

# DIPLOMARBEIT

Gesamtprojekt

## Find**ME** – Das Digitale FundBüro

**Relationale vs. NoSQL-Datenbanken: Vergleich der Modelle, Anwendungsfälle und Performance.**

Maximilian Fenusz 6CAIF

Betreuer: Rene Wenz

**Best Practices für modernes User Interface Design (UI): Prinzipien und Strategien für ein nutzerfreundliches UI-Design.**

Mihajlo Ilicin 6CAIF

Betreuer: Rene Wenz

**Datenvisualisierung im Web: Best Practices und Tools zur Darstellung komplexer Datenmengen.**

Dominik Nadrasky 6CAIF

Betreuer: Rene Wenz

**Vergleich klassischer und agiler Projektmanagement-Methoden: Effizienz, Flexibilität und Praxistauglichkeit in modernen IT-Projekten**

Sebastian Stiller 6CAIF

Betreuer: Rene Wenz

**Barrierefreiheit im Webdesign: Techniken und Standards für die Umsetzung barrierefreier Webanwendungen.**

Daniel Wagner 6CAIF

Betreuer: Rene Wenz

---

Abgabevermerk:

Datum: 27.06.2025

übernommen von:

 SPENGERGASSE	<b>HTBLVA Wien V, Spengergasse</b> Aufbaulehrgang für Informatik	<b>Reife- und Diplomprüfung</b>
---	---	-------------------------------------

## EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen Hilfsmittel als die angegebenen benützt habe. Die Stellen, die anderen Werken (gilt ebenso für Werke aus elektronischen Datenbanken oder aus dem Internet) wörtlich oder sinngemäß entnommen sind, habe ich unter Angabe der Quelle und Einhaltung der Regeln wissenschaftlichen Zitierens kenntlich gemacht. Diese Versicherung umfasst auch in der Arbeit verwendete bildliche Darstellungen, Tabellen, Skizzen und Zeichnungen. Für die Erstellung der Arbeit habe ich auch folgende Hilfsmittel generativer KI-Tools

ChatGPT von OpenAI: <https://openai.com/de-DE/chatgpt/overview/>

(z. B. ChatGPT, Grammarly Go, Midjourney) zu folgendem Zweck verwendet:

- Zusammenfassung der Themengebiete
- Strukturierung der Themengebiete
- Fachbegriffserläuterung

Die verwendeten Hilfsmittel wurden vollständig und wahrheitsgetreu inkl. Produktversion und Prompt ausgewiesen.

Wien, am 27.06.2025

Verfasser / Verfasserinnen:

Maximilian Fenusz

Mihajlo Ilicin

Dominik Nadrasky

Sebastian Stiller


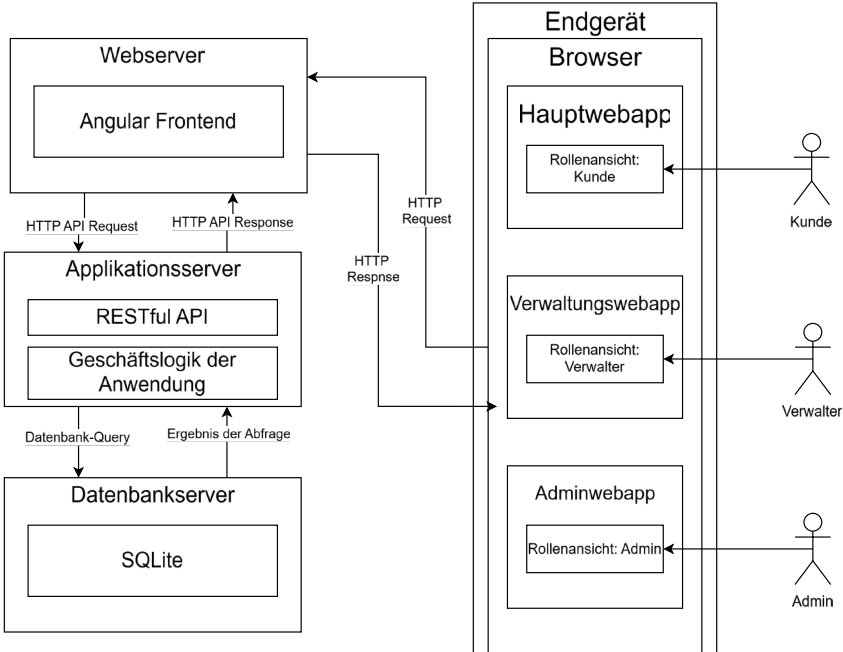
Daniel Wagner


## DIPLOMARBEIT DOKUMENTATION

Namen der Verfasser/innen	Maximilian Fenusz; Mihajlo Ilicin; Dominik Nadrasky; Sebastian Stiller; Daniel Wagner
Jahrgang Schuljahr	6CAIF 2024-2025
Thema der Diplomarbeit	Die Optimierung der derzeitigen Fundsachen Situation an der HTBLVA Spengergasse durch Erstellung und eventueller Implementierung einer Web-basierten EDV-Lösung zur Erfassung und Verwaltung von Fundsachen. Dies soll die Menge an Fundsachen, welche am Ende des Schuljahres übrig bleiben, verringern und eine längerfristige Lösung zur Warenverschwendung bieten. Für eine bessere, grünere Zukunft.
Kooperationspartner	HTBLVA Spengergasse: Spengergasse 20, 1050 Wien

Aufgabenstellung	<p>Die Hauptaufgabe des Teams ist es, eine Webapplikation zu entwickeln, die es Personen ermöglicht, verlorene Gegenstände innerhalb der Schule besser zu finden. Diese Applikation soll den Prozess der Fundsachen Aufnahme zu erleichtern und zu modernisieren.</p> <p>Dabei ergeben sich folgende Fragen:</p> <p><b>Dominik Nadrasky/Maximilian Fenusz</b> Wie wird ein Backend für eine Web-App effizient und funktionsorientiert entwickelt? Welche Frameworks gibt es?</p> <p><b>Daniel Wagner/Mihajlo Ilicin</b> Wie gestalte ich ein User-Interface so, dass große Datenmengen optimal und Übersichtlich dargestellt werden können.</p> <p>Sebastian Stiller</p>
------------------	--

Realisierung	Die Webapplikation „FindMe“ wird in einem modernen Technologie-Stack entwickelt, um die Aufbereitung und Öffentlichmachung zu modernisieren und zu erleichtern. Das Backend wird mit ASP DOTNET (C#) umgesetzt, um eine leistungsfähige und skalierbare RESTful API zu bieten. Für die Datenverwaltung wird eine flexible und effiziente Relationale SQLite Datenbank verwendet.
--------------	--

 SPENGERGASSE	<b>HTBLVA Wien V, Spengergasse</b> <b>Aufbaulehrgang für Informatik</b>	<b>Reife- und Diplomprüfung</b>
	Das Frontend wird mit Angular, einem leistungsstarken TypeScript-Framework, erstellt, das ein dynamisches und reaktionsschnelles Benutzerinterface ermöglicht. Besonderes Augenmerk liegt auf der Verfügbarkeit der Web-App.	
Ergebnisse	<p><b>Dominik Nadrasky/Maximilian Fenusz</b>  Betriebsbereites, fertig implementiertes Backend, welches mit der Datenbank interagieren und Anfragen des Frontend über eine REST-API beantwortet und die Daten vor unbefugtem Zugriff und Manipulation geschützt sind.</p> <p><b>Daniel Wagner/Mihajlo Ilicin</b>  Realisiertes Frontend als Web-App mit einem benutzerfreundlichen User Interface. Welches die Daten in einer geordneten und der Betriebslogik entsprechenden Weise für den Benutzer aufbereitet und Darstellt.</p> <p><b>Sebastian Stiller</b>  Erarbeitung einer Prozedur, welche die Einbindung der Web-App in bereits bestehende Abläufe integriert und die bestehenden Prozeduren modernisiert.</p>	
Typische Grafik, Foto etc. (mit Erläuterung)	<p>Die User verbinden sich auf ihrem Endgeräten (z.B. Computer, Laptop etc.) über einen Browser mit der Webapp. Der Webserver stellt die Webapp zur Verfügung und verwaltet die Kommunikation zwischen dem Browser und dem übrigen Teil der Anwendung. Der Applikationsserver übernimmt die Geschäftslogik und interagiert mit dem Datenbankserver. Der Datenbankserver speichert die Daten und wird vom Applikationsserver genutzt, um Daten zu lesen, zu schreiben, zu aktualisieren oder zu löschen.</p>  <pre> graph TD     subgraph Endgerät         subgraph Browser             HW[Hauptwebapp]             VW[Verwaltungswebapp]             AW[Adminwebapp]         end         K((Kunde))         V((Verwalter))         A((Admin))     end      subgraph ServerStack         subgraph Webserver             AF[Angular Frontend]         end         subgraph Applikationsserver             RA[RESTful API]             GL[Geschäftslogik der Anwendung]         end         subgraph Datenbankserver             SL[SQLite]         end     end      K --&gt; HW     V --&gt; VW     A --&gt; AW      HW -- "HTTP Request" --&gt; AF     AF -- "HTTP API Response" --&gt; RA     RA -- "HTTP Response" --&gt; HW     VW -- "HTTP Request" --&gt; AF     AF -- "HTTP API Response" --&gt; RA     RA -- "HTTP Response" --&gt; VW     AW -- "HTTP Request" --&gt; AF     AF -- "HTTP API Response" --&gt; RA     RA -- "HTTP Response" --&gt; AW      RA -- "Datenbank-Query" --&gt; SL     SL -- "Ergebnis der Abfrage" --&gt; GL </pre>	
Teilnahme an Wettbewerben, Auszeichnungen	Keine	

 SPENGERGASSE	<b>HTBLVA Wien V, Spengergasse</b> <b>Aufbaulehrgang für Informatik</b>	<b>Reife- und Diplomprüfung</b>
---	--	-------------------------------------

Möglichkeiten der Einsichtnahme in die Arbeit	Bibliothek der HTL-Spengergasse
---	---------------------------------

Approbation (Datum / Unterschrift)	Prüfer/Prüferin  <b>Rene Wenz</b>	Direktor/Direktorin Abteilungsvorstand/Abteilungsvorständin
---------------------------------------	---	--

## DIPLOMA THESIS

### Documentation

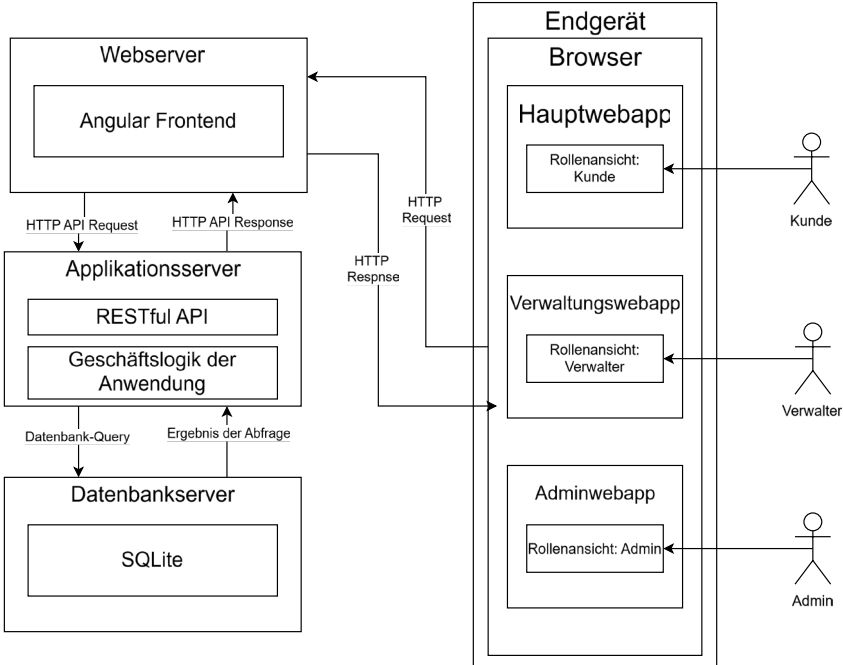
Author(s)	Maximilian Fenusz; Mihajlo Ilicin; Dominik Nadrasky; Sebastian Stiller; Daniel Wagner
Form Academic year	2024-2025
Topic	The optimization of the current lost property situation at the HTBLVA Spengergasse by creating and possibly implementing a web-based IT solution for recording and managing lost property. This should reduce the amount of lost property left over at the end of the school year and provide a longer-term solution to wastage. For a better, greener future.
Co-operation partners	HTBLVA Spengergasse: Spengergasse 20, 1050 Vienna

Assignment of tasks	<p>The main task of the team is to develop a web application that enables people to better find lost items within the school. This application is intended to facilitate and modernize the process of lost property collection.</p> <p>The following questions arise:</p> <p><b>Dominik Nadrasky/Maximilian Fenusz</b> How is a backend for a web app developed in an efficient and function-oriented way? Which frameworks are available?</p> <p><b>Daniel Wagner/Mihajlo Ilicin</b> How do I design a user interface so that large amounts of data can be displayed optimally and clearly?</p> <p><b>Sebastian Stiller</b></p>
---------------------	--

Realisation	<p>The web application "FindMe" is developed in a modern technology stack to modernize and facilitate the preparation and publication. The backend is implemented with ASP DOTNET (C#) to provide a powerful and scalable RESTful API. A flexible and efficient relational SQLite database is used for data management.</p> <p>The frontend is created with Angular, a powerful TypeScript framework that enables a dynamic and responsive user interface. Particular attention is paid to the availability of the web app.</p>
-------------	---

Results	<p><b>Dominik Nadrasky/Maximilian Fenusz</b> Ready-to-use fully implemented backend that interacts with the database and responds to frontend requests via a REST API and protects the data from unauthorized access and manipulation.</p> <p><b>Daniel Wagner/Mihajlo Ilicin</b> Realized frontend as a web app with a user-friendly user interface. Which prepares and displays the data for the user in an organized way that corresponds to the operating logic.</p>
---------	--

	<b>Sebastian Stiller</b> Development of a procedure that integrates the web app into existing processes and modernizes the existing procedures.
--	--

Illustrative graph, photo (incl. explanation)	<p>Users connect to the web app on their end devices (e.g. computer, laptop, etc.) via a browser. The web server makes the web app available and manages the communication between the browser and the rest of the application. The application server handles business logic and interacts with the database server. The database server stores the data and is used by the application server to read, write, update or delete data.</p>  <pre> graph TD     subgraph Endgerät         subgraph Browser             HW[Hauptwebapp]             VW[Verwaltungswebapp]             AW[Adminwebapp]         end         K((Kunde))         V((Verwalter))         A((Admin))     end      subgraph ServerStack         WS[Webserver]         AS[Applikationsserver]         DS[Datenbankserver]     end      subgraph WS_Contents [Webserver]         AF[Angular Frontend]     end      subgraph AS_Contents [Applikationsserver]         RA[RESTful API]         GL[Geschäftslogik der Anwendung]     end      subgraph DS_Contents [Datenbankserver]         SL[SQLite]     end      K --&gt; HW     V --&gt; VW     A --&gt; AW      HW -- "HTTP Request" --&gt; WS     WS -- "HTTP API Response" --&gt; AS     AS -- "Datenbank-Query" --&gt; DS     DS -- "Ergebnis der Abfrage" --&gt; AS     AS -- "HTTP Response" --&gt; WS     WS -- "HTTP Request" --&gt; VW     VW -- "HTTP Response" --&gt; AS     AS -- "HTTP Request" --&gt; AW     AW -- "HTTP Response" --&gt; AS       </pre>
--	---

Participation in competitions Awards	NONE
---	------

Accessibility of diploma thesis	Library of HTL Spengergasse
------------------------------------	-----------------------------

Approval (date / signature)	Examiner  <b>Rene Wenz</b>	Head of College / Department
--------------------------------	----------------------------------	------------------------------





# Inhaltsverzeichnis

Themenstellung von Sebastian Stiller .....	68
Einleitung .....	68
Klassisches Projektmanagement: Das Wasserfallmodell .....	68
Agiles Projektmanagement: Scrum und Kanban .....	74
Vergleich der Methoden .....	78
Fazit, Ausblick und Anwendung im Projekt .....	81
Quellen .....	82

# **Relationales vs. NoSQL-Datenbanken**

Verfasser: Maximilian A. Fenusz

Klasse: 6CAIF

Betreuer: Wenz Rene

Jahrgang: 2024/25

HTBLVA, Wien V, Spengergasse

Höhere Lehranstalt für Informatik

Ausbildungsschwerpunkt Informatik für Erwachsene

Abgabedatum: 27.06.2025

## Inhalt

1.	Einleitung .....	3
1.1.	Projektüberblick.....	3
1.2.	Motivation und Zielsetzung .....	3
1.3.	Technologien im Projekt „FindMe“ .....	4
2.	Theorie.....	5
2.1.	Grundlagen relationaler Datenbanken.....	5
2.2.	Grundlagen von NoSql-Datenbanken.....	6
2.3.	Relationale vs. NoSql-Datenbanken .....	7
2.3.1.	Struktur und Datenmodell.....	7
2.3.2.	Konsistenz- und Transaktionsmodelle .....	7
2.3.3.	Skalierbarkeit und Performance.....	7
2.3.4.	Flexibilität und Einsatzbereiche.....	8
3.	Anwendungsfälle und Einsatzszenarien .....	8
3.1.	Wann sind relationale Datenbanken besser geeignet? .....	8
3.2.	Wann sind NoSql-Datenbanken im Vorteil?.....	9
3.3.	Typische Praxisbeispiele und bekannte Anwendungen.....	9
4.	Praktische Umsetzung.....	10
4.1.	Architekturüberblick .....	10
4.2.	Verwendung von MongoDB im Backend.....	10
4.3.	Integration mit ASP.NET Core .....	10
4.4.	Datenmodell und Datenzugriff.....	11
4.5.	Kommunikation zwischen Frontend und Backend (REST API) .....	11
4.6.	Vor- und Nachteile der NoSQL-Nutzung im Projekt.....	11
5.	Lessons Learned .....	11
5.1.	Herausforderungen während der Entwicklung .....	12
5.2.	Erkenntnisse aus der Arbeit mit MongoDB .....	12
5.3.	Reflexion: Wäre eine relationale Lösung sinnvoll gewesen? .....	12
6.	Fachbegriffe .....	13
6.1.	NoSQL.....	13

6.2.	MongoDB.....	13
6.3.	Relationale Datenbank.....	13
6.4.	ACID.....	13
6.5.	BASE .....	13
6.6.	CRUD .....	13
6.7.	REST API.....	14
7.	Quellen und Literaturverzeichnis .....	14
7.1.	Fachartikel und Online-Ressourcen.....	14
7.2.	Projektspezifische Quellen.....	14

## 1. Einleitung

### 1.1. Projektüberblick

Das Projekt „FindMe“ ist eine digitale Webapplikation, die als modernes Fundbüro für den Gebrauch in der Schule konzipiert wurde. Nutzer können verlorene Gegenstände melden oder gefundene Objekte erfassen. Ziel ist es, eine zentrale Plattform zur Verwaltung und Rückgabe von Fundsachen bereitzustellen, um den Prozess für Schüler\*innen und Schulpersonal effizienter und transparenter zu gestalten.

Die Applikation wurde als Fullstack-Webprojekt umgesetzt:

- Frontend: React in Kombination mit Material UI für ein modernes und responsives Design.
- Backend: ASP.NET Core für die API-Logik und Serverkommunikation.
- Datenbank: MongoDB als NoSQL-Datenbank zur flexiblen Speicherung von Benutzer- und Objektdaten.

Ein zentraler Bestandteil der Arbeit ist der Vergleich von NoSQL- und relationalen Datenbanken, um zu begründen, warum für dieses Projekt auf eine dokumentenbasierte Lösung (MongoDB) gesetzt wurde.

### 1.2. Motivation und Zielsetzung

In vielen Bildungseinrichtungen geht täglich eine Vielzahl von Gegenständen verloren – sei es Kleidung, Schlüssel, USB-Sticks oder technische Geräte. Häufig

werden diese Funde zwar abgegeben, aber selten erfolgreich ihren Besitzer\*innen zugeordnet, da ein strukturierter digitaler Prozess fehlt.

Die Motivation hinter dem Projekt „FindMe“ war es daher, eine benutzerfreundliche Webanwendung zu entwickeln, die das Verwalten und Wiederfinden von verlorenen Gegenständen einfach und zugänglich macht.

Zielsetzung der Arbeit:

- Entwicklung einer funktionalen Webapplikation zur Verwaltung von Fund- und Verlustmeldungen.
- Einsatz moderner Technologien (React, ASP.NET, MongoDB) zur praxisnahen Umsetzung.
- Vergleich von NoSQL- und relationalen Datenbanksystemen, um deren Vor- und Nachteile im Kontext des Projekts zu analysieren.
- Auswahl des optimalen Datenbankmodells für das Szenario anhand realer Anforderungen: Flexibilität, Skalierbarkeit und einfache Integration in ein agiles Frontend.

Diese Ziele spiegeln sowohl die technische Herausforderung als auch den wissenschaftlichen Vergleich der Datenbankmodelle wider.

### 1.3. Technologien im Projekt „FindMe“

Für die Umsetzung des Projekts „FindMe“ wurde auf moderne und bewährte Technologien gesetzt, um eine performante und skalierbare Webapplikation zu entwickeln. Die gewählte Architektur folgt dem klassischen Three-Tier-Modell (Frontend – Backend – Datenbank), was eine klare Trennung von Zuständigkeiten ermöglicht.

Frontend

- React: JavaScript-Bibliothek für die Erstellung dynamischer Benutzeroberflächen.
- Material UI: Design-Framework für ein konsistentes und modernes UI mit Fokus auf Benutzerfreundlichkeit und Barrierefreiheit.
- Axios: Für die Kommunikation mit der REST-API im Backend.

Backend

- ASP.NET Core: Moderne, plattformunabhängige Web-API-Technologie von Microsoft.

- Verwendung des MVC-Prinzips zur Trennung von Logik, Daten und Darstellung.
- Implementierung von REST-Endpunkten für CRUD-Operationen.

### Datenbank

- MongoDB: Dokumentenbasierte NoSQL-Datenbank, die JSON-ähnliche Datenstrukturen (BSON) speichert.
- Besonders geeignet für Anwendungen mit sich dynamisch verändernden Datenstrukturen (z. B. Benutzerprofile, Objekte mit optionalen Eigenschaften).

Diese Technologieauswahl ermöglicht eine flexible, wartbare und zukunftssichere Lösung, die den Anforderungen eines modernen Schulprojekts gerecht wird.

## 2. Theorie

### 2.1. Grundlagen relationaler Datenbanken

Relationale Datenbanken sind das klassische Modell zur strukturierten Datenspeicherung und werden seit den 1970er-Jahren in der IT breit eingesetzt. Sie basieren auf der Idee, Daten in Tabellen (Relationen) zu organisieren, wobei jede Tabelle eine definierte Struktur besitzt.

Wesentliche Merkmale:

- Tabellenstruktur: Daten werden in Zeilen und Spalten gespeichert.
- Feste Schemata: Jede Tabelle hat ein vordefiniertes Datenmodell – z. B. „Benutzer“ mit den Spalten „ID“, „Vorname“, „Nachname“, „Email“.
- Primärschlüssel: Jede Tabelle hat einen eindeutigen Identifikator (z. B. ID), um Datensätze zu unterscheiden.
- Fremdschlüssel: Relationen zwischen Tabellen werden über Verweise hergestellt (z. B. ein Benutzer hat viele Bestellungen).

Vorteile:

- Datenintegrität durch klare Schemata und Schlüssel.
- Transaktionssicherheit (ACID): garantierte Konsistenz bei parallelen Operationen.
- Ausgereifte Tools und Community (z.B. MySQL, PostgreSQL, Oracle)

Nachteile:

- Geringe Flexibilität: Änderungen am Schema sind aufwändig.
- Schwierige horizontale Skalierung.
- Nicht ideal für unstrukturierte oder sich stark ändernde Datenformate.

## 2.2. Grundlagen von NoSql-Datenbanken

NoSQL-Datenbanken („Not only SQL“) sind ein moderner Ansatz zur Datenspeicherung, der sich von den starren Tabellenstrukturen relationaler Systeme löst. Sie wurden entwickelt, um die Anforderungen an Skalierbarkeit, Flexibilität und Performance in modernen Web- und Cloud-Anwendungen zu erfüllen.

Charakteristische Merkmale:

- Schemafrei: Keine feste Tabellenstruktur – jedes „Dokument“ kann andere Felder besitzen.
- Datenformate: Meistens JSON, BSON oder Schlüssel-Wert-Paare.
- Horizontale Skalierbarkeit: Daten werden auf mehrere Server verteilt (Sharding).
- Hohe Verfügbarkeit und Geschwindigkeit, auch bei großen Datenmengen.

Arten von NoSQL-Datenbanken:

- Dokumentbasiert: Speicherung von JSON-ähnlichen Dokumenten (z.B. MongoDB)
- Key-Value-Stores: Einfache Zuordnung von Schlüssel zu Wert (z.B. Redis)
- Spaltenorientiert: Optimiert für Lesezugriffe auf viele Spalten (z.B. Cassandra)
- Graphdatenbanken: Fokus auf Beziehungen zwischen Datenpunkten (z.B. Neo4j)

Vorteile:

- Hohe Flexibilität bei Datenmodellierung.
- Ideal für unstrukturierte oder sich ändernde Daten.
- Leicht skalierbar für große Nutzerzahlen und Datenmengen.
- Schnelle Entwicklungszyklen, da keine Migrationen bei Feldänderungen notwendig sind.

Nachteile:

- Weniger standardisiert als SQL.
- Weniger geeignet für komplexe Transaktionen.
- Eventual Consistency statt starker ACID-Garantien (bei verteilten Systemen).

## 2.3. Relationale vs. NoSql-Datenbanken

Die Wahl zwischen relationalen und NoSQL-Datenbanken hängt stark vom Anwendungsszenario und den Anforderungen der Applikation ab. In diesem Abschnitt werden die wichtigsten Unterschiede anhand verschiedener Kriterien dargestellt.

### 2.3.1. Struktur und Datenmodell

Kriterium	Relationale DB	NoSQL DB
Datenmodell	Tabellen mit festen Schemata	Dokumente, flexibel
Beziehungen	Explizit mit Fremdschlüssel	Implizit oder eingebettet
Änderung am Modell	Aufwendig (Migration Notwendig)	Flexibel, on-the-fly

### 2.3.2. Konsistenz- und Transaktionsmodelle

Kriterium	Relationale DB	NoSQL DB
Konsistenzmodell	ACID	BASE / Eventual Consistency
Transaktionen	Mehrstufige, garantierte	Eingeschränkt oder optional
Datenintegrität	Sehr hoch (Constraints)	Muss manuell umgesetzt werden

### 2.3.3. Skalierbarkeit und Performance

Kriterium	Relationale DB	NoSQL DB
Skalierbarkeit	Vertikal (stärkere Server)	Horizontal (mehr Server)
Schreibzugriffe	Langsamer bei viel Last	Sehr schnell
Lesegeschwindigkeit	Gut mit Indexen	Sehr gut bei flachen Datenstrukturen



### 2.3.4. Flexibilität und Einsatzbereiche

Kriterium	Relationale DB	NoSQL DB
Flexibilität	Niedrig – starres Schema	Hoch – strukturlose Speicherung
Einsatzbereiche	Finanzwesen, ERP, klassische Business-Apps	Web-Apps, IoT, Big Data, Startups
Beispielsysteme	MySQL, PostgreSQL, SQL Server	MongoDB, Redis, Firebase, Cassandra

## 3. Anwendungsfälle und Einsatzszenarien

Die Wahl der richtigen Datenbank hängt nicht nur von theoretischen Vor- und Nachteilen ab, sondern vor allem davon, wofür die Datenbank eingesetzt wird. In diesem Kapitel werden typische Anwendungsfälle beschrieben, bei denen sich relationale oder NoSQL-Datenbanken jeweils besser eignen.

### 3.1. Wann sind relationale Datenbanken besser geeignet?

Relationale Datenbanken sind ideal, wenn:

- Starke Datenkonsistenz und Validierung entscheidend sind (z. B. bei Geldbeträgen, Bestellungen).
- Die Datenstruktur statisch und vorhersehbar ist.
- Komplexe Abfragen (Joins, Aggregationen) benötigt werden.
- Es viele Beziehungen zwischen Entitäten gibt (z. B. Kunden, Bestellungen, Rechnungen).

Typische Einsatzgebiete:

- Banken und Finanzdienstleister
- Buchhaltungssysteme
- Behörden, Registrierungsdatenbanken
- Lagerverwaltungssysteme

## 3.2. Wann sind NoSql-Datenbanken im Vorteil?

NoSQL-Datenbanken glänzen bei:

- Dynamischen Datenstrukturen (z. B. Benutzer mit optionalen Feldern, Objekte mit Bildern, Tags etc.)
- Hohen Datenmengen mit schnellen Schreib-/Lesezugriffen.
- Verteilten Systemen (Cloud, Microservices).
- Agilen Entwicklungsprozessen, bei denen das Datenmodell oft geändert wird.

Typische Einsatzgebiete:

- Webapplikationen mit vielen gleichzeitigen Nutzern
- Social Media Plattformen
- Mobile Apps
- IoT-Systeme (Sensoren, Logs etc.)
- Content-Management-Systeme (CMS)

## 3.3. Typische Praxisbeispiele und bekannte Anwendungen

Szenario	Empfohlene Datenbank	Grund
E-Commerce-System mit Warenkorb	Relationale DB (SQL)	Transaktionen, Lagerstand, Rechnungen
Messaging-App	NoSQL (z. B. MongoDB)	Flexible Nachrichtenstruktur, hohe Performance
Schüler-Fundbüro (FindMe)	NoSQL (MongoDB)	Unterschiedliche Objekttypen, einfache Erweiterung
Personalverwaltung	Relationale DB	Strukturierte Stammdaten, viele Relationen

## 4. Praktische Umsetzung

Im Rahmen der praktischen Arbeit wurde die Webapplikation „FindMe“ als digitales Fundbüro umgesetzt. Das Projekt verfolgt das Ziel, verlorene Gegenstände effizient zu erfassen, sichtbar zu machen und zurückzugeben. Der Fokus lag nicht nur auf der technischen Umsetzung, sondern auch auf der bewussten Auswahl der passenden Datenbanktechnologie – in diesem Fall MongoDB als NoSQL-Datenbank.

### 4.1. Architekturüberblick

Die Applikation basiert auf einer klassischen Client-Server-Architektur:

- Frontend: React + Material UI
- Backend: ASP.NET Core (C#)
- Datenbank: MongoDB

Die Kommunikation erfolgt über eine REST-API, die JSON-Daten zwischen Frontend und Backend austauscht

### 4.2. Verwendung von MongoDB im Backend

- MongoDB wurde verwendet, um die Daten wie Fundobjekte, Benutzer, Nachrichten und Gruppen flexibel zu speichern.
- Es wurden keine festen Tabellen definiert – jede Entität (z. B. Task, User, WorkSlot) ist ein eigenes JSON-Dokument.
- Dank der schemalosen Struktur konnten Felder je nach Objekt erweitert oder weggelassen werden – perfekt für Fundobjekte mit optionalen Eigenschaften (z. B. Farbe, Fundort, Bild, Beschreibung).

### 4.3. Integration mit ASP.NET Core

- Das Backend wurde mithilfe von ASP.NET Core Web API erstellt.
- Die Anbindung an MongoDB erfolgte über den offiziellen MongoDB C# Driver.
- CRUD-Operationen wurden über Services, Controller und Repository-Schichten implementiert

## 4.4. Datenmodell und Datenzugriff

- Es wurden mehrere Entitätsklassen erstellt, z. B. „User“, „Group“, „Task“, „Message“, „Profile“.
- Die meisten Klassen erben von einer BaseEntity, die automatisch eine ID, Erstellungs- und Änderungszeit speichert.
- Die Daten werden als BSON-Dokumente in MongoDB gespeichert, was schnelles Abfragen, Erstellen und Bearbeiten ermöglicht.

## 4.5. Kommunikation zwischen Frontend und Backend (REST API)

- Über HTTP-Endpunkte wie „/api/user“, „/api/task“, „/api/message“ konnten Daten abgerufen und bearbeitet werden.
- Authentifizierung und Autorisierung wurden ebenfalls im Backend umgesetzt.
- Das Frontend verwendet Axios für die Kommunikation mit dem Backend.

## 4.6. Vor- und Nachteile der NoSQL-Nutzung im Projekt

Vorteile:

- Flexibles Datenmodell für heterogene Fundobjekte.
- Schnellere Entwicklung durch fehlende Migrationen.
- Gute Performance auch bei vielen Einträgen.

Nachteile:

- Datenintegrität (z. B. eindeutige Referenzen) muss manuell geprüft werden.
- Keine Transaktionen über mehrere Sammlungen hinweg.
- Weniger strukturierter Überblick bei großen Datenmengen ohne zusätzliches Management-Tool.

## 5. Lessons Learned

Während der Entwicklung von „FindMe“ konnten viele praktische Erfahrungen gesammelt werden – sowohl im Umgang mit modernen Webtechnologien als auch beim direkten Vergleich relationaler und NoSQL-Datenbanksysteme.

## 5.1. Herausforderungen während der Entwicklung

- Die schemalose Struktur von MongoDB war anfangs ungewohnt. Ohne vorgegebenes Datenmodell mussten Validierungen im Code selbst durchgeführt werden.
- Die Trennung von Geschäftslogik und Datenzugriff in ASP.NET erforderte eine saubere Strukturierung (Controller → Services → Repositories).
- Einige Features wie Referenzen zwischen Dokumenten (z. B. User ↔ Tasks) mussten manuell modelliert werden, da MongoDB keine klassischen Fremdschlüssel kennt.

## 5.2. Erkenntnisse aus der Arbeit mit MongoDB

- MongoDB eignet sich hervorragend für Anwendungen mit dynamischen Daten, wie es bei Fundobjekten der Fall ist.
- Die Arbeit mit JSON-Dokumenten war besonders angenehm in Kombination mit React, da beide Seiten (Client/Server) dieselbe Datenstruktur nutzen.
- Die Geschwindigkeit beim Lesen/Schreiben war deutlich höher als in vergleichbaren relationalen Setups – besonders bei Listenoperationen und einfachem Filtern.

## 5.3. Reflexion: Wäre eine relationale Lösung sinnvoll gewesen?

In diesem konkreten Fall: Nein.

Ein relationales System hätte:

- Ein komplexes und starres Datenmodell erfordert.
- Häufige Migrationen bei Änderung von Feldern verursacht.
- Die agile Entwicklung unnötig verlangsamt.

MongoDB war die richtige Wahl für dieses Projekt, weil es maximale Flexibilität, schnelle Entwicklung und gute Performance bei gleichzeitig überschaubarem Datenvolumen ermöglichte.

## 6. Fachbegriffe

### 6.1. NoSQL

Sammelbegriff für Datenbanksysteme, die nicht auf dem relationalen Modell basieren. NoSQL-Datenbanken sind in der Regel schemafrei und besonders gut für unstrukturierte oder sich häufig ändernde Daten geeignet.

### 6.2. MongoDB

Eine weit verbreitete dokumentenorientierte NoSQL-Datenbank, bei der Daten als BSON (Binary JSON) gespeichert werden. Sie bietet hohe Flexibilität und einfache Skalierbarkeit.

### 6.3. Relationale Datenbank

Ein Datenbanksystem, das auf dem Tabellenmodell mit festem Schema basiert. Daten werden in Relationen (Tabellen) gespeichert, die über Schlüssel miteinander verknüpft sind. Bekannte Vertreter: MySQL, PostgreSQL, Oracle.

### 6.4. ACID

Abkürzung für vier Eigenschaften von Transaktionen in relationalen Datenbanken:

- Atomicity (Unteilbarkeit)
- Consistency (Konsistenz)
- Isolation (Isolation paralleler Vorgänge)
- Durability (Dauerhaftigkeit)

### 6.5. BASE

Gegenmodell zu ACID, das oft bei NoSQL-Systemen verwendet wird:

- Basically available
- Asoft state
- Eventual consistency

### 6.6. CRUD

Steht für die vier grundlegenden Operationen auf Daten:

- Create
- Read

- Update
- Delete

## 6.7. REST API

Ein Kommunikationsmodell zwischen Client und Server, bei dem HTTP-Methoden (GET, POST, PUT, DELETE) verwendet werden, um Daten zwischen Frontend und Backend auszutauschen

## 7. Quellen und Literaturverzeichnis

### 7.1. Fachartikel und Online-Ressourcen

1. MongoDB – Offizielle Dokumentation (22.06.2025)  
<https://www.mongodb.com/docs>
2. PostgreSQL – Dokumentation (22.06.2025)  
<https://www.postgresql.org/docs/>
3. Microsoft ASP.NET Core API Doku (22.06.2025)  
<https://learn.microsoft.com/en-us/aspnet/core/>
4. MongoDB vs SQL: Key Differences (22.06.2025)  
<https://www.mongodb.com/nosql-explained/nosql-vs-sql>
5. REST API Grundlagen – Mozilla Developer Network (22.06.2025)  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

### 7.2. Projektspezifische Quellen

6. Eigene Implementierung und Praxiserfahrung im Projekt „FindMe“  
(HTL Spengergasse, Jahrgang 2024/25)
7. Unterrichtsskripten der HTL Spengergasse (Fächer: DBI, POS, WMC, DSAI)

# UI/UX

Verfasser: Mihajlo Ilicin

Klasse: 6CAIF

Betreuer: Wenz Rene

Jahrgang: 2024/25

HTBLVA, Wien V, Spengergasse

Höhere Lehranstalt für Informatik

Ausbildungsschwerpunkt Informatik für Erwachsene

Abgabedatum: 27.06.2025



## Inhalt

1.	Einleitung .....	5
2.	Theorie: UI/UX-Design.....	5
2.1.	Was ist User Experience (UX)? .....	5
2.1.1.	Bestandteile der User Experience .....	6
2.1.2.	Ziele der User Experience.....	6
2.1.3.	UX in schulischen Anwendungen.....	7
2.2.	Was ist User Interface (UI)? .....	7
2.2.1.	Typische Bestandteile eines UI .....	7
2.2.2.	UI-Ziele und Anforderungen.....	7
2.2.3.	UI-Ansatz im Projekt FindMe.....	8
2.3.	Usability – Benutzerfreundlichkeit als Erfolgsfaktor.....	8
2.3.1.	Ziele einer guten Usability.....	9
2.3.2.	Kriterien für hohe Usability .....	9
2.3.3.	Usability im Projekt FindMe .....	9
2.3.4.	Vorteile durch hohe Usability.....	10
2.4.	Informationsarchitektur im UX-Kontext.....	10
2.4.1.	Aufgaben der Informationsarchitektur.....	10
2.4.2.	Strukturprinzipien einer guten IA .....	11
2.4.3.	Informationsarchitektur in FindMe .....	11
2.5.	Konsistenz im Design .....	12
2.5.1.	Arten von Konsistenz .....	12
2.5.2.	Vorteile durch Konsistenz.....	12
2.5.3.	Konsistenz in FindMe .....	13
2.6.	Aktuelle Trends im modernen UI-Design .....	13
2.6.1.	Material Design & Komponentenbibliotheken .....	13
2.6.2.	Responsive Design (Mobile First) .....	14
2.6.3.	Dark Mode & Farbkontraste.....	14
2.6.4.	Microinteractions.....	14
2.6.5.	Minimalismus & Flat Design .....	14

2.6.6.	Barrierefreiheit & Inclusive Design .....	14
2.7.	Tools & Methoden zur UX-Verbesserung .....	15
2.7.1.	User Personas & Zielgruppenanalyse .....	15
2.7.2.	Wireframes & UI-Prototyping.....	15
2.7.3.	Usability-Testing.....	15
2.7.4.	Snackbar-Feedback & Interaktionsrückmeldungen .....	16
2.7.5.	Iteration & kontinuierliche Verbesserung .....	16
3.	Praktische Umsetzung.....	16
3.1.	Zielgruppenanalyse & UX-Ziele .....	16
3.2.	Farbwahl und Kontraste.....	17
3.3.	Schriftarten und Typografie .....	18
3.3.1.	Verwendete Schriftart.....	18
3.3.2.	Schriftgrößen und Struktur.....	19
3.3.3.	Zeilenhöhe und Lesbarkeit .....	19
3.3.4.	Konsistenz durch Theme-Vererbung .....	19
3.4.	Aufbau der Informationsarchitektur .....	19
3.4.1.	Strukturierung der Anwendung.....	19
3.4.2.	Navigationskonzept.....	20
3.4.3.	Informationszugriff & Orientierung.....	20
3.4.4.	Visuelle Unterstützung.....	21
3.5.	Konsistenz durch Wiederverwendbarkeit von Komponenten .....	21
3.5.1.	Was sind wiederverwendbare Komponenten? .....	21
3.5.2.	Vorteile der Wiederverwendbarkeit.....	22
3.5.3.	Technische Umsetzung in FindMe.....	22
3.5.4.	Einheitliche Benutzerführung.....	22
3.6.	Mobile First & Responsives Design .....	<b>Error! Bookmark not defined.</b>
3.6.1.	Grundlagen des responsiven Designs .....	<b>Error! Bookmark not defined.</b>
3.6.2.	Umsetzung in FindMe .....	<b>Error! Bookmark not defined.</b>
3.6.3.	Testgeräte und Breakpoints .....	<b>Error! Bookmark not defined.</b>

3.7.	Getestete UX-Optimierungen und Userflows.....	<b>Error! Bookmark not defined.</b>
3.7.1.	Beispielhafte Userflows.....	<b>Error! Bookmark not defined.</b>
3.7.2.	Optimierungen aus Tests.....	<b>Error! Bookmark not defined.</b>
4.	Lessons Learned .....	23
4.1.	Erkenntnisse aus der Entwicklung.....	23
4.2.	Herausforderungen.....	23
4.3.	Persönliche Learnings .....	23
5.	Fachbegriffe .....	24
6.	Fazit und Ausblick.....	25

## 1. Einleitung

Digitale Anwendungen durchdringen heute alle Lebensbereiche – auch den Schulalltag. Ob Stundenplan, Hausübungen oder administrative Prozesse: Webanwendungen sollen Informationen zugänglich machen und die Zusammenarbeit vereinfachen. Doch selbst die technisch beste Anwendung verfehlt ihren Zweck, wenn sie kompliziert, unübersichtlich oder frustrierend zu bedienen ist. Hier setzt das Fachgebiet UI/UX-Design an.

Diese Diplomarbeit beschäftigt sich mit der Entwicklung und Gestaltung der Webapplikation FindMe, einem digitalen Fundbüro für eine Schule. Im Fokus steht die Frage, wie eine benutzerfreundliche Oberfläche (UI) und ein positives Nutzererlebnis (UX) systematisch geplant, umgesetzt und getestet werden können – besonders unter Berücksichtigung einer heterogenen Zielgruppe wie Schüler:innen, Lehrer:innen und Verwaltungspersonal.

Neben der theoretischen Auseinandersetzung mit zentralen UI/UX-Prinzipien wie Informationsarchitektur, Konsistenz und Barrierefreiheit, wird die praktische Umsetzung im Projekt FindMe detailliert beschrieben. Ziel ist es zu zeigen, wie gutes UI/UX-Design zur Effizienz, Zufriedenheit und Akzeptanz einer Anwendung beiträgt – gerade in einem schulischen Umfeld.

## 2. Theorie: UI/UX-Design

### 2.1. Was ist User Experience (UX)?

Der Begriff **User Experience (UX)** – zu Deutsch *Nutzererlebnis* – beschreibt die Gesamtheit aller Erfahrungen, Eindrücke und Interaktionen eines Nutzers mit einem digitalen Produkt oder Service. Dabei geht es nicht nur um die reine Funktionalität der Anwendung, sondern vor allem um das subjektive Empfinden während der Nutzung: Ist die Anwendung intuitiv bedienbar? Macht die Nutzung Spaß? Wird der Nutzer frustriert oder motiviert?

Im Fokus der UX steht der Mensch – und seine Erwartungen, Fähigkeiten, Bedürfnisse sowie emotionale Reaktionen. Bei digitalen Produkten wie **FindMe**, einer Webapplikation zur Verwaltung von verlorenen und gefundenen Gegenständen an Schulen, ist eine gute UX entscheidend, um sicherzustellen,

dass alle Benutzer – egal ob Schüler, Lehrpersonal oder Fundbüro – das System effizient und gerne verwenden.

### *2.1.1. Bestandteile der User Experience*

Um eine positive UX zu schaffen, müssen laut Fachliteratur drei zentrale Bestandteile berücksichtigt werden:

#### **1. Funktionalität**

Die Anwendung muss stabil, zuverlässig und technisch einwandfrei funktionieren. Fehlerhafte Eingaben, Abstürze oder Ladezeiten beeinträchtigen die UX erheblich. In „FindMe“ wurde daher darauf geachtet, dass alle Formulare valide Eingaben verlangen und Ladezeiten gering gehalten werden.

#### **2. Nützlichkeit**

Die Applikation soll dem Nutzer einen echten Mehrwert bieten. Das kann durch eine klare Navigation, verständliche Informationen und hilfreiche Features geschehen – etwa das strukturierte Einreichen von Fundmeldungen oder die gezielte Suche nach verlorenen Objekten.

#### **3. Ästhetik**

Das Design soll ansprechend, modern und gleichzeitig übersichtlich wirken. Farben, Icons und Layout tragen dazu bei, eine angenehme Stimmung zu erzeugen und Vertrauen in die Anwendung zu stärken. Dabei wurde bei „FindMe“ auf ein minimalistisches, flaches Design gesetzt, das moderne UI-Prinzipien berücksichtigt.

### *2.1.2. Ziele der User Experience*

Die Zielsetzung von UX-Design besteht darin, dem User eine reibungslose, angenehme und motivierende Interaktion mit dem Produkt zu ermöglichen.

Eine gute UX bewirkt:

- höhere **Zufriedenheit** bei der Nutzung
- stärkere **Nutzerbindung** und Wiederverwendung
- **Effizienz** bei der Zielerreichung
- Abgrenzung gegenüber Konkurrenzprodukten
- indirekt auch eine **Steigerung der Nutzung** und des Erfolgs der Applikation

### 2.1.3. *UX in schulischen Anwendungen*

Gerade in schulischen Kontexten, in denen unterschiedlichste Nutzergruppen mit variierendem technischen Vorwissen arbeiten, ist eine gute UX entscheidend. Schülerinnen und Schüler sollen das System genauso einfach bedienen können wie Verwaltungspersonal oder Lehrkräfte – ohne lange Einlernzeiten. In „FindMe“ wurde daher besonderes Augenmerk auf eine klare Benutzerführung, verständliche Icons und Feedbackmechanismen gelegt.

## 2.2. Was ist User Interface (UI)?

Das **User Interface (UI)** – zu Deutsch *Benutzeroberfläche* – ist die zentrale Schnittstelle zwischen Mensch und Software. Es ist der Teil einer Anwendung, den die Nutzer sehen und mit dem sie aktiv interagieren. Dabei geht es um visuelle Gestaltungselemente wie Buttons, Formulare, Navigationsleisten, Farben und Typografie – aber auch um deren sinnvolle Anordnung und Funktion.

Während die User Experience (UX) das Gesamtgefühl bei der Nutzung eines digitalen Produkts umfasst, stellt das UI die konkrete *technische und visuelle Umsetzung* dieses Erlebnisses dar. Ein gut gestaltetes UI trägt wesentlich dazu bei, dass eine Anwendung leicht verständlich, angenehm zu bedienen und effizient nutzbar ist.

### 2.2.1. *Typische Bestandteile eines UI*

- **Navigationselemente:** z. B. Menüs, Breadcrumbs, Sidebar
- **Interaktive Komponenten:** Buttons, Textfelder, Drop-downs, Checkboxes
- **Rückmeldungen:** Ladeanimationen, Erfolgsmeldungen, Validierungsfehler
- **Visuelle Gestaltung:** Farben, Schriftarten, Icons, Layout-Raster
- **Responsives Design:** automatische Anpassung an verschiedene Bildschirmgrößen

### 2.2.2. *UI-Ziele und Anforderungen*

Ein gelungenes UI erfüllt folgende Anforderungen:

- **Intuitive Bedienung:** Der User versteht auf Anhieb, wie die Anwendung funktioniert
- **Effiziente Navigation:** Wege zum Ziel sind klar, logisch und kurz

- **Ästhetik und Klarheit:** Design wirkt modern, aufgeräumt und visuell angenehm
- **Konsistenz:** Wiederkehrende Designelemente schaffen Vertrauen und Orientierung
- **Fehlertoleranz:** Nutzerfehler werden abgefangen oder nachvollziehbar kommuniziert

### 2.2.3. *UI-Ansatz im Projekt FindMe*

In der Webapplikation **FindMe** wird das UI mit dem Framework **React** umgesetzt, wobei die Gestaltung über die UI-Komponentenbibliothek **Material UI (MUI)** erfolgt. MUI bietet ein modernes, barrierearmes Designsystem mit vorgefertigten Komponenten, das sich stark an den **Material Design Guidelines** von Google orientiert.

Die Anwendung gliedert sich in mehrere klar strukturierte Bereiche:

- eine Navigationsleiste für schnellen Zugriff auf Hauptfunktionen
- Seiten für das Melden, Durchsuchen und Verwalten von Fundgegenständen
- administrative Komponenten für Lehrkräfte oder Schulverwaltung
- responsive Layouts, damit die Anwendung auch auf Laptops oder Tablets gut funktioniert

Die verwendeten UI-Komponenten (z. B. Button, TextField, Dialog, Snackbar) wurden konsequent auf **Konsistenz**, **Barrierefreiheit** und **Responsivität** ausgelegt. Durch MUI-Designprinzipien wie visuelles Feedback, klarer Kontrast und visuelle Hierarchien wurde ein Interface geschaffen, das SchülerInnen, LehrerInnen und Verwaltungspersonal intuitiv bedienen können.

## 2.3. Usability – Benutzerfreundlichkeit als Erfolgsfaktor

**Usability**, also die *Gebrauchstauglichkeit* einer Anwendung, beschreibt, wie einfach, effizient und zufriedenstellend ein Nutzer mit einem System interagieren kann. Im Zentrum steht dabei die Frage: Wie leicht ist es für jemanden, eine Aufgabe erfolgreich zu erledigen – ohne Frust, Umwege oder technische Hürden?

Gute Usability ist nicht nur ein „Nice to have“, sondern ein zentraler Erfolgsfaktor für digitale Produkte. In schulischen Anwendungen wie **FindMe** ist

es besonders wichtig, dass Nutzer mit unterschiedlichem technischen Hintergrund (z. B. Schüler, Lehrkräfte, Sekretariat) das System ohne lange Einarbeitung oder Hilfestellung verstehen und nutzen können.

### *2.3.1. Ziele einer guten Usability*

Eine benutzerfreundliche Anwendung verfolgt folgende Ziele:

- **Einfache Erlernbarkeit:** Auch Erstnutzer sollen sofort verstehen, wie die Applikation funktioniert
- **Effizienz:** Regelmäßige Nutzer sollen Aufgaben schnell und zielgerichtet erledigen können
- **Fehlertoleranz:** Fehler sollen vermieden oder leicht korrigierbar sein
- **Zufriedenheit:** Die Nutzung soll angenehm sein und ein positives Erlebnis vermitteln

### *2.3.2. Kriterien für hohe Usability*

Laut ISO 9241-11 basiert gute Usability auf drei Kerndimensionen:

- **Effektivität:** Aufgaben werden korrekt und vollständig erledigt
- **Effizienz:** Der Aufwand zur Erledigung ist gering (z. B. wenige Klicks, logische Navigation)
- **Zufriedenheit:** Nutzer empfinden die Anwendung als angenehm und unterstützend

### *2.3.3. Usability im Projekt FindMe*

Im Projekt **FindMe** wurde die Benutzerfreundlichkeit durch folgende Maßnahmen gezielt umgesetzt:

- **Klar strukturierte Navigation:** Die Hauptfunktionen sind über eine übersichtliche Seitenleiste erreichbar
- **Eindeutige Benutzerführung:** Buttons sind beschriftet, Aktionen geben sofort visuelles Feedback
- **Formularvalidierung:** Eingabefehler werden direkt beim Ausfüllen angezeigt
- **Responsives Design:** Die Anwendung ist auch bei kleineren Displays gut bedienbar



- **Einheitliche UI-Komponenten:** Durch die Nutzung von Material UI wirkt alles vertraut und konsistent
- **Zugänglichkeit:** Kontrastreiche Farben und ausreichende Button-Größen erleichtern die Nutzung für alle

Beispielsweise kann ein Nutzer mit nur wenigen Klicks einen Fundgegenstand melden – ohne unnötige Informationen eingeben zu müssen. Auch die Suche nach verlorenen Objekten wurde so gestaltet, dass diese schnell über Kategorien und Schlagwörter gefiltert werden kann.

#### *2.3.4. Vorteile durch hohe Usability*

Eine benutzerfreundliche Anwendung wie **FindMe** bietet zahlreiche Vorteile:

- **Weniger Schulungsaufwand**
- **Geringere Supportkosten**
- **Höhere Akzeptanz und Nutzungsrate**
- **Weniger Fehlerquellen im täglichen Gebrauch**

Gerade im schulischen Alltag, wo Zeit und Aufmerksamkeit oft knapp sind, macht gute Usability den Unterschied zwischen einer gerne genutzten Anwendung und einer, die ignoriert wird.

## 2.4. Informationsarchitektur im UX-Kontext

Die **Informationsarchitektur (IA)** ist ein zentrales Element innerhalb der User Experience. Sie befasst sich mit der **Strukturierung, Organisation und Benennung** von Inhalten innerhalb einer digitalen Anwendung. Ziel ist es, dass Nutzer Informationen schnell finden, verstehen und damit interagieren können – ohne sich „verirrt“ zu fühlen.

Gerade bei Anwendungen wie **FindMe**, die verschiedene Funktionen (z. B. Fundmeldung, Suche, Verwaltung) für unterschiedliche Nutzerrollen anbieten, ist eine durchdachte Informationsarchitektur entscheidend, um Übersichtlichkeit, Orientierung und Effizienz sicherzustellen.

#### *2.4.1. Aufgaben der Informationsarchitektur*

- **Inhalte logisch gruppieren**  
Ähnliche Informationen werden zusammengefasst (z. B. „Alle Fundmeldungen“, „Meine Funde“)

- **Navigation klar strukturieren**  
Hauptbereiche der Anwendung sollen jederzeit zugänglich sein
- **Bezeichnungen verständlich formulieren**  
Technische Begriffe oder Abkürzungen sollen vermieden werden
- **Userflows planen**  
Wie kommt der Nutzer von A nach B – mit möglichst wenigen Klicks und ohne Umwege?

#### 2.4.2. *Strukturprinzipien einer guten IA*

- **Hierarchien:** Inhalte sind logisch von oben nach unten gegliedert
- **Kategorisierung:** Objekte werden sinnvoll gruppiert (z. B. Fundarten wie „Elektronik“, „Kleidung“)
- **Suchbarkeit:** Inhalte lassen sich leicht durchsuchen und filtern
- **Orientierung:** Der Nutzer weiß immer, wo er sich gerade befindet

#### 2.4.3. *Informationsarchitektur in FindMe*

In der Umsetzung von **FindMe** wurde besonderes Augenmerk auf die IA gelegt. Die Benutzeroberfläche ist in folgende Hauptbereiche gegliedert:

- **Startseite:** Übersicht über neueste oder relevante Funde
- **Fund melden:** Schrittweiser Formularprozess zur Eingabe von Funddaten
- **Meine Funde:** Übersicht der eigenen Meldungen (filterbar und editierbar)
- **Verwaltung:** Nur für Admins sichtbar – zur Nachverfolgung und Archivierung
- **Hilfe & Kontakt:** Erklärungen zur Bedienung und Kontakt zur Schulverwaltung

Durch den Einsatz von **React Router** wurden die Seiten logisch verknüpft. Jede Aktion (z. B. auf einen Button klicken) führt konsistent zu einer neuen URL/Ansicht – das sorgt für Klarheit. Ergänzend ermöglichen Filterfunktionen eine gezielte Eingrenzung nach Kategorie, Funddatum oder Beschreibung.

Zusätzlich wird durch farbliche Trennung und sprechende Icons (z. B. Lupe, Papierkorb, Checkmark) visuell verdeutlicht, was zu welchem Bereich gehört. Damit orientieren sich auch unerfahrene Nutzer leicht.

## 2.5. Konsistenz im Design

**Konsistenz** ist ein zentrales Gestaltungsprinzip im UI/UX-Design. Sie beschreibt die einheitliche Gestaltung und Funktionsweise von Benutzeroberflächen – sowohl visuell als auch funktional. Wenn sich Designelemente innerhalb einer Anwendung wiederholen oder vorhersehbar verhalten, entsteht für den Nutzer ein Gefühl von Vertrauen, Kontrolle und Sicherheit.

Ein konsistentes Design unterstützt die **Lernfähigkeit**, **Orientierung** und **Effizienz** in der Nutzung. Gerade in einer schulischen Applikation wie **FindMe**, die von sehr unterschiedlichen Benutzergruppen (Schüler:innen, Lehrkräften, Verwaltungspersonal) verwendet wird, spielt Konsistenz eine entscheidende Rolle für die Akzeptanz und Nutzbarkeit der Anwendung.

### 2.5.1. *Arten von Konsistenz*

#### 1. **Visuelle Konsistenz**

Farben, Schriftarten, Abstände und Layouts werden einheitlich verwendet. In FindMe wurden durch **Material UI** Designregeln wie feste Abstände, Farbpaletten und Typografie-Vorgaben standardisiert.

#### 2. **Funktionale Konsistenz**

Buttons, Filter und Interaktionen funktionieren über die gesamte Anwendung hinweg gleich. Wenn man z. B. ein Formular ausfüllt, wird bei allen Feldern dieselbe Validierungslogik angewendet.

#### 3. **Sprachliche Konsistenz**

Begriffe und Bezeichnungen bleiben gleich (z. B. „Fund melden“ statt einmal „Eintragen“, einmal „Neu hinzufügen“). In FindMe wurden alle Texte und Fehlermeldungen bewusst vereinheitlicht.

#### 4. **Interaktionskonsistenz**

Navigations- und Steuerungselemente reagieren stets gleich: Ein Klick auf ein Icon führt zu einer Detailseite, ein Snackbar zeigt Rückmeldungen immer unten links, etc.

### 2.5.2. *Vorteile durch Konsistenz*

- **Schnellere Orientierung:** Nutzer müssen sich nicht ständig neu orientieren
- **Weniger Fehler:** Konsistente Prozesse vermeiden Missverständnisse
- **Besseres Look-and-Feel:** Die App wirkt professioneller und durchdachter

- **Höhere Wiederverwendung von Komponenten:** Spart Entwicklungszeit und verbessert Wartbarkeit

### 2.5.3. *Konsistenz in FindMe*

In **FindMe** wurde Konsistenz auf mehreren Ebenen umgesetzt:

- Alle Buttons sind in Farbe, Form und Verhalten identisch (z. B. Primär-Buttons in Rot, sekundäre grau)
- Die Farbwelt basiert auf einem festen MUI-Theme mit dunklem Rot (Primär) und Türkis (Akzent)
- Eingabefelder, Fehlermeldungen und Layouts folgen einer einheitlichen Struktur (z. B. Padding, Margin)
- Wiederkehrende Komponenten wie Header, Card, Dialog oder Snackbar wurden als **Reusable Components** implementiert

Durch diese Herangehensweise wird die Anwendung als stabil, vertrauenswürdig und durchgehend nachvollziehbar wahrgenommen – ein wichtiger Erfolgsfaktor für jede UX-orientierte Anwendung.

## 2.6. Aktuelle Trends im modernen UI-Design

Das UI-Design unterliegt einem ständigen Wandel – technologische Entwicklungen, veränderte Nutzergewohnheiten und neue Designphilosophien beeinflussen die Gestaltung digitaler Benutzeroberflächen. Moderne UI-Trends zielen dabei nicht nur auf visuelle Attraktivität ab, sondern vor allem auf eine **verbesserte User Experience**, höhere Zugänglichkeit und effizientere Interaktionen.

Auch in der Entwicklung von **FindMe** wurden einige dieser aktuellen Trends bewusst eingesetzt, um eine moderne, intuitive und barrierearme Webapplikation für den Schulalltag zu schaffen.

### 2.6.1. *Material Design & Komponentenbibliotheken*

Ein dominierender Trend ist der Einsatz von **Designsystemen** wie **Material Design**, das von Google entwickelt wurde. In FindMe kommt **Material UI (MUI)** zum Einsatz – eine React-Implementierung dieses Designsystems. MUI bietet klare visuelle Hierarchien, responsive Komponenten und durchdachte Animationen, die den Nutzer sicher durch die Anwendung führen.

**Vorteile:**

- Einheitliches Look & Feel
- Vordefinierte Interaktionsmuster
- Schnellere Entwicklung durch wiederverwendbare Komponenten

### 2.6.2. *Responsive Design (Mobile First)*

Moderne UIs müssen sich dynamisch an verschiedene Bildschirmgrößen anpassen – vom Schul-PC bis zum Tablet. In **FindMe** wurde ein responsives Layout umgesetzt, das auf **Grid-Systemen**, **Flexbox** und **Media Queries** basiert. Buttons, Texte und Abstände skalieren automatisch und sorgen für eine optimale Darstellung auf allen Geräten.

### 2.6.3. *Dark Mode & Farbkontraste*

Immer mehr Anwendungen bieten einen **Dark Mode** oder farbliche Varianten zur Anpassung an Lichtverhältnisse. Auch wenn FindMe aktuell einen hellen Standardmodus verwendet, wurde bei der Farbwahl auf **hohen Kontrast** geachtet (z. B. Rot #B0151F auf weißem Hintergrund), um die Lesbarkeit zu maximieren.

### 2.6.4. *Microinteractions*

**Kleine, animierte Rückmeldungen** – wie Hover-Effekte, Ladeanimationen oder Snackbar-Meldungen – steigern die Benutzerzufriedenheit erheblich. In FindMe zeigen Snackbar-Komponenten z. B. Bestätigungen für erfolgreich abgeschickte Formulare oder Fehlermeldungen bei falschen Eingaben.

### 2.6.5. *Minimalismus & Flat Design*

Aktuelle UI-Designs setzen auf klare Linien, viel Weißraum und **reduzierte Oberflächen**. In FindMe wurden unnötige visuelle Ablenkungen bewusst vermieden. Die Benutzeroberfläche bleibt so fokussiert auf den Inhalt – etwa beim Melden eines Fundgegenstands.

### 2.6.6. *Barrierefreiheit & Inclusive Design*

Ein zunehmend wichtiger Aspekt ist die **Zugänglichkeit für alle Nutzergruppen** – unabhängig von körperlichen oder kognitiven Einschränkungen. Auch wenn FindMe keine vollständige WCAG-konforme Umsetzung bietet, wurde auf:

- Tastaturbedienbarkeit
- ausreichende Schriftgröße
- Farbkontraste  
geachtet – eine solide Basis für eine inklusivere Anwendung.

## 2.7. Tools & Methoden zur UX-Verbesserung

Die Gestaltung einer optimalen User Experience ist kein einmaliger Schritt, sondern ein **iterativer Prozess**. Um das Nutzererlebnis laufend zu verbessern, stehen Designern und Entwicklern zahlreiche Methoden, Werkzeuge und Testverfahren zur Verfügung. Sie helfen dabei, reale Probleme frühzeitig zu erkennen und Lösungen nutzerzentriert umzusetzen.

Auch im Projekt **FindMe** wurden einige dieser Prinzipien und Methoden angewendet bzw. bewusst eingeplant, um eine benutzerfreundliche und funktionale Anwendung für den Schulalltag zu schaffen.

### 2.7.1. *User Personas & Zielgruppenanalyse*

Ein zentrales UX-Werkzeug ist die Erstellung von **User Personas** – fiktiven Nutzerprofilen, die typische Anwendergruppen repräsentieren. Für FindMe wurden unter anderem folgende Zielgruppen definiert:

- **Schüler:innen** (Gegenstände verloren / wiederfinden)
- **Lehrer:innen** (Fundgegenstände melden oder prüfen)
- **Schulsekretariat** (Verwaltung & Archivierung)

Die Struktur und Sprache der Benutzeroberfläche wurde gezielt auf diese Gruppen abgestimmt – z. B. durch einfache Formulare, verständliche Labels und klare Navigation.

### 2.7.2. *Wireframes & UI-Prototyping*

Ein effektives Mittel zur Visualisierung und Testung von Designideen sind **Wireframes** oder **Prototypen**. Auch bei FindMe wurden einzelne UI-Elemente zuerst skizziert oder als React-Komponenten lokal getestet, bevor sie produktiv eingesetzt wurden. Das ermöglichte schnelle Feedback und gezielte Anpassungen im Layout oder der Interaktion.

### 2.7.3. *Usability-Testing*

Bereits einfache **Usability-Tests** mit wenigen Testpersonen können aufschlussreiche Hinweise auf Schwächen im Design geben. In FindMe wurden vor dem Projektabschluss gezielt **Feedbackrunden** mit Mitschüler:innen durchgeführt – z. B. zur Menüstruktur, zur Verständlichkeit der Formulare oder zur Logik der Filterfunktion.

Typische Fragen, die dabei geklärt wurden:

- „Finde ich sofort, wo ich melden kann?“

- „Verstehe ich, was ein Feld von mir will?“
- „Wie leicht komme ich wieder zurück zur Startseite?“

#### 2.7.4. *Snackbar-Feedback & Interaktionsrückmeldungen*

Ein einfaches, aber effektives UX-Werkzeug ist das sogenannte **Immediate Feedback**. In FindMe wird z. B. nach erfolgreichem Absenden eines Formulars eine **Snackbar** eingeblendet, die dem Nutzer eine Bestätigung gibt. Auch Fehler (z. B. leere Felder) werden klar und konsistent visualisiert.

#### 2.7.5. *Iteration & kontinuierliche Verbesserung*

UX ist nie „fertig“. Eine moderne Webanwendung wie FindMe muss regelmäßig angepasst werden – etwa bei technischen Updates, geänderten Anforderungen oder nach Feedback von Nutzern. Durch den modularen Aufbau mit React-Komponenten und eine saubere Strukturierung des Backends (ASP.NET) kann die Anwendung leicht erweitert und verbessert werden.

## 3. Praktische Umsetzung

Im folgenden Kapitel wird die praktische Umsetzung des Projekts FindMe im Hinblick auf eine möglichst gute User Experience (UX) beschrieben. Dabei liegt der Fokus insbesondere auf den Aspekten UI-Design, Informationsarchitektur, Konsistenz, Responsivität und Usability-Optimierungen.

FindMe ist eine Webapplikation zur Verwaltung von verlorenen und gefundenen Gegenständen an einer Schule. Sie richtet sich an verschiedene Benutzergruppen – unter anderem Schüler:innen, Lehrkräfte und die Verwaltung – und wurde mit React im Frontend sowie ASP.NET im Backend umgesetzt. Für die Gestaltung der Benutzeroberfläche kam das Framework Material UI zum Einsatz.

### 3.1. Zielgruppenanalyse & UX-Ziele

Bereits in der frühen Planungsphase wurde eine Zielgruppenanalyse durchgeführt. Die wichtigsten Nutzergruppen lassen sich wie folgt kategorisieren:

- **Schüler:innen** → wollen verlorene Gegenstände schnell wiederfinden



- **Lehrkräfte** → melden gefundene Gegenstände oder helfen beim Suchen
- **Verwaltungspersonal** → organisiert, archiviert und verwaltet die Einträge zentral

Basierend auf diesen Gruppen wurden folgende UX-Ziele definiert:

- **Schnelle Orientierung:** alle Nutzergruppen sollen innerhalb weniger Sekunden verstehen, wie die Anwendung funktioniert
- **Klare Navigation:** ein einfaches, konsistentes Menü leitet die Nutzer intuitiv durch alle Funktionen
- **Effiziente Abläufe:** möglichst wenige Klicks bis zur gewünschten Aktion (z. B. Fundmeldung)
- **Optische Klarheit:** Farben, Abstände und Icons sollen gezielt zur Orientierung beitragen
- **Barrierearme Nutzung:** durch klare Kontraste, ausreichende Schriftgröße und Tastaturbedienung

---

## 3.2. Farbwahl und Kontraste

Bei der Gestaltung der Benutzeroberfläche wurde großer Wert auf **Lesbarkeit und Kontraste** gelegt. Die gewählten Farben entsprechen modernen UI-Standards und sind kompatibel mit den **Material Design Guidelines**.

### Primärfarbe:

- #B0151F – ein dunkler Rotton zur Hervorhebung von Buttons und aktiven Elementen

### Akzentfarbe:

- #005F63 – ein dunkles Türkis für sekundäre Hinweise und passive Elemente

### Schriftfarbe:

- Standardmäßig Schwarz auf weißem Hintergrund für maximalen Kontrast

Diese Farbkombinationen wurden bewusst gewählt, um ein kontrastreiches, aber dennoch ruhiges Farbschema zu ermöglichen. Die Einhaltung der **WCAG-Richtlinien** (Web Content Accessibility Guidelines) für Kontrastverhältnisse wurde mithilfe von Tools wie Adobe Color getestet. So wurde z. B.



sichergestellt, dass Textfarbe und Hintergrund ein Kontrastverhältnis von mindestens **7:1** erreichen.

Alle Farben wurden als Variablen in der MUI-Theme-Konfiguration definiert, wodurch Änderungen konsistent über die gesamte Anwendung hinweg übernommen werden.

### 3.3. Schriftarten und Typografie

Die **Typografie** ist ein zentraler Bestandteil einer guten User Experience. Sie bestimmt, wie leicht Inhalte gelesen, verstanden und interpretiert werden können. In **FindMe** wurde besonders darauf geachtet, dass Texte gut lesbar, strukturiert und barrierearm dargestellt werden – sowohl auf Desktop- als auch auf mobilen Endgeräten.

#### 3.3.1. *Verwendete Schriftart*

Als globale Schriftart wurde "**Roboto**" definiert, ergänzt durch "**Arial**" als Fallback sowie das generische "sans-serif". Diese Schriftkombination garantiert maximale Lesbarkeit und breite Kompatibilität auf allen Geräten.

In der Theme-Konfiguration von Material UI wurde dies wie folgt umgesetzt:

```
const theme = createTheme({
  palette: {
    primary: {main: "#c62828", },
    background: { default: "#ffffff" },
    text: { primary: "#000000" } },
  typography: { fontFamily: "Roboto, Arial, sans-serif", fontSize: 14,
    button: { textTransform: "none" }, }, components: { MuiButton: {
      defaultProps: { variant: "contained", color: "primary" }, }, } });
```

#### **Merkmale dieser Wahl:**

- *Roboto* ist eine moderne, gut lesbare Webschrift mit klaren Proportionen
- *Arial* dient als gängige Fallback-Schriftart für Systeme, die Roboto nicht installiert haben
- *Sans-serif* stellt sicher, dass bei allen Ausfällen zumindest eine serifenlose Schrift genutzt wird

Zusätzlich wurde die automatische Umwandlung von Button-Texten in Großbuchstaben (textTransform: "uppercase") deaktiviert, um besser lesbare und benutzerfreundliche Beschriftungen zu ermöglichen.

### *3.3.2. Schriftgrößen und Struktur*

- Die Basis-Schriftgröße wurde global auf **14px** gesetzt
- Für Überschriften und wichtige Elemente kommen gestaffelte Größen zum Einsatz (h1–h4)
- Platzhaltertexte und Hinweise nutzen hellere Grautöne, um sich vom Hauptinhalt abzuheben
- Alle Texte sind kontrastreich – z. B. schwarze Schrift auf weißem Hintergrund (#000000 auf #ffffff)

### *3.3.3. Zeilenhöhe und Lesbarkeit*

Durch Material UI wird automatisch ein harmonischer **line-height** generiert, der je nach Komponente variiert. In langen Texten oder Formularbeschreibungen wird dadurch eine angenehme Lesbarkeit erreicht.

### *3.3.4. Konsistenz durch Theme-Vererbung*

Da das Theme über den gesamten React-Komponentenbaum per `<ThemeProvider>` eingebunden wurde, sind Schriftart, Größe und Stil in der gesamten Anwendung konsistent. Das betrifft sowohl systemweite Komponenten wie Button, TextField, Typography als auch benutzerdefinierte Inhalte.

## **3.4. Aufbau der Informationsarchitektur**

Die **Informationsarchitektur (IA)** beschreibt die Strukturierung, Organisation und Navigation innerhalb einer digitalen Anwendung. Ziel ist es, dass Nutzer Inhalte schnell und intuitiv finden, ohne durch unklare Menüs oder verschachtelte Seiten den Überblick zu verlieren.

Gerade bei einer Anwendung wie **FindMe**, die mehrere zentrale Funktionen abbildet (Fund melden, Suchen, Verwaltung), ist eine durchdachte Informationsarchitektur entscheidend für die **Benutzerfreundlichkeit**.

### *3.4.1. Strukturierung der Anwendung*

Die Applikation ist in folgende Hauptbereiche gegliedert:

1. **Startseite**  
Kurze Einführung und Übersicht – zentraler Einstiegspunkt für alle Nutzergruppen.
2. **Fund melden**  
Interaktives Formular zur Einreichung eines Fundgegenstands inkl. Beschreibung, Fundort und Bild.
3. **Suche / Liste der Funde**  
Öffentliche Fundliste mit Filterfunktionen (z. B. nach Kategorie, Datum, Beschreibung).
4. **Meine Meldungen**  
Anzeige und Verwaltung aller vom eingeloggten Nutzer erstellten Fundmeldungen.
5. **Adminbereich (nur für Berechtigte)**  
Verwaltungsansicht zur Archivierung, Statusänderung oder Löschung von Einträgen.
6. **Hilfe & Kontakt**  
Anleitungen, häufige Fragen und Kontaktformular für Rückfragen an die Schulverwaltung.

### *3.4.2. Navigationskonzept*

Die Navigation erfolgt über eine klar strukturierte **Sidebar** (linke Seitenleiste), die auf allen Seiten sichtbar bleibt. Sie enthält Icons mit erklärenden Labels, wodurch auch technisch weniger erfahrene Nutzer sofort erkennen, welche Aktion sich hinter welchem Menüpunkt verbirgt.

Für das Routing wurde **React Router** verwendet. Dadurch ist gewährleistet:

- **Konsistente URL-Struktur** (z. B. /report, /search, /admin)
- **Browsernavigation ist möglich** (Vor/Zurück-Buttons, Lesezeichen)
- **Komponentenbasiertes Laden** → bessere Performance und Übersichtlichkeit

### *3.4.3. Informationszugriff & Orientierung*

Zusätzlich zur Navigation gibt es folgende unterstützende Elemente:

- **Breadcrumbs (optional)** für tiefer verschachtelte Inhalte
- **Such- und Filterfunktion** mit Live-Feedback in der Fundliste

- **Visuelles Feedback** (z. B. aktive Sidebar-Icons, Hover-Effekte)

Der gesamte Seitenaufbau folgt dem Prinzip: **Zuerst Übersicht, dann Aktion, dann Detail** – etwa in der Fundliste: erst alle Funde, dann gezielt filtern, dann Details aufrufen.

#### 3.4.4. *Visuelle Unterstützung*

- Farblich hervorgehobene Buttons lenken gezielt zur nächsten logischen Aktion
- Einheitliche Icons (z. B. Lupe für Suche, Pluszeichen für „neu“) erleichtern die Orientierung
- Alle Inhaltsblöcke sind visuell sauber getrennt (Cards, Container, Divider)

### 3.5. Konsistenz durch Wiederverwendbarkeit von Komponenten

Ein wichtiger Aspekt eines konsistenten UI-Designs ist die **Wiederverwendbarkeit von Komponenten**. In modernen Webapplikationen wie **FindMe** wird dieses Prinzip durch den Einsatz von **React-Komponenten** konsequent umgesetzt. Dadurch entsteht ein einheitliches Erscheinungsbild, das sich positiv auf die **User Experience, Wartbarkeit** und **Entwicklungsgeschwindigkeit** auswirkt.

#### 3.5.1. *Was sind wiederverwendbare Komponenten?*

In React können UI-Elemente – z. B. Buttons, Formulare, Dialoge – einmal als Komponente erstellt und dann an beliebigen Stellen der Applikation wiederverwendet werden. Dabei bleiben Stil und Verhalten konsistent, was nicht nur den Code sauber hält, sondern auch eine einheitliche Bedienung für die Nutzer sicherstellt.

Beispiele für wiederverwendbare Komponenten in **FindMe**:

- **CustomButton**: z. B. für „Melden“, „Speichern“, „Löschen“ – immer gleiche Farbe, Größe, Verhalten
- **FundCard**: Darstellung eines Fundobjekts (Bild, Beschreibung, Status, Datum)
- **SnackbarMessage**: Einheitliche Rückmeldung bei Erfolg, Fehler oder Warnung

- **ConfirmDialog:** Pop-up zur Bestätigung von Aktionen wie „Löschen“ oder „Archivieren“

### 3.5.2. *Vorteile der Wiederverwendbarkeit*

- **Visuelle Konsistenz:** Gleiche Aktionen sehen überall gleich aus (z. B. alle Hauptbuttons in Primärfarbe #c62828)
- **Reduzierter Entwicklungsaufwand:** Einmal gebaut, mehrfach einsetzbar
- **Einfachere Wartung:** Änderungen an einem zentralen Ort wirken sich überall aus
- **Bessere UX:** Nutzer erkennen vertraute Muster schneller und handeln sicherer

### 3.5.3. *Technische Umsetzung in FindMe*

Die Komponenten wurden in einem eigenen components/-Ordner organisiert. Sie sind modular aufgebaut und übernehmen automatisch das definierte globale MUI-Theme (Farben, Schrift, Button-Verhalten).

Beispiel: CustomButton-Komponente mit Theme-Integration

```
export default function CustomButton({ label, onClick }) {  
  return (<Button variant="contained" color="primary" onClick={onClick}>  
    {label} </Button> );  
}
```

Diese Komponente wird überall dort eingesetzt, wo Aktionen erforderlich sind – sei es beim Abschicken eines Formulars oder bei Verwaltungsaktionen im Adminbereich.

### 3.5.4. *Einheitliche Benutzerführung*

Dank der wiederverwendeten Komponenten ist das Verhalten vorhersehbar:

- Alle Buttons lösen visuelles Feedback aus
- Formularfelder validieren Eingaben auf gleiche Weise
- Dialogfenster verhalten sich konsistent – egal, von welcher Seite aus sie geöffnet werden

Dies trägt dazu bei, dass sich Nutzer schnell zurechtfinden und Vertrauen in die Anwendung aufbauen.

## 4. Lessons Learned

Im Laufe der Umsetzung von **FindMe** konnte viel praktisches Wissen im Bereich **UI/UX-Design** gesammelt und angewendet werden. Besonders die Verbindung von theoretischen Prinzipien mit realem Nutzerfeedback war dabei entscheidend.

### 4.1. Erkenntnisse aus der Entwicklung

- **UX ist mehr als nur „schönes Design“:** Es geht um klare Strukturen, verständliche Abläufe und echtes Nutzerverständnis.
- **Testen bringt oft überraschende Erkenntnisse:** Dinge, die für Entwickler logisch wirken, sind für Nutzer manchmal unverständlich – deshalb war das Feedback von außen besonders wertvoll.
- **Komponentenbasiertes Arbeiten erhöht die Konsistenz:** Durch wiederverwendbare UI-Bausteine konnten Designfehler und Redundanzen vermieden werden.
- **Ein gutes Theme spart Zeit und Nerven:** Die zentrale Definition von Farben, Schriftarten und Komponentenverhalten in Material UI machte spätere Anpassungen einfach und effektiv.

### 4.2. Herausforderungen

- **Mobile Optimierung war aufwändiger als erwartet:** Besonders die Darstellung komplexer Komponenten auf kleinen Displays erforderte Nacharbeit.
- **Nutzerführung im Adminbereich:** Die Anforderungen der Verwaltung unterschieden sich stark von denen der Schüler:innen – dafür mussten separate UX-Logiken umgesetzt werden.
- **Zeitmanagement:** UX-Optimierungen kosten Zeit. Ohne klare Priorisierung hätten viele Details zu viel Aufwand bedeutet.

### 4.3. Persönliche Learnings

- UX beginnt beim Verstehen der Nutzer, nicht beim Coden.
- Schon einfache Tools wie Live-Feedback, gute Typografie und klare Navigation machen einen riesigen Unterschied.
- Struktur, Planung und Wiederverwendbarkeit sind das Fundament jedes guten Frontends.

- Es lohnt sich, früh und oft zu testen – nicht erst am Ende.

## 5. Fachbegriffe

### User Interface (UI)

Das **User Interface** ist die Benutzerschnittstelle einer Software. Es umfasst alle grafischen und interaktiven Elemente, mit denen der Nutzer arbeitet – z. B. Buttons, Eingabefelder, Navigationselemente oder Icons.

---

### User Experience (UX)

Die **User Experience** beschreibt das gesamte Nutzungserlebnis eines Users mit einem digitalen Produkt. Sie umfasst Aspekte wie Verständlichkeit, Reaktionszeit, Gestaltung, Emotionen und Effizienz.

---

### Responsives Design

Ein **responsives Design** passt sich automatisch an verschiedene Bildschirmgrößen an. Dadurch wird die Anwendung sowohl auf Smartphones als auch auf Tablets oder Desktops optimal dargestellt.

---

### Informationsarchitektur (IA)

Die Informationsarchitektur beschreibt die Struktur und Organisation von Inhalten innerhalb einer Anwendung. Sie stellt sicher, dass Nutzer benötigte Informationen schnell finden und verstehen.

---

### Material UI (MUI)

**Material UI** ist ein React-Komponenten-Framework, das auf Googles „Material Design“-System basiert. Es bietet vorgefertigte, responsive UI-Komponenten mit hoher Gestaltungsqualität.

---

### React

**React** ist eine JavaScript-Bibliothek zur Erstellung von Benutzeroberflächen. Sie ermöglicht die Entwicklung modularer Komponenten und reagiert dynamisch auf Änderungen im Datenmodell.

---

### Theme

Ein **Theme** definiert globale Designwerte wie Farben, Schriftarten oder Buttonverhalten. In FindMe wurde über das Theme z. B. festgelegt, dass alle Buttons dieselbe Primärfarbe (#c62828) verwenden.

---

### Snackbar

Eine **Snackbar** ist ein kleines Benachrichtigungsfeld, das dem Nutzer eine Rückmeldung gibt (z. B. „Fund erfolgreich gemeldet“). Sie erscheint temporär und wird meist automatisch ausgeblendet.

---

### Komponente

In React bezeichnet eine **Komponente** einen wiederverwendbaren Baustein der Benutzeroberfläche. Beispiele in FindMe sind etwa FundCard, CustomButton oder ConfirmDialog.

---

### Barrierefreiheit (Accessibility)

Barrierefreiheit beschreibt die Gestaltung digitaler Inhalte so, dass sie auch von Menschen mit Einschränkungen problemlos genutzt werden können – z. B. durch hohe Kontraste, Tastaturbedienung oder klare Sprache.

## 6. Fazit und Ausblick

Die Entwicklung von **FindMe** hat eindrucksvoll gezeigt, wie entscheidend gutes **UI/UX-Design** für die Akzeptanz und Nutzbarkeit einer Webanwendung ist. Gerade in einem schulischen Umfeld mit unterschiedlichen Nutzergruppen – von technikaffinen Schüler:innen bis hin zu administrativem Personal – kommt es darauf an, **klare Strukturen, verständliche Abläufe und konsistentes Design** zu schaffen.



Im Rahmen dieses Projekts wurden nicht nur theoretische UX-Prinzipien wie **Informationsarchitektur, Responsivität, Konsistenz und Barrierefreiheit** erfolgreich umgesetzt, sondern auch durch konkrete Maßnahmen im Code (z. B. Wiederverwendung von Komponenten, MUI-Theme, responsives Grid-System) in der Praxis angewandt.

Besonders hilfreich war der regelmäßige Austausch mit realen Testpersonen, durch den Schwachstellen früh erkannt und gezielt verbessert werden konnten. Kleine Anpassungen – wie klarere Buttonbeschriftungen, sichtbare Pflichtfeldmarkierungen oder optimierte Filterfunktionen – hatten dabei oft eine große Wirkung.

### **Ausblick**

Obwohl FindMe in seiner aktuellen Version bereits voll funktionsfähig ist, bieten sich für die Zukunft weitere Potenziale:

- **Dark Mode:** Ein optionaler Dunkelmodus würde die Zugänglichkeit und visuelle Ergonomie weiter verbessern.
- **Mehrsprachigkeit:** Eine sprachliche Erweiterung (z. B. Englisch, Türkisch, Bosnisch) könnte die Nutzung für alle Schüler:innen erleichtern.
- **Barrierefreiheit (WCAG):** Eine noch stärkere Ausrichtung an Barrierefreiheitsstandards würde die Anwendung inklusiver machen.
- **Mobile App:** Eine native App-Umsetzung für Android/iOS wäre denkbar – auf Basis der bestehenden React-Komponenten.

Zusammenfassend lässt sich sagen: **Gutes UI/UX ist kein Extra – es ist der Schlüssel zu erfolgreicher Software.** Das Projekt FindMe hat bewiesen, dass schon einfache Maßnahmen große Wirkung erzielen können, wenn man den Menschen ins Zentrum der Entwicklung stellt.

# Datenvisualisierung im Web

Verfasser: Nadrasky Dominik

Klasse: 6CAIF

Betreuer: Wenz Rene

Jahrgang: 2024/25

HTBLVA, Wien V, Spengergasse

Höhere Lehranstalt für Informatik

Ausbildungsschwerpunkt Informatik für Erwachsene

Abgabedatum: 27.06.2025

## Inhalt

1.	Einleitung .....	4
2.	Theorie.....	4
2.1.	Was ist Datenvisualisierung? .....	4
2.2.	Relevanz der Datenvisualisierung im Web .....	5
2.2.1.	Visualisierung mit React .....	5
2.3.	Arten der Datenvisualisierung.....	5
2.3.1.	Statistische Diagramme.....	5
2.3.2.	Zeitreihen und Verlaufsdarstellungen .....	7
2.4.	Interaktive Datenvisualisierung .....	7
2.4.1.	Statische Visualisierungen.....	8
2.4.2.	Interaktive Visualisierungen .....	8
2.5.	Technologische Grundlagen.....	8
2.5.1.	API-gestützte Datenbereitstellung .....	8
2.5.2.	JSON als Standardformat.....	9
2.6.	Gestaltung und Wirkung von Visualisierungen.....	10
2.6.1.	Designprinzipien.....	10
2.7.	Herausforderungen bei großen Datenmengen .....	10
2.7.1.	Techniken zur Optimierung .....	11
3.	Praktische Umsetzung.....	11
3.1.	Einleitung.....	11
3.1.1.	FindMe Aufbau.....	12
3.2.	Visualisierende Komponenten im Web-Frontend .....	12
3.2.1.	Darstellung von Fundstücken .....	12
3.2.2.	Interaktive Filtermöglichkeiten .....	14
3.2.3.	Live-Anzeige .....	15
3.3.	Backend-Struktur zur Datenbereitstellung .....	16
3.3.1.	REST-Endpunkte für Datenzugriff .....	16
3.3.2.	Modellstruktur .....	18
3.3.3.	Filterlogik im Frontend .....	19

3.4.	Benutzerbezogene Visualisierung und Adminfunktionen.....	20
3.4.1.	Anzeigen & Löschen in der Adminansicht.....	20
4.	Lessons Learned .....	21
5.	Fachbegriffe .....	22
6.	Quellen und Literaturverzeichnis .....	24
6.1.	Abbildungsverzeichnis .....	24
6.2.	Abbildungsquellen .....	25
6.3.	Literaturverzeichnis .....	26

# 1. Einleitung

Die vorliegende projektspezifische fachwissenschaftliche Diplomarbeit handelt von Datenvisualisierung im Web im Zusammenhang der Entwicklung eines digitalen Fundbüros.

**Des Weiteren sind diese Aspekte zu klären:**

- Welche Technologien und Tools werden zur Datenvisualisierung im Web häufig verwendet?
- Welche gängigen Methoden und Visualisierungstechniken eignen sich besonders gut für die Darstellung von Daten im Web?
- Wie unterscheiden sich interaktive von statischen Datenvisualisierungen und welche Vorteile bietet interaktive Visualisierung?
- Wie kann die Auswahl einer Visualisierungsform die Verständlichkeit und Aussagekraft der präsentierten Daten beeinflussen?
- Welche Herausforderungen stellt die Darstellung großer Datenmengen dar, und wie können sie durch Optimierungstechniken gelöst werden?

## 2. Theorie

### 2.1. Was ist Datenvisualisierung?

Datenvisualisierung umfasst die graphische Darstellung von Informationen, um Erkenntnisse zu gewinnen und Entscheidungsprozesse zu unterstützen. Dies kann durch Diagramme, Karten oder interaktive Visuals erfolgen. Ziel ist es, Datenmuster, Trends oder Ausreißer schnell verständlich zu machen.

Neben den komplexeren Diagrammen gibt es auch einfachere und grundlegende Visualisierungsformen, die in vielen Anwendungsbereichen sehr effektiv sind. Diese einfachen Visualisierungen bieten eine schnelle Möglichkeit, Daten auf verständliche Weise darzustellen, wie herkömmliche Auflistungen.

## 2.2. Relevanz der Datenvisualisierung im Web

Webanwendungen stellen Informationen nicht nur in Textform da. Dank modernen Technologien, wie das React Framework, lassen sich interaktive Grafiken oder andere Visualisierungen direkt im Browser rendern. Dies erlaubt dem User, die Daten leicht zu verstehen und die Anwendung dynamisch zu nutzen. Selbst einfache Formen der Darstellung, wie Listen oder Aufzählungen, sind essenziell für Großteils des Internets.

### 2.2.1. Visualisierung mit React

Es lassen sich mit wenigen Code Zeilen, gut dargestellte Grafiken erstellen, die mit dem Nutzer oder Nutzerin interagieren. Wenn der Mauszeiger über bestimmte Bereiche oder Felder bewegt wird, erscheinen weitere Informationen oder Felder. Als Beispiel, lässt sich folgende interaktive Visualisierung mittels React erstellen:

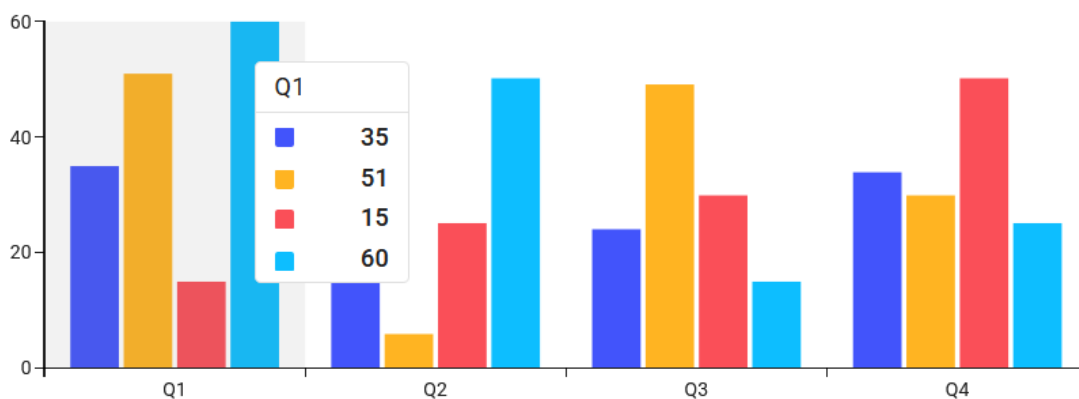


Abbildung 1: Interaktives Balkendiagramm

## 2.3. Arten der Datenvisualisierung

### 2.3.1. Statistische Diagramme

Statistische Diagramme werden gerne verwendet um so einfach und effizient wie möglich, Informationen zu vermitteln. Zu diesen gehören, als Beispiel, folgende klassische Visualisierungen wie:

- Balken- und Säulendiagramme
- Liniendiagramme
- Kreisdiagramme

Beispiel:

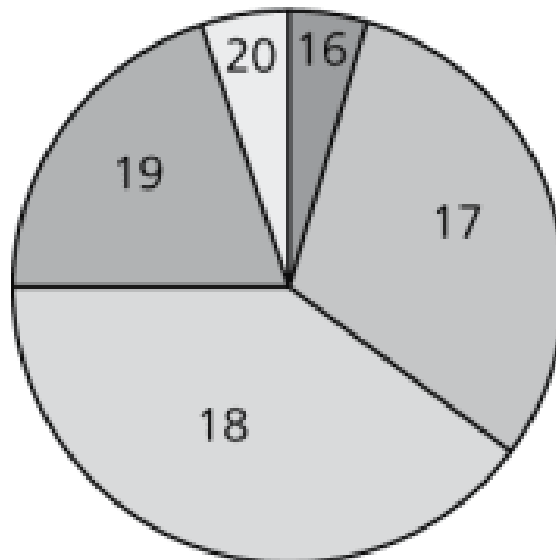


Abbildung 2: statisches Kreisdiagramm

Beispiel:

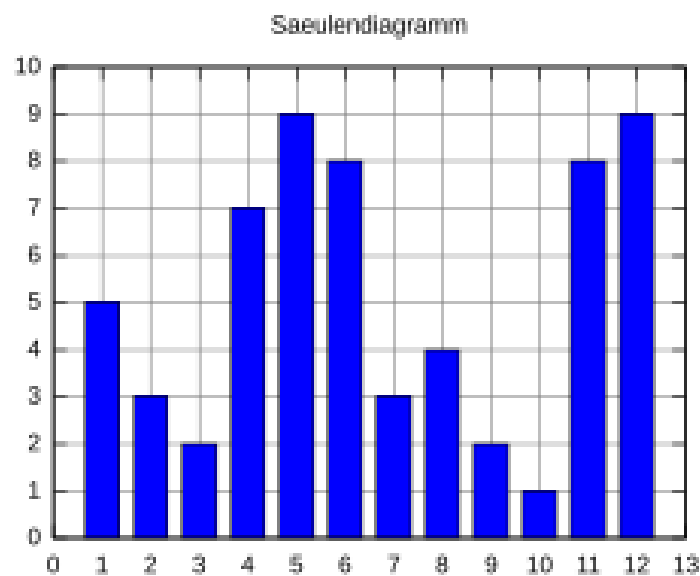


Abbildung 3: statisches Säulendiagramm

### 2.3.2. *Zeitreihen und Verlaufsdarstellungen*

Zu dieser Art der Datendarstellung gehören meistens nur statische Visualisierungstechniken. Es wird mit Daten im Hintergrund gearbeitet, um ein „Ergebnis“ präsentieren zu können oder auch nur bestehende Daten wie ein Verlauf dargestellt.

Als passendes Beispiel kann man das FindME Projekt betrachten. Dort wird laufend visualisiert, wie viele Gegenstände derzeit in der Datenbank als „Verloren“ eingetragen sind.

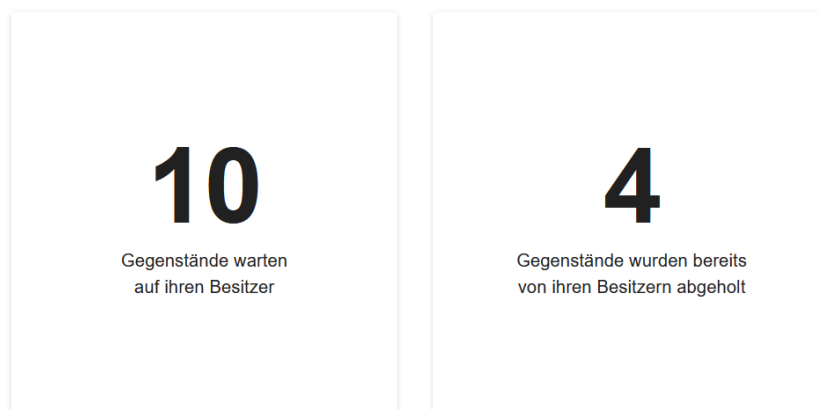


Abbildung 4: FindME Gegenstände Anzeigen

## 2.4. Interaktive Datenvisualisierung

Gegenüber der statischen Datenvisualisierung ist eine interaktive Visualisierung viel Benutzerfreundlicher. Der User kann meist selbst entscheiden welche Informationen er genau aus dem Diagramm oder anderen Darstellungsformen lesen möchte.

Bei der Gegenstandsauflistung, auf der FindME Webseite, kann der Nutzer oder die Nutzerin beispielsweise aktiv mit den präsentierten Daten interagieren. Er darf nach Kategorien und Fundorte filtern und so die Visualisierung nach seinen Interessen verändern.



### 2.4.1. *Statische Visualisierungen*

- einmalig generiert (z. B. Screenshot oder Bilddatei)
- geringere Komplexität, aber weniger flexibel
- ideal für einfache Präsentationen
- hohe Sicherheit

### 2.4.2. *Interaktive Visualisierungen*

- dynamisch, reagieren auf Nutzereingaben
- Zoom, Filtern, Mouseover, erweiterte Informationen
- insbesondere in Webprojekten wie **FindME** sehr vorteilhaft, da Benutzer filtern oder einzelne Datenbereiche untersuchen können

## 2.5. Technologische Grundlagen

React oder andere Frameworks bieten schon selbst viele Möglichkeiten für statische, aber auch interaktive Visualisierungen. Es lassen sich auch einfach, weitere Visualisierungsbibliotheken integrieren wie Chart.js, D3.js, Recharts, Google Charts oder viele weitere. Mit diesen ist es möglich, Daten nach seinen Wünschen zu präsentieren.

### 2.5.1. *API-gestützte Datenbereitstellung*

Das Backend stellt Daten, als Beispiel, über [/api/statistics/byMonth](#) bereit. Diese Endpunkte aggregieren Daten so, dass sie sofort von Visualisierungsbibliotheken genutzt werden können.

Die Daten sind meistens als JSON-Objekt formatiert und werden so übergeben.

### 2.5.2. *JSON als Standardformat*

In der Regel wird mit dem JSON-Format gearbeitet, was mit Schlüssel-Wert-Paaren aufgebaut ist.

JSON steht für JavaScript Object Notation und ist heute das Standardformat für den Datenaustausch zwischen Server und Client, besonders im Web. Es ist leichtgewichtig, menschenlesbar, gut strukturiert und sehr gut mit JavaScript-basierten Frameworks (wie React) kompatibel.

Im Kontext der Datenvisualisierung ist JSON das verbindende Element zwischen dem Backend, das Daten bereitstellt, und dem Frontend, dass diese Daten visuell aufbereitet. Egal, ob es sich um einfache numerische Werte oder komplexe Objektstrukturen handelt. JSON stellt eine Struktur zur Verfügung, um alle Arten von Informationen zu übertragen.

*Beispiel JSON-Objekt:*

```
[  
{ "monat": "Januar", "anzahl": 12 },  
{ "monat": "Februar", "anzahl": 9 },  
{ "monat": "März", "anzahl": 17 },  
]
```

## 2.6. Gestaltung und Wirkung von Visualisierungen

Die Wahl der richtigen Darstellungsmöglichkeit ist entscheidend für viele Aspekte, wie Aussagekraft oder Verständlichkeit. Dabei kann man grundlegende Regeln einhalten, um die am besten passende Visualisierung zu nutzen. Wenn man Verteilungen zeigen will, nutzt man oft Kreisdiagramme, aber auch Balkendiagramme erzielen häufig dieselbe Aussagekraft. Doch wenn man, als Beispiel, einen Zeitverlauf präsentieren möchte, sollte man ein Liniendiagramm verwenden, ein Balkendiagramm würde hier nicht passend sein.

### 2.6.1. *Designprinzipien*

Wichtig bei der Gestaltung der Visualisierungen ist jeder Teil der Darstellung, doch meist erzielen die einfachsten Präsentationsmittel, die beste Wirkung. Doch selbst die falsche Farbe könnte einen Nutzer oder einer Nutzerin daran hindern, Informationen wie gewünscht oder auch schnell zu verstehen.

#### **Prinzipien:**

- Farbcodierung, Lesbarkeit, Kontraste
- Tooltips für Details
- Reduktion auf das Wesentliche

## 2.7. Herausforderungen bei großen Datenmengen

Bei großen Datenmengen oder Datenpunkten entstehen schnell Performance- und Darstellungsprobleme. Übliche Problematiken, die auftreten werden:

- Browserleistung sinkt
- Unübersichtliche Darstellungen („Datenüberflutung“)
- Lange Ladezeiten bei API-Aufrufen

### 2.7.1. *Techniken zur Optimierung*

Auf fast jeder Webseite, die mit größeren Datenmengen arbeitet, werden Techniken zur Optimierung eingesetzt, um das Nutzererlebnis performant zu halten.

Selbst einfachere Techniken können schon vieles bewirken. Bei den meisten Darstellungen lassen sich „Verbesserungen“ oder „Vereinfachungen“ vornehmen, wie standardmäßig nur wenige Daten zu zeigen und den Nutzer oder Nutzerin selbst interaktiv, die gewollten Daten auszuwählen oder zu filtern. Auch kann man oft Informationen summieren und zusammenfassen, was die Leistung erheblich verbessern kann.

Komplexere Techniken wie „Lazy Loading“ oder serverseitige Filterung sind stark optimierende Möglichkeiten. Man kann „harte“ Beschränkungen direkt am Server einstellen, so dass, als Beispiel, nur mehr Daten von diesem Jahr gezeigt werden.

Als „Lazy Loading“ bezeichnet man die moderne Technik, nur die Daten zu laden die auch im Sichtbereich sind.

## 3. Praktische Umsetzung

### 3.1. Einleitung

Die Webanwendung *FindME* dient der Erfassung, Verwaltung und Suche von Fundgegenständen in einer Bildungseinrichtung. User können Fundstücke einsehen und zurückfordern, Administratoren pflegen die Daten.

In diesem Projekt sind auch mehrfach erwähnte Visualisierungstechniken genutzt worden, um dem Nutzer oder der Nutzerin der Anwendung, die nötigen Informationen bestmöglich zu vermitteln.

### **3.1.1. FindMe Aufbau**

#### **Backend:**

Das Backend von FindME wurde mit ASP.NET entwickelt und klassische Entwicklungsstrukturen eingehalten. Es wurde eine REST-API mit Endpunkten für Items, Kategorien, Räume und auch Benutzerverwaltung implementiert und eine Datenbankverbindung mittels Entity Framework Core verwendet.

#### **Frontend:**

Das Frontend ist mittels React.js mit MUI (Material UI) geschrieben worden. Durch Komponentenbasierte UI, haben wir all unsere nötigen Visualisierungen erstellen können. Dazu gehören einfache Auflistungen, aber auch Eingabemasken, Filterfelder oder Dialoge.

#### **Format:**

Als Schnittstelle zwischen dem Frontend und dem Backend wurde JSON genutzt. Beide Bereiche können mit diesem Format arbeiten und so optimal die Daten anzeigen, verändern oder zu speichern.

## **3.2. Visualisierende Komponenten im Web-Frontend**

In der Anwendung FindME wurden keine klassischen Diagramme oder Charts verwendet, um Informationen zu liefern. Stattdessen nutzen wir interaktive Listen, Filter oder Item Cards zur Darstellung.

### **3.2.1. Darstellung von Fundstücken**

Die Liste der Fundstücke wurde mit dynamischen Item-Kacheln (Item-Cards) aufgebaut. Diese enthalten jeweils immer ein Bild, Titel, Kategorie, Fundort und eventuell eine Beschreibung. Bei einem Klick auf eines der Gegenstand-Kacheln, öffnet sich ein Detaildialog (vergrößerte Detailansicht) mit mehr Informationen.

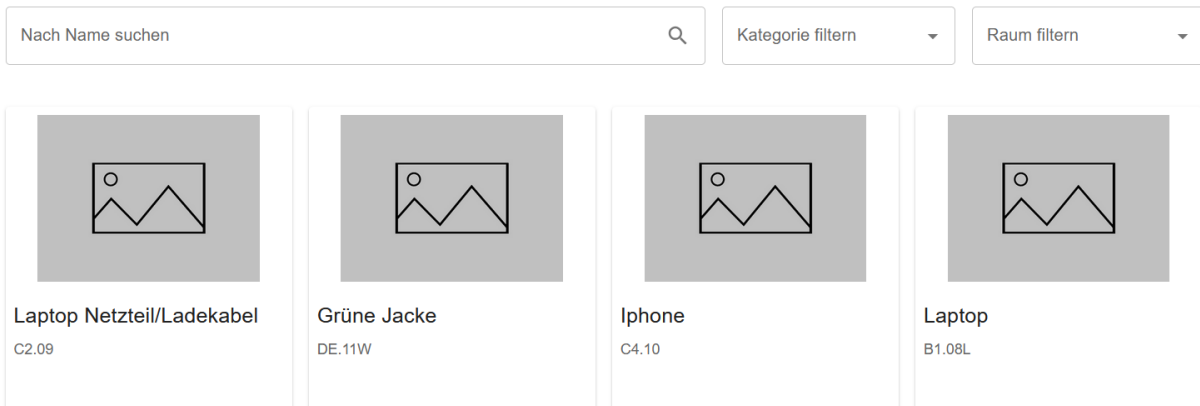


Abbildung 5: FindME Fundstücke (Testdaten)

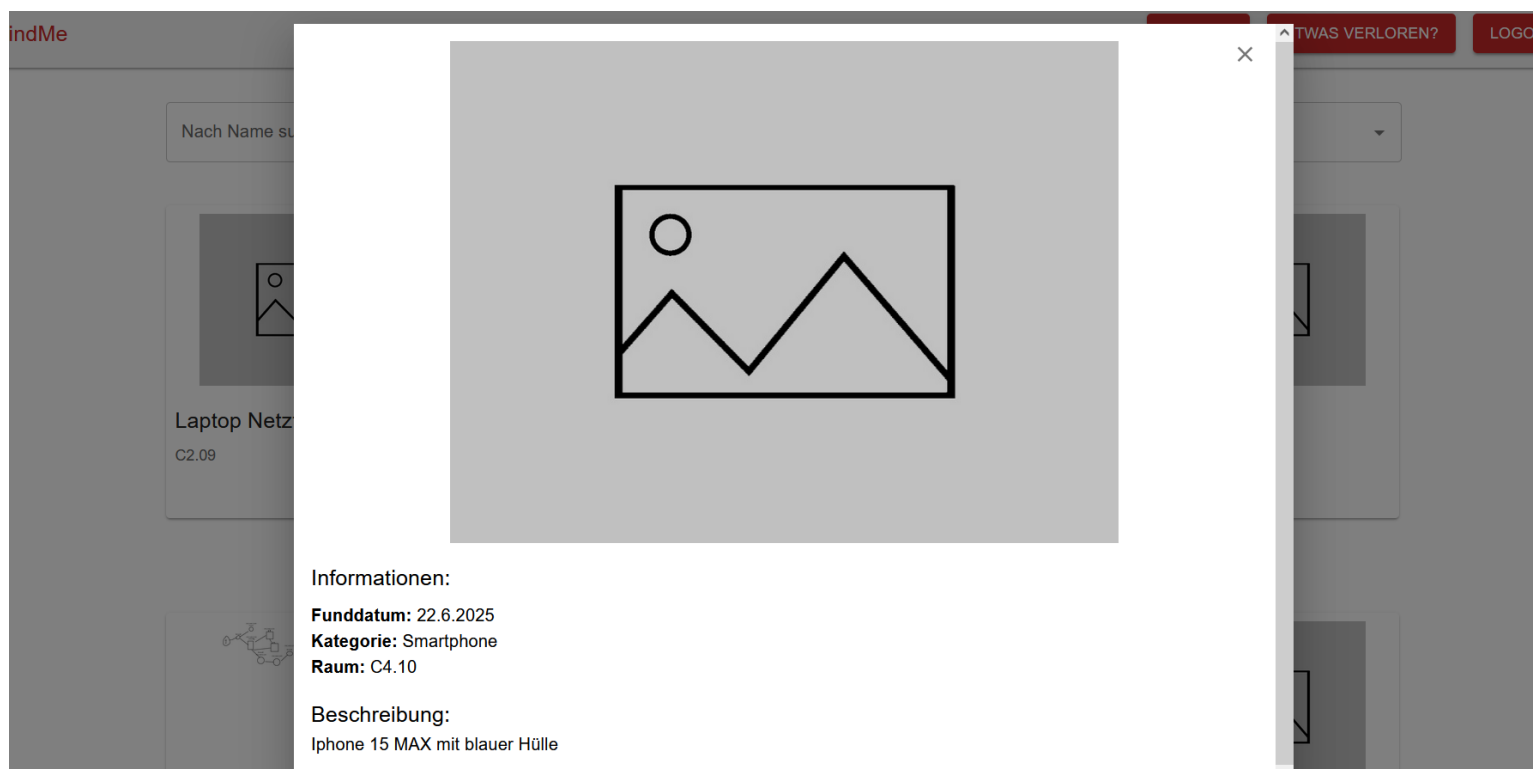


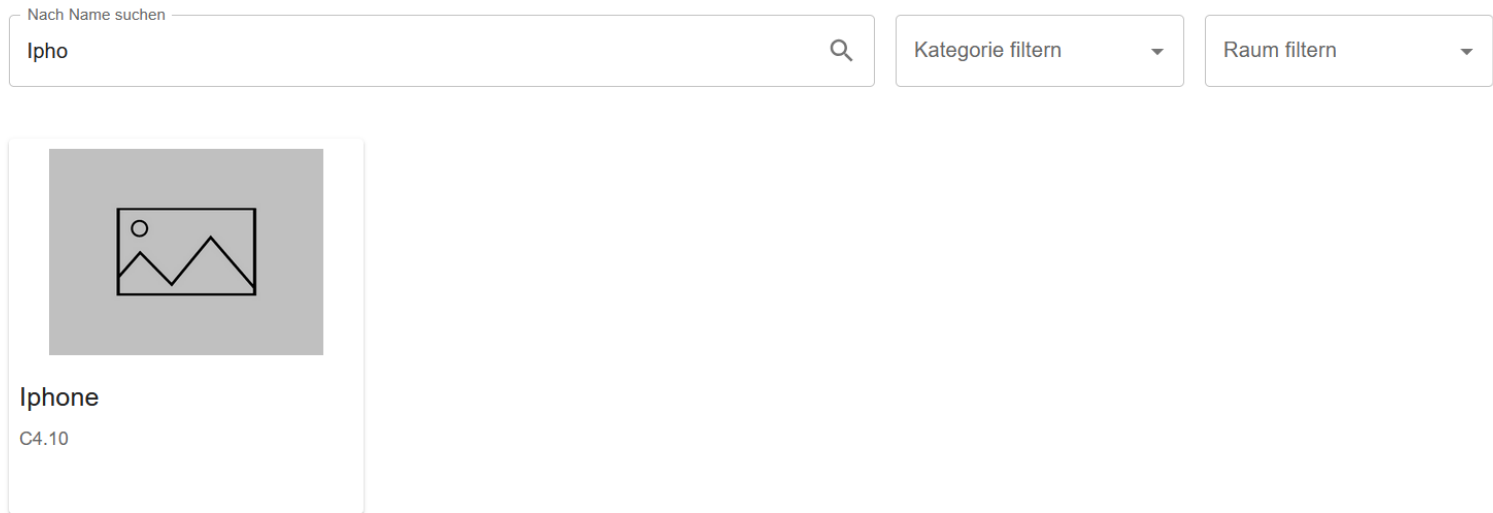
Abbildung 6: FindME Detailansicht (Testdaten)

(Da dies Testdatensätze sind, gibt es keine Bilder, was im Betrieb nicht möglich ist)

### 3.2.2. Interaktive Filtermöglichkeiten

Interaktive Filtermöglichkeiten erlauben es dem User die Fundliste nach seinen Interessen anzupassen. Dies wurde mit Client-seitiger Filterung umgesetzt.

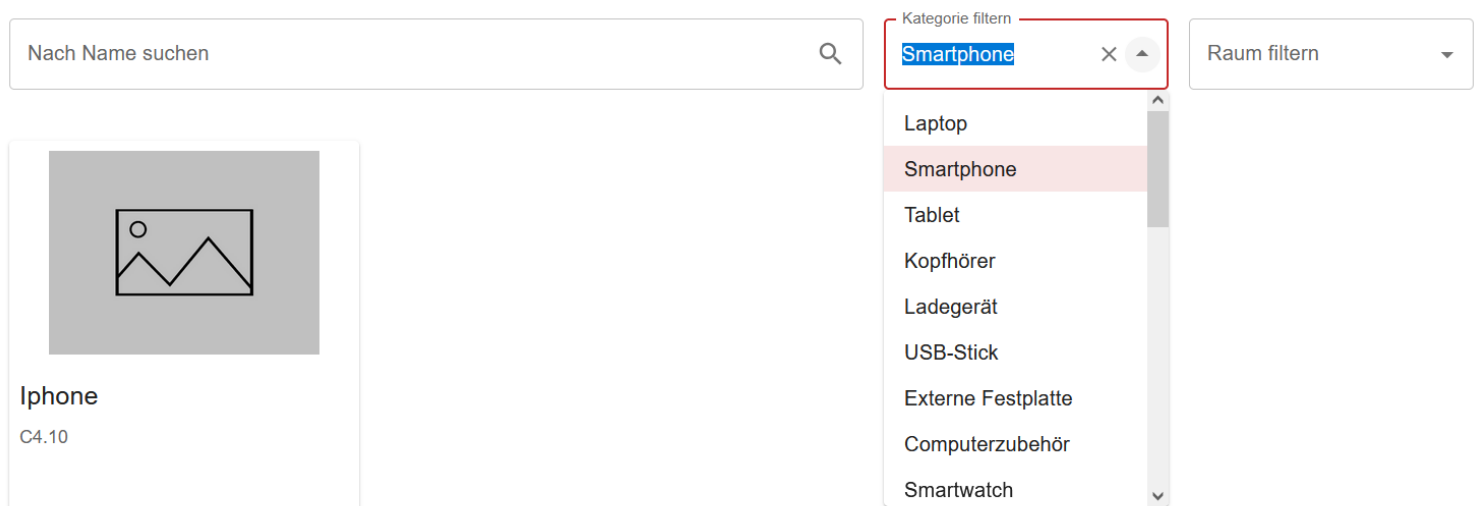
**Suche nach Name:** Textfeld, das Client-seitig die Liste filtert



The screenshot shows the top section of the FindME application. On the left, there is a search bar labeled 'Nach Name suchen' containing the text 'Ipho'. To its right are two filter buttons: 'Kategorie filtern' and 'Raum filtern'. Below the search bar, a card displays a placeholder image and the text 'Iphone' with 'C4.10' underneath.

Abbildung 7: FindME Namefilter

**Filter nach Raum oder Kategorie:** Dropdown-Listen (Autocomplete)



This screenshot shows the same interface as before, but with the 'Kategorie filtern' dropdown menu open. The dropdown list contains the following items: Laptop, Smartphone (highlighted in pink), Tablet, Kopfhörer, Ladegerät, USB-Stick, Externe Festplatte, Computerzubehör, and Smartwatch. The search bar still contains 'Ipho'.

Abbildung 8: FindME Kategoriefilter (mit Dropdown)

### 3.2.3. *Live-Anzeige*

Die Daten werden im Frontend nach einer Eingabe sofort neu dargestellt und somit laufend aktuell gehalten. Dies ist ein perfektes Beispiel für eine einfache Form von interaktiver Datenvisualisierung und zeigt ebenfalls, wie effektiv solche Darstellungen sind.

Jeder der die FindME Seite besucht, wird in den ersten Sekunden erkennen können wie viele Gegenstände derzeit im Fundbüro liegen und erlaubt es schnell Informationen auf einem Blick zu vermitteln.

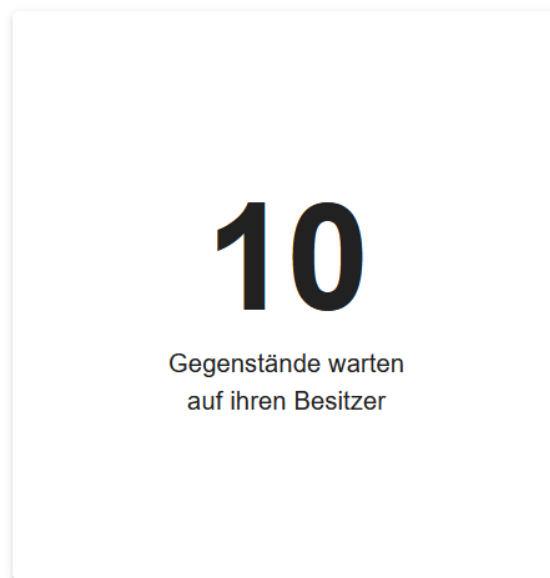


Abbildung 9: FindME Fundstücke Card



## 3.3. Backend-Struktur zur Datenbereitstellung

### 3.3.1. REST-Endpunkte für Datenzugriff

Um die Daten, die man im Frontend sieht, auch von dem Backend zu bekommen sind REST-Endpunkte nötig gewesen. Folgende Endpunkte wurden erstellt, damit von dem jeweiligen Endpunkt die korrekten Daten im JSON-Format an das Frontend geleitet werden. Folgende Endpunkte wurden implementiert und übermitteln bei einem Aufruf die JSON-Daten:

- **GET /api/item/mapped** – Alle Fundstücke mit zugehöriger Kategorie & Raum
- **GET /api/category** – Kategorien
- **GET /api/location** – Räume

```
▼ 0:
  name: "Laptop"
▼ 1:
  name: "Smartphone"
▼ 2:
  name: "Tablet"
▼ 3:
  name: "Kopfhörer"
▼ 4:
  name: "Ladegerät"
▼ 5:
  name: "USB-Stick"
▼ 6:
  name: "Externe Festplatte"
▼ 7:
  name: "Computerzubehör"
▼ 8:
  name: "Smartwatch"
▼ 9:
  name: "Taschenrechner"
▼ 10:
  name: "Elektronik"
▼ 11:
  name: "Laborausrüstung"
```

Abbildung 10: API Category Endpunkt JSON

```
▼ 0:
  name: "A1.03"
▼ 1:
  name: "A1.05"
▼ 2:
  name: "A1.06"
▼ 3:
  name: "A1.14"
▼ 4:
  name: "A1.15"
▼ 5:
  name: "A1.16"
▼ 6:
  name: "A1.17"
▼ 7:
  name: "A1.23"
▼ 8:
  name: "A2.04"
▼ 9:
  name: "A2.05"
▼ 10:
  name: "A2.06"
▼ 11:
  name: "A2.13"
▼ 12:
  name: "A2.14"
▼ 13:
  name: "A2.20"
▼ 14:
  name: "A3.04"
▼ 15:
  name: "A3.05"
▼ 16:
  name: "A3.12"
```

Abbildung 11: API Location Endpunkt JSON

```

▼ 0:
  itemId:      1
  name:        "Test Item"
  category:    "Tablet"
  location:    "A1.14"
  foundDate:   "2025-05-31T00:00:00"
  isClaimed:   false
  imagePath:   "https://localhost:51645/img/503610bf-d04e-4ff1-86d6-f75f23dabc38.JPG"
  description: "TEST"
▼ 1:
  itemId:      2
  name:        "test2"
  category:    "Ausweis/ID"
  location:    "A3.13"
  foundDate:   "2025-06-02T00:00:00"
  isClaimed:   false
  imagePath:   "https://localhost:51645/img/99a33f58-7cd4-4027-ab14-0908a1aeb09e.JPG"
  description: "test ausweis"
▼ 2:
  itemId:      3
  name:        "klein"
  category:    "Anderes"
  location:    "A1.05"
  foundDate:   "2025-06-02T00:00:00"
  isClaimed:   false
  imagePath:   "https://localhost:51645/img/45f42d32-7d65-4b11-80d2-4f4607ffd768.JPG"
  description: "test"

```

Abbildung 12: API Item/mapped Endpunkt JSON

### 3.3.2. Modellstruktur

Die Modellstruktur wurde einfach, kompakt und effizient gestaltet. Es gibt insgesamt vier Datenmodelle. Dazu gehören Category, Location und Item, wobei hier eine 1:n Beziehung besteht, damit Items den Kategorien und Fundorten zugeordnet werden können.

Es existiert ebenfalls eine Accountstruktur, wobei zwei statische Account implementiert wurden. Diese dient aber rein zur leichten Verwaltung und Authentifizierung des „Admin“ und „Verwalter“ Account.

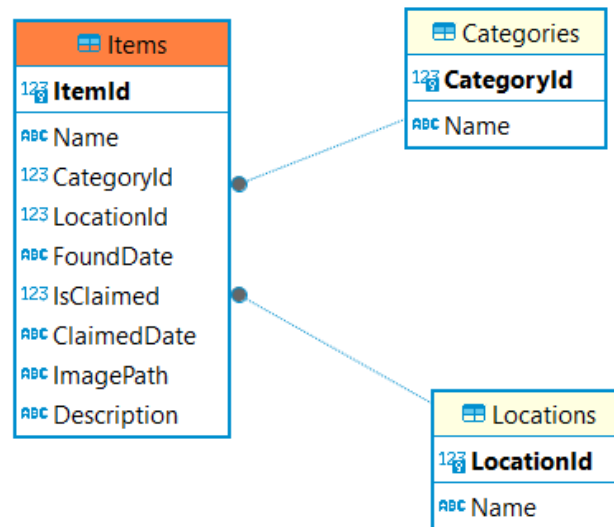


Abbildung 13: FindME DB ER-Diagramm

Die Account Klasse wird nicht als eigene Tabelle gespeichert, da die Daten statisch sind.

### 3.3.3. Filterlogik im Frontend

Die Filterlogik kam häufig zum Einsatz und wurde in mehreren React Komponenten implementiert. Als Beispiel, nenne ich hier die schon Erwähnte Fundstück-Liste, die sich durch folgende Eingabemöglichkeiten steuern lässt:

- Ein Texteingabefeld für die Namenssuche
- Ein Dropdown zur Auswahl der Kategorie
- Ein Dropdown zur Auswahl des Raums

Die Filterung erfolgt im Frontend durch folgende Logik:

```
54
55  ✓ const filteredItems = items.filter((item) => {
56      const matchesName = item.name.toLowerCase().includes(searchName.toLowerCase());
57      const matchesCategory = !selectedCategory || item.category === selectedCategory;
58      const matchesLocation = !selectedLocation || item.location === selectedLocation;
59      return matchesName && matchesCategory && matchesLocation;
60  });
```

Abbildung 14: Frontend Filterlogik

Diese drei Bedingungen (Z. 56, 57, 58) sorgen dafür, dass die Ergebnisliste reaktiv angepasst wird, sobald ein Filter gesetzt oder verändert wird.

Die Filter wirken dabei kombinatorisch, das heißt alle aktiven Filterbedingungen müssen gleichzeitig erfüllt werden, um das jeweilige Ergebnis anzuzeigen.

## 3.4. Benutzerbezogene Visualisierung und Adminfunktionen

Die statischen Accounts Admin und Verwalter haben mehr Funktionen und Zugriffe als ein üblicher User. Der Admin darf in den Adminbereich, wo auch exklusive Visualisierungen und Funktionen zur Verfügung stehen.

### 3.4.1. *Anzeigen & Löschen in der Adminansicht*

Hier ist klar die Verwaltung im Vordergrund, jedoch wurden auch hier Filterfunktionen zur erleichterten Verwaltung implementiert. Wie bei der herkömmlichen Gegenstand-Suche, sind die Filter- und Suchfelder reaktiv. Als Beispiel, kann man mit wenig Aufwand Kategorien suchen und ebenfalls mit einem Klick löschen. Dabei werden Dialogfenster aufgerufen, die das Löschen bestätigen und sicherer macht.

## Kategorien

Suchen

Sm

Smartphone



Smartwatch



[Weniger anzeigen](#)

Abbildung 15: Adminbereich Kategorien Verwaltung

Bei einem Klick auf das Lösch-Icon, öffnet sich ein Dialogfenster zur Bestätigung, dieses prüft auch noch ob Items existieren, die diesen Wert noch in der Datenbank verwenden. Falls dies der Fall ist, wird der Löschvorgang nicht zugelassen.

## 4. Lessons Learned

Ich habe sehr viele neue Dinge lernen können und dass nicht nur in Technischer Hinsicht. Das allgemeine Entwickeln von einem Backend, API's und Datenbanken bzw. wie diese miteinander arbeiten oder in Verbindung stehen, verstehe ich jetzt weitreichend. Vor dem Projekt habe ich rückblickend nicht wirklich verstanden, was es heißt, wenn das Frontend mit dem Backend interagiert oder was genau eigentlich mit den Endpunkten passiert. Nun bin ich in der Lage, eine moderne Backendstruktur aufzubauen, die nötigen Methoden, Klassen, aber auch grundlegende Backend Konfigurationen zu implementieren.

Auch habe ich sehr viel lernen können, wie man unter „realen“ Bedingungen in einem Team arbeitet und ein Projekt umsetzt. Die Schwierigkeiten einer guten Planung, Umsetzung und Absprache, zeigten mir was wirklich wichtig bei so einer Gruppenarbeit ist und worauf in der Zukunft mehr bzw. weniger Wert legen sollte.

## 5. Fachbegriffe

**Frontend:** Der Teil einer Webanwendung, der für den Benutzer sichtbar und zugänglich ist. Es umfasst das Design und die Benutzeroberfläche (UI), die direkt mit dem Nutzer interagiert.

**Backend:** Der Teil einer Webanwendung, der für die Logik, die Datenverarbeitung und die Kommunikation mit der Datenbank zuständig ist. Das Backend stellt APIs zur Verfügung und verarbeitet die Anfragen vom Frontend.

**API:** Eine Schnittstelle, die es verschiedenen Softwarekomponenten ermöglicht, miteinander zu kommunizieren. APIs stellen Funktionen bereit, die von externen Programmen (wie dem Frontend) genutzt werden können, um Daten zu erhalten oder zu senden.

**REST-API:** Eine spezifische Art von API, die auf den Prinzipien von REST (Representational State Transfer) basiert und Webanwendungen eine einfache und flexible Möglichkeit bietet, mit anderen Anwendungen zu kommunizieren.

**Datenmodell:** Eine Struktur, die beschreibt, wie Daten gespeichert und organisiert werden.

**Mappen:** Der Prozess des Zuordnens oder Abgleichens von Daten aus verschiedenen Quellen oder Formaten.

**React.js:** Eine JavaScript-Bibliothek zur Erstellung von Benutzeroberflächen, die hauptsächlich für das Frontend verwendet wird. React ermöglicht es, wiederverwendbare UI-Komponenten zu erstellen, die mit dem Zustand und den Daten einer Anwendung interagieren.

**Material UI (MUI):** Ein React-Komponenten-Framework, das eine Sammlung vorgefertigter UI-Komponenten bietet, die den Material Design-Richtlinien von Google folgen. Es erleichtert das Erstellen von Benutzeroberflächen, indem es Design-Komponenten für Buttons, Dialoge, Formulare usw. bereitstellt.

**Lazy Loading:** Eine Technik zur Optimierung der Ladegeschwindigkeit von Webseiten, bei der nur die Daten geladen werden, die sich im sichtbaren Bereich der Webseite befinden. Alle anderen Daten werden erst nach Bedarf nachgeladen.

**Entity Framework Core:** Ein Open-Source ORM (Object-Relational Mapping) Framework für .NET, das die Interaktion mit einer relationalen Datenbank vereinfacht.

**Item-Karten (Item Cards):** Eine Art der Darstellung von Fundstücken im Frontend, die relevante Informationen wie Bild, Titel, Kategorie und Fundort enthalten. Diese Karten sind interaktiv und ermöglichen es dem Benutzer, mehr Details durch Klicken zu sehen.

**Datenüberflutung (Data Overload):** Ein Zustand, bei dem eine große Menge an Daten zu einer Überforderung des Nutzers führen kann, was die Interpretation der Daten erschwert.



## 6. Quellen und Literaturverzeichnis

### 6.1. Abbildungsverzeichnis

Abbildung 1: Interaktives Balkendiagramm.....	5
Abbildung 2: statisches Kreisdiagramm.....	6
Abbildung 3: statisches Säulendiagramm .....	6
Abbildung 4: FindME Gegenstände Anzeigen .....	7
Abbildung 5: FindME Fundstücke (Testdaten) .....	13
Abbildung 6: FindME Detailansicht (Testdaten) .....	13
Abbildung 7: FindME Namefilter .....	14
Abbildung 8: FindME Kategoriefilter (mit Dropdown) .....	14
Abbildung 9: FindME Fundstücke Card .....	15
Abbildung 10: API Category Endpunkt JSON .....	16
Abbildung 11: API Location Endpunkt JSON .....	17
Abbildung 12: API Item/mapped Endpunkt JSON .....	18
Abbildung 13: FindME DB ER-Diagramm .....	19
Abbildung 14: Frontend Filterlogik .....	20
Abbildung 15: Adminbereich Kategorien Verwaltung.....	21

## 6.2. Abbildungsquellen

Abbildung 1: <https://mui.com/x/react-charts/>

Abbildung 2: <https://learnattack.de/schuelerlexikon/mathematik/statistische-diagramme>

Abbildung 3: <https://de.wikipedia.org/wiki/S%C3%A4ulendiagramm>

Abbildung 4: Von FindME Projekt

Abbildung 5: Von FindME Projekt

Abbildung 6: Von FindME Projekt

Abbildung 7: Von FindME Projekt

Abbildung 8: Von FindME Projekt

Abbildung 9: Von FindME Projekt

Abbildung 10: Von FindME Projekt

Abbildung 11: Von FindME Projekt

Abbildung 12: Von FindME Projekt

Abbildung 13: Von FindME Projekt

Abbildung 14: Von FindME Projekt

Abbildung 15: Von FindME Projekt

## 6.3. Literaturverzeichnis

Datenvisualisierung (17.06.2025) Von <https://www.ibm.com/de-de/topics/data-visualization>

Designprinzipien. (kein Datum). Von <https://www.ibm.com/de-de/topics/rest-api> abgerufen

MUI (2025 08 03) Von <https://mui.com/x/react-charts/>

IT, R. (10. 11 2022). Frontend vs. Backend Development: Breaking Down the Difference. Von <https://medium.datadriveninvestor.com/frontend-vs-backend-development-breaking-down-the-difference-f9b1c09ae8ff> abgerufen



# Themenstellung von Sebastian Stiller

Vergleich klassischer und agiler Projektmanagement-Methoden: Effizienz, Flexibilität und Praxistauglichkeit in modernen IT-Projekten

## Einleitung

In unserer heutigen Welt, in der wir uns als Menschen immer schneller weiterentwickeln müssen, ist es immer wichtiger, nicht nur für Unternehmen, sondern für alle Menschen, ihre Projekte so effizient und flexibel wie möglich zu organisieren. Stetig wechselnde Anforderungen und Umstände machen es nötig, dass die Art, wie ein Projekt organisiert wird, ebenfalls immer flexibler und effizienter wird. Hierdurch stellt sich immer wieder die Frage: Was ist heutzutage besser? Klassisches hierarchisches oder modernes und agiles Projektmanagement?

## Klassisches Projektmanagement: Das Wasserfallmodell

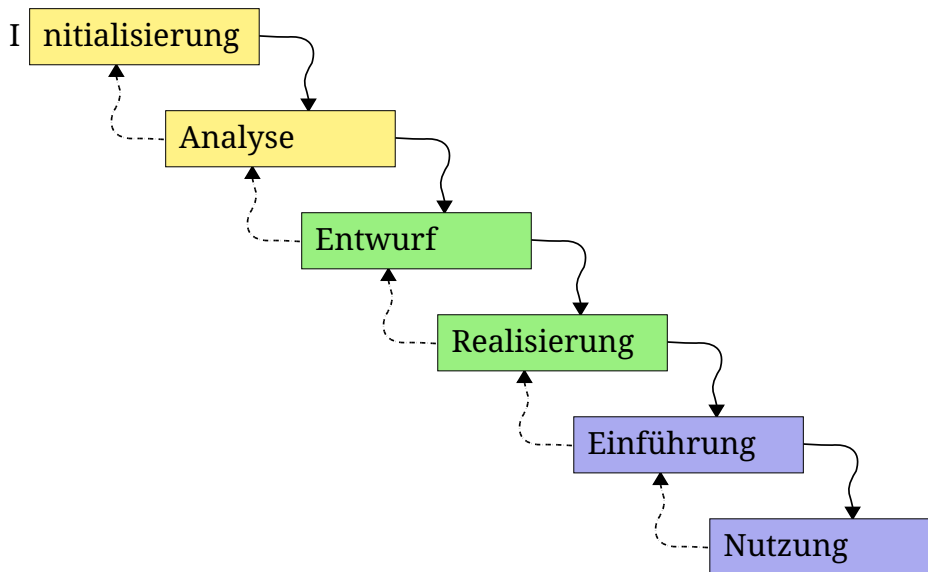
Im klassischen Projektmanagement sind die Rollen klar geregelt und jeder hat seine Aufgabe im Projekt. Eines der berühmtesten Modelle zum "**klassischen**" Projektmanagement ist das Wasserfallmodell (vgl. [1](#))

### Beschreibung der Methode

Bei dem Wasserfallmodell handelt es sich um ein sequentielles Vorgehensmodell, welches in Phasen eingegliedert ist. Das Modell ist sequenziell aufgebaut, gemeint ist damit, dass die einzelnen Phasen des Projektes größtenteils nacheinander abgearbeitet werden müssen, um das Projekt erfolgreich abzuschließen. Das Modell wird deshalb als Wasserfall bezeichnet, da die Darstellung einem Wasserfall sehr stark ähnelt. (vgl. [2](#))

### Phasen des Wasserfallmodells (vgl. [4](#))

Das Wasserfallmodell besteht aus sechs Phasen, welche sequentiell durchlaufen werden müssen. Die Phasen sind:



(img. 1)

## Anforderungsanalyse

In dieser Phase werden die Anforderungen an das Projekt gesammelt und dokumentiert. Wichtig ist, dass alle Anforderungen und Ziele des Projektes und die Organisation klar, deutlich und für alle Parteien verständlich formuliert sind um Missverständnisse oder Interpretationsspielraum nicht aufkommen zu lassen.

Eine gute Herangehensweise ist es, die Ziele des Projektes mit Hilfe des SMART-Abfrageschemas zu formulieren.

**Spezifisch:** Die Ziele sollten klar und eindeutig formuliert sein.

**Messbar:** Die Ziele sollten eine messbare Komponente enthalten, um einen Projektfortschritt erfassen zu können

**Attraktiv:** Die Ziele sollten für alle Beteiligten attraktiv und motivierend sein.

**Realistisch:** Die Ziele sollten realistisch und erreichbar sein.

**Terminiert:** Die Ziele sollten einen klaren Zeitrahmen haben, um den Fortschritt des Projektes zu verfolgen.

(vgl. 3)

## Entwurf

In dieser Phase wird das Konzept für das Projekt erstellt. Hierbei werden die Anforderungen und Ziele des Projektes in ein machbares Konzept umgesetzt. Zuerst wird eine Machbarkeitsanalyse durchgeführt um zu prüfen, ob die Ziele des Projektes mit den vorhandenen Ressourcen erreicht werden kann. Anschließend wird mit diesen Ergebnissen ein Konzept erstellt, welches die Anforderungen und Ziele enthält, welche mit einer höheren Sicherheit erreicht werden können.

## Ressourcen

Ressourcen sind generell alle Mittel, die für die Erfüllung eines Projektes zur Verfügung stehen. Diese Ressourcen werden generell in drei Kategorien eingestuft:

- \* **Personelle Ressourcen:** Alle Personen, die an dem Projekt beteiligt sind. Hierzu zählen sowohl die Projektmitarbeiter, als auch die Stakeholder und das Management.
- \* **Sachliche Ressourcen:** Alle Werkzeuge, Maschinen, Materialien, Infrastruktur und Software-Tools, welche für eine reibungslose Abwicklung des Projektes zur Verfügung stehen.
- \* **Finanzielle Ressourcen:** Das ist das Budget, also das Geld, welches oft vom Auftraggeber oder oberen Management zur Verfügung gestellt wird, um das Projekt zu finanzieren. Hierbei ist es wichtig, dass das Budget immer eher höher angesetzt wird, als eigentlich erforderlich ist, da man nie genau von Anfang an mit 75%iger Sicherheit sagen kann, ob nicht etwas unvorhergesehenes eintritt und das Budget übersteigt. Eine Regel an dieser Stelle ist grundsätzlich Murphy's Gesetz. Dieses beschreibt, dass auch wenn man glaubt alles bedacht zu haben, wird immer etwas passieren, was die Planung über den Jordan gehen lässt.

## Implementierung

Das ist die eigentliche Arbeit am Projekt. Hier werden die Anforderungen und Ziele des Projektes nach den Qualitätsstandards des Projektes so umgesetzt, wie es im Konzept vom Auftraggeber abgesegnet wurde. In dieser Phase wird also das Produkt erstellt, welches am Ende des Projektes vom Auftraggeber abgenommen und an den Kunden übergeben wird. Hier ist es wichtig regelmäßige Meetings mit dem Auftraggeber und den Stakeholdern zu haben, um den Fortschritt des Projektes zu überwachen und gegebenenfalls Maßnahmen zu ergreifen, wenn etwas nicht nach Plan läuft. Ein gutes Hilfsmittel hierfür ist der Statusbericht mit einer Meilenstein-trendanalyse. Hierbei wird der Fortschritt des Projektes anhand der Meilensteine des Projektes dokumentiert und es wird geschaut, ob die Meilensteine im Zeitplan liegen oder ob es zu Verzögerungen oder Vorsprüngen kommt.

## Test

In dieser Phase wird das Produkt, welches in der Implementierungsphase erstellt worden ist, auf die funktionalen und qualitativen Anforderungen des Projektes getestet. Hier wird geprüft, ob das Produkt den Anforderungen und Zielen des Projektes, welche vom Auftraggeber vorgegeben wurden, entspricht. Hierbei ist es wichtig, dass die verschiedenen Abschnitte des Produktes einzeln getestet werden und anschließend das gesamte Produkt und das Zusammenspiel der einzelnen Abschnitte miteinander getestet wird. Dies wird oft in der Form von Komponententests, Integrationstests und Systemtests durchgeführt. Zu jedem dieser Tests muss ein Testprotokoll erstellt werden, welches die Anforderungen und Ergebnisse der Tests dokumentiert. Diese Protokolle sind wichtig, um später nachweisen zu können, dass alle Anforderungen und Qualitätsstandards des Produktes, welche vom Auftraggeber vorgegeben wurden, erfüllt sind. Dies ist wichtig, um sagen zu können, dass das Produkt bei der Übergabe an den Kunden keine Mängel aufweist.

## Abnahme

Bei der Abnahme wird das Produkt vom Auftraggeber auf die Erfüllung der Qualitätsanforderungen und der Ziele des Projektes geprüft. Hierbei wird das Produkt auf die Funktionalität, die Qualität und die Erfüllung der Anforderungen getestet. Wenn das Produkt diesen Anforderungen entspricht, wird es vom Auftraggeber abgenommen und zur Inbetriebnahme oder zur Übergabe an den Kunden freigegeben. Hierbei ist es wichtig, ein Abnahmeprotokoll zu erstellen, welches dokumentiert, welche Anforderungen und Qualitätsmerkmale das Produkt aufweisen muss, um abgenommen zu werden. Des Weiteren wird in diesem Protokoll dokumentiert, welche Anforderungen und Qualitätsmerkmale das Produkt tatsächlich erfüllt oder nicht erfüllt. Abschließend wird das Protokoll vom Auftraggeber und dem Projektleiter unterschrieben, um von beiden Parteien zu bestätigen, dass das Produkt den Anforderungen und Qualitätsmerkmalen eben entspricht oder nicht entspricht und welche Maßnahmen gegebenenfalls ergriffen werden müssen, um das Produkt den Anforderungen und Qualitätsmerkmalen anzupassen und dadurch die Abnahme erfolgreich durchzuführen. Ist dies der Fall, wird das Produkt an den Kunden übergeben und das eigentliche Projekt gilt als abgeschlossen.

## Inbetriebnahme

In dieser Phase, welche grundsätzlich nicht mehr in die eigentliche Projektarbeit fällt, wird das Produkt in Betrieb genommen und kommt somit in den produktiven Einsatz. Hierbei kommt es oft zu Einschulungen und Workshops, um die Mitarbeiter des Kunden mit dem Produkt vertraut zu machen und die mögliche Umstellung von einem alten Produkt auf das neue zu gewährleisten. Hierbei ist es wichtig, dass die Mitarbeiter des Kunden das neue Produkt verstehen und damit arbeiten können. Oftmals werden auch Schulungsunterlagen erstellt, um den Mitarbeitern des Kunden die Arbeit mit dem neuen Produkt zu erleichtern.

(vgl. 4)

## Rollen und Verantwortlichkeiten

Im klassischen Projektmanagement sind die Rollen und Verantwortlichkeiten klar und Hierarchisch geregelt. Grundsätzlich gibt es neun Rollen, welche in einem "klassisch" geführtem Projekt vorkommen können. Hier ist jedoch zu beachten, dass viele dieser Rollen bei kleineren Projekten nicht vorkommen oder vom Projektleiter übernommen werden. Diese Rollen sind:

- \* **Projektauftraggeber:** Der AuftraggeberIn ist die Person oder Organisation, die das Projekt in Auftrag gibt und die finanziellen Mittel bereitstellt. Er ist derjenige der als einziger sagen kann ob das Projekt geglückt ist oder als gescheitert wird und somit als Verlust abgeschrieben wird. Er ist derjenige, welcher gewisse Dokumente, wie Projektauftrag, Lasten und Pflichtenheft und das Abnahmeprotokoll absegnet und somit die Grundlage für das Projekt schafft.
- \* **Projektleiter:** Der Projektleiter/Die Projektleiterin ist die Person, die das Projekt leitet und die Verantwortung für die Planung, Durchführung und Kontrolle des Projektes trägt. Er ist derjenige, welcher das Projektteam führt und die Kommunikation zwischen dem Auftraggeber und dem Projektteam sicherstellt.
- \* **Der Teilprojektleiter/Abschnittsleiter:** Der Teilprojektleiter oder auch als Abschnittsleiter bezeichnet, ist ein Teammitglied, welches, gerade bei sehr großen Projekten mit vielen Abschnit-

ten, die Verantwortung für seinen Abschnitt des Projektes trägt. Er ist also eine Verlängerung des Projektleiters und hat die gleichen Aufgaben und Privilegien wie der Gesamtprojektleiter, jedoch eben nur für seinen Abschnitt des Projektes. Diese Rolle ist vor allem wichtig, wenn die einzelnen Abschnittsteams nicht direkt mit dem Projektleiter kommunizieren oder das Projekt eine Größe aufweist, bei der es für den Projektleiter nicht mehr zumutbar ist, alle Teammitglieder zu führen. Diese Rolle fällt bei kleineren Projekten generell weg.

\* **Der Qualitätsmanager:** Der QualitätsmanagerIn ist die Person in einem Projektteam, welche die Verantwortung für die Qualität des Endproduktes trägt. Er ist also dafür verantwortlich, dass das Produkt den Anforderungen und Qualitätsmerkmalen des Projektauftraggebers entspricht und die Abnahme des Produktes sichergestellt werden kann. Er kontrolliert laufend die Qualität des Produktes und kann bei Bedarf Maßnahmen ergreifen, um die Qualität des Produktes, laut Lastenheft sicherzustellen.

\* **Der Projektkaufmann:** Der Projektkaufmann/Die Projektkauffrau ist die Person im Projektteam, welche das Budget des Projektes verwaltet und vor allem überwacht. Er stellt also sicher, dass das Projekt im Rahmen des Budgets bleibt und somit ein Scheitern des Projektes aufgrund von wegfallenden finanziellen Ressourcen nicht eintritt. In dieser Rolle muss der PKM immer das Budget im Auge behalten und wenn nötig, Maßnahmen ergreifen, um entweder das Budget zu erhöhen, die Kosten zu senken oder das Projekt als gescheitert zu erklären, wenn das Budget nicht mehr gerettet werden kann.

\* **Der Projektcontroller:** Ähnlich wie der Projektkaufmann, ist der Projektcontroller/Die Projektcontrollerin dafür verantwortlich, dass das Projekt zustande kommt. Der Unterschied ist, dass der Projektcontroller/Die Projektcontrollerin nicht das Budget im Auge hat, sondern die Termine und Meilensteine des Projektes. Er ist also für den generellen Fortschritt des Projektes verantwortlich und stellt sicher, dass das Projekt zum ausgemachten Zeitpunkt fertiggestellt werden kann. Hierbei ist es wichtig zu erwähnen, dass der Projektcontroller/Die Projektcontrollerin ebenfalls das Projekt als gescheitert erklären kann, wenn die Termine und Meilensteine unter keinem Umstand mehr eingehalten werden können. Die finale Entscheidung obliegt in beiden Fällen (Budget und Termine) jedoch immer dem Projektleiter bzw. dem Projektauftraggeber.

\* **Die Projektbüromitarbeiter:** Die Projektbüromitarbeiter/Die Projektbüromitarbeiterin ist die Person im Projektteam, welche die administrativen Aufgaben des Projektes übernimmt. Hierzu zählen unter anderem die Dokumentation des Projektes, die Erstellung von Protokollen und Berichten, die Organisation von Meetings und die Kommunikation mit dem Auftraggeber und den Stakeholdern. Sie sind also für die gesamte Bürokratie verantwortlich und haben somit eine der **wichtigsten** Aufgaben, da die Dokumentation des Projektes vor allem aufgrund der rechtlichen Konsequenzen wichtig ist, wenn entweder die Dokumentation nicht vollständig ist oder etwas darin steht, was nicht der Wahrheit entspricht.

\* **Der Fachausschuss:** Der Fachausschuss ist die Zusammenkunft von Experten, welche aus allen beteiligten Abteilungen des Projektes besteht. Diese Experten sind für die fachliche Beratung des Projektteams verantwortlich und stellen sicher, dass das Projekt den fachlichen Anforderungen entspricht. Eine der wichtigsten Bereiche des Fachausschusses ist die Rechtsabteilung, Logistikabteilung. Diese beiden Bereiche sind wichtig, da sie in der Regel die meisten Probleme in einem Projekt verursachen können und wo sich das Projektteam am wenigsten auskennt. Der Fachausschuss ist also eine Art Expertenrat, welcher dem Projektteam beratend zur Seite steht und bei Bedarf auch Entscheidungen treffen kann, wenn es um fachliche Fragen geht.



## Projektrollen Light (Abgespeckt)

Bei kleineren Projekten, welche nicht die Größe und Komplexität eines großen Projektes aufweisen, werden oft nicht alle Rollen benötigt oder können zusammengefasst werden. Daraus resultiert die folgende Liste:

- **Projektauftraggeber:** Diese Rolle bleibt immer gleich, da der Auftraggeber die Person ist, ohne die kein Projekt zustande kommt.
- **Projektleiter:** Diese Rolle ist ebenfalls immer gleich, da der Projektleiter/die Projektleiterin die Person ist, welche das Projekt leitet und die Verantwortung für die Planung, Durchführung und Kontrolle des Projektes trägt.
- **Projektteammitglied:** Diese Rolle ist einfach gesagt die Person, welche die eigentliche Arbeit am Projekt verrichtet. Sie ist also für die Umsetzung der Anforderungen und Ziele des Projektes verantwortlich.
- **Projektmitarbeiter:** Diese Rolle ist im Grunde der Fachausschuss.

## Vor- und Nachteile (vgl. 7)

Grundsätzlich hat alles seine Vor- und Nachteile. Das Wasserfallmodell ist hierbei keine Ausnahme.

### Vorteile

**Klarheit und Struktur:** Das Wasserfallmodell bietet aufgrund seiner klaren hierarchischen Struktur eine hohe Klarheit und Struktur im Projektablauf. Alle Rollen und Verantwortlichkeiten sind klar definiert und es gibt sehr wenig Interpretationsspielraum. Jeder Beteiligte weiß genau, was er zu tun hat und welche Aufgaben er zu erfüllen hat. Dies führt zu einer hohen Effizienz im Projektablauf.

**Sicherheit:** Durch die klaren Anforderungen und Ziele des Projektes, welche in der Anforderungsanalyse, dem Lastenheft und final im Pflichtenheft festgehalten werden, ist das Wasserfallmodell sehr sicher. Wenn alle Ziele und Anforderungen dem SMART-Abfrageschema entsprechen, ist die Wahrscheinlichkeit für Interpretationsspielraum sehr gering und somit auch die Wahrscheinlichkeit für Missverständnisse und Fehler im Projektablauf.

**Zeitplansicherheit:** Dadurch dass das Wasserfallmodell ein sequentielles Vorgehensmodell ist, ist der Zeitplan des Projektes sehr sicher.

### Nachteile

**Unflexibilität:** Das Wasserfallmodell ist aufgrund der starren Struktur beim Wegfall von Teammitgliedern sehr unflexibel. Die Reaktionsfähigkeit bei solchen Ereignissen ist daher beeinträchtigt.

**Geringe Fehlererkennungsfähigkeit:** Aufgrund der fixen Berichtszyklen werden Fehler oft erst sehr spät erkannt und können dann nur schwer behoben werden, ohne Verzögerungen im Projektablauf zu verursachen.

**Hohe Komplexität:** Das Wasserfallmodell ist nicht geeignet für sehr komplexe Projekte, da man hier oft keine oder nur eine sehr geringe Planungssicherheit aufweisen kann.

**Hoher Dokumentationsaufwand:** Das Wasserfallmodell erfordert eine Vielzahl an Dokumenten, welche für die Durchführung des Projektes notwendig sind. Diese Dokumente müssen erstellt, gepflegt und aktualisiert werden. Zusätzlich müssen diese Dokumente so geschrieben werden, dass sie für alle Beteiligten verständlich sind. Dies bedeutet, wenn das Projekt über mehrere Jahre und mit wechselnden Teammitgliedern und Stakeholdern aus möglicherweise verschiedenen Ländern durchgeführt werden soll, wächst die Dokumentation und muss auch übersetzt werden, wodurch die Dokumentation sich immer verdoppelt und der Aufwand immer komplizierter wird.

**Keine Iterativität:** Das Wasserfallmodell ist ein sequentielles Verfahren, welches keine Iterationen vorsieht. Damit gemeint ist, dass es keine Möglichkeit für Feedback- oder Korrekturschleifen gibt. Ein tatsächliches Feedback ist erst nach der Beendigung des Projektes möglich, was zu einem erhöhten Risiko für Fehler und Unstimmigkeiten führt.

**Ausfallvulnerabilität:** Wenn ein Teammitglied ausfällt (Krankheit, Unfall oder Ausscheiden aus dem Projekt) kann das Projekt ins Stocken bringen und somit eine Zeitverzögerung hervorrufen. Dies hat den Grund da eine Kompensation sehr kompliziert werden kann (vor allem später im Projekt) da Arbeitspakete Reevaluiert werden müssen um sie gegebenenfalls an andere Teammitglieder umzuverteilen oder aus dem Projekt zu streichen.

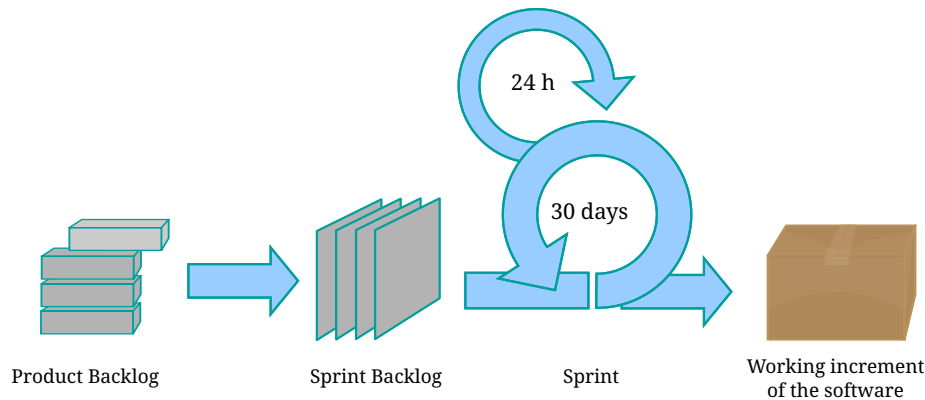
## Agiles Projektmanagement: Scrum und Kanban

Gerade in der IT-Branche und in der Softwareentwicklung oder bei Projekten, die eine Hohe Flexibilität und Anpassungsfähigkeit erfordern, hat sich ein neueres Modell des Projektmanagements etabliert. Dieses Modell ist das Agile Projektmanagement. Hierbei gibt es nicht nur eine Methode, sondern Mehrere, wie zum Beispiel Extreme Programming (XP), Feature-Driven Development (FDD), Dynamische Systementwicklung (DSDM).

### Beschreibung der Methode

Beim agilen Projektmanagement handelt es sich, anders als bei der klassischen Projektmanagementmethode, um eine moderne Variante, deren Ursprung bereits im Jahr 1957 liegt, damals jedoch nur teilweise, aber ab 1970 dann erst wirklich relevant wurde, vor allem im Bereich der Softwareentwicklung. Hierbei arbeiten die Teams in kurzen, aber konzentrierten Projekteinheiten, um flexibel auf Veränderungen reagieren zu können.

### Phasen des Agilen Projektmanagements



(img. 3)

Beim Scrum Framework wird das Projekt in sogenannte Sprints eingeteilt. Diese Sprints sind also ähnlich der Berichtszyklen im klassischen Projektmanagement. Dennoch gibt es vier essenzielle Bestandteile welche hier ihre Verwendung finden. Diese Bestandteile sind **das Product Backlog, das Sprint Backlog, der Sprint** und **das Working Increment of the Product**. Durch diese Bestandteile(Stufen) werden die einzelnen Arbeitspakete durchgeführt und das anhand ihres Fertigstellungsgrades oder Ihrer Priorität.

### Product Backlog

Das **Produkt Backlog** ist der **Aufgabenpool** des gesamten Projektes. Hier befinden sich alle Aufgabenpakete welche im **gesamten** Projekt anfallen. Es zeigt also den Gesamtfortschritt des Projektes auf. Je weniger Aufgaben im Product Backlog vorhanden sind desto weiter ist das Projekt vorangeschritten. Neue Aufgaben, welche während des Projektverlaufs aufkommen landen automatisch hier und werden hier im Sinne ihrer Priorität evaluiert und eingestuft. Mit dieser Einstufung wird dann überlegt wann im Verlauf der nächsten Stufe das Arbeitspaket bearbeitet wird.

### Sprint Backlog

Das **Sprint Backlog** ist ähnlich des **Product Backlog** ein Aufgabenpool der die Aufgabenpakete für den Sprint sammelt. Hier befinden sich also alle Aufgaben welche im **aktuellen** Sprint anfallen. Ist diese Hürde genommen wird ein Arbeitspaket durch einen Entwickler abgearbeitet und entsprechen fertiggestellt. Es stellt also die nächst höhere Stufe dar und hier wird weniger auf Priorität geachtet sondern auf die Reihenfolge. Diese Hürde ist aber nicht entgültig und wenn es notwendig ist kann auch während des Sprints eine Reevaluierung stattfinden und ein Arbeitspaket kann wieder im Sprint Backlog landen. Eine Rückstufung in das Product Backlog ist zwar selten aber nicht unmöglich.

### Sprint

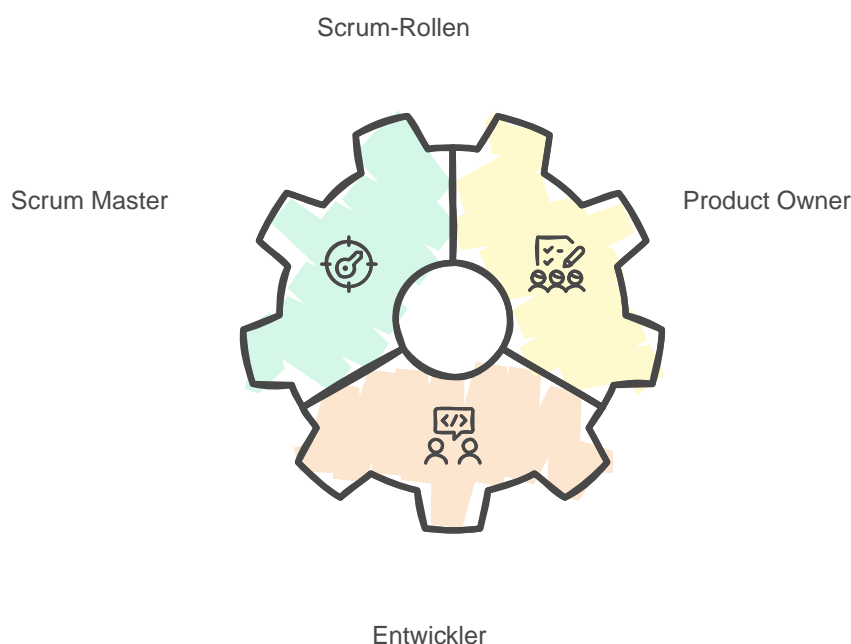
Der **Sprint** ist die Eigentliche Arbeitszeit in der Die Arbeitspakete Abgearbeitet werden. Dieser Sprint hat eine fix definierte Länge von bis zu 30 Tage (längere Sprints sind nicht ideal). Nach

diesen 30 Tagen gibt es ein Meeting mit allen Beteiligten (**Product Owner, Scrum Master und dem Team**) und es wird geschaut was wurde erledigt, was ist noch offen und wo gab es möglicherweise Probleme gibt welche Angesprochen werden müssen. Dieses Meeting wird auch als **Sprint Review** bezeichnet. Nach Beendigung dieses Meetings beginnt der Nächste Sprint und das ganze Prozedere wird wiederholt. Innerhalb des Sprints gibt es einen weiteren Zeitraum der wichtig ist. Denn während eines Sprint gibt es alle 24 Stunden ein Meeting, genannt Daily Stand-Up, bei welchen das Team intern abspricht was jeder an diesem Tag erreichen möchten, was er am Gestrigen Tag geschafft hat, was er nicht geschafft hat und wo es möglicherweise Probleme gab. Auf Grundlage dieser Beiden Meetings wird der Projektfortschritt gemessen.

### Working increment of the Product (WIP)

Der **Working increment of the Product (WIP)** oder auch **Work in Progress** stellt den Fertigstellungsgrad des Produktes dar. Hier landen also alle fertigen Arbeitspakete aus den Sprints. Es wird also bei jedem Sprint Meeting der Vergleich zwischen Product Backlog, Sprint Backlog und WIP gemacht und das Ergebnis dieses Vergleiches ist dann der Status des Projektes. Am Ende des Projektes ist dann also das Fertige Produkt der Inhalt dieser Stufe. Grundsätzlich ist es so, dass Arbeitspakete die einmal im **WIP** sind werden nicht mehr angerührt und sind daher immer Vollständig. Wenn also z.B. der Produkt Owner sich eine weitere Funktion wünscht von einem Arbeitspaket welches bereits im **WIP** ist so wird dieses nicht raus genommen sondern es wird ein komplett neues Arbeitspaket gemacht welches mit diesem in Relation gestellt wird. Dieses Neue Arbeitspaket durchläuft dann einmal den kompletten Evaluierungsprozess und kann sogar wenn es vom Team so eingestuft wird zur Gänze abgelehnt werden. Das bedeutet die Funktion kann oder wird nicht implementiert.

### Rollen und Verantwortlichkeiten



(img. 2)

Da es keine einheitliche Definition für agiles Projektmanagement gibt, sondern verschiedene Frameworks wie zum Beispiel **Scrum**. In diesem Framework werden drei große Rollen vergeben:

### 1. Produkt Owner

Der Product Owner ist die Interessensvertretung des Kunden im Projekt. Er ist also der Auftraggeber und kann auch der Kunde selbst sein. Seine Aufgaben sind es das Backlog zu verwalten und zu überlegen welche Tasks möglicherweise zuerst erledigt werden müssen. Er ist auch dafür verantwortlich, dass das Produkt in der höchst möglichen Qualität erstellt wird, da er effektiv gesehen für das Produkt zahlt. Der Product Owner ist auch für die **Sprint**-Abnahme verantwortlich.

### 2. Scrum Master

Der Scrum Master ist als Unterstützung für das Team da. Er achtet darauf, dass die richtigen Verfahren angewandt werden. Er organisiert die Projektmeetings, beschafft Räumlichkeiten und ist für die Anschaffung von benötigter Software und Hardware. Er ist zusätzlich auch eine Unterstützung des Product Owners. Einfach gesagt der Scrum Master übernimmt die Aufgaben eines Projektleiters und des Projektbüromitarbeiter (Operative und Bürokratische Verwaltung des Projektes) und ist somit essentiell für das voranschreiten des Projektes. Der Scrum Master sorgt auch für die Einhaltung der Scrum-Prinzipien und kann entsprechend nachschärfen wenn es zu Missverständnissen dahingehend kommt.

### 3. Team

Das Team (oder auch die Entwickler) ist der Kern des Projektes. Sie sind die Boots-on-the-Ground Worker. Sie machen die eigentliche Arbeit für die Erarbeitung des Produktes also die Bauarbeiter der Software. Sie sind also die Personen die das Produkt nach den Vorgaben des Product Owners und in einer Weise, welche den Scrum-Prinzipien entspricht und vom Scrum Master vorgegeben wird. Sie teilen sich die Verantwortung für das Scheitern oder den Erfolg des Projektes. Das Team ist auch für die verschiedenen Aspekte des Projektes (Analyse, Entwurf, Entwicklung, Tests und Dokumentation) verantwortlich. Das Team besteht je nach Größe des Projektes aus mehreren Personen und verwaltet sich grundsätzlich intern.

## Vor- und Nachteile (vgl. 7)

Auch beim Agilen Projektmanagement gibt es (auch wenn es die modernere Variante ist) Vor- und Nachteile.

### Vorteile

**Zeitliche Anpassbarkeit:** Grundsätzlich ist beim Agilen Projektmanagement kein exaktes Enddatum festgelegt. Es ist also variabel wann das Projekt tatsächlich zu einem Ende kommt. Dadurch fällt der Druck den ein fixes Enddatum bietet weil man nicht die ganze Zeit im Hinterkopf behalten muss.

**Kundenmitsprache:** Der Kunde hat aufgrund der Sprint Reviews die Möglichkeit auch während dem Projektverlaufes Fragen, Wünsche, Anregungen oder Beschwerden vorzubringen und muss nicht hoffen, dass das Endprodukt dem Entspricht was er möchte. Es ermöglicht auch die Vorzeitige Beendigung des Projektes (wenn nicht mehr gewollt, Finanzierungsprobleme oder der Kunde nicht mehr existiert (z.B. in Konkurs geht)).

**Flache Hierarchie und Motivation:** Aufgrund der flacheren Hierarchie (Das Team teilt sich den Kern der Verantwortung untereinander) ist die Motivation im Team weil man sich sein eigenes Tempo wählen kann und somit auch hier der Druck ("Ich muss das jetzt bis xxx machen") fällt.

**Höhere Kundenzufriedenheit:** Da der Kunde mehr direkter in das Projekt integriert ist und eine deutlich ausgeprägtere Kommunikation zwischen Team und Kunde (und Stakeholder) stattfindet ist dieser am Ende deutlich Zufriedener und hat auch selbst weniger Stress (Angst ob Produkt wirklich das ist was er möchte). Er ist also um einiges weniger Angespant als anders vielleicht.

**Iterativität:** Dadurch dass, gerade bei Scrum, das Projekt laufend Reevaluiert werden kann ist eine bessere Reaktionszeit möglich. Hierdurch kann man auch besser auf Unvorhergesehenes reagieren und somit ein Scheitern des Projektes verhindern.

**Ausgeprägtere Kommunikation:** Da das ganze Konzept auf Kommunikation beruht gibt es weniger Spielraum für Interpretationen, da man bei Fragen leichter entweder den Product Owner oder Kunden kontaktieren kann und somit die benötigten Informationen leichter erhalten werden.

## Nachteile

**verminderte Dokumentation:** Aufgrund des Verringerten Dokumentationsaufwandes beim Agilen Projektmanagement ist eine Rechtfertigung schwerer möglich. Das bedeutet wenn am Ende des Projektes doch Fragen oder Missverständnisse aufkommen sind die Handlungen des Teams während des Projektes deutlich schwerer möglich als bei der Klassischen Methodik.

**Kein fixes Endergebnis:** Da das Produkt im Laufe des Projektes sich vielleicht mehrmals ändert (entweder durch zusätzliche oder weniger Funktionen) ist nicht lässt sich kein eindeutiges und Fixes Endergebnis festlegen. Einzig ein grundsätzlicher Funktionsvorschlag kann vorgebracht werden.

**erschwerte Langfristigkeit:** Langfristige Projekte erfordern ein erhöhtes Maß an durchaus genauer und Penibler Planung bis ins kleinste Detail. Dies ist aufgrund der Iterativität und Inkremtierten Bereitstellung des agilen Projektmanagements schwer oder garnicht möglich, da agiles PM nur kurzfristig eine genaue Planung ermöglicht was langfristig zu komplikationen führt.

## Vergleich der Methoden

Um beide Methoden effektiv zu vergleichen muss man sich zuerst anschauen was die größten



Unterschiede beide Methodiken aufweisen und in welchen Bereichen welche Methodik am besten angewendet wird. Erst dann kann man einen exakten Vergleich anstellen. Weiters ist zu sagen das keine der Methodiken besser ist als die andere sondern es Situationsbedingt ist ob man lieber "klassisch" oder agil vorgeht. Dann gibt es noch zu sagen, dass kein Projekt 100%ig "klassisch" oder 100%ig agil geführt wird es handelt sich in der Regel in den meisten Fällen um eine Mischform.

## Unterschiede

### 1. Iterativität

Einer der größten Unterschiede zwischen "klassischem" Projektmanagement und agilem Projektmanagement ist und bleibt die Iterativität. Diese bedeutet das innerhalb des Projektes eine "Phase" immerwieder durchlaufen wird und nach jedem Durchlauf eine Reevaluierung der Projektziele stattfindet. Der Begriff findet hierbei hauptsächlich in der Softwareentwicklung gebrauch und beschreibt dort das wiederholte durchlaufen einer Methode (z.B. in einer Schleife) um auch hier nach jedem Durchlauf eine Reevaluierung eines Maßgeblichen Parameters (z.B. die Frage ob der Inhalt eines Warenkorb den Maximal zulässigen Warenwert erreichen kann) zuzulassen. Dies ist jedoch muss bei Langfristigen Projekten nicht immer ein Nachteil sein, da man so immer Evaluieren kann ob ein Aspekt noch benötigt wird oder ob möglicherweise Maßnahmen getroffen werden um das Projekt wieder in eine Bestimmte Richtung zu Lenken.

### 2. Dokumentationsaufwand

Bei einem "klassisch" geführten Projekt steht die ausführliche und stichhaltige Dokumentation immer im Vordergrund. Dies hat den Grund das diese Dokumentation oft der **einzigste** Rechtfertigungsgrund darstellt. Das wird vor allem dann wichtig wenn das Projekt scheitert, da zwischen dem Kunden und dem Projektteam grundsätzlich ein Kooperationsvertrag geschlossen wird und dieser in den meisten Fällen eine je nach dem wie kritisch die Fertigstellung des Produktes ist eine Vertragsstrafe durchaus normal ist. Diese macht oft ein Vielfaches des Projektwertes aus. Um also zu verhindern, dass man möglicherweise diese Strafe zahlen muss (was gerade für ein kleines Team den Ruin bedeuten kann) ist eine lückenlose und schlüssige Dokumentation durchaus wichtig.

Im agilen Projektmanagement wird dies dezent durch die direkte Entscheidungsfindung und Beteiligung des Kunden am Projekt abgedeckt, da hier der Kunde selbst eine gewisse Verantwortung am Erfolg oder Misserfolg am Projekt trägt und auch immer wieder verlauten kann wenn er etwas am Projekthinhalt ändern möchte. Das heißt die Rechtfertigungspflicht liegt hier nicht ausschließlich beim Projektteam sondern teilt sich zwischen den einzelnen Parteien (Kunde, Product Owner, Stakeholder, Scrum Master und Entwickler) auf.

Auch die Anzahl an Dokumenten die benötigt werden unterscheidet sich grundlegend. So braucht man bei einem "klassisch" geführten Projekt zum einen ein Projekthandbuch (enthält alle relevanten Aspekte des Projektes und wird über das gesamte Projekt adaptiert und aktualisiert), Statusberichte mit einer Meilensteintrendanalyse, welche eindeutig aufzeigt ob alle Mei-

lensteine immernoch planmäßig erreicht werden können. Diese Statusberichte müssen in einem vordefinierten Zeitraum vom PL oder PBA erstellt werden und an den Kunden gesendet werden. Dies ist bei Agilen Projekten nicht notwendig da der Kunde aufgrund seiner direkten Involvierung generell immer auf dem neusten Stand ist.

### 3. Kontakt-Prinzipien

Beim "klassischen" Projektmanagement ist es generell so das der Kunde gerade am Anfang, am Ende und wenn der Statusbericht kommt wirklichen Kontakt zum Projektteam hat. Jetzt könnte man sagen: "dann macht doch alle 14 Tage oder drei Wochen einen Statusbericht". Das ist nur leider nicht klug da der Aufwand des Statusberichtes mit der Größe des Projektes wächst. Aus diesem Grund sagt man auch, dass die "klassische" Methodik einem Wal ähnelt weil dieser oft 2-3 Monaten untertaucht und nicht zu sehen ist.

Anders ist es hier bei der Agilen Methodik. Hier ist der Kunde direkt in die Entscheidungsfindung eingebunden und die Sprints sind oft je nach grundlegender Gesamtlänge des Projektes so gewählt das ein häufiger Kontakt möglich ist. Vorallem ist anzumerken, dass die Sprint-Gliederung (wobei ein Sprint auch eine Woche sein kann) diese Möglichkeit noch verstärkt und es Tatsache ist dass ein direkter Kontakt mit dem Projektteam um einiges Vorteilhafter ist als der Erhalt eines Dokumentes. Der Soziale Kontakt ist einfach besser für die Projektkultur.

### 4. Fehler- & Änderungstoleranz

Grundsätzlich ist die "klassische" Methodik zwar Rechtfertigungsfreudiger bei Fehlern, aufgrund der im besten Fall lückenlosen Dokumentation, aber hat ihre Grenzen was die Fehler- bzw. Änderungstoleranz angeht. Das wird vor allem klar wenn man sich das Grundsätzliche Prozedere einer Änderung im Projektinhalt ansieht. In der "klassischen" Methodik muss ein Änderungsantrag vom Projektteam an den Kunden geschickt werden, dabei muss geschaut werden ob diese Änderung das Projekt nicht bezogen auf die Ressourcenverwaltung gefährdet. Wenn dass alles passt und der Kunde den Änderungsantrag unterzeichnet, dann kann die Änderung vorgenommen werden.

Bei der Agilen Methodik ist dies um einiges vereinfacht da der Kunde direkt Feedback zu der Änderung geben kann und sogar selber Änderungen vorschlagen und die Gefährdung des Projektes ist um einiges geringer als bei der "klassischen" Methodik. Außerdem ist es so das durch die erhöhte Kommunikation mit dem Kunden um einiges schneller auf Fehler oder Ereignisse reagiert werden kann als bei der "klassischen" Methodik. Dadurch wird das Risiko des Scheitern deutlich verringert.

## Anwendungsbereiche

Grundsätzlich sind die Anwendungsbereiche der "klassischen" Methodik und der Agilen Methodik deutlich Unterschiedlicher. Das bedeutet aber nicht, dass es nicht auch gemeinsame Anwendungsbereiche gibt. Weiters darf man nicht vergessen, dass man nie entweder "klassisch" oder agil vorgeht sondern es immer gewisse Mischformen gibt.

Aber grundlegend kann man schon annehmen, dass Projekte die Rund um Kritische Infra-



struktur passieren eher zur "klassischen" Methodik tendieren sollten. Auch Projekte an denen eine Hohe Geldsumme hängen sollten die Vorteile des "klassischen" Methodik nutzen.

Grundsätzlich ist agiles Projektmanagement für die Softwareentwicklung konzipiert, da die Entwicklung eines Softwareprogramms nie genau nach plan verläuft.

## Fazit, Ausblick und Anwendung im Projekt

### Fazit

Abschließend ist zu sagen, dass auch wenn beide Methodiken ihre Vorteile und Ihre Nachteile haben und auch beide haben Ihre Daseinsberechtigung haben. Dennoch darf man nicht vergessen das es immer ein Aufwiegen ist welche Aspekte am Besten zum Projekt passen um es am Effektivsten zu Nutzen. Somit lässt sich zwar sage, dass beide Methodiken doch große Unterschiede haben können sie auch gut zusammenarbeiten und am Ende des Tages ist doch wichtig, dass das Produkt im Mittelpunkt steht und man muss grundsätzlich selber entscheiden was für die Herstellung des Produktes am besten passt. Ob es jetzt die Rechtfertigungsfähigkeit der "klassischen" Methodik oder die Flexibilität und Änderungstoleranz der Agilen Methodik.

### Ausblick (wie wird es Weitergehen )

Grundsätzlich kann man sagen, dass die Welt sich rapide Weiterentwickelt und es immer mehr Anpassungsfähigkeit braucht um im Kampf zu bleiben. Ob es so sein wird, dass die "klassische" Methodik ausstirbt oder immer wichtiger wird ist schwer zu sagen. Eine Sache die man aber mit Sicherheit sagen kann ist, dass die Agile Methode immer wichtiger wird und auch außerhalb des Projektmanagement immer mehr Unternehmen eine agile Unternehmenskultur anstreben um weiterhin relevant am Markt zu bleiben.

### Anwendung im Projekt "FindME - Das Digitale Fundbüro"

Das Projekt "FindME - Das Digitale Fundbüro" ist entstanden aufgrund der Tatsache, dass die Fundsachensituation innerhalb der HTBLVA Spengergasse eher ungünstig war. Den Gegenstände, welche innerhalb der Schule verloren wurden, wurden grundsätzlich in einer Kiste oder im Sekretariat gelagert und wurden nach dem Ende des Jahres entsorgt.

Unser Projekt beinhaltet die Erstellung einer Web-Applikation mit welcher die Gegenstände zentral und Öffentlich einsehbar erfasst werden können. Dadurch kann jeder direkt Nachschauen wenn er etwas verloren hat.

Die Web-Applikation beinhaltet eine Auflistung aller Gegenstände mit Informationen wie Bezeichnung, eine Beschreibung, der Ort wo der Gegenstand gefunden wurde, wo der Besucher nachschauen kann ob sein Gegenstand gefunden wurde oder nicht.

In Bezug auf das Projektmanagement mussten wir also beachten, dass wir die Balance zwischen beider Methodiken finden, da wir grundlegend beide Aspekte abdecken müssen um den Kontext der Schulausbildung nicht zu vernachlässigen. Weiters war es ein großes Problem, dass wir uns nicht 100%ig auf das Projekt konzentrieren konnten da der restliche Unterricht immer wieder

dazwischen kam wodurch wir immer wieder Phasen hatten in denen wir schlichtweg nicht die Ressourcen aufbringen konnten um am Projekt zu arbeiten. Dadurch sind wir bereits mit einer gewissen Verzögerung in das Projekt gestartet. Dennoch war es uns möglich die weniger intensive Zeit

## Quellen

### Literatur

### Internetquellen

- Klassisches Projektmanagement im Überblick, URL: <https://projekte-leicht-gemacht.de/projektmanagement/klassisches-projektmanagement/> (abgerufen am 18.06.2025)
- Das Wasserfallmodell einfach erklärt, projekte-leicht-gemacht.de, URL: <https://projekte-leicht-gemacht.de/blog/projektmanagement/klassisch/wasserfallmodell/#~:text=Das%20Wasserfallmodell%20ist%20ein%20lineares%20Vorgehensmodell%20im%20Projektmanagement.,grafisch%20oft%20in%20Form%20eines%20Wasserfalls%20dargestellt%20werden.> (abgerufen am 18.06.2025)
- SMART-Regel/SMART-Methode, URL: [https://www.orghandbuch.de/Webs/OHB/DE/OrganisationshandbuchNEU/4\\_MethodenUndTechniken/Methoden\\_A\\_bis\\_Z/SMART\\_Regel\\_Methode/SMART\\_Regel\\_Methode\\_node.html](https://www.orghandbuch.de/Webs/OHB/DE/OrganisationshandbuchNEU/4_MethodenUndTechniken/Methoden_A_bis_Z/SMART_Regel_Methode/SMART_Regel_Methode_node.html) (abgerufen am 18.06.2025)
- Phasen des Wasserfallmodells, URL: <https://projekte-leicht-gemacht.de/blog/projektmanagement/klassisch/wasserfallmodell/#~:text=Das%20Wasserfallmodell%20ist%20ein%20lineares%20Vorgehensmodell%20im%20Projektmanagement.,grafisch%20oft%20in%20Form%20eines%20Wasserfalls%20dargestellt%20werden> (abgerufen am 18.06.2025)
- Die beliebtesten agilen Methoden im Überblick, URL: <https://projekte-leicht-gemacht.de/blog/projektmanagement/agil/agile-methoden-ueberblick/> (abgerufen am 18.06.2025)
- Agiles Projektmanagement: Definition, Eckpfeiler, Fakten, URL: <https://www.marketinginstitut.biz/blog/agiles-projektmanagement/#~:text=Agiles%20Projektmanagement%20fand%20seinen%20Ursprung,Branche%20der%20Software%2DEntwicklung%20eingesetzt.> (abgerufen am 18.06.2025)
- Agiles vs. klassisches Projektmanagement: Individuelle Vor- und Nachteile, URL: <https://topsoft.ch/themen/agiles-vs-klassisches-projektmanagement-individuelle-vor-und-nachteile/> (abgerufen am 21.06.2025)

### Abbildungen

- Wasserfallmodell, URL: <https://commons.wikimedia.org/wiki/File:Wasserfallmodell.svg> (abgerufen am 18.06.2025)
- Scrum Rollen, URL: <https://www.pureconsultant.de/de/scrum/scrum-rollen/> (abgerufen am 21.06.2025)



- Scrum Process, URL: [https://upload.wikimedia.org/wikipedia/commons/5/58/Scrum\\_process.svg](https://upload.wikimedia.org/wikipedia/commons/5/58/Scrum_process.svg) (abgerufen am 21.06.2025)

# Barrierefreiheit im Webdesign

Verfasser: Daniel Wagner

Klasse: 6CAIF

Betreuer: Rene Wenz

Jahrgang: 2024/25

HTBLVA Spengergasse

Ausbildungsschwerpunkt Informatik für Erwachsene

Abgabedatum: 27.06.2025

# 1. Inhaltsverzeichnis

2. Einleitung .....	3
3. Theorie .....	4
3.1. Was versteht man unter Barrierefreiheit im Webdesign? .....	4
3.2. Warum ist Barrierefreiheit im Webdesign wichtig? .....	4
3.3. Web Content Accessibility Guidelines .....	5
3.3.1. Konformitätsstufen .....	6
3.4. Hindernisse und Barrieren .....	6
3.4.1. Visuelle Barrieren.....	7
3.4.2. Motorische Barrieren.....	8
3.4.3. Kognitive Barrieren .....	9
4. Praktische Umsetzung.....	10
4.1. Schrift .....	10
4.2. Farbe .....	11
4.3. Responsive Darstellung und Zoom .....	13
4.4. Testung der Barrierefreiheit im FindMe Projekt .....	13
4.4.1. Lighthouse Test.....	13
4.4.2. axe DevTools Test .....	15
5. Lessons Learned .....	16
6. Fachbegriffe .....	17
7. Literaturverzeichnis.....	18
8. Abbildungsverzeichnis.....	19

## 2. Einleitung

Diese wissenschaftliche Arbeit widmet sich mit der Barrierefreiheit im Webdesign und den Techniken sowie Standards, die erforderlich sind, um eine barrierefreie Webanwendung zu erstellen.

Im theoretischen Teil werden die Grundlagen der Barrierefreiheit erklärt. Es wird gezeigt, warum Barrierefreiheit wichtig ist, welche gesetzlichen Vorgaben es gibt und welche Standards berücksichtigt werden müssen.

Im praktischen Teil wird demonstriert, wie diese Prinzipien im Projekt „FindMe“ umgesetzt wurden. Dabei werden unter anderem Schriftwahl, das Farbschema, responsives Design und Accessibility-Tests mit Lighthouse und axe DevTools behandelt. Das Ziel ist es, ein Bewusstsein für barrierefreies Webdesign zu fördern und konkrete Möglichkeiten zur Umsetzung aufzuzeigen.

## 3. Theorie

### 3.1. Was versteht man unter Barrierefreiheit im Webdesign?

Barrierefreiheit im Webdesign bedeutet, Webseiten so zu gestalten, dass sie für alle Menschen unabhängig von körperlichen, sensorischen, kognitiven oder technischen Einschränkungen zugänglich und nutzbar sind.<sup>1</sup>

Dazu gehört, dass Webinhalte sowohl von Menschen mit Sehbehinderungen über Screenreader gelesen werden können als auch für Menschen mit motorischen Einschränkungen mit der Tastatur bedienbar sind.

Barrierefreiheit ist nicht nur für Menschen mit dauerhaften Behinderungen relevant, sondern verbessert auch die Nutzung für ältere Personen, Menschen mit temporären Einschränkungen oder jene in besonderen Nutzungssituationen.<sup>2</sup>

### 3.2. Warum ist Barrierefreiheit im Webdesign wichtig?

Barrierefreiheit im Webdesign ist ein entscheidender Aspekt, um allen Menschen den gleichberechtigten Zugang zu Informationen und digitalen Diensten zu ermöglichen. Allein in Deutschland und Österreich leiden rund 8 % der Männer und 0,4 % der Frauen an einer Rot-Grün-Sehchwäche – eine der häufigsten Formen von Farbsehstörungen.<sup>3</sup>

Für diese Nutzergruppe können schlecht konzipierte Farbschemata bedeuten, dass zentrale Inhalte oder Elemente nicht wahrgenommen werden können.

Neben dem sozialen Aspekt spielt Barrierefreiheit auch wirtschaftlich eine Rolle. Webseiten mit besserer Zugänglichkeit erreichen ein breiteres Publikum. Mehr Menschen können die Webseite nutzen die sie zuvor nicht erreichen konnten. Ebenfalls hilft eine Barrierefreie Webseite bei der Suchmaschinenoptimierung (SEO) und minimieren rechtliche Risiken, da viele Länder gesetzliche Anforderungen an barrierefreie Webangebote stellen.

---

<sup>1</sup> URL: <https://www.ionos.at/digitalguide/websites/web-entwicklung/barrierefreies-webdesign/>

<sup>2</sup> URL: <https://www.ionos.at/digitalguide/websites/web-entwicklung/barrierefreies-webdesign/>

<sup>3</sup> URL: <https://www.derstandard.at/story/2000093943222/wie-farbenblinde-menschen-die-welt-sehen>

### 3.3. Web Content Accessibility Guidelines

Die Web Content Accessibility Guidelines (WCAG) sind ein international anerkannter Standard zur barrierefreien Gestaltung von Webinhalten. Sie werden vom World Wide Web Consortium (W3C) entwickelt und regelmäßig aktualisiert. Ziel der WCAG ist es, Websites so zu gestalten, dass sie für Menschen mit unterschiedlichsten Einschränkungen zugänglich und bedienbar sind.<sup>4</sup>

Die Richtlinien basieren auf vier Prinzipien, die auch als POUR-Prinzipien bekannt sind:<sup>5</sup>

- **Perceivable** (Wahrnehmbar): Inhalte müssen für alle Sinne zugänglich sein (z. B. Alternativtexte für Bilder).
- **Operable** (Bedienbar): Webseiten müssen mit verschiedenen Eingabemethoden bedient werden können (z. B. mit Tastatur oder Screenreader).
- **Understandable** (Verständlich): Informationen und Navigation müssen klar und vorhersehbar sein.
- **Robust** (Robust): Inhalte müssen auch mit zukünftigen Technologien kompatibel bleiben.

---

<sup>4</sup> URL: <https://www.ionos.at/digitalguide/websites/web-entwicklung/wcag-richtlinien-fuer-die-barrierefreiheit-im-web/>

<sup>5</sup> URL: <https://ialabs.ie/understanding-the-pour-principles-of-accessibility/>



### 3.3.1. Konformitätsstufen

Die WCAG definiert drei Konformitätsstufen, die angeben, wie umfassend eine Website barrierefrei ist:

Konformitätsstufe	Definition	Zugänglichkeitsniveau
A	Eine Website entspricht der Konformitätsstufe A, wenn alle Erfolgskriterien der Stufe A erfüllt sind oder eine alternative Version der Website zur Verfügung steht, die die entsprechenden Kriterien erfüllt.	Niedriges Zugänglichkeitsniveau
AA	Eine Website entspricht der Konformitätsstufe AA, wenn alle Erfolgskriterien der Stufen A und AA erfüllt sind oder eine alternative Version der Website zur Verfügung steht, die die entsprechenden Kriterien erfüllt.	Mittleres Zugänglichkeitsniveau
AAA	Eine Website entspricht der Konformitätsstufe AAA, wenn alle Erfolgskriterien der Stufen A, AA und AAA erfüllt sind oder eine alternative Version der Website zur Verfügung steht, die die entsprechenden Kriterien erfüllt.	Hohes Zugänglichkeitsniveau

Abbildung 1: Konformitätsstufen laut WCAG <sup>6</sup>

Dabei ist zu beachten, dass die WCAG-Konformität sich immer auf jede einzelne Unterseite einer Website bezieht. Wenn auch nur eine Seite die Anforderungen nicht erfüllt, gilt die gesamte Website als nicht konform.

## 3.4. Hindernisse und Barrieren

Um eine Webseite barrierefrei zu gestalten, muss zunächst ein Verständnis dafür geschaffen werden, welche konkreten Barrieren für verschiedene Nutzergruppen existieren. Menschen mit Behinderungen oder Einschränkungen sind beim Zugriff auf digitale Inhalte oft auf besondere Gestaltung und technische Anpassungen angewiesen.

Die Hindernisse, die am meisten aufkommen, lassen sich in drei Kategorien einteilen: visuelle, motorische und kognitive Barrieren.

In den folgenden Abschnitten wird näher auf die drei Kategorien eingegangen.

<sup>6</sup> URL: <https://www.ionos.at/digitalguide/websites/web-entwicklung/wcag-richtlinien-fuer-die-barrierefreiheit-im-web/>

### 3.4.1. Visuelle Barrieren

Visuelle Barrieren betreffen Menschen mit eingeschränktem Sehvermögen, Farbsehschwäche oder völliger Blindheit. Diese Nutzer haben Schwierigkeiten mit Webinhalten, wenn beispielsweise der Kontrast zwischen Text und Hintergrund zu gering ist oder wenn Texte und Layouts sich nicht vergrößern lassen. Ein klassisches Problem stellt etwa ein roter Text auf schwarzem Hintergrund dar, da für Menschen mit Rot-Grün-Sehschwäche ein solcher Kontrast kaum erkennbar ist.



Abbildung 2: Rot-Grün Farbschwäche <sup>7</sup>

Damit Inhalte auch für sehbehinderte Menschen zugänglich bleiben, sollten Webseiten einen hohen Farbkontrast bieten. Dadurch wird sichergestellt, dass auch bei Sehschwächen Texte deutlich erkennbar bleiben. Ebenso wichtig ist ein responsives Design, das eine flexible Skalierung von Texten erlaubt.

---

<sup>7</sup> URL: <https://ajda.gregorcic.eu/2021/03/11/raising-awareness-about-colour-blindness/>

### 3.4.2. Motorische Barrieren

Menschen mit motorischen Einschränkungen sind häufig auf alternative Eingabemethoden angewiesen, etwa nur die Nutzung der Tastatur, Sprachsteuerung oder Geräte wie spezielle Schalter oder Joysticks. Für sie stellen enge Abstände zwischen zum Beispiel Buttons oder kleinen Klickflächen ein ernsthaftes Hindernis dar. Auch Funktionen, die Drag-and-Drop oder präzises klicken erfordern, können schwer oder gar nicht nutzbar sein.<sup>8</sup>

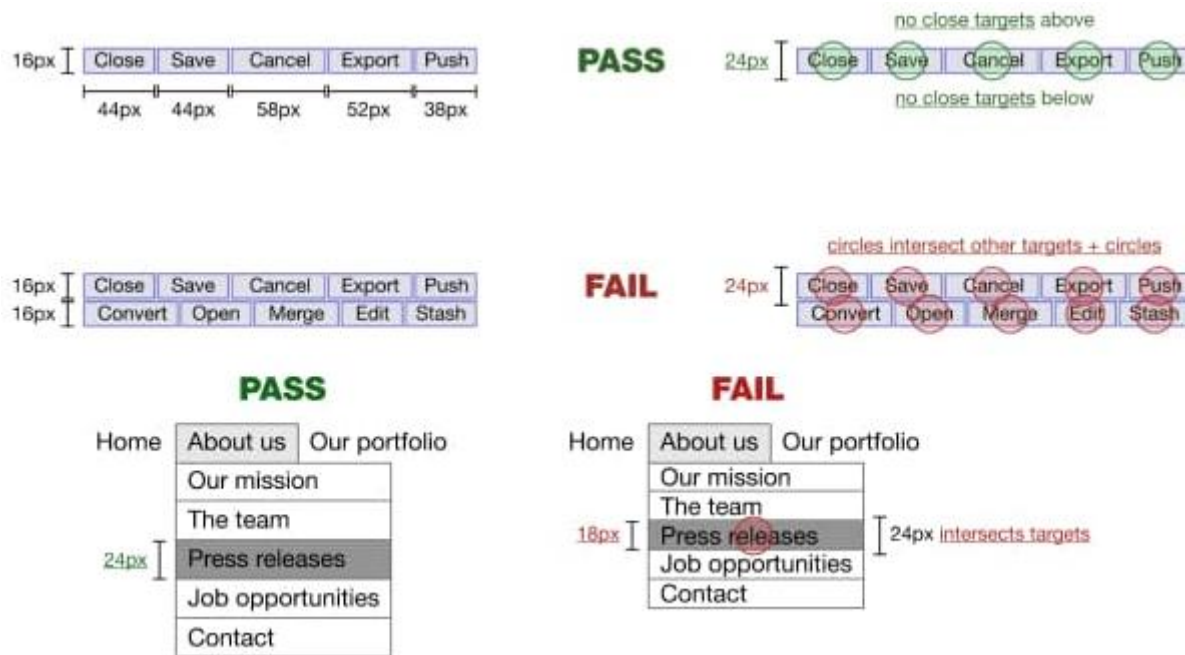


Abbildung 3: W3G.org Size requirement<sup>9</sup>

Um motorische Barrieren zu vermeiden, sollten Bedienelemente ausreichend groß gestaltet und mit einem angemessenen Abstand zueinander versehen werden. Alle Interaktionen auf der Website sollten auch vollständig mit der Tastatur durchführbar sein.

<sup>8</sup> URL: <https://www.barrierefreies-webdesign.de/spezial/multimediale-inhalte/behinderung-und-internet.html#:~:text=Nach%20Möglichkeit%20sollten%20allzu%20kleine%20Schaltflächen%20und,muss%20ausschließlich%20über%20die%20Tastatur%20möglich%20sein.>

<sup>9</sup> URL: <https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html#intent>

### 3.4.3. Kognitive Barrieren

Kognitive Barrieren betreffen Menschen mit Lernschwierigkeiten, Konzentrationsstörungen, Demenz oder anderen geistigen Beeinträchtigungen. Sie benötigen vor allem eine klare Struktur, einfache Sprache und eine konsistente Navigation. Webseiten mit komplizierten Begriffen, unübersichtlichem Aufbau oder wechselnden Layouts können bei dieser Zielgruppe schnell zur Überforderung führen.<sup>10</sup>

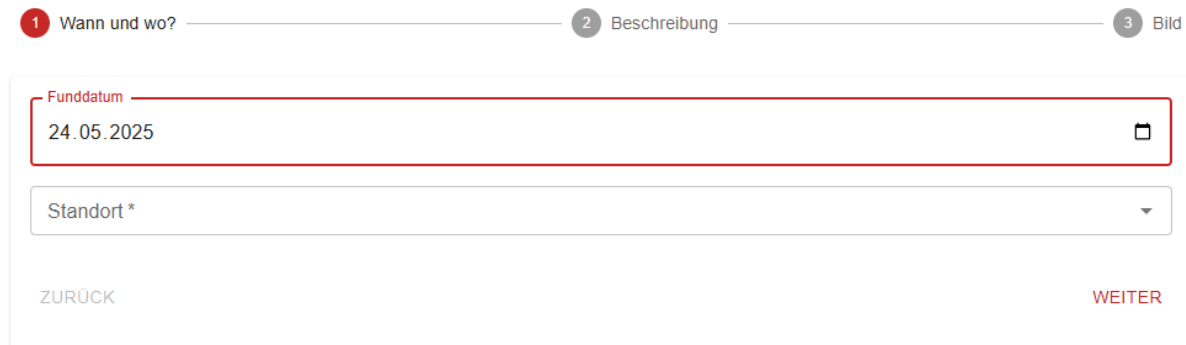


Abbildung 4: FindMe Funddatumseingabe

Ein konkretes Beispiel aus dem FindMe-Projekt wäre das Eingabeformular bei der Erstellung eines Fundstücks. Wenn dort beispielsweise das Feld nicht „Funddatum“, sondern „Datum“ heißen würde, dann könnten Nutzer mit kognitiven Einschränkungen Schwierigkeiten haben zu verstehen, ob damit das Datum des Fundes oder der Tag des Hochladens gemeint ist.

Durch gezielte sprachliche Vereinfachung, klare visuelle Trennung von Eingabebereichen und die Vermeidung technischer Fachbegriffe können auch kognitiv eingeschränkte Nutzer das System leichter bedienen.

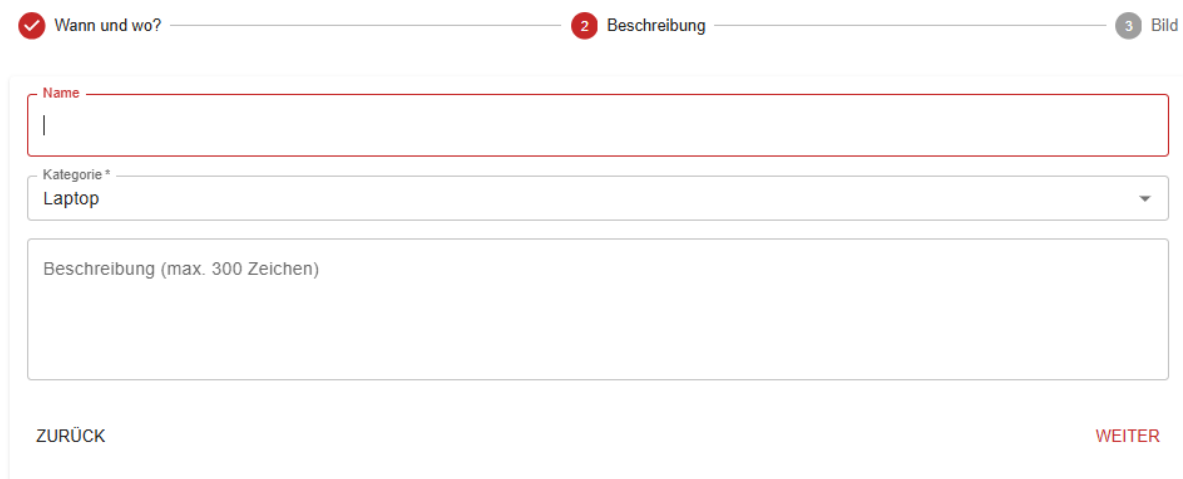
<sup>10</sup> URL: <https://barrierefreiheitcheck.de/blog/kognitive-barrieren-in-der-webgestaltung>

## 4. Praktische Umsetzung

Die Umsetzung barrierefreier Webdesign-Prinzipien im FindMe-Projekt erfolgte entlang der vier Grundprinzipien der WCAG. Ziel war es, die Anwendung so zu gestalten, dass auch Nutzerinnen und Nutzer mit Einschränkungen die Seite ohne Barrieren bedienen können.

### 4.1. Schrift

Ein wichtiger Aspekt für barrierefreies Webdesign ist eine gut lesbare Schrift. Im FindMe-Projekt wurden serifenlose Schriftarten sowie ausreichend große Schriftgrößen verwendet. Mit der TextField-Komponente von Material UI wurde sichergestellt, dass die Eingabefelder einheitlich, fokussierbar und gut sichtbar sind.



✓ Wann und wo? — 2 Beschreibung — 3 Bild

Name

Kategorie \*

Laptop

Beschreibung (max. 300 Zeichen)

ZURÜCK WEITER

Abbildung 5: FindMe Beschreibungseingabe

Der Fokus ist durch einen gut sichtbaren Rahmen hervorgehoben, was der WCAG-Anforderung 2.4.7 – Sichtbarer Fokus entspricht.<sup>11</sup>

<sup>11</sup> URL: <https://www.w3.org/WAI/WCAG22/Understanding/focus-visible>

## 4.2. Farbe

Farben wurden so gewählt, dass sie hohen Kontrast bieten und nicht alleinige Träger von Informationen sind.

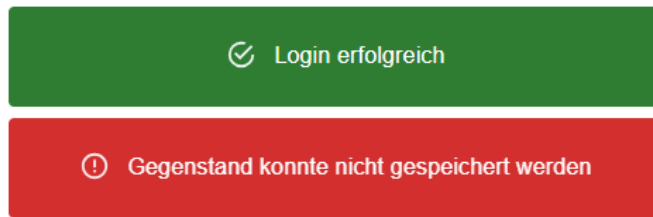


Abbildung 6: FindMe Alerts

Fehlermeldungen z. B. sind rot dargestellt, aber zusätzlich mit Text und Icon versehen, wodurch sie auch von Menschen mit Farbsehschwächen erkannt werden können.

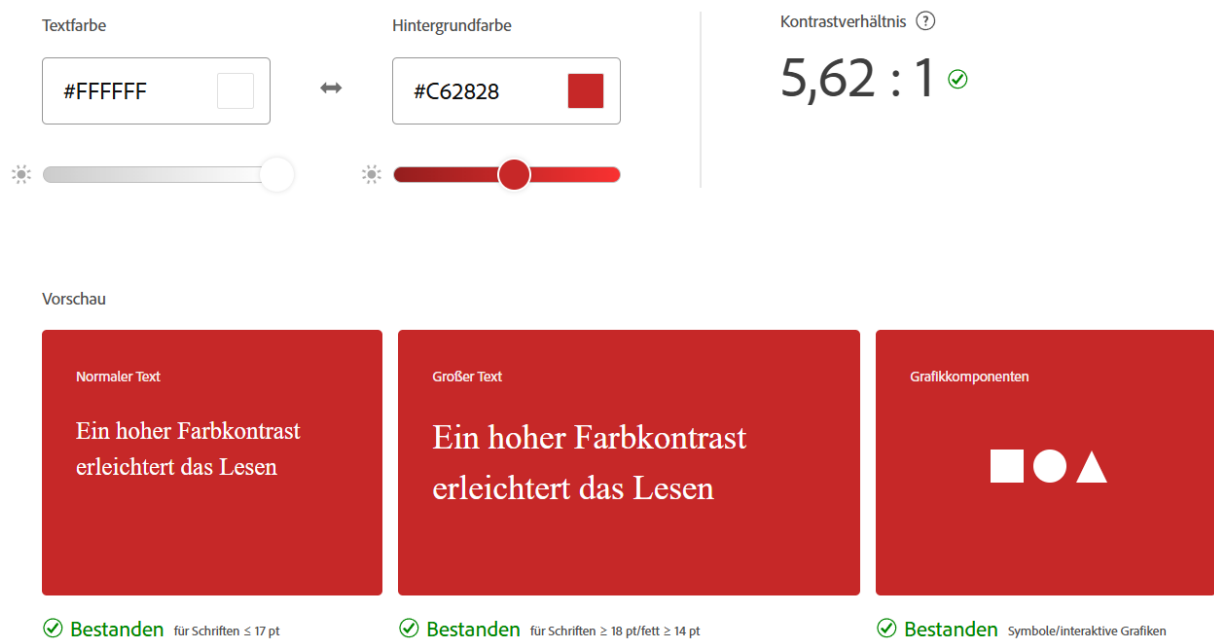


Abbildung 7: Adobe Color – Kontrastprüfung <sup>12</sup>

WCAG 1.4.3 – Kontrast (Minimum) verlangt, dass Textinhalte ein Kontrastverhältnis von mindestens 4.5:1 aufweisen.<sup>13</sup> Durch die Wahl kräftiger Farben auf hellem Hintergrund wird dieses Kriterium in unserem Projekt erfüllt. Ein Test durch den Adobe Color Kontrastprüfer hat gezeigt, dass unsere Webseite ein Kontrastverhältnis von 5,62:1 hat.

<sup>12</sup> URL: <https://color.adobe.com/de/create/color-contrast-analyzer>

<sup>13</sup> URL: <https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum>

```
const theme = createTheme({
  palette: {
    primary: {
      main: "#c62828",
    },
    background: {
      default: "#ffffff"
    },
    text: {
      primary: "#000000"
    }
  },
  typography: {
    fontFamily: "Roboto, Arial, sans-serif",
    fontSize: 14,
    button: {
      textTransform: "none"
    },
  },
  components: {
    MuiButton: {
      defaultProps: {
        variant: "contained",
        color: "primary"
      },
    },
  },
});
```

Mithilfe von Material UI ist es möglich, ein zentrales Theme zu definieren. Dieses Theme regelt globale Designparameter wie Primärfarbe, Schriftart und Komponentenverhalten. Der Vorteil ist, dass Farbgebung und Schriftart überall einheitlich sind, was die Barrierefreiheit unterstützt.

### 4.3. Responsive Darstellung und Zoom

Das Layout wurde mit Box-Containern und flexibler Breite umgesetzt. Die Seite passt sich damit verschiedenen Bildschirmgrößen an, was Zoom und mobile Nutzung unterstützt – ganz im Sinne von WCAG 1.4.10 – Reflow.<sup>14</sup>

```
<Box sx={{ display: "flex", flexDirection: { xs: "column", md: "row" }, gap: 2 }}>
```

Durch „flexDirection“ wird der Inhalt auf kleineren Geräten, wie Handys oder Tablets, untereinander dargestellt und auf großen Geräten, wie Computern, nebeneinander angezeigt. Durch Material UI wird dieses Verhalten automatisch im Code integriert, ohne dass zusätzliche CSS-Dateien benötigt werden.

So bleiben Texte und Bedienelemente auch bei starker Vergrößerung zugänglich, was besonders für sehbehinderte Nutzer entscheidend ist.

### 4.4. Testung der Barrierefreiheit im FindMe Projekt

Die Umsetzung barrierefreier Webdesign-Prinzipien sollte nicht nur konzeptionell berücksichtigt, sondern auch mithilfe Tools überprüft werden. Im FindMe-Projekt wurden zwei gängige Werkzeuge zur Testung der Barrierefreiheit eingesetzt: Google Lighthouse und axe DevTools. Beide ermöglichen eine automatisierte Analyse von Webseiten hinsichtlich Barrierefreiheit, Struktur, Benutzerfreundlichkeit und anderen Qualitätsmerkmalen.

#### 4.4.1. Lighthouse Test

Lighthouse ist ein Open-Source-Tool von Google, das direkt in die Chrome Developer Tools integriert ist.<sup>15</sup> Es ermöglicht die Prüfung von Webseiten in verschiedenen Kategorien, darunter Leistung, Barrierefreiheit, Best Practices und SEO. Für das FindMe-Projekt wurde ein Lighthouse-Audit der Startseite durchgeführt.<sup>16</sup>

<sup>14</sup> URL: <https://www.w3.org/WAI/WCAG22/Understanding/reflow>

<sup>15</sup> URL: <https://github.com/GoogleChrome/lighthouse>

<sup>16</sup> URL: <https://chromewebstore.google.com/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=de>



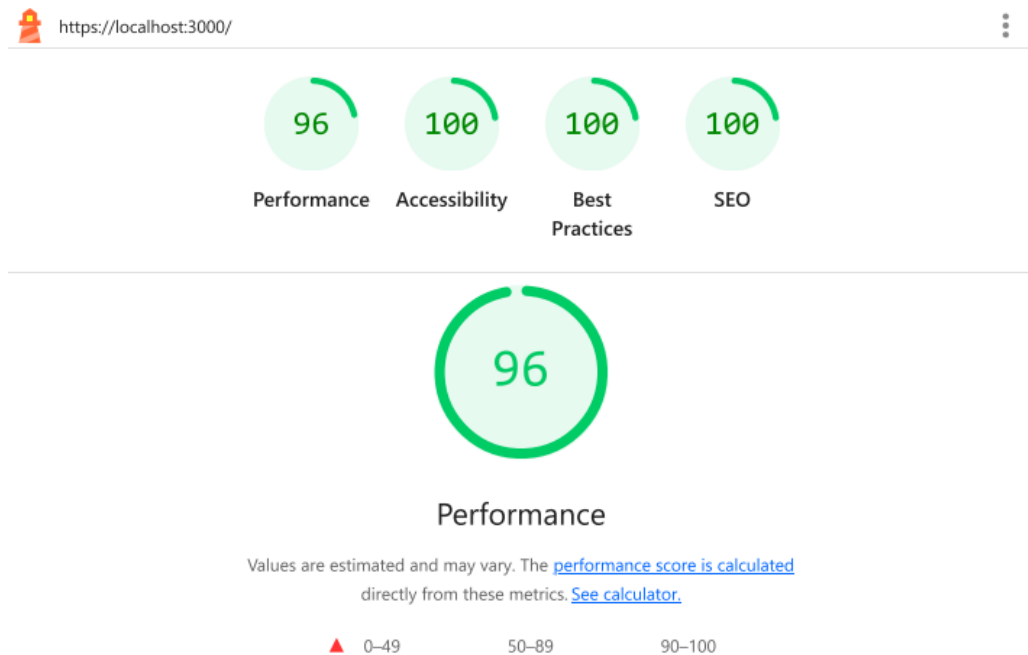


Abbildung 8: Lighthouse Report

Der Accessibility-Score betrug 100 von 100 Punkten, was ein sehr gutes Ergebnis darstellt. Das bedeutet, dass die Webseite in Bezug auf Barrierefreiheit alle Kriterien der WCAG 2.1 erfüllt.

Zusätzlich zeigt der Report eine klare, semantisch strukturierte HTML-Nutzung, was die Bedienung mit Screenreadern erleichtert. Auch die Fokusführung beim Navigieren per Tastatur war korrekt implementiert.

#### 4.4.2. axe DevTools Test

Axe DevTools ist ein weiteres leistungsstarkes Tool zur Barrierefreiheitsanalyse, das von der Firma Deque Systems entwickelt wurde. Für die Tests wurde die kostenlose Browser-Erweiterung genutzt, die grundlegende Checks der Webinhalte erlaubt.<sup>17</sup>

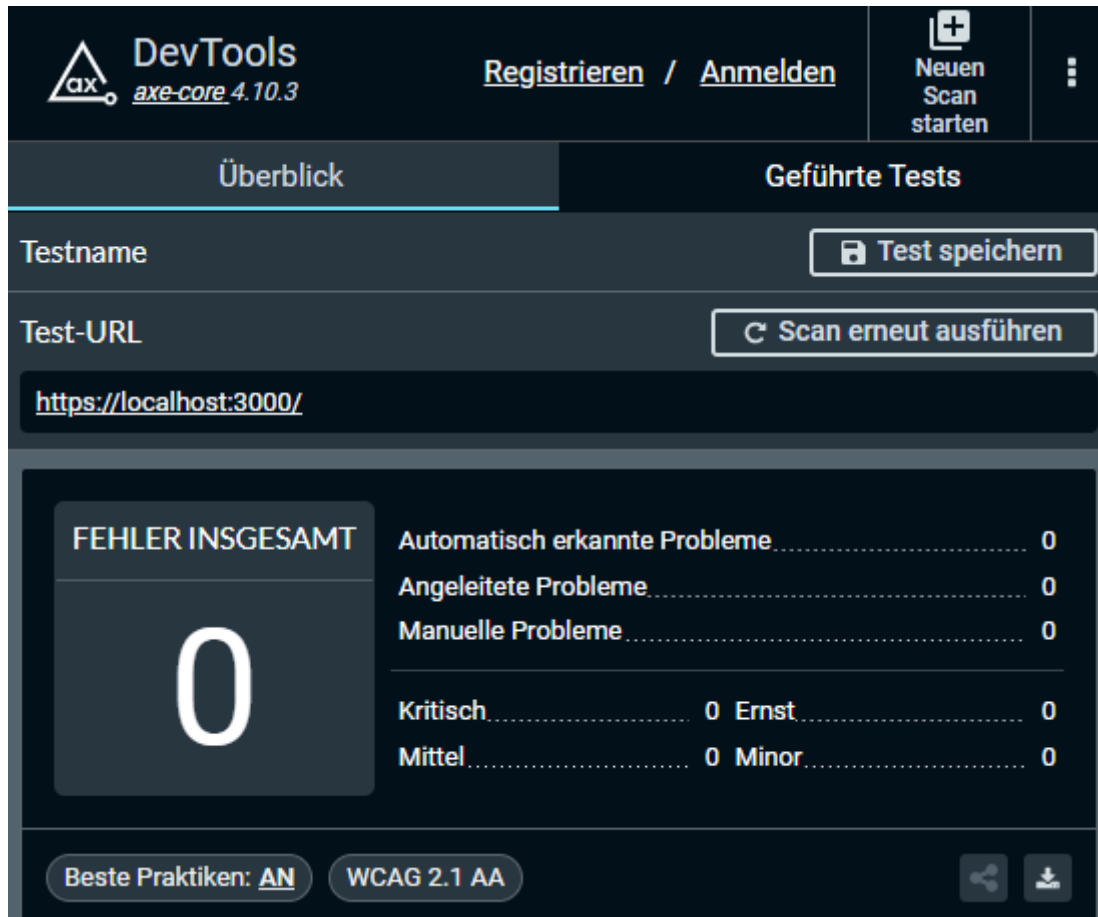


Abbildung 9: axe DevTools Report

Das Ergebnis war ebenfalls sehr positiv: 0 Fehler wurden erkannt. Die Analyse zeigte, dass keine Probleme mit Kontrast, Alternativtexten oder Fokus-Management bestehen. Zudem wurde die Konformität mit der WCAG 2.1 auf Stufe AA bestätigt, was den Industriestandard für Barrierefreiheit darstellt.

<sup>17</sup> URL: <https://chromewebstore.google.com/detail/axe-devtools-web-accessib/lhdoppojpmngadmndnejfpokejbdd>

## 5. Lessons Learned

Während der Entwicklung des FindMe-Projekts und der Auseinandersetzung mit barrierefreiem Webdesign konnten viele wertvolle Erfahrungen gesammelt werden. Bereits in der frühen Planungsphase zeigte sich, dass Barrierefreiheit ein zentraler Bestandteil moderner Webentwicklung sein sollte.

Besonders durch den Einsatz der Tools Google Lighthouse und axe DevTools konnte nachvollzogen werden, wie konkrete technische und gestalterische Aspekte der Barrierefreiheit überprüft werden können. Dabei wurde deutlich, dass bereits mit überschaubarem Aufwand ein hoher Grad an Zugänglichkeit erreicht werden kann. Der hohe Accessibility-Score von Lighthouse und die fehlerfreie Analyse durch axe DevTools bestätigen, dass viele der im Projekt umgesetzten zur Barrierefreiheit beigetragen haben.

Ein zentrales Learning war auch, dass Tests nicht alle möglichen Barrieren erfassen können. Beispielsweise die Verständlichkeit der Inhalte oder die Nutzerführung. Dies kann nur mittels Tests mit Betroffenen getestet werden.

Zusammenfassend kann festgehalten werden, dass Barrierefreiheit kein Add-on ist, das nachträglich ergänzt wird, sondern integraler Bestandteil jedes gut geplanten Webprojekts sein sollte. Die frühzeitige Auseinandersetzung mit WCAG-Richtlinien und unterstützenden Tools hat nicht nur die Qualität der Anwendung verbessert, sondern auch das Bewusstsein für inklusive Gestaltung geschärft.

## 6. Fachbegriffe

**WCAG (Web Content Accessibility Guidelines):** Ein internationaler Standard für barrierefreies Webdesign.

**W3C (World Wide Web Consortium):** Das W3C ist ein internationales Gremium, das Webstandards entwickelt. Ziel ist es, das Internet für alle Menschen zugänglich und nutzbar zu machen.

**Material UI:** Material UI ist eine weit verbreitete React-Komponentenbibliothek, die auf den Designrichtlinien von Google basiert. Sie ermöglicht eine konsistente Gestaltung moderner Webanwendungen.

**Screenreader:** Ein Software-Tool, das blinden oder sehbehinderten Menschen den Inhalt einer Webseite vorliest.

**Semantisches HTML:** Die Verwendung von HTML-Elementen entsprechend ihrer Bedeutung (z. B. <header>, <nav>, <main>, <footer>) zur besseren Orientierung für Screenreader und andere assistive Technologien.

**Kontrastverhältnis:** Der Helligkeitsunterschied zwischen Vorder- und Hintergrundfarbe. Ein zu geringer Kontrast erschwert Menschen mit Sehschwächen das Lesen.

**Responsive Design:** Eine Methode zur Gestaltung von Webseiten, die sich automatisch an die Bildschirmgröße des Endgerätes anpasst.

**SEO (Search Engine Optimization):** Maßnahmen zur Verbesserung der Sichtbarkeit und Auffindbarkeit einer Website in Suchmaschinen.

**Best Practices:** Empfohlene Methoden und Strategien für die Entwicklung qualitativ hochwertiger, sicherer und wartbarer Webanwendungen.

**Google Lighthouse:** Ein Analyse-Tool zur Bewertung von Webseitenqualität in den Bereichen Performance, Barrierefreiheit, Best Practices und SEO. Es zeigt Stärken und Schwächen mit konkreten Verbesserungsvorschlägen auf.

**axe DevTools:** Ein Tool zur automatischen Analyse von Barrierefreiheit. Es prüft Webseiten auf WCAG-Konformität und bietet eine benutzerfreundliche Fehlerdarstellung mit Lösungsansätzen.

**ARIA (Accessible Rich Internet Applications):** ARIA ist eine Sammlung von Attributen, die Webinhalte für Menschen mit Behinderungen besser zugänglich machen. Sie wird verwendet, um zusätzlichen Kontext für Screenreader bereitzustellen.

## 7. Literaturverzeichnis

*Ajda Gregorcic.* (17. 05 2025). Von <https://ajda.gregorcic.eu/2021/03/11/raising-awareness-about-colour-blindness/> abgerufen

*barrierefreies-webdesign.de.* (13. 06 2025). Von <https://www.barrierefreies-webdesign.de/spezial/multimediale-inhalte/behinderung-und-internet.html#:~:text=Nach%20Möglichkeit%20sollten%20allzu%20kleine%20Schaltflächen%20und,muss%20ausschließlich%20über%20die%20Tastatur%20möglich%20sein.> abgerufen

*Der Standard.* (08. 03 2025). *Der Standard.* Von <https://www.derstandard.at/story/2000093943222/wie-farbenblinde-menschen-die-welt-sehen> abgerufen

*Github.* (07. 06 2025). Von <https://github.com/GoogleChrome/lighthouse> abgerufen

<https://barrierefreiheitcheck.de>. (13. 06 2025). Von <https://barrierefreiheitcheck.de/blog/kognitive-barrieren-in-der-webgestaltung> abgerufen

*ialabs.ie.* (14. 04 2025). Von <https://ialabs.ie/understanding-the-pour-principles-of-accessibility/> abgerufen

*IONOS Digital Guide.* (08. 03 2025). Von <https://www.ionos.at/digitalguide/websites/web-entwicklung/barrierefreies-webdesign/> abgerufen

*IONOS Digital Guide.* (08. 03 2025). Von <https://www.ionos.at/digitalguide/websites/web-entwicklung/wcag-richtlinien-fuer-die-barrierefreiheit-im-web/> abgerufen

*MDN Web Docs.* (31. 05 2025). Von <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA> abgerufen

*MDN Web Docs.* (31. 05 2025). Von <https://developer.mozilla.org/en-US/docs/Glossary/SEO> abgerufen

*w3.org.* (14. 04 2025). Von <https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum> abgerufen

*w3.org.* (14. 04 2025). Von <https://www.w3.org/WAI/WCAG22/Understanding/reflow> abgerufen

*w3.org.* (15. 04 2025). Von <https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html#intent> abgerufen

## 8. Abbildungsverzeichnis

Abbildung 1: Konformitätsstufen laut WCAG .....	6
Abbildung 2: Rot-Grün Farbschwäche .....	7
Abbildung 3: W3G.org Size requirement .....	8
Abbildung 4: FindMe Funddatumseingabe .....	9
Abbildung 5: FindMe Beschreibungseingabe .....	10
Abbildung 6: FindMe Alerts .....	11
Abbildung 7: Adobe Color – Kontrastprüfung .....	11
Abbildung 8: Lighthouse Report.....	14
Abbildung 9: axe DevTools Report.....	15

Abbildung 1: <https://www.ionos.at/digitalguide/websites/web-entwicklung/wcag-richtlinien-fuer-die-barrierefreiheit-im-web/>

Abbildung 2: <https://ajda.gregorcic.eu/2021/03/11/raising-awareness-about-colour-blindness/>

Abbildung 3: <https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html#intent>

Abbildung 5: <https://color.adobe.com/de/create/color-contrast-analyzer>