



PROJEKTARBEIT T2000

des Studiengangs Informatik (B.Sc.)
an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Nils-Christopher Wiesenauer

9. September 2021

Bearbeitungszeitraum

15.12.2020 bis 09.09.2021

Matrikelnummer, Kurs

7344312, TINF19C

Ausbildungsunternehmen

evia solutions GmbH, Stuttgart

Betreuer

Gennadii Svitlychnyi

Gutachter der Dualen Hochschule

Prof. Dr. Mario Babilon

Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: „Konzeptionierung und Realisierung eines Bürgerportals mit intelligenter Suche“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Stuttgart, 9. September 2021

Ort, Datum

A handwritten signature in black ink, consisting of stylized, cursive letters, positioned above a horizontal line.

Unterschrift

Abstract

Das Bürgerportal bietet dem/der Nutzer*in die Möglichkeit, auf einfache Art und Weise Suchanfragen mit erweiterten Informationen zu senden. Mit Hilfe der integrierten künstlichen Intelligenz können neben der Beschreibung der Leistung auch mögliche weitere Zusatzfunktionen konfiguriert werden, z.B. die Art der Kontaktaufnahme, Bezahlungsmöglichkeiten oder die Begleitung und Nachverfolgung der Leistungserbringung. Die Anfragen werden sortiert und nach den Filtern, die der/die Nutzer*in eingestellt hat, zurückgegeben. Basierend auf diese Suchanfragen der Nutzer*innen lernt die künstliche Intelligenz mit, um zukünftig bessere und effizientere Ergebnisse liefern zu können. Das Projekt beinhaltet mehrere Aspekte:

- Frontend-Web-App zur Verwirklichung der Suche und der Ergebnisse
- Backend mit KI für eine effiziente und schnelle Suche

Inhaltsverzeichnis

Erklärung	I
Abstract	II
Inhaltsverzeichnis	III
Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	IX
Quellcodeverzeichnis	X
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Zielsetzung	1
1.3 Aufbau	2
2 Konzeptionierung des Bürgerportals	3
2.1 Anforderungen an das Projekt	3
2.2 Anfertigung Wireframes	4
2.3 Entwicklung Styleguide	6
2.4 Softwarearchitektur und Clean Code	7
2.5 Technologieevaluation Frontend-Framework	9
2.5.1 React	10
2.5.2 Angular	12
2.5.3 Vue	16
2.5.4 Entscheidung	18
3 Künstliche Intelligenz	19
3.1 Historischer Kontext	19
3.2 Nutzung KI in Deutschland	20
3.3 Definition künstliche Intelligenz	20
3.4 Teilgebiete	21

3.4.1	Wissensbasierte Systeme	21
3.4.2	Robotik	21
3.4.3	Musteranalyse und Mustererkennung	21
3.4.4	Mustervorhersage	21
3.4.5	Modellierung anhand der Entropiekraft	22
3.4.6	Künstliches Leben	22
3.5	Die starke und schwache KI-Hypothese	23
3.5.1	Schwache KI-Hypothese	23
3.5.2	Starke KI-Hypothese	23
3.5.3	Turing-Test	24
4	Umsetzung des Bürgerportals	27
4.1	Verwaltung mit Git	27
4.2	Kanban-Entwicklungsmethode	28
4.3	Coding Style Guideline	29
4.3.1	Nachgestelltes Komma	30
4.3.2	Tabulator Breite	30
4.3.3	Semikolon bei Anweisungsende	31
4.3.4	Einfache Anführungszeichen	31
4.3.5	Maximale Zeilenlänge	32
4.4	Darstellung des Bürgerportals	33
4.4.1	Architektur in Angular	33
4.4.2	Styling mit SCSS	35
4.4.3	Responsive Webdesign	36
4.4.4	Internationalisierung mit ngx-translate	37
4.4.5	Textdarstellung mit Markdown	39
4.4.6	Angular Material	41
4.5	Entwicklung einer REST API	42
4.5.1	Geplante Umsetzung	43
4.5.2	Erweiterung mit JSend	43
4.5.3	Postman	45
4.5.4	Anbindung an eine MongoDB Datenbank	46

4.5.5	Authentifizierungs-System	48
4.5.6	Konkrete Umsetzung	51
4.5.7	Brain.js	53
5	Corona Daten	59
6	Fazit und Ausblick	60
	Literaturverzeichnis	67
	Anhang	68
	Scribble Startseite	68
	Scribble Corona-Infoseite	69
	Wireframe Startseite	70
	Bürgerportal GUI - Startseite (1/2)	71
	Bürgerportal GUI - Startseite (2/2)	72
	Bürgerportal GUI - Mobile Ansicht	73
	Bürgerportal GUI - Corona Unterseite	74
	Bürgerportal GUI - Suche	75
	Bürgerportal GUI - Artikel	76
	Bürgerportal GUI - Register Form	77
	Bürgerportal GUI - Login Form	77
	Bürgerportal GUI - Dashboard	78
	Bürgerportal GUI - Dashboard: Account-Einstellungen	79
	Bürgerportal GUI - Dashboard: KI Training	80
	Bürgerportal GUI - Dashboard: Artikel Übersicht	81
	Bürgerportal GUI - Dashboard: Artikel Edit Dialog	82
	Bürgerportal GUI - Dashboard: Translate Dialog	83
	Bürgerportal GUI - Dashboard: Translate Dialog (markdown)	84
	Bürgerportal GUI - Dashboard: Bilder Galerie	85

Abkürzungsverzeichnis

AI	Artificial Intelligence
AL	Artificial Life
API	Application Programming Interface
CSS	Cascading Style Sheets
DI	Dependency Injection
DOM	Document Object Model
DVCS	Distributed Version Control System, verteiltes Versionskontrollsystem
ES	ECMAScript
GRU	Gated Recurrent Unit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IDE	Integrated Development Environment
JS	JavaScript
JSON	JavaScript Object Notation
JSX	JavaScript XML
JWS	JSON Web Signature
JWT	JSON Web Token
KI	künstliche Intelligenz
LSTM	Long Short Term Memory
MVC	Model View Controller

MVVM	Model View ViewModel
npm	Node Package Manager
NoSQL	not only SQL bzw. non-SQL
npm	Node Package Manager
REST	Representational State Transfer
RIA	Rich Internet Application
RNN	Recurrent Neural Network
RSA	Rivest–Shamir–Adleman
Sass	Syntactically Awesome Style Sheets
SCSS	Sassy CSS
SHA	Secure Hash Algorithm, Sicherer Hash-Algorithmus
SOLID	Single Responsibility Prinzip, Open-Closed-Prinzip, Liskovsches Substitutionsprinzip, Interface Segregation Prinzip, Dependency Inversion Prinzip
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locators
WiP	Work in Process

Abbildungsverzeichnis

1	Scribble - Startseite	4
2	Wireframe - Startseite	5
3	Bürgerportal Logo	6
4	MEAN-Stack	15
5	Kommunikation während Turing-Test	24
6	Erstelltes Kanbanboard für das Projekt T2000	28
7	Citizienportal UI - Startseite	36
8	Citizienportal API - Postman	45
9	Relationale Datenbank vs. Dokumentenbasierte Datenbank	46
10	Feedforward - Neuronales Netzwerk	54
11	Recurrent Neural Network: Rückkopplungsarten	56
12	Long Short Term Memory Neural Network	57
13	Gated Recurrent Unit Neural Network	58
14	Corona-Daten chart.js	59
15	Scribble der Startseite	68
16	Scribble der Corona-Infoseite	69
17	Wireframe Startseite	70
18	Bürgerportal GUI - Startseite (1/2)	71
19	Bürgerportal GUI - Startseite (2/2)	72
20	Bürgerportal GUI - Mobile Ansicht	73
21	Bürgerportal GUI - Corona Unterseite	74
22	Bürgerportal GUI - Suche	75
23	Bürgerportal GUI - Artikel	76
24	Bürgerportal GUI - Register Form	77
25	Bürgerportal GUI - Login Form	77
26	Bürgerportal GUI - Dashboard	78
27	Bürgerportal GUI - Dashboard: Account-Einstellungen	79
28	Bürgerportal GUI - Dashboard: KI Training	80

29	Bürgerportal GUI - Dashboard: Artikel Übersicht	81
30	Bürgerportal GUI - Dashboard: Artikel Edit Dialog	82
31	Bürgerportal GUI - Dashboard: Translate Dialog	83
32	Bürgerportal GUI - Dashboard: Translate Dialog (markdown)	84
33	Bürgerportal GUI - Dashboard: Bilder Galerie	85

Quellcodeverzeichnis

2.1	React.js - Input	10
2.2	React.js - Output	11
2.3	Angular - Komponente	14
2.4	Vue.js - Komponente	17
4.1	Prettier - Konfigurations Datei: prettierrc.json	29
4.2	Prettier - Option trailingComma	30
4.3	Prettier - Option tabWidth	30
4.4	Prettier - Option semi	31
4.5	Prettier - Option singleQuote	31
4.6	ngx-translate - Translate Direktive	37
4.7	ngx-translate - Translate Pipe	37
4.8	ngx-translate - i18n Beispiel-Datei (de.json)	38
4.9	ngx-translate - Translate Service	38
4.10	Markdown - Statisches Modell	40
4.11	Markdown - Remote-Datei (.md)	40
4.12	Markdown - Variable Binding	40
4.13	REST-API - Erfolgreiche Antwort als JSend-Objekt	44
4.14	JWT - Header	48
4.15	JWT - Payload	49
4.16	JWT - Signatur	49
4.17	REST-API - Training Routes	52
4.18	Brain.js - NeuralNetwork	55
4.19	Brain.js - LSTM Neuronales Netzwerk	58

1 Einleitung

1.1 Ausgangssituation

In der heutigen Zeit spielt die künstliche Intelligenz eine immer wichtiger werdende Rolle in der Informatik. In solchen Programmen wird versucht, menschliche Entscheidungsstrukturen nachzubilden, um so effizient und schnell das beste Resultat zu liefern. Gerade in den letzten Jahren, in denen das Corona Virus ein wichtiger Bestandteil ist, war es teils schwer für Bürger*innen des Landes schnell wichtige Informationen der Kommune zu erhalten.

Veraltete Webseiten und uneffiziente Suchfunktionen bieten leider nicht immer die Möglichkeit, die eigentlich gesuchte Information zu liefern. Falls ein*e Bürger*in sich in einer solchen Lage befindet, gibt es nur den Weg der telefonischen Kontaktaufnahme mit der Kommune selbst. Um diesen Fall vorzubeugen, hat die evia solutions GmbH mich, Nils-Christopher Wiesenauer, beauftragt, ein funktionstüchtiges Bürgerportal als Web-Applikation zu konzeptionieren und umzusetzen.

Eine weitere Maßnahme für eine effiziente Ergebnisausgabe stellt das automatisierte Trainieren von Suchbegriffen dar. Da die Kundschaft der evia solutions GmbH dieses Projekt als Auftrag erhalten hat, entfallen im Rahmen dieser Projektarbeit Erläuterungen, welche den Kunden betreffen.

1.2 Zielsetzung

Es gibt viele Möglichkeiten zur Umsetzung oder Integration einer künstlichen Intelligenz. In der Vergangenheit fing alles damit an, dass man das menschliche Denken automatisieren oder mechanisieren wollte. Man versuchte eine Illusion eines künstlichen Menschens zu vermitteln. Im Jahre 1956 fand die Veranstaltung "Dartmouth Conference" statt, welche als Gründungsstunde der künstlichen Intelligenz gilt. In einem sechswöchigen Workshop namens "Dartmouth Summer Research Project on Artificial Intelligence" erschien erstmals der Begriff Artificial Intelligence (AI). Seither kommen immer mehr Technologien auf den Markt, die eine künstliche Intelligenz in verschiedenen Bereichen ermöglichen.

Ziel dieser Arbeit ist somit, durch die Erstellung eines Bürgerportals mit integrierter künstlichen Intelligenz den Bürger*innen die bestmöglichen Informationen zu liefern und mit manuellen Suchen die künstliche Intelligenz weiter lernen zu lassen, damit zukünftig noch effizientere Resultate geliefert werden können.

Zu keinen oder falschen Suchergebnissen kommt es meistens, wenn der/die betroffene Nutzer*in einen Fehler in der Rechtschreibung oder aufgrund der technischen Gegebenheiten sich nicht im Portal auskennt.

Zu den Vorteilen und Zielen, welche mit dem künstlichen Bürgerportal verknüpft sind, zählt die einfache Bedienung für die Nutzer*innen, das automatisierte Trainieren durch ausgewählte Suchbegriffe, die Internationalisierung, durch das Hinzufügen von Übersetzungen der gewünschten Sprachen und der modulare Aufbau des Projektes zur einfachen Wartung und Weiterforschung des Programmes.

1.3 Aufbau

In der Einleitung wird die Bedeutung der Arbeit für die evia solutions GmbH sowie das damit verbundene Ziel verdeutlicht. Des weiteren klärt sich der inhaltliche Aufbau der vorliegenden Projektarbeit.

Aufgeteilt wird diese Ausarbeitung in einen theoretischen und praktischen Block. Der theoretische Teil ist notwendig, um die Zielsetzung der Arbeit zu erreichen, Methoden der Konzeptionierung und die praktische Umsetzungsmöglichkeiten aufzeigen zu können. Zuerst werden begriffliche Grundlagen geklärt. Anschließend werden in dem Kapitel 2.5 die Herangehensweise dargestellt, wie es zu den jeweiligen Technologien und Ideen gekommen ist.

In dem zweiten großen Block, der praktischen Umsetzung, wird in einem ersten Schritt erläutert, wie das zu entwickelnde Bürgerportal aussehen soll, damit es folgend effektiv genutzt werden kann. Im nächsten Schritt wird auf Grundlage der in der Theorie begründeten Technologien ein strukturiertes Vorgehen erstellt. Die einzelnen Komponenten und benötigten Dependencies werden definiert. Der praktische Block endet mit der Umsetzung des Bürgerportals und den dazugehörigen Darstellung. Das Fazit und der Ausblick bilden den Abschluss der Arbeit, welche neben einer Zusammenfassung der wichtigsten Ergebnisse auch Limitierungen der Arbeit und weitere Bearbeitungsaspekte aufzeigen. Im Anhang befinden sich zudem Ausschnitte des finalen Projektes.

2 Konzeptionierung des Bürgerportals

In diesem Kapitel wird näher auf die Konzeptionierung des Bürgerportals eingegangen, welche Ideen verfolgt wurden und wie sich der Weg bis zum fertigen Endprodukt gestaltet hat.

2.1 Anforderungen an das Projekt

Innerhalb dieses Abschnitts werden die Ziele des Vorhabens und die Eckdaten des Projekts beschrieben.

Das Projekt soll den Bürger*innen die Möglichkeit geben, auf einfache Art und Weise Suchanfragen mit erweiterten Informationen zu senden. Diese Suchanfragen sollen dann mit Hilfe einer integrierten künstlichen Intelligenz die möglichst besten Ergebnisse liefern. Das Ganze soll modular entwickelt werden. Entwickler*innen sollen in Zukunft an diesem Projekt weiterarbeiten können, auch wenn sie nur Erfahrung in einer der verwendeten Technologien besitzen.

Aufgrund der aktuellen Lage soll eine zusätzliche Unterseite für das Corona-Virus erstellt werden, in welcher die aktuelle Situation für Deutschland, Baden-Württemberg und dessen Land- und Stadtkreise gezeigt sind.

Die Internationalisierung ist ein weiterer Teil des Projektes. Das Bürgerportal soll für den/die Nutzer*in in den Sprachen deutsch und englisch verfügbar sein.

2.2 Anfertigung Wireframes

Wie es in verschiedenen Internetagenturen der Fall ist, wurden die ersten Scribbles in einem Kickoff mit dem Product Owner zu Beginn des Projektes gestaltet.

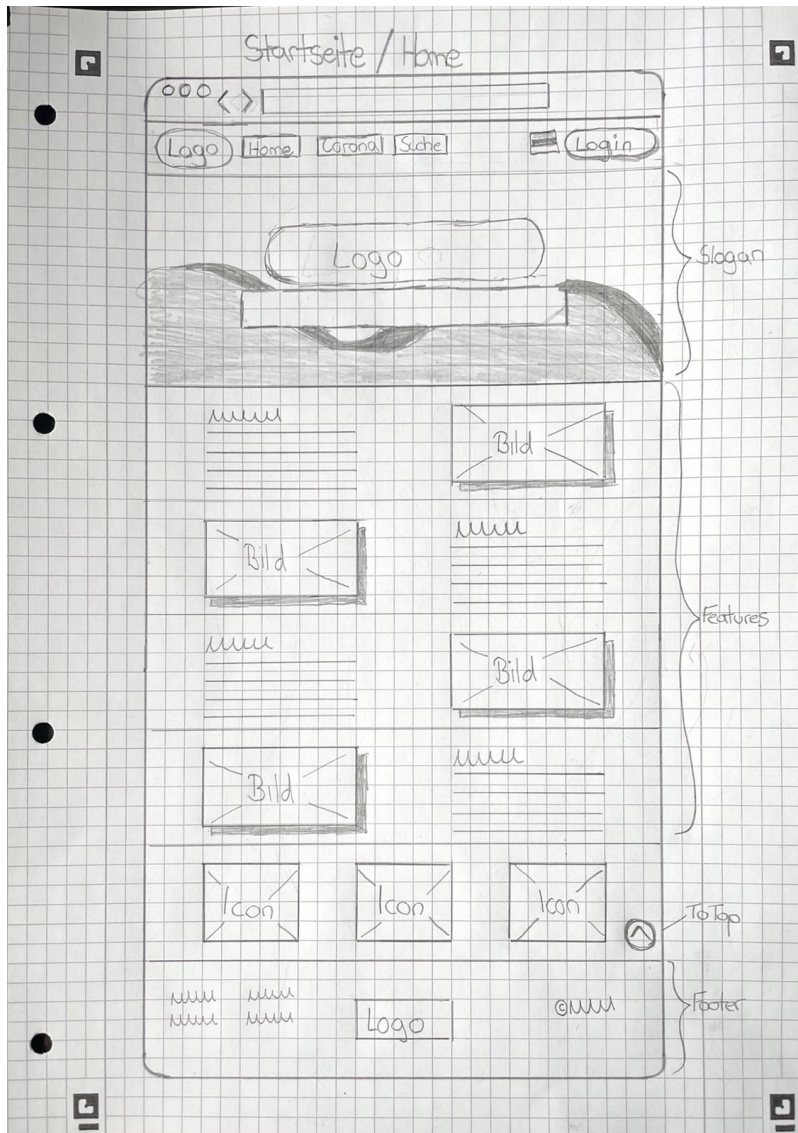


Abbildung 1: Scribble - Startseite

Quelle: Eigene Darstellung

Bei Scribbles handelt es sich um einen ersten Entwurf, der von Hand gezeichnet wird. Dies illustriert, wie das Bürgerportal aufgebaut und dargestellt werden könnte. Das Ziel von Scribbles ist es, die ersten Ideen und Vorschläge schnell und einfach festzuhalten und zu visualisieren. Man kann auch von ersten Skizzen der Webseite sprechen. [1] Im Beispiel wird die Startseite des Bürgerportals gezeigt. Alle erstellten Scribbles befinden sich im Anhang.

Mit den vorgefertigten Scribbles wurden dann im zweiten Schritt Wireframes für das spätere User Interface (UI) des Bürgerportals in draw.io umgesetzt. Ein Wireframe ist ein visueller Leitfaden, der das Grundgerüst einer Website darstellt [2].

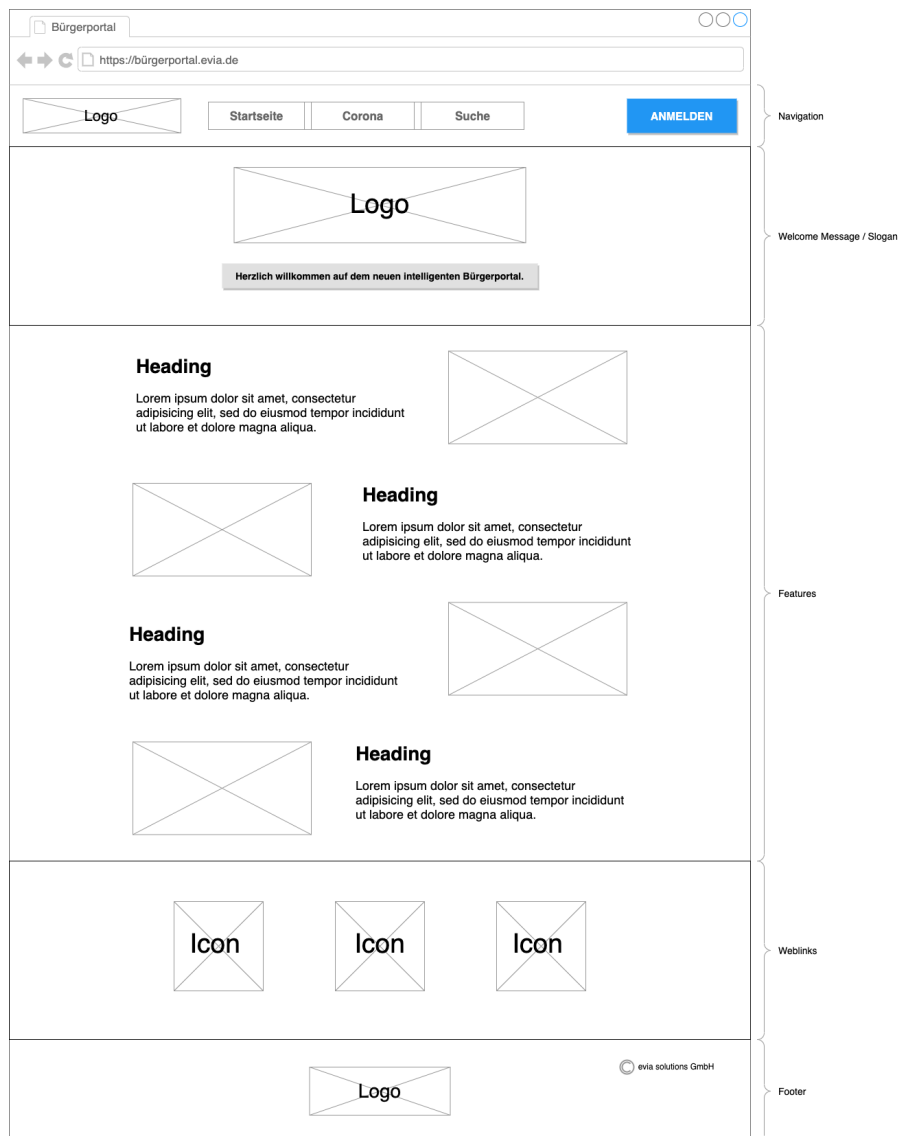


Abbildung 2: Wireframe - Startseite

Quelle: Eigene Darstellung

Wireframes werden erstellt, um Elemente so anzuordnen, dass sie einen bestimmten Zweck am besten erfüllen. Der Zweck wird normalerweise durch ein Geschäftsziel und eine kreative Idee bestimmt. Der Wireframe stellt das Seitenlayout oder die Anordnung des Inhalts der Website dar, einschließlich der Schnittstellenelemente und Navigationssysteme und wie sie zusammenarbeiten. [3]

Das Seitenlayout des Bürgerportals, zu sehen in Abbildung 2, wurde so gewählt, dass Bürger*innen zu Beginn der Webseite direkt an alle wichtigen Informationen gelangen.

2.3 Entwicklung Styleguide

Zur Verwirklichung des Bürgerportals entschied man sich einen Styleguide zu entwickeln.

Ein Styleguide ist ein Dokument, in dem festgelegt ist, wie die verschiedenen Gestaltungselemente des Projektes in der Öffentlichkeit verwendet werden sollen. Er umfasst unter anderem das Logo, die Farben und Schriften und legt fest, in welchem Zusammenhang und auf welche Weise sie zum Einsatz kommen. Durch den Einsatz eines Styleguides lassen sich Fehler und Frustrationen vermeiden, das Vertrauen in das System erhöhen, den Lernaufwand vermindern, die Produktivität und die Kundenzufriedenheit erhöhen und den Bekanntheitsgrad einer Marke oder eines Produktes steigern. Außerdem sorgt der Styleguide für ein einheitliches Auftreten. [4] Aufgrund der Wirkung der Farbe blau, die unter anderem für Ruhe, Vertrauen und Sehnsucht steht und als Lieblingsfarbe der Deutschen gilt (Vgl. [5]), wählte man blau als Primärfarbe der Farbpalette.



Abbildung 3: Bürgerportal Logo

Quelle: Eigene Darstellung

In Bezug darauf wurde das Logo, zu sehen in Abbildung 3, erstellt. Das Logo stellt den*die Bürger*in dar, der*die in Verbindung mit dem Buch-Icon Wissen mit Bezug auf ein Lexikon vermitteln soll. Zusätzlich wählte man die Font Roboto als Schriftart sowohl im Logo als auch in der Applikation selbst.

Roboto ist eine Schriftart der serifenlosen Linear-Antiqua Gruppe. Linear-Antiqua sind Satzschriften für das lateinische Alphabet bei denen die Buchstaben keine oder nur unwesentliche Strichstärkenunterschiede aufweisen. [6]

Bei Serifen handelt es sich um An- und Abstriche bzw. Anfangs- und Endstriche von Buchstaben. Weisen Buchstaben diese Striche auf, nennt man sie Serifenschrift, andernfalls Serifenlose.

2.4 Softwarearchitektur und Clean Code

Heutzutage existieren zur Entwicklung einer innovativen und selbsterklärenden Web-Applikation verschiedene Architekturen und Prinzipien. Als Ausgangspunkt für eine gute Softwarearchitektur haben sich die SOLID-Prinzipien des amerikanischen Softwareentwicklers und IT-Beraters Robert C. Martin bewährt. Diese Prinzipien sind: Single Responsibility Prinzip, Open-Closed-Prinzip, Liskovsches Substitutionsprinzip, Interface Segregation Prinzip, Dependency Inversion Prinzip (SOLID). Sie bilden den Kern einer ganzen Reihe von Prinzipien.

Single-Responsibility-Prinzip

Das Single-Responsibility-Prinzip besagt, dass jede Komponente nur eine einzige Verantwortung haben bzw. für eine Aufgabe verantwortlich sein und auch die gesamte Funktionalität einer Aufgabe abdecken soll.

“There should never be more than one reason for a class to change.”

Werden Änderungen an der Software vorgenommen, sollte es immer einen Grund dafür geben. In der Praxis hat eine Komponente dann die richtige Größe, wenn sich die Änderungen aufgrund einer User Story nur auf eine Komponente auswirken. Mehr als eine Verantwortung für eine Komponente führt zu mehreren Bereichen, in denen zukünftige Änderungen notwendig werden können. Die Wahrscheinlichkeit, dass die Komponente zu einem späteren Zeitpunkt geändert werden muss, steigt zusammen mit dem Risiko, sich bei solchen Änderungen subtile Fehler einzuhandeln. Dieses Prinzip führt in der Regel zu Komponenten mit hoher Kohäsion, in denen alle Methoden einen starken gemeinsamen Bezug haben. [7]

Open-Closed-Prinzip

Das Open-Closed-Prinzip lässt sich sowohl im kleinen sowie großen Rahmen anwenden. Erweiterungen an Klassen sollten nicht durch Änderung der Klasse durchgeführt, sondern über Vererbung oder besser durch Delegation gelöst werden.

“Modules should be both open (for extension) and closed (for modification).”

Bei größeren Softwareeinheiten ist ein Plugin-Konzept sinnvoll, über das zusätzliche Klassen zur Erweiterung der Funktionalität hinzugefügt werden können. [8]

Liskovsches Substitutionsprinzip

“Let $q(x)$ be a property provable about objects x of type T . Then $q(y)$ should be true for objects y of type S where S is a subtype of T .” [9]

Eine abgeleitete Klasse muss stets auch im Kontext ihrer Basisklasse eingesetzt werden können. Sie darf das Verhalten der Basisklasse nur erweitern, aber nicht einschränken. Der*die Entwickler*in darf bei Verwendung mit der Basisklasse oder einer abgeleiteten Klasse keine Unannehmlichkeiten hervorrufen. [10]

Interface-Segregation-Prinzip

Der Umfang eines Interfaces wird durch die Anforderungen des Client bestimmt und nicht umgekehrt. Ein Client darf nicht gezwungen werden, Funktionalität zu implementieren, die gar nicht benötigt wird:

“Clients should not be forced to depend upon interfaces that they do not use.” [11]

Damit wird der Zusammenhalt von Modulen gestärkt und deren Kopplung reduziert.

Dependency-Inversion-Prinzip

In einer geschichteten Architektur existieren Klassen und Module auf unterschiedlichen Abstraktionsebenen. In Klassen niedriger Ebenen werden Implementierungsdetails, bspw. die Datenbankansbindung, gekapselt. Auf den höheren Ebenen geht es dagegen um die abstrakte Business-Logik. [12]

“A. High-level modules should not depend on low level modules. Both should depend on abstractions. B. Abstractions should not depend upon details. Details should depend upon abstractions.” [13]

Das Prinzip der Dependency-Inversion besagt, dass Module auf höheren Abstraktionsebenen nicht von Modulen niedrigerer Abstraktionsebenen abhängen, sondern sich jeweils auf Interfaces beziehen sollten. Interfaces hingegen sollten nicht von Details abhängen, sondern die Details müssen eine Abhängigkeit von den jeweiligen Interfaces aufweisen.

2.5 Technologieevaluation Frontend-Framework

Es gibt eine Vielzahl an Programmiersprachen zur Darstellung im Benutzerbereich. Aufgrund der vorhandenen Expertise im Bereich JavaScript wurde das Bürgerportal in der evia solutions GmbH der Abteilung JavaScript (JS)-Delivery zugewiesen. Der entworfene Prototyp soll möglichst in allen Punkten in der Programmiersprache JavaScript umgesetzt werden.

JavaScript ist eine heutzutage nicht wegzudenkende Webtechnologie, die es erlaubt, Webseiten interaktiv zu gestalten. Die von Netscape entwickelte Skriptsprache wurde 1995 für dynamisches Hypertext Markup Language (HTML) in Webbrowsern entwickelt, um Benutzerinteraktionen auszuwerten, Inhalte zu ändern, nachzuladen oder zu generieren. Dadurch werden die Möglichkeiten von HTML und Cascading Style Sheets (CSS) erweitert. [14]

Obwohl JavaScript als Präfix den Namen der populärsten Programmiersprache Java enthält, hat sie nur im Geringsten mit ihr zu tun. JavaScript erstellt keine Applets oder Standalone-Anwendungen, ist in der allgemeinsten Form in HTML-Dokumenten integriert und bietet Interaktivitätsgrade für Webseiten, die mit einfachen HTML-Dokumenten nicht erzielt werden können. JavaScript ist die Sprache von Webbrowsern. Es verwendet das Application Programming Interface (API) des Browsers, das sogenannte Document Object Model (DOM). [15]

Die wichtigsten Unterschiede zwischen Java und JavaScript sind erstens, dass Java eine objektorientierte Programmiersprache ist, während JavaScript eine objektorientierte Skriptsprache ist. Im Vergleich muss der Code in Java kompiliert werden, JavaScript-Code wird im Text selbst interpretiert. Java erstellt Anwendungen, die auf einem virtuellen Rechner oder Browser ausgeführt werden, während JavaScript-Code nur auf einem Browser mit Hilfe des DOM ausgeführt wird.

Um die Entwicklung, die dazugehörige Modularisierung und das objektorientierte Arbeiten zu erleichtern, gibt es die Möglichkeit Frameworks beziehungsweise Bibliotheken zu nutzen. In der folgenden Sektion werden die drei bekanntesten JavaScript Frameworks angeschaut und diskutiert.

2.5.1 React

React.js ist eine Open-Source JavaScript-Bibliothek, die zum Erstellen von Benutzeroberflächen speziell für Single-Page-Anwendungen verwendet wird. Es wird für die Handhabung der Ansichtsebene für Web- und Mobile-Apps verwendet. React gilt als Nachfolger des von Jordan Walke und Jamie Kyle entwickelte Framework namens FaxJS (Vgl. [16]). Mit React kann man unter anderem wiederverwendbare UI-Komponenten erstellen. React wurde ursprünglich von Jordan Walke, einem Softwareingenieur, der für Facebook arbeitet, entwickelt. React wurde erstmals 2011 im Newsfeed von Facebook und 2012 auf Instagram eingesetzt [17].

```
1 <div id="myFirstReactApp"></div>
2 <script type="text/babel">
3   function HelloWorld(props) {
4     return <h1>{props.text}</h1>;
5   }
6   let App = <HelloWorld text="Hello World!" />;
7   ReactDOM.render(App, document.getElementById('myFirstReactApp'));
8 </script>
```

Quellcode 2.1: React.js - Input

Bei Quellcode 2.1 handelt es sich um ein rudimentäres Beispiel für die Verwendung von React in HTML mit JavaScript XML (JSX) und JavaScript. Die "HelloWorld" Funktion in Zeile 3 ist eine React-Komponente, die die Eigenschaft "text" akzeptiert. Die Variable "App" in Zeile 6 ist eine Instanz der "HelloWorld"-Komponente, in der die "text"-Eigenschaft auf "Hello World!" festgelegt wird. Die "ReactDOM.render"-Methode in Zeile 7 rendert dann unsere HelloWorld-Komponente innerhalb des DOM-Elements mit der ID "myFirstReactApp".

```
1 <div id="myFirstReactApp">
2   <h1>Hello World!</h1>
3 </div>
```

Quellcode 2.2: React.js - Output

Das erläuterte Beispiel wird dann wie in Quellcode 2.2 gezeigt, gerendert. Hierbei wird ein simpler HTML-Container mit der ID "myFirstReactApp" mit der Überschrift "Hello World!" erzeugt.

React ermöglicht es Entwickler*innen, große Webanwendungen zu erstellen, die Daten ändern können, ohne die Seite neu laden zu müssen. Der Hauptzweck von React besteht darin, schnell, skalierbar und einfach zu sein. Es funktioniert nur auf Benutzeroberflächen in der Anwendung. Dies entspricht der Ansicht in der Model View Controller (MVC)-Vorlage. Es kann mit einer Kombination anderer JavaScript-Bibliotheken oder Frameworks wie Angular in MVC verwendet werden. [18]

2.5.2 Angular

Angular ist ein TypeScript basierendes kostenloses Open-Source Frontend-Framework, das vom Angular-Team bei Google und der Community von Einzelpersonen und Unternehmen als komplette Neufassung von AngularJS entwickelt wurde. [19]

Die neueste zurzeit veröffentlichte Version "v12.2.1" von Angular wurde auf GitHub am 11.08.2021 durch Andrew Kushnir getaggt. Mit dieser Version wurden Fehler des Router behoben. Es wird nun sichergestellt, dass die Prüfung auf Übereinstimmung der Optionen mit der Umbenennung von Eigenschaften kompatibel ist. [20]

Mit Angular steht ein Framework zur Auswahl, das sehr gut an die Bedürfnisse der Entwickler*innen komplexerer Single-Page-Applications zugeschnitten ist. Dependency Injection, Annotationsbasierte Konfiguration, ein Komponenten- und Modulsystem, sowie gute Unterstützung für Unit- und Integrationstests sorgen dafür, dass wenige Abstriche gemacht werden müssen. Da Angular in erster Linie für Single-Page-Applications entwickelt wurde, ist es möglich Anwendungen zu entwickeln, die sich wie Rich-Clients bedienen lassen, trotzdem aber die Vorteile von Web-Clients mitbringen. Keine Installation ist notwendig, automatische Updates können veröffentlicht werden und das Ganze erfolgt plattformunabhängig. [21]

Angular wurde mit Blick auf moderne Webstandards entwickelt und bietet volle Flexibilität bei der Auswahl der Sprache und bietet volle Unterstützung für ECMAScript (ES)6 und TypeScript. [22]

Typescript ist eine von Microsoft Coporation entwickelte Open-Source-Sprache. TypeScript steht in einer ungewöhnlichen Beziehung zu JavaScript. Es bietet alle Funktionen von JavaScript und darüber hinaus noch eine weitere Ebene: das Typsystem von TypeScript selbst. Das bedeutet, dass vorhandener funktionierender JavaScript-Code auch TypeScript-Code ist. Deshalb gilt Typescript als Superset von JavaScript. Typescript bringt im Gegensatz zu JavaScript, wie der Name schon vermuten lässt, eine statische Typisierung mit. Um Interoperabilität, also die Zusammenarbeit unterschiedlicher Systeme, mit ungetyptem JavaScript Code zu ermöglichen sind diese Typisierungen optional. [23]

Der Hauptvorteil von TypeScript besteht darin, dass es unerwartetes Verhalten in Code hervorheben kann, wodurch die Wahrscheinlichkeit von Fehlern verringert

wird. JavaScript stellt beispielsweise Sprachprimitive wie "string" und "number" bereit, überprüft jedoch nicht, ob diese konsistent zugewiesen wurden. TypeScript wiederum tut es. [24]

Typen bieten eine Möglichkeit, die Form eines Objektes zu beschreiben, eine bessere Dokumentation bereitzustellen und es TypeScript zu ermöglichen, zu überprüfen, ob der erstellte Code richtig funktioniert. [23]

So ist es der Integrated Development Environment (IDE) möglich, den Code vor dem Compilen nach Fehler der verwendeten Typen zu überprüfen. Die Typisierung muss somit von den Entwickler*innen eingehalten werden und erleichtert das Programmieren enorm. Für viele JavaScript-Bibliotheken existieren Typdefinitionen für Typescript, die die möglichen Typen der JavaScript APIs einschränken und so besser mit Typescript Code benutzbar machen. Da Browser kein Typescript unterstützen, wird der Quellcode vor der Ausführung im Browser zu JavaScript compiliert. Übernommen wird dies von der Angular Tool-Chain. So steht dem*der Entwickler*in eine Sprache mit starkem Typsystem wie aus Java zur Verfügung, die zudem alle aktuellen Sprachfeatures von ECMAScript 2017 (ES8) einschließt, ohne auf Kompatibilität mit Browsern Rücksicht nehmen zu müssen. Die starke Typisierung ermöglicht das Entdecken von Fehlern zur Compilezeit, die bessere Code Analyse durch Tools und somit viele aus dem Backend bekannte Komfortfunktionen wie Autovervollständigung, navigierbare Methoden, Klassen- und Refactoring-Unterstützung. (Vgl. [25])

```
1 import { Component, Input, Output, EventEmitter } from '@angular/core'
2 @Component({
3   selector: 'app-user-list',
4   template: `
5     <ul>
6       <li *ngFor="let user of users" (click)="onSelectUser(user.id)">
7         {{ user.name }}
8       </li>
9     </ul>
10  `,
11 })
12 export class UserListComponent {
13   @Output() selectUser = new EventEmitter<{ id: string; name: string }>();
14   @Input() users: { id: string; name: string }[];
15   onSelectUser(id: string) {
16     this.selectUser.emit(this.users.find(u => u.id === id));
17   }
18 }
```

Quellcode 2.3: Angular - Komponente

In Quellcode 2.3 wird eine ziemlich triviale Komponente namens "UserListComponent" gezeigt, die die Eigenschaftsbindung (@Input()) verwendet, um Daten bereitzustellen, die von außen festgelegt werden sollen, die Ereignisbindung (@Output()), um Ereignisse auszugeben und eine Direktive (ngFor), um eine Liste von Daten zu rendern.

Es ist in Zeile 2 zu sehen, dass Angular eine TypeScript-Funktion namens Decorators (@Component) verwendet, um zusätzliche Daten an normale Klassen anzuhängen. Man schreibt Code wie diesen und Angular kümmert sich hinter den Kulissen darum, das echte DOM zu manipulieren und zu verbinden. Als Entwickler*in muss man nie Code schreiben, der direkt Elemente zum und vom DOM erstellt oder entfernt. Man erstellt stattdessen Angular-Komponenten mit der oben gezeigten Syntax, damit Angular den Rest übernimmt. Man kann Eingaben (Eigenschaften) und Ausgaben (Ereignisse) von Komponenten definieren und mithilfe von Angular einen komponentenspezifischen oder app-weiten Status verwalten.

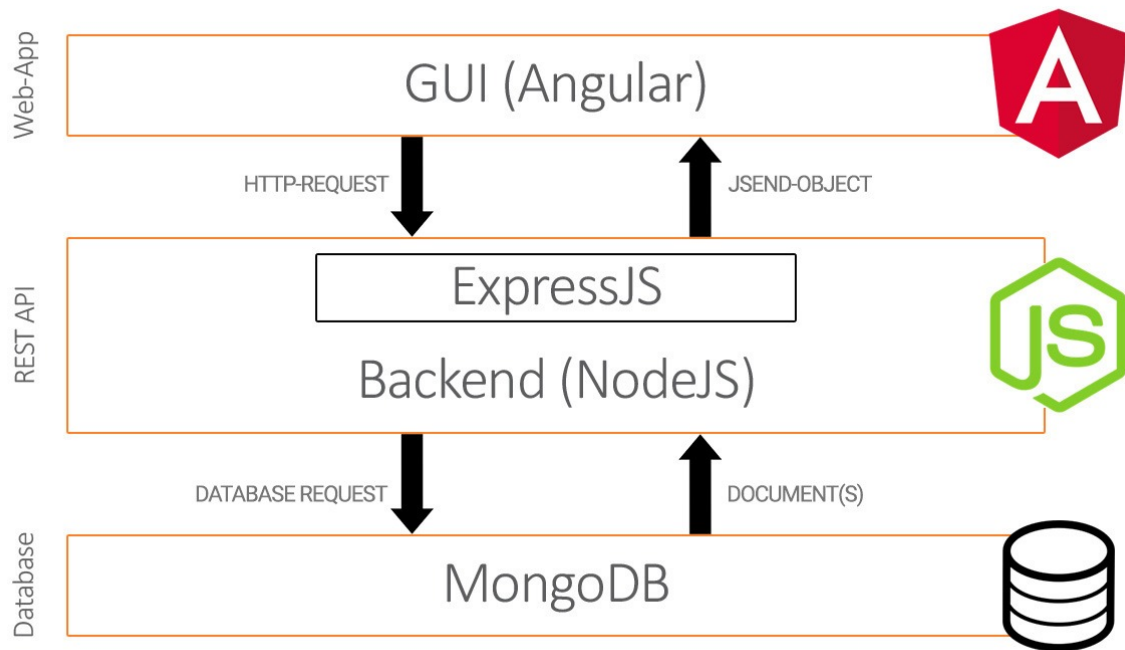


Abbildung 4: MEAN-Stack

Quelle: Eigene Darstellung

Angular wird sehr oft als Frontend des sogenannten MEAN-Stacks verwendet, der aus der MongoDB-Datenbank, dem Express.js-Webanwendungsserver-Framework, Angular selbst und der Node.js-Serverlaufzeitumgebung besteht. Wie in Abbildung 2 gezeigt, sendet die GUI eine Hypertext Transfer Protocol (HTTP)-Anfrage an die Representational State Transfer (REST)-API. Diese validiert die Anfrage und sendet ggfs. eine Datenbankabfrage an die verbundene MongoDB. Sofern die Anfrage erfolgreich ist, wird das gefundene Dokument an das Backend zurückgeliefert. Hier wird das Ergebnis dann als JavaScript Object Notation (JSON)-Objekt zurück an das Frontend geliefert. Im Frontend erfasst, wird das Ergebnis dann je nach Anwendung verarbeitet und visualisiert.

2.5.3 Vue

Vue ist ein progressives Framework zum Erstellen von Benutzeroberflächen. Das von Evan You entwickelte Framework geht nach dem Model View ViewModel (MVVM)-Muster, das als Entwurfsmuster zur Trennung der Darstellung und Logik entwickelt wurde, vor. Im Gegensatz zu anderen monolithischen Frameworks ist Vue von Grund auf so konzipiert, dass es inkrementell adaptierbar ist.

Die Kernbibliothek konzentriert sich nur auf die Ansichtsebene und ist leicht zu übernehmen und in andere Bibliotheken oder vorhandene Projekte zu integrieren. Auf der anderen Seite ist Vue auch perfekt in der Lage, anspruchsvolle Single-Page-Anwendungen zu betreiben, wenn es in Kombination mit modernen Tools und unterstützenden Bibliotheken verwendet wird. [26]

Wie auch React benutzt Vue ein virtuelles DOM, stellt reaktive und zusammensetzbare View Components bereit und behält den Schwerpunkt in der Kernbibliothek, wobei Dinge wie Routing und globale Zustandsverwaltung von Begleitbibliotheken behandelt werden. [27]

Zur Verwendung von Vue ist keinespezielle Einrichtung erforderlich. Man kann ganz einfach beginnen, indem man die Vue-Bibliothek in eine HTML-Datei importiert. Dadurch ist kein spezielles Projekt-Setup oder eine Kompilierung hinter den Kulissen wie bei Angular und React erforderlich. Vue verwendet nur JavaScript und HTML und allgemein auch keine große spezielle Syntax, die man zuerst lernen muss. Man erweitert das vorhandene HTML mit Direktiven wie v-for. Diese Verbesserungen sind für erfahrene Entwickler*innen zum größten Teil selbsterklärend. [28]

```
1 <template>
2   <ul>
3     <li v-for="user in users" :key="user.id"
4       @click="selectUser(user.id)">
5       {{ user.name }}
6     </li>
7   </ul>
8 </template>
9 <script>
10  export default {
11    props: ['users'],
12    methods: {
13      selectUser: function(userId) {
14        this.$emit('selectUser', this.users.find(u => u.id === userId))
15      },
16    },
17  }
18 </script>
19 <style>...</style>
```

Quellcode 2.4: Vue.js - Komponente

Das Vue Snippet 2.4 ist ein gleichnamiges Code Snippet zum Rendern einer Liste der Nutzer*innen, wie das des Angular Code Snippets 2.3. Wenn man eine Komponente in einer .vue-Datei auf der einfachsten Ebene erstellt, findet man drei Abschnitte, in die man Code einfügen kann. Der template-Abschnitt (Zeile 1 - 8) fungiert als erweiterte Version von HTML-Code und dient als Richtlinie für das Framework, wie das endgültige Markup der Komponente auf der Grundlage des internen Zustands zu erstellen ist. Ähnlich wie in Angular kommt hier eine Direktive (v-for, Zeile 3) zum Einsatz, um alle Nutzer*innen der Liste "users" zu listen.

Im zweiten Abschnitt namens "script" (Zeile 9 - 18) befindet sich die Logik der Anwendung. In diesem Abschnitt werden Low-Level-Konzepte wie beispielsweise Eigenschaften eingegeben und Funktionen definiert oder importiert.

Im letzten Abschnitt "style" wird CSS zur optionalen Gestaltung und Strukturierung des Komponenten ähnlich wie im allgemeinen HTML Head-Tag hinzugefügt.

2.5.4 Entscheidung

Nach eingängiger Auseinandersetzung mit den drei Technologien React, Angular und Vue ergibt sich für die finale Technologieentscheidung folgende Bewertung:

Auswahlkriterien anhand der angeschauten Technologien			
	React	Angular	Vue
Veröffentlichung	2013	2010	2014
Aktuelle Version	17.x	11	3.0.x
K1: Popularität (GitHub)	★★★(173Tsd.)[29]	★★(75.8Tsd.)[30]	★★★(187Tsd.)[31]
K2: Größe und Ladezeit	★★★	★★	★★★
K3: Lernaspekt	★★	★★★	★
K4: Komponenten	★★★	★★★	★★★
K5: Form-Validierung	★	★★★	★
K6: HTTP-Client	★	★★★	★
Durchschnitt	2.0★	2.5★	2.0★

Nach ausgiebiger Diskussion, welche Technologie respektive welches Framework für die bevorstehende Implementation des Bürgerportals die am geeignetste sein könnte, wurden die Auswahlkriterien näher betrachtet.

Vergleicht man die Kandidaten anhand der Auswahlkriterien, sehen die Bewertungen meist ziemlich ausgeglichen aus. Die Popularität, gemessen an der Anzahl der vergebenen Sterne auf GitHub, ist bei allen Frameworks sehr hoch [29, 30, 31]. Die Betrachtung der Kriterien K4, K5 und K6 lässt jedoch im Bezug auf die grundlegende Idee dieser Arbeit, der Entwicklung eines Bürgerportals, bereits eine ziemlich gute Filterung zu. So wird klar, dass Angular eher in Frage kommt.

Die Betrachtung der verbleibenden Kandidaten verdeutlicht, dass React und Vue keines Wegs keine Alternative darstellen. Alle angeschauten Frameworks bieten die Möglichkeit zur Implementierung des Vorhabens, jedoch besitzt Angular mehr direkt eingebaute Features, die dabei helfen, das Bürgerportal effizient umzusetzen. Zusätzlich war die Neugier auf eine unverbrauchte Technologie, die Verwendung des MEAN-Stacks und die Herausforderung, etwas neues auszuprobieren sehr groß.

Aus diesen Gründen geht Angular mit dem bereits vorhandenen Routing, HTTP-Client und Form-Validierung als Gewinner der Technologieevaluation hervor.

3 Künstliche Intelligenz

In diesem Kapitel wird der Begriff der künstlichen Intelligenz genau erörtert. Um im weiteren Vorgehen die Implementierung der Suche mit künstlicher Intelligenz aufzeigen zu können, ist es unvermeidlich, eine konkrete Vorstellung zu schaffen, was unter einer künstlicher Intelligenz verstanden werden kann.

3.1 Historischer Kontext

Zu den bedeutensten Gründer*innen der künstlichen Intelligenz gilt der Mathematiker Alan Turing. Im Jahre 1936 kam Turing zum Ergebnis, dass in der Theorie eine Rechenmaschine möglich sein müsse, in der Lage zu sein, alle Probleme zu lösen. In Bezug auf eine Intelligenz bedeutet dies, dass künstliche Intelligenz dann möglich ist, wenn intelligentes Denken als mathematische Probleme beschrieben werden kann. Herr Turing ging noch weiter in seinen Überlegungen und beschäftigte sich zudem mit der Frage, wie man bei Vorhandensein einer entsprechenden künstlichen Intelligenz beweisen könnte, ob diese auf dem Niveau der menschlichen Intelligenz operiert. Hierzu entwarf er den Turing-Test“. Dieser gilt als bestanden, wenn eine Versuchsperson mit einem Menschen und einer Maschine kommuniziert ohne diese zu sehen und nicht feststellen kann, welches der beiden die menschliche und welches die künstliche Intelligenz ist. [32]

Ungeachtet, dass das Thema Intelligenz die Menschheit schon seit Jahrtausenden beschäftigte und auch bereits im antiken Griechenland Philosophen sich damit auseinander gesetzt haben, wurde der Begriff der künstlichen Intelligenz erst durch den Namen einer von John McCarty organisierten Forschungskonferenz im Jahr 1956 geprägt und popularisiert. Aufgrund der damaligen Gegebenheiten ging die Entwicklung der künstlichen Intelligenz nur schleppend voran.

Erst seit Mitte der 1970er Jahre begann man damit, reale Einsatzmöglichkeiten von künstlicher Intelligenz zu finden, wobei man seitdem mit immer steigenden technischen Kapazitäten und Investitionen immer größere Erfolge erzielen konnte. In der heutigen Informatik investieren Firmen gewaltige Summen für die Entwicklung von künstlicher Intelligenz. [33]

3.2 Nutzung KI in Deutschland

Zwar ist die Zahl mit 6 Prozent der Unternehmen, die künstliche Intelligenz (KI) zum Einsatz bringen noch relativ gering, dennoch gaben 17 Prozent (Vgl. [34]) der Betriebe an, KI zu testen oder zumindest Projekte mit KI-Einsatz zu planen.

Laut Google Trends, einem von Google entwickelten Programm zur Analyse von Suchbegriffen im Bereich der Popularität, verzeichnet das Interesse für den Suchbegriff "künstliche Intelligenz" seit August 2016 einen großen Anstieg. [35]

3.3 Definition künstliche Intelligenz

Es gibt keine einheitlich anerkannte Definition der künstlichen Intelligenz. Vielmehr gibt es unterschiedliche Sichtweisen, was eine künstliche Intelligenz ist. Für die eindeutige Definition der künstlichen Intelligenz fehlt es an Wissen, wie die Intelligenz selbst zu beschreiben ist. Aufgrund dessen wird in dieser Arbeit auf charakteristische Merkmale der künstlichen Intelligenz gesetzt. Hierzu wird zuerst gezeigt, welche verschiedene wissenschaftlichen Stoßrichtungen insbesondere welche Teilgebiete es zu diesem Thema gibt. Im Anschluss daran wird eine Gegenüberstellung von schwachen und starken künstlicher Intelligenzen näher ausgeführt. Schließlich werden anfängliche Ideen erläutert und mit dem bekannten Turing-Test abgeschlossen.

3.4 Teilgebiete

In der Allgemeinheit wird die künstliche Intelligenz in ihren Anwendungsfällen in verschiedene Teilgebiete untergeordnet:

3.4.1 Wissensbasierte Systeme

Dieses Teilgebiet ist bei der Entwicklung des Bürgerportals von großer Bedeutung. Hier wird eine Form von rationaler Intelligenz modelliert, damit aus den Suchanfragen der Nutzer*innen auf Grundlage formalisierten Fachwissens und daraus gezogener logischer Schlüsse Antworten geliefert werden können. [36]

Im Beispiel des zu entwickelten Bürgerportals, soll es anhand von Suchbegriffen die jeweilige passende Kategorie finden. Die angegebenen Suchbegriffe sind hierbei der Input und die resultierende Kategorie der Output. Diese sollen im Dashboard des Bürgerportals manuell hinzugefügt werden können.

3.4.2 Robotik

Die Robotik beschäftigt sich mit manipulativer Intelligenz. Anwendung findet dieses Teilgebiet unter anderem bei der Minensuche oder beim Schweißen und Lackieren bei immer gleich auftretenden Manipulationen. Man versucht hier die intelligenten Verhaltensweisen von Lebewesen nachzuvollziehen. [37]

3.4.3 Musteranalyse und Mustererkennung

Hier geht es darum, Formen beziehungsweise Bilder zu erkennen und zu analysieren. Bekannte Anwendungsbeispiele sind Handschrifterkennung, Gesichtserkennung, Abgleich von Fingerabdrücken, zur industriellen Qualitätskontrolle oder zur Fertigungsautomation. [38]

3.4.4 Mustervorhersage

Hierbei handelt es sich um eine Erweiterung der Mustererkennung. Sie stellt die Grundlage der Abbildung einiger Eigenschaften des Neocortex dar. „Vorhersage ist nicht einfach nur eines der Dinge, die dein Gehirn tut. Sie ist die Hauptfunktion des Neocortex und das Fundament der Intelligenz.“ [39] Vorteile dieses Systems ist die Vorhersage, wohin sich das erkannte Objekt bewegt beziehungsweise als nächstes aufhält.

3.4.5 Modellierung anhand der Entropiekraft

Basierend auf der Arbeit des Physikers Alexander Wissner-Gross lassen sich intelligente Systeme mit Hilfe von Entropiekräften modellieren. Der intelligente Agent versucht, seine Umgebung (Zustand X_0) durch Aktion (Kraftfeld F) zu beeinflussen, um im zukünftigen Zustand X die größtmögliche Handlungsfreiheit (Entropie S) zu erreichen. [40]

3.4.6 Künstliches Leben

Artificial Life (AL) überschneidet sich mit der künstlichen Intelligenz. Daher wird diese als Teildisziplin davon und umgekehrt als übergeordnete Disziplin angesehen. [41]

3.5 Die starke und schwache KI-Hypothese

Im Rahmen der künstlichen Intelligenz unterscheidet die Philosophie die schwache und starke KI-Hypothese.

3.5.1 Schwache KI-Hypothese

Bei der schwachen KI-Hypothese handelt es sich um eine Behauptung, dass Maschinen genau so agieren können, als ob diese intelligent seien. Von den meisten wird die schwache KI-Hypothese als gegeben hingenommen. Konkret geht es darum, dass konkrete Anwendungsprobleme gemeistert werden. Sie soll eine Unterstützung zwischen dem menschlichen Denken und den technischen Anwendungen bieten. [42]

3.5.2 Starke KI-Hypothese

Die starke KI-Hypothese geht im Gegensatz davon aus, dass Maschinen wirklich denken können. Hierbei geht man davon aus, dass sie nicht nur das Denken simulieren. Die Hypothese einer starken künstlichen Intelligenz steht meist nicht im Vordergrund der künstlichen Intelligenzforschung, sondern ist eher eine pragmatische Sichtweise.

Eine starke KI muss nicht viele Gemeinsamkeiten mit dem Menschen haben. Sie wir ein anderen Aufbau für das kognitive Denken haben und in den Anfängen ebenfalls nicht mit dem des menschlichen Denkens auf gleicher Ebene sein. [43]

Als Forscher*in der künstlichen Intelligenz sollte man sich jedoch der ethischen und moralischen Konsequenzen bewusst sein.

3.5.3 Turing-Test

Ein Test, der die starke KI-Hypothese nachweisen könnte, könnte der folgende von Alan Turing vorgeschlagene Test sein, welcher bereits im Kapitel 3.1 Historischer Kontext angeschnitten wurde.

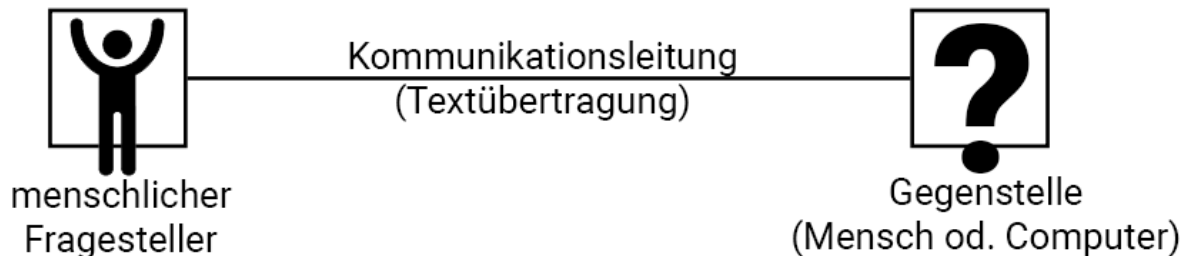


Abbildung 5: Kommunikation während Turing-Test

Quelle: Eigene Darstellung

Beim sogenannten Turing-Test stellt ein*e menschliche*r Fragesteller*in einem Computer eine schriftliche Frage, welcher darauf antwortet (visualisiert in Abbildung 5). Kann der*die Fragesteller*in nicht erkennen, ob die Antwort von einem Computer oder einer Person stammt, gilt der Test als bestanden, d.h. die Gegenstelle sollte abwechselnd den Computer und dann die Person widerspiegeln. [44]

Das große Problem dieses Tests ist, dass er subjektiv ist. Er ist nämlich mit der Denkweise des Menschen verknüpft, insbesondere vom Wissen über die aktuellen Fähigkeiten des Computers. Das Ergebnis des Tests kann durch mehrfaches Wiederholen des Experiments mit mehreren Fragesteller*innen erhöht werden.

Dieser Test ist so entwickelt, dass keine physische Interaktion zwischen Mensch und Computer gibt, da diese Erweiterung keine Anforderung einer Intelligenz ausmacht. In diesem Fall sieht der Mensch den Computer nicht und die Fragen werden schriftlich gestellt. Im Vergleich hierzu verwendet der totale Turing-Test ein zusätzliches Videosignal, damit der Computer auch den*die menschliche*n Fragesteller*in analysieren kann, z.B. durch Gestenerkennung oder auffälliges Verhalten. Andersrum weiß der*die Fragesteller*in dann, ob der Befragte wahrnehmungsorientierte Fähigkeiten besitzt. Zudem kann der*die Fragesteller*in verschiedene physische Objekte übergeben, weshalb der Computer weitere Fähigkeiten zur Erkennung von Objekten und generell auch von bewegten Objekten haben muss.

Der Turing-Test wird bei Forscher*innen nicht stark als angestrebtes Ziel verfolgt, aber aufgrund seines simplen Konzepts führt er vor, was künstliche Intelligenz bedeutet.

Loebner-Preis

Ein Wettbewerb für solch einen Turing-Test ist der seit 1991 von Hugh Gene Loebner ausgeschriebene Loebner-Preis. Hierbei werden entwickelte Programme einem*einer Schiedsrichter*in gegenüber gestellt, der darüber entscheidet, ob er mit einer Person oder einem Computer schreibt. Der mit 100.000 Dollar dotierte Preis für eine KI die 25 Minuten standhält, wurde bis dato noch nicht vergeben. [45] Das informative an den Turing-ähnlichen Tests ist, dass sie ein halbwegs einsichtiges Kriterium bestimmen, um zu entscheiden ob eine Maschine menschliches Handeln interpretieren kann oder nicht. Gegenargumente für den Turing-Test sind vielfältig, z.B. könnte man ein System mit Verknüpfung an eine riesige Datenbank verwenden, die auf jede Frage vorgefertigte Antworten aus der Datenbank liest. Andererseits existieren Programme die manche Personen in Analogie zum Turing-Test täuschen können.

ELIZA

Im Jahr 1966 entwickelte Joseph Weizenbaum ein solches für den Turing-Test beschriebene Computerprogramm namens ELIZA, welches die Möglichkeiten zur Kommunikation zwischen Mensch und Computer über natürliche Sprache aufzeigt. Im Allgemeinen kann man sagen, dass ELIZA eine*n humanistische*n Psychotherapeut*in simuliert.

Es verwendet ein strukturiertes Wörterbuch, welches nach Satzeingabe durchsucht wird. Falls ein Wort enthalten ist, wird nach Synonymen, meistens aber nach Oberbegriffen gesucht. Zusätzlich hierzu enthält ELIZA eine Sammlung an verschiedenen Phrasen, oftmals Fragen oder Aufforderungen, weiterzureden.

Folgend wird ein Beispieldialog mit ELIZA gezeigt:

Mensch: "Ich habe ein Problem mit meiner Mutter."

ELIZA: "Erzählen Sie mehr über Ihre Familie."

In diesem Fall hat ELIZA im Wörterbuch eine festgelegte Zuordnung für "Mutter" zu dessen Oberbegriff "Familie" umgesetzt und eine Phrase zum Themenkomplex "Familie" herausgesucht. Im Beispiel gezeigt wird eine grammatikalisch richtige Antwort geliefert, welche aber leider für den Mensch keinen Sinn ergibt. Aufgrund dieses Problems kann ELIZA eine für Menschen unverständliche Antwort liefern. Falls es kein Oberbegriff oder Phrase zum Themenkomplex findet bzw. nicht mit der Fragen anzufangen weiß, antwortet es mit Ausweichsphrasen, z.B. mit "Interessant, erzählen Sie mehr". [46]

4 Umsetzung des Bürgerportals

Dieser Teil umfasst mehrere große und praktische Aspekte, zum Ersten die Entwicklung eines komponentenbasierten Frontends, eines Backends zur Kommunikation mit dem Frontend und der damit verknüpften Datenbankbindung und der Integration einer künstlichen Intelligenz für die zukünftigen Suchanfragen der Bürger*innen.

4.1 Verwaltung mit Git

Git dient zur Versionsverwaltung des Bürgerportals. Git selbst wurde im Jahre 2005 von Linus Torvalds als Open-Source-Projekt entwickelt. Eine sehr hohe Anzahl an Softwareprojekten vertraut Git zur Versionskontrolle. Aufgrund der verteilten Architektur ist Git ein Beispiel für ein Distributed Version Control System, verteiltes Versionskontrollsystem (DVCS). Git ermöglicht Entwickler*innen, Arbeitskopien zu erstellen und bietet gleichzeitig die Möglichkeit ein Repository, das den kompletten Verlauf aller Änderungen enthält, zu verwalten. [47]

Für die Nutzung von Git kam der Einsatz von GitHub, einem webbasierten Repository Hosting Service. Hierbei ist es möglich, Code und Projektdaten zu speichern und zu verwalten sowie Änderungen am Code selbst zu verfolgen und zu kontrollieren. Zusätzlich bietet GitHub die Möglichkeiten einer Wiki und eines Kanbanboards. Mehr zur Nutzung von Kanban im Kapitel 4.2 Kanban-Entwicklungsmethode.

Die in GitHub integrierte Versionskontrolle unterstützt Entwickler*innen bei der Verfolgung und Verwaltung von Änderungen am Code in der IDE oder online im Browser. Im Wesentlichen macht GitHub es Endverbrauchern einfacher, Git für die Versionskontrolle und Zusammenarbeit zu nutzen. [48]

4.2 Kanban-Entwicklungsmethode

Zur Entwicklung des Bürgerportals kam die Kanban-Entwicklungsmethode zum Einsatz. Kanban selbst ist eine agile Methode für evolutionäres Change Management, was bedeutet, dass das bestehende Probleme in immer weitere kleineren Schritte zerlegt wird. Hierbei wird das Risiko für jedes einzelne Probleme reduziert.

Der erste Schritt bei der Einführung von Kanban besteht darin, die definierten Anforderungen beziehungsweise Probleme kleinere Aufgaben in ein Kanban-Board als Karte hinzuzufügen. Jede Karte repräsentiert hierbei eine Aufgabe. Dadurch erzielt man Transparenz über den Umfang und Verteilung der Arbeit sowie bestehende Engpässe. Im zweiten Schritt wird der Work in Process (WiP) begrenzt, also die Anzahl der parallelen Tasks. Das reduziert zum einen das Multitasking, zum anderen kann durch weniger WiP jede einzelne Aufgabe schneller erledigt werden als bisher. Kanban kennt die verschiedenen Mechanismen der WiP-Restriktion, die gängigste Art ist die sogenannten Spaltenrestriktion. [49]

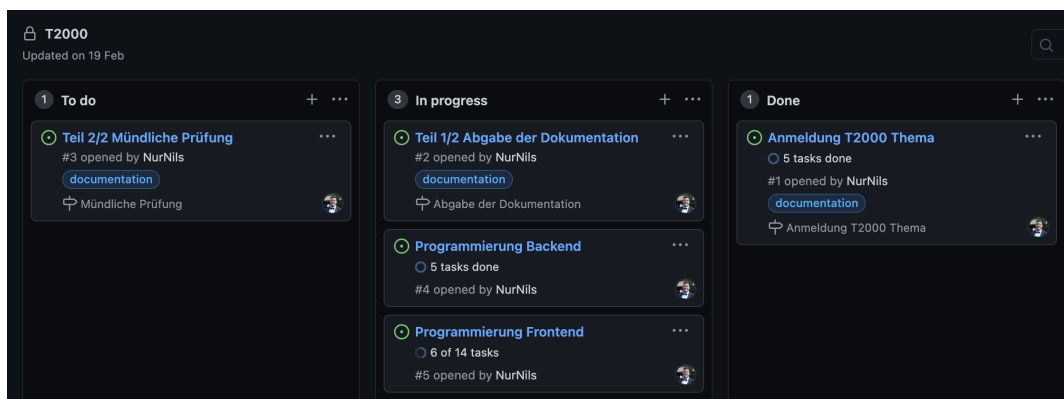


Abbildung 6: Erstelltes Kanbanboard für das Projekt T2000

Quelle: Eigene Darstellung

In Abbildung 6 gezeigt, dürfen beispielsweise nur vier Karten in der Spalte "In progress" stehen. Diese Restriktion bewirkt, dass Probleme aktiv gelöst und nicht umgangen werden. Der Kern von Kanban ist das Konzept von Flow. Das bedeutet, dass Karten, sogenannte Tickets, möglichst gleichmäßig durch das System fließen sollen, ohne lange Wartezeiten oder Blockaden. Alles, was den Flow behindert, sollte sorgfältig überprüft werden. Kanban kennt verschiedene Techniken, Metriken und Modelle. Werden diese konsequent angewendet, kann Kanban dabei helfen, den Entwicklungsprozess zu beschleunigen.

4.3 Coding Style Guideline

Basierend auf den Coding Style Guidelines von Projekten der evia solutions GmbH wurde sich gemäss der Präferenzen für den Einsatz von Prettier entschlossen. Prettier ist ein Code-Formatter für diverse Dateitypen. Er wurde als Node Package Manager (npm)-Package in den zwei Projekten eingebunden. Prettier ist opinionated und dient zur Formattierung des Quellcodes. Er kümmert sich unter anderem um Einrückungen, die Einrückungsgröße, Zeilenumbrüche und Semikolons. [50]

```
1 {  
2   "trailingComma": "es5",  
3   "tabWidth": 2,  
4   "semi": true,  
5   "singleQuote": true,  
6   "printWidth": 100,  
7 }
```

Quellcode 4.1: Prettier - Konfigurations Datei: prettierrc.json

Zur Konfiguration von Prettier wird eine JSON-Datei mit dem Namen ".prettierrc.json" im initialen Projektordner erstellt. In dieser Datei können die gewünschten Konfigurationen, Prettier Optionen genannt, definiert werden. [51]

Im Falle des Bürgerportals wurden folgende Optionen, wie im Quellcode 4.1 zu sehen, verwendet: trailingComma als es5, tabWidth, semi, singleQuote und printWidth.

4.3.1 Nachgestelltes Komma

```
1 // Input
2 obj = {
3   prettier: 'example',
4   hello: 'world'
5 };
6
7 // Prettier output
8 obj = {
9   prettier: 'example',
10  hello: 'world', //<-- Nachgestelltes Komma
11 };
```

Quellcode 4.2: Prettier - Option trailingComma

Die Option "trailingComma" von Prettier ergänzt ein nachgestelltes Komma, wo immer in ES5 gültig, zusehen in Zeile 10 des Quellcodes 4.2.

4.3.2 Tabulator Breite

```
1 // Input
2 obj = {
3   prettier: 'example',
4   hello: 'world',
5 };
6
7 // Prettier output
8 obj = {
9   prettier: 'example', //<-- Einrueckung ergaenzt
10  hello: 'world', //<-- Einrueckung ergaenzt
11 };
```

Quellcode 4.3: Prettier - Option tabWidth

Die Option "tabWidth" definiert die Anzahl der Leerzeichen pro Einrückungsebene. Die Konfigurationsdatei setzt diesen Wert auf 2, weshalb zwei Leerzeichen zur Einrückung in Zeile 9 und 10 des Quellcodes 4.3 verwendet werden.

4.3.3 Semikolon bei Anweisungsende

```
1 // Input
2 obj = {
3   prettier: 'example',
4   hello: 'world',
5 }
6
7 // Prettier output
8 obj = {
9   prettier: 'example',
10  hello: 'world',
11 }; //<-- Ergaenzung Semikolons
```

Quellcode 4.4: Prettier - Option semi

Die Option "semi" ergänzt fehlende Semikolons am Ende von jeder Anweisung. Im Quellcode 4.4 gezeigt, wird das fehlende Semikolon von "obj" aus Zeile 5 in Zeile 11 ergänzt.

4.3.4 Einfache Anführungszeichen

```
1 // Input
2 obj = {
3   prettier: "example", //<-- Doppelte Anfuehrungszeichen
4   hello: "world", //<-- Doppelte Anfuehrungszeichen
5 };
6
7 // Prettier output
8 obj = {
9   prettier: 'example', //<-- Einfache Anfuehrungszeichen
10  hello: 'world', //<-- Einfache Anfuehrungszeichen
11 };
```

Quellcode 4.5: Prettier - Option singleQuote

Die Option "singleQuote" ersetzt doppelte Anführungszeichen mit einfachen, wie im Quellcode 4.5 veranschaulicht werden beide Strings in Zeile 3 und 4 manipuliert.

4.3.5 Maximale Zeilenlänge

In Code-Styleguides sind die Regeln für die maximale Zeilenlänge oft auf 100 oder 120 festgelegt. Wenn Menschen Code schreiben, streben sie jedoch nicht danach, die maximale Anzahl von Spalten in jeder Zeile zu erreichen. Entwickler*innen verwenden oft Leerzeichen, um lange Zeilen zur besseren Lesbarkeit aufzulockern. In der Praxis endet die durchschnittliche Zeilenlänge oft weit unterhalb des Maximums.

Die Option "printWidth" funktioniert nicht auf die gleiche Weise. Sie ist nicht die harte Obergrenze für die erlaubte Zeilenlänge. Sie ist eine Möglichkeit, Prettier mitzuteilen, wie lang man die Zeilen ungefähr haben möchte. Prettier wird sowohl kürzere als auch längere Zeilen erzeugen, aber im Allgemeinen versuchen, die angegebene maximale Zeilenlänge einzuhalten. [52]

4.4 Darstellung des Bürgerportals

Zur Darstellung des Bürgerportals erfolgten verschiedene Implementationen, welche im folgenden Text erklärt werden. Hinzufügend wird auf den verwendeten Architekturansatz des Angular Projektes eingegangen.

4.4.1 Architektur in Angular

Die Architektur einer Angular-Anwendung beruht auf bestimmte grundlegende Konzepte. Die Grundbausteine des Angular-Frameworks sind Angular-Components, die in NgModulen organisiert sind. NgModule sammeln zusammengehörigen Code in Funktionssätzen. Eine Angular-Anwendung wird durch eine Reihe von NgModulen definiert. Eine Anwendung verfügt immer über mindestens ein Root-Modul, das das Bootstrapping ermöglicht und verfügt in der Regel über viele weitere Funktionsmodule. Modules, Components und Services sind Klassen, die Decorators verwenden. Diese Dekoratoren markieren ihren Typ und stellen Metadaten bereit, die Angular mitteilen, wie sie verwendet werden sollen. [53]

NgModule

NgModule sind Container für einen zusammenhängenden Codeblock, der einer Anwendungsdomäne, einem Workflow oder einem eng verwandten Satz von Funktionen gewidmet ist. Sie können Components, Service Providers und andere Codedateien enthalten, deren Geltungsbereich durch das enthaltende NgModule definiert wird. Sie können Funktionen importieren, die von anderen NgModulen exportiert wurden und ausgewählte Funktionen zur Verwendung durch andere NgModule exportieren. [54]

Jede Angular-Applikation hat mindestens eine NgModule-Klasse, das Root-Modul, das herkömmlicherweise AppModule heißt. Es befindet sich in der Datei mit dem Namen app.module.ts. Die Anwendung wird durch das Bootstrapping des Root-NgModules gestartet.

Während kleinere Anwendungen möglicherweise nur über ein NgModule verfügen, verfügen die meisten großen Anwendungen über viele weitere Funktionsmodule.

Component

Die Metadaten für eine Komponentenkategorie verknüpft man mit einer Vorlage, die eine Ansicht definiert. Eine Vorlage kombiniert normales HTML mit Angular-Anweisungen und binding-Markup, die es Angular ermöglichen, den HTML-Code zu ändern, bevor er für die Anzeige gerendert wird. [55] Optional können noch Styling Dateien, wie beispielsweise eine CSS-Datei, verknüpft werden, die dann nur für diese Vorlage Änderungen vornimmt.

Angular erstellt, aktualisiert und zerstört Komponenten, während sich der/die Benutzer*in durch die Anwendung bewegt. Die Anwendung kann zu jedem Zeitpunkt im Lebenszyklus durch optionale Lebenszyklus-Hooks wie `ngOnInit()` und `ngOnDestroy()` Maßnahmen ergreifen.

Service

Die Metadaten für eine Serviceklasse stellen die Informationen bereit, die Angular benötigt, um die Komponenten durch Dependency Injection (DI) zur Verfügung zu stellen. [56]

Directive

Direktiven sind Klassen, die Elementen in Angular-Anwendungen zusätzliches Verhalten hinzufügen. Mit den integrierten Direktiven von Angular kann man unter anderem Formulare, Listen und das, was Benutzer*innen sehen, verwalten. [57] Mehr dazu im Unterkapitel 4.4.4.

4.4.2 Styling mit SCSS

Zur Erstellung der Styling Dateien der zu programmierenden Komponenten für die Angular-Web-Applikation entschied man sich für die Verwendung von Sassy CSS (SCSS). SCSS ist die neuere Syntax von Syntactically Awesome Style Sheets (Sass). Sass ist eine Erweiterung von CSS, die verschachtelte Regeln, Variablen, Mixins, Selektorvererbung und mehr hinzufügt. SCSS verwendet eine Blockformatierung wie die von CSS. Es verwendet geschweifte Klammern, um Codeblöcke zu kennzeichnen und Semikolons, um Regeln innerhalb eines Blocks zu trennen. Die ältere Syntax, die als die eingerückte Syntax bezeichnet wird, verwendet Einrückungen, um Codeblöcke zu trennen und Newline- Zeichen, um Regeln zu trennen. Diese erweiterten CSS-Dateien erhalten traditionell die Erweiterungen .sass bzw. .scss. [58]

4.4.3 Responsive Webdesign

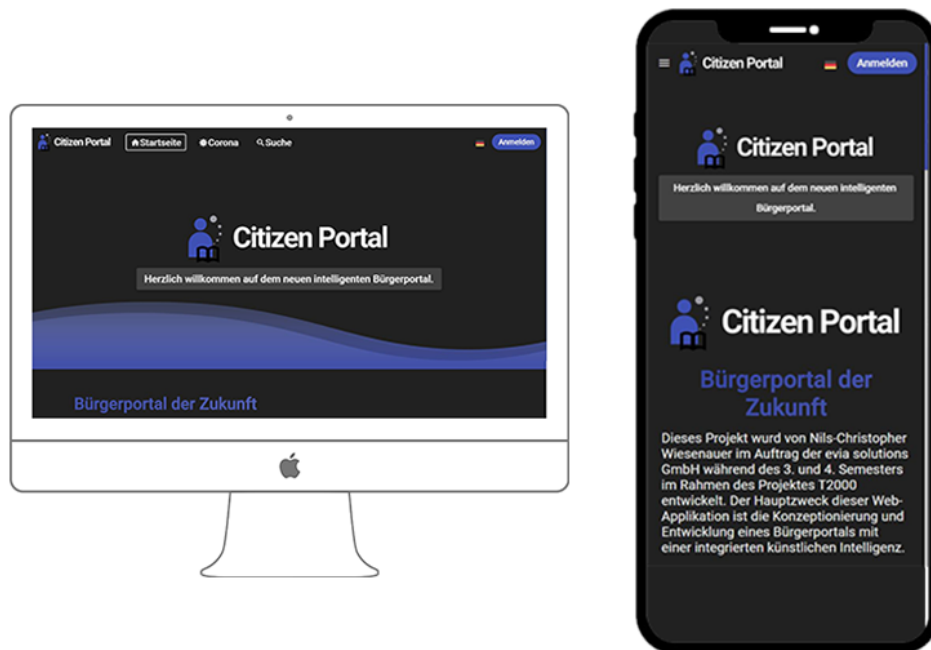


Abbildung 7: Citizienportal UI - Startseite

Quelle: Eigene Darstellung

Da das Bürgerportal möglichst auf vielen Endgeräte verfügbar sein soll, wurde durch Anwendung von mixins verschiedene Bildschirmgrößen für ein responsive Design definiert.

Mit Mixins werden Stile definiert, die im Stylesheet wiederverwendet werden können. Sie machen es einfach, nicht-semantische Klassen zu vermeiden und Sammlungen von Stilen in Bibliotheken zu verteilen.

Mixins werden mit der `@mixin` at-Regel definiert. Der Name eines Mixins kann ein beliebiger Sass-Bezeichner sein und jede andere Anweisung als Anweisungen der obersten Ebene enthalten. Sie können verwendet werden, um Stile zu kapseln, die in einer einzigen Stilregel abgelegt werden können und eigene Stilregeln enthalten, die in andere Regeln verschachtelt oder auf der obersten Ebene des Stylesheets eingefügt werden können. Oder man verwendet mixins lediglich, um Variablen zu ändern. Mixins werden mit der `@include` at-Regel in den aktuellen Kontext eingebunden, gefolgt vom definierten mixin-Namen. [59]

4.4.4 Internationalisierung mit ngx-translate

Da das Bürgerportal möglichst viele Menschen erreichen soll, wurde ngx-translate implementiert und die Sprachen deutsch und englisch gepflegt. In der i18n-Bibliothek ngx-translate werden Übersetzungen zur Laufzeit durchgeführt. Dafür liefert die Bibliothek eine Direktive, eine Pipe und ein Service mit. Durch den flexiblen Aufbau ist es möglich, Übersetzungen auszuwechseln, ohne die Applikation neu starten zu müssen. [60]

Directive

```
1 <div translate [translateParams]="{value: 'Welt'}">HALLO</div>
```

Quellcode 4.6: ngx-translate - Translate Direktive

Die Direktive "translate" ermöglicht die direkte Anweisung eines Translations-Keys in ein HTML-Element. Zu sehen in Quellcode 4.6 wird der Translate-Key "HALLO" mit Translate-Parameter verwendet. Bei den angegebenen Translate-Parametern handelt es sich um ein Objekt, welches verschiedene Werte annehmen kann. Je nachdem, wie der Translation-Key in der Übersetzungs JSON-Datei hinterlegt ist, können optional Parameter übergeben werden. In diesem Beispiel wird der Parameter value auf den String "Welt" gesetzt.

translate Pipe

```
1 <h2>{{ 'HALLO' | translate: { value: 'Welt' } }}</h2>
```

Quellcode 4.7: ngx-translate - Translate Pipe

Das gezeigte Beispiel der Translate-Direktive kann auch mit Hilfe der hinzugefügten Translate-Pipe erfolgen. Im Quellcode 4.7 wird hierzu in einem HTML-Element die Translate-Pipe nach dem Translation-Key hinzugefügt. Optional wie in Quellcode 4.6 können nach der Pipe durch Trennung mit einem Doppelpunkt die Translate-Parameters übergeben werden.

```
1 { "HALLO" : "Hallo {{value}}" }
```

Quellcode 4.8: ngx-translate - i18n Beispiel-Datei (de.json)

Durch Anwendung der Translate-Pipe oder der Translate-Direktive können Parameter in den Übersetzungstext übernommen werden. In den Beispielen gezeigt, wird der Wert für value gesetzt und an den eindeutigen Translate-Key "HALLO" übergeben. Der Output lautet dann "Hallo Welt".

Translate Service

```
1 export class AppComponent {
2   constructor(public translate: TranslateService) {
3     translate.addLangs(['de', 'fr']);
4     translate.setDefaultLang('de');
5     const browserLang = translate.getBrowserLang();
6     translate.use(browserLang.match(/de|fr/) ? browserLang : 'de');
7   }
8 }
```

Quellcode 4.9: ngx-translate - Translate Service

Für die Verwendung des Translate Services kann man diesen in der globalen Komponente, im Sourcecode 4.9 Zeile 2, im Konstruktor importieren. Dort können nun Sprachen hinzugefügt (Zeile 3) und eine Standardsprache (Zeile 4) ausgewählt werden. Für eine mehrsprachige Webseite ist es sinnvoll, Zugriffe aus der ganzen Welt zu erkennen und folgend die Sprache des Landes auszuwählen. Dies erfolgt in den Zeile 5 und 6. Zuerst wird mit Hilfe der vom TranslateService eingebauten Funktion "getBrowserLang()" die Sprache des Browsers erkannt. Nach Erkennung der Sprache wird geschaut, ob diese in der Applikation verfügbar ist. Ist dies der Fall, wird diese gesetzt, andernfalls wird auf die Standardsprache zurückgegriffen.

4.4.5 Textdarstellung mit Markdown

Zur Gliederung und Formatierung von editierbaren Texten in Artikeln und Beiträge einigte man sich auf die Verwendung von Markdown.

Markdown wurde von Aaron Swartz und John Gruber entwickelt und Ende 2004 veröffentlicht. Nach monatelanger Arbeit gelang es den beiden die Syntaxdetails niederzuschreiben. Die erstellte Implementation für Perl benannten sie Markdown. Das Ziel von Markdown war, das Schreiben einfacher Webseiten und Webblock-Einträge so einfach wie das Schreiben einer E-Mail zu machen, indem man den gleichen Syntax verwendet und diesen automatisch in HTML konvertiert. [61]

Um Markdown in unserem Angular Projekt zu verwenden, fügte man die Angular Bibliothek ngx-markdown mithilfe von npm hinzu. Die erste Version dieser Bibliothek wurde 2017 von Jean-Francois Cere, einem in Kanada lebenden Fullstack-Developer veröffentlicht und bis heute von ihm verwaltet. [62]

Nachfolgend sind ein paar Funktionen von ngx-markdown gelistet:

- Syntax-Highlighting mit Prism.js, einem leichtgewichteter, erweiterbarer Syntax-Highlighter, der mit Blick auf moderne Webstandards entwickelt wurde. Er wird in Millionen von Websites verwendet. [63]

Sogar Emojis können durch Aktivierung eines Emoji-Toolkits als Shortcode (bspw. :heart: für das rote Herz) hinzugefügt werden

- Rendering von Matheformeln durch Einbindung der KaTeX JavaScript Bibliothek
- Angular HttpClient Unterstützung, um eine Remote-Datei direkt laden zu können

Zur Nutzung dieser Bibliothek benötigt man folgenden Import in der app.module.ts Datei: MarkdownModule.forRoot(). Man kann nun die ngx-markdown-Komponente in drei verschiedenen Weisen verwenden. [64]

```
1 <!-- Static markdown -->
2 <markdown>
3   # Put your markdown here
4 </markdown>
```

Quellcode 4.10: Markdown - Statisches Modell

Die erste Weise (Quellcode 4.10) wird für statischen Markdown verwendet, indem man es direkt in das HTML-Element hinzufügt.

```
1 <!-- Loaded from Remote-URL -->
2 <markdown
3   [src]='path/to/your/file.md' "
4   (load)="onLoad($event)"
5   (error)="onError($event)">
6 </markdown>
```

Quellcode 4.11: Markdown - Remote-Datei (.md)

Beim zweiten Fall, Quellcode 4.11, wird eine Remote-Datei durch einer Uniform Resource Locators (URL) geladen, indem man den Inhalt der gewünschten Datei über die src-Eigenschaft lädt.

```
1 <!-- Variable Binding -->
2 <markdown
3   [data]="objekt[translateService.currentLang]"
4   (ready)="onReady()">
5 </markdown>
```

Quellcode 4.12: Markdown - Variable Binding

Im letzten Fall wird eine Variable über die data-Eigenschaft an der Komponente gebunden, zu sehen in Zeile 3 des Quellcodes 4.12. Im Falle des Bürgerportals kam hier der Einsatz der von ngx-translate bereitgestellten Funktion der Internationalisierung. Es wird der Wert der derzeitig ausgewählten Sprache durch Einsatz von translateService.currentLang genutzt.

4.4.6 Angular Material

Angular Material ist eine UI-Komponentenbibliothek für Angular Entwickler. Angular Material-Komponenten helfen beim Erstellen attraktiver, konsistenter und funktionaler Webseiten unter Einhaltung moderner Webdesign-Prinzipien wie Browser-Portabilität, und Geräteunabhängigkeit. Zudem helfen sie bei der Erstellung schneller, schöner und reaktionsschneller Websites. Bereitgestellt wird Angular Material von Google LLC, das durch Inspiration des Google Material Designs entstand. [65]

Mit dem Theming-System von Angular Material können Farb- und Typografiestile für Komponenten in der Anwendung angepasst werden. Man kann entweder bereits vorgefertigte Designs verwenden oder ein benutzerdefiniertes Design hinzufügen. Zudem gibt es die Auswahl zwischen einem Light- und Dark-Theme. Damit die Änderungen global in der Anwendung zu sehen sind, werden die Theming-Einstellungen in einer globalen Angular-Styling Datei ergänzt. [66]

Die Komponenten von Angular Material werden von einem separaten Components-Team entwickelt, welches Teil des Angular-Teams bei Google ist. Das Team umfasst sowohl Google-Mitarbeiter*innen als auch Community-Mitwirkende aus der ganzen Welt. Das Team hat zwei Aufgaben. Es erstellt hochwertige UI-Komponenten, die Entwickler in ihre vorhandene Anwendung einfügen können und es stellt Tools bereit, mit denen Entwickler*innen ihre eigenen benutzerdefinierten Komponenten mit gemeinsamen Interaktionsmustern erstellen können.

Unter einer hochwertigen UI-Komponenten für Angular Material versteht man eine Komponente, die Internationalisiert ist, damit alle Benutzer*innen sie verwenden können. Sie muss die Nutzung von einfachen APIs haben, die Entwickler*innen nicht verwirren. Die Komponente soll sich bei einer Vielzahl von Anwendungsfällen wie erwartet ohne Fehler verhalten. Dieses Verhalten muss sowohl mit Unit- als auch mit Integrationstests gut getestet werden. Sie soll anpassbar im Rahmen der Material Design-Spezifikation sein und die Leistungskosten minimieren. Zudem muss der Code sauber und gut dokumentiert sein, um als Beispiel für Angular-Entwickler*innen zu dienen. [67]

4.5 Entwicklung einer REST API

Zur Verwirklichung des Backends und dem damit verbundenen API kam die grundsätzliche Denkweise von REST. REST ist ein Architekturstil, der beschreibt, wie die verteilten Systeme miteinander kommunizieren können. Dieser Architekturstil hat Auswirkungen auf die Strukturierung einer komplexen Client-Server Applikation und ist durch mehrere Randbedingungen gegeben:

- Kommunikation zwischen Client und Server ist zustandslos
- HTTP wird als Transportprotokoll und semantische Grundlage verwendet
- Uniform Resource Identifier (URI)s sind Identifikatoren für Ressourcen auf dem Server
- Datenformat (XML, JSON, etc.) ist nicht vorgegeben

Durch die Zustandslosigkeit von REST werden Implementationen von skalierbaren Schnittstellen ermöglicht. Aufgrund der meist angewandten Versionierung einer REST-API können Client und Server mehrheitlich unabhängig voneinander entwickelt werden. Sobald eine neue Version erstellt wurde und Rückwärtskompatibilität alterer Versionen garantiert wird, kann die Client-Software (das Frontend) weiterentwickelt werden. REST baut auf HTTP auf. Explizit werden sogenannte HTTP-Verben (GET, PUT, POST, DELETE etc.) verwendet, um Ressourcen zu erhalten oder zu manipulieren. Durch Einsatz des HTTP-Standards wird das Zwischenspeichern, das sogenannte Chaching, ermöglicht, vorausgesetzt die Software sendet den entsprechenden Cache-Header. Damit eine bestimmte Instanz der Ressource angesprochen werden kann, muss der richtige und eindeutige URI verwendet werden. Jeder Objekttyp und jedes Objekt sollte eindeutig identifizierbar sein. Das Datenformat von Objekten und Collections ist nicht definiert. Die Serialisierung der Daten bleibt dem/der Softwareentwickler*in überlassen. Heute ist JSON weit verbreitet, da es sich um ein kompaktes und leicht lesbares Datenformat handelt. [68]

4.5.1 Geplante Umsetzung

Die API für das Bürgerportal soll REST mit JSON als Datentyp verwenden. Jede Ressource, die über das Frontend angefragt werden kann, soll auch über die notwendigen REST-API-Endpunkte verfügen. Um den Quellcode möglichst verständlich und erweiterbar zu gestalten, soll auf der Server-Seite die API verwirklicht werden. Somit ist die API entkoppelt von sonstigem Quellcode. Eine Versionierung der API und das Caching der API-Zugriffe ist geplant.

4.5.2 Erweiterung mit JSend

Es gibt viele Webdienste, die JSON-Daten bereitstellen und jeder hat seine eigene Art, Antworten zu formatieren. Zudem versuchen Entwickler oftmals neue Ideen zu kreieren, wenn es darum geht, Daten von ihren Servern zu übermitteln. Es gibt zwar viele gemeinsame Muster für die Strukturierung dieser Daten, dennoch gibt es keine Konsistenz bei Dingen wie der Namensgebung oder den Antworttypen. Außerdem fördert eine Konsistenz die Zufriedenheit und Einheitlichkeit zwischen Backend-Entwicklern und Frontend-Designern, da jeder einen gemeinsamen Ansatz für die Interaktion mit dem anderen erwarten kann. Zur Definierung des allgemeinen Nachrichtenaustausches entschloss man sich für die Erweiterung der JSON-Antworten mit JSend.

JSend ist eine von Robert Treat entwickelte Spezifikation, die einige Regeln festlegt, wie JSON-Antworten von Webservern formatiert werden sollten. JSend konzentriert sich hierbei auf das Messaging auf der Anwendungsebene (im Gegensatz zu Protokoll- oder Transportebene), was es ideal für den Einsatz in REST-artigen Anwendungen und APIs macht. Hierbei besteht ein sogenanntes JSend-Objekt immer aus einem Typ und dem Status der Anfrage. [69]

JSend-Typen

Beim Typ "success" für erfolgreiche Anfragen, wenn beispielsweise Daten zurückgeliefert werden, müssen zusätzlich Status "status" und die Daten als "data" deklariert werden.

Der Typ "fail" wird genommen, wenn es ein Problem mit den zu übermittelten Daten gab oder eine Vorbedingung des API-Aufrufes nicht erfüllt wurde. Hier müssen auch zusätzlich Status und Data gesetzt werden.

Im letzten Falle, falls ein Fehler mit der Bearbeitung der Anfrage aufgetreten ist, wird der Typ "error" geliefert. Hier muss ein Status und eine Fehlernachricht als "message" gesetzt werden. Zusätzlich können hier auch der dazugehörige HTTP-Statuscode als "code" und Data gesetzt werden.

```
1 {
2   status: 'success',
3   data: {
4     training: [{
5       _id: '...',
6       search: 'Covid-19',
7       category: 'corona',
8       userId: '...',
9       trained: true
10    }],
11    overview: {
12      total: 120,
13      categories: ['corona', 'kfz']
14    }
15  }
16 }
```

Quellcode 4.13: REST-API - Erfolgreiche Antwort als JSend-Objekt

Im Falle des Bürgerportals sieht eine erfolgreiche Antwort wie in Quellcode 4.13 aus. Der in der Datenbank gespeicherte Trainingsdatensatz für das Stichwort "Covid-19" wird in "data" hinterlegt. Zusätzlich wird das Objekt mit einer Übersicht aller Trainingsdaten zurückgeliefert.

4.5.3 Postman

Die Anfragen der REST-API werden mit dem Programm Postman getestet. Bei Postman handelt es sich um ein API-Entwicklungstool, das für das Testen, Erstellen und Ändern von APIs verwendet wird. HTTP-Anfragen (GET, POST, PUT, DELETE) können mit Postman durchgeführt werden. Zudem können Integrationstests mit Postman durchlaufen werden, damit die API wie erwartet funktioniert. [70]

In Postman selbst wurden Umgebungsvariablen hinterlegt, damit unter anderem bei jeder API-Anfrage die richtige API-URL für das lokale oder produktive System verwendet wird. So ist für den lokalen Zustand die URL "http://localhost:3000/" hinterlegt. Im Produktivmodus steht die URL "https://api.citizienportal.com/".

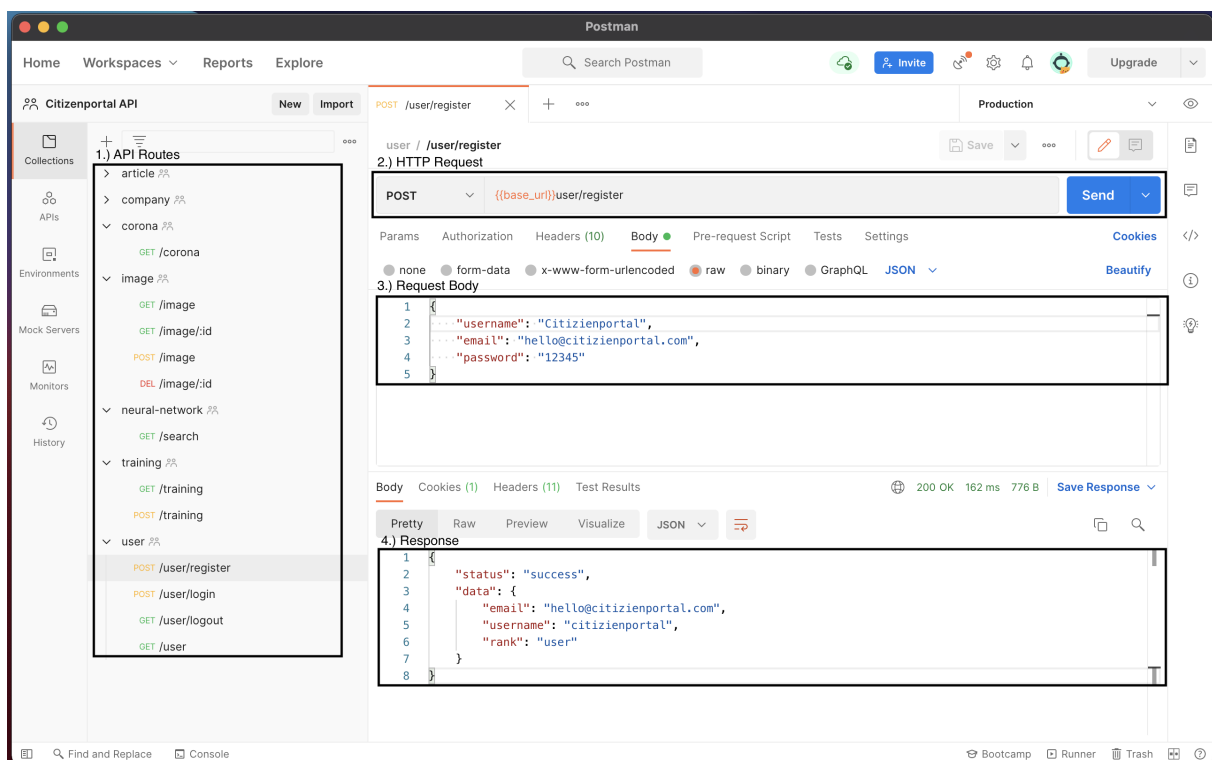


Abbildung 8: Citizienportal API - Postman

Quelle: Eigene Darstellung

Im Beispiel der Abbildung 8 gezeigt, sind auf der linken Seite alle API Routen der Bürgerportal API definiert. Eine Route besitzt hierbei die Informationen, um welche HTTP Anfrage es sich handelt, die zu abfragende URL, optionale Parameter und ggfs. den Request Body. Im Beispiel ist eine POST Anfrage für die Registrierung eines neuen Account zu sehen. Hierzu erfolgt Anfrage an den Endpoint "/user/register/" mit den definierten Accountdaten als Request Body.

4.5.4 Anbindung an eine MongoDB Datenbank

Obwohl Node.js gut mit MySQL-Datenbanken funktioniert, ist die perfekte Kombination eine NoSQL-Datenbank wie MongoDB, bei dem das Schema nicht gut strukturiert sein muss. MongoDB ist eine verteilte Datenbank, die Ad-hoc-Abfragen, Echtzeitintegration und effiziente Indizierung ermöglicht. Darüber hinaus ist MongoDB Open Source und perfekt für sich häufig ändernde Daten. Es bietet auch serverseitige Datenvalidierung an. [71]

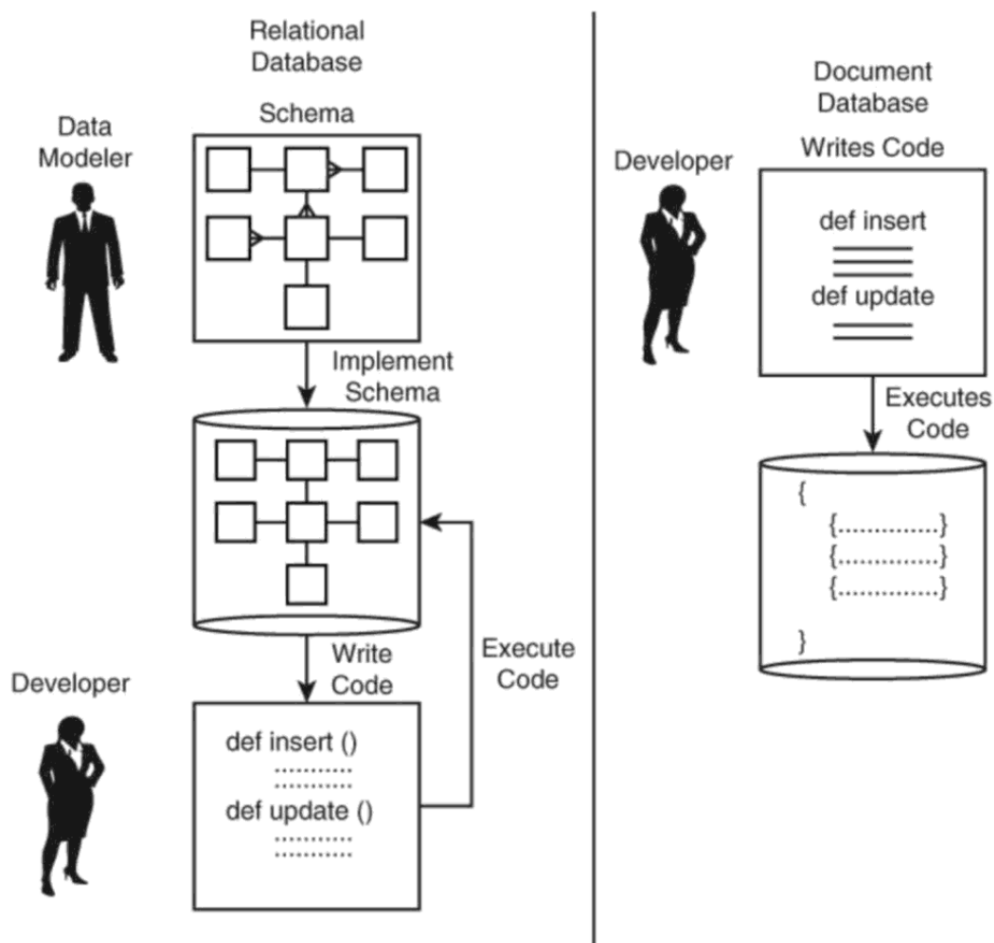


Abbildung 9: Relationale Datenbank vs. Dokumentenbasierte Datenbank

Quelle: Raj R. Parmar und Sudipta Roy [72]

Dokumentendatenbanken sind so konzipiert, dass sie Variationen von Dokumenten innerhalb einer Sammlung, der sog. Collection, berücksichtigen können. Da sich jedes Dokument von allen zuvor eingefügten Dokumenten unterscheiden kann, ist es nicht sinnvoll vor dem Aufbau und der Befüllung der Datenbank eine Modellierung vorzunehmen. Diese Freiheit, zu sehen auf der rechten Seite in Abbildung 9, von der

Notwendigkeit, die Datenbankstruktur im Voraus zu definieren, wird durch den Begriff "schemalos" ausgedrückt, der häufig zur Beschreibung von Dokumentendatenbanken verwendet wird. [72]

Somit können Entwickler*innen direkt mit der Arbeit beginnen, ohne dass eine Datenbankmodellierung, wie es der Fall bei einer relationalen Datenbank ist.

Infolgedessen wurde sich für MongoDB als Datenbank entschieden. MongoDB ist eine dokumentenorientierte not only SQL bzw. non-SQL (NoSQL)-Datenbank, die für die Speicherung großer Datenmengen verwendet wird. Aufgrund dieser Eigenschaft ist MongoDB passend für Anwendungen im Bereich der künstlichen Intelligenz. Zum Trainieren einer KI benötigt man viele Daten zu einer genauen Entscheidung des Outputs. In der MongoDB selbst werden Daten in JSON ähnlichen Dokumenten verwaltet. Der Aufbau eines MongoDB Prozesses besteht aus Datenbanken und Collections.

Ein Prozess kann mehrere Datenbanken verwalten. Eine Datenbank wiederum mehrere Collections, analog zu MySQL, in der mehrere Tabellen in einer Datenbank vorhanden sein können. Die Datenbank und Collection zusammen mit einem Punkt bilden den Namespace, einem Namensraum in welchem das Objekt eindeutig bezeichnet wird. Für die Bürgerportal-Datenbank, um alle notwendigen Daten zu verwalten und eine Collection, die die Training-Daten enthält wurde der Name "database.training" gewählt.

Vergleichbar mit einer Tabelle einer relationalen Datenbank enthält die Collection Dokumente. Ein wesentlicher Unterschied im Vergleich zur relationalen Datenbank können in einer Collection Dokument völlig unterschiedlich aufgebaut sein. Sie müssen weder einem Schema noch müssen die Werte desselben Schlüssels vom gleichen Datentyp sein.

Zur Abfrage von Daten aus der Datenbank erlaubt es Mongo, jederzeit jedes Feld abzufragen. Zudem bietet Mongo die Unterstützung bei Bereichsabfragen, das Suchen von regulären Ausdrücken und weitere spezielle Suchabfragen. Die genannten Abfragen lassen auch benutzerdefinierte JavaScript-Funktionen zu. [73]

Abfragen können auch nur einen Teil des Dokumentes wiedergeben, die Ergebnisse sortieren, überspringen oder limitieren. Zusätzlich können Aggregationen ähnlich dem GROUP BY von SQL angewandt werden.

4.5.5 Authentifizierungs-System

Für das Authentifizierungssystem und der damit wichtigen sicheren Übertragung zwischen Frontend und Backend kommt der Einsatz von JSON Web Token (JWT). JWT ist ein offener Standard, der eine kompakte und in sich geschlossene Methode zur sicheren Übertragung von Informationen zwischen Parteien als JSON definiert. Diese Informationen können verifiziert werden, da sie eine digitale Signatur haben. Ein JWT besteht aus drei Teilen: Dem Header, Payload und der Signatur. [74]

Header

```
1 {  
2   "alg": "HS256",  
3   "typ": "JWT"  
4 }
```

Quellcode 4.14: JWT - Header

Beim Header handelt es sich um ein JSON-Element, welches beschreibt, um welchen Token-Typ es sich handelt und welche Verschlüsselungsmethode zum Einsatz kommt. Insgesamt vier Felder können hierbei gesetzt werden.

Der Typ beschreibt den Internet Assigned Numbers Authority (IANA) Medientyp des Tokenes. Um den Medientyp "application/jwt" zu beschreiben, wird dieser immer auf "JWT" gesetzt. Der Content Typ wird benötigt, wenn das JWT ein anderes JWT als Payload enthält. Das Algorithm Feld gibt an, welche Signaturmethode zum Einsatz kommt. Normalerweise kommen hier HMAC mit Secure Hash Algorithm, Sicherer Hash-Algorithmus (SHA)-256 (HS256) und das Rivest–Shamir–Adleman (RSA) Verfahren mit SHA-256 (RS256) zum Einsatz. [74]

Payload

```
1 {  
2   "iss": "DHBW",  
3   "_id": "1234567890",  
4   "name": "Username",  
5   "admin": true  
6 }
```

Quellcode 4.15: JWT - Payload

Der Payload ist wie der Header ein JSON-Element. Dieses beschreibt die Claims. Einige Claims sind bereits vordefiniert. Darunter fallen Issuer (iss) für den Aussteller des Tokens, Subject (sub) für welches Subjekt die Claims gelten, Audience (aud) beschreibt, für welche Zieldomäne das Token ausgestellt wurde, Expiration Time (exp) für das Ablaufdatum des Tokens in Unixzeit, Not Before (nbf) für den Start der Gültigkeit, Issued At (iat) für das Ausstellungsdatum des Tokens und die JWT ID (jti) zur eindeutigen Identifikation des Tokens. [74]

Signatur

```
1 const encodedString = `${base64UrlEncode(header)}.${base64UrlEncode(payload)}`;  
2 const hash = HS256(encodedString, secret);
```

Quellcode 4.16: JWT - Signatur

Die Signatur wird durch JSON Web Signature (JWS) aufgebaut und mit Hilfe von Base64 erstellt. Hierbei wird der Header und der Payload in Base64 kodiert und durch einen Punkt in einen String zusammengefügt. Dieser String wird dann in der definierten Hashmethode beispielsweise in HMAC mit SHA-256 (HS256) gehashed. Hierbei wird oftmals ein selbstdefiniertes Secret für die Kodierung und Dekodierung verwendet. [74]

Übertragung

Das generierte JWT-Token wird dem Frontend nach Erstellung übermittelt, damit die Nutzer*innen sicher angemeldet bleiben beziehungsweise Anfragen, die eine Authentifizierung benötigen, mitsenden können. Es gibt zwei unterschiedliche Arten, wie das JWT-Token im HTTP-Header übertragen werden kann. Der erste Fall ist über das Authorization-Feld. Hier wird ein Präfix "Bearer" vor dem eigentlichen Token gesetzt, damit der Typ des Tokens klar sichtbar ist. Der zweite Fall ist über ein Cookie-Feld. Hierbei wird ein Cookie mit dem Name "token" gesetzt. [74]

4.5.6 Konkrete Umsetzung

Die API-Routen des Bürgerportals wurden mit Hilfe von Express.js umgesetzt. Express ist ein einfaches und flexibles Node.js-Framework von Webanwendungen, das zahlreiche leistungsfähige Features und Funktionen für Webanwendungen und mobile Anwendungen bereitstellt. Hierbei bestehen die Routen aus jeweils einem oder mehreren Requesthandler. Ein Client übergibt eine HTTP-Anforderungsnachricht an den Server, welcher wiederrum eine HTTP-Antwortnachricht im ausgewählten JSend-Format zurückgibt. Es Anforderungsmethode wie bspw. GET, PUT, POST und DELETE definiert sein. Durch Einsatz von Middlewares kommt die Authentifizierung zum Einsatz. Eine Middleware ist nichts anderes als eine Funktion, die über die Weiterleitungsebene in Express vor dem letzten Anforderungshandler aufgerufen wird. Deshalb befindet sich diese Funktion in der Mitte zwischen einer unformatierten Anforderung und der endgültigen beabsichtigten Weiterleitung [75].

```
1 const express = require('express');
2 const router = express.Router();
3 const TRAINING = require('../models/training');
4
5 router.get('/', authenticate, (req, res) => {
6   TRAINING.getByUserId(req.user._id, (err, training) => { ... });
7 });
8
9 router.post('/', authenticate, (req, res) => {
10   req.body.userId = req.user._id;
11   req.body.trained = false;
12   TRAINING.add(req.body, (err, training) => { ... });
13 });
14
15 module.exports = router;
```

Quellcode 4.17: REST-API - Training Routes

Das Beispiel im Quellcode 4.17 zeigt zwei Definitionen der Training API-Route mit der Middleware `authenticate` zur Authentifizierung des Clients.

Bei Zeile 5 wird eine HTTP-GET Anfrage für das Erhalten aller Trainings des authentifizierten Accounts definiert. Wie man sehen kann, werden zuerst alle vom Account hinterlegten Training-Dokumente aus der MongoDB geladen. Danach werden alle bisherigen Trainings gezählt und als Overview mit den Account Trainings-Daten im JSend-Format als Data-Objekt zurückgeliefert.

Zeile 9 definiert eine HTTP-POST Anfrage für das Erstellen eines Training-Datensatzes. Auch hier wird überprüft, ob der/die Nutzer*in authentifiziert ist. Sobald der Training-Datensatz gespeichert wurde, wird er zurück an den/die Nutzer*in als JSend-Objekt geliefert, damit er in der Übersicht im Frontend auftauchen kann.

Aus Zeitgründen wurde kein Caching und keine Versionierung implementiert. Durch das Einbinden eines zusätzlichen Präfixes für die API-Route, wäre eine Versionierung problemlos möglich.

4.5.7 Brain.js

Zur Erweiterung des Bürgerportals mit einer intelligenten Suche und dem damit verbundenen maschinellen Lernen dient brain.js. Brain.js ist eine GPU-beschleunigte Bibliothek für Neuronale Netze, die in JavaScript geschrieben ist. [76]

Hierbei handelt es sich um eine Fortsetzung der von harthur entwickelten JavaScript-Bibliothek für Neuronale Netzwerke namens brain. Brain wurde seit dem 21. Juni 2015 nicht mehr gepflegt. Aufgrund der Einfachheit dieser Bibliothek wollten mehrere Entwickler*innen die Arbeit von harthur fortsetzen. Ionică Bizău entschloss sich am 13. Oktober 2015 die Maintainerin zu kontaktieren mit Erfolg. Kurt Ericson, Abhishek Soni und Ionică Bizău schlossen sich zusammen und gründeten die Heather Arthur Organization "harthur-org" für die Weiterentwicklung von brain, welche nach einer Zeit auf GitHub in den Namen BrainJS geändert wurde. [77]

Der neue Name brain.js für das Repository entstand aus einer Abstimmung mit 9 Vorschläge und 28 Stimmen. Brain.js wurde damals als klarer Gewinner mit 13 Stimmen gewählt. Am 15.04.2016 wurde dann das erste npm-Package unter dem Namen brain.js durch Ionică Bizău veröffentlicht. [78]

Die Open-Source Bibliothek ist auf GitHub unter der MIT-Lizenz verfügbar. Das Training des neuronalen Netzes in brain.js ist sehr rechenintensiv, daher sollte man versuchen, das Netzwerk offline zu trainieren und als JSON-Datei abzuspeichern, um das vortrainierte Netzwerk folgend in der Webapplikation einzubinden. Zurzeit gibt es vier verschiedene Typen, die sich in insgesamt acht verschiedene Arten unterteilen, ein neuronales Netzwerk mittels brain.js zu erstellen, da verschiedene neuronale Netze für unterschiedliche Dinge gut sein können: Feedforward-Netzwerk (NeuralNetwork, NeuralNetworkGPU), Recurrent Neural Network (RNNTimestep, RNN), Long Short Term Memory Neural Network (LSTMTimestep, LSTM) und Gated Recurrent Unit (GRUTimestep, GRU). [79]

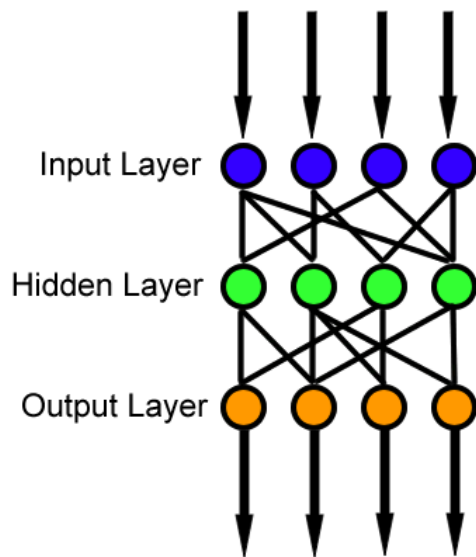


Abbildung 10: Feedforward - Neuronales Netzwerk

Quelle: Paskari, Nov. 2006

Bei NeuralNetwork und NeuralNetworkGPU handelt es sich um Feedforward neuronales Netzwerke, bei denen die Verbindungen zwischen den Knoten keinen Zyklus bilden. [80]

In einem Feedforward-Netzwerk bewegen sich somit die Informationen immer in eine Richtung und gehen nie zurück. Veranschaulicht wird das in Abbildung 10. Ein Feedforward Neural Network kann einfache Dinge sehr gut klassifizieren. Es besitzt kein Gedächtnis für frühere Aktionen und variiert die Ergebnisse unendlich.

Im Code muss für das NeuralNetwork und NeuralNetworkGPU jedes Trainingsmuster, Pattern genannt, einen Input und Output besitzen. Die Beiden können entweder ein Array aus Zahlen von 0 bis 1 oder ein Hash von Zahlen im Bereich 0 bis 1 sein.

```
1  const net = new brain.NeuralNetwork();
2  net.train([
3    { input: { r: 0.03, g: 0.7, b: 0.5 }, output: { black: 1 } },
4    { input: { r: 0.16, g: 0.09, b: 0.2 }, output: { white: 1 } },
5    { input: { r: 0.5, g: 0.5, b: 1.0 }, output: { white: 1 } },
6  ]);
7  const output = net.run({ r: 1, g: 0.4, b: 0 }); // { white: 0.99, black: 0.002 }
```

Quellcode 4.18: Brain.js - NeuralNetwork

Im Quellcode 4.18 wird eine Demo für die Untersuchung des Farbkontrastes einer Farbe gezeigt. Dabei wird ein feedforward neuronales Netzwerk durch brain.js erzeugt. Der Input kann dabei variieren. Nicht überall müssen die gleichen Eigenschaft vorhanden sein. In Zeile 1 wird das neuronale Netzwerk erstellt. Trainiert wird es dann in Zeile 2-6, wobei drei Farbkontraste mit dem Output "black" oder "white" hinterlegt sind. Nachfolgend wird anhand eines anderen Farbkontrastes untersucht, zu wie viel Prozent er schwarz oder weiß ist. Die Unterschiede zwischen NeuralNetwork und NeuralNetworkGPU ist die Art des Trainings. Mit dem Zusatz von GPU wird die GPU anstelle der CPU verwendet.

Typ zwei, das Recurrent Neural Network (RNN), hat eine endliche Anzahl von Ergebnissen. Wenn es eine eingebaute Zeitschrittfunktion besitzt, wie es bei RNNTimeStep der Fall ist, kann es auch zukünftige Werte vorhersagen.

Das auf deutsch rückgekoppelte neuronale Netz bildet den multisensorischen und motorischen Teil der Großhirnrinde von Säugetieren, den sog. Neocortex nach. Im Vergleich zum Feedforward-Netz kann es auch zurückkoppeln.

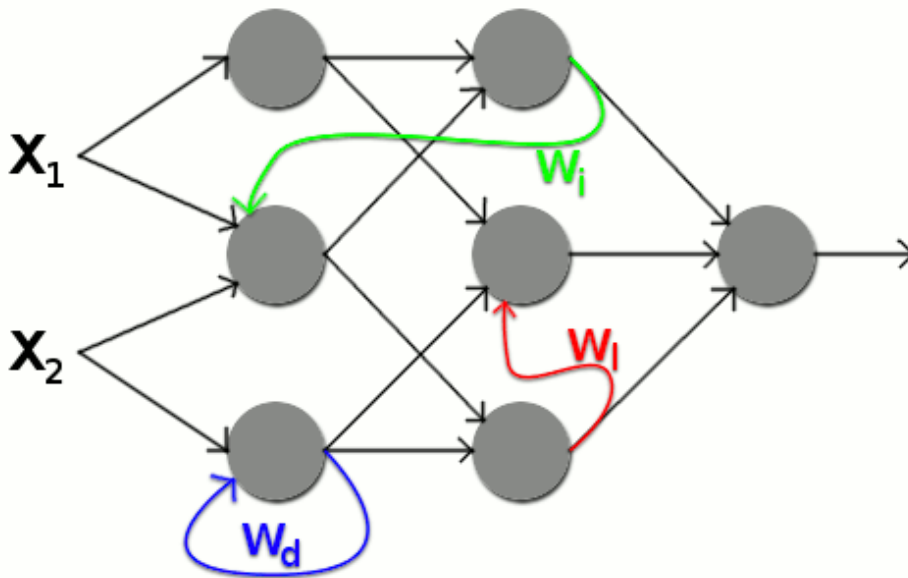


Abbildung 11: Recurrent Neural Network: Rückkopplungsarten

Quelle: Mercyse, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=47449500>

Im Allgemeinen gibt es drei verschiedene Rückkopplungsarten. Die indirekte Rückkopplung (Abbildung 11, Farbe grün) verbindet den Ausgang des Knotens mit einem Knoten der vorhergehenden Schicht. Die direkte Rückkopplung (Abbildung 11, Farbe blau) nutzt den Ausgang des derzeitigen Knotens als weiteren Eingang. Der letzte Typ, die seitliche Rückkopplung (Abbildung 11, Farbe rot) verknüpft den derzeitigen Knoten mit einem Knoten derselben Schicht.

Der dritte Typ, das Long Short Term Memory (LSTM), bedeutet langes Kurzzeitgedächtnis und ist eine Unterkategorie der RNNs die in der Lage sind, langfristige Abhängigkeiten zu lernen. Eingeführt wurden die LSTMs erstmals im Jahre 1997 von Sepp Hochreiter und Jürgen Schmidhuber. Die Idee der beiden besteht darin, drei Tore zu verwenden. Ein Eingangstor (Input Gate), Merk- und Vergesstor (Forget Gate) und ein Ausgangstor (Output Gate). Auf diese Weise ist es dem LSTM möglich im Vergleich zu allgemeinen RNNs Ereignisse an frühere Erfahrungen zu merken. Dadurch entsteht das lange Kurzzeitgedächtnis. [81]

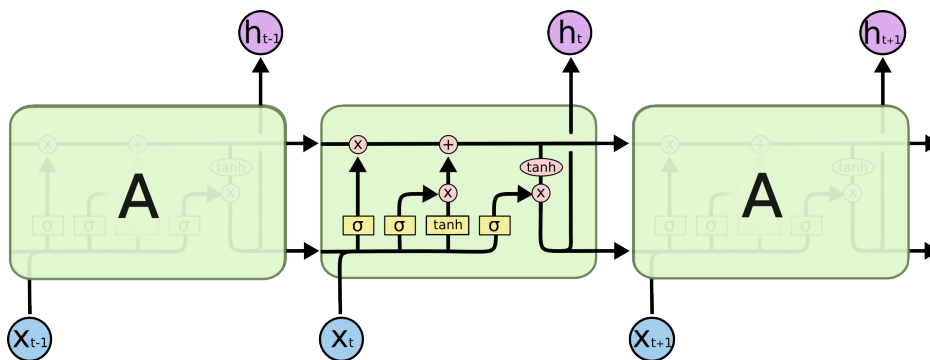


Abbildung 12: Long Short Term Memory Neural Network

Quelle: Christopher Olah, <https://github.com/colah/colah.github.io>

LSTMs haben auch eine kettenartige Struktur der RNNs, aber das sich wiederholende Modul hat eine andere Struktur. Anstelle einer einzelnen neuronalen Netzwerkschicht gibt es vier, die auf ganz besondere Weise interagieren. Der erste Schritt in unserem LSTM besteht darin, zu entscheiden, welche Informationen wir aus dem Zellzustand wegwerfen. Diese Entscheidung wird von einer Forget-Gate-Schicht (Erster gelber Block in Abbildung 12) getroffen. Der nächste Schritt besteht darin, zu entscheiden, welche neuen Informationen wir im Zellenstatus speichern. Dies hat zwei Teile. Zuerst entscheidet die Input-Gate-Schicht, welche Werte wir aktualisieren (2. gelber Block). Als Nächstes erstellt ein Tanh-Layer (3. gelber Block) einen Vektor mit neuen Kandidatenwerten, die dem Zustand hinzugefügt werden könnte. Im nächsten Schritt werden diese beiden kombiniert, um eine Aktualisierung des Status zu erstellen. Schließlich muss die Entscheidung getroffen werden, was genau ausgegeben wird. Diese Ausgabe basiert auf unserem Zellstatus. Hierbei wird die vierte gelbe Schicht ausgeführt, die entscheidet, welche Teile des Zellzustands ausgegeben wird. [82]

Ein solches LSTM kann mit Hilfe von brain.js sehr einfach umgesetzt werden. Hierbei ist kein Vorwissen notwendig.

```
1 const net = new brain.recurrent.LSTMTimestep();  
2 net.train([[1, 2, 3]]);  
3 const output = net.run([1, 2]); // 3
```

Quellcode 4.19: Brain.js - LSTM Neuronales Netzwerk

Im Quellcode 4.19 zu sehen, wird ein LSTM neuronales Netzwerk erstellt und mit den Daten 1, 2 und 3 in Zeile 2 trainiert. In- und Output besteht hierbei immer aus einem Array von Zahlen oder aus einem Array aus Arrays von Zahlen. Der Output konzentriert sich auf die bereits trainierte Zahlenfolge. Somit ist klar, dass mit dem Input [1, 2] der Output 3 folgt. Um die Art des rekurrenten Netzwerkes zu ändern könne man "LSTMTimestep" in Zeile 1 durch RNNTimestep oder GRUTimestep austauschen, um das neuronale Netzwerk in den anderen Typen aufzubauen. [79]

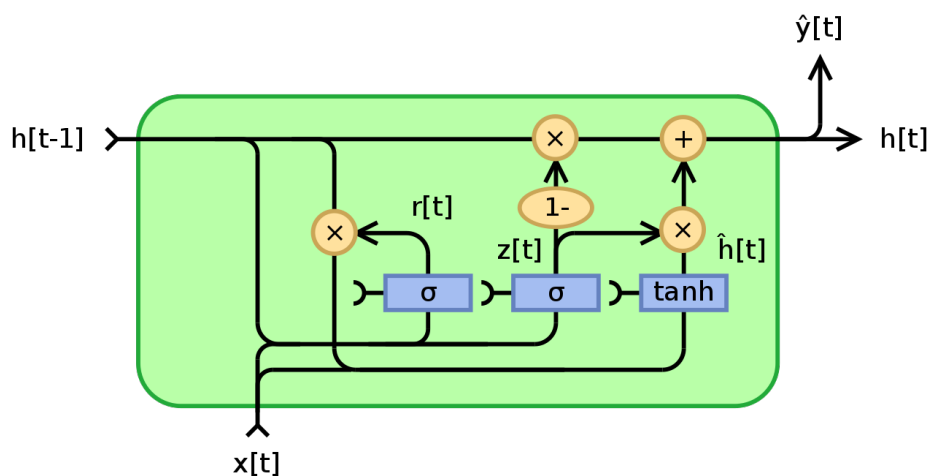


Abbildung 13: Gated Recurrent Unit Neural Network

Quelle: Jeblad, CC BY-SA 4.0, https://en.wikipedia.org/wiki/Gated_recurrent_unit#/media/File:Gated_Recurrent_Unit,_base_type.svg

Der letzte Typ, die Gated Recurrent Unit (GRU), ist eine Erweiterung der erwähnten LSTMs. Im GRU sind weniger Parameter als beim LSTM vorhanden, da ein Ausgangsgate (Zu sehen in Abbildung 13) fehlt [83]. Im Vergleich zum LSTM fehlt hier die vierte Schicht. Es wurde gezeigt, dass GRUs bei bestimmten kleineren und weniger häufigen Datensätzen eine bessere Leistung zeigen. [84]

5 Corona Daten

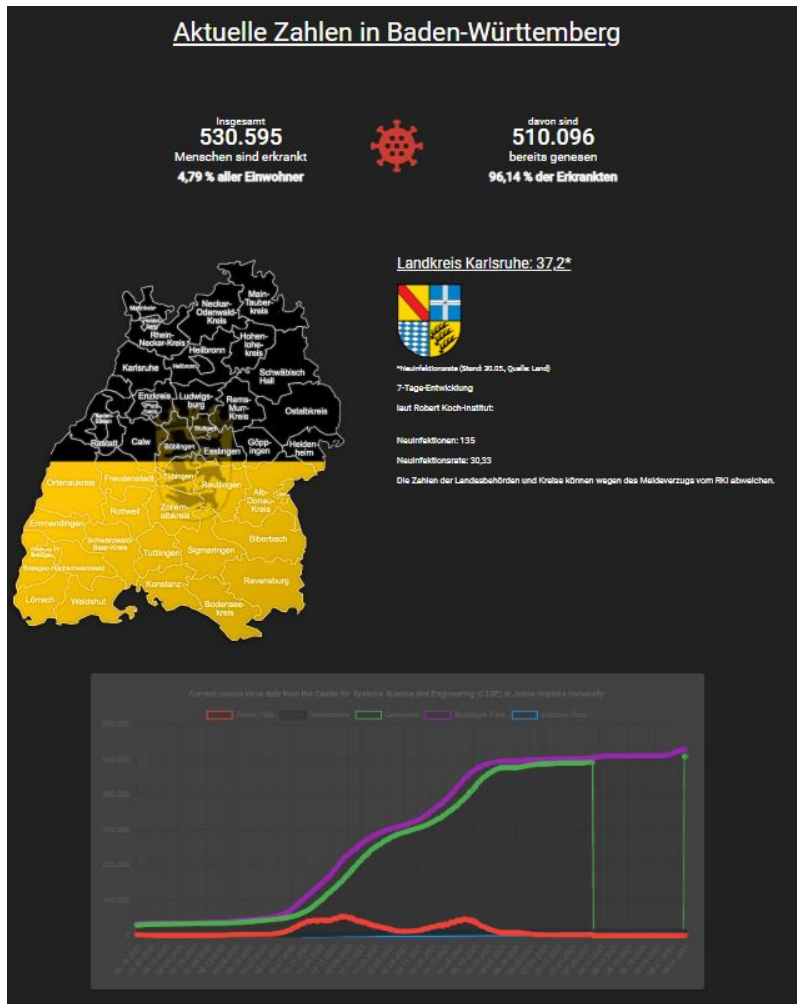


Abbildung 14: Corona-Daten chart.js
Quelle: Eigene Darstellung

Diese Statistik, zu sehen in Abbildung 14, wird durch einen eigenen Endpunkt der REST-API bereitgestellt. Visualisiert wird die Statistik in der Angular-Anwendung mit chart.js.

Die im Jahre 2013 von Nick Downie veröffentlichte Open-Source JavaScript-Bibliothek dient zur Visualisierung von Daten. [85]

Chart.js verfügt über acht verschiedene Diagramme. Darunter sind Linien-, Balken-, Donut-, Kreis-, Blasen-, Streu- und Flächendiagramm. Chart.js rendert in HTML5 Canvas und wird weithin als eine der besten Datenvisualisierungsbibliotheken angesehen. [86]

6 Fazit und Ausblick

Mit dieser wissenschaftlichen Arbeit sollte ein Bürgerportal mit intelligenter Suche konzeptioniert und umgesetzt werden. Mithilfe einer detaillierten Untersuchung der derzeitigen vorhandenen Technologien zur Entwicklung einer solchen Web-Applikation wurde dieses veranschaulicht, diskutiert und verwirklicht.

Die heutzutage frei verfügbaren Mittel zur Entwicklung einer Web-Applikation sind sehr groß. Trotz des großen Lernaspekts von Programmiersprachen im Allgemeinen, ist es möglich mit dem Wissen der aktuellen Zeit, auf viele verschiedene Frameworks zurückzugreifen, die nicht selten Ähnlichkeiten aufweisen.

Durch Auswahl des MEAN-Stacks war es möglich, das Bürgerportal modularisiert zu entwickeln. Somit können Entwickler*innen, die beispielsweise nur Erfahrung in einem Gebiet der verwendeten Technologien (MongoDB, Express.js, Angular, Node.js und Brain.js) haben, daran in Zukunft weiter forschen.

Die Entwicklung einer komplett funktionierenden künstlichen Intelligenz, die beispielsweise den Turing-Test besteht, ist eine immer intensiv werdende noch nicht ganz erforschte Thematik. Besonders die Dartmouth Conference hat die Untersuchungen zur Entstehung einer künstlichen Intelligenz in den letzten 50 Jahren angestoßen. Mit neuen Einsichten über automatische Computer und Neuronale Netzwerke haben sie dabei eine neue Perspektive eingenommen. Gerade die Anzahl der vorhandenen Gebiete, in der eine KI zum Einsatz kommt, ist sehr vielfältig.

Durch die spezifische Betrachtung der Beziehung zwischen den Technologien untereinander haben die Ergebnisse dieser Arbeit neue Erkenntnisse über künstliche Intelligenz und über die Entwicklung einer Rich Internet Application (RIA) aufgezeigt.

Die vorliegende Arbeit betrachtet ausschließlich die Entwicklung eines Bürgerportals mit integrierter intelligenter Suche. Daher kann es sinnvoll sein, die Zusammenhänge und Wichtigkeit der künstlichen Intelligenz auch in anderen Projekten auszuweiten.

Literaturverzeichnis

- [1] S. J. Schmidt, *Handbuch Werbung*. Lit. Verlag Münster, 2004, S. 163, ISBN: 9783825875404.
- [2] D. M. Brown, *Communicating design: developing web site documentation for design and planning*. New Riders, 2010, ISBN: 9780321712462.
- [3] J. J. Garrett, *The elements of user experience: user-centered design for the web and beyond*. Pearson Education, 2010.
- [4] S. Morandi, S. Niclasse und C. Turttschi, "Online-Styleguide," Diss., HSR Hochschule für Technik Rapperswil, 2011, S. 14.
- [5] Institut für Demoskopie Allensbach (IfD), *Welche ist Ihre Lieblingsfarbe?* 2002. Adresse: <https://de.statista.com/statistik/daten/studie/50/umfrage/eigene-liebblingsfarbe/> (besucht am 01.06.2021).
- [6] C. Robertson, *The New Roboto*, 16. Juli 2014. Adresse: <https://developers.googleblog.com/2014/07/the-new-roboto.html> (besucht am 28.05.2021).
- [7] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. Prentice Hall, 2002, S. 149–153, ISBN: 0135974445.
- [8] B. Meyer, *Object-oriented Software Construction*. Prentice Hall, 1988, S. 57–61.
- [9] B. H. Liskov und J. M. Wing, *Family Values: A Behavioral Notion of Subtyping*. Pittsburgh, 16. Juli 1993.
- [10] J. M. W. Barbara H. Liskov, *Behavioral Subtyping Using Invariants and Constraints*. Pittsburgh, 1999.
- [11] R. C. Martin, *The Interface Segregation Principle*. Object Mentor, 1996.
- [12] —, *Object Oriented Design Quality Metrics. an analysis of dependencies*. 1994.
- [13] —, *The Dependency Inversion Principle*. Object Mentor, 1996.
- [14] S. Koch, *JavaScript: Einführung, Programmierung und Referenz-inklusive Ajax*. dpunkt. Verlag, 2007, ISBN: 9783898645942.

- [15] D. Crockford, *Das Beste an JavaScript*. O'Reilly Germany, 2008, S. 2–4, ISBN: 9783897218765.
- [16] J. Walke und J. Kyle, *FaxJS*, *GitHub repository*, <https://github.com/jordwalke/FaxJs>, GitHub, 23. Mai 2017. (besucht am 16.01.2021).
- [17] C. Gackenheim, *Introduction to React*. Apress, 2015, S. 1–3, ISBN: 9781484212455.
- [18] A. Banks und E. Porcello, *Learning React: functional web development with React and Redux*. O'Reilly Media, Inc., 2017, S. 178–182, ISBN: 9781491954621.
- [19] Angular Team, *Angular Contributors, Building For the Future*, 2021. Adresse: <https://angular.io/about?group=Angular> (besucht am 12.04.2021).
- [20] A. Kushnir, *v12.2.1*, *GitHub release*, <https://github.com/angular/angular/releases/tag/12.2.1>, GitHub, 11. Aug. 2021. (besucht am 18.08.2021).
- [21] Angular Team, *Angular, The modern web developer's platform*, 2021. Adresse: <https://angular.io/> (besucht am 12.04.2021).
- [22] Y. Fain und A. Moiseev, *Angular 2 Development with TypeScript*. Manning Publications Company, 2017, S. 9, ISBN: 978-1-78588-207-4.
- [23] G. Bierman, M. Abadi und M. Torgersen, “Understanding typescript,” in *European Conference on Object-Oriented Programming*, Springer, 2014, S. 257–281.
- [24] C. Nance, *TypeScript Essentials*. Packt Publishing Ltd, 2014, S. 16–18.
- [25] F. Malcher, J. Hoppe und D. Koppenhagen, *Angular: Grundlagen, fortgeschrittene Themen und Best Practices – inklusive NativeScript und NgRx (iX Edition)*. dpunkt. Verlag, 2017, ISBN: 9783864906466.
- [26] E. You, *What is Vue.js? Website*, 2021. Adresse: <https://vuejs.org/v2/guide/#What-is-Vue-js> (besucht am 26.05.2021).
- [27] —, *Comparison with Other Frameworks, Website*, 2021. Adresse: <https://vuejs.org/v2/guide/comparison.html> (besucht am 26.05.2021).
- [28] E. Hanchett und B. Listwon, *Vue.js in Action*. Simon und Schuster, 2018, ISBN: 9781638356097.

- [29] Facebook Inc., *React*, *GitHub repository*, <https://github.com/facebook/react>, GitHub, 18. Aug. 2021. (besucht am 18.08.2021).
- [30] Angular Team, *Angular*, *The modern web developer's platform*. <https://github.com/angular/angular>, GitHub, 18. Aug. 2021. (besucht am 18.08.2021).
- [31] E. You, *Vue*, *GitHub repository*, <https://github.com/vuejs/vue>, GitHub, 18. Aug. 2021. (besucht am 18.08.2021).
- [32] A. Hodges, "Alan Turing and the Turing test," in *Parsing the Turing Test*, Springer, 2009, S. 13–22.
- [33] J. McCarthy, "Artificial intelligence, logic and formalizing common sense," in *Philosophical logic and artificial intelligence*, Springer, 1989, S. 161–190.
- [34] Bundesministerium für Wirtschaft und Energie, "Einsatz von Künstlicher Intelligenz in der Deutschen Wirtschaft, Website," 30. März 2020. Adresse: <https://www.bmwi.de/Redaktion/DE/Publikationen/Wirtschaft/einsatz-von-ki-deutsche-wirtschaft.html> (besucht am 04. 12. 2020).
- [35] Google LLC, *Google Trends Statistik für künstliche Intelligenz*, Website, 2021. Adresse: <https://trends.google.de/trends/explore?date=all&geo=DE&q=k%C3%BCnstliche%20Intelligenz> (besucht am 27. 12. 2020).
- [36] C. Beierle und G. Kern-Isberner, *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. Springer-Verlag, 2014.
- [37] K. Mainzer, *Leben als Maschine? Von der Systembiologie zur Robotik und Künstlichen Intelligenz*. Mentis Verlag, 2010, S. 777–780.
- [38] P. E. Hart, D. G. Stork und R. O. Duda, *Pattern classification*. Wiley Hoboken, 2000.
- [39] J. Hawkins und S. Blakeslee, *On Intelligence*. Owl Books, 2005, S. 89, ISBN: 9780805078534.
- [40] A. D. Wissner-Gross und C. E. Freer, "Causal entropic forces," *Physical review letters*, Jg. 110, Nr. 16, 2013.
- [41] T. Bäck, J. N. Kok und G. Rozenberg, *Handbook of Natural Computing*. Heidelberg: Springer, 2012, ISBN: 978-3-540-92909-3.

- [42] N. Nilsson, "The quest for artificial intelligence: A history of ideas and achievements, The Quest for Artificial Intelligence: A History of Ideas and Achievements.," CBO9780511819346, 2011.
- [43] N. Bostrom, "Superintelligenz: Szenarien einer kommenden Revolution. 3," *Aufl. Berlin: Suhrkamp*, S. 42, 2018.
- [44] G. Oppy und D. Dowe, "The turing test," 2003.
- [45] Netz-Depeschen, "Geschnatter um Authentizität, Website," *Süddeutsche Zeitung*, 16. Aug. 2011. Adresse: <https://www.sueddeutsche.de/kultur/netz-depeschen-geschnatter-um-authentizitaet-1.1131504> (besucht am 13.02.2021).
- [46] J. Weizenbaum, *ELIZA - a computer program for the study of natural language communication between man and machine*. ACM New York, NY, USA, 1966, S. 36–45.
- [47] S. Chacon und J. Long, *Git: About, Website*, 2021. Adresse: <https://git-scm.com/about> (besucht am 10.01.2021).
- [48] K. Finley, *The Problem With Putting All the World's Code in GitHub, Website*, 29. Juni 2015. Adresse: <https://www.wired.com/2015/06/problem-putting-worlds-code-github/> (besucht am 13.01.2021).
- [49] D. J. Anderson, *Kanban: Evolutionäres Change Management für IT-Organisationen*. dpunkt. verlag, 2011.
- [50] J. Long, *What is Prettier? Website*, 2021. Adresse: <https://prettier.io/docs/en/> (besucht am 03.04.2021).
- [51] —, *Prettier Options, Website*, 2021. Adresse: <https://prettier.io/docs/en/options.html> (besucht am 05.04.2021).
- [52] —, *Prettier Options - Print Width, Website*, 2021. Adresse: <https://prettier.io/docs/en/options.html#print-width> (besucht am 06.04.2021).
- [53] Google LLC, *Introduction to Angular concepts, Website*, 2021. Adresse: <https://angular.io/guide/architecture> (besucht am 10.07.2021).
- [54] —, *Introduction to modules, Website*, 2021. Adresse: <https://angular.io/guide/architecture-modules> (besucht am 10.07.2021).

- [55] —, *Introduction to components*, Website, 2021. Adresse: <https://angular.io/guide/architecture-components> (besucht am 11.07.2021).
- [56] —, *Introduction to services*, Website, 2021. Adresse: <https://angular.io/guide/architecture-services> (besucht am 11.07.2021).
- [57] —, *Built-in directives*, 2021. Adresse: <https://angular.io/guide/built-in-directives> (besucht am 13.07.2021).
- [58] N. Weizenbaum, D. Smart und A. Sottile, *Sass*, GitHub repository, <https://github.com/sass/sass>, GitHub, 30. Dez. 2020. (besucht am 19.07.2021).
- [59] Sass team, *@mixin and @include*, Website, 2021. Adresse: <https://sass-lang.com/documentation/at-rules/mixin> (besucht am 21.07.2021).
- [60] O. Combe (ocombe), *@ngx-translate/core*, npm package, npm, 2021. Adresse: <https://www.npmjs.com/package/@ngx-translate/core> (besucht am 23.05.2021).
- [61] A. Swartz, *Markdown*, Website, 19. März 2004. Adresse: <http://www.aaronsw.com/weblog/001189> (besucht am 01.06.2021).
- [62] J.-F. Cere (jfcere), *ngx-markdown*, GitHub repository, <https://github.com/jfcere/ngx-markdown>, GitHub, 22. Mai 2021. (besucht am 02.06.2021).
- [63] L. Verou, Golmote, J. DiGioia und S. Michael, *Prism*, Website, 2021. Adresse: <https://prismjs.com/> (besucht am 03.06.2021).
- [64] J.-F. Cere (jfcere), *ngx-markdown: Usage*, GitHub repository, <https://github.com/jfcere/ngx-markdown#usage>, GitHub, 22. Mai 2021. (besucht am 02.06.2021).
- [65] Google LLC, *Angular Material*, Website, 2021. Adresse: <https://material.angular.io/> (besucht am 15.07.2021).
- [66] —, *Theming Angular Material*, Website, 2021. Adresse: <https://material.angular.io/guide/theming> (besucht am 16.07.2021).
- [67] Angular Team, *Official components for Angular*, GitHub repository, <https://github.com/angular/components>, GitHub, 2021. (besucht am 17.07.2021).

- [68] M. Alabor, A. Joly und M. Weibel, *Architekturkonzepte moderner Web-Applikationen, Bachelorarbeit*, Hochschule für Technik Rapperswil, 14. Juni 2013. Adresse: <http://hsr-ba-ajw-2013.github.io/BA-Dokumentation/thesis.pdf> (besucht am 05. 03. 2021).
- [69] R. Treat (xzilla), *JSend, GitHub repository*, <https://github.com/omniti-labs/jsend>, GitHub, 12. Jan. 2019. (besucht am 12. 04. 2021).
- [70] A. Soni und V. Ranga, "API features individualizing of web services: REST and SOAP," *International Journal of Innovative Technology and Exploring Engineering*, Jg. 8, S. 668, 2019.
- [71] MongoDB, Inc, "Mongodb," Jg. 9, 2014. Adresse: <https://www.mongodb.com/> (besucht am 15. 05. 2021).
- [72] R. R. Parmar und S. Roy, "MongoDB as an efficient graph database: An application of document oriented NOSQL database," in *Data Intensive Computing Applications for Big Data*, IOS Press, 2018, S. 331–358.
- [73] M. Schwarz, *Konzeption und Implementierung eines NoSQL Laborversuches anhand einer MongoDB auf Basis eines Docker Containers, Bachelorarbeit*, Hochschule Offenburg, 2019. Adresse: https://opus.hs-offenburg.de/frontdoor/deliver/index/docId/3411/file/BA_MSchwarz_SS19.pdf (besucht am 21. 05. 2021).
- [74] Auth0, *Introduction to JSON Web Tokens, Website*, 2021. Adresse: <https://jwt.io/introduction> (besucht am 16. 03. 2021).
- [75] OpenJS Foundation, *Express, Website*, 2010. Adresse: <https://expressjs.com/> (besucht am 02. 03. 2021).
- [76] M. U. Raza, R. Plummer, B. Admoni und V. Karpov, *Brain.js: GPU accelerated Neural networks in JavaScript, Website*, 2021. Adresse: <https://brain.js.org/#/>.
- [77] I. Bizău (IonicaBizau), *Why do you not maintain these popular projects anymore? Website*, 13. Okt. 2015. Adresse: <https://github.com/harthur/brain/issues/72> (besucht am 12. 06. 2021).
- [78] —, *Why do you not maintain these popular projects anymore? Website*, 15. Apr. 2016. Adresse: <https://github.com/harthur/brain/issues/72#issuecomment-210477059> (besucht am 12. 06. 2021).

- [79] —, *brain.js*, *GitHub repository*, <https://github.com/BrainJS/brain.js>, GitHub, 12. Jan. 2019. (besucht am 15.06.2021).
- [80] A. Zell, *Simulation neuronaler netze*. Addison-Wesley Bonn, 1994, Bd. 1, S. 73.
- [81] S. Hochreiter und J. Schmidhuber, “Long short-term memory,” *Neural computation*, Jg. 9, Nr. 8, S. 1735–1780, 1997.
- [82] C. Olah, *colah.github.io*, *GitHub repository*, GitHub, 2021. Adresse: <https://github.com/colah/colah.github.io> (besucht am 26.08.2021).
- [83] D. Britz, *Recurrent Neural Network Tutorial, Part 4, Implementing a GRU/LSTM RNN with Python and Theano*, 27. Okt. 2015. Adresse: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/> (besucht am 28.08.2021).
- [84] C. K. Chung Junyoung Gulcehre Caglar und B. Yoshua, *Recurrent Neural Network Tutorial, Part 4, Implementing a GRU/LSTM RNN with Python and Theano*, 2014. Adresse: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/>.
- [85] N. Downie, *Nick Downie on Twitter, Website*, 18. März 2013. Adresse: https://twitter.com/_nnnick/status/313599208387137536 (besucht am 28.07.2021).
- [86] H. Da Rocha, *Learn Chart.js: Create interactive visualizations for the web with chart.js 2*. Packt Publishing Ltd, 2019, ISBN: 9781789342154.

Anhang

Scribble Startseite

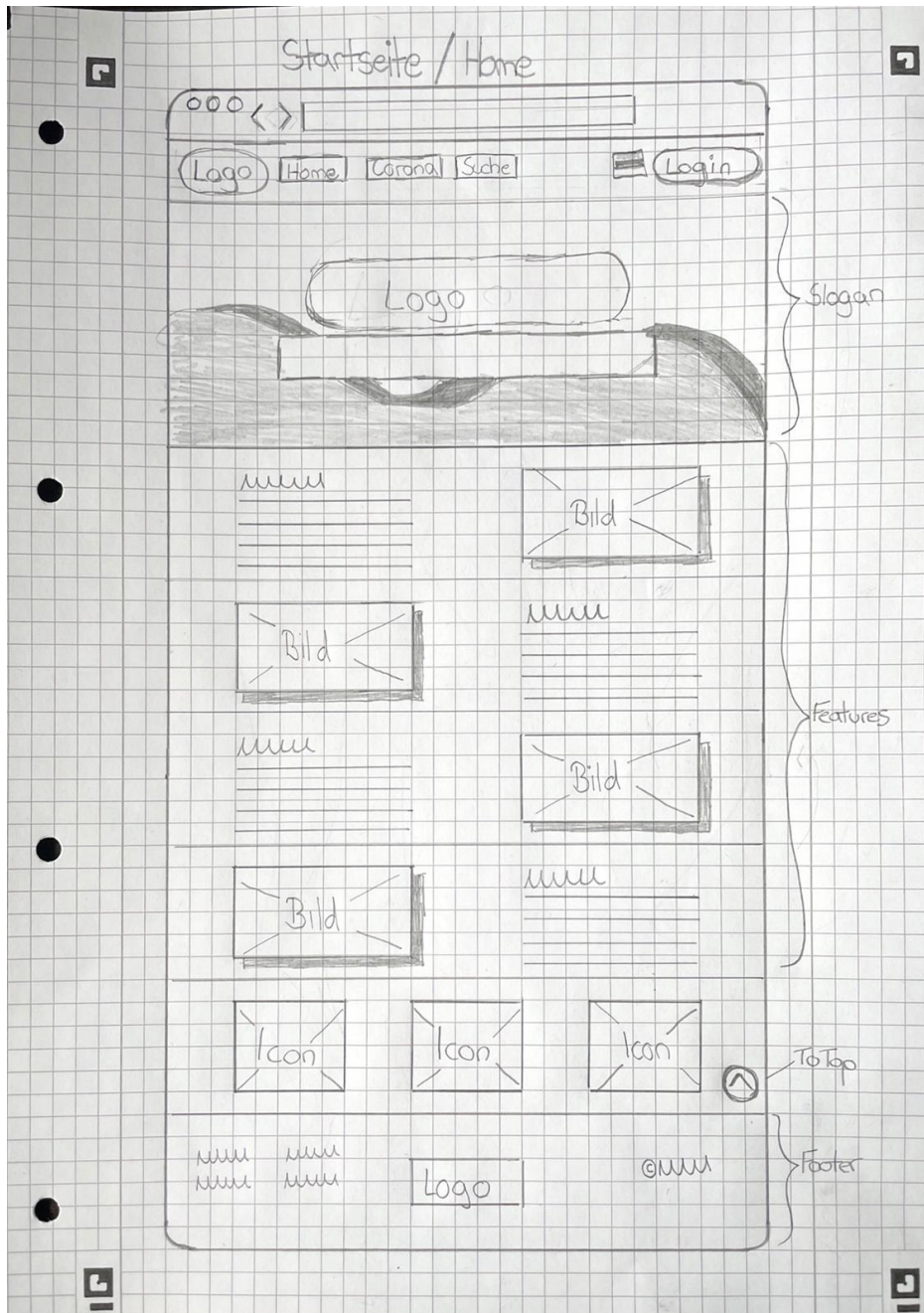


Abbildung 15: Scribble der Startseite
Quelle: Eigene Darstellung

Scribble Corona-Infoseite

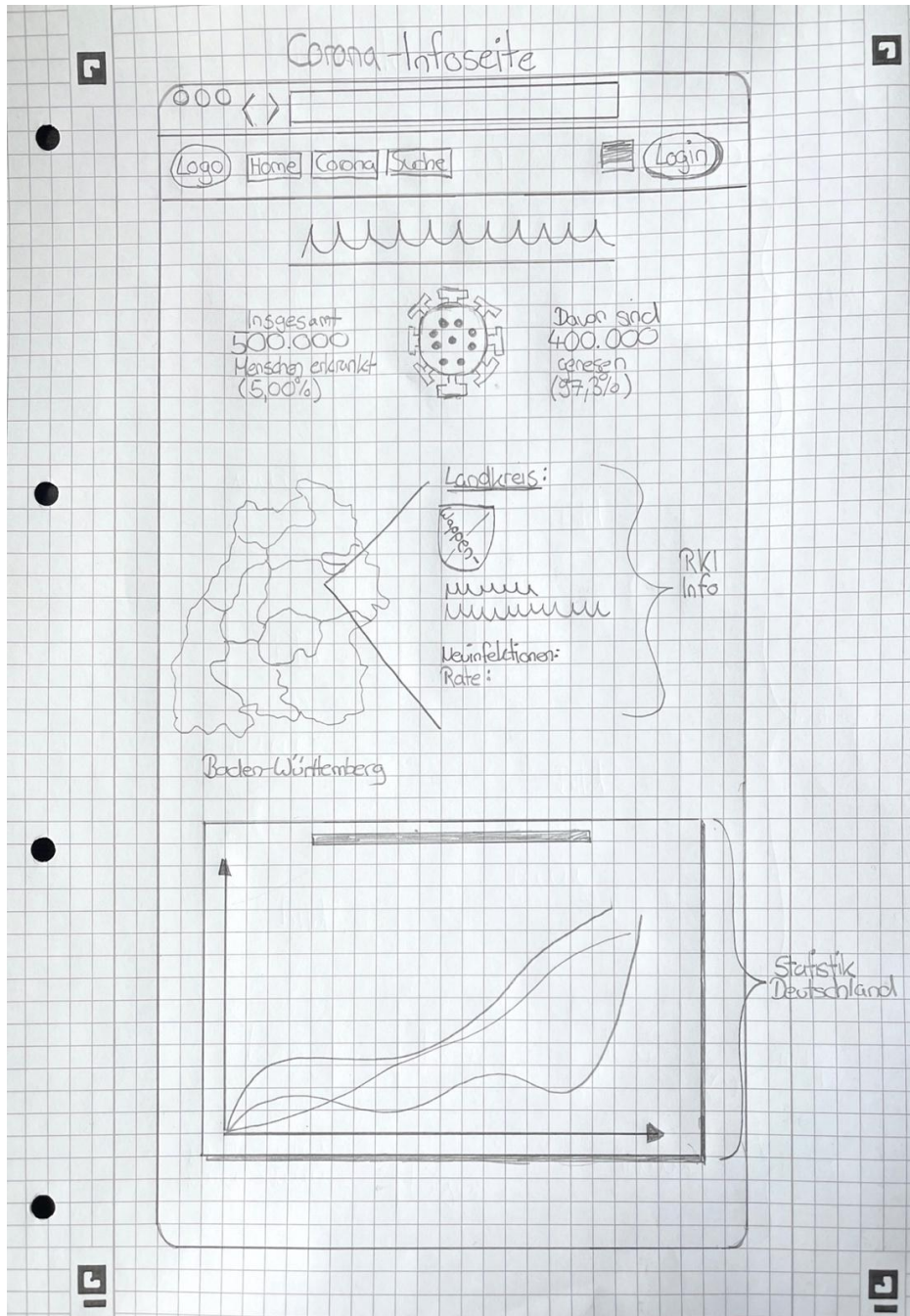


Abbildung 16: Scribble der Corona-Infoseite
Quelle: Eigene Darstellung

Wireframe Startseite

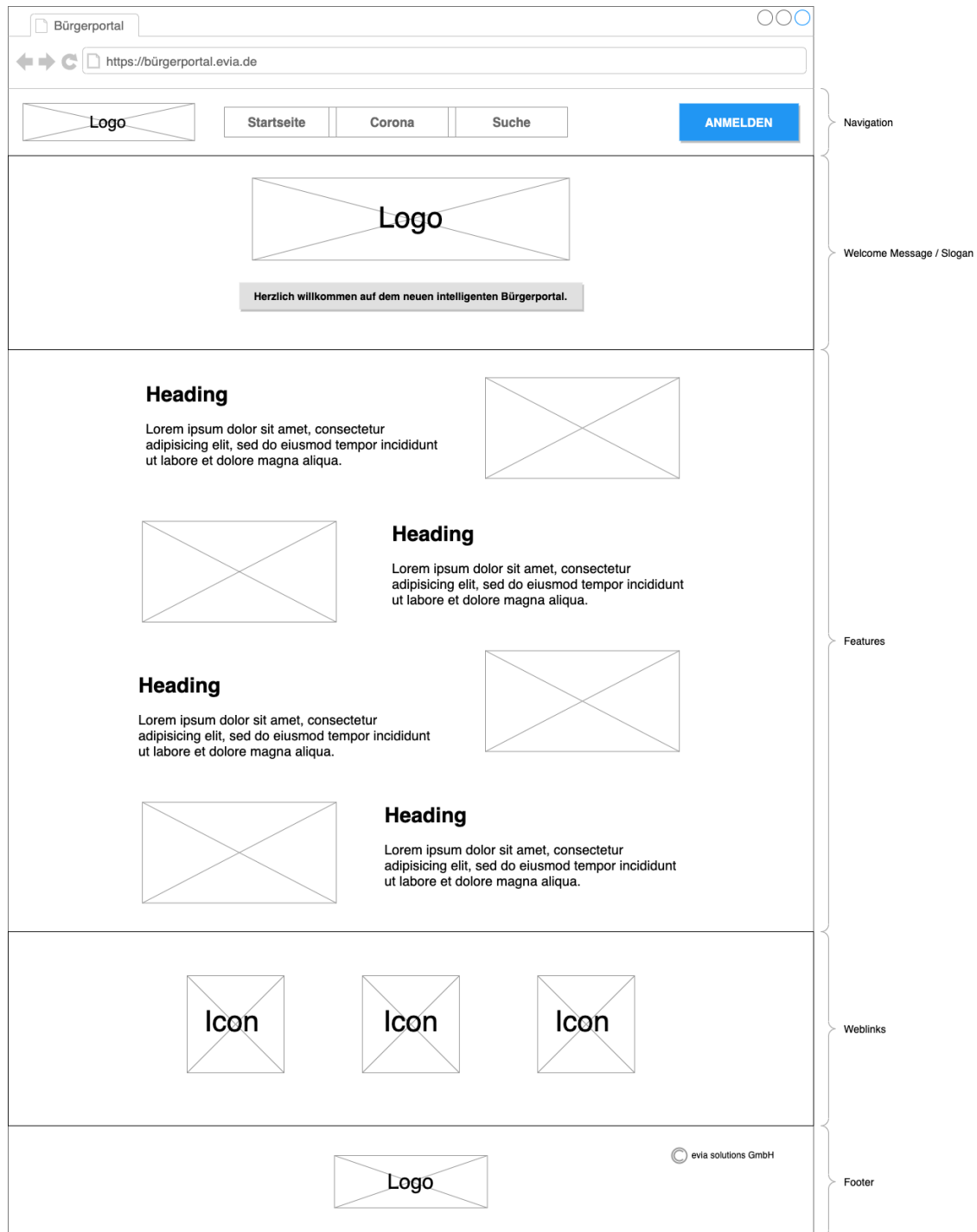


Abbildung 17: Wireframe Startseite
Quelle: Eigene Darstellung

Bürgerportal GUI - Startseite (1/2)

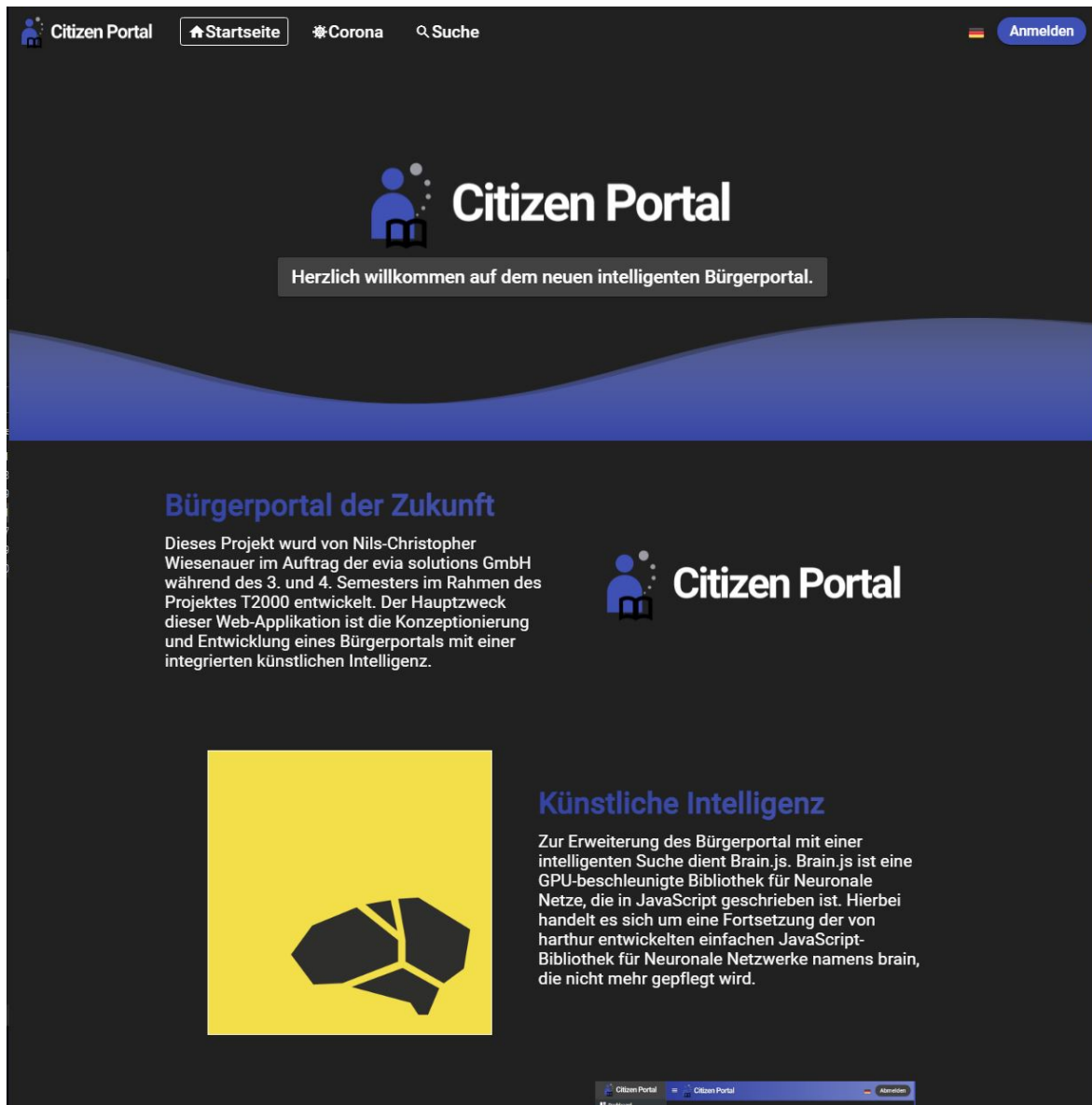


Abbildung 18: Bürgerportal GUI - Startseite (1/2)
Quelle: Eigene Darstellung

Bürgerportal GUI - Startseite (2/2)

Dashboard

Um das Wissen der Künstlichen Intelligenz und somit die Suchergebnisse zu verbessern, können Sie sich ganz einfach einen Account erstellen. Zudem existiert die Möglichkeit eigene Unternehmen zu registrieren und Artikel zu veröffentlichen. Wir wünschen viel Spaß dabei.

Corona

Erhalten Sie Informationen über die aktuellen Fallzahlen sowie Infos und Lockerungen zu den jeweiligen Stadt- und Landkreisen in Baden-Württemberg. Jeden Tag aktualisieren wir unsere Daten mit den aktuellen, die uns das Robert-Koch-Institut (RKI) und das Center for System Science and Engineering (CSSE) der John Hopkins Universität liefert.

Unternehmen

Dieses Projekt wurde im Auftrag der evia solutions GmbH entworfen.

GitHub

Das Repository und der dazugehörige Quellcode befindet sich auf GitHub.

Twitter

Informationen und Neuigkeiten zum Projekt finden Sie auf Twitter.

[Startseite](#)
[Anmelden](#)

[Corona](#)
[Suche](#)

[Impressum](#)
[AGB](#)
[Datenschutzklärung](#)

© 2020-2021 evia solutions GmbH

Erstellt von NurNils

Abbildung 19: Bürgerportal GUI - Startseite (2/2)
Quelle: Eigene Darstellung

Bürgerportal GUI - Startseite Mobile Ansicht

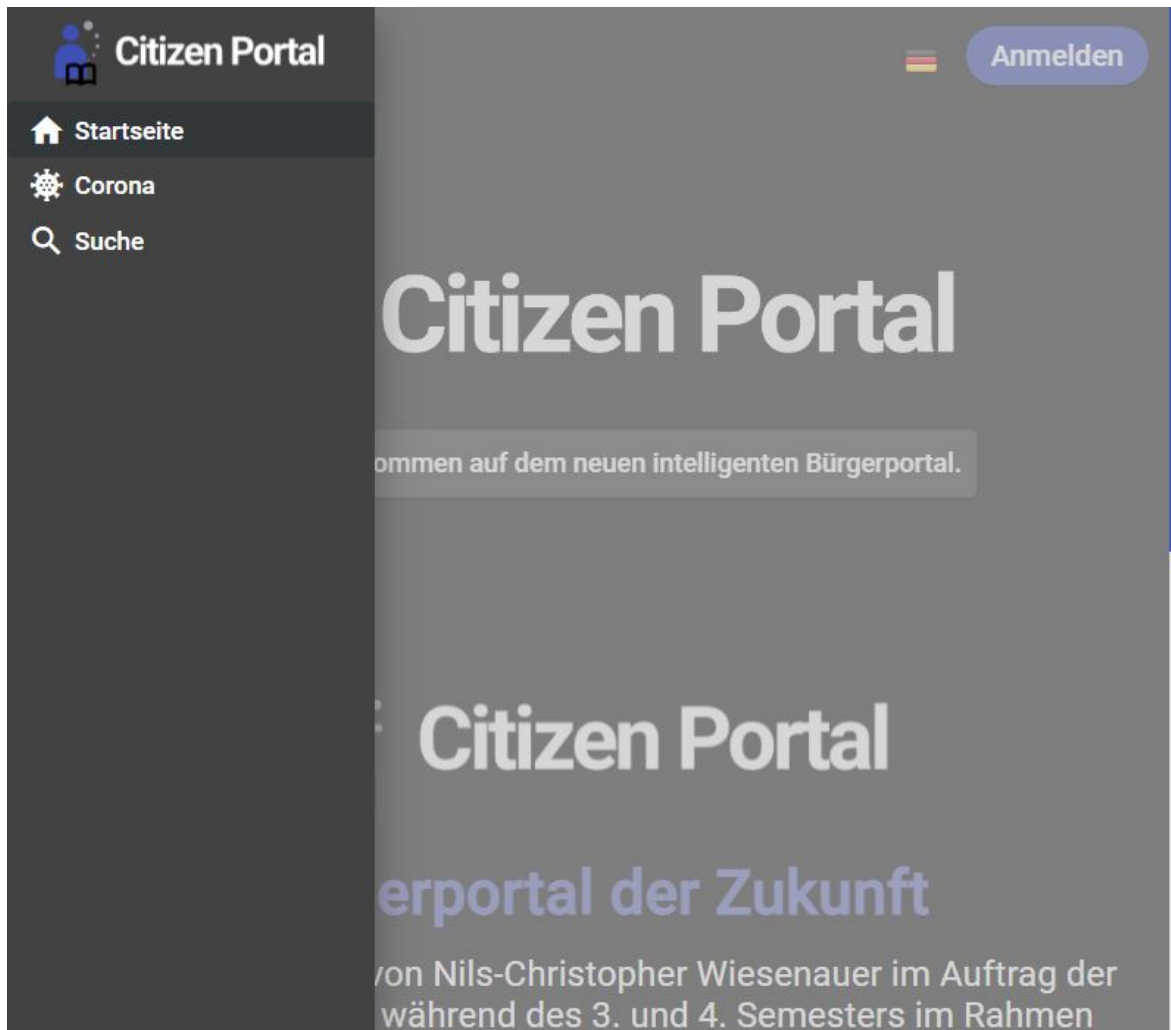


Abbildung 20: Bürgerportal GUI - Mobile Ansicht
Quelle: Eigene Darstellung

Bürgerportal GUI - Corona Unterseite

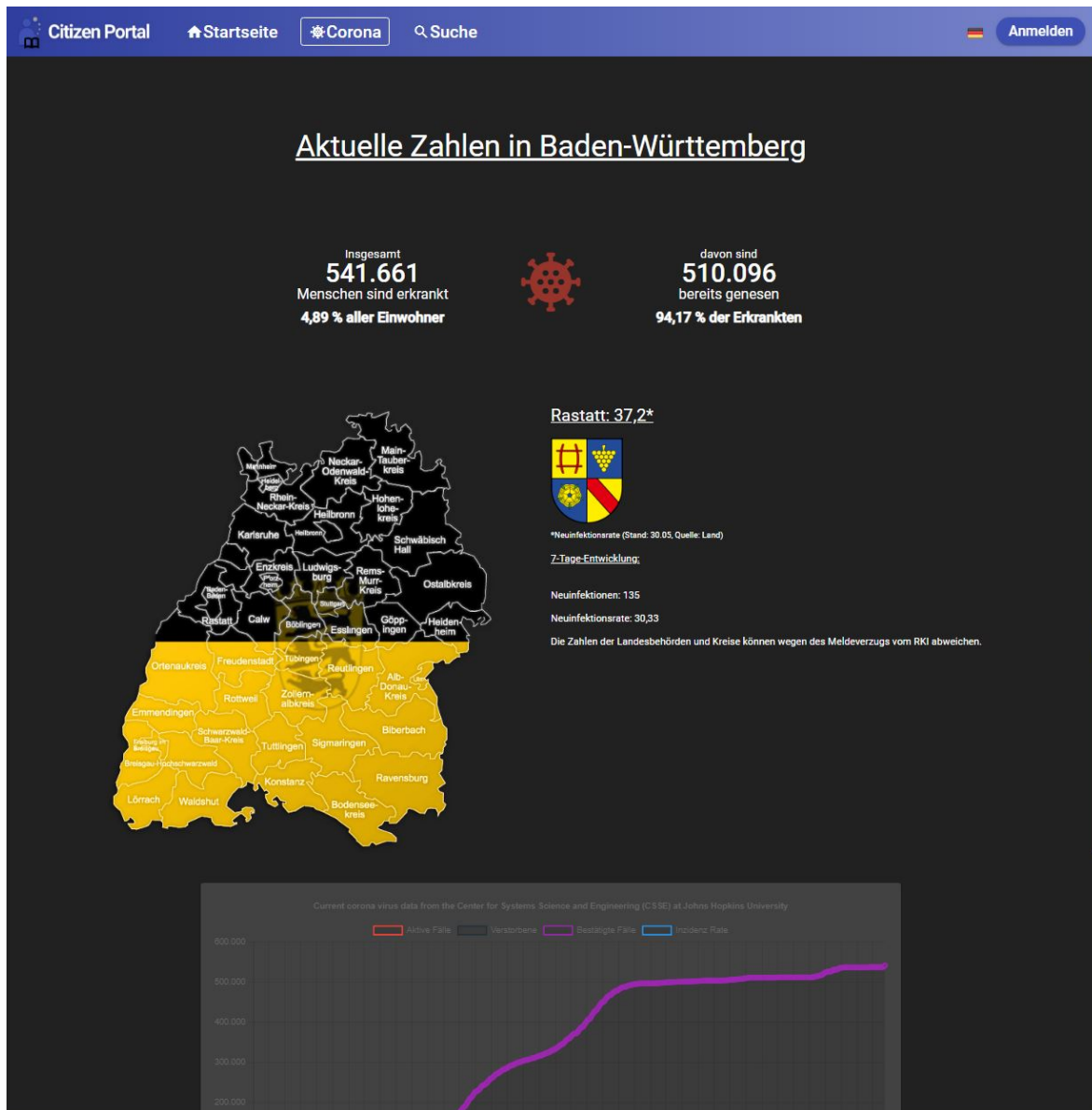


Abbildung 21: Bürgerportal GUI - Corona Unterseite
Quelle: Eigene Darstellung

Bürgerportal GUI - Suche

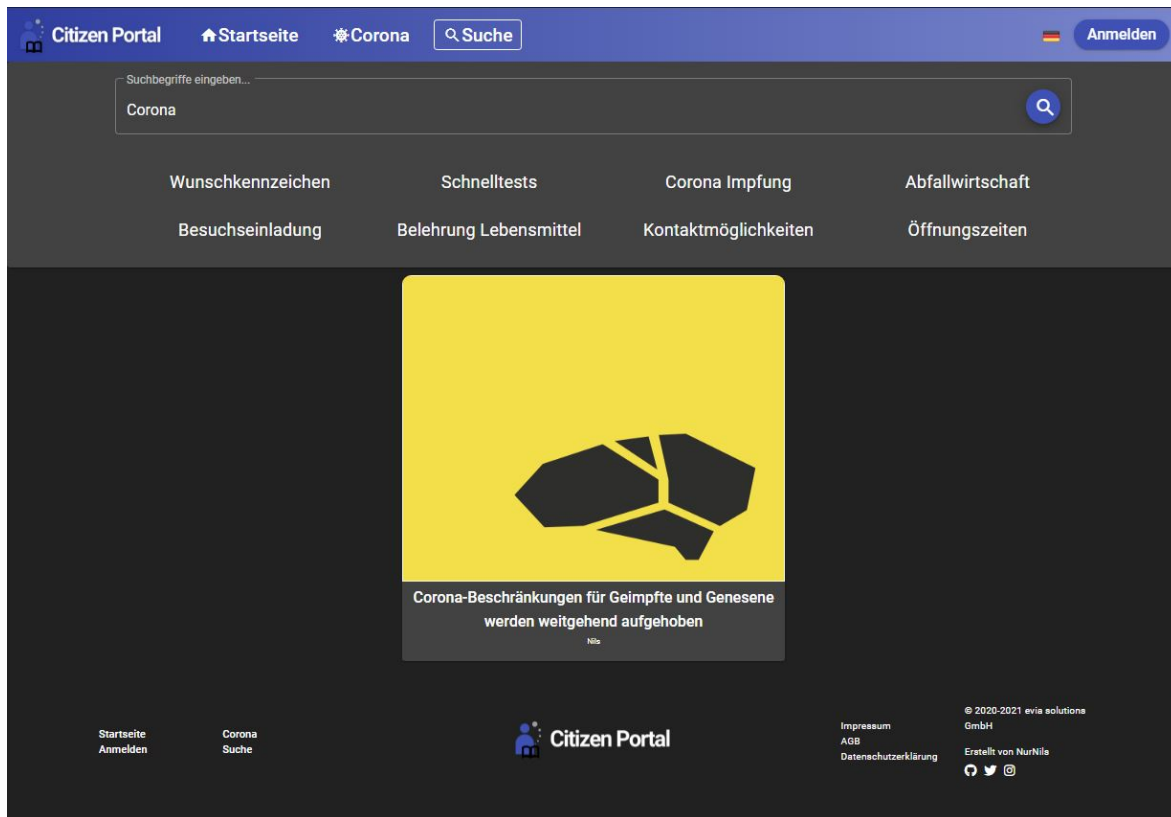


Abbildung 22: Bürgerportal GUI - Suche
Quelle: Eigene Darstellung

Bürgerportal GUI - Artikel

 Citizen Portal

Startseite

Corona

Suche

 Anmelden

CORONA_KFZ

Corona-Beschränkungen für Geimpfte und Genesene werden weitgehend aufgehoben



Baden-Württemberg setzt die Beschlüsse der Ministerpräsidentenkonferenz vom 10. August umgehend um. Ab dem 16. August gelten die neuen Regelungen, die vor allem für geimpfte und genesene Personen die allermeisten Einschränkungen aufheben.

Bund und Länder haben sich am 10. August 2021 darauf geeinigt, die Corona-Beschränkungen zu lockern (PDF). Vor allem für vollständig geimpfte sowie genesene Personen entfallen die allermeisten Beschränkungen. Ebenso entfallen in Baden-Württemberg die bisherigen vier Inzidenzstufen.

Erhalten bleibt für alle jedoch weiter die Maskenpflicht in ihrer jetzigen Form. Das heißt, in geschlossenen Räumen – mit Ausnahme des privaten Bereichs – und im Freien, wenn der Abstand von 1,5 Metern zu anderen Personen nicht dauerhaft eingehalten werden kann, gilt weiterhin die Maskenpflicht. Kinder bis einschließlich fünf Jahre sind auch künftig von der Maskenpflicht befreit. Auch die Abstands- und Hygieneregeln bleiben bestehen. Pflicht bleibt auch die Erfassung der Kontaktdaten.

Infektionsgeschehen wird weiter berücksichtigt

Abbildung 23: Bürgerportal GUI - Artikel
Quelle: Eigene Darstellung

Bürgerportal GUI - Register Form

The screenshot shows the 'Citizen Portal' registration interface. The header bar is blue with the portal logo, navigation links for 'Startseite', 'Corona', and 'Suche', a German flag, and an 'Anmelden' button. The main content area is dark grey and features a 'Registrieren' form. This form includes four input fields: 'E-Mail-Adresse *' (with a 0/100 character count), 'Nutzername *' (with a 0/25 character count), 'Passwort *', and 'Passwort wiederholen *'. Below the fields is a 'Registrieren' button and a link 'Bereits ein Konto? Hier anmelden'. The footer contains links for 'Startseite' and 'Anmelden', 'Corona' and 'Suche', the 'Citizen Portal' logo, and legal links for 'Impressum', 'AGB', and 'Datenschutzerklärung'. It also includes copyright information for 2020-2021 evia solutions GmbH, the creator 'Erstellt von NurNils', and social media icons for GitHub, Twitter, and Instagram.

Abbildung 24: Bürgerportal GUI - Register Form
Quelle: Eigene Darstellung

Bürgerportal GUI - Login Form

The screenshot shows the 'Citizen Portal' login interface. The header bar is identical to the registration page. The main content area is dark grey and features an 'Anmelden' form. This form has two input fields: 'Nutzername oder E-Mail-Adresse *' (containing the text 'Nils' and having a 4/100 character count) and 'Passwort *' (masked with dots). Below the fields is an 'Anmelden' button and a link 'Noch kein Konto? Hier registrieren'. The footer is identical to the registration page, containing navigation links, the portal logo, legal links, copyright information, the creator 'Erstellt von NurNils', and social media icons.

Abbildung 25: Bürgerportal GUI - Login Form
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard

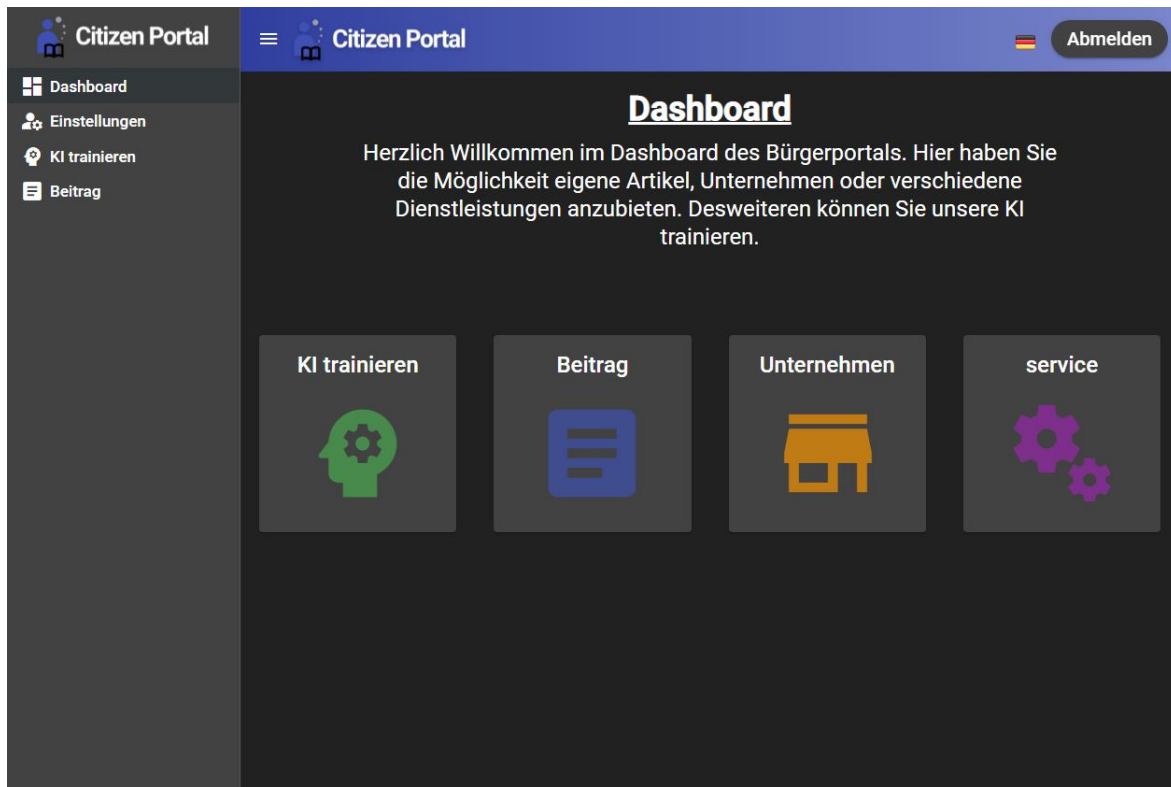


Abbildung 26: Bürgerportal GUI - Dashboard
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Account-Einstellungen

The screenshot displays the 'Citizen Portal' interface with a dark theme. On the left is a sidebar menu with options: 'Dashboard', 'Einstellungen', 'KI trainieren', and 'Beitrag'. The main content area is titled 'Citizen Portal' and includes a German flag and an 'Abmelden' button. Three form sections are visible:

- Benutzername ändern**: Contains a 'Nutzername *' field with the value 'Nils' and a character count '4 / 100'. Below it is a 'Passwort *' field with masked characters. A blue 'Aktualisieren' button is at the bottom.
- E-Mail-Adresse ändern**: Contains an 'E-Mail-Adresse *' field with the value 'Nils' and a character count '4 / 100'. Below it is a 'Passwort *' field with masked characters. A blue 'Aktualisieren' button is at the bottom.
- Passwort ändern**: Contains two 'Passwort *' fields for confirmation. A greyed-out 'Aktualisieren' button is at the bottom.

Abbildung 27: Bürgerportal GUI - Dashboard: Account-Einstellungen
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: KI Training

Citizen Portal

Dashboard

Einstellungen


KI trainieren

Beitrag

Citizen Portal

Abmelden

Insgesamt
2 Kategorien
sind bereits angelegt:
corona, kfz


51
Insgesamte Trainings

davon sind
49
von dir beigetragen

Vielen Dank, dass Sie uns dabei unterstützen, unsere Suche noch effizienter und besser zu gestalten. Unten finden Sie alle Informationen für die Eingabe und den damit resultierenden Trainings für unsere Suche mit künstlicher Intelligenz.

Kategorie wählen
Corona

Suchbegriffe eingeben...
3G Regel

+

_id ↓	Kategorie	Suche	userId	Status
6139e518716575498cc8f5a4	corona	3G Regel	60b4a98caa749a44a06edac2	🔴
61375edb3229cc5c044eb4c6	kfz	gena	60b4a98caa749a44a06edac2	✅
61374dad9e62b132004398ac	corona	12345	60b4a98caa749a44a06edac2	✅
61374da49e62b132004398aa	kfz	test2	60b4a98caa749a44a06edac2	✅
6137304d741a5a5fb4e87ea9	kfz	test	60b4a98caa749a44a06edac2	✅
6135c765a6c6288ca85bc1b3	corona	corna	60b4a98caa749a44a06edac2	✅
60be32261233e96218f82b64	kfz	auto	60b4a98caa749a44a06edac2	✅
60bdb7aa6b45d5308cb2f017	kfz	3 Tages Kennzeichen	60b4a98caa749a44a06edac2	✅
60bdb7506b45d5308cb2f015	corona	Antigentests	60b4a98caa749a44a06edac2	✅
60bdb74e6b45d5308cb2f014	corona	3G Regel	60b4a98caa749a44a06edac2	✅

Training erfolgreich hinzugefügt. Wir werden unsere KI bald damit trainieren. Vielen Dank

ok

Items per page: 10 1 - 10 of 49 < >

Abbildung 28: Bürgerportal GUI - Dashboard: KI Training
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Artikel Übersicht

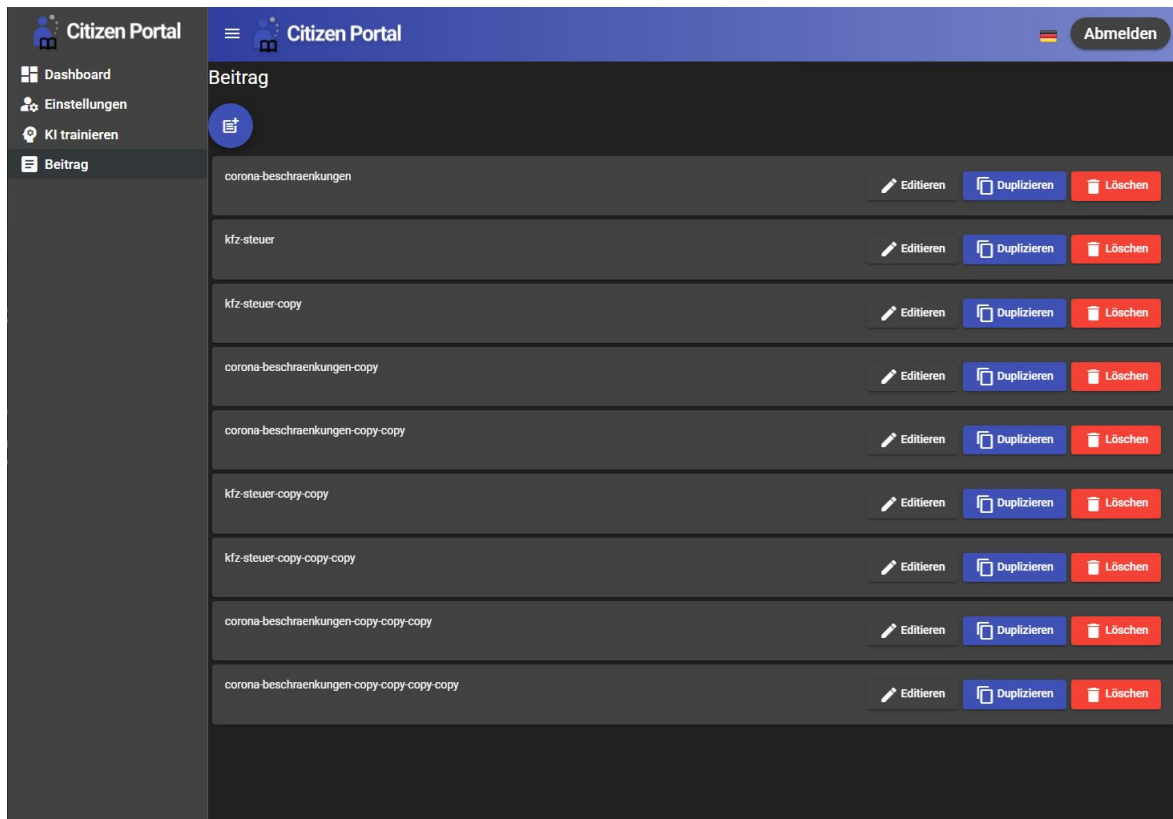


Abbildung 29: Bürgerportal GUI - Dashboard: Artikel Übersicht
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Artikel Edit Dialog

Artikel erstellen/editieren

Bitte gebe die gewünschten Informationen an.

☒ Aktivieren?

Einzigartiger Name *

corona-beschaenkungen

Titel

Corona restrictions for vaccinated and convalescents la

Beschreibung

Beschreibung...

Thumbnail *

<http://localhost:3000/image/6134bc524cdeab5e28b9b>

Kategorie

Corona (+1 weiteres)

Abbrechen Speichern

Abbildung 30: Bürgerportal GUI - Dashboard: Artikel Edit Dialog
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Translate Dialog

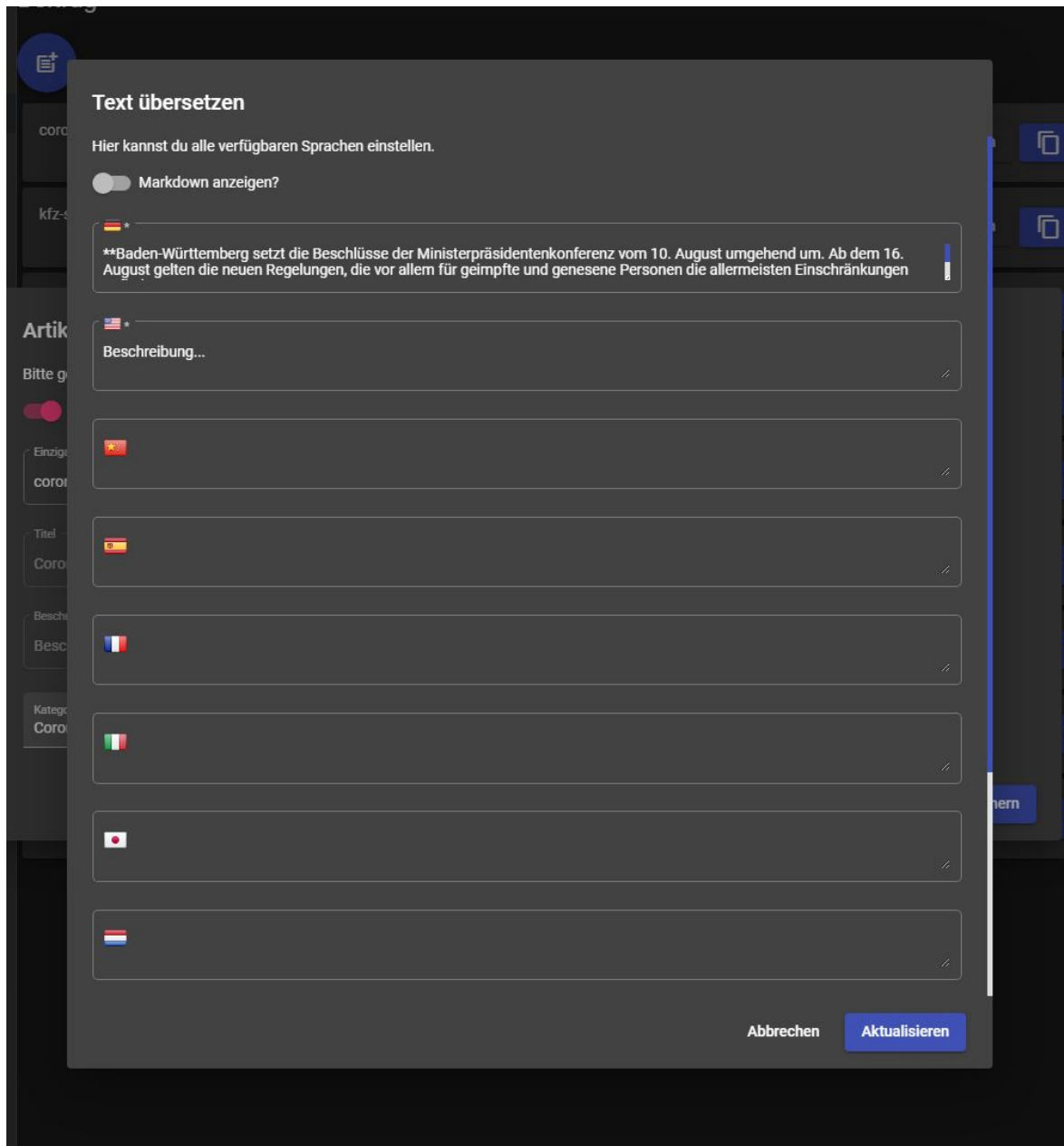


Abbildung 31: Bürgerportal GUI - Dashboard: Translate Dialog
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Translate Dialog mit aktiviertem markdown

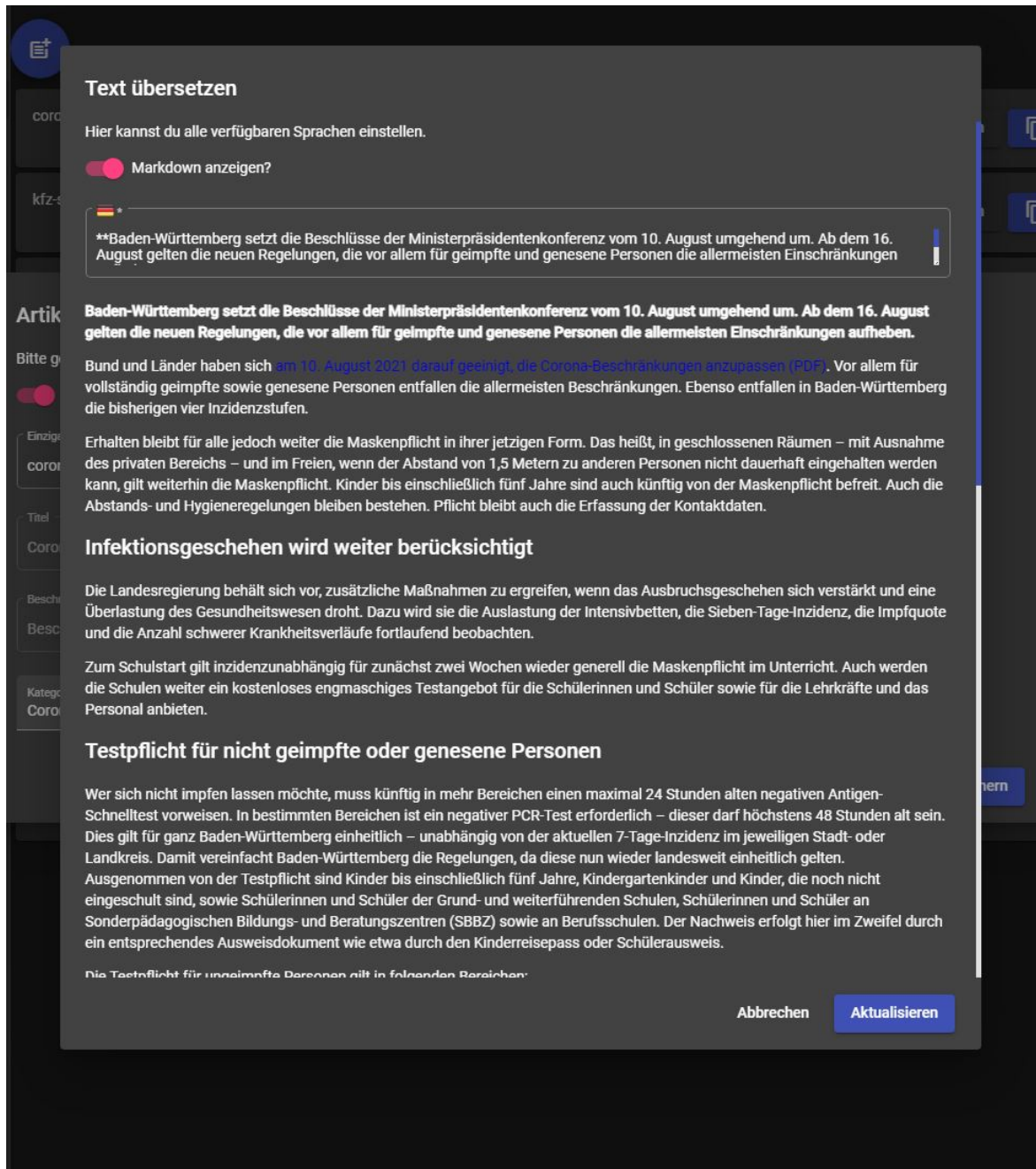


Abbildung 32: Bürgerportal GUI - Dashboard: Translate Dialog (markdown)
Quelle: Eigene Darstellung

Bürgerportal GUI - Dashboard: Bilder Galerie

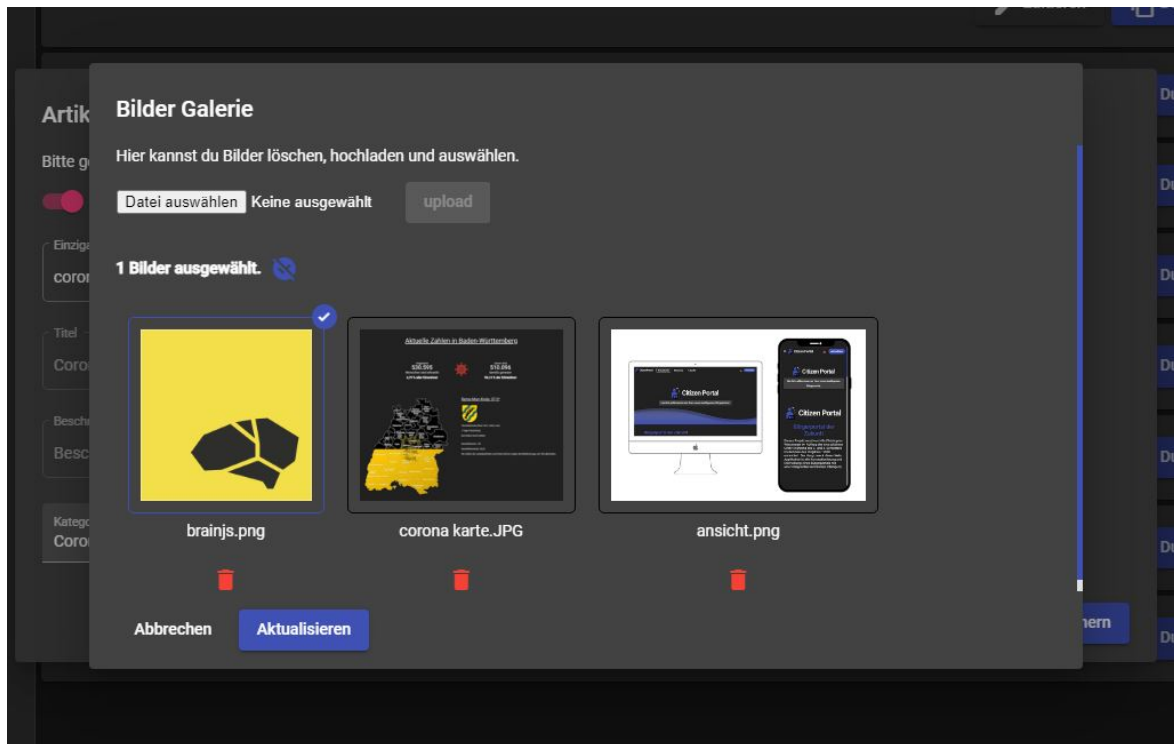


Abbildung 33: Bürgerportal GUI - Dashboard: Bilder Galerie
Quelle: Eigene Darstellung