

# ENGEL - Translate

## DIPLOMARBEIT

Höhere Abteilung für Informatik

01/07/2024 – 04/04/2025

**Projektmitglieder:** Elisabeth Brandstetter  
Olivia Kamptner

**Betreuerin:** Prof.in. OStRin Maga. Gabriela Danner



# Eidesstattliche Erklärung

Hiermit versichern wir, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von uns angegebenen Quellen angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Bei der Erstellung der Arbeit haben wir die generativen KI-Tool ChatGPT zu folgendem Zweck verwendet: Recherche. Entsprechende Stellen sind gekennzeichnet.

03. April, 2025

Elisabeth Brandstetter

03. April, 2025

Olivia Kamptner

# Gendererklärung

Zur besseren Lesbarkeit wird in dieser Diplomarbeit durchgehend die weibliche Form verwendet. Diese sprachliche Entscheidung dient der Vereinfachung und soll das Verständnis des Textes fördern. Selbstverständlich sind damit stets Personen aller Geschlechter gleichermaßen gemeint. Diese Formulierung ist bewusst gewählt und soll keine Geschlechterdiskriminierung symbolisieren.

03. April, 2025

Elisabeth Brandstetter

03. April, 2025

Olivia Kamptner

# Danksagung

An dieser Stelle möchten wir uns herzlich bei allen Personen bedanken, die uns während der Erstellung dieser Diplomarbeit mit ihrer Unterstützung begleitet haben.

Ein großer Dank gilt der Firma ENGEL und unserer Betreuerin Gudrun Rockenschaub, die uns die Möglichkeit gegeben haben, diese spannende Arbeit durchzuführen. Ein besonderer Dank gebührt dabei Richard Erda, der uns vor Ort fachlich unterstützt und mit seiner Erfahrung maßgeblich zum Erfolg der Diplomarbeit beigetragen hat.

Ein weiterer Dank geht an unseren ursprünglichen Auftraggeber Thomas Gahleitner sowie an die spätere Auftraggeberin Christina Haider, die uns jederzeit mit konstruktivem Feedback unterstützt haben.

Wir möchten uns außerdem bei unserer Betreuerin, Frau Prof. Danner, für ihre wertvolle Begleitung, ihr Organisationstalent und Leitung und die stets offenen Ohren während der gesamten Projektzeit bedanken.

Ebenso möchten wir Herrn Professor Otto danken, der uns mit seinem fundierten Wissen über Microsoft-Technologien unterstützt und damit wesentlich zur Umsetzung unserer Arbeit beigetragen hat.

Unser besonderer Dank gilt unserem guten Bekannten Karl C. G., der uns stets mit hilfreichen Ratschlägen zur Seite stand und uns motivierte, unser Bestes zu geben.

Allen Beteiligten möchten wir aufrichtig für ihr Engagement und ihre Unterstützung danken – ohne Sie wäre diese Arbeit in dieser Form nicht möglich gewesen.

# Zusammenfassung

Die Firma ENGEL AUSTRIA GmbH stellt Spritzgussmaschinen sowie benötigte Automatisierungsanlagen für Kundinnen auf der ganzen Welt her. Sie hat den Firmensitz im in Oberösterreich liegenden Schwertberg und beschäftigt mehr als 7000 Mitarbeiterinnen.

The logo for ENGEL, consisting of the word "ENGEL" in a bold, black, sans-serif font.

Da ENGEL Standorte weltweit hat und die Übersetzung von Texten sowie Dokumenten eine große Rolle spielt, um sich nicht nur verständigen zu können, sondern auch um die Wiederverwendbarkeit von bereits existierenden Dokumenten zu gewährleisten und somit Zeit zu sparen, entstand der Wunsch einer einheitlichen Übersetzungsmöglichkeit. Das Anliegen der Kundin war es, ein Tool zu erhalten, das beides, also Texte und Dokumente, übersetzen kann. Ebenso ein Anliegen dabei war ihr das Glossar einzubinden, um Fachbegriffe richtig zu übersetzen, um Missverständnisse zu verhindern und folgende Probleme zu vermeiden.

Auf der Webseite ENGEL-Translate ist es den ENGEL-Mitarbeiterinnen möglich, Texte einzugeben sowie Dokumente hochzuladen und diese übersetzen zu lassen. Die Seite basiert auf Benutzerfreundlichkeit und ermöglicht es den Benutzerinnen die Oberflächensprache auf Englisch oder Deutsch umzustellen.

Für das Übersetzen selbst ist eine Übersetzer-Ressource in Microsoft Azure verantwortlich. Wer Zugriff auf das Übersetzungstool hat, wird von Microsoft Azure Entra geregelt und der Zugriff wird daher externen Personen nicht gewährt.

Ziel der Arbeit war in den grundlegenden Funktionalitäten eine Webanwendung für die Firma zu entwickeln. Darüber hinaus sollten verschiedene Erweiterungen (Abrechnung, Datenbankanbindung, Einbindung des firmeneigenen Glossars) im Rahmen eines Proof of Concept soweit vorbereitet werden, dass Entscheidungen diesbezüglich von der Firma ENGEL in einer Weiterentwicklung genutzt werden können.

# Abstract

ENGEL AUSTRIA GmbH manufactures injection moulding machines and the necessary automation systems for customers all over the world. It is headquartered in Schwertberg, Upper Austria, and employs more than 7000 people.



Since ENGEL has locations all over the world and the translation of texts and entire documents plays a major role, not only to be able to communicate, but also to ensure the reusability of existing documents and thus save time, the desire for a uniform translation option arose. The customer's request was to have a tool that could translate both texts and documents. It was also important to her to integrate the glossary for translating technical terms correctly in order to prevent misunderstandings and avoid subsequent problems.

On the ENGEL-Translate website, ENGEL employees can enter texts, upload documents, and have them translated. The site is based on user-friendliness and allows users to change the interface language to English or German.

A translator resource in Microsoft Azure is responsible for the translation itself. It is governed by Microsoft Azure Entra, who has access to the translation tool and therefore access is not granted to external parties.

The aim of the work was to develop a web application for the company with basic functionality. Furthermore, various extensions (billing, database connectivity, integration of the company's own glossary) were to be prepared as part of a proof of concept to the extent that ENGEL could use these decisions in further development. <sup>1</sup>

---

<sup>1</sup>Generated with the help of ENGEL-Translate.

# Inhaltsverzeichnis

<b>1</b>	<b>Ausgangslage</b>	<b>1</b>
<b>2</b>	<b>Zielsetzung</b>	<b>2</b>
<b>3</b>	<b>Entwurfsphase 1</b>	<b>4</b>
3.1	Architekturbild . . . . .	4
3.2	Komponenten . . . . .	4
3.3	Spezifiaktionen . . . . .	6
3.4	Backend - Frontend . . . . .	6
3.5	Design . . . . .	7
3.6	Fazit der Phase . . . . .	11
<b>4</b>	<b>Entwurfsphase 2</b>	<b>12</b>
4.1	Übersetzungs-Ressource . . . . .	12
4.2	Datenbank . . . . .	13
4.3	Frontend Framework [1][2] . . . . .	15
4.4	Abrechnung . . . . .	17
4.5	Fazit der Phase . . . . .	18
<b>5</b>	<b>Implementierung</b>	<b>19</b>
5.1	Programmierung . . . . .	19
5.2	Datenbank . . . . .	23
5.3	Abrechnung . . . . .	23
5.4	Fazit . . . . .	24
<b>6</b>	<b>Deployment</b>	<b>25</b>
6.1	Azure . . . . .	25
6.2	Deployment des Backends [3] [4] . . . . .	25
6.3	Deployment der Authorization-Applikation . . . . .	26
6.4	Fazit . . . . .	27

<b>7</b>	<b>Machbarkeitsstudie Glossar</b>	<b>28</b>
7.1	Einleitung . . . . .	28
7.2	Fragestellungen und Überlegungen . . . . .	28
7.3	Ergebnis . . . . .	29
7.4	Zusammenfassung . . . . .	32
7.5	Fazit . . . . .	32
<b>8</b>	<b>Ergebnis Produkt</b>	<b>33</b>
8.1	Technologien . . . . .	33
8.2	Funktionalität . . . . .	36
<b>9</b>	<b>Zusätzliche Ergebnisse</b>	<b>39</b>
9.1	Technologien . . . . .	39
9.2	Inhalt . . . . .	41
<b>10</b>	<b>Organisation</b>	<b>44</b>
10.1	Vorgehensmodell [5] . . . . .	44
10.2	Technologien . . . . .	47
10.3	Projektteam . . . . .	52
<b>11</b>	<b>Resümee</b>	<b>55</b>
11.1	Resümee Auftraggeberin . . . . .	55
11.2	Resümee Brandstetter Elisabeth . . . . .	57
11.3	Resümee Kamptner Olivia . . . . .	57
	<b>Glossar</b>	<b>V</b>
	<b>Literaturverzeichnis</b>	<b>VIII</b>
	<b>Abbildungsverzeichnis</b>	<b>X</b>
	<b>Tabellenverzeichnis</b>	<b>XI</b>
	<b>Anhang</b>	<b>XII</b>
A	Aufgabenverteilung . . . . .	XII
B	Meilensteine . . . . .	XII
C	ChatGPT . . . . .	XII

# 1 Ausgangslage

Die Firma Engel hat 146 Standorte die verteilt sind auf Europa, Asien, Nordamerika, Südamerika und Australien. Die meist gesprochenen beziehungsweise benötigten Sprachen sind Deutsch, Englisch, Tschechisch, Chinesisch, Koreanisch und Indisch welche eine gewisse Sprachbarriere bilden. Diese Barriere wird derzeit mit Google Translate zur Textübersetzung, DeepL zur Text und Dokumentenübersetzung oder ChatGPT und anderen Tools überwunden. Dies kostet natürlich Zeit und ist nicht immer kostenlos. Mitarbeiterinnen übersetzen sensible Dateien mit firmen-externen Tools, welche ein Risiko darstellen. Da hoher Wert auf Datensicherheit gelegt wird, entstand der Wunsch nach einem eigenen internen Übersetzungstools, um Texte sowie Dokumente auf derselben Seite übersetzen zu können, auf die jede Engel-Mitarbeiterin Zugriff hat.

# 2 Zielsetzung

## Produktziele

Das Ziel ist, eine Webanwendung zu implementieren, die für alle Engel Mitarbeiterinnen erreichbar und verfügbar ist und ihren Bedürfnissen entspricht, sich jedoch auch von bereits eingesetzten Übersetzungstools abhebt. Diese Webanwendung soll Fließtext, sowie Dokumente übersetzen, ohne das Format zu verändern. Dabei werden Services, die auf KI basieren verwendet, um die Qualität und Effizienz zu steigern. Weiters soll die Website selbst auf Englisch, sowie Deutsch verfügbar sein. Nicht zu vergessen ist ein benutzerfreundliches Layout, um die Verwendung des Tools für Mitarbeiterinnen aller Altersgruppen attraktiv zu machen. Die Angaben der Auftraggeberin wurden wie folgt klar kommuniziert und dokumentiert:

Im Anmeldebereich wird Single-Sign-On verwendet.

Auf der Startseite soll das ENGEL Logo, sowie der Name des Übersetzungstools - ENGEL Translate - verwendet werden. Weiters soll das ENGEL-Grün eine wichtige Grundlage des Designs sein und darf dabei nicht fehlen.

Die Navigationsleiste (links) ermöglicht der Benutzerin das einfache Navigieren auf der Webseite. Es gibt eine kleine Beschreibung über technische Details und Informationen zum Tool selbst, ebenfalls wird ein Link zum IT-Helpdesk eingebunden, um bei Fragen Hilfe zu bekommen.

Der Sicherheitshinweis, dass die Übersetzung selbst überprüft werden muss, soll ebenso ein wichtiger Bestandteil sein, um Folgefehler zu vermeiden. Das Übersetzungsfeld soll zentral eingebettet sein und Benutzerinnen sollen einen Text eingeben können. Zum Auswählen der gewünschten Zielsprache gibt es ein Dropdown-Menü, ebenso für die Eingabesprache. Diese Dropdown-Menüs sind sehr wichtig, da sie an das Backend und somit den Übersetzungsdienst im Hintergrund weitergeleitet werden und zur Übersetzung benötigt werden. Den übersetzten Text kann man mittels eines Buttons in die Zwischenablage kopieren.

Die Auswahl der Sprachen funktioniert für die Dokumentenübersetzung genau gleich wie bei der Textübersetzung. Darüber hinaus gibt es eine Drop-in Funktion, sowie eine Durchsuch-Funktion

für Dokumente. Das übersetzte Dokument wird direkt einfach und simpel heruntergeladen. Es werden Dateien mit dem Dateiformat Word (.docx), PowerPoint (.pptx), und anderen, auf der Webseite angeführten Dateiformaten, unterstützt, jedoch wird nur ein Dokument (pro Übersetzung) bis zu einer maximalen Dateigröße von 25 MB genehmigt. Die Übersetzung erfolgt für folgende Sprachen: Deutsch, Englisch, Tschechisch, Chinesisch, Koreanisch, Indisch.

Ein ENGEL spezifischen Glossars mit Fachbegriffen wird automatisch im Hintergrund eingelesen.

Es wird eine Schnittstelle zum Anbinden von Drittsystemen zur Verfügung gestellt. Ziel ist es am Ende eine Website auf Azure zu hosten. Diese soll einen Blob-Service zum Zwischenspeichern der Dokumente und ASPX-Seiten zum Anzeigen der Benutzeroberfläche verwenden. Eine nutzungsbasierte Zahlung auf Kostenstellenbasis und Auswertungsmöglichkeit im Azure Portal, um zu evaluieren welche Abteilung beziehungsweise Kostenstelle das Tool am meisten verwendet, soll implementiert werden.

Die Website wird auf Deutsch und Englisch bereitgestellt. Dabei dürfen importierte Daten nicht außerhalb des ENGEL Netzwerks verwendet oder gespeichert werden. Ebenfalls werden dafür geeignete KI-Service wie z.B. Azure Cognitive Services, Azure Translator Text API oder Azure Open AI Service eingesetzt.

### **Geschäftsziele**

Ziel von ENGEL AUSTRIA ist es, die Übersetzung selbst intern zu machen damit keine sensiblen Dokumente an externe Übersetzer geschickt werden müssen. Somit gewährleistet man die Kontrolle über die eigenen Dateien. Ein weiterer Vorteil, der sich ergibt, sind die Kosten. Da nun keine Lizenzen oder Abonnements mehr bezahlt werden müssen, sondern nur mehr pro Wort/MB bezahlt werden muss, spart das viel Geld ein. Beispielsweise wenn eine Mitarbeiterin nur ein mal im Monat ein Dokument übersetzt, muss trotzdem der gesamte Preis für die Lizenz oder da Abonnement bezahlt werden. Mit Single-Sign-On wird ebenso Zeit gespart, weil man sich nicht einloggen muss.

# 3 Entwurfsphase 1

## 3.1 Architekturbild

Die Auftraggeberin und das Projektteam haben gemeinsam ein Architekturbild erstellt, welches darstellt, wie das Projekt aufgebaut sein soll.

## 3.2 Komponenten

### 3.2.1 Backend

Das Backend dient der Verbindung von Übersetzungs-Ressource zum Frontend. Es kümmert sich um die Authentifikation des Requests welcher vom Frontend an das Backend gesendet wird. Dazu werden Token verwendet, welche dem Frontend Zugriff auf die API gewährleisten.



Abbildung 1: Gemeinsame Erarbeitung der Architektur

## Azure Cognitive Services

Azure Cognitive Services bieten eine Sammlung von KI-Diensten und kognitiven APIs, die es Entwicklerinnen ermöglichen, Anwendungen zu erstellen, die sehen, hören, sprechen, verstehen und sogar menschliche Entscheidungen nachahmen können. Diese Dienste sind darauf ausgerichtet, Entwicklerinnen zu helfen, KI-Funktionen ohne tiefgreifendes Fachwissen über Datenwissenschaft zu integrieren. Sie umfassen verschiedene Kategorien wie Vision, Sprache, Spracherkennung, und Entscheidungsfindung. Ein Beispiel ist die Azure Translator Text API, die Textübersetzungsdienste anbietet.

### Azure OpenAI Service

Azure OpenAI Service hingegen ist eine speziellere Plattform, die auf OpenAI's fortschrittlichen KI-Modellen basiert, wie z.B. GPT-3 und Codex. Diese Dienste sind darauf ausgelegt, um Entwicklerinnen Zugang zu leistungsstarken Sprachmodellen zu geben, die für eine Vielzahl von Anwendungen verwendet werden können, von der natürlichen Sprachverarbeitung bis hin zur Codegenerierung. Der Azure OpenAI Service bietet Entwicklerinnen die Möglichkeit, diese fortschrittlichen KI-Modelle in ihre Anwendungen zu integrieren, um komplexe Aufgaben wie das Verstehen und Generieren von menschlicher Sprache zu bewältigen.[6]

Anforderung	Cognitive Services	Open AI
Fließtextübersetzung	ja	ja
Dokumentenübersetzung	ja	nein
Dokumente bleiben formatiert	ja	nein
Glossar Einbindung	ja	ja (umständlich)
Codesprache	Java	Python
Abfrage	JSON, Sprachenkürzel in URL	ganze Sätze (ChatGPT)

Tabelle 1: Tabellarischer Vergleich von Cognitive Services und Open AI

### Fazit

Zusammenfassend lässt sich sagen, dass Azure Cognitive Services eine breitere Palette von KI-Diensten für verschiedene Zwecke bietet, während Azure OpenAI Service spezialisierte KI-Modelle für fortgeschrittene Sprach- und Textaufgaben zur Verfügung stellt. Beide Dienste haben das Ziel, die Integration von KI-Funktionen in Anwendungen zu vereinfachen und Entwicklerinnen leistungsstarke Tools zur Verfügung zu stellen. Jedoch passen die Eigenschaften der Azure Cognitive Services für unseren Anwendungsfall besser.

### 3.2.2 Benutzeroberfläche (UI)

Hier können Benutzerinnen Texte eingeben oder Dokumente hochladen, die übersetzt werden sollen. Es soll eine einfach bedienbare Webanwendung sein. Empfehlung für Programmiersprachen: .NET oder Java. Da die Firma ENGEL schon mit diesen Programmiersprachen arbeitet, ist es eine Anforderung an das Projekt, auch hierbei einer dieser Sprachen zu verwenden.

### 3.2.3 API-Gateway

Dient als Schnittstelle zwischen der Benutzeroberfläche und den Backend-Diensten. Optional soll es noch eine zweite API-Schnittstelle geben, welche für Drittanbieterinnen zur Verfügung steht.

### 3.2.4 Datenverwaltung (DB)

Speichert Übersetzungen und Benutzerinnenpräferenzen. Zu diesen zählt zum Beispiel die Einstellung, in welcher Sprache die Webseite erreichbar ist. Diese Art von Informationen sollen in einer Datenbank gespeichert werden.

### 3.2.5 Authentifizierungsservice (ID, Azure ENTRA)

Verwaltet Benutzerinnenkonten und Zugriffsrechte (jwt Token, Identity Provider). Das ist relevant für den Abrechnungsdienst im Backend.[7]

Der Azure Tenant soll sicherstellen, dass verschiedene Microsoft-Dienste, wie Azure-Abonnements und Microsoft 365, durch eine geschützte Authentifizierung und Autorisierung genutzt werden können.

## 3.3 Spezifikationen

Die Wünsche der Kundin an das Projekt. (ENGELAzureTranslateSpezifikation.pdf)

Die Nachfolgenden Punkte beschreiben die Wünsche und Anforderungen der Kundin für das Frontend, Backend und inhaltliche Anforderungen.

## 3.4 Backend - Frontend

In Verbindung mit den Anforderungen, welche oben im verlinkten PDF-Dokument (befindet sich im Anhang) beschrieben werden, wurden im Team bei einer Besprechung weitere Anforderungen beschlossen.

### 3.4.1 Anforderungen an das Backend

- Es soll API-Calls zur Translator Ressource machen.
- Die Authentifizierung des Users erfolgt durch Azure Entra (API).
- Eine Verbindung zur Datenbank soll User-Daten, beziehungsweise Einstellungen der Sprache auf der Webseite holen.

- Es stellt die Verbindung zwischen Frontend und Translator Ressource dar.
- OPTIONAL soll es noch eine API-Schnittstelle für Drittanbieterinnen bereitstellen.

### 3.4.2 Anforderungen an das Frontend

- Für die Erstellung der Oberfläche in Java sollen JSP-Seiten verwendet werden.
- Das Frontend verwaltet und verwendet XML-Files für die Spracheinstellung auf der Webseite.
- Kommunikation mit Backend-API
  - Das Frontend sendet Text/Datei an das Backend zum Übersetzen.
  - Der übersetzten Text/die übersetzte Datei wird dem User zur Verfügung gestellt. Der übersetzte Text kann in die Zwischenablage kopiert werden. Die übersetzte Datei wird im Browser heruntergeladen.
- Style der Webseite
  - Es sollen Farben definiert und überall gleich verwendet werden (ENGEL-Grün!)
  - Logos einbinden (ENGEL & ET)
- Sicherheitshinweise
  - Eine Garantie der Richtigkeit der Übersetzung ist nicht zu 100 Prozent gewährleistet und sollte bei sensiblen Texten auf jeden Fall überprüft werden.
  - IT-Support-Link für den Help-Desk von ENGEL, falls etwaige Probleme auftreten sollten.
- Feedback Funktion
- Schulungsvideo einbetten

## 3.5 Design

### 3.5.1 Logo

Das Logo wurde vom Projektteam entworfen und von der Kundin akzeptiert. Die einzige Bitte an das Logo war, das offizielle Grün der Firma ENGEL zu verwenden. Das Logo stellt die Anfangsbuchstaben des Namens des Projektes dar. Dabei wurde darauf geachtet, dass das 'E', welches für ENGEL steht, auch in dessen bestimmter Farbe dargestellt wird. Der Trennstrich zwischen den Buchstaben soll die Trennung von zwei Übersetzungen darstellen und gleichzeitig klar machen, dass es zwei Wörter sind, aus denen das Logo besteht.



Abbildung 2: Endgültiges Logo

### 3.5.2 Mockup

Die Abbildung unten verwendet noch den Namen ETT, welcher mit Rücksprache mit dem Kunden durch ET ersetzt wurde.

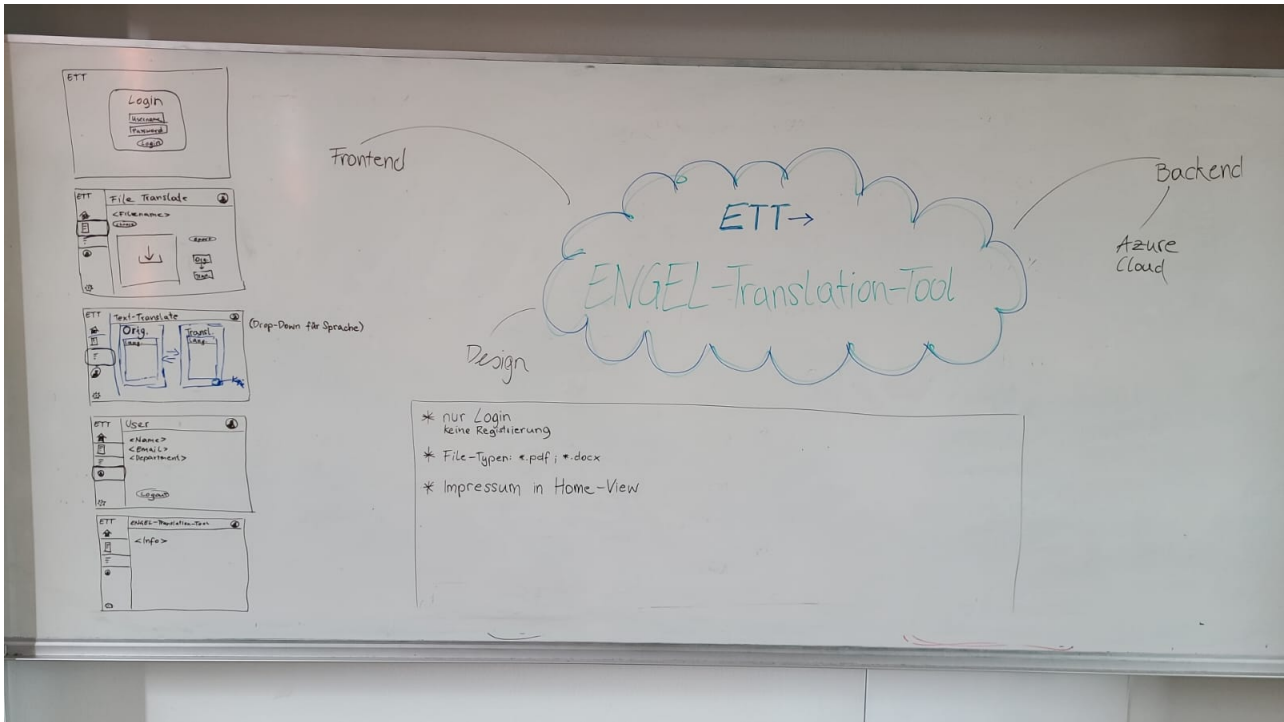


Abbildung 3: Foto vom Brainstorming des Mockups

### Login

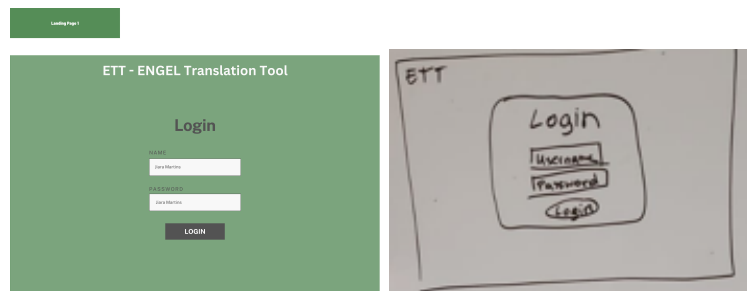


Abbildung 4: Login Seitendesign in Canva (links)

Die Login-Page soll durch das SSO ersetzt werden, und dadurch die Authentifizierung für die Benutzerin vereinfachen.

## Home

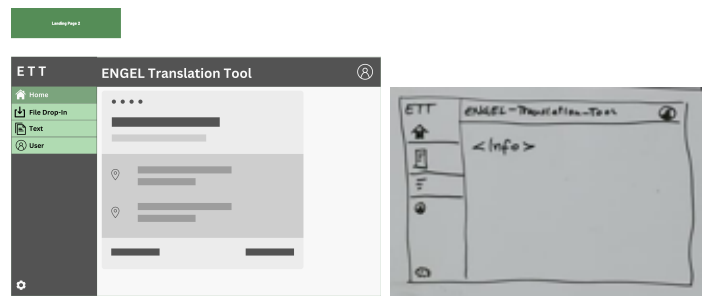


Abbildung 5: Home Seitendesign in Canva (links)

Auf der Home-Page sollen Hinweise und allgemeine Informationen angeführt werden. Dazu zählen eine kurze Information, was dieses Tool überhaupt macht und der Hinweis, dass die Übersetzungen dieses Tools von einer KI gemacht werden und nicht zu 100% verlässlich sein können. Außerdem gibt es eine Möglichkeit mit einem Link eine Email an den Helpdesk der Firma ENGEL zu schreiben.

## Textübersetzung



Abbildung 6: Textübersetzung Seitendesign in Canva (links)

Diese Seite lehnt sich vom Layout her stark an Google Translate an, da dieses Layout die meisten Benutzerinnen bereits kennen und sich damit zurechtfinden können.

## Dokumentenübersetzung



Abbildung 7: Dokumentenübersetzung Seitendesign in Canva (links)

### 3.5 Design

Dokumente können mithilfe des Hochladen Buttons oder mit Drag-and-Drop eingefügt werden. Die unterstützten Dateiformate sind hierbei angeführt. Das Dokument wird nach dem Übersetzen sofort heruntergeladen.

#### User



Abbildung 8: User Seitendesign in Canva (links)

Die User-Page bietet eine Möglichkeit Daten zum eingeloggtten User anzuzeigen. Diese Seite ist nach Besprechung mit der Kundin als überflüssig markiert worden, und daher nicht implementiert.

## 3.6 Fazit der Phase

In der ersten Entwurfsphase des Projekts wurde die grundlegende Architektur definiert und ein klares Verständnis der Komponenten erarbeitet. Die zentrale technische Grundlage bilden die Azure Cognitive Services für die KI-gestützte Übersetzung. Die Backend-Architektur wurde so konzipiert, dass sie eine effiziente Kommunikation zwischen Frontend, API-Gateway und Datenbank ermöglicht. Wichtige Sicherheitsaspekte, insbesondere die Authentifizierung über Azure Entra, wurden ebenfalls abgestimmt.

Im Bereich der Benutzeroberfläche wurde das Design durch Mockups veranschaulicht und in Abstimmung mit der Kundin optimiert. Die Entscheidung, sich am bekannten Layout von Google Translate zu orientieren, soll eine einfache Nutzung ermöglichen. Zudem wurden zentrale Anforderungen, wie die Unterstützung verschiedener Dateiformate und eine API-Schnittstelle für Drittanbieterinnen, spezifiziert.

Die Entwurfsphase hat eine solide Basis für die weitere Entwicklung geschaffen.

## 4 Entwurfsphase 2

### 4.1 Übersetzungs-Ressource

Zum Übersetzen der Fließtexte und Dokumente soll eine Ressource in Azure verwendet werden, welche mittels API erreichbar ist.

#### 4.1.1 Azure Translator [8]

Der Microsoft Translator ist ein mehrsprachiger, cloudbasierter Dienst für maschinelle Übersetzung, der zur Azure AI Services-Familie gehört. Er ermöglicht die Übersetzung von Texten und gesprochenen Inhalten in über 100 Sprachen und wird in verschiedene Microsoft-Produkte wie Bing, Microsoft Office, SharePoint, Microsoft Edge, Skype und andere integriert.



Abbildung 9: Translator Ressource

Für Entwicklerinnen und Unternehmen bietet der Translator eine API, die eine einfache Integration von Übersetzungsfunktionen in Anwendungen ermöglicht. Der Dienst unterstützt sowohl Text- als auch Sprachübersetzungen und bietet Funktionen wie Transliteration, Spracherkennung und die Bereitstellung alternativer Übersetzungen.

Damit man den Microsoft Translator nutzen kann, muss man im Azure-Portal eine Ressource erstellen. Nach der Bereitstellung erhält man Zugriff auf den API-Schlüssel und die Endpunkt-URLs, die für die Authentifizierung und Nutzung des Dienstes gebraucht werden.

Zusätzlich bietet Microsoft die Möglichkeit, benutzerdefinierte Übersetzungsmodelle zu erstellen, um spezifische Anforderungen zu erfüllen. Dies ermöglicht es Unternehmen, Übersetzungen an ihre individuellen Bedürfnisse anzupassen und die Genauigkeit für fachspezifische Inhalte zu erhöhen.

Insgesamt stellt der Microsoft Translator eine umfassende Lösung für die Integration von

Übersetzungsfunktionen in Anwendungen und Diensten dar, unterstützt durch die Skalierbarkeit und Sicherheit der Azure-Plattform.

## 4.2 Datenbank

### 4.2.1 Anforderungen an die Datenbank

- Hauptsächlich werden Selects ausgeführt (Daten holen)
- Kommunikation muss einfach gehen
- Schnelle Beschaffung der Informationen
- Es werden nur ID des Users und das Kürzel der Spracheinstellung gespeichert
- Verwendung von SQL, da uns diese Sprache schon bekannt ist und gut verwendet werden kann.

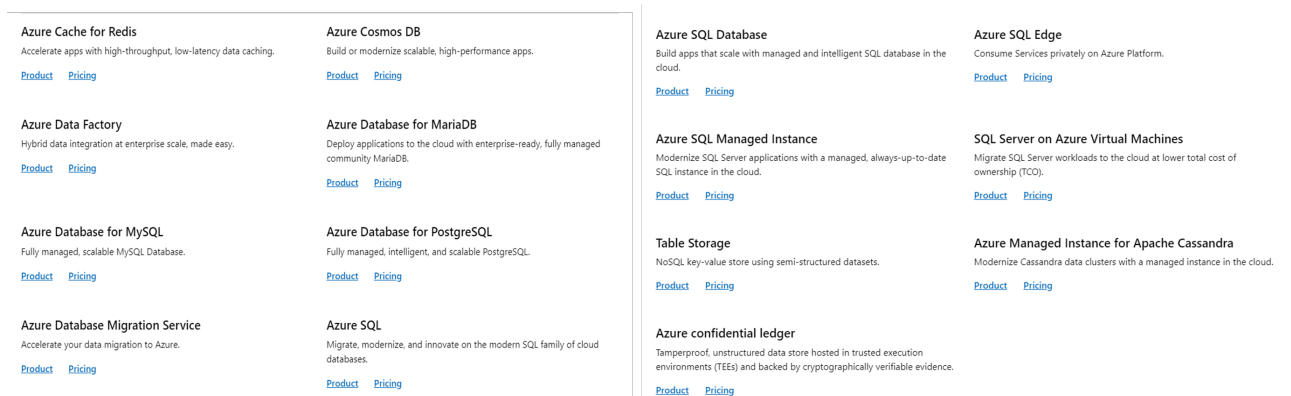


Abbildung 10: Datenbankauswahl in Azure

### Welche DBs kommen in Frage?[9]

- Azure Database for MySQL
- Azure SQL Database

Aufgrund der bereits erworbenen Kenntnisse der Diplomandinnen in der Schule, sind diese beiden Datenbanken in Erwägung gezogen worden. Die Familiarität im Umgang dieser Datenbanken spielte dabei einen wichtigen Faktor, wodurch eine ausgiebige Recherche der beiden Modelle folgte.

### 4.2.2 Gegenüberstellung

#### Azure Database for MySQL:[10]

- Es handelt sich um einen vollständig verwalteten Datenbankdienst für MySQL, der die Open-Source-Version von MySQL verwendet.
- Bietet intelligente Leistungsoptimierungen, integrierte KI, Sicherheits- und Compliance-Funktionen sowie automatisierte Patches und Updates.
- Verfügbar als Single Database, Elastic Pool, Managed Instance und Hyperscale.

#### Typische Anwendungsfälle:

- Ideal für Enterprise Anwendungen, die SQL Server verwenden, wie ERP-, CRM- und LOB Anwendungen.
- Häufig verwendet für Data Warehousing, Business Intelligence und Reporting-Lösungen, insbesondere in Verbindung mit Power BI.
- Entwicklerinnen, die SaaS Lösungen erstellen, profitieren von der Skalierbarkeit und den integrierten Verwaltungsfunktionen.
- Unternehmen, die ihre bestehenden On-Premise SQL Server-Datenbanken in die Cloud migrieren möchten, um von den Vorteilen einer vollverwalteten Plattform zu profitieren.

#### Azure SQL Database[11]

- Es handelt sich um einen vollständig verwalteten Datenbankdienst, der auf der Microsoft SQL Server-Technologie basiert und daher vollständig kompatibel mit SQL-Server ist.
- Bietet intelligente Leistungsoptimierungen, integrierte KI, Sicherheits- und Compliance-Funktionen sowie automatisierte Patches und Updates.
- Verfügbar als Single Database, Elastic Pool, Managed Instance und Hyperscale.

#### Typische Anwendungsfälle:

- Unternehmensanwendungen: Ideal für Enterprise-Anwendungen, die SQL Server verwenden, wie ERP-, CRM- und LOB-Anwendungen.
- Datenanalyse und BI: Häufig verwendet für Data Warehousing, Business Intelligence und Reporting-Lösungen, insbesondere in Verbindung mit Power BI.
- Skalierbare SaaS-Anwendungen: Entwicklerinnen, die SaaS Lösungen erstellen, profitieren von der Skalierbarkeit und den integrierten Verwaltungsfunktionen.

## 4.3 Frontend Framework [1][2]

- Modernisierung von Altsystemen: Unternehmen, die ihre bestehenden On-Premises SQL Server-Datenbanken in die Cloud migrieren möchten, um von den Vorteilen einer vollverwalteten Plattform zu profitieren.

### 4.2.3 Zusammenfassung und Entscheidung

- Azure Database for MySQL ist ideal für Anwendungen, die MySQL als Datenbank bevorzugen, insbesondere für Open-Source-Projekte und Webanwendungen.
- Azure SQL Database eignet sich hervorragend für Unternehmensanwendungen und -lösungen, die auf SQL Server basieren und fortgeschrittene Funktionen für Leistung, Sicherheit und Verwaltung erfordern.

Aufgrund dieser Erkenntnisse entscheiden wir uns für Azure Database for MySQL. Da unsere Anwendung im Web läuft, ist MySQL eine passende Wahl, da es sich gut für Webanwendungen eignet. Außerdem haben wir bereits Erfahrung mit MySQL, was die Entwicklung erleichtert. Die Kombination aus einer vertrauten Datenbank und den Vorteilen einer verwalteten Cloud-Lösung macht diese Option ideal für das Projekt.

## 4.3 Frontend Framework [1][2]

Ein wichtiger Bestandteil von jeder Webanwendung ist das Frontend, das für die Darstellung und Interaktion mit den Nutzerinnen verantwortlich ist. Die Wahl des passenden Frameworks beeinflusst einerseits die Entwicklungsweise andererseits auch die Benutzerfreundlichkeit der Anwendung. In diesem Abschnitt werden die Technologien JSP und JSF vorgestellt, die beide zur Entwicklung von Webanwendungen mit Java genutzt werden. Während beide Ansätze dynamische Inhalte bereitstellen, unterscheiden sie sich in ihrer Architektur und ihrem Anwendungsbereich.

JSP wird hauptsächlich für die serverseitige Generierung von Webseiten mithilfe von Skripten eingesetzt, während JSF ein komponentenbasiertes Framework ist, das eine strukturierte Entwicklung interaktiver Benutzeroberflächen ermöglicht. Im Folgenden werden die Eigenschaften, Vor- und Nachteile dieser Technologien näher erklärt.

### 4.3.1 JSP

- **Technologie:** JSP ist eine serverseitige Scripting-Technologie, die HTML mit eingebettetem Java-Code ermöglicht. JSP-Seiten erhalten durch Servlets eine Funktionalität und werden vom Server verarbeitet.
- **Zweck:** Hauptsächlich verwendet für die dynamische Generierung von HTML-Inhalten. Es eignet sich gut für die Erstellung einfacher, dynamischer Webseiten.
- **Komponenten:** JSP verwendet Standard-HTML und eingebetteten Java-Code (Scriptlets), EL und JSP-Tags.
- **Architektur:** JSP ist eher seitenzentriert. Es gibt weniger Struktur und Unterstützung für das Trennen von Logik und Präsentation.
- **Zustandsverwaltung:** JSP-Seiten sind zustandslos. Zustandsinformationen müssen manuell verwaltet werden (z. B. durch Sessions oder Cookies).
- **Navigation:** Navigation wird typischerweise durch das Verlinken von Seiten oder durch Servlet-Weiterleitungen gehandhabt.
- **Entwicklungsaufwand:** Einfacher und schneller für kleine, weniger komplexe Anwendungen.
- **Flexibilität:** Mehr Kontrolle über den generierten HTML-Code, aber weniger Unterstützung für Wiederverwendbarkeit und Wartbarkeit.

### 4.3.2 JSF

- **Technologie:** JSF ist ein komponentenbasiertes Framework für die Benutzeroberfläche (UI) in Java-Webanwendungen. Es verwendet JSP oder Facelets als Ansichts-Technologien.
- **Zweck:** Entwickelt für die Erstellung komplexer, wiederverwendbarer UI-Komponenten und zur Verwaltung des Anwendungszustands. Es bietet ein vollständiges MVC-Framework.
- **Komponenten:** JSF bietet eine umfangreiche Bibliothek von UI-Komponenten (z.B.: Formulare, Tabellen, Menüs) und ermöglicht die Erstellung eigener Komponenten.
- **Architektur:** JSF folgt dem komponentenbasierten Ansatz und unterstützt MVC. Es gibt eine klare Trennung von Logik (Managed Beans) und Präsentation (Seiten).
- **Zustandsverwaltung:** JSF ist zustandsbehaftet. Es verwaltet automatisch den Zustand von UI-Komponenten und Benutzereingaben zwischen Anfragen.
- **Navigation:** JSF bietet eine deklarative Navigation mit Navigation-Regeln und kann komplexe Anwendungsabläufe einfach handhaben.
- **Entwicklungsaufwand:** Komplexer und möglicherweise mehr initialer Aufwand, aber geeignet für große und komplexe Anwendungen.

## 4.4 Abrechnung

- **Flexibilität:** Höhere Abstraktion und Wiederverwendbarkeit durch Komponenten, jedoch weniger direkte Kontrolle über den generierten HTML-Code.

## 4.4 Abrechnung

Das Projekt ist Teil der Abteilung IT in der Firma ENGEL. Die Abteilung hat vor, diese Software an Mitarbeiterinnen zu geben und dabei eine Abrechnung zu erstellen. Die Abrechnung sollte nach User funktionieren. Also beim SSO sollte der User mit dokumentiert werden und die Anzahl der Zeichen, welche übersetzt wurden, gespeichert werden. Die Abrechnung erfolgt monatlich.

## 4.5 Fazit der Phase

In der zweiten Entwurfsphase wurden wichtige technische Entscheidungen getroffen, die die Grundlage für die weitere Entwicklung der Anwendung bilden. Die Wahl der Datenbank fiel auf Azure Database for MySQL, da sie eine gute Kombination aus Vertrautheit, Leistungsfähigkeit und einfacher Verwaltung bietet. Diese Entscheidung ermöglicht eine effiziente Datenhaltung, insbesondere für die Speicherung der Benutzerinnen-IDs und Spracheinstellungen.

Zudem wurde eine gründliche Gegenüberstellung von JSP und JSF durchgeführt. Während JSP für einfachere Webanwendungen mit dynamischer HTML-Erzeugung geeignet ist, bietet JSF durch seinen komponentenbasierten Ansatz und integrierte Zustandsverwaltung klare Vorteile für komplexere Anwendungen. Die Entscheidung für eine dieser Technologien hängt von den spezifischen Anforderungen der Anwendung ab und ist am Ende der Phase auf JSP gefallen.

Schließlich wurde ein Konzept für die Abrechnung erarbeitet. Hierbei wird das Single-Sign-On genutzt, um Nutzeraktivitäten zu erfassen und die Nutzung des Übersetzungstools basierend auf der Anzahl übersetzter Zeichen monatlich abzurechnen. Dies ermöglicht eine transparente Kostenverteilung und eine gezielte Steuerung der Zugriffsrechte innerhalb des Unternehmens.

# 5 Implementierung

In der dritten Entwurfsphase wurde die technische Umsetzung der Anwendung finalisiert. Dabei wurden sowohl das Backend als auch das Frontend mit den zuvor festgelegten Technologien entwickelt und getestet. Besonderes Augenmerk lag auf der Integration des Azure Translator-Dienstes, der Implementierung einer sicheren Authentifizierung sowie der Gestaltung einer benutzerfreundlichen Oberfläche.

Während der Entwicklung wurden einige zuvor geplante Systemkomponenten erneut hinterfragt und angepasst. Dies betraf insbesondere die Verwendung einer Datenbank zur Speicherung von Spracheinstellungen sowie die Implementierung einer Abrechnungsfunktion für die Nutzungskosten der Übersetzungs-API. Nach sorgfältiger Analyse und Rücksprache mit unserer Projektleiterin sowie Experten aus dem Bereich Microsoft Azure wurden diese beiden Funktionen verworfen, um die Komplexität des Systems zu reduzieren und eine effizientere Umsetzung zu gewährleisten. Die folgenden Abschnitte erläutern die getroffenen Entscheidungen und den Entwicklungsprozess im Detail.

## 5.1 Programmierung

### 5.1.1 Backend [12] [13]

Im Rahmen der Entwicklung des Backends wurde zunächst das gemeinsam erarbeitete Architekturkonzept analysiert. Ziel war die Implementierung einer API, die den Azure Translator-Dienst nutzt, um sowohl Fließtexte als auch Dokumente in eine gewünschte Zielsprache zu übersetzen.

Die Umsetzung begann mit der Einrichtung der Azure Translator-Ressource innerhalb der Microsoft Azure-Cloud. Dabei wurden alle relevanten Zugangsdaten, einschließlich des API-Schlüssels und der geografischen Region, dokumentiert. Um eine erste Verbindung zur Ressource herzustellen und deren Funktionsweise zu validieren, wurden Testanfragen über die Kommandozeile (cmd.exe) mittels curl durchgeführt, basierend auf den offiziellen Microsoft-Dokumentationen.

Für die Entwicklung der API wurde das Java-Framework Spring Boot verwendet, da es eine

schnelle und effiziente Implementierung von RESTful-Webservices ermöglicht. Die Spring-Boot-Anwendung basiert auf einem Controller, der HTTP-Anfragen entgegennimmt und verarbeitet. Dieser Controller stellt zwei Endpunkte bereit:

- Einen Endpunkt zur Übersetzung von **Texten**
- Einen Endpunkt zur Übersetzung von **Dokumenten**

Innerhalb dieser Methoden werden die eingehenden Daten aufbereitet, um einen gültigen API-Request an die Azure Translator-Ressource zu senden. Die Antwort der Übersetzungs-API wird anschließend extrahiert und in einem einheitlichen Format an den Client zurückgegeben.

Ein API-Aufruf an das Backend muss dabei nicht nur den zu übersetzenden Text oder das Dokument enthalten, sondern auch die Ausgangssprache und die Zielsprache. Zwar bietet der Azure Translator die Möglichkeit, die Ausgangssprache automatisch zu erkennen, jedoch unterscheidet sich in diesem Fall die Struktur des Response-Bodys. Um eine konsistente Verarbeitung der API-Antworten zu gewährleisten und unnötige Sonderfälle in der Datenverarbeitung zu vermeiden, wurde festgelegt, dass beide Sprachen in der Anfrage explizit angegeben werden müssen. Dadurch bleibt der Aufbau des Response-Objekts immer gleich, unabhängig von der Sprachkombination.

Zur kontinuierlichen Validierung und Fehleranalyse wurde Postman eingesetzt, um die verschiedenen API-Funktionen während der Entwicklung ausgiebig zu testen. Zusätzlich wurde zur Absicherung des Backends OAuth2-Authentifizierung implementiert. Diese stellt sicher, dass jede eingehende Anfrage über einen gültigen Token verfügt, bevor sie weiterverarbeitet wird. Dadurch wird gewährleistet, dass nur autorisierte Clients Zugriff auf die API erhalten.

### 5.1.2 Frontend

Das Design und die Struktur des Frontends basieren auf den zuvor erstellten Mockups, die in der Entwurfsphase entwickelt wurden. Diese Mockups dienen als visuelle Grundlage für das Layout und die Benutzerinnenführung der Anwendung. Zudem wurde in der Entwurfsphase eine Analyse verschiedener Technologien zur Umsetzung des Frontends durchgeführt, insbesondere ein Vergleich zwischen JSP (JavaServer Pages) und JSF (JavaServer Faces). Basierend auf dieser Analyse fiel die Entscheidung auf JSP, da diese Technologie bereits aus dem Unterricht bekannt war und eine effiziente Umsetzung ermöglichte.

Für die Entwicklung der Weboberfläche wurde WildFly als Applikationsserver eingesetzt,

## 5.1 Programmierung

da er bereits in früheren Projekten verwendet wurde und die Erstellung von JSP-basierten Webseiten erleichterte. Dadurch konnte die Entwicklung direkt beginnen, ohne sich erst in ein neues Framework einarbeiten zu müssen.

Während das Backend noch in der Implementierungsphase war, wurde zunächst die grundlegende Struktur des Frontends entsprechend den Mockups aufgebaut. Das Layout der Webseite ist wie folgt strukturiert:

- Eine Informationsleiste am oberen Rand, die das Firmenlogo (ENGEL) sowie das eigene Projektlogo (ET) enthält.
- Eine vertikale Navigationsleiste auf der linken Seite, mit der zwischen den verschiedenen Unterseiten gewechselt werden kann.

Die Anwendung selbst besteht aus drei Hauptseiten:

- **Startseite:**

Die Startseite enthält allgemeine Informationen zur Anwendung und erläutert deren Zweck – die Übersetzung von Texten und Dokumenten mithilfe von KI. Zudem wird ein Hinweis darauf gegeben, dass die Übersetzungen durch eine künstliche Intelligenz erfolgen und daher nicht immer zu 100% korrekt sein können. Zusätzlich ist auf dieser Seite ein Link zum ENGEL-internen Help-Desk integriert, über den Nutzerinnen bei Fragen oder Problemen Unterstützung anfordern können.

- **Textübersetzung:**

Die Benutzeroberfläche dieser Seite orientiert sich am Layout von Google Translate, da viele Nutzerinnen bereits mit dessen Bedienkonzept vertraut sind. Dadurch wird eine intuitive Verwendung der Anwendung ermöglicht. Die wichtigsten Funktionen dieser Seite sind:

- Ein Dropdown-Menü für die Auswahl der Ausgangssprache sowie eines für die Zielsprache.
- Ein Textfeld zur Eingabe des zu übersetzenden Textes.
- Ein Übersetzen-Button, der die Eingabe an das Backend sendet und den übersetzten Text im darunterliegenden Feld ausgibt.
- Ein Button zum Kopieren der Übersetzung in die Zwischenablage, um das Ergebnis weiterverwenden zu können.

- **Dokumentenübersetzung:**

Der grundlegende Aufbau dieser Seite ähnelt der Textübersetzungsseite, insbesondere hin-

sichtlich der Auswahl der Sprachen über zwei Dropdown-Menüs. Zusätzlich bietet diese Seite:

- Ein Dateiupload-Feld, das Drag-and-Drop sowie eine Durchsuchen-Funktion unterstützt, um ein Dokument zur Übersetzung auszuwählen.
- Eine Beschränkung, die sicherstellt, dass immer nur ein einzelnes Dokument hochgeladen werden kann.
- Eine Infobox, die über unterstützte Dateiformate informiert.
- Einen Übersetzen-Button, der das Dokument an das Backend sendet. Das übersetzte Dokument wird anschließend automatisch als Download im Browser bereitgestellt, unter Beibehaltung des ursprünglichen Formates und Dateityps.

Sobald das Backend fertiggestellt war, wurde die Anbindung des Frontends an die API umgesetzt, um die Übersetzungsfunktionalität vollständig zu integrieren. Dabei wurde sichergestellt, dass API-Anfragen mit einem gültigen Token authentifiziert werden müssen. Da das Backend OAuth2 zur Authentifizierung verwendet, musste das Frontend zunächst einen gültigen Token anfordern, bevor es eine Anfrage an das Backend senden konnte. Die Verwaltung dieses Tokens wird durch ein Authorization-Programm übernommen, das in einem späteren Abschnitt detaillierter beschrieben wird.

Beim Design der Benutzeroberfläche wurde darauf geachtet, dass die Farbgebung der Anwendung mit den Farben des ENGEL-Logos übereinstimmt, um ein einheitliches und professionelles Erscheinungsbild zu gewährleisten.

### 5.1.3 Authorization [14] [15] [16]

Das Authorization-Programm wurde entwickelt, um dem Frontend einen gültigen Token bereitzustellen, der für die Kommunikation mit dem Backend erforderlich ist. Da das Backend durch OAuth2 geschützt ist, müssen alle API-Anfragen mit einer autorisierten Identität erfolgen.

Die Implementierung basiert auf Azure App Registrations innerhalb von Microsoft Entra ID (ehemals Azure Active Directory). Für die Authentifizierung wurde eine App-Registrierung in Entra ID erstellt, über die eine Client-ID und ein Client-Secret generiert wurden. Diese Anmeldeinformationen ermöglichen es dem Authorization-Programm, über eine Client-Credentials-Authentifizierung einen Access-Token abzurufen.

Das Programm selbst stellt eine einfache REST-API mit einem einzigen GET-Endpoint bereit.

Dieser Endpunkt wird vom Frontend aufgerufen, um einen gültigen Token zu erhalten. Die API selbst besitzt keine weiteren Endpunkte oder komplexe Logik – sie übernimmt ausschließlich die Aufgabe, einen Token von Microsoft abzurufen und diesen an das Frontend weiterzugeben.

## 5.2 Datenbank

In einer früheren Entwurfsphase wurde detailliert dargelegt, welche Datenbanktechnologie für das Projekt am besten geeignet wäre und welche Kriterien zur Auswahl führten. Nach einer erneuten Evaluierung der Anforderungen und einer Abwägung der Implementierungskosten wurde jedoch entschieden, keine Datenbank in das System zu integrieren.

Der ursprüngliche Hauptgrund für die Einbindung einer Datenbank war die Speicherung von Spracheinstellungen der Nutzerinnen. Nach eingehender Analyse wurde jedoch festgestellt, dass eine persistente Speicherung dieser Daten nicht zwingend erforderlich ist. Stattdessen wird die Spracheinstellung für die jeweilige Sitzung festgelegt und standardmäßig auf Englisch gesetzt. Diese Entscheidung reduziert die Systemkomplexität und vermeidet den Wartungsaufwand, der mit einer Datenbankverwaltung einhergeht. Da keine weiteren Anwendungsfälle für eine Datenbank identifiziert wurden, wurde sie letztendlich aus dem Systemdesign entfernt.

## 5.3 Abrechnung

Ein weiterer ursprünglich geplanter Aspekt war die Implementierung einer Abrechnungsfunktion zur Erfassung und Abgrenzung der Nutzungskosten für die Translator-Ressource in Microsoft Azure. Im Rahmen eines Meetings mit unserer Projektleiterin und einem Experten aus dem Bereich Microsoft Azure, von der Firma ENGEL, wurde die Möglichkeit einer differenzierten Abrechnung nach Benutzerinnen untersucht. Dabei stellte sich heraus, dass eine eindeutige und nachvollziehbare Zuweisung der Nutzungskosten nicht mit der gewünschten Genauigkeit umsetzbar ist.

Auf Grundlage dieser Erkenntnisse entschied die Projektleiterin, dass eine solche Abrechnung für die Diplomarbeit nicht erforderlich sei. Folglich wurde diese Funktion nicht weiter verfolgt und nicht in das System implementiert. Diese Entscheidung ermöglichte eine effizientere Ressourcenverwendung und reduzierte den Implementierungsaufwand, ohne die Kernfunktionalität des Systems zu beeinträchtigen.

## 5.4 Fazit

Im Rahmen dieser Implementierung wurde die Applikation erfolgreich umgesetzt und ausführlich getestet. Zur Validierung der API wurden Tools wie Postman und die Kommandozeile (cmd.exe) genutzt, um die Funktionsweise der Schnittstellen zu überprüfen und mögliche Fehler frühzeitig zu identifizieren. Das Zusammenspiel zwischen Backend und Frontend wurde iterativ verbessert, sodass eine stabile und benutzerfreundliche Anwendung entstanden ist.

Insgesamt konnte die Applikation erfolgreich realisiert werden. Sie bietet eine zuverlässige Möglichkeit zur maschinellen Übersetzung von Texten und Dokumenten und erfüllt die Anforderungen des Projekts.

# 6 Deployment

## 6.1 Azure

Das Deployment der Webanwendung auf Microsoft Azure stellte eine bedeutende Herausforderung dar, insbesondere aufgrund der Vielzahl von Technologien, die für die verschiedenen Komponenten der Anwendung verwendet werden. Die Webanwendung wurde in mehrere Komponenten unterteilt, die jeweils separat auf Microsoft Azure bereitgestellt werden sollten. Der Deployment-Prozess wurde mit der Azure CLI durchgeführt, wobei verschiedene Ansätze für das Backend, die Authentifizierungsanwendung sowie den Einsatz von Docker und JBoss EAP evaluiert wurden.

## 6.2 Deployment des Backends [3] [4]

Das Deployment des Backends erfolgte in Azure App Service. Dazu wurde im Azure-Portal ein entsprechender Dienst erstellt. Nach dem Build-Prozess, bei dem eine ausführbare .jar-Datei generiert wurde, wurde diese mit der Azure CLI deployed. Laut den Ausgaben im Terminal verlief der Deployment-Prozess erfolgreich.

Nach dem erfolgreichen Deployment wurden beide Translation-Endpoints mit Postman getestet. Während die Dokumentübersetzung korrekt funktionierte und das übersetzte Dokument zurückgegeben wurde, führte der Aufruf der Textübersetzung zu einem Serverfehler (500 HTTP Error). Um die Ursache einzugrenzen, wurde der Code lokal ausgeführt, wo beide Endpunkte wie erwartet funktionierten. Da die Dokumentenübersetzung in Azure erfolgreich war, konnte ein Problem mit dem benötigten Token ausgeschlossen werden, da dieser für beide Endpunkte gleich ist. Es liegt somit nahe, dass die Fehlerquelle im Code oder in der Konfiguration innerhalb der Azure-Umgebung zu finden ist.

## 6.3 Deployment der Authorization-Applikation

Für die Wildfly-basierte Authorization-Applikation wurde zunächst dieselbe Methode wie beim Backend angewendet. Das Deployment wurde in der Azure CLI erfolgreich durchgeführt, jedoch lieferte der Endpoint bei Tests mit Postman lediglich eine HTML-Informationseite zurück. Um die Funktionsfähigkeit des Codes sicherzustellen, wurde die Anwendung lokal gestartet und erneut mit Postman getestet, wobei die erwartete Antwort – ein Token – zurückgegeben wurde. Dies bestätigte, dass die Applikation grundsätzlich korrekt arbeitet, jedoch in der Azure-Umgebung nicht wie gewünscht ausgeführt wurde.

### 6.3.1 Deployment als Wildfly-Server

Um eine mögliche Ursache in fehlenden Wildfly-Laufzeitkomponenten in Azure auszuschließen, wurde die Anwendung als vollständiger Wildfly-Server gepackt und in dieser Form deployed. Dies sollte sicherstellen, dass alle notwendigen Komponenten innerhalb des Deployments enthalten sind. Das Deployment verlief erfolgreich, jedoch wurde beim erneuten Test mit Postman weiterhin nur die HTML-Seite zurückgegeben. Dies deutet darauf hin, dass entweder die Konfiguration von Wildfly oder die Art, wie Azure mit Wildfly umgeht, nicht kompatibel ist.

### 6.3.2 Deployment mit Docker [17]

Ein weiterer Ansatz bestand darin, die Applikation als Docker-Container zu deployen. Dazu wurde ein bereits existierendes Wildfly-Image von quay.io verwendet. Nach dem Build-Prozess und der Erstellung eines entsprechenden App Services in Azure wurde das Deployment erneut ausgeführt. Hierbei trat jedoch eine Fehlermeldung auf, dass .war-Dateien nicht direkt in den App Service deployed werden können. Dies machte eine alternative Methode erforderlich, die eine andere Umgebung für das Deployment vorsieht.

### 6.3.3 Deployment mit JBoss EAP

Bei einem Gespräch mit Professor Otto kam noch die Idee auf, das Deployment mit JBoss EAP zu versuchen. Das KI Tool ChatGPT schlug vor, JBoss EAP als Laufzeitumgebung in Azure App Service auszuwählen. Diese Option war jedoch in der Azure-Umgebung nicht verfügbar, weshalb dieser Ansatz nicht weiterverfolgt werden konnte. Da JBoss EAP für die Ausführung von Jakarta EE-Anwendungen optimiert ist, hätte dies eine mögliche Lösung dargestellt. Allerdings wurde dieser Weg aufgrund der fehlenden Unterstützung in Azure nicht weiterverfolgt.

## 6.4 Fazit

Zusammenfassend zeigten die Tests, dass das Backend grundsätzlich erfolgreich deployed werden konnte, jedoch noch Fehler bei einzelnen Endpunkten bestehen.

Die Wildfly-Applikationen erwiesen sich als schwieriger zu deployen, da Azure standardmäßig keine direkte Unterstützung für Wildfly bietet. Während das Deployment technisch (teilweise) erfolgreich war, funktionierte keiner der Endpunkte wie gewünscht.

Die durchgeführten Tests zeigten, dass das Deployment lokal einwandfrei funktionierte, jedoch in Azure nicht das erwartete Verhalten zeigte. Die Ursachen hierfür können in der Laufzeitumgebung, der Konfiguration oder der Art des Deployments liegen.

# 7 Machbarkeitsstudie Glossar

## 7.1 Einleitung

ENGEL stellte uns ein Glossar zur Verfügung, welches deutsche und englische Übersetzungen für Fachbegriffe und Anweisungen enthält. Nach einigen Testungen und Recherche sind wir jedoch zu dem Entschluss gekommen, dass es nicht möglich ist, das bestehende ENGEL Glossar einzubinden.

Beim Verwenden des Glossars wird einfach eine CSV-Datei beim Übersetzen mitgeschickt. Die Microsoft Azure Translator Resource verlangt dafür aber eine mit ',' getrennte Datei. In der CSV-Datei befinden sich aber ebenfalls Sätze mit Beistrichen als Satzzeichen, welche dann genauso als Trennzeichen zwischen dem Deutschen und Englischen Worten angesehen werden. Deswegen kann das Glossar nicht verwendet werden.

## 7.2 Fragestellungen und Überlegungen

Mit einer kleinen Test CSV-Datei ohne Sätze wurden nachfolgende Fragestellungen erarbeitet.

### Glossar Test Fragestellungen:

- Wird auch richtig übersetzt, wenn das Wort falsch geschrieben wurde?

Zum Testen, ob auf auch menschliche Fehler eingegangen wird.

- Wird auch richtig übersetzt, wenn das Wort vor oder nach einem Beistrich steht?

Zum Testen, ob Satzzeichen die Erkennung des Wortes beeinflusst oder nicht.

- Wird auch richtig übersetzt, wenn das Wort neben Satzzeichen steht wie „?, !, .,“ usw. steht?

Zum Testen, ob es Unterschiede gibt, wenn das Wort am Ende des Satzes neben einem Satzzeichen steht anstatt mitten im Satz wie bei einem Beistrich.

- Wird auch richtig übersetzt, wenn das Wort nachfolgende oder führende Leerzeichen hat?

## 7.3 Ergebnis

Zum Erkennen, ob das Erkennen des Wortes durch Leerzeichen beeinflusst wird.

- Wird auch richtig übersetzt, wenn das Wort leicht verändert ist? zB. Auto, Autos, AUTO, auto...

Zum Prüfen, ob die Groß-Klein-Schreibung berücksichtigt wird bzw. ob es die Mehrzahl eines Wortes als eigenes oder anderes Wort identifiziert und nicht nach Glossar übersetzt.

- Wird auch richtig übersetzt, wenn das Wort am Satzanfang steht und somit großgeschrieben wird?

## 7.3 Ergebnis

**Das Dokument welches zum Übersetzten mit dem Glossar mitgeschickt wurde enthielt folgende Sätze:**

- Das Auto fährt sehr schnell.

Um zu überprüfen, ob das Wort in einem normalen unveränderten Satz richtig übersetzt wird.

- Autos werden in der Werkstatt repariert.

Um zu überprüfen, ob das Wort übersetzt wird, wenn es am Satzanfang steht und anders geschrieben wird.

- Ich finde Autos, Fahrräder und Boote cool.

Um zu überprüfen, ob das Wort richtig ist, wenn Satzzeichen im Satz vorkommen und es anders geschrieben wird.

- Darf ich auto fahren?

Um zu überprüfen, ob das Wort richtig ist, wenn es im Satz klein geschrieben wird.

- Das Kind schreit: „Auto“!

Um zu überprüfen, ob das Wort übersetzt wird, wenn es unter Hochkommas steht.

- Wird das Auto gerade von jemandem gefahren?

Um zu überprüfen, ob das Wort übersetzt wird, wenn Leerzeichen davor sind.

### 7.3 Ergebnis

- Auto?

Um zu überprüfen, ob das Wort richtig ist, wenn es mit einem Satzzeichen kombiniert wird.

- Autos?

Um zu überprüfen, ob das Wort richtig ist, wenn es mit einem Satzzeichen kombiniert wird und ein Buchstabe hinzugefügt wurde.

- Autokorrektur

Um zu überprüfen, ob das Wort richtig ist, wenn es ein zusammengesetztes Wort ist.

- autokorrektur

Um zu überprüfen, ob das Wort richtig ist, wenn es ein zusammengesetztes Wort ist und klein geschrieben wurde.

- autobahn

Um zu überprüfen, ob das Wort richtig ist, wenn es am Satzanfang steht und ein zusammengesetztes Wort ist und klein geschrieben wurde.

- Autobahn

Um zu überprüfen, ob das Wort richtig ist, wenn es am Satzanfang steht und ein zusammengesetztes Wort ist.

- Audo auto autu Autu Aoto Aauto Autoo

Um zu überprüfen, ob das Wort richtig ist, wenn das Wort in jeglicher Art falsch geschrieben wurde.

- Darf ich Audo fahren?

Um zu überprüfen, ob das Wort richtig ist, wenn ein Buchstabe falsch geschrieben wurde.

- Ich darfAuto fahren.

Um zu überprüfen, ob das Wort richtig ist, wenn es mit dem vorgehenden Wort zusammen geschrieben wurde.

- AUTO

Um zu überprüfen, ob das Wort richtig ist, wenn es in Großbuchstaben geschrieben wurde.

### 7.3.1 Vergleich Ausgangslage vs Übersetzung

In der nachstehenden Grafik ist das verwendete Glossar dargestellt, welches in weiterer Folge zur Übersetzung verwendet wurde.

Deutsch,English  
 Auto,carre  
 gefahren,drivee  
 fahren,drovee  
 schreien,screamm

Abbildung 11: CSV-Test-Glossar

Hier sind die Ergebnisse der Übersetzung, mithilfe des Glossars dargestellt.

Deutsch - Ausgangslage	Englisch - Übersetzung
Das Auto fährt sehr schnell.	The car drives very fast.
Autos werden in der Werkstatt repariert.	Cars are repaired in the workshop.
Ich finde Autos, Fahrräder und Boote cool.	I think cars, bicycles and boats are cool.
Darf ich auto fahren?	Can I drove?
Das Kind schreit: „Auto“!	The child screams: "Carre"!
Wird das Auto gerade von jemandem gefahren?	Is the car being driven by someone right now?
Auto?	Car?
Autos?	Automobiles?
Autokorrektur	Autocorrect
autokorrektur	Autocorrect
autobahn	motorway
Autobahn	Motorway
Audo auto autu Autu Aoto Aauto Autoo	Audo auto autu Autu Aoto Aauto Autoo
Darf ich Audo fahren?	Can I drove Audo?
Ich darfAuto fahren.	I am allowed to drove a car.
AUTO	CAR

Tabelle 2: Vergleich Glossar - Ausgangslage und Übersetzung

## 7.4 Zusammenfassung

Das Glossar hat nur in einem Fall richtig funktioniert, das ist, wenn man das Wort unter Hochkomma schreibt. Wenn das Wort in Großbuchstaben geschrieben ist, wird es nicht übersetzt. Ebenso nicht, wenn es falsch geschrieben ist und wenn es unabsichtlich mit dem vorherigen Wort zusammengeschrieben ist.

Manchmal wird das Wort übernommen aber nur zur Hälfte richtig übersetzt. Wenn das Wort in einem anderen enthalten ist, wird dieses nicht richtig übersetzt. Aufgrund der oben genannten Analyse und aufgrund der Beschaffenheiten des ENGEL Glossars, kann das ENGEL - Glossar nicht zum Übersetzen von Fachbegriffen verwendet werden. Schlussfolgernd kann man sagen, wenn man ein Glossar einbinden möchte, dann sollte man nur Übersetzungen für einzelne Worte und keine ganzen Sätze in eine CSV-Datei schreiben. Hilfreich bei der Übersetzung ist ebenfalls ein einwandfreies Ausgangsdokument um Folgefehler zu vermeiden.

## 7.5 Fazit

Wenn das Wort unter Hochkommas geschrieben wird, beispielsweise das Wort „Auto“ in der obigen Tabelle so wird es richtig übersetzt zu „Carre“, dies kann man auch in dem Dokument „Machbarkeitsstudie Glossar“ im Anhang nachlesen (siehe grün markiertes Wort). Weiters werden Wörter zu einem gewissen Teil richtig übersetzt, das merkt man am Wort „drove“ (siehe obige Tabelle). Im Satz „Darf ich Audo fahren?“ wurde das Wort „Auto“ nicht erkannt und überhaupt nicht übersetzt und das Wort „fahren“ wurde nur zum Teil richtig übersetzt. Das Wort „fahren“ wurde mit dem Wort „drove“ übersetzt, was im Grammatikalischen nicht stimmt sowie nicht der Übersetzung laut Glossar entspricht.

# 8 Ergebnis Produkt

## 8.1 Technologien

### 8.1.1 Programmiersprache

#### Java [18]

Java ist eine der am weitesten verbreiteten objektorientierten Programmiersprachen. Entwickelt wurde sie von James Gosling, welcher das Prinzip „Write once, run anywhere“ anwendete. Dieses Prinzip heißt, dass Java-Code auf allen Plattformen ausführbar ist. Natürlich nur, wenn diese Java unterstützen. Die Java-Technologien bestehen aus mehreren Komponenten. Zu diesen Komponenten zählen die JVM, also die Laufzeitumgebung, in welcher der Java-Bytecode ausgeführt wird. Zuständig für das Ausführen von Java-Anwendungen ist das JRE welches die JVM und viele Standardbibliotheken beinhaltet. Für das Erstellen von Java-Anwendungen ist dann noch das JDK wichtig, welches Werkzeuge wie zum Beispiel den Compiler umfasst. Beim Compilieren wird der JIT verwendet.



Abbildung 12: Java Logo

### 8.1.2 Verwendete Software

#### Microsoft Azure [19]

Microsoft Azure ist eine Cloud-Plattform, die eine Vielzahl an Diensten bereitstellt und Entwicklerinnen, ebenso wie Unternehmen flexible Lösungen für verschiedenste Anforderungen bietet. Zu Leistungen in der Infrastruktur zählen zum Beispiel virtuelle Server, Speicherlösungen und unterschiedliche Netzwerke, welche eine sichere und dazu noch skalierbare Umgebung für Anwendungen schaffen. Weiters bietet Microsoft Azure Datenbankdienste für alle Arten von Datenbanken an. Von relational bis zu NoSQL



Abbildung 13: Azure Logo

ist eine schnelle und effiziente Speicherung von Daten möglich. Dazu kommen dann noch etwaige Analyse-Tools und KI-Funktionen, welche es vereinfachen, große Datenmengen zu verarbeiten und auszuwerten. Microsoft Azure bietet im Generellen eine breite Palette an Software und Werkzeugen von und für Entwicklerinnen, die die Dienste einfach und schnell in der Cloud zur Verfügung gestellt bekommen.

### IntelliJ IDEA [20]

IntelliJ IDEA wurde von JetBrains entwickelt und ist eine Entwicklungsumgebung (IDE) für Java und noch viele weitere Programmiersprachen. IntelliJ bietet den Entwicklerinnen einige Funktionen, wie zum Beispiel Codevervollständigung, Codeanalyse und noch viele weitere Werkzeuge speziell für die Entwicklung. Dieses Programm ist besonders für die Entwicklung moderner Softwareprojekte geschaffen, welche durch Build-Tools, Frameworks und guten Versionskontrollsystemen unterstützt werden.

Weiters werden Technologien wie zum Beispiel Spring, Hibernate, Maven und Gradle von der Entwicklungsumgebung unterstützt. Außerdem bietet die IDE anpassbare Workflows, welche mit Funktionen, wie automatisches Refactoring, Debugging-Werkzeuge und starken Suchfunktionen optimiert werden.

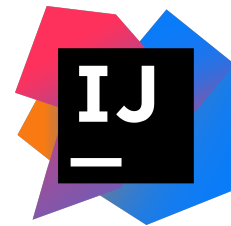


Abbildung 14: IntelliJ Logo

### Postman [21]

Postman ist ein beliebtes Tool zum Testen von APIs. Es hilft Entwicklerinnen, HTTP-Anfragen zu erstellen, zu senden und die Antworten zu analysieren – ohne dafür extra Code schreiben zu müssen.

Mit Postman kann man verschiedene API-Methoden wie GET, POST, PUT und DELETE einfach ausprobieren. Man gibt die URL ein, fügt Parameter oder einen Body hinzu und sieht sofort die Antwort vom Server. Besonders praktisch ist, dass man Anfragen speichern, organisieren und automatisieren kann.



Abbildung 15: Postman Logo

Für die Entwicklung mit Java ist Postman nützlich, um REST-APIs zu testen, die zum Beispiel

mit Spring Boot erstellt wurden. Man kann API-Endpoints prüfen, die Authentifizierung testen (z. B. mit JWT oder OAuth) und sogar automatische Tests für APIs schreiben.

### Spring Boot [22]

Spring Boot ist ein Framework, das die Entwicklung von Java-Anwendungen deutlich vereinfacht. Es basiert auf dem Spring-Framework und ermöglicht die schnelle Erstellung von Web- und Unternehmensanwendungen, ohne dass Entwicklerinnen viele Konfigurationen vornehmen müssen.



Abbildung 16: Springboot Logo

Ein großer Vorteil von Spring Boot ist, dass es viele gängige Funktionen bereits integriert, wie zum Beispiel eine eingebaute Webserver-Unterstützung, Datenbankanbindungen mit Spring Data und eine einfache REST-API-Entwicklung mit Spring Web. Durch automatische Konfiguration und vordefinierte Einstellungen kann man direkt mit der Entwicklung starten, ohne sich um komplexe Setups kümmern zu müssen.

Spring Boot ist besonders beliebt für Microservices, da es leichtgewichtig ist und gut mit Containern wie Docker sowie Plattformen wie Kubernetes zusammenarbeitet.

Mit seiner einfachen Handhabung, flexiblen Struktur und starken Integration in das Java-Ökosystem ist Spring Boot eine ideale Wahl für Entwicklerinnen, die schnelle, skalierbare und moderne Java-Anwendungen erstellen möchten.

### WildFly [23]

WildFly ist ein leistungsstarker Server für Java-Programme. Er wurde als Nachfolger von JBoss entwickelt und dient als Umgebung für moderne Unternehmensanwendungen. WildFly ist schnell, flexibel und kann gut in der Cloud genutzt werden.



Abbildung 17: Wildfly Logo

Der Server unterstützt Jakarta EE (früher Java EE) vollständig

und bietet wichtige Technologien wie JPA für Datenbanken, CDI für einfache Abhängigkeiten und JAX-RS für Webservices. Außerdem arbeitet WildFly gut mit beliebten Entwicklertools wie Maven, Gradle, IntelliJ IDEA und Eclipse zusammen.

Ein besonderes Merkmal von WildFly ist die einfache Konfiguration. Man kann ihn über eine Web-Oberfläche, eine Befehlszeile oder XML-Dateien anpassen. Dank Unterstützung für MicroProfile-APIs eignet er sich auch gut für Cloud-Anwendungen, wie zum Beispiel mit Kubernetes oder OpenShift.

## 8.2 Funktionalität

Die Webanwendung besteht aus zwei zentralen Komponenten: dem Backend, das die gesamte Logik und Verarbeitung der Übersetzungen übernimmt, und dem Frontend, das für eine intuitive Benutzerinnenerfahrung sorgt. Außerdem ist eine dritte Komponente, das Authorization Programm, für die Authentifikation vom Frontend im Backend verantwortlich.

### 8.2.1 Backend

Im Backend laufen alle Prozesse ab, die für die eigentliche Übersetzung notwendig sind. Sobald eine Benutzerin einen Text oder ein Dokument zur Übersetzung einreicht, empfängt das Backend die Anfrage und verarbeitet sie entsprechend.

Bei der Textübersetzung nimmt das System den eingegebenen Text sowie die gewählten Sprachen aus dem Request entgegen und leitet die Anfrage an den Übersetzungsdienst weiter. Sobald die Übersetzung zurückkommt, wird sie direkt an das Frontend zurückgegeben. Für die Dokumentenübersetzung sieht der Ablauf ähnlich aus: Das Backend nimmt das hochgeladene Dokument entgegen und sendet es mit den erforderlichen Informationen an die Translator Ressource weiter. Diese prüft, ob das Dateiformat erlaubt ist (diese Informationen sind direkt im Durchsuchen-Feld auf der Webseite ersichtlich).

Nach der Verarbeitung wird das übersetzte Dokument zurückgegeben, wobei die Ressource in Azure sicherstellt, dass Formatierungen sowie der ursprüngliche Dateityp vollständig erhalten bleiben. Außerdem stellt sie sicher, dass alle unterstützten Sprachen – Deutsch, Englisch, Tschechisch, Chinesisch, Koreanisch und Indisch – sowohl für Text- als auch für Dokumentenübersetzungen verfügbar sind.

### 8.2.2 Frontend

Das Frontend bildet die Schnittstelle zur Benutzerin und macht die Anwendung einfach und effizient nutzbar. Die wesentlichen Seiten sind:



Abbildung 18: Startseite (Deutsch)

Die Startseite ist klar strukturiert und enthält das ENGEL-Logo sowie einen Hinweis, dass die Übersetzungen maschinell erstellt werden und daher nicht zu 100% fehlerfrei sind. Zusätzlich gibt es einen Link, über den direkt eine E-Mail an den ENGEL-Helpdesk geöffnet werden kann. Eine Sprachumschaltung befindet sich oben rechts und ermöglicht das Umschalten zwischen Deutsch und Englisch über eine kleine Flaggen-Schaltfläche.

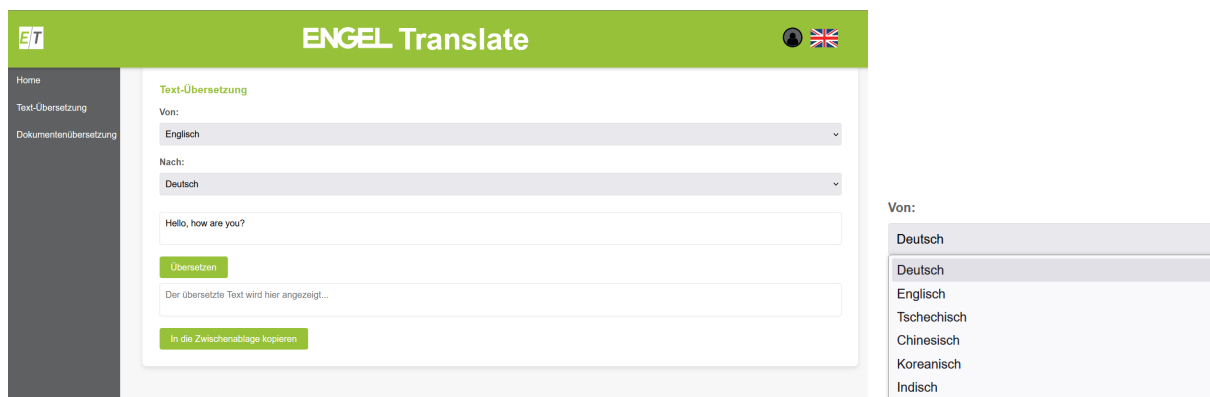


Abbildung 19: Textübersetzung (Deutsch) - Sprachauswahldialog

Für die Textübersetzung steht ein zentrales Eingabefeld zur Verfügung, in das die Benutzerin ihren Text eingeben kann. Die Eingabe- und Zielsprache lassen sich über zwei Dropdown-Menüs

## 8.2 Funktionalität

auswählen. Nach dem Absenden wird der übersetzte Text direkt auf der Seite angezeigt. Falls die Benutzerin den übersetzten Text weiterverwenden möchte, kann sie ihn mit einem Klick auf einen Button in die Zwischenablage kopieren, woraufhin ein kleines Info-Pop-up erscheint, das die erfolgreiche Aktion bestätigt.



Abbildung 20: Dokumentenübersetzung (Deutsch)

Die Dokumentenübersetzung ist ebenso einfach gehalten. Ein Dokument kann entweder per Drag-and-Drop oder über die Durchsuchen-Funktion hochgeladen werden. Die Auswahl der Sprachen erfolgt auch hier über die Dropdown-Menüs. Nach dem Absenden wird die Anfrage an das Backend weitergeleitet, und sobald die Übersetzung abgeschlossen ist, gibt das Frontend das Dokument direkt zum Download frei. Dabei bleibt die gesamte Formatierung erhalten, sodass das übersetzte Dokument optisch und strukturell gleich zum Original bleibt.

Dank dieser Struktur bietet die Webanwendung eine sichere, schnelle und benutzerfreundliche Lösung für die Übersetzung von Texten und Dokumenten, ohne dass externe Tools verwendet werden müssen.

# 9 Zusätzliche Ergebnisse

Ergänzend zur Webanwendung wurde ein individuelles Logo entwickelt, das die visuelle Identität des Projekts unterstützt. Darüber hinaus wurde ein Werbevideo produziert, das die Funktionsweise der Anwendung veranschaulicht und deren Nutzen vermittelt.

## 9.1 Technologien

Mithilfe dieser Technologien konnte die Umsetzung eines Logos und eines Werbevideos realisiert werden.

### 9.1.1 Adobe Illustrator [24]

Adobe Illustrator ist ein vektorbasiertes Grafikdesign-Programm von Adobe, das sich besonders für die Erstellung von Logos, Illustrationen und anderen grafischen Elementen eignet. Es bietet eine Vielzahl an Werkzeugen zur präzisen Bearbeitung von Vektorgrafiken, darunter Pfadwerkzeuge, Typografie-Funktionen und Farbverwaltungssysteme. Ein großer Vorteil von Illustrator ist die verlustfreie Skalierbarkeit der erstellten Designs, da Vektorgrafiken unabhängig von der Auflösung sind. Dadurch können Logos und andere Gestaltungselemente in verschiedenen Größen ohne Qualitätsverlust verwendet werden.



Abbildung 21: Illustrator Logo

Für die Gestaltung unseres Logos haben wir Adobe Illustrator genutzt, da wir bereits im Unterricht mit dem Programm gearbeitet haben und mit dessen Funktionen vertraut waren. Dies ermöglichte uns eine effiziente und präzise Umsetzung unseres Designs.

### 9.1.2 Adobe Photoshop [25]

Adobe Photoshop ist ein pixelbasiertes Bildbearbeitungsprogramm von Adobe, das hauptsächlich zur Bearbeitung und Optimierung von Fotos sowie zur Erstellung digitaler Grafiken genutzt wird. Es bietet eine Vielzahl an Werkzeugen für Retusche, Farbkorrekturen, Ebenenbearbeitung und Bildkomposition. Durch Funktionen wie Masken, Filter und Effekte können Bilder professionell angepasst und kreativ gestaltet werden. Besonders für die Bearbeitung und Kombination verschiedener Bildelemente ist Photoshop ein leistungsstarkes Tool.



Abbildung 22: Photoshop Logo

Im Rahmen unseres Projekts haben wir Adobe Photoshop verwendet, um grafische Elemente für unser Werbevideo zu erstellen und zu bearbeiten. Da wir bereits im Unterricht mit dem Programm gearbeitet hatten, konnten wir effizient damit umgehen und unsere Designs gezielt umsetzen.

### 9.1.3 Adobe Premiere Pro [26]

Adobe Premiere Pro ist ein professionelles Videobearbeitungsprogramm von Adobe, das für den Schnitt, die Nachbearbeitung und die Optimierung von Videomaterial eingesetzt wird. Es bietet zahlreiche Werkzeuge für den präzisen Videoschnitt, Farbkorrekturen, Effekte sowie die Bearbeitung von Tonspuren. Durch die Unterstützung verschiedener Dateiformate und die Möglichkeit zur nahtlosen Integration mit anderen Adobe-Programmen eignet sich Premiere Pro besonders für die Erstellung hochwertiger Video-Projekte.



Abbildung 23: Premiere Pro Logo

Für die Bearbeitung unseres Werbevideos haben wir Adobe Premiere Pro genutzt, um das aufgezeichnete Material zu schneiden und zu optimieren. Da wir im Rahmen eines schulischen Freigegegenstands bereits Erfahrung mit dem Programm gesammelt hatten, konnten wir den Videoschnitt effizient und zielgerichtet umsetzen.

## 9.2 Inhalt

In unserer Diplomarbeit haben wir ein Werbevideo entwickelt, das die Funktionalität des Tools „ENGEL Translate“ auf anschauliche Weise demonstriert.

### 9.2.1 Idee

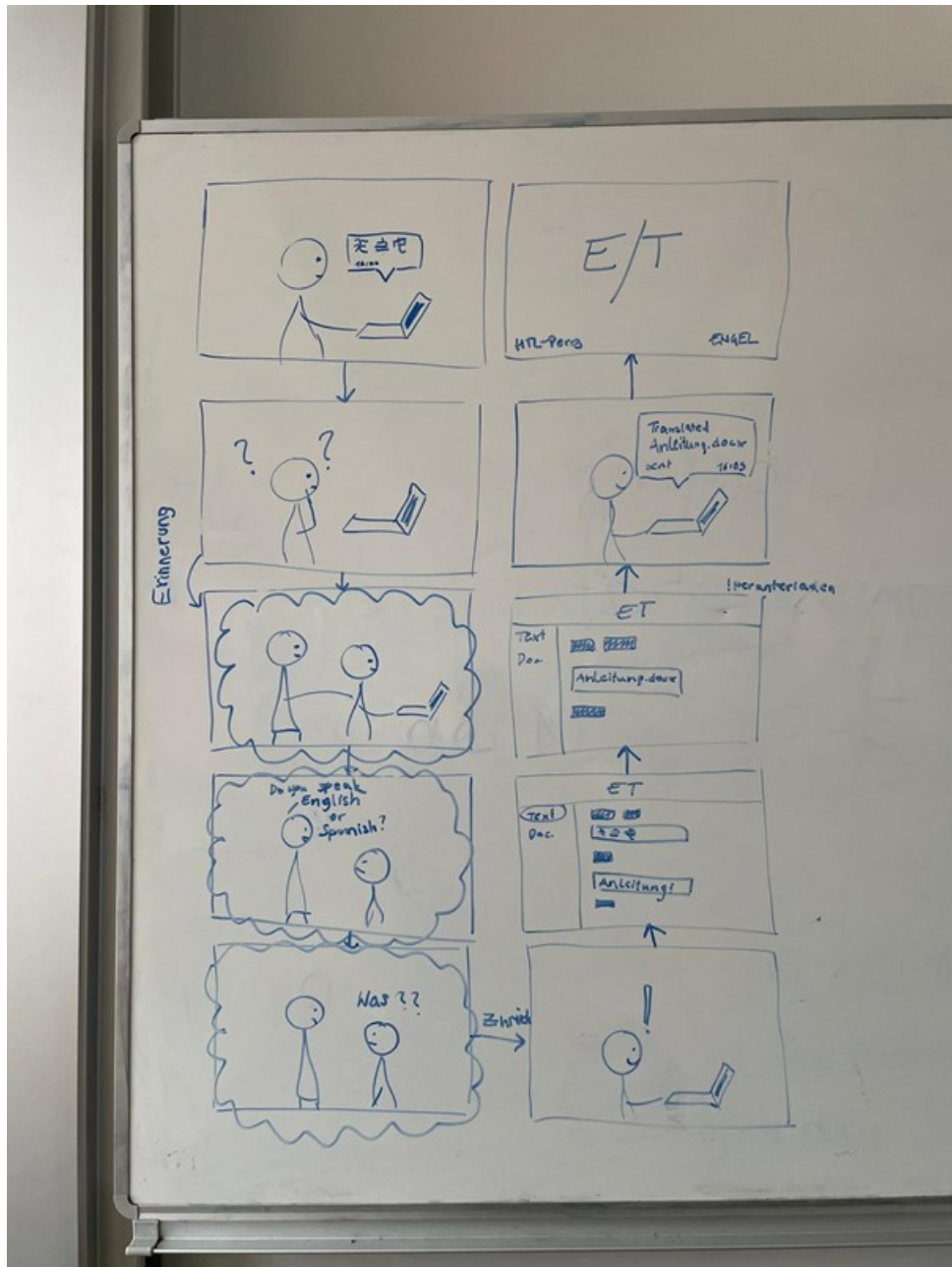


Abbildung 24: Storyboard

In diesem Szenario wird die Anwendung des ENGEL Translate Tools in einem beruflichen Kontext dargestellt, um eine Sprachbarriere zu überwinden. Die Handlung beginnt, als Diana eine Nachricht von Person A erhält, die sie nicht versteht, da die Nachricht auf Chinesisch

verfasst ist und Diana diese Sprache nicht spricht. Verwirrt darüber, was die Nachricht bedeutet, denkt Diana an ein früheres Erlebnis zurück, das ihr nun eine Lösung bietet.

Das frühere Szenario wird eingeblendet. Person B sitzt an ihrem Schreibtisch und arbeitet konzentriert an einer Aufgabe, als Diana zu ihr hinget und sie an der Schulter antippt, um mit ihr zu sprechen. Doch Diana stellt fest, dass Person B die von ihr gesprochene Sprache nicht versteht. In einem Moment der Unsicherheit fragt Diana höflich: „English or Spanish?“ Die Frage lässt Person B kurz stocken, und ein kleines Ladezeichen erscheint über Person B, um den Moment der Verwirrung darzustellen. Nach dieser kurzen Pause fällt Person B eine Lösung ein. Sie öffnet unser Tool und kann sich so mit Diana unterhalten. Dies ist der Moment, in dem die Lösung für Sprachprobleme durch das Tool sichtbar wird.

Zurück in der Gegenwart, hat Diana nun eine konkrete Lösung für das Problem, das sie gerade erlebt. Die Erinnerung an das vergangene Szenario führt sie dazu, die Webseite von ENGEL Translate aufzurufen. Diana beginnt die Nachricht von Person A, die eine Betriebsanleitung auf Chinesisch benötigt, zu übersetzen. Sie öffnet die Betriebsanleitung in englischer Sprache und verwendet das ENGEL Translate Tool, um diese in die gewünschte Sprache, Chinesisch, zu übersetzen. Nachdem die Übersetzung abgeschlossen ist, schickt sie die chinesische Version der Betriebsanleitung an Person A.

Zusätzlich sendet Diana eine Nachricht an Person A, in der sie den Link zur ENGEL Translate Webseite teilt. So kann Person A in Zukunft selbstständig das Tool verwenden, um weitere Übersetzungen vorzunehmen, ohne auf Dianas Hilfe angewiesen zu sein.

In der finalen Szene wird das ENGEL Translate-Logo eingeblendet, das in einer Animation von „ET“ zu „ENGEL-Translate“ morphet. Dies verdeutlicht, wie die Marke und das Tool miteinander verbunden sind und ihre Funktionalität hervorheben. Zum Abschluss des Videos werden sowohl das ENGEL-Logo als auch das Logo der HTL Perg eingeblendet, gefolgt von den Namen der beteiligten Personen. Diese finale Szene rundet das Video ab und lässt den Zuschauer mit einem klaren Eindruck von der Effizienz und dem Nutzen des ENGEL Translate Tools zurück.

Das gesamte Szenario zeigt auf anschauliche Weise, wie das ENGEL Translate Tool Sprachbarrieren in einem beruflichen Umfeld schnell und einfach überwindet und somit die Kommunikation zwischen verschiedenen Sprachräumen ermöglicht. Es wird eine Lösung für ein alltägliches Problem präsentiert, die den Arbeitsalltag erleichtert und die Zusammenarbeit optimiert.

### **9.2.2 Video-Dreh & Video-Schnitt**

Das Video wurde im Rahmen eines kleinen Freigegegenstandes an der Schule gedreht und geschnitten. Dabei haben wir den EDV-Raum der HTL Perg als Drehort gewählt, um eine authentische und professionelle Arbeitsumgebung darzustellen. Für die Bearbeitung und den Schnitt des Videos haben wir Adobe Premiere Pro verwendet, um die Szenen präzise zusammenzufügen und mit den nötigen visuellen Effekten sowie Übergängen auszustatten. So konnten wir die gesamte Handlung flüssig und ansprechend gestalten, um die Funktionalität des ENGEL-Translate Tools überzeugend zu präsentieren.

# 10 Organisation

## 10.1 Vorgehensmodell [5]

### 10.1.1 Theorie SCRUM

#### Hauptmerkmale

- Drei Rollen
- Sammlung aller Anforderungen in einem Backlog (Eine Art Vision wird festgelegt vom Product Owner und von der Kundin. Außerdem beschreibt der Product Owner mit User Stories welche Funktionen die Software erfüllen soll.)
- Iterative Produktentwicklung von Zyklen, welche zeitlich eindeutig definiert sein müssen.
- Ein Team mit gleichberechtigten Mitgliedern, welches eigenständig arbeiten kann.

#### Zentrale Rollen

- Product Owner: ist die Schnittstelle zur Auftraggeberin (z.B.: Kundin) und arbeitet im Team mit -> verwaltet und erstellt Anforderungen
- Scrum Master: die Moderatorin, ist bzw. kann Mitglied des Teams sein und stellt einen reibungslosen organisatorischen Ablauf sicher.
- Team: Entwicklerinnen, welche die Anforderungen umsetzen

Eine Projektmanagerin gibt es nicht. Diese Aufgaben übernehmen Product Owner und Team.

#### Ablauf nach Scrum

**Sprint Planning Meeting:** Hier wird das Ziel des folgenden Sprints festgelegt. Anforderungen werden als Tasks in den Backlog geschrieben.

**Sprint:** Der Sprint ist die Entwicklungsphase, in der das Team arbeitet. Am Ende eines Sprints muss immer eine lauffähige Version der Software vorhanden sein. Der letzte Sprint (mit Übergabe an die Kundin) wird Release-Sprint genannt.

**Daily Scrum:** Das ist ein Meeting, welches am Anfang jeden Arbeitstages für maximal

15 Minuten durchgeführt wird. Es wird besprochen, was am letzten Arbeitstag vorangegangen ist, und was als nächstes geplant ist. Außerdem wird besprochen, ob es Probleme mit dem Vorankommen der aktuellen Tasks gibt.

**Sprint Review:** In dieser Phase sind alle Betroffenen des Projektes anwesend und das Team präsentiert die aktuelle Version der Software. Es wird festgestellt, ob die Version den Ansprüchen der Kundin entspricht.

**Sprint Retrospektive:** Die Scrum Masterin moderiert ein Meeting, in dem die positiven und negativen Aspekte der Arbeit im vergangenen Sprint besprochen werden. Es werden Maßnahmen für folgende Sprints gesetzt.

### Fortschrittskontrolle mithilfe von

- **User Stories:** Eine Form der Beschreibung von Anforderungen aus der Sicht von Kundin oder Anwenderin (z.B.: „Als Lagerarbeiterin möchte ich einen Artikel nach seiner Artikelnummer finden, um einen Kundinnenauftrag kommissionieren zu können.“)
- **Sprint Backlogs:** Die User Stories werden im Backlog in die Kategorien „offen“, „in Arbeit“ und „fertig“ eingeteilt. Man verwendet dazu gerne eine Pinnwand oder extra Software zum Visualisieren.
- **Burndown Charts:** Das sind Grafiken, welche die Abarbeitung der Tasks im Sprint darstellt. (Die Entwicklungsgeschwindigkeit wird als Velocity bezeichnet.)
- **Timeboxing:** Das heißt, dass Deadlines unbedingt einzuhalten sind. Z.B.: Daily Scrum dauert 15 Minuten und nicht länger. Ein Sprint dauert 21 Tage und wird normal nicht verlängert.
- **Definition of Done:** Eine Definition, welche beschreibt, wann eine User Story oder Anforderung als fertig angesehen ist. Zum Beispiel durch Tests, Code-Reviews, Abnahme durch Product Owner. werden Fortschritte kontrolliert.

## 10.1.2 Theorie KANBAN

### Definition

Kanban ist eine Methode, die ursprünglich in den 1950er-Jahren bei Toyota entwickelt wurde, um Produktionsprozesse zu optimieren. Sie wird auch als Teil der „lean production“ bezeichnet. Der Fokus liegt darauf, unnötigen Ballast aus Prozessen zu entfernen, indem Arbeitsmaterialien und Aufgaben nur nach Bedarf bereitgestellt werden.

- **Kernidee:** Arbeitsfluss wird durch bedarfsabhängige Steuerung und Signalkarten („Kanban“) organisiert.
- **Hauptanwendung:** Heute wird Kanban auch in der Software-Entwicklung eingesetzt, um Prozesse schlank und flexibel zu gestalten.

### Regeln

**Pull-Prinzip:** Arbeitsaufgaben werden aktiv von den Mitarbeitenden „gezogen“ (genommen), anstatt sie ihnen aufzuzwingen. Dies stellt sicher, dass nur Aufgaben bearbeitet werden, die aktuell notwendig sind.

**Begrenzung der parallelen Aufgaben (WIP – Work in Progress):** Es wird eine maximale Anzahl an Aufgaben definiert, die sich gleichzeitig in Bearbeitung befinden dürfen. Dadurch soll Überlastung vermieden und der Arbeitsfluss stabil gehalten werden.

**Visualisierung der Arbeitsabläufe:** Arbeitsprozesse werden auf einem Kanban-Board sichtbar gemacht. Typische Spalten: „To Do“, „In Progress“, „Done“. Dies hilft Teams, den Fortschritt und Engpässe auf einen Blick zu erkennen.

**Erhebung von Prozessdaten:** Wichtige Kennzahlen wie die Cycle Time einer Aufgabe werden gemessen. Die Cycle Time ist die Zeit von Beginn einer Aufgabe bis Ende oder Abschluss einer Aufgabe. Diese Daten unterstützen die kontinuierliche Optimierung.

**Kontinuierliche Verbesserung:** Prozesse werden laufend angepasst, um Effizienz und Produktivität zu steigern. Probleme werden als Chance zur Verbesserung betrachtet.

### Cycle Time

**Ziel:** Die Cycle Time soll möglichst kurzgehalten werden, um schnell Ergebnisse zu erzielen und Engpässe zu vermeiden.

Wenn ein Arbeitsschritt blockiert ist, wird dies schnell erkannt:

- Es gibt keine fertiggestellten Aufgaben für den nächsten Schritt.
- Das WIP-Limit im betroffenen Schritt ist erreicht.
- Rückstaus entstehen in den vorherigen Schritten.
- Solche Probleme können durch Umverteilung von Kapazitäten gelöst werden, um den Fluss wiederherzustellen.

### 10.1.3 Verwendete Aspekte

Als Vorgehensmodell wurde während des Praktikums Scrum verwendet.

Beim Praktikum wurde das Daily Standup Meeting mit den anderen Mitarbeiterinnen und Mitarbeitern der Abteilung jeden Tag abgehalten.

Zum Erstellen eines Backlogs, also einer Liste aller Tasks, wurde Jira verwendet. Dies ist eine einfache Software, welche es ermöglicht, Tasks anzulegen und die Tasks aus dem Backlog den Sprints zuzuweisen. Ein Sprint wurde nach einer Woche abgeschlossen.

Nach dem Praktikum wurde Kanban verwendet, da aus zeitlichen Gründen keine täglichen Meetings oder praktischen Arbeiten mehr möglich waren und sich SCRUM daher nicht mehr rentierte. So wurde auf Kanban umgestiegen da wir damit etwas flexibler arbeiten konnten.

Gemeinsam wurden Todo's erstellt. Wenn ein Todo als fertig gekennzeichnet wird, kann dann ein neues Todo von der 'To Do' Liste, auf die 'In Progress' Liste wandern.

Ein traditionelles Kanban-Board wurde nicht verwendet. Die Priorisierung und das Testen von Todo's wurden gemeinsam bearbeitet, da die Arbeitsgruppe nur aus zwei Personen besteht.

## 10.2 Technologien

Hier werden alle Technologien beschrieben, welche zu Organisationszwecken verwendet wurden. Es wurde Azure DevOps zur Übergabe und Sicherung des Codes, Clockify zur Erfassung der gearbeiteten Zeit, Confluence zur Dokumentation der Arbeit, GitLab zur Sicherung des Codes und aller Dokumente, Jira zur Erstellung und Zuteilung von Tasks und Microsoft Teams zur Kommunikation verwendet.

### 10.2.1 Azure DevOps [27]

Azure DevOps ist eine Plattform von Microsoft, die eine Kultur der Zusammenarbeit sowie eine Reihe von Prozessen unterstützt, um Entwicklerinnen, Projektmanagerinnen und weitere Beteiligte bei der Softwareentwicklung zusammenzubringen. Dies ermöglicht es Organisationen, Produkte schneller zu entwickeln und zu verbessern als mit herkömmlichen Ansät-



Abbildung 25: Azure DevOps Logo

zen.

Die Plattform bietet verschiedene Dienste, die über einen Webbrowser oder einen IDE-Client zugänglich sind:

- Azure Boards: Agile-Tools zur Planung und Nachverfolgung von Aufgaben, Codefehlern und Problemen mithilfe von Kanban- und Scrum-Methoden.
- Azure Repos: Bereitstellung von Git-Repositorys oder der Team Foundation-Versionskontrolle (TFVC) für die Quellcodeverwaltung.
- Azure Pipelines: Build- und Releasedienste zur Unterstützung der kontinuierlichen Integration und Bereitstellung von Anwendungen.
- Azure Test Plans: Tools zum Testen von Anwendungen, einschließlich manueller und explorativer Tests sowie kontinuierlicher Tests.
- Azure Artifacts: Möglichkeit für Teams, Pakete wie Maven, npm, NuGet und mehr aus öffentlichen und privaten Quellen zu teilen und in die Pipelines zu integrieren.

Azure DevOps kann entweder als Cloud-Dienst (Azure DevOps Services) oder lokal (Azure DevOps Server) genutzt werden, je nach den spezifischen Anforderungen und Präferenzen einer Organisation.

### 10.2.2 Clockify [28]

Clockify ist ein Zeiterfassungstool, das von Teams und Einzelpersonen genutzt wird, um Arbeitszeiten zu tracken und Projekte besser zu verwalten. Es hilft dabei, die auf Aufgaben verwendete Zeit zu erfassen und Berichte zu erstellen.



Abbildung 26: Clockify Logo

Mit Clockify kann man Zeiten manuell eingeben oder über einen Timer automatisch erfassen. Die erfassten Zeiten können Projekten und Kunden zugeordnet werden, was besonders für Freiberufler und Unternehmen nützlich ist.

In der Java-Entwicklung wird Clockify oft verwendet, um den Zeitaufwand für verschiedene Entwicklungsaufgaben, wie das Programmieren mit Spring Boot oder das Testen von APIs, zu messen. Es kann mit anderen Tools wie Jira, Trello oder Microsoft Teams integriert werden, um Arbeitsabläufe zu optimieren.

Durch seine einfache Bedienung, detaillierten Berichte und flexible Nutzung ist Clockify eine

praktische Lösung für alle, die ihre Arbeitszeit besser organisieren und auswerten möchten.

### 10.2.3 Confluence [29]

Confluence ist ein Wissensmanagement- und Kollaborationstool, das von Teams genutzt wird, um Dokumentationen, Anleitungen und Projektpläne zentral zu speichern und gemeinsam zu bearbeiten.



Abbildung 27: Confluence Logo

Mit Confluence kann man Seiten und Dokumente erstellen, die Inhalte strukturieren und mit dem Team teilen. Es eignet sich besonders gut für Softwareentwicklerinnen, um Spezifikationen, API-Dokumentationen oder technische Konzepte festzuhalten – oft in Verbindung mit Jira, um Entwicklungsaufgaben und Dokumentation zu verknüpfen.

Für Java-Teams ist Confluence nützlich, um z. B. Spring Boot-Architekturen, REST-API-Beschreibungen oder Deployment-Prozesse für WildFly zu dokumentieren. Dank Integration mit anderen Atlassian-Tools wie Jira und Bitbucket lässt sich Confluence gut in den Entwicklungsworkflow einbinden.

Durch seine einfache Bedienung, starke Team-Funktionen und gute Suchbarkeit ist Confluence ein ideales Tool für Unternehmen, die ihr Wissen zentral verwalten und effizient im Team arbeiten möchten.

### 10.2.4 GitLab [30]

GitLab ist eine Plattform für Versionskontrolle, CI/CD und DevOps, die Teams hilft, Software effizient zu entwickeln, zu testen und bereitzustellen. Es basiert auf Git und bietet eine zentrale Umgebung für Code-Verwaltung, Zusammenarbeit und Automatisierung.



Abbildung 28: GitLab Logo

Mit GitLab können Entwicklerinnen ihren Code in Repositories verwalten, Merge Requests erstellen und Änderungen im

Team besprechen. Es enthält integrierte Funktionen für Continuous Integration und Deployment (CI/CD), mit denen sich Software automatisch testen und ausliefern lässt.

In der Java-Entwicklung wird GitLab oft mit Spring Boot, WildFly oder anderen Frameworks genutzt, um Code zu verwalten und Build-Prozesse zu automatisieren. Durch die Integration mit Tools wie Jira, Docker und Kubernetes eignet es sich besonders für moderne, containerisierte Anwendungen.

Dank seiner starken Versionskontrolle, integrierten Automatisierung und flexiblen Nutzungsmöglichkeiten ist GitLab eine leistungsfähige Lösung für Entwicklerteams, die ihre Software effizient und sicher entwickeln möchten.

### 10.2.5 Jira [31]

Jira ist ein leistungsstarkes Tool für Projektmanagement und Fehlerverfolgung. Es wird vor allem in der Softwareentwicklung genutzt, um Aufgaben zu organisieren, den Fortschritt zu verfolgen und Teams effizient zusammenarbeiten zu lassen.



Abbildung 29: Jira Logo

Mit Jira kann man Tickets für Aufgaben, Bugs oder neue Features erstellen und diese mit Status wie 'To Do', 'In Progress' oder 'Done' verwalten. Besonders in Agile-Methoden wie Scrum und Kanban ist Jira sehr beliebt, da es Sprint-Planung, Backlogs und Burndown-Charts unterstützt.

In der Java-Entwicklung wird Jira oft zusammen mit Spring Boot oder WildFly genutzt, um Software-Projekte strukturiert zu managen. Es kann mit Git, CI/CD-Tools und Entwicklungsplattformen wie IntelliJ IDEA oder VS Code integriert werden.

Dank seiner Flexibilität, Anpassungsmöglichkeiten und starken Team-Kollaboration ist Jira ein unverzichtbares Tool für Unternehmen, die ihre Entwicklungsprozesse effizient steuern möchten.

## 10.2.6 Microsoft Teams [32]

Microsoft Teams ist ein Tool für Kommunikation und Zusammenarbeit, das von Unternehmen und Teams genutzt wird, um effizient zusammenzuarbeiten. Es ermöglicht Chat-Nachrichten, Videoanrufe, Datei-Sharing und die Integration mit anderen Microsoft- und Drittanbieter-Tools.



Abbildung 30: Teams Logo

Mit Microsoft Teams können Teams in Kanälen organisiert arbeiten, Nachrichten austauschen und Dokumente direkt in der Anwendung bearbeiten. Besonders für Softwareentwicklerinnen ist es praktisch, da es sich mit Tools wie Jira, Confluence und GitHub kombinieren lässt.

In der Java-Entwicklung wird Microsoft Teams oft genutzt, um Code-Reviews zu besprechen, Projektfortschritte zu koordinieren und Meetings zu organisieren. Es bietet auch Benachrichtigungen und Workflows, die helfen, Aufgaben und Deployments im Blick zu behalten.

Durch seine einfache Nutzung, starke Integration mit Microsoft 365 und viele Anpassungsmöglichkeiten ist Microsoft Teams eine gute Wahl für Unternehmen, die eine zentrale Plattform für Kommunikation und Zusammenarbeit suchen.

## 10.3 Projektteam

### 10.3.1 Vorstellung des Teams

Das Diplomarbeitsteam setzt sich aus zwei engagierten Schülerninnen der HTL Perg mit dem Schwerpunkt Informatik zusammen: Elisabeth Brandstetter und Olivia Kamptner. Gemeinsam haben wir uns das Ziel gesetzt, eine Webanwendung für die Firma ENGEL zu entwickeln, so wie wir es gemeinsam mit der Kundin in einem Proof of Concept erarbeitet haben. Unsere Zusammenarbeit basiert auf einer klaren Aufgabenverteilung, die es uns ermöglicht, unser Wissen und unsere Fähigkeiten optimal zu kombinieren, während wir uns gleichzeitig gegenseitig unterstützen und regelmäßig austauschen. Dieser kooperative Ansatz hat uns geholfen, eine effiziente und produktive Arbeitsumgebung zu schaffen.

Während des gesamten Projektverlaufs haben wir uns darauf konzentriert, sowohl individuelle Stärken als auch die jeweiligen fachlichen Kompetenzen einzubringen. Wir haben durch unsere verschiedenen, aber komplementären Fähigkeiten die Möglichkeit, uns in unterschiedlichen Bereichen des Projekts zu spezialisieren und dennoch stets im Team zusammenzuarbeiten. Diese enge Zusammenarbeit hat uns nicht nur dabei unterstützt, die Ziele der Diplomarbeit zu erreichen, sondern auch unsere eigenen Fähigkeiten weiterzuentwickeln und neue technische und organisatorische Herausforderungen erfolgreich zu meistern.

### 10.3.2 Sicherheit und Backend

Olivia Kamptner hat einen Großteil der technischen Umsetzung des Projekts übernommen und sich insbesondere auf die Entwicklung und Optimierung des Backends konzentriert. Dabei war ihr Hauptfokus die Implementierung der Geschäftslogik, welche als Grundlage für die funktionalen Anforderungen der Anwendung dient. Eine ihrer zentralen Aufgaben war es, die Kommunikation zwischen den verschiedenen Modulen der Webanwendung zu ermöglichen, wobei ein besonderer Wert auf die Skalierbarkeit und Performance gelegt wurde. Ein weiterer wichtiger Aspekt ihrer Arbeit war die Integration der Authentifizierung, die für den sicheren Zugang zur Anwendung sorgt.

Ein wichtiger Bestandteil ihrer Arbeit war die intensive Auseinandersetzung mit der Azure Translator Ressource, einem Cloud-basierten Dienst von Microsoft Azure, der für die maschinelle Übersetzung von Texten verwendet wird. Olivia war verantwortlich dafür, diese Ressource in die Webanwendung zu integrieren und sicherzustellen, dass der Übersetzungsprozess effizient und stabil funktioniert. Sie sorgte dafür, dass die Anwendung die Übersetzungsanforderungen zuverlässig und in Echtzeit umsetzen kann, was durch die leistungsstarke Azure-Technologie

ermöglicht wurde.

Zusätzlich zu diesen Aufgaben kümmerte sich Olivia um die Bereitstellung einer API, die als Schnittstelle zwischen dem Backend und dem Frontend fungiert. Diese API war von zentraler Bedeutung für den reibungslosen Austausch von Daten und Informationen zwischen den beiden Bereichen der Anwendung. Um die Performance und Sicherheit der Anwendung zu gewährleisten, stellte Olivia sicher, dass die API optimiert und gegen mögliche Sicherheitslücken abgesichert wurde. Die API-Programmierung erfolgte unter Verwendung von Java, wodurch eine stabile und skalierbare Lösung geschaffen wurde.

Ein weiteres zentrales Element von Olivias Arbeit war das Deployment der Anwendung auf der Azure-Plattform.

### 10.3.3 Frontend und Organisation

Elisabeth Brandstetter übernahm die Verantwortung für das Frontend der Webanwendung und setzte ihre Expertise in der Gestaltung einer benutzerfreundlichen Oberfläche gezielt ein. Ihr Ziel war es, eine intuitive und ansprechende Benutzeroberfläche zu entwickeln, die den Anwenderinnen eine einfache Navigation ermöglicht und gleichzeitig die technischen Anforderungen der Anwendung unterstützt. Elisabeth legte großen Wert darauf, dass das Frontend nicht nur optisch ansprechend, sondern auch funktional ist, um die bestmögliche Nutzerinnenerfahrung zu gewährleisten. Die Verwendung von modernen Webtechnologien und Frameworks war hierbei von entscheidender Bedeutung, um eine reaktionsschnelle und performante Oberfläche zu schaffen, die auf Endgeräten problemlos genutzt werden kann.

Ein zentraler Bestandteil von Elisabeths Arbeit war die Sicherstellung, dass die Benutzeroberfläche mit der Backend-API nahtlos zusammenarbeitet. Sie arbeitete eng mit Olivia zusammen, um die API-Schnittstellen optimal in das Frontend zu integrieren, sodass die Benutzerinnen die gewünschten Funktionen der Anwendung einfach und effizient nutzen konnten. Hierbei war es wichtig, dass die Anwendung sowohl funktional als auch ästhetisch überzeugte, was durch eine enge Zusammenarbeit und regelmäßige Abstimmung im Team ermöglicht wurde.

Neben der Entwicklung des Frontends kümmerte sich Elisabeth auch um die organisatorischen Aspekte des Projekts. Dies beinhaltete das Verwalten und Organisieren der Protokolle, die während der Meetings erstellt wurden, sowie das Dokumentieren aller wichtigen Entscheidungen und Fortschritte, die im Verlauf des Projekts getroffen wurden. Elisabeth führte die

### 10.3 Projektteam

Verantwortung für die strukturierte Dokumentation des Projekts, was sicherstellte, dass alle Teammitglieder und Betreuerinnen stets über den aktuellen Stand des Projekts informiert waren und dass alle relevanten Informationen für spätere Auswertungen und Nachvollziehbarkeit zur Verfügung standen.

Die Fähigkeit von Elisabeth, sowohl technische als auch organisatorische Aufgaben zu übernehmen, stellte einen wichtigen Erfolgsfaktor für das Team dar. Sie half dabei, die Kommunikation im Team zu optimieren und die Arbeitsabläufe zu koordinieren, sodass das Projekt effizient vorangetrieben werden konnte. Ihre strukturierte Herangehensweise und ihr Fokus auf Details trugen maßgeblich dazu bei, dass alle Aufgaben planmäßig erledigt wurden und das Projekt in die gewünschte Richtung lief.

# 11 Resümee

## 11.1 Resümee Auftraggeberin

Beurteilung des Proof of Concept (POC) zur Erstellung eines zentralen, KI-basierten Übersetzungsdienstes

### **Projektübersicht:**

Die Schülerinnen Elisabeth Brandstetter und Olivia Kamptner haben im Rahmen ihrer Diplomarbeit einen Proof of Concept (POC) für einen zentralen, websitebasierten Übersetzungsdienst entwickelt.

### **Zielsetzung und Anforderungen:**

Das Ziel des Projekts war es, einen nutzerfreundlichen und effizienten Übersetzungsdienst zu entwickeln, der in der Lage ist, Texte und Dokumente schnell und zuverlässig zu übersetzen. Die Schülerinnen haben das Thema sehr gut verstanden und die Zielsetzung klar und präzise formuliert. Die Option zur Integration eines ENGEL Glossars sowie die Auswahl geeigneter Azure Cognitive Services wie Textübersetzung und weitere KI-Dienste wurden sinnvoll und durchdacht in die Arbeit integriert.

### **Methodik und Umsetzung:**

Der gesamte SW-Entwicklungsprozess von der Anforderungserhebung bis hin zur Abschlusspräsentation und Übergabe wurde erfolgreich abgewickelt und dokumentiert. Die Schülerinnen haben die Anforderungen an den Übersetzungsdienst sorgfältig analysiert und die Entscheidung, Azure Cognitive Services zu nutzen, begründet getroffen. Azure Cognitive Services bieten eine skalierbare und leistungsfähige Lösung, die es ermöglicht, Übersetzungsdienste effizient bereitzustellen. Der Übersetzungsdienst stützt sich auf den Azure Translate Service, eine leistungsstarke KI-Technologie, die eine automatische Übersetzung von Texten und Dokumenten in zahlreiche Sprachen ermöglicht.

### **Technische und organisatorische Umsetzung:**

Die Schülerinnen haben sehr erfolgreich mit den technischen Ansprechpartnern bei ENGEL zusammengearbeitet. Die Kommunikation und regelmäßigen Abstimmungen im Rahmen des Praktikums waren hervorragend organisiert, was eine effiziente und zielorientierte Umsetzung des Projekts ermöglichte.

### **Zusammenarbeit und Arbeitsweise:**

Die Zusammenarbeit mit den Schülerinnen war vorbildlich. Elisabeth und Olivia haben das 4-wöchige Praktikum sehr gut genutzt und sich von Beginn an als äußerst engagiert und motiviert gezeigt. Sie waren stets freundlich, wissbegierig und zeigten eine hohe Selbstständigkeit bei der Arbeit. Ihr Interesse an der Materie war spürbar, und sie haben sich aktiv in die Diskussionen eingebracht, um die bestmögliche Lösung zu erarbeiten. Ihre Arbeitsweise war stets strukturiert und zielorientiert, wodurch sie den vorgegebenen Zeitplan zufriedenstellend einhalten konnten.

#### **11.1.1 Fazit:**

Die Arbeit von Elisabeth Brandstetter und Olivia Kamptner ist äußerst gelungen und erfüllt die Anforderungen des Projekts. Sie haben ein funktionierendes, gut dokumentiertes Proof of Concept für einen zentralen Übersetzungsdienst entwickelt, das auf Azure Cognitive Services und Azure Translate basiert. Das POC bietet eine vielversprechende Grundlage für eine mögliche Erweiterung des Übersetzungsdienstes bei ENGEL, insbesondere durch mögliche Ausbaustufen wie Drittsystemanbindung oder Abrechnungsvorschrift. Die Schülerinnen haben nicht nur fachlich überzeugt, sondern auch durch ihre exzellente Zusammenarbeit und Arbeitsweise während des gesamten Projekts.

#### **Note:**

Sehr gut

## 11.2 Resümee Brandstetter Elisabeth

Bereits in der Vorbereitungsphase vor dem Praktikum hatten wir viel Kontakt mit der Firma ENGEL. Dadurch konnten wir uns optimal auf die bevorstehenden Aufgaben vorbereiten und ohne Verzögerung in die praktische Arbeit starten. Auch nach dem Wechsel des Auftraggebers blieb die Kommunikation stets reibungslos, und es gab keinerlei Schwierigkeiten, Termine zu koordinieren. Es standen uns nicht nur in der Zeit während des Praktikums, sondern auch davor sowie danach immer Ansprechpartner\*innen zur Verfügung. Wenn diese keine Zeit hatten, wurden wir zu anderen Ansprechpartnern\*innen weitergeleitet.

Die praktische Arbeit selbst gestaltete sich herausfordernd, insbesondere aufgrund des hohen Rechercheaufwands. Dennoch betrachte ich diese Erfahrung als wertvollen Lernprozess, der mir nicht nur vertiefte Fachkenntnisse, sondern auch verbesserte Selbstlernkompetenzen gebracht hat – Fähigkeiten, die mir auf meinem weiteren Karriereweg ohne Zweifel von Nutzen sein werden. Das Thema unserer Diplomarbeit gefällt mir besonders da es einen echten Nutzen und Mehrwert für die Firma ENGEL hat und sich mit dem Zukunftsthema KI befasst.

## 11.3 Resümee Kamptner Olivia

Die Zusammenarbeit mit der Kundin war durchweg positiv. Von Beginn an verlief die Kommunikation reibungslos, da uns viele Ansprechpartner\*innen zur Verfügung standen. Besonders hervorzuheben ist die Hilfsbereitschaft der Mitarbeiter\*innen bei ENGEL. Egal, an wen wir uns mit unseren Fragen gewandt haben, wir wurden stets freundlich unterstützt. Die offene und angenehme Atmosphäre im Unternehmen hat das Arbeiten zusätzlich erleichtert.

Das Projekt selbst brachte einige Herausforderungen mit sich. Besonders die Arbeit mit Azure erwies sich als komplexer als zunächst erwartet. Es gab einige Hürden zu überwinden, und nicht alles funktionierte auf Anhieb. Dennoch konnten wir durch kontinuierliches Testen und Anpassen eine funktionierende Lösung erarbeiten, deren Ergebnisse, wie in einem anderen Kapitel beschrieben, unseren Erwartungen entsprachen.

Trotz der Herausforderungen war die gesamte Erfahrung sehr wertvoll. Ich konnte viel Neues lernen und mir wertvolle Erkenntnisse für zukünftige Projekte mitnehmen. Besonders erfreulich war die gute Organisation innerhalb unseres Teams. Die Abstimmung und Aufgabenverteilung haben reibungslos funktioniert, was zum erfolgreichen Abschluss des Projekts beigetragen hat.

# Glossar

**API** - Application Programming Interface - Schnittstelle, über die Programme miteinander kommunizieren können

**Business Intelligence** - Auswertung von Geschäftsdaten zur Unterstützung von Entscheidungen

**CDI** - Custodial Detention Index - Java-Framework für die Verwaltung von Abhängigkeiten zwischen Komponenten

**CI/CD** - Continuous Integration/Continuous Development - Automatisierte Prozesse zur Softwareentwicklung und -bereitstellung

**Compliance-Funktionen** - Funktionen, die sicherstellen, dass gesetzliche oder unternehmensinterne Regeln eingehalten werden

**CRM** - Customer Relationship Management - Systeme zur Verwaltung von Kundenbeziehungen und -daten

**Data Warehousing** - Speicherung und Analyse großer Datenmengen aus verschiedenen Quellen

**DB** - Datenbank - strukturierte Sammlung von Daten

**EL** - Expression Language - Sprache zur einfachen Einbindung von dynamischen Inhalten in Java-Webanwendungen

**Elastic Pool** - Datenbanklösung, bei der mehrere Datenbanken sich Ressourcen teilen können

**Enterprise Anwendungen** - Softwarelösungen für Unternehmen zur Unterstützung ihrer Geschäftsprozesse

**ERP** - Enterprise Resource Planning - Software zur Verwaltung von Geschäftsprozessen

**ET** - ENGEL-Translate

**ETT** - ENGEL-Translation-Tool

**HTML** - Hyper Text Markup Language - Standardauszeichnungssprache für Webseiten

**Hyperscale** - skalierbare Datenbankarchitektur für große Datenmengen und hohe Leistung

- JDK** - Java Development Kit - Softwarepaket zur Entwicklung von Java-Anwendungen
- JIT** - Just In Time Compiler - Compiler, der Code erst zur Laufzeit übersetzt
- JPA** - Java Persistence API - Java-Spezifikation, die das Speichern, Abrufen und Verwalten von Daten in relationalen Datenbanken vereinfacht
- JRE** - Java Runtime Environment - Umgebung zur Ausführung von Java-Anwendungen
- JSF** - Java Server Faces - komponentenbasiertes Java-Framework zur Erstellung interaktiver Webanwendungen
- JSP** - Java Server Pages - Java-Technologie zur Erstellung dynamischer Webseiten mit serverseitigen Skripten
- JVM** - Java Virtual Machine - Laufzeitumgebung, die Java-Code ausführt
- LOB** - line-of-business - Geschäftsanwendungen, die speziell für ein Unternehmen oder eine Branche entwickelt wurden
- Managed Instance** - verwaltete SQL-Datenbanklösung mit umfassenden Funktionen für Unternehmen
- MVC** - Model-View-Controller - Architekturmodell zur Trennung von Daten, Darstellung und Logik einer Anwendung
- On-Premise** - Datenbank, die lokal auf eigenen Servern betrieben wird, statt in der Cloud
- Open-Source** - Software, deren Quellcode öffentlich zugänglich ist und von jedem verändert werden kann
- Power BI** - Microsoft-Tool zur Datenanalyse und Visualisierung
- REST** - Representational State Transfer - Architekturstil für Web-APIs zur Kommunikation zwischen Systemen
- SaaS** - Software as a Service - Software, die über das Internet genutzt wird, ohne dass eine Installation nötig ist
- Single Database** - eigenständige Datenbank, die für eine Anwendung oder einen Tenant genutzt wird
- SQL** - Structured Query Language - Sprache zur Abfrage und Bearbeitung von Datenbanken

**SSO** - Single-Sign-On - Anmeldeverfahren, bei dem sich Nutzer nur einmal anmelden müssen, um mehrere Dienste zu nutzen

**Tenant** - Eine logische Einheit, welche Benutzer, Anwendungen, Daten usw. einer Organisation verwaltet.

**UI** - User Interface - Schnittstelle, über die Nutzer mit einer Software oder Website interagieren

**XML** - X-Markup Language - Sprache zur Strukturierung und Speicherung von Daten

# Literaturverzeichnis

- [1] P. Pedamkar, *JSP vs JSF*, 2023. Online verfügbar: <https://www.educba.com/jsp-vs-jsf/>
- [2] tutorialspoint, *JSF - Managed Beans*. Online verfügbar: [https://www.tutorialspoint.com/jsf/jsf\\_managed\\_beans.htm](https://www.tutorialspoint.com/jsf/jsf_managed_beans.htm)
- [3] *App Service: Übersicht*. Online verfügbar: <https://learn.microsoft.com/de-de/azure/app-service/overview>
- [4] JavaInUse, *Deploy a Spring Boot App (JAR) to Azure App Service*, 2024. Online verfügbar: [https://www.youtube.com/watch?v=EkhRs\\_PbWeE](https://www.youtube.com/watch?v=EkhRs_PbWeE)
- [5] I. S.-M. Felix Schwab, *SYSTEMPLANUNG UND PROJEKTENTWICKLUNG*, 2013. Online verfügbar: <https://hoelzel.at/flip/160775/index.html>
- [6] mrbullwinkle, „What is Azure OpenAI Service?“ *Microsoft Learn*, 2024. Online verfügbar: <https://learn.microsoft.com/en-us/azure/ai-services/openai/overview>
- [7] barclayn, „What is Microsoft Entra ID?“ *Microsoft Learn*, 2024. Online verfügbar: <https://learn.microsoft.com/en-us/entra/fundamentals/whatis>
- [8] *Azure Document Translator client library for Java*. Online verfügbar: <https://learn.microsoft.com/en-us/java/api/overview/azure/ai-documenttranslator-readme?view=azure-java-legacy>
- [9] *Azure Products*. Online verfügbar: <https://azure.microsoft.com/en-us/products/#databases>
- [10] *Azure Database for MySQL*. Online verfügbar: <https://azure.microsoft.com/en-us/products/mysql/>
- [11] *Azure SQL Database*. Online verfügbar: <https://azure.microsoft.com/en-us/products/azure-sql/database/>
- [12] *What is Azure Text Translation?*, 2025. Online verfügbar: <https://learn.microsoft.com/en-us/azure/ai-services/translator/text-translation/overview>
- [13] *What is Document Translation?*, 2025. Online verfügbar: <https://learn.microsoft.com/en-us/azure/ai-services/translator/document-translation/overview>
- [14] B. Hofmann, *So funktioniert OAuth2*, 2012. Online verfügbar: <https://entwickler.de/security/so-funktioniert-oauth2>
- [15] barclayn, *Was ist Microsoft Entra ID?*, 2024. Online verfügbar: <https://learn.microsoft.com/de-de/entra/fundamentals/whatis>
- [16] paulliew, *Tutorial: Register an app with Microsoft Entra ID*, 2025. Online verfügbar: <https://learn.microsoft.com/en-us/power-apps/developer/data-platform/walkthrough-register-app-azure-active-directory>

- [17] J. Freeberg, *Azure-Samples/app-service-wildfly*. Online verfügbar: <https://github.com/Azure-Samples/app-service-wildfly>
- [18] Sebastian.Dietrich, *Java (Programmiersprache)*, 2025. Online verfügbar: [https://de.wikipedia.org/wiki/Java\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Java_(Programmiersprache))
- [19] *Entdecken Sie die Azure-Welt*, 2025. Online verfügbar: <https://azure.microsoft.com/de-de/explore/>
- [20] *The IDE for Professional Development*, 2025. Online verfügbar: <https://www.jetbrains.com/idea/>
- [21] *Postman Docs*, 2025. Online verfügbar: <https://learning.postman.com/>
- [22] *Spring Boot*, 2025. Online verfügbar: <https://docs.spring.io/spring-boot/>
- [23] *WildFly Documentation*, 2025. Online verfügbar: <https://docs.wildfly.org/>
- [24] *Designen im Rekordtempo*. Online verfügbar: <https://www.adobe.com/de/products/illustrator.html>
- [25] *Erschaffe einzigartige Bilder schneller als je zuvor*. Online verfügbar: <https://www.adobe.com/de/products/photoshop.html>
- [26] *Videobearbeitung mit Adobe – da steckt mehr für dich drin*. Online verfügbar: <https://www.adobe.com/de/products/premiere.html>
- [27] steved0x, *MicrosoftDocs/azure-devops-docs*, 2025. Online verfügbar: <https://github.com/MicrosoftDocs/azure-devops-docs>
- [28] *Kostenloses Zeiterfassungssystem*, 2025. Online verfügbar: <https://clockify.me/de/time-entry-system>
- [29] *Goodbye silos,hello teamwork*, 2025. Online verfügbar: <https://www.atlassian.com/software/confluence>
- [30] *GitLab*, 2025. Online verfügbar: <https://gitlab.com/gitlab-org/gitlab>
- [31] *Great outcomes start with Jira*, 2025. Online verfügbar: <https://www.atlassian.com/software/jira>
- [32] *Microsoft Teams-Hilfe und -Learning*, 2025. Online verfügbar: <https://support.microsoft.com/de-de/teams>

# Abbildungsverzeichnis

1	Gemeinsame Erarbeitung der Architektur . . . . .	4
2	Endgültiges Logo . . . . .	7
3	Foto vom Brainstorming des Mockups . . . . .	8
4	Login Seitendesign in Canva (links) . . . . .	8
5	Home Seitendesign in Canva (links) . . . . .	9
6	Textübersetzung Seitendesign in Canva (links) . . . . .	9
7	Dokumentenübersetzung Seitendesign in Canva (links) . . . . .	9
8	User Seitendesign in Canva (links) . . . . .	10
9	Translator Ressource . . . . .	12
10	Datenbankauswahl in Azure . . . . .	13
11	CSV-Test-Glossar . . . . .	31
12	Java Logo . . . . .	33
13	Azure Logo . . . . .	33
14	IntelliJ Logo . . . . .	34
15	Postman Logo . . . . .	34
16	Springboot Logo . . . . .	35
17	Wildfly Logo . . . . .	35
18	Startseite (Deutsch) . . . . .	37
19	Textübersetzung (Deutsch) - Sprachauswahlsdialog . . . . .	37
20	Dokumentenübersetzung (Deutsch) . . . . .	38
21	Illustrator Logo . . . . .	39
22	Photoshop Logo . . . . .	40
23	Premiere Pro Logo . . . . .	40
24	Storyboard . . . . .	41
25	Azure DevOps Logo . . . . .	47
26	Clockify Logo . . . . .	48
27	Confluence Logo . . . . .	49
28	GitLab Logo . . . . .	49
29	Jira Logo . . . . .	50
30	Teams Logo . . . . .	51

# Tabellenverzeichnis

1	Tabellarischer Vergleich von Cognitive Services und Open AI . . . . .	5
2	Vergleich Glossar - Ausgangslage und Übersetzung . . . . .	31

# Anhang

## A Aufgabenverteilung

## B Meilensteine

## C ChatGPT

Bei der Erstellung der entsprechend gekennzeichneten Teile dieser Diplomarbeit wurde ChatGPT-4 eingesetzt. Dieses Tool ist unter der Adresse <https://chat.openai.com/chat> online verfügbar. In den folgenden Abschnitten sind die Benutzereingaben und die dazugehörigen Ausgaben von ChatGPT dargestellt.

### C.1 Azure DevOps

**Eingabe:** Schreibe eine kurze Zusammenfassung von Azure DevOps mithilfe nachfolgender Quelle: <https://learn.microsoft.com/de-de/azure/devops/user-guide/what-is-azure-devops>

**Ausgabe:** Azure DevOps ist eine Plattform von Microsoft, die eine Kultur der Zusammenarbeit sowie eine Reihe von Prozessen unterstützt, um Entwickler, Projektmanager und weitere Beteiligte bei der Softwareentwicklung zusammenzubringen. Dies ermöglicht es Organisationen, Produkte schneller zu entwickeln und zu verbessern als mit herkömmlichen Ansätzen.

Die Plattform bietet verschiedene Dienste, die über einen Webbrowser oder einen IDE-Client zugänglich sind:

- Azure Boards: Agile-Tools zur Planung und Nachverfolgung von Aufgaben, Codefehlern und Problemen mithilfe von Kanban- und Scrum-Methoden.
- Azure Repos: Bereitstellung von Git-Repositorys oder der Team Foundation-Versionskontrolle (TFVC) für die Quellcodeverwaltung.
- Azure Pipelines: Build- und Releasedienste zur Unterstützung der kontinuierlichen Integration und Bereitstellung von Anwendungen.
- Azure Test Plans: Tools zum Testen von Anwendungen, einschließlich manueller und explorativer Tests sowie kontinuierlicher Tests.
- Azure Artifacts: Möglichkeit für Teams, Pakete wie Maven, npm, NuGet und mehr aus öffentlichen und privaten Quellen zu teilen und in die Pipelines zu integrieren.

Azure DevOps kann entweder als Cloud-Dienst (Azure DevOps Services) oder lokal (Azure DevOps Server) genutzt werden, je nach den spezifischen Anforderungen und Präferenzen einer Organisation.

**Eingabe:** Nenne eine Quelle für die obige Formulierung

**Ausgabe:** Die Informationen in der vorherigen Antwort stammen aus dem Artikel "Was ist Azure DevOps?" auf Microsoft Learn.

Quelle: <https://learn.microsoft.com/de-de/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>

### C.2 L<sup>A</sup>T<sub>E</sub>X

**Eingabe:** Erkläre in einem Satz was Latex ist.

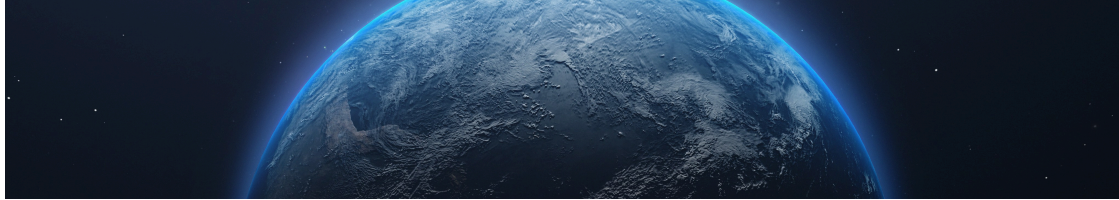
**Ausgabe:** LaTeX ist ein textbasiertes Satzsystem, das hauptsächlich zur Erstellung von wissenschaftlichen Dokumenten verwendet wird, da es präzise Formatierung und mathematische Formeln unterstützt.

**Eingabe:** Nenne eine Quelle für die obige Formulierung

**Ausgabe:** Die Formulierung basiert auf einer allgemeinen Definition von LaTeX. Eine ähnliche Beschreibung findest du auf der offiziellen LaTeX-Projektseite:

Quelle: <https://www.latex-project.org/about/>

# Proof of Concept



## 24.10001 AzureTranslate: POT Spezifikation

Diese Spezifikationen sollen als Grundlage für die Entwicklung des Online-Übersetzers dienen und sicherstellen, dass der Dienst den Bedürfnissen der Nutzer entspricht und sich von anderen - bereits bei ENGEL eingesetzten Tools - abhebt. Die Integration von KI gestützten Services wird die Qualität und Effizienz des Übersetzungsdienstes maßgeblich steigern und eine solide Basis für die Skalierbarkeit, Abrechnung und Sicherheit bieten.

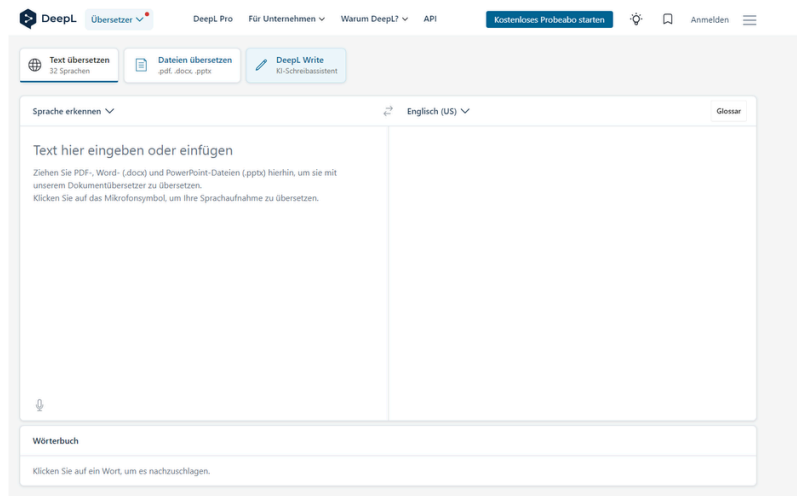
### Aufbau der Startseite [↗](#)

Die Startseite eines Übersetzungsdienstes sollte folgende Informationen enthalten:

- **Logo und Name des Dienstes:** ENGEL spezifisches Design: ENGEL Farbcode verwenden (HEX: #97C138), ENGEL Logo und der Name des Übersetzungsdienstes: ENGEL Translate (Vorschlag - eure kreativen Ideen sind gefragt)
- **Navigationsleiste:** Eine intuitive Navigationsleiste mit Links zu verschiedenen Seiten: Features & Vorteile (Technische Details/Infos, Über Uns, ...), Verrechnung, Kontakt ([it-helpdesk@engel.at](mailto:it-helpdesk@engel.at); Feedbackmöglichkeiten), Hilfe (ev. Schulungsvideo mit Powtoon) usw.
- **Guidelines für den Übersetzungsservice:** Hinweise zu den Grenzen der Maschinenübersetzung und Verantwortung beim Redakteur (Plausibilitätscheck).
- **Übersetzungsfeld:** Ein zentrales Feld, in dem Benutzer Text eingeben und Sprache wählen kann (Dropdown-Menüs oder Schaltflächen zur Auswahl der Ausgangs- und Zielsprache.)
- **Dokument-Upload:** Dokumente für die Übersetzung hochladen können und Sprache wählen kann. (Dropdown-Menüs oder Schaltflächen zur Auswahl der Ausgangs- und Zielsprache)
- **Anmelde-/Registrierungsbereich:** Single-Sign-On Anmeldung integrieren.

Diese Elemente tragen dazu bei, dass die Startseite informativ, benutzerfreundlich und einladend für neue und wiederkehrende Benutzer ist.

### Orientierung an bekannten, großen "Playern" (Beispiel DeepL) [↗](#)



Beispiel "DeepL"

### Funktionale Anforderungen [↗](#)

Entwicklung eines benutzerfreundlichen, effizienten und präzisen Online-Übersetzungsdienstes, der Texte, Dokumente in Echtzeit übersetzen kann.

- **Textübersetzung (Prio 1):** Schnelle und genaue Übersetzung von Texten in verschiedenen Sprachen (Deutsch, Englisch, Tschechisch, Chinesisch, Koreanisch, Indisch).
- **Dokumentenübersetzung (Prio 2):** Unterstützung für gängige Dateiformate wie Word (.docx), PowerPoint (.pptx), ggf. PDF (.pdf). Max. 1 Datei pro Übersetzung, Max. 25 MB Dateigröße, Automatischer Browserdownload der übersetzten Datei.
- **OPTIONAL: Integration eines Glossars:** Optionale Integration eines ENGEL Glossars; Datenbank des Glossars verwenden, die Befüllung soll im Hintergrund passieren.
- **OPTIONAL: Übersetzungsdienst für Drittsysteme:** Anbindung von anfordernden ENGEL Systemen/Tools die Übersetzungen abrufen wollen (Serviceuser, REST-Schnittstelle)

### Technische Anforderungen [↗](#)

- **Website-Implementierung:** Hosting der Website als Ressource im Azure, Verwendung von Blob-Service und ASPX-Seiten.
- **Abrechnungsvorschrift:** Nutzungsbasierte Zahlung auf Kostenstellenbasis und Auswertungsmöglichkeit im Azure Portal.
- **Mehrsprachigkeit:** Bereitstellung der Website in Deutsch und Englisch.
- **Datensicherheit:** Die importierten Daten dürfen nicht außerhalb des ENGEL Netzwerkes gespeichert und verwendet werden.
- **Einsatz geeigneter KI Services:** Azure Cognitive Services, Azure Translator Text API oder Azure Open AI Service

# Beispiel Besprechungsprotokoll



ET – ENGEL Translate

**Datum:** 10.07.2024, 11.07.2024

**Beteiligte Personen:** Elisabeth, Olivia, Gudrun

**Ort:** ENGEL Schwertberg

**Anlass:**

- Auswahl der Ressource besprechen (mit Begründung)
- Datenbank Auswahl

**Ergebnisse (Projektteam):**

- Tasks für die Sprints sind erstellt und am Laufen
- Protokolle, Recherchen und Ausarbeitungen auf Confluence veröffentlicht
- Erster Sprint abgeschlossen – neuer begonnen
- Recherche zu OpenAI gemacht und die Ressource mit der Translator Ressource verglichen (Confluence Seite)
- Glossar erhalten – Zugriff bekommen und analysiert
- Datenbank ausgewählt und in Azure angelegt (MySQL – flexible Server)
- Testung Ressource im Intellij -> Fließtext-Übersetzung funktioniert
- Ausarbeitung von Entra ID und Single-Sign-On
- Grobe Planung von Backend-Architektur (Klassen usw.) auf Papier

**Was steht an:**

- Datenbankschema (Abrechnung Besprechung) -> [REDACTED]
- Testung Ressource im Intellij -> File-Übersetzung
- User Authentifizierung in Azure
- Datenbank verwenden
- Grundstruktur der Frontend Oberfläche

**Aufgaben (Projektteam):**

- Besprechung mit [REDACTED] / [REDACTED] wegen Azure AI Ressourcen
  - AI-Hub wurde mit Azure Ressource gemacht
  - Vergleich: Was ist AI-Hub? Was machen wir (ET)?

**Aufgaben (Auftraggeber):**

- Mehr Details über Abrechnung

02.04.2025

1

ET – ENGEL Translate



## Fragen:

- **Abrechnung**
  - Wie geht das?
  - Was müssen wir speichern?
  - Wie lange müssen wir was speichern?
  - Wen dürfen wir Fragen/Ansprechen?

## Mitschrift:

Kontakt für Ressourcen Vergleich Besprechung:

██████████ (Chef)

██████████ (Analytics-Technology) Abrechnung auf Bereichsebene -> ██████████  
██████████

## **Nächste Schritte:**

██████████ kontaktieren, um mit ihm Azure OpenAI und Azure Translator zu besprechen. → Expertenmeinung einholen

Fragen, wie sie (██████████ und ██████████) die Webseite AI-Hub gemacht haben und ob wir Zugriff auf die Seite bekommen.

## Ergebnis der Machbarkeitsstudie des Glossars

### Glossar Fragen:

Wird auch richtig übersetzt, wenn das Wort falsch geschrieben wurde?

Wird auch richtig übersetzt, wenn das Wort vor nach einem Beistrich steht?

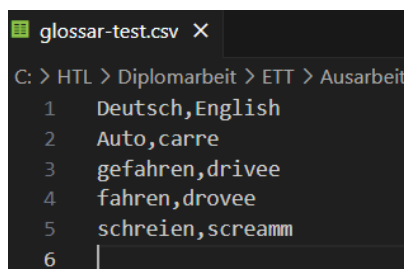
Wird auch richtig übersetzt, wenn das Wort neben Satzzeichen steht wie „?, !, .,“ usw.?

Wird auch richtig übersetzt, wenn das Wort am Satzanfang steht und somit großgeschrieben wird?

Wird auch richtig übersetzt, wenn das Wort nachfolgende oder führende Leerzeichen hat?

Wird auch richtig übersetzt, wenn das Wort leicht verändert ist? zB. Auto, Autos, AUTO, auto, ...

Test-CSV:



```
glossar-test.csv X
C: > HTL > Diplomarbeit > ETT > Ausarbeit
1 Deutsch,English
2 Auto,carre
3 gefahren,drivee
4 fahren,drovee
5 schreien,screamm
6
```

Die verschiedenen Testfälle bzw. die Eingabe: Die Ausgabe bzw. das Ergebnis:

Das <b>Auto</b> fährt sehr schnell.	The <b>car</b> drives very fast.
<b>Autos</b> werden in der Werkstatt repariert.	<b>Cars</b> are repaired in the workshop.
Ich finde <b>Autos</b> , Fahrräder und Boote cool.	I think <b>cars</b> , bicycles and boats are cool.
Darf ich <b>auto fahren</b> ?	Can I <b>drove</b> ?
Das Kind <b>schreit</b> : „ <b>Auto</b> “!	The child <b>screams</b> : " <b>Carre</b> "!
Wird das <b>Auto</b> gerade von jemandem <b>gefahren</b> ?	Is the <b>car</b> being <b>driven</b> by someone right now?
<b>Auto</b> ?	<b>Car</b> ?
<b>Autos</b> ?	<b>Automobiles</b> ?
<b>Auto</b> korrektur	<b>Auto</b> correct
<b>auto</b> korrektur	<b>Auto</b> correct
<b>auto</b> bahn	motorway
<b>Auto</b> bahn	Motorway
<b>Audo auto autu Autu Aoto Aauto Autoo</b>	<b>Audo auto autu Autu Aoto Aauto Autoo</b>
Darf ich <b>Audo</b> fahren?	Can I <b>drove Audo</b> ?
Ich darf <b>Auto</b> fahren.	I am allowed to <b>drove a car</b> .
<b>AUTO</b>	<b>CAR</b>

Das Glossar hat nur in einem Fall richtig funktioniert, das ist, wenn man das Wort unter Hochkomma schreibt. (siehe Vergleich grün maskiertes Wort)

Wenn das Wort in Großbuchstaben geschrieben ist, wird es nicht übersetzt.

Ebenso nicht wenn es falsch geschrieben wurde und wenn es unabsichtlich mit dem vorherigen Wort zusammengeschrieben wurde.

Manchmal wird es übernommen aber nicht in der richtigen form so ist es nicht richtig übersetzt mit dem Glossar und auch nicht richtig übersetzt, wenn es ohne Glossar gemacht worden wäre. Wenn das Wort in einem anderem enthalten ist, wird dieses nicht mit dem Glossar übersetzt.

Das Engel Glossar kann nicht verwendet werden zum Übersetzen von Fachbegriffen, aufgrund der oben genannten Analyse und aufgrund des Beschaffenheiten des Engel Glossars. Wenn Sätze im Glossar stehen, die einen Beistrich haben werden diese aufgetrennt und die eine hälfte des Satzes wird mit die andere Übersetzt. Da Azure als Trennzeichen einen Beistrich verwendet.