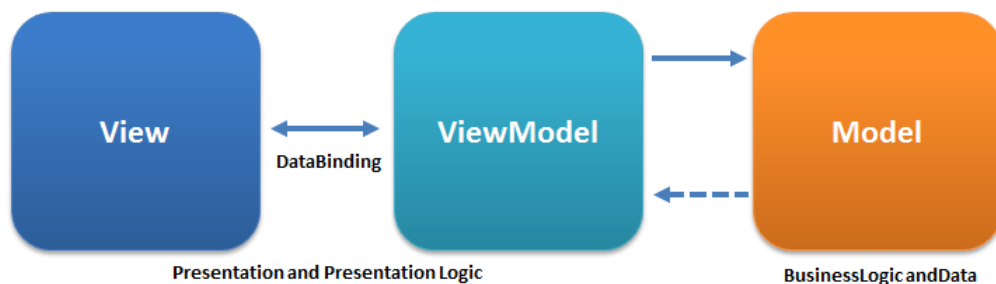


Abbildung 3: Darstellung von Branches in einem Repository  
<https://upload.wikimedia.org/wikipedia/commons/8/87/MVVMPattern.png>

- **View:** Enthält alle Elemente die durch die Benutzeroberfläche angezeigt werden. Es bindet sich an das ViewModel, welches die Eigenschaften der View bestimmt.
- **ViewModel:** Es enthält die Logik des UI's. Es tauscht sich mit dem Model aus und benützt seine Methoden und Dienste. Gleichzeitig gibt es der View Eigenschaften, die dem Model entsprechen. Es bindet Daten mit der View und sich selbst (DataBinding).
- **Model:** Diese Schicht enthält alle Daten die der Benutzer manipuliert oder aufruft. Es enthält die gesamte Geschäftslogik.[11]



## Vue Instance

Jede Vue Applikation beginnt mit der Erstellung einer Vue Instanz.

Listing 1: Vue Instanz

```
1   var vm = new Vue({
2     // options
3   })
```

Die Variable `vm` steht für ViewModel, was unsere Vue Instanz darstellt. Man kann jeder Instanz Optionen zuweisen, um sie zu konfigurieren. Diese Instanz wird auch als Root Instanz bezeichnet und bildet den Stamm eines Baumes mit Komponenten.

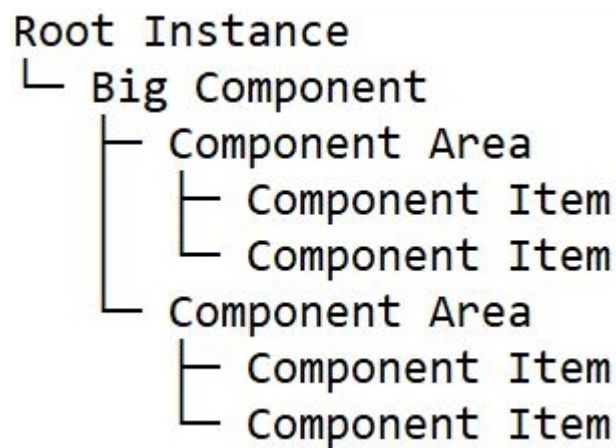


Abbildung 4: Der Stammbaum einer Root Instanz

Zu einer Instanz gehört auch der `data`-Bereich. Dieser beherbergt alle Properties einer Instanz und diese Properties reagieren auf Veränderungen im Code. Noch dazu kann jede Instanz Methoden haben.

Jede Instanz hat auch seine Lifecycle Hooks, dies sind Methoden, die zu bestimmten Zeitpunkten einer Instanz ausgeführt werden. [12] Diese sind:

- **created**
- **mounted**
- **updated**
- **destroyed**

## Lifecycle Diagram

Das Diagramm hier stellt den Ablauf einer Erstellung einer neuen Vue Instanz dar.

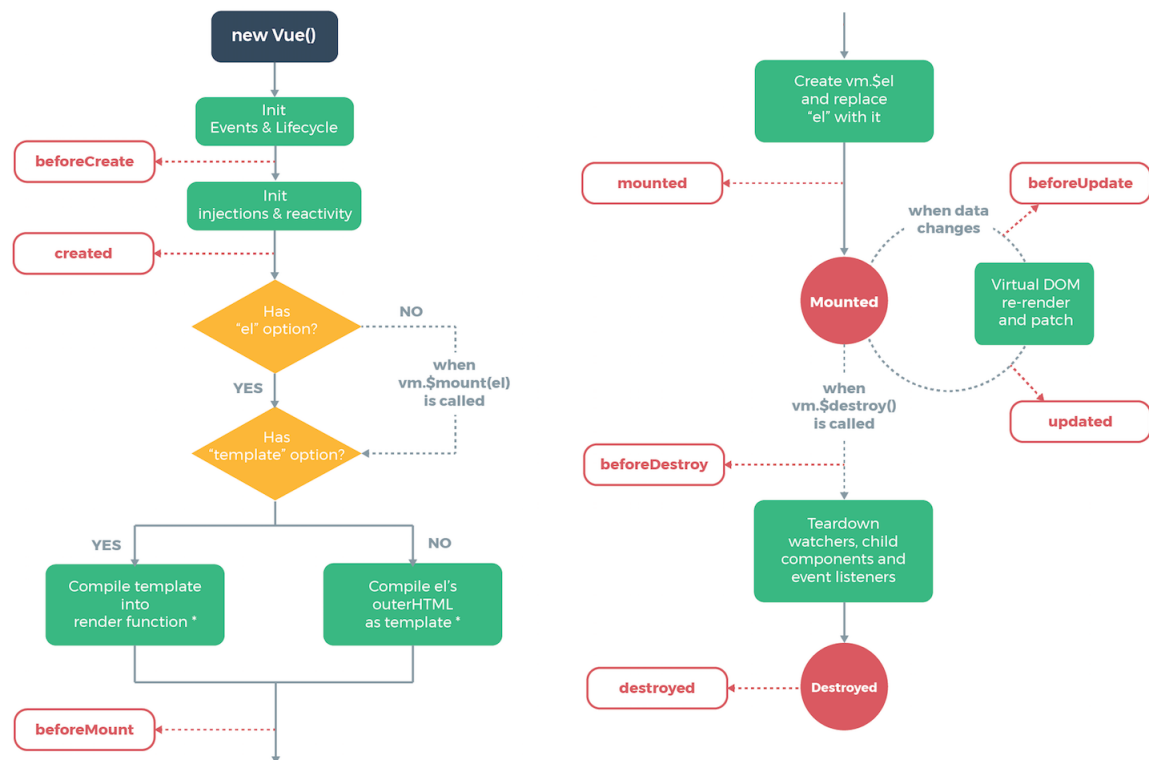


Abbildung 5: Diagramm des Ablaufs einer Vue Instanz

<https://www.oreilly.com/library/view/full-stack-vuejs-2/9781788299589/assets/9f308e86-bbbe-489c-9f93-06abe2675081.png>

## Vue Components

Komponenten sind wiederverwendbare Vue Instanzen mit einem eigenen Namen. Diese Komponenten können mittels HTML-Tags in anderen Komponenten verwendet werden.

## **4.11 Angular vs. Vue**

## **4.12 HTML**

## **4.13 CSS**

## **4.14 Javascript**

## **4.15 Json Web Token (JWT)**

## **4.16 Progressive Web App(PWA)**

## **4.17 Google Charts**

**Deepest**

Vermeide mich.

## 5 Implementierung

Siehe tolle Daten in Tab. 1.

Siehe und staune in Abb. 6. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

	Regular Customers	Random Customers
Age	20-40	>60
Education	university	high school

Tabelle 1: Ein paar tabellarische Daten

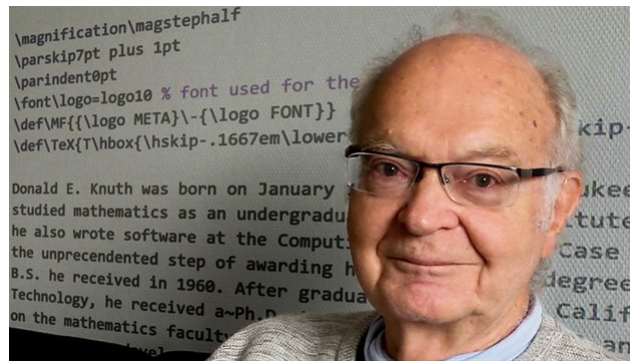


Abbildung 6: Don Knuth – CS Allfather

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus. Dann betrachte den Code in Listing 2.

## Listing 2: Some code

```

1  # Program to find the sum of all numbers stored in a list (the not-Pythonic-way)
2
3  # List of numbers
4  numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
5
6  # variable to store the sum
7  sum = 0
8
9  # iterate over the list
10 for val in numbers:
11     sum = sum+val
12
13 print("The sum is", sum)

```

# 6 Zusammenfassung

Aufzählungen:

- Itemize Level 1
  - Itemize Level 2
    - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
  - a. Enumerate Level 2
    - i. Enumerate Level 3 (vermeiden)

**Desc** Level 1

**Desc** Level 2 (vermeiden)

**Desc** Level 3 (vermeiden)



# Literaturverzeichnis

- [1] P. Rechenberg, G. Pomberger *et al.*, *Informatik Handbuch*, 4. Aufl. München – Wien: Hanser Verlag, 2006.
- [2] Association for Progressive Communications, „Wireless technology is irreplaceable for providing access in remote and scarcely populated regions,” 2006, letzter Zugriff am 23.05.2021. Online verfügbar: <http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access>
- [3] JetBrains, „IntelliJ IDEA,” letzter Zugriff am 5.11.2021. Online verfügbar: <https://www.jetbrains.com/de-de/idea/>
- [4] Kinsta, „Github erklärt,” letzter Zugriff am 5.11.2021. Online verfügbar: <https://kinsta.com/de/wissensdatenbank/was-ist-github/>
- [5] J. Strumpflohner, „Git Explained,” 4 2013, letzter Zugriff am 8.11.2021. Online verfügbar: <https://juristr.com/blog/2013/04/git-explained/>
- [6] StefanT123, „Git Commands Explained,” *dev.to*, 6 2020, letzter Zugriff am 8.11.2021. Online verfügbar: <https://dev.to/stefant123/basic-git-commands-explained-1cjd>
- [7] „GitHub Seite,” letzter Zugriff am 8.11.2021. Online verfügbar: <https://github.com/>
- [8] „Vue Guide,” letzter Zugriff am 10.11.2021. Online verfügbar: <https://vuejs.org/v2/guide/>
- [9] G. Roden, „Was man über Vue.js wissen sollte,” *heise.de*, 11 2020, letzter Zugriff am 10.11.2021. Online verfügbar: <https://www.heise.de/developer/artikel/Was-man-ueber-Vue-js-wissen-sollte-4969211.html>
- [10] „Vue.js,” letzter Zugriff am 10.11.2021. Online verfügbar: <https://de.wikipedia.org/wiki/Vue.js>
- [11] „Model-View-ViewModel,” letzter Zugriff am 10.11.2021. Online verfügbar: [https://de.wikipedia.org/wiki/Model\\_View\\_ViewModel](https://de.wikipedia.org/wiki/Model_View_ViewModel)
- [12] „Vue Guide Instance,” letzter Zugriff am 10.11.2021. Online verfügbar: <https://vuejs.org/v2/guide/instance.html>



# Abbildungsverzeichnis

1	Darstellung des Git Workflows . . . . .	6
2	Darstellung von Branches in einem Repository . . . . .	8
3	Darstellung von Branches in einem Repository . . . . .	10
4	Der Stammbaum einer Root Instanz . . . . .	11
5	Diagramm des Ablaufs einer Vue Instanz . . . . .	12
6	Don Knuth – CS Allfather . . . . .	15

# Tabellenverzeichnis

1	Ein paar tabellarische Daten . . . . .	14
---	--	----

# Anhang