n wir uns für die von Amazon entwickelte API namens “Polly” entschieden. Diese bietet verschiedene Sprachen und dazu jeweils unterschiedliche Akzente oder Dialekte an.  
**TP / Informatik Projekt: Text-to-Speech**

Inhaltsverzeichnis

[Themenbeschreibung 3](#_Toc105023010)

[Unsere Vorstellung 3](#_Toc105023011)

[Potenziell nutzbare APIs 3](#_Toc105023012)

[Vorgehensmodell: Wasserfallmodell 4](#_Toc105023013)

[Erklärung 4](#_Toc105023014)

[Anforderungsanalyse 5](#_Toc105023015)

[Entwurf erstellen 6](#_Toc105023016)

[User Case Diagramm 6](#_Toc105023017)

[Diagramme 9](#_Toc105023018)

[Quellen 10](#_Toc105023019)

[Dokumentation Anforderungsanalyse 11](#_Toc105023020)

[Erklärung 11](#_Toc105023021)

[Minimale Anforderungen 11](#_Toc105023022)

[Potenzielle Erweiterungen 11](#_Toc105023023)

[Fazit 12](#_Toc105023024)

[Arbeitstagebuch (Planung) 13](#_Toc105023025)

[*22.03.2022* 13](#_Toc105023026)

[*29.03.2022* 13](#_Toc105023027)

[*31.03.2022* 13](#_Toc105023028)

[*05.04.2022* 13](#_Toc105023029)

[*07.04.2022* 13](#_Toc105023030)

[*11.04.2022* 14](#_Toc105023031)

[*21.04.2022* 14](#_Toc105023032)

[*26.04.2022* 14](#_Toc105023033)

[*28.04.2022* 14](#_Toc105023034)

[*03.05.2022* 14](#_Toc105023035)

[05.05.2022 14](#_Toc105023036)

[Arbeitstagebuch (Umsetzung) 15](#_Toc105023037)

[Einteilung 15](#_Toc105023038)

[09.05.2022 15](#_Toc105023039)

[10.05.2022 15](#_Toc105023040)

[12.05.2022 15](#_Toc105023041)

[13.05.2022 16](#_Toc105023042)

[16.05.2022 16](#_Toc105023043)

[18.03.2022 16](#_Toc105023044)

[19.05.2022 16](#_Toc105023045)

[20.05.2022 16](#_Toc105023046)

[23.05.2022 17](#_Toc105023047)

[24.05.2022 17](#_Toc105023048)

[25.05.2022 – 31.05.2022 17](#_Toc105023049)

[Kundentests 18](#_Toc105023050)

[Einführung 18](#_Toc105023051)

[Befragte Personen und Interviews 18](#_Toc105023052)

# 

# **Themenbeschreibung**

## Unsere Vorstellung

Bei dem Thema des Projekts handelt es sich um eine Desktop Anwendung, welche in der Lage sein soll, die Eingabe eines Textfelds oder eine Textdatei (txt, docx…) in eine Audiodatei (wav, mp3) umwandeln. Dieses Prinzip ist auch unter TTS(=Text to speech) bekannt. Ein Text-to-Speech-System (TTS) (oder Vorleseautomat) wandelt Fließtext in eine akustische Sprachausgabe um.

Die genaue Definition der Features, wie beispielsweise Sprachen, Dialekte, Dateiformate, muss noch mit dem Kunden abgeklärt werden, da man sich bei der Anforderungsanalyse für die Interviewtechnik entschied. Grundsätzlich erfolgt die Umwandlung mit Hilfe von APIs. Eine API (Application Programming Interface) ist ein Satz von Befehlen, Funktionen, Protokollen und Objekten, die Programmierer verwenden können, um eine Software zu erstellen oder mit einem externen System zu interagieren. Dadurch muss die Umwandlung nicht selbst implementiert werden, was Kenntnisse im Bereich Machine Learning erfordert hätte. Im Prinzip wird im Hintergrund eine API-Anfrage geschickt. Abhängig von der API werden die verschiedenen Parameter wie beispielsweise Sprache mitgegeben. Nachdem die API die Anfrage bearbeitet hat, sendet sie den umgewandelten Text in Form einer Audiodatei zurück. Mittels des UIs (User Interface, zu Deutsch: Benutzeroberflächer) hat der User die Möglichkeit die Datei durch das Drücken eines Buttons herunterzuladen. Im Vorfeld musste er Benutzer in der Benutzeroberfläche gewünschte Parameter für das Generieren der Audiodatei auswählen.

## Potenziell nutzbare APIs

* <https://www.ispeech.org>
* <https://cloud.google.com/text-to-speech?hl=de#section-12>
* <https://docs.microsoft.com/en-us/archive/msdn-magazine/2019/june/speech-text-to-speech-synthesis-in-net>
* <https://www.deepl.com/de/docs-api/>

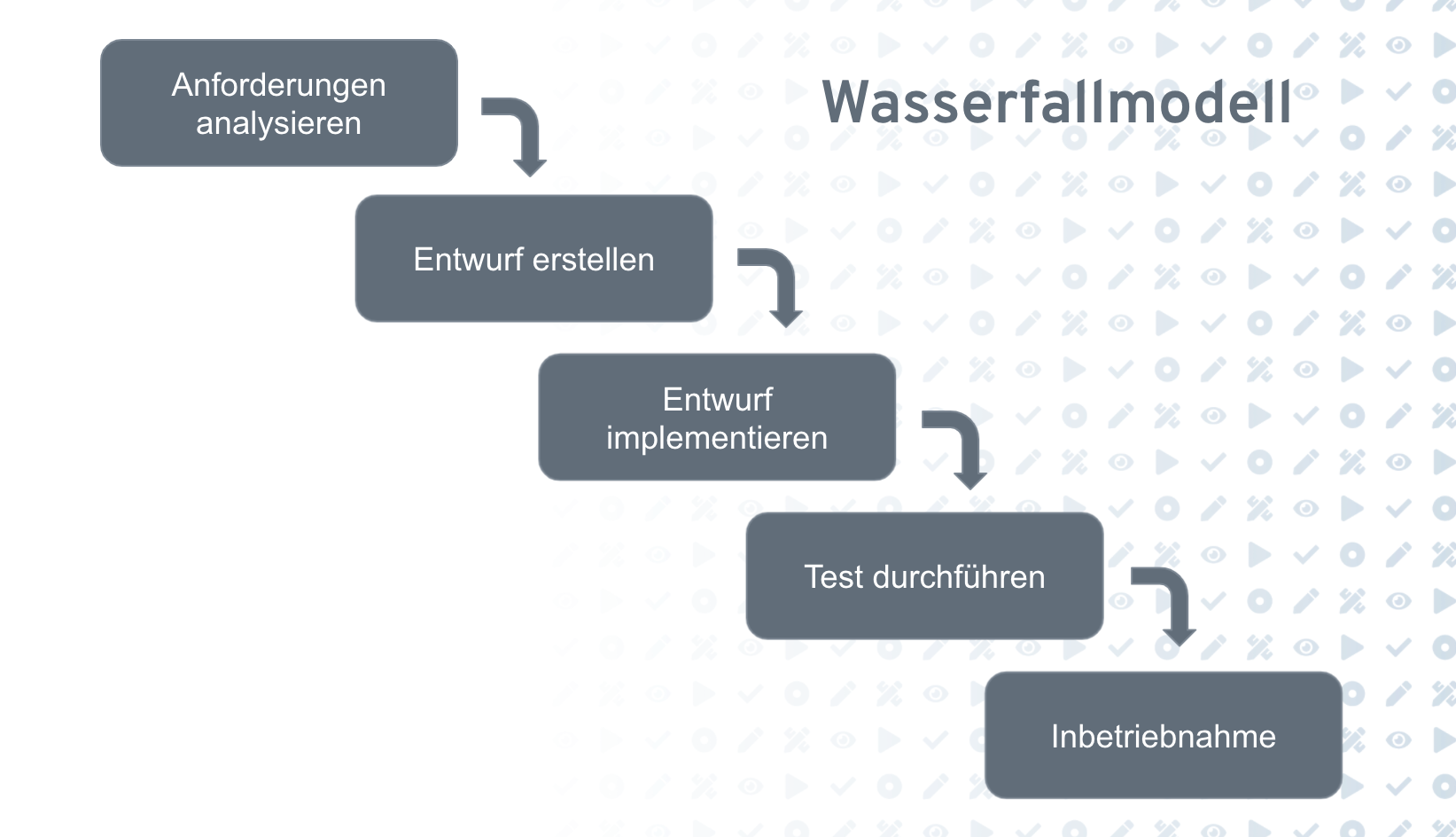
# 

# **Vorgehensmodell: Wasserfallmodell**

## *Erklärung*

Als Vorgehensmodell für unser Projekt haben wir das Wasserfallmodell gewählt. Die Wahl kam aufgrund der Vorteile des Wasserfallmodells zustande. Dieses Vorgehensmodell hat eine einfache Struktur, welche durch klar abgegrenzte Projektphasen verdeutlicht wird. Eine gute Dokumentation des Entwicklungsprozesses durch klar definierte Meilensteine ermöglicht zudem ein angenehmes und genaues Arbeiten.

Wie man in der Abbildung sehen kann, ist das Wasserfallmodell in fünf Schritte eingeteilt, die wir im Laufe des Projektes befolgen müssen. Zu Beginn werden die Anforderungen, die die Software haben soll, analysiert. Hierfür gibt es verschiedene Techniken. In unserem Fall wenden wir die Interviewtechnik für die Anforderungsanalyse an. Anschließend erstellt man einen Entwurf. Basierend auf den Anforderungen wird in diesem Schritt ein Lösungskonzept in Form von einem Entwurfsdokument entwickelt. Der dritte Schritt besteht darin, den Entwurf zu implementieren. Die Entwicklung des Produkts steht im Vordergrund und es werden Strukturen und Abläufe realisiert, um das Konzept umzusetzen. Anschließend werden in der vierten Phase Tests durchgeführt. Das Produkt wird hierbei zum ersten Mal im Ganzen getestet und überprüft, ob es die Anforderungen erfüllt. In der Inbetriebnahme, welche die letzte Phase ist wird das fertige Produkt in den Betrieb aufgenommen und anschließend an den Kunden ausgeliefert.



## *Anforderungsanalyse*

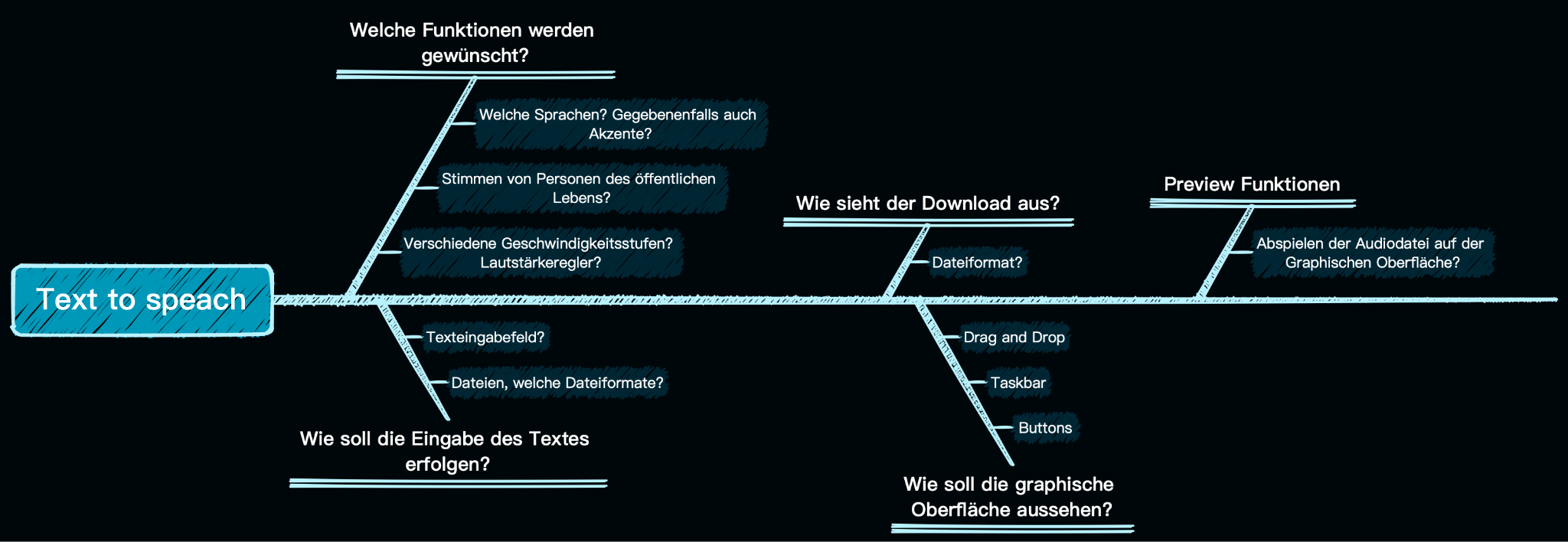
Jedes Softwareprojekt beginnt mit einer Analysephase, die eine Machbarkeitsstudie und eine Anforderungsdefinition umfasst. In der Machbarkeitsstudie wird zum Beispiel die Realisierbarkeit eingeschätzt. Anschließend kommt man zu einer detaillierten Anforderungsdefinition, bei der man Konzepte definiert, welche Funktionen und Eigenschaften das Softwareprodukt enthalten soll, um die Anforderungen zu erfüllen.

Für diesen Schritt der Planung haben wir die Interviewtechnik ausgewählt. Hierbei werden künftige Anwender in einem vorgegebenen und formalisierten Verfahren befragt. Mit dieser Technik sollte es möglich sein, verschiedene Gruppen zu bilden und schwer quantifizierbare, quantifizierbare und ergänzende Nutzenpotentiale abfragen. Es ist dabei unerlässlich für die Quantifizierung der Nutzenpotenziale alle betroffenen Bereiche einzubeziehen und aktiv mitzuwirken. Mit der Interviewmethode versucht man die konkrete Bedeutung der Produktattribute wie “einfach”, “innovativ”, “kontrollierbar” oder “eindrucksvoll” für ein mögliches Softwareprodukt zu verstehen.

Um die Interviewtechnik erfolgreich umzusetzen, haben wir wichtige Fragen und Punkte zusammengeschrieben, um diese dann in Form von einer Mindmap den künftigen Anwendern zu präsentieren. Hierfür haben wir und entschieden, das Tool XMind zu verwenden. Dieses stellt neben den kostenpflichtigen Pro Versionen auch eine kostenfreie Version mit vielen Eigenschaften zu Verfügung. Wir haben uns auch andere Mindmap Programme angeschaut, sind aber zum Entschluss gekommen, dass die Layouts und die Arbeitsoberfläche von XMind sich am besten für uns anbietet. Dieses professionelle Mindmap Tool sollte uns das Umsetzen der Interviewtechnik erleichtern und uns dabei helfen, diesen Schritt der Planung vernünftig zu visualisieren.

Anschließend haben wir noch ein Dokument verfasst, in dem wir die Anforderungsanalyse schriftlich dokumentiert haben.

Mindmap für das Interview:

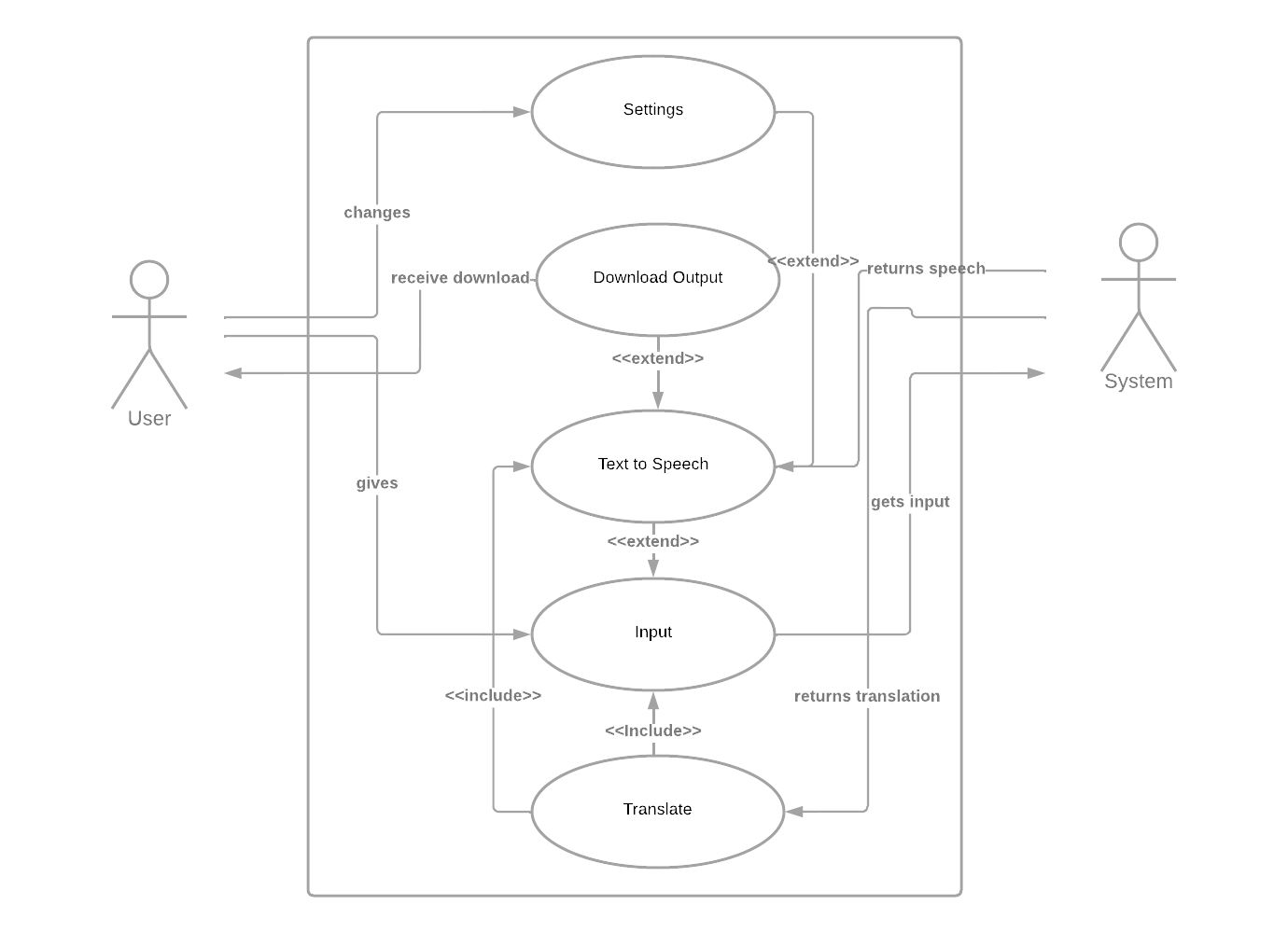


## *Entwurf erstellen*

Die Design-Phase des Wasserfallmodells dient der Ausbreitung eines konkreten Lösungskonzeptes auf Basis der Anforderungen, welche bei der Anforderungsanalyse gestellt wurden. Hierbei erstellt man einen Entwurf, um die Aufgaben und Strategien der Software umzusetzen. hierbei konzentriert man sich auf die Umsetzung der Software, Softwarearchitektur, Bibliotheken, Schnittstellen und der Graphischen Oberfläche.

Für das Erstellen des Entwurfs haben wir ein User Case Diagramm erstellt und ein UML Diagramm. Zudem haben wir eine Skizze der GUI skizziert und diese anschließend mit dem Tool Figma professioneller designend.

## *User Case Diagramm*

****

**UML Klassendiagramm**

Die Klasse User ist eine Hilfsklasse die es uns erlaubt, User als Objekte zu erstellen mit den notwendigen Attributen. In der User Klasse befinden sich sonst eigentlich nur Getter und Setter Methoden, mit denen wir verschiedene Informationen über User im Programm erhalten und abändern können.

Die Klasse UserList arbeitet mit verschiedenen Details von den Benutzern. Die liest die Benutzer aus der user.txt Datei formatiert aus, erstellt die User als Objekte und arbeitet dann mit der erstellten User-Liste. Es gibt die Methoden zum Hinzufügen und löschen von Benutzern aus der UserList und weitere Methoden zum Überprüfen, ob es einen benutzter bereits gibt oder ob die Passwörter gültig sind. Des Weiteren werden Passwörter von Usern die sich neu Anmelden mit einem Hashsalz gehasht, um die Passwörter zu schützen.

Startet man das Programm so öffnet sich die LoginPage. Über die LoginPage kann man sich einloggen, um mit dem Programm zu arbeiten, sich einen neuen Account anlegen, falls man noch keinen hat oder seinen Account löschen, wenn man diesen nicht mehr haben möchte.

Um einen Account zu erstellen, wird die AccountCreation Klasse verwendet. Diese enthält verschiedene Methoden, um zu schauen ob die Benutzereingaben richtig sind wie zum Beispiel passwordCheck() und accountCreatable(). Klickt man auf „confirm“ und alles wurde richtig eingegeben so sind die Accountdaten des neuen Accounts in users.txt eingetragen worden. Die Sprachen sind defaultmäßig initialisiert und werden anschließend beim Verwenden des Programms abgespeichert. Die zuletzt verwendete Sprache und Sprecher werden beispielsweise auch in der user.txt Datei abgespeichert und wenn man sich einloggt, ist automatisch die zuletzt verwendete Sprache eingestellt.

Um einen Account zu löschen, wird die AccountDelete Klasse verwendet. In dieser muss man den Benutzername korrekt eingeben und zweimal das Passwort. Auch hier wird überprüft, ob die Passwörter übereinstimmen und ob es den Benutzer überhaupt gibt. Kann der Benutzer erfolgreich gelöscht werden, so wird dieser von der user.txt Datei entfernt und kann sich anschließend nicht mehr anmelden.

Die Klasse Menue enthält nur Methoden, um zwischen den verschiedenen Formen zu wechseln. Sobald man sich eingeloggt hat, kann man auf das Menue zugreifen und von dort aus entschieden, was man gerade machen möchte. Man hat die Auswahl eine Textdatei zu einer Audiodatei zu konvertieren, einen eingegebenen Text zu einer Audiodatei zu konvertieren, eine Datei zu übersetzen oder einen Text zu übersetzen.

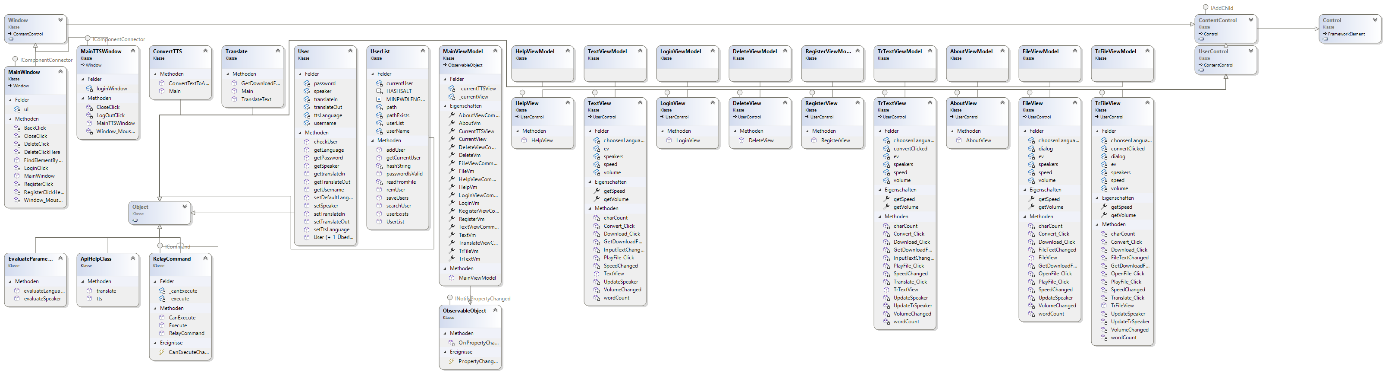
Für das TTS wurde eine Klasse namens InputTTS geschrieben. Wenn man im Menue auswählt, dass man ein File zu Audio konvertieren möchte, wird die InputTTS Klasse so aufgerufen, dass man über einen Button mit dem FileBrowserDialog eine Datei auswählen kann, die man konvertieren möchte. Will man nur eine Eingabe konvertieren so wird beim Anzeigen der InputTTS GUI einfach der Button versteckt, sodass man nicht unnötig zwei nahezu identische Klassen programmieren muss. Das gleiche gilt auch für die Klasse TranslateInput. Zurück zuer InputTTS Klasse: Diese stellt Methoden zum Überprüfen der Länge der Eingabe zur Verfügung, die mit Hilfe einer Hilfklasse VerifyInput überprüft werden. Desweiteren sind die ganzen Methoden der verschiedenen Buttonclicks ausprogrammiert, was passieren sollte wenn der Benutzer beispielsweise Convert klickt.

Die Klasse TranslateInput ist für das Übersetzen zuständig. Man kann Text eingeben, aus einer Datei auslesen, Sprache ändern und weiteres. Auch hier wird die Länge der Eingabe überprüft mithilfe der Klasse VerifyInput. Auch die Methoden der verscheidenen Buttonclicks für TranslateInput sind hier ausprogrammiert.

Um eine Schnittstelle zwischen den Klassen, mit denen der Benutzer interagiert und den Klassen, mit denen die APIs interagieren zu realisieren haben wir die ApiHelpClass programmiert. Die Methoden regeln noch einige Sachen wie zum Beispiel die Arbeit mit Dateien und anschließend werden die Klassen der APIs mit den richtigen Parametern aufgerufen.

Die Klassen Translate.cs und ConvertTTS.cs sind anschließend die Klassen, die mit den APIs arbeiten. Hier werden die APIs richtig aufgerufen, bekommen die Parameter übergeben, um die gewünschten Vorgänge zu erledigen und geben dem Programm das zurück, was zum Weiterarbeiten benötigt wird.

Die Klasse EvaluateParameters ist eine Hilfklasse, die bei dem TTS Konvertieren verwendet wird. Diese gibt je nach Sprachauswahl die Sprecher zurück, die für die ausgewählte Sprache zur Verfügung stehen.



Entwurf Implementieren

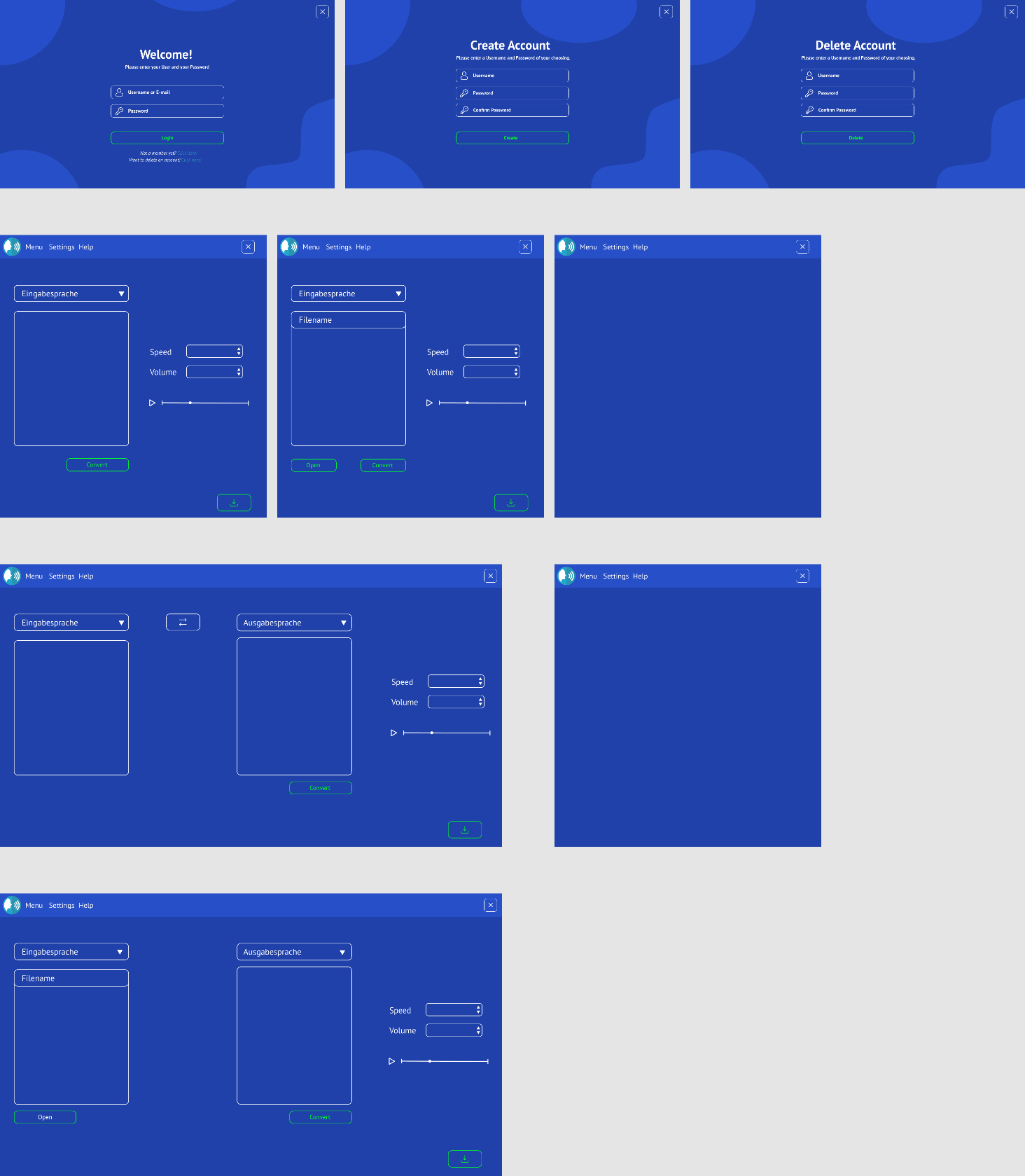
In der Implementierungsphase wird die Software für das Projekt in der gewünschten Programmiersprache programmiert. Einzelne Softwarekomponenten werden separat entwickelt, im Rahmen von Modultests überprüft und Schritt für Schritt entsteht dadurch das Gesamtprodukt.

Tests durchführen:

In Bezug auf die Tests haben wir uns darauf geeinigt, manuelle Tests durchzuführen, da wir im Gegensatz zu anderen Programmen keine Berechnungen oder ähnlich durchzuführen haben. Bei den Tests haben wir verschiedenste Konfigurationen in Kombination mit verschiedenen Text- oder Fileinputs probiert. Anschließend wurden die Ausgaben kontrolliert. Außerdem wurde der Login Bereich auf verschiedenste Wege getestet.

Inbetriebnahme:  
Zur Inbetriebnahme haben wir die Publish Funktion in Visual Studio 2022 verwendet, was bedeutet, dass nur 3 Dateien nur Installation nötig sind.

## *Diagramme*

* **Use Case Diagram:** <https://lucid.app/lucidchart/e304ba8a-882c-4a3d-9a51-385612e15085/edit?invitationId=inv_96c403cd-b505-4d08-a87e-38e0a1cbec22>
* <https://lucid.app/lucidchart/b395ef65-5d04-4d34-84e4-ca2e464ad7d6/edit?invitationId=inv_986bf351-b9da-429d-a255-17b3e36f4e80>
* **UI Entwurf:**

## *Quellen*

* <http://download.gsb.bund.de/BundesCIO/V-Modell_XT_Bund/V-Modell%20XT%20Bund-2.0-HTML/1e7a146fcccafec.html>
* <https://www.ionos.de/digitalguide/websites/web-entwicklung/wasserfallmodell/>
* <https://www.factro.de/blog/wasserfallmodell/>
* <https://xmind.works>

# **Dokumentation Anforderungsanalyse**

## *Erklärung*

Im Folgenden wurden kurz die minimalen Anforderungen des Projekts, sowie potenzielle Erweiterungen aufgezeigt. Die minimalen Anforderungen sind der Grundbestand des Projekts, wobei es das Ziel ist, abhängig vom Zeitlichen, so viele Erweiterungen wie möglich zu implementieren, um ein umfangreiches Produkt bieten zu können. Bei dem Thema des Projekts handelt es sich um eine Desktop Anwendung.

## *Minimale Anforderungen*

Grundsätzlich sollte der Benutzer die Möglichkeit haben, eine Textdatei oder eine Eingabe als Input zu übergeben. Dieser Input wird anschließend, wenn nötig formatiert und eine API-Anfrage getätigt. Als Basis haben wir uns für die von Amazon entwickelte API namens “Polly” entschieden. Diese bietet verschiedene Sprachen und dazu jeweils unterschiedliche Akzente oder Dialekte an. Als Rückgabe soll die API eine Audiodatei liefern. Das Dateiformat ist “.wav”. Wenn gewünscht wird diese Datei auch in “.mp3” konvertiert. Zum Schluss hat man die Möglichkeit eine Vorschau der Datei anzuhören und diese bei Zufriedenstellung herunterzuladen. Die heruntergeladene Datei soll im File Explorer im Ordner “Downloads” gespeichert werden.

## *Potenzielle Erweiterungen*

Eine mögliche Erweiterung für unsere Software wäre ein Übersetzer, welcher ebenfalls durch die Verwendung einer API realisiert werden kann. Im Prinzip soll der Benutzer die Möglichkeit haben eine Sprache für die Ausgabedatei festzulegen. Sind beispielsweise Ein- und Ausgabe Sprache verschieden, kann die Eingabe mithilfe des integrierten Übersetzers in die Zielsprache umgewandelt werden. Weiteres haben wir uns über einen Lautstärke- und ein Speech-Geschwindigkeitsregler Gedanken gemacht, wobei die Änderung der Lautstärke und der Geschwindigkeit nicht auf die herunterladbaren Dateien übertragen werden soll. In Bezug auf API-Usage haben wir zahlreiche Erweiterungsmöglichkeiten. Mittlerweile existieren viele ausgereifte APIs welche Stimmen von Personen des öffentlichen Lebens, Charakteren aus Videospielen und vielen weiteren Bereichen imitieren können. Daher steht uns eine Vielzahl von Möglichkeiten offen.

# **Fazit**

Zu Beginn kann man sagen, dass dieses Projekt allgemein sehr lehrreich war. Zunächst kamen wir zum ersten Mal mit dem praktischen Aspekt der Planung eines Projekts in Kontakt. Wie bereits erklärt haben wir das Wasserfallmodell als Vorgehensmodell verwendet. Bei einem nächsten Mal wäre es sicher vom Vorteil ein etwas dynamisches Vorgehensmodell zu verwenden. Uns wurde recht schnell bewusst, wie schnell sich Ideen und Anforderungen ändern können, wobei das Ausmaß in unserem Fall nicht besonders groß war. Daher musste die Planung ohne Fehler und nahezu perfekt über die Bühne gehen, was uns auch Großteils gelungen ist.   
Ein großer Faktor, warum wir uns für dieses Projekt entschieden haben, ist der, dass man mit APIs in Kontakt treten wollte. Vorerst war nur die Umwandlung von Text zu Sprache vorgesehen und die Übersetzungsfunktion nur eine Erweiterung. Aufgrund guter Zeiteinteilung und Planung konnten wir diese Funktion noch zusätzlich implementieren.   
Die zwei größten Herausforderungen waren zum einen die Einarbeitung in das Framework Xamarin Form zum UI Design. Dieses Bot nämlich viel mehr Möglichkeiten ein gutaussehendes und funktionierendes UI umzusetzen. Zweitens musste man sich mit der Verwendung von API in C# vertraut machen, wobei es bei der API für die Übersetzung sehr schwierig war, eine kostenlose und gut funktionierende zu finden. Die Dokumentation war zwar teilweise recht komplex, jedoch umfangreich sehr hilfreich. Damit konnten beide APIs erfolgreich verwendet werden.

# **Arbeitstagebuch (Planung)**

### *22.03.2022*

Am Donnerstag den 22.03. haben wir mit den Arbeiten an unserem Projekt begonnen. Nachdem wir die Gruppen zusammengestellt hatten, mussten wir uns Ideen für unser Projekt ausdenken. Wir konnten uns recht schnell dafür einigen, ein TTS Programm mit APIs zu schreiben. Nach der Einigung haben wir uns verschiedene Vorgehensmodelle angeschaut, welche für die Planung gut sein könnten. Am meisten hat uns dabei das Wasserfallmodell angesprochen. Die Vorteile dieses Modells wie eine einfache Struktur und eine gute Einteilung in verschiedene Planungsphasen haben zur Entscheidung beigetragen.

### *29.03.2022*

Die Arbeitsstunde am Dienstag den 29.03. hat uns dabei geholfen, einen Einblick in die Anforderungsanalyse zu bekommen. Diese ist die erste Phase der Planung des Wasserfallmodells. Wir haben uns in verschiedene Techniken der Anforderungsanalyse einlesen und haben uns dafür entschieden, die Interviewtechnik zu verwenden, um diesen Planungsschritt zu realisieren. Nach der Einigung haben wir uns die Interviewtechnik genauer angeschaut und damit begonnen, Fragen an die künftigen Anwender aufzuschreiben, welche wir anschließend visualisiert darstellen werden.

### *31.03.2022*

Am 31.03. haben wir mit der Visualisierung für das Interview begonnen. Hierfür haben wir das Mindmap Tool XMind ausgewählt, da uns dieses aufgrund verschiedener Features am meisten angesprochen hat. hiermit haben wir eine Mindmap zu den Interviewfragen erstellt, welches wir voraussichtlich in der nächsten führen werden.

### *05.04.2022*

Am 05. April haben wir das Interview mit Herr Mitterrutzner geführt. Dabei haben wir die Mindmap vorgetragen und die wichtigsten Punkte zu unserem Programm geklärt. Bei der Besprechung kamen wir zum Entschluss, dass wir ein Dokument vorbereiten mit den minimalen Anforderungen und mit zusätzlichen Features. Anschließend werden wir noch ein User Case Diagramm vorbereiten.

### *07.04.2022*

In diesen Arbeitsstunden haben wir uns mit dem Dokument für die minimalen Anforderungen und möglichen Erweiterungen befasst und diesen Unterpunkt der Planung vollendet. Anschließend haben wir uns noch mit dem User Case Diagramm und mit dem UML Klassendiagramm beschäftigt. Für das Realisieren der Diagramme haben wir uns das Online Tool Lucidchart ausgesucht. Mit diesem ist es möglich, verschiedene Form Bibliotheken für verschiedene Arten von Diagrammen zu erstellen.

### *11.04.2022*

Diese Arbeitsstunde gab uns die Möglichkeit, noch ein Gespräch über unser Projekt zu führen. Anschließend haben wir bereits damit begonnen, uns mit der Grafischen Oberfläche zu beschäftigen. Hierfür haben wir mit Paint eine Skizze angelegt, welche wir in Zukunft mit dem Tool Figma genauer und detaillierter designen werden.

### *21.04.2022*

In den heutigen Stunden haben wir uns konkreter mit der GUI beschäftigt. Wir haben noch die Skizze besprochen und zu ende gestellt, sodass wir mit dem Designen mit Figma beginnen konnten.

### *26.04.2022*

In dieser letzten Stunde der Planung gab es für uns noch genug zu tun. Wir mussten das Designen der GUI mit Figma fertig umsetzen, das UML Diagramm vervollständigen und auch noch die Dokumentation fertigschreiben. Um effizient weiterzuarbeiten und alles abschließen zu können haben wir uns die Arbeit aufgeteilt und so konnten wir die Planung des Projektes erfolgreich beenden.

### *28.04.2022*

Heute haben wir eine Einführung in GIT bekommen. GIT ist eine freie Software zur Verteilung von Softwareversionen. Wir haben gelernt ein GIT-Repository anzulegen, den Repository Status zu überprüfen, neue Dateien zur Versionsverwaltung hinzuzufügen, Änderungen via Commits zu bestätigen und noch weiteres. Die Software GIT sollte uns beim Programmieren weiterhalfen, um die Programmversionen miteinander zu teilen und gemeinsam am Projekt zu arbeiten.

### *03.05.2022*

In der heutigen Unterrichtsstunde haben wir uns weiter mit GIT beschäftigt. Um die Befehle in GIT zu verstehen und anschließend erfolgreich zu arbeiten haben wir online Übungen zu verschiedenen Aufgabestellungen gelöst. Diese haben und geholfen die Logik bei GIT besser zu verstehen und wir haben einen guten Einblick in die Arbeit mit GIT bekommen, was uns beim Projekt anschließend weiterhelfen sollte.

### 05.05.2022

Heute haben wir noch Zeit dafür gefunden, alles zu verfeinern und fertigzustellen. Michael hat noch das Design der GUI verbessert und verfeinert, Patrick hat sich damit beschäftigt, das Testdrehbuch zu schreiben. Für das Testdrehbuch haben wir uns an der Anforderungsanalsye orientiert, die wir durchgeführt haben. Durch die Anforderungen aus der Anforderungsanalyse kann man sich gut ausdenken, wie man die Anforderungen am besten testen kann.

# **Arbeitstagebuch (Umsetzung)**

## Einteilung

Für das Umsetzen des Programmes haben wir uns die Aufgaben folgendermaßen eingeteilt: Michael informiert sich über das GUI programmieren / darstellen mit XAML und wird anschließend die GUI des Projektes erstellen. Patrick beschäftigt sich damit, die richtigen APIs auszuwählen, die wir für das Projekt verwenden. Zudem war Patrick dafür zuständig, die Klasen translate.cs und ConvertTTS.cs zu programmieren. Die restlichen Klassen hat Alex programmiert. Er war für die Programmlogik zuständig.

### 09.05.2022

Heute war die erste Stunde in der uns effektiv mit dem Umsetzen des Projektes beschäftigt haben. Wir haben die EVAZ-Stunde dafür benutzt, uns ein Git-Repository einzurichten und das Projekt zu erstellen. Anschließend haben Patrick und Michael sich mit der GUI beschäftigt. Alex hat mit dem Erstellen der Klassen, variablen und Methoden anhand des UML Diagrammes begonnen. Anschließend hat er zu Hause damit begonnen, das Login-System umzusetzen. Um Benutzer und ihre Daten zu speichern, wird eine XML-Datei verwendet. Neue Benutzer werden in die XML-Datei eingetragen, zu Löschende Benutzer entfernt, und Benutzer, die sich einloggen werden in der XML-Datei gesucht, um die Eingabedaten zu verifizieren.

### 10.05.2022

In der heutigen Arbeitsstunde haben wir das, was wir am Vortag angefangen haben versucht abzuschließen. Das Login System funktioniert nun einwandfrei und bereits erstellte User können sich nun mit dem Benutzername und Passwort anmelden. Michael hat begonnen, mit XAML die GUI zu erstellen während Patrick damit begonnen hat, sich mit APIs zu beschäftigen.

11.05.2022

Heute hat Michael an der GUI weitergearbeitet in dem er sich Tutorials zu XAML angeschaut hat und Patrick hat sich einen Code zum Arbeiten mit der API durchgelesen. Dabei hat Patrick verstanden, wie man die Text to Speech API richtig aufruft und verwendet. Der Code muss noch etwas umgeschrieben werden, um ihn an unser Programm anzupassen.

### 12.05.2022

In den heutigen Stunden haben Michael und Alex an ihren Teilen weitergearbeitet. Patrick war abwesend. Michael hat sich mit XAML beschäftigt und ein Icon für das Programm in die XAML GUI importiert. Alex hat sich damit auseinandergesetzt, den eingegebenen Text nach der Länge zu überprüfen. Dies wird anhand eines Wörterlimits und eines Zeichenlimits geregelt, sodass die API beim Arbeiten nicht überfordert wird. Zu Hause hat Alex noch das System zum Auslesen aus einer Datei mit verschiedenen Abfragen zur textlänge und weiterem programmiert.

### 13.05.2022

Am heutigen Nachmittag hat Patrick uns erste Resultate von der API liefern können. Hierbei hat er mit Alex die Installationen der NuGet Pakete durchgeführt, die wir für das Projekt benötigen. Nachdem alles erfolgreich installiert war, haben Patrick und Alex zusammen angefangen ein Paar Tests für die API zu machen, um zu schauen, ob alles wie geplant funktioniert. Michael war weiterhin mit XAML beschäftigt.

### 16.05.2022

Weiterarbeiten an dem API-Code und an der GUI.

### 18.03.2022

Heute waren Alex und Patrick abwesend. Michael hat an der GUI weitergearbeitet. Er hat sich vor allem mit der GUI für das Login Fenster beschäftigt und die Buttons und Labels dafür erstellt. Alex hat zu Hause etwas am Klassendiagram weitergeschrieben, da wir zusätzliche Funktionen eingebaut haben, die erst beim Programmieren dazugekommen sind. Zudem hat er noch einige Feinheiten in den verschiedenen Klassen abgeändert und verbessert.

### 19.05.2022

Heute hat Patrick sich mit der Textübersetzung auseinandergesetzt. Da die meisten APIs kostenpflichtig sind, war es schwierig eine zu finden. Zu Hause hat er es geschafft, eine funktionierende und kostenfreie API zu finden die funktioniert. Alex hat sich damit befasst, den Code noch zu verfeinern und zu schauen, was man noch verbessern oder ändern könnte. Es gab noch den Bug, dass beim Logout das Passwort und der Benutzername noch in den Textboxen standen. Zu Hause hat Alex noch die Funktion des Play-Buttons programmiert. Hierfür braucht man ein NuGet packet für die WMP Bibliothek.

### 20.05.2022

In der heutigen Doppelstunde hatten wir noch einiges vor. Wir mussten noch die Translate API mit dem Programm zusammenfügen und die GUI war auch noch nicht ganz fertig. Wir haben uns die Arbeit so eingeteilt, dass Michael die GUI verfeinert, Patrick mit der API weiterarbeitet und Alex die Programmlogik zum Erhalten der Parameter für die Translate API programmiert. Dabei mussten Alex verstehen, welche Parameter die API braucht, wie man diese vom Benutzer richtig übergeben bekommt und anschließend die API ohne Fehler aufgerufen werden kann.

### 23.05.2022

Heute wurde uns noch eine Informatikstunde zur Verfügung gestellt, um am Projekt weiterzuarbeiten. In dieser Stunde haben wir versucht, den Code von Patrick für die API mit dem Code von Alex (Programmlogik, abrufen der Parameter vom Benutzer und weiteres) zu verbinden, sodass die API richtig aufgerufen wird und den Text erfolgreich übersetzen kann.

### 24.05.2022

Die heutige TP-Stunde durften wir noch am Projekt weiterarbeiten. Alex hat die Stunde genutzt, das Tagebuch und die Dokumentation weiterzuführen und noch zusätzliche Punkte zu bearbeiten. Patrick musste noch ein bisschen mit der Zusammenarbeit der API mit dem Programm weiterarbeiten während Michael damit beschäftigt war, die GUI mit dem Code zu verbinden. Am Nachmittag hat Alex noch einige Tests durchgeführt und ist dabei auf ein paar Fehler gestoßen, die behoben werden müssen. Es musste noch die Arbeit mit Textdateien umgeändert werden, dass das Programm mit relativen Pfaden arbeiten kann. Zudem hat Alex noch das Feature hinzugefügt, dass bei jedem Benutzer die zuletzt verwendete Sprache und der Sprecher beim TTS und die Input Sprache und Output Sprache beim Übersetzer abgespeichert werden. Es gab noch einen kleinen Bug, dass beim Sprache wechseln vom TTS die Sprecher der vorherigen Sprache noch zur Verfügung standen. Dies ließ sich leicht beheben.

### 25.05.2022 – 31.05.2022

In den letzten Tagen hatten wir die Chance noch einiges zu verbessern was zu verbessern war. Die Dokumentation musste noch fertiggeschrieben werden, der Text für die About Seite im Programm musste geschrieben werden und weiteres. Auch das UML Diagramm musste noch richtiggestellt werden, da neue Klassen und Funktionen hinzugekommen sind. Michael und Patrick brauchten noch etwas Zeit, um die GUI mit dem Programm zusammenzusetzen während Alex sich um den Rest gekümmert hat und noch gewisse Bugs behoben hat. Am Tag der Abgabe haben wir gemeinsam noch einige Tests durchgeführt. Die XAML GUI hat nämlich bei gewissen Bedingungen nicht richtig mit dem Programm zusammengearbeitet. Dies musste unbedingt noch behoben werden, da es mit der Winform GUI schon funktioniert hat.

# Kundentests

### Einführung

Um zu bestätigen, dass das Produkt gut funktioniert, keine Bugs hat und die Kunden mit dem Service zufrieden sind, hat Alex noch einigen Personen aus verschiedenen Altersgruppen das Programm zum Testen gegeben. Unten sind die Personen aufgelistet und einige Aussagen mitgeschrieben, die die Meinungen einiger der Befragten widerspiegeln sollen. Bei den Aussagen / Interviews habe ich jeweils eine Person pro Altersgruppe gewählt um einen

### Befragte Personen und Interviews

Großmutter, Großvater, Onkel, Cousine, Freundin, Vater;

Großmutter: „Trotz, dass ich keine guten technischen Kenntnisse haben und nur selten mein Handy / iPad verwende, bin ich sehr gut mit dem Programm zurechtgekommen. Ich denke, dass die Auswahl der implementierten Sprachen gut ist, da das Programm mit den wichtigsten Sprachen gut arbeiten kann. Ich kann die Gruppe für die gute Arbeit nur loben und hoffe, dass das Projekt auch bei den Professoren gut ankommt.“

Vater: „Das Programm funktioniert einwandfrei. Ich finde es sehr gut und wichtig, dass die Passwörter nicht als Passwort in der Benutzer Datei abgespeichert werden, sondern dass sie verschlüsselt werden. Aus der Sicht der Sicherheit also auf jedem Fall Positiv. Zudem habe ich die Anforderungsanalyse durchgelesen und feststellen können, dass das Programm die gestellten Anforderungen erfüllt.“

Freundin: „Ich finde die Programmidee sehr gut und auch die Umsetzung. Das Tool kann ich zum Beispiel sehr gut verwenden, wenn ich die Aussprache eines Worts auf Englisch für eine Prüfung brauch. Und ein Übersetzer ist sowieso immer sehr wichtig. Mir wurde zudem gesagt, dass DeepL zum Übersetzen verwendet wird. Ich verwende immer das online Tool von DeepL und finde auch dieses sehr gut, weshalb ich die Entscheidung der DeepL API sehr vorteilhaft finde.“