

Entwicklung eines Stammdatenmoduls in Angular mit .NET-API-Anbindung unter Berücksichtigung von Usability und Datenintegrität

Hausarbeit

Hochschule Rhein-Waal
Fakultät: **Kommunikation und Umwelt**
Studiengang: **Medieninformatik**
Abgabe: **23.11.2025**

Workshop II: **Dozent**
Vorgelegt von: **Andreju Douvletis**
Matrikelnummer: **30233**

Contents

| | | |
|----------|--|----------|
| 1 | Einleitung | 1 |
| 1.1 | Relevanz der Arbeit | 1 |
| 1.2 | Vorgehensweise und Aufbau der Arbeit | 1 |
| 2 | Theoretische Grundlagen und Stand der Forschung | 2 |
| 3 | Methodisches Vorgehen | 2 |
| 4 | Konzeption des Artefakts | 2 |
| 4.1 | Anforderungsanalyse | 2 |
| 5 | Umsetzung des Artefakts | 3 |
| 6 | Evaluation | 4 |
| 7 | Reflexion | 4 |
| 8 | Fazit und Ausblick | 4 |

Abstract

Die Entwicklung von Frontend-Anwendungen spielt eine zentrale Rolle bei der Effizienzsteigerung und Prozessoptimierung in modernen Unternehmen. Dieser Hausarbeit mit dem Titel „Entwicklung eines Stammdatenmoduls in Angular mit .NET-API-Anbindung unter Berücksichtigung von Usability und Datenintegrität“ untersucht die Herausforderungen und Potenziale, die mit der Implementierung benutzerfreundlicher und leistungsfähiger Benutzeroberflächen verbunden sind.

Im Fokus stehen sowohl technische als auch organisatorische Aspekte, die während der Migration und Neuentwicklung von Anwendungskomponenten eine Rolle spielen. Dabei werden Strategien zur Strukturierung des Entwicklungsprozesses, zur Entscheidungsfindung im Hinblick auf visuelle Gestaltung und logische Funktionalitäten sowie zum Umgang mit typischen technischen Schwierigkeiten analysiert.

Die Ergebnisse verdeutlichen, dass der Einsatz moderner Frameworks wie Angular, Angular Material und Tailwind maßgeblich zur Standardisierung und Effizienz beiträgt. Gleichzeitig zeigt sich, dass neben der technologischen Umsetzung auch Faktoren wie klare Projektstrukturen und effektive Kommunikation innerhalb des Teams entscheidend für den Projekterfolg sind.

Der Bericht liefert damit nicht nur eine Reflexion über den individuellen Entwicklungsprozess, sondern auch praxisnahe Erkenntnisse, die für zukünftige Projekte im Bereich der Frontend-Entwicklung von Relevanz sein können.

Keywords: Angular, .NET-API-Anbindung, Frameworks, Usability, Datenintegrität

1 Einleitung

In dem ersten Abschnitt der Arbeit geht es um die Problemstellung, die Motivation, die Zielsetzung, die Abgrenzung und die Vorgehensweise. Das zu lösende Problem ist, dass die veraltete Access-Anwendung zu einem erhöhten Arbeitsaufwand führt. Dieses Problem soll durch eine Modernisierung der Seite gelöst werden. Das alte Tool ähnelt einer Monolith Anwendung und soll deswegen durch die Modernisierung mehr in Richtung Microservices entwickelt werden. Ziel des Prototyps ist, eine Weboberfläche zu entwickeln die effizienter und benutzerfreundlicher, als die Access Anwendung, für das Unternehmen ist. Die Forschungsfrage „Wie können UI- und UX-Designentscheidungen bei der Entwicklung eines Angular-basierten Stammdatenmoduls mit .NET-API-Anbindung zur Verbesserung der Usability beitragen?“ soll die technische Umsetzung in den Vordergrund stellen. Dabei sollen die Wirkung der UI- und UX-Designentscheidungen von Quellen und spätere Usertests belegt werden.

1.1 Relevanz der Arbeit

Um einen wissenschaftlichen Beitrag zu leisten, muss die Relevanz der geplanten Arbeit erläutert werden. Dabei wird die Relevanz einmal in einen praktischen und einen theoretischen Bereich aufgeteilt. Ein gutes wissenschaftliches Thema sollte in beiden Bereichen einen Beitrag leisten können. Die praktische Relevanz dieser Arbeit entsteht durch das Verlangen des Unternehmens, eine Access-Anwendung zu modernisieren. Dadurch, dass viele Unternehmen eine Vielzahl an Software verwenden, ist es üblich, dass diese oft gewartet oder von Grund auf neu erstellt werden müssen. Dieses Verhältnis zeigt auf, dass das Thema auch theoretische Relevanz hat, weil es anderen Entwicklern dabei helfen könnte ihre Projekte zu modernisieren. Bei der Entwicklung neuer Lösungen spielt Usability für die Akzeptanz und Benutzerfreundlichkeit eine große Rolle. Zudem ist die Datenintegrität bei Softwarelösungen für das Datenmanagement in jedem Unternehmen ein wichtiges Thema. Deswegen wäre eine Bachelorarbeit mit diesem Thema ein wissenschaftlicher Beitrag, der praktische und auch theoretische Relevanz hat.

1.2 Vorgehensweise und Aufbau der Arbeit

Diese Hausarbeit ist in den ersten Schritten auf die Konzeption und technische Einordnung des Themas fokussiert. Danach wird die Umsetzung des Artefaktes mit Fokus auf die Entwicklungsentscheidungen thematisiert. Darauf wird das erzeugte Artefakt evaluiert. Am Ende wird dann reflektiert und ein Fazit gezogen.

2 Theoretische Grundlagen und Stand der Forschung

Im zweiten Abschnitt geht es um die Grundlagen, verwandte Arbeiten und bestehende Ansätze und relevante Theorien, Modelle oder Designprinzipien. In Bezug auf die Grundlagen werden die Frameworks und nennenswerte Funktionalitäten kurz beschrieben. Bei verwandten Arbeiten und bestehenden Ansätzen wird nach ähnlichen wissenschaftlichen Beiträgen gesucht und der Fund beschrieben. Hier können dann Inspirationen unter anderem aufgeführt werden. Um die Grundlagen abzurunden, werden noch relevante Theorien, Modelle oder Designprinzipien aufgezeigt und erläutert. Dabei sind nicht nur die Methoden, sondern auch ihre Anwendung in der Arbeit relevant.

Der erste Literaturansatz wäre das Buch zu Human Centered Design (Kurosu, 2011). Die Herangehensweise von HDC ist es, das Design nach dem Nutzer auszurichten. Zusätzlich kann die Quelle (Nielsen, 2024) bei der Konzeption des Artefakts berücksichtigt werden. Weitere Ansätze wären Quellen wie (Killisch, 2022) die einen technischen Einblick in die Verwendung von EF Core geben. Weiterhin werden die Quellen (Garcia, 2023) und (Vogel, 2021) für die technische Umsetzung verwendet. Die Quelle (Garcia, 2023) ist nicht aktuell, kann aber für die strukturelle Entwicklung mit Angular verwendet werden. Beim arbeiten mit dem Sourcecode der alten Oberfläche könnte die Quelle (Ala-Hulkko, 2025) hilfreich sein.

3 Methodisches Vorgehen

Die Bachelorarbeit verfolgt eine prototypische Methodik, das bedeutet, dass die Dokumentation des Entwicklungsprozesses ein zentraler Aspekt der Arbeit ist. Die praktische Umsetzung der Oberfläche sollte einen tieferen Einblick in die Wichtigkeit der Entwicklungsentscheidungen liefern. Im Kern soll die Korrelation zwischen UI- und UX-Designentscheidungen und der späteren Usability des Artefaktes bestimmt werden. Um diese Korrelation zu bestimmen, werden Designprinzipien wie „Human Centered Design“ und Usertests verwendet.

Hier wird auch die Verwendung der DSR Methode im Vorgehen der Arbeit beschrieben.

4 Konzeption des Artefakts

Im folgenden Abschnitt geht es um die Planung für die Entwicklung des Artefakts. Um das Artefakt gut planen zu können, sollte vorher eine Anforderungsanalyse angefertigt werden, um herauszufinden, welche Aspekte des Prototyps priorisiert werden sollten. Es ist ebenfalls wichtig, zu planen, welche Designprinzipien oder Methoden während der Entwicklung verwendet werden sollen. Außerdem sind ein konzeptionelles Modell und ein Plan für die Systemarchitektur hilfreich, um bei der Umsetzung Zeit zu sparen. Zusätzlich zu der Anforderungsanalyse sollten hier auch geplante Funktionalitäten und Interaktionen dokumentiert werden.

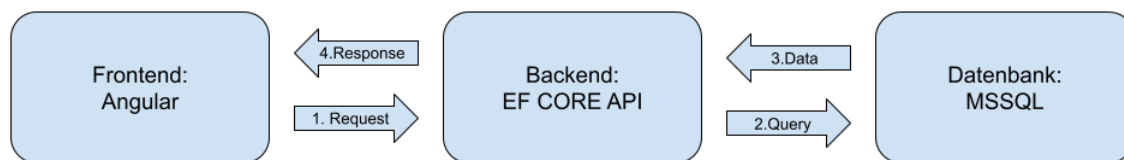


Figure 1: Datenfluss in dem Projekt

4.1 Anforderungsanalyse

Die Anforderungsanalyse ist ein wichtiger Bestandteil der Konzeption des Artefakts. In diesem Abschnitt werden die funktionalen und nicht-funktionalen Anforderungen an das System beschrieben. Funktionale

Anforderungen beziehen sich auf die spezifischen Funktionen und Features, während nicht-funktionale Anforderungen Aspekte wie Leistung, Sicherheit und Benutzerfreundlichkeit abdecken.

Der Prototyp soll eine moderne Weboberfläche für die Stammdatenverwaltung bieten. Zu den funktionalen Anforderungen gehören das erstellen, bearbeiten und löschen von Stammdaten. Nicht-funktionale Anforderungen umfassen eine intuitive Benutzeroberfläche, schnelle Ladezeiten und die Gewährleistungen der Datenintegrität.

5 Umsetzung des Artefakts

Bei der technischen Umsetzung des Artefaktes geht es vor allem um die Entwicklungsentscheidungen, Implementierungsdetails und Herausforderungen. Die Begründung der Entwicklungsentscheidungen und das Aufzeigen von Alternativen sind für diesen Abschnitt entscheidend.

Die Umsetzung des Prototyps erfolgt mit dem Angular Framework für das Frontend und .NET für das Backend. Dabei werden Angular Material und Tailwind CSS für die grafische Gestaltung der Oberfläche verwendet. Die Anbindung an die Datenbank erfolgt über eine .NET-API die Entity Framework Core nutzt.

Um die Oberfläche zu erstellen werden für die unterschiedlichen Seiten Komponenten in Angular erstellt. Diese Komponenten sind für die Darstellung und Interaktion der Benutzeroberfläche verantwortlich. Die Kommunikation zwischen dem Frontend und dem Backend erfolgt über HTTP-Anfragen. Es werden GET, POST, PUT und DELETE Requests verwendet, um Daten abzurufen, zu erstellen, zu aktualisieren und zu löschen (Garcia, 2023). Für die unterschiedlichen Abfragen wird in .NET ein Controller erstellt, der die Abfragen entgegennimmt und in der Service Datei im Backend verarbeitet.

Entity Framework Core ist ein ORM (Object-Relational Mapping) Framework, das die Datenmodelle des Backends mit der Datenbank vergleicht. Änderungen in den Datenmodellen werden durch die SaveChanges-Methode (Vogel, 2021) zur Datenbank übermittelt.

Änderungen werden durch LINQ-Abfragen (Vogel, 2021) durchgeführt, die eine einfache und lesbare Möglichkeit bieten, Datenbankoperationen durchzuführen. Es gibt zwei Arten von Syntax, mit der LINQ Queries geschrieben werden können. Zum einen können LINQ Queries in der Methodensyntax 2 geschrieben werden, zum anderen in der Abfragesyntax.

```
var persons = from p in Persons
               where p.firstName == "Susan"
               select p;
```

Listing 1: Beispiel für eine LINQ Query in Abfragesyntax (Vogel, 2021)

Die Syntax in Listing 1 zeigt eine LINQ Query in der Abfragesyntax, die alle Personen mit dem Vornamen „Susan“ aus der Personen Tabelle abrufen. Die Syntax gleicht einer SQL-Abfrage und ist für Entwickler, die mit SQL vertraut sind, leicht verständlich. Die Methodensyntax verwendet Fluentapi (Vogel, 2021), um die Abfrage aneinander zu ketten.

```
var persons = Persons.Where(x => x.LastName == "Smith").ToList();
```

Listing 2: Beispiel für eine LINQ Query in Methodensyntax (Vogel, 2021)

Die Syntax in Listing 2 zeigt eine LINQ Query in der Methodensyntax, die alle Personen mit dem Nachnamen „Smith“ aus der Personen Tabelle abrufen. In der Abfrage wird eine Arrowfunktion verwendet, um die Query Bedingung zu definieren.

Um die Usability des Prototyps zu verbessern, werden UI- und UX-Designprinzipien angewendet. Dazu gehört die Verwendung von klaren und konsistenten Layouts, intuitiven Navigationselementen und

ansprechenden visuellen Designs. Die Entwicklung orientiert sich am Human Centered Design Ansatz (Kurosu, 2011), um sicherzustellen, dass die Bedürfnisse und Erwartungen der Benutzer berücksichtigt werden. Damit die UI- und UX-Designentscheidungen fundiert sind, wird auf etablierte Designprinzipien und Heuristiken zurückgegriffen (Nielsen, 2024).

6 Evaluation

Um die Anforderungen zu evaluieren, wird die zuvor erstellte Anforderungsanalyse als Grundlage verwendet. Es wird überprüft, ob alle funktionalen und nicht-funktionalen Anforderungen erfüllt wurden. Die Ergebnisse der Evaluation werden dokumentiert und im Team diskutiert. Zusätzlich werden Usertests durchgeführt, um die Usability des Prototyps zu bewerten. Dadurch kann festgestellt werden, ob die UI- und UX-Optimierungen tatsächlich zu einer Verbesserung der Usability geführt haben. Nach der Evaluation werden über mögliche Verbesserungen und Anpassungen diskutiert.

7 Reflexion

Im vorletzten Abschnitt wird der Entwicklungsprozess reflektiert. Hierbei werden die Design- und Entwicklungsentscheidungen nochmal beleuchtet und kritisch hinterfragt. Auch die Limitationen und Erkenntnisse werden für zukünftige Arbeiten beschrieben. Zum Beispiel könnte die Entscheidung Methodensyntax für LINQ Queries zu verwenden reflektiert werden. Als Alternative hätte der Autor die Abfragesyntax verwenden können. Hier können die Vor- und Nachteile der beiden Syntaxen diskutiert werden. Weiterhin können die getroffenen Designentscheidungen reflektiert werden. Wurden die Designprinzipien eingehalten? Wären andere Entscheidungen angemessener gewesen? Konnte der Zeitplan eingehalten werden? Gab es unvorhergesehene Herausforderungen? Auch die tatsächliche Zeiteinteilung kann hier im Bezug zum Zeitplan aufgeführt werden. Außerdem werden Empfehlungen abgeleitet, die bei Projekten in der Zukunft berücksichtigt werden sollten.

8 Fazit und Ausblick

Im letzten Abschnitt wird das Ergebnis der Arbeit zusammengefasst und die Forschungsfrage beantwortet. Zum Schluss wird ein Ausblick auf die zukünftigen Entwicklungen formuliert.

References

- [1] Tero Ala-Hulkko. *Modernization of a legacy codebase*. 2025. URL: https://www.theseus.fi/bitstream/handle/10024/894263/Ala-Hulkko_Tero.pdf?sequence=2.
- [2] Victor Hugo Garcia. *Getting Started with Angular*. Apress, 2023.
- [3] Benjamin Uwe Killisch. *Effiziente Behandlung von Daten mit rekursiven Beziehungen in ORM-Frameworks am Beispiel von Entity Framework Core*. 2022. URL: <https://htwk-leipzig.qucosa.de/api/qucosa%3A83322/attachment/ATT-0/?L=1>.
- [4] Massaki Kurosu. *Human Centered Design*. Springer, 2011.
- [5] Jakob Nielsen. *10 Usability Heuristics for User Interface Design*. 2024. URL: <https://www.nngroup.com/articles/ten-usability-heuristics>.
- [6] Eric Vogel. *Beginning Entity Framework Core 5*. Apress, 2021.