Translation app

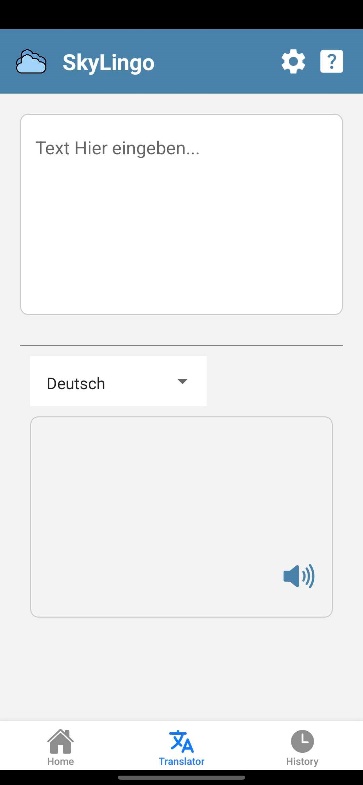
## SkyLingo

### Version 1.0.0, Datum: 21. Februar 2025

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Text, Elektronik, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abstract (Kurzbeschreibung)

Das Projekt befasst sich mit einer Translator-App, welche sehr nahe an den klassischen Modellen, wie Google-Translate angelehnt ist. Wir haben dafür die Open-Source-API LibreTranslate verwendet. In jener kann man spezifisch Sprachsets herunterladen. In Form von einem Textfeld kann man den gewünschten Satz oder das gewünschte Wort eingeben, mittels Dropdowns die Zielsprache auswählen und es wird mit einer kleinen Verzögerung übersetzt. Als Zusatz kann man mit dem Lautsprecher-Button auch noch eine Audioversion des übersetzten Teils bekommen. Jener funktioniert aber leider nicht auf alle Sprachen, da dies von Expo Speech bereitgestellt wird. Anschliessend kann man die Übersetzung im Historytab nachschauen und durch die Detailansicht auch eingehender auf grössere Übersetzungen eingehen.

user stories

* Als User möchte ich Wörter und Sätze in anderen Sprachen übersetzen, damit ich deren Definition in anderen Sprachen kenne.
* Als User möchte ich in mehreren Sprachen übersetzen können, damit ich in verschiedenen Sprachen die Definition kenne.
* Als User möchte ich meine, Verlauf der Übersetzung sehen, damit ich sehen kann, was ich früher übersetzt habe.

**Optional:**

* Als User möchte ich mit der Mikrofonfunktion direkt Texte übersetzen können, damit ich nichts tippen muss.
* Als User kann ich die Daten aus der History einsehen, damit ich die Übersetzung im Detail anschauen kann
* Als User kann ich die Daten in der History löschen, damit ich wieder die Übersicht habe.

Mockups

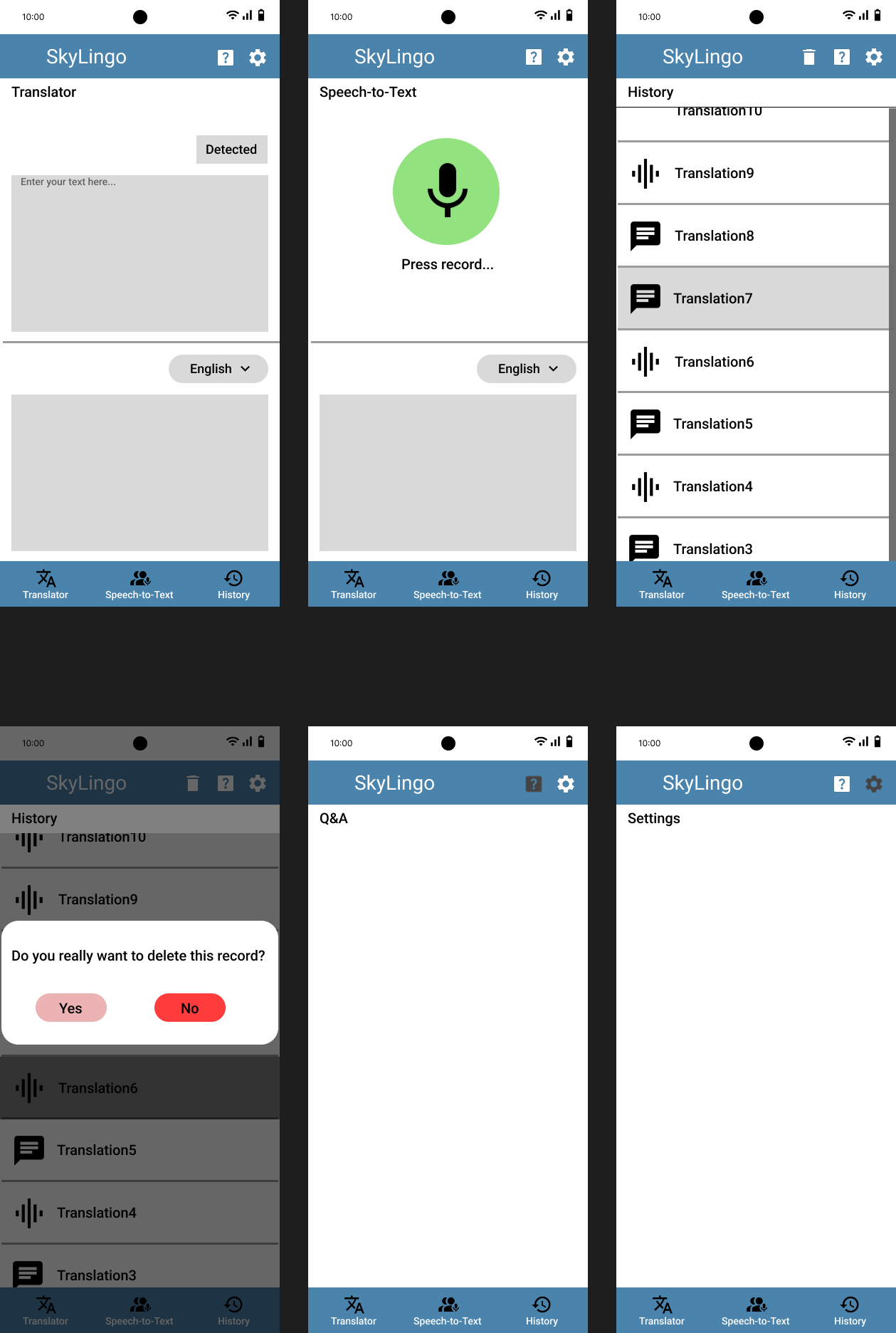
In den folgenden Bildern erkennt man, wie der Verlauf von den ersten Ideen in Form von Scribbles in ein tatsächliches Mockup migriert wurden.

Während im Scribble erstmals die Funktionen wie Navigation im Zentrum standen kamen im Mockup spezifischer die Input-/Output-Felder und Farben zur Geltung und die App nahm Form an.

Ein Bild, das Text, Papier, Notebook Notizbuch, Buch enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Screens wurden auf drei Hauptseiten aufgeteilt. Ein normaler Translator, ein Speech-to-Text-to-Speech-Translator, welcher wir auf Grund von keiner passenden API vorläufig auf Eis gelegt haben und einem History-Verlauf, um vergangene Aufrufe noch einmal anzuschauen.



In der Umsetzung gab es dann noch einige Abweichungen zu den Mockups. Unteranderem farblich wie auch funktional, sodass wir eine Homepage als Startseite für den User eingerichtet haben und die Funktionen des Text-to-Speech mit dem Translator zusammengenommen haben.

Technische Realisierung

#### Welche API haben wir benutzt und wie haben wir diese eingesetzt?

#### Wir haben die Open-Source-Übersetzungs-API *LibreTranslate* verwendet. Diese API ist grundsätzlich kostenlos nutzbar, allerdings fallen Kosten an, wenn man sie über einen gehosteten Dienst verwenden möchte. Um uneingeschränkt und ohne zusätzliche Kosten Übersetzungsanfragen senden zu können, haben wir LibreTranslate in einem lokalen Docker-Container gehostet. Dadurch können wir beliebig viele Anfragen stellen. Die API liefert ihre Antworten als JSON-String und zeichnet sich durch hohe Geschwindigkeit und Zuverlässigkeit aus.

#### Wie haben wir die Daten gespeichert?

Alle durchgeführten Übersetzungen werden in einer Datenbank in der Tabelle *Verlauf* gespeichert. Dafür nutzen wir *SQLite*, eine leichtgewichtige, einfach zu bedienende relationale Datenbank, die sich besonders für kleinere Anwendungen eignet. Der Datenbankserver wird mit *Node.js* gestartet, und die SQL-Statements werden über *Axios* an die Datenbank gesendet. Bei jedem Serverstart wird die Tabelle automatisch neu erstellt, falls sie noch nicht existiert.

Testing

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 1 |
| Anforderungen | Als User möchte ich Wörter und Sätze in anderen Sprachen übersetzen, damit ich deren Definition in anderen Sprachen kenne. |
| Vorbedingungen | Man muss sich auf der Übersetzungsseite befinden und eine Verbindung zum LibreTranslate API haben |
| Ablauf | * Auf die Obere Inputbox klicken * Ein Wort oder Satz eingeben |
| Erwartetes Resultat | Der eingegebene Text sollte übersetzt werden und in der unteren Output box ersichtlich sein. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 2 |
| Anforderungen | Als User möchte ich meinen Verlauf der Übersetzung sehen, damit ich sehen kann, was ich früher übersetzt habe. |
| Vorbedingungen | Eine Datenbank Verbindung mit der languageapp.db muss vorhanden sein. Eine Übersetzung muss gemacht worden sein. |
| Ablauf | * Auf den Verlauf in der Bottom Navigation klicken |
| Erwartetes Resultat | Es sollte der Verlauf der Übersetzungen gezeigt werden |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 3 |
| Anforderungen | Als User möchte ich in mehreren Sprachen übersetzen können, damit ich in verschiedenen Sprachen die Definition kenne. |
| Vorbedingungen | Verbindung mit der API und ein Dropdownmenu der Sprachen |
| Ablauf | * Sprache im Dropdownmenu auswählen |
| Erwartetes Resultat | Es sollte eine andere Sprache ausgewählt sein und der Text sollte übersetzt sein |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 4 |
| Anforderungen | Als User möchte ich mit der Mikrofonfunktion direkt Texte übersetzen können, damit ich nichts tippen muss. |
| Vorbedingungen | * Funktionierendes Mikrofon |
| Ablauf | Auf das Mikrofon Icon klicken und einen Satz oder Wort ins Mikrofon sprechen |
| Erwartetes Resultat | Es sollte den gesprochenen Satz und das Wort Übersetzen |

Testprotokoll

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen |
| ST-01 | Noa Henne 21.02.2025 | Ja | Hat Funktioniert. TTS klingt nicht wie erwartet. |
| ST-02 | Noa Henne  21.02.2025 | Ja | Falsche Sprache bei Source Language |
| ST-03 | Noa Henne  21.02.2025 | Ja | Hat Geklappt |
| ST-04 | Noa Henne  21.02.2025 | Nein | Noch nicht implementiert |
| ST-01 | Eray da Silva Burmaci | Ja | Tastatur ist nicht ideal und verwirrt. |
| ST-02 | Eray da Silva Burmaci | Ja | Hat Funktioniert |
| ST-03 | Eray da Silva Burmaci | Ja | Hat Funktioniert |
| ST-04 | Eray da Silva Burmaci | Nein | Noch nicht implementiert |
|  |  |  |  |

fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.

#### Fazit Adrian Orlamünde

Ich bin sehr zufrieden mit unserem Projekt und mit dem, was wir geleistet haben. Besonders gut fand ich, wie wir uns gegenseitig unterstützt haben, wenn jemand Fragen hatte, und wie wir gemeinsam Probleme lösen konnten. Leider konnten wir die optionalen User Storys, die wir geplant hatten, nicht abschließen. Aber insgesamt bin ich sehr zufrieden. Während des ÜK konnte ich viel über React lernen und darüber, wie man eine einfache, moderne Mobile App erstellt. Leider fehlte uns die nötige Zeit, um alle gewünschten Features und das geplante Design vollständig zu implementieren.

#### Fazit Larissa Richvalsky

Gut:

* Haben mit LibreTranslate eine tolle Übersetzungs-API gefunden => sind daher schnell vorangekommen und konnten noch viel an optionalen Features ausprobieren
* Gute Teamarbeit 😊

Schlecht:

* Haben leider keine API für Speech to Text (in unserem finanziellen Rahmen) gefunden

Ich bin im Allgemeinen sehr zufrieden mit unserem Endergebnis. Auch wenn ich es grossartig gefunden hätte, wenn etwas mehr Zeit zur Verfügung gestanden wäre, vor allem am Anfang, um uns in die Basics einzulesen und üben. Meinerseits brachte ich keine direkte Vorerfahrung mit Mobile Applikationen programmieren mit und hätte dieses Wissen gut gebrauchen können.

Angekoppelt an oben genannt, habe ich daher sehr viel gelernt und es hat eine Menge Spass gemacht dies in Form eines Projektes anzuwenden.

Momentan fehlen uns noch die Refresh-Funktion nach dem Löschen der History, die Möglichkeit einzelne Historyelemente zu löschen, Speech-To-Text-Funktion, die Einstellungsseite und die Q&A Seite.