



MODULHANDBUCH

Front-End



**Mohamed Benchellal, 924283, mohamed.benchellal@stud.fra-uas.de
William Iradukunda, 1222667, iradukun@stud.fra-uas.de**

ABGABEDATUM: 15.09.2023

SS 2023

Inhalt

1. Einführung.....	2
1.1 Produktbeschreibung.....	2
1.2 Werkzeuge.....	2
2. Zeitplan.....	4
3. Herausforderungen.....	5
4. Benutzeroberfläche und Anwendung.....	5
5. Statutarische Erklärung.....	12

1. Einführung

1.1 Produktbeschreibung

Als Team konzentrierten wir uns auf ein benutzerfreundliches Front-End-Design, die ein Tools repräsentiert, dass durchs Hochladen von Dateien ein fertiges Modulhandbuch ausgibt. In unserem Projekt haben wir einen Hauptnutzer: Der Verantwortliche für die Erstellung der Modulhandbücher. Der Verantwortliche hat die Möglichkeit, über die Anwendung 6 Templates in Word Format und 1 Excel Datei hochzuladen und ein fertiges Modulhandbuch mit den Formaten und Daten der Templates und Excel-Datei wird generiert.

1.2 Werkzeuge

Die Wahl, in welcher Sprache wir unser Projekt realisieren wollten, war sehr schnell getroffen. Auch auf die IDE konnten wir uns sehr schnell einigen. Wir haben unsere Anwendung mit React Framework mit Javascript realisiert, da wir bereits über einige Erfahrung verfügten. **React** ist eine JavaScript-Bibliothek zur Erstellung von Benutzeroberflächen. Die Bibliothek erfordert die Eingaben eines Benutzers, wodurch andere Prozesse reibungslos gesteuert werden. Jede React-Webanwendung besteht aus wiederverwendbaren Komponenten, die Teile der Benutzeroberfläche bilden. Warum react? Mit React Js brauchen Entwickler weniger Zeit, um eine interaktive Webseite zu entwickeln und bieten viel weniger Möglichkeiten für Fehlerbehebung und Bugs zu fixen. Außerdem ist React viel leichter zu lernen. Mit React spart man viel Zeit benötigt man viel weniger Codierung, weil man viel Sache nicht wiederholen muss und ist es somit effizienter. Zum Beispiel das virtuelle DOM von (Dokumentobjektmodell) aktualisiert nur Komponenten, die aktualisiert werden müssen. In den meisten Fällen wird für die Erstellung von React-App die Syntax JSX (JavaScript XML) verwendet, eine Syntaxerweiterung von JavaScript. Damit kann man die Logik von JavaScript und die Logik der Benutzeroberfläche leichter kombinieren. JavaScript ist eine Programmiersprache, die es ermöglicht, komplexe Funktionen in Web-Design zu implementieren

Dann haben wir uns auch für **Visual Studio Code als IDE** entschieden, weil diese IDE uns grundlegende Funktionen bot, die wir für unser Projekt für notwendig hielten. Visual Studio Code ist eine integrierte Entwicklungsumgebung von Microsoft. Die Entwicklungsumgebung kann von der Homepage www.code.visualstudio.com heruntergeladen werden. Die Community-Version ist kostenlos. Vor der Installation ist es notwendig, ein eigenständiges React Bibliothek herunterzuladen.

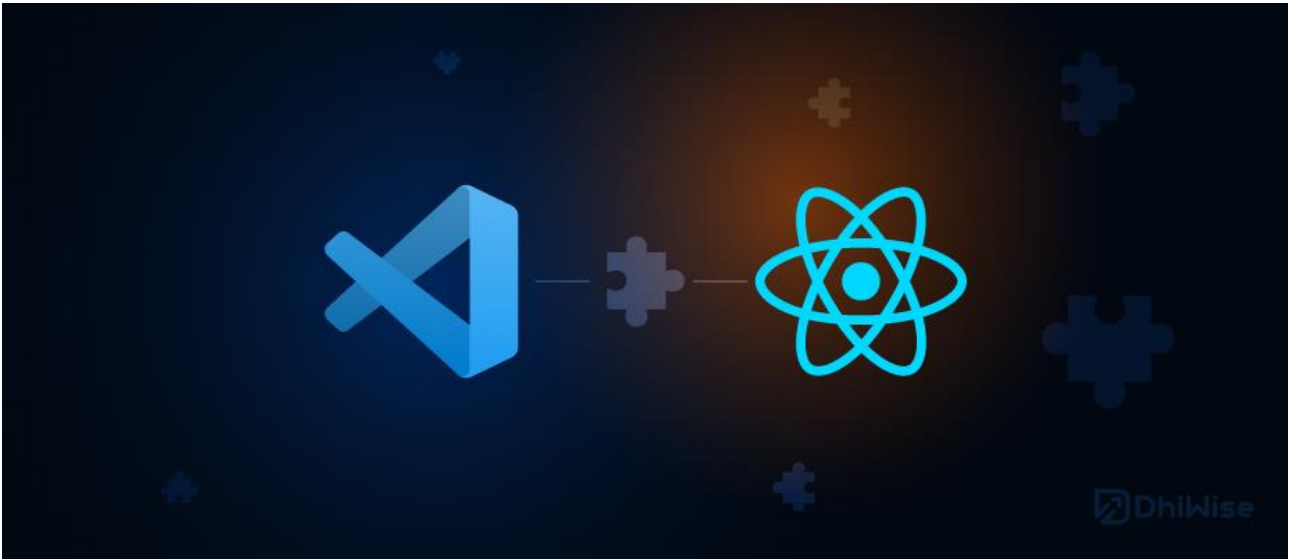


Bild.1 : Bild von Visual Studio Code und React

Copyrighted by Microsoft

Die Integration von **Sharepoint**, einem System von Microsoft, das Speicherplatz für Dateien und Versionskontrollsoftware bietet, um unseren geschriebenen Code und Dokumente zwischen unseren Teammitgliedern zu teilen, war notwendig, um unsere täglichen/wöchentlichen Scrums zu erfüllen.



Bild. 2: Bild von SharePoint

Copyrighted by Microsoft

GitLab ist eine Webanwendung zur Versionsverwaltung für Softwareprojekte auf Git-Basis. In unserem Projekt wurde ein Intern GitLab für Frankfurt University of Applied Sciences(fb2) verwendet. Gitlab erleichterte uns, unser Projekt im Team leichter zu managen. Mit den Funktionen der GitLab war es leichter unseren Code zu verwalten, Probleme zu lösen und die tägliche Kommunikation in Bezug auf Code war viel angenehmer.



Bild. 3: Bild von GitLab

2. Zeitplan

In diesem Abschnitt wird zunächst der Beginn des Projekts beschrieben, bei dem wir eine grobe Vorstellung davon hatten, wie das Endprodukt aussehen sollte. Am Ende des Abschnitts wird deutlich, wie wir unsere Zusammenarbeit gemeistert und ein voll funktionsfähiges Projekt realisiert haben. Alles begann mit ein paar Treffen, bei denen wir uns als Team zusammensetzten und über mögliche Ansätze für das Projekt nachdachten. Zuerst mussten wir entscheiden, wie unser Front-End aussehen sollte.

Nach weiteren Diskussionen und Überlegungen hatten wir unsere Ideen dokumentiert und hatten eine viel klarere Vorstellung von unserer Oberfläche aussehen sollte.

Natürlich gab es Anforderungen, die durch die Projektbeschreibung vorgegeben waren, aber diese mussten noch detaillierter dargestellt werden. Es gab viele Anforderungen, die wir hinzugefügt haben, um die Anwendung für die Benutzer attraktiver zu machen. Mit all diesen Anforderungen, die von den Teammitgliedern vorgeschlagen wurden und die für das Funktionieren und die Skalierbarkeit der Oberfläche notwendig waren, hatten wir eine Menge zu planen. Zunächst unterschieden wir zwischen Anforderungen, die absolut notwendig waren, und solchen, die es nicht waren. Dann haben wir die Anforderungen in eine Rangfolge von sehr wichtig bis weniger wichtig gebracht. Die Kriterien für diese Einstufung waren in erster Linie die Bedeutung der Anforderung für mögliche Interessengruppen, die Komplexität der Anforderung und unsere Bereitschaft, die Umsetzung der Anforderung durchzuführen. Nachdem wir eine Rangliste aller vorgeschlagenen Anforderungen erstellt hatten, war es viel einfacher, einige herauszunehmen und sich auf die wichtigen Anforderungen zu konzentrieren. An diesem Punkt entschieden wir uns für einen agilen Ansatz. Wir begannen mit der Erstellung einer Vorlage mit den dazugehörigen Seiten der Oberfläche. Wir teilten uns die Aufgabenbereiche und legten Deadlines an. Da wir immer montags den Termin mit der ganzen Projektgruppe hatten, hielten wir es für sinnvoll, danach kurz über unsere Aufgaben zu diskutieren, analysieren und Verbesserungen anzusprechen und neue Ideen innerhalb der Gruppe aufzuteilen damit jeder von uns daran arbeiten konnte. Als die Oberfläche fertig war, trafen wir uns regelmäßig für das Testen der Oberfläche, dafür erstellten wir eine einfache API um zu testen ob die Verbindung zwischen frontend und backend besteht. Die zugewiesenen Aufgaben haben sich als sehr nützlich erwiesen, denn auf diese Weise konnte die Arbeitslast gerecht verteilt werden. Hätten wir diese Rollen nicht eingeführt, wäre es höchstwahrscheinlich zu Konflikten gekommen, wenn wir an der gleichen Implementierung gearbeitet hätten.

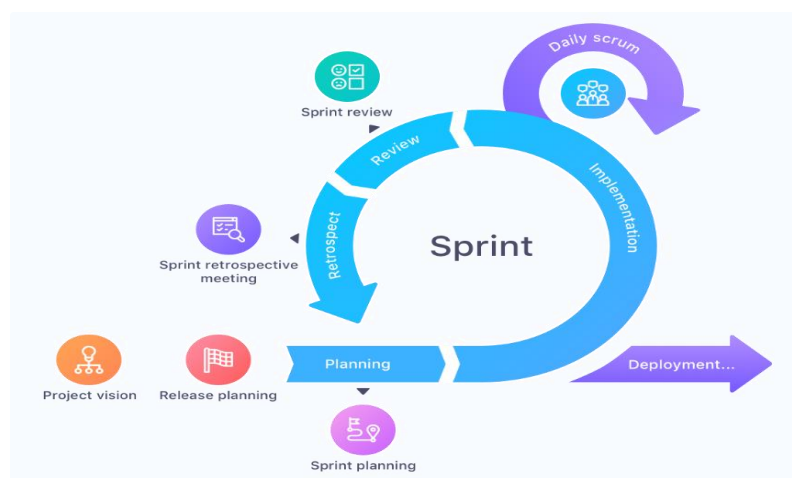


Bild.4: Bild von Scrum

3. Herausforderungen

Gute Organisation und Zusammenarbeit sind der Schlüssel zu einem erfolgreichen Projekt. Als Team haben wir viel Wert auf diese Aspekte gelegt, aber es gab dennoch einige, denen wir uns während des Projekts stellen mussten. In diesem Abschnitt gehen wir näher auf diese Herausforderungen ein und erläutern, wie wir sie gemeistert haben. Die erste Herausforderung, der wir uns stellen mussten, war die Entscheidung der Programmiersprachen bzw. Framework und Werkzeuge. Jedes Teammitglied hatte bereits einige Erfahrung mit bestimmten Programmiersprachen und Werkzeuge, und wir erkannten schnell, dass diese Entscheidung nicht einfach sein würde. Also begannen wir alle vorgeschlagenen Programmiersprachen, Frameworks und Werkzeuge zu vergleichen. Das Aufschreiben der Vor- und Nachteile half uns, eine gezieltere Diskussion zu führen, und zeigte uns, welche Sprachen und Werkzeuge die Anforderungen unserer Anwendung am besten erfüllen würde.

Dann war es an der Zeit, den einzelnen Teammitgliedern Rollen zuzuweisen, so dass jedes Teammitglied für einen Teil der Anwendung verantwortlich war, wie es im Zeitplan vorgesehen war. Normalerweise wäre dies kein Problem, aber in unserem Fall hatten wir wenig Erfahrung mit dem React Framework.

Aus diesem Grund war es schwierig, sich für eine Rolle zu entscheiden, auch wenn wir wussten, dass dieser Schritt für eine effektive Entwicklungsphase wichtig war. Nach einigen Diskussionen gelang es uns, das Team auf die entsprechenden Rollen aufzuteilen. Auch wenn jeder eine Rolle hatte, agierten wir als Team und halfen uns gegenseitig.

Die letzte Herausforderung kam, als wir mit der Entwicklung begannen und dann feststellten, dass wir eine Änderung in der Oberfläche brauchten. Wie bereits erwähnt, haben wir viel Zeit in das anfängliche Design investiert. Auf diese Weise konnten wir viele Unannehmlichkeiten vermeiden, die während der Entwicklungsphase auftreten konnten, aber offenbar haben wir einen Punkt übersehen. Wir mussten nun einen Teil des Designs ändern. Glücklicherweise geschah dies alles in den frühen Phasen der Entwicklung. Und es war auch von Vorteil, dass wir eine gute Struktur in unserem Projekt hatten. Schließlich konnten wir das Problem lösen, und wir erkannten, dass es viel schlimmer hätte kommen können, wenn wir nicht die nötige Zeit in die Strukturierung unseres Projekts investiert hätten.

In solchen Phasen haben wir eine gute Organisation und Zusammenarbeit zwischen den Teammitgliedern erlebt.

4. Benutzeroberfläche und Anwendung

Die fertige Benutzeroberfläche sieht folgermaßen aus:

- **Die Startseite**

Auf der Startseite erscheint folgende Oberfläche. Ein Klick auf “Excel-Datei einlesen” gelangen Sie auf die nächste Seite in der Sie ein Modulhandbuch generieren können.

Modulhandbuch



Excel-Datei einlesen

Hier können Sie ihre Excel-Datei einlesen

Bild. 5: Screenshot von Startseite

- **Die Anwendungsseite**

Nach dem Klick in der Startseite gelangen Sie zu dieser Seite. Sie haben hier die Möglichkeit auf der rechten Seite 6 Templates von Ihrem lokalen Rechner hochzuladen. Auf der linken Seite können Sie eine lokale Excel-Datei hochladen. Im “Verzeichnis für Ausgabe auswählen” können Sie einen lokalen Speicherort auswählen in der das generierte Modulhandbuch abgespeichert wird. Ein Klick auf “Start” lässt dann das Modulhandbuch erstellen. Mit “Löschen” werden die hochgeladenen Dateien gelöscht und Sie können neue Dateien hochladen. Mit “Zurück” gelangen Sie wieder zur Startseite

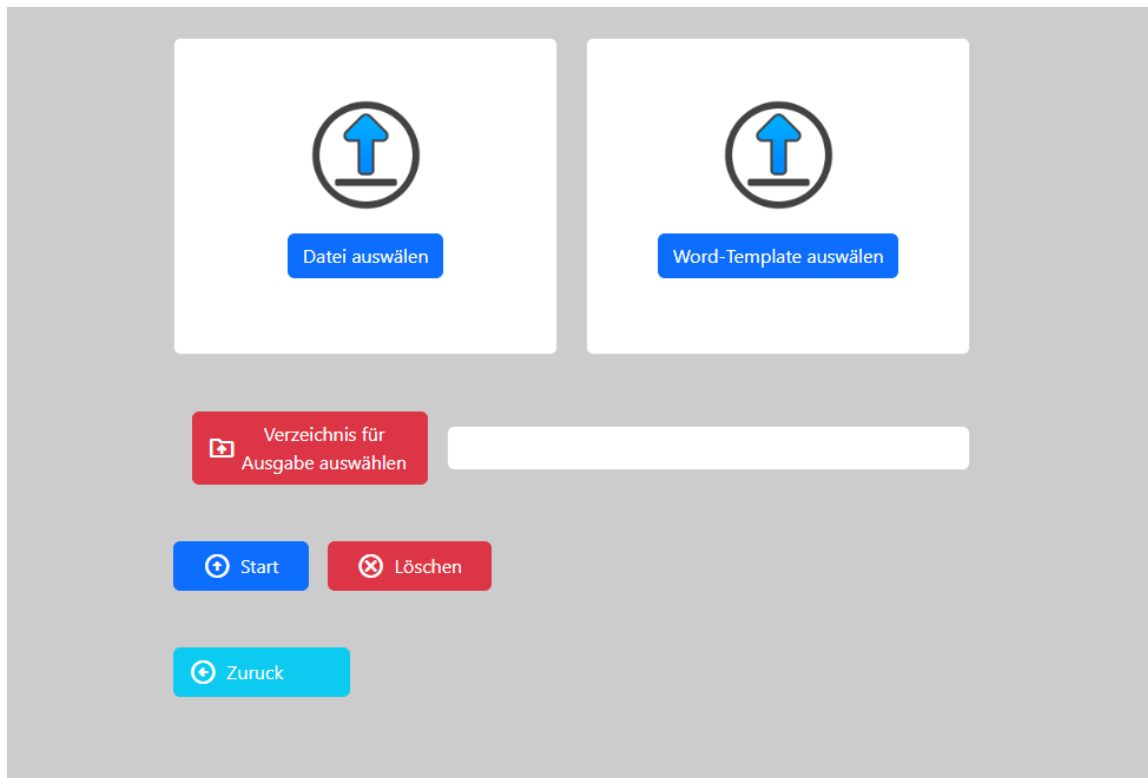


Bild. 6: Screenshot von Anwendung-Seite

- **Fehlermeldungen**

Beim Hochladen der Dateien achten Sie bitte auf das richtige Format sonst erscheint wie im Bild abgebildet eine Fehlermeldung. Auf der linken Seite sind ausschließlich nur Excel-Dateien zulässig und auf der rechten Seite sind ausschließlich 6 Word-Templates erlaubt. Wenn Sie die Anwendung starten ohne ein Verzeichnis für die Ausgabe auszuwählen, erscheint eine weitere Fehlermeldung mit der bitte ein Verzeichnis auszuwählen wie unten abgebildet.

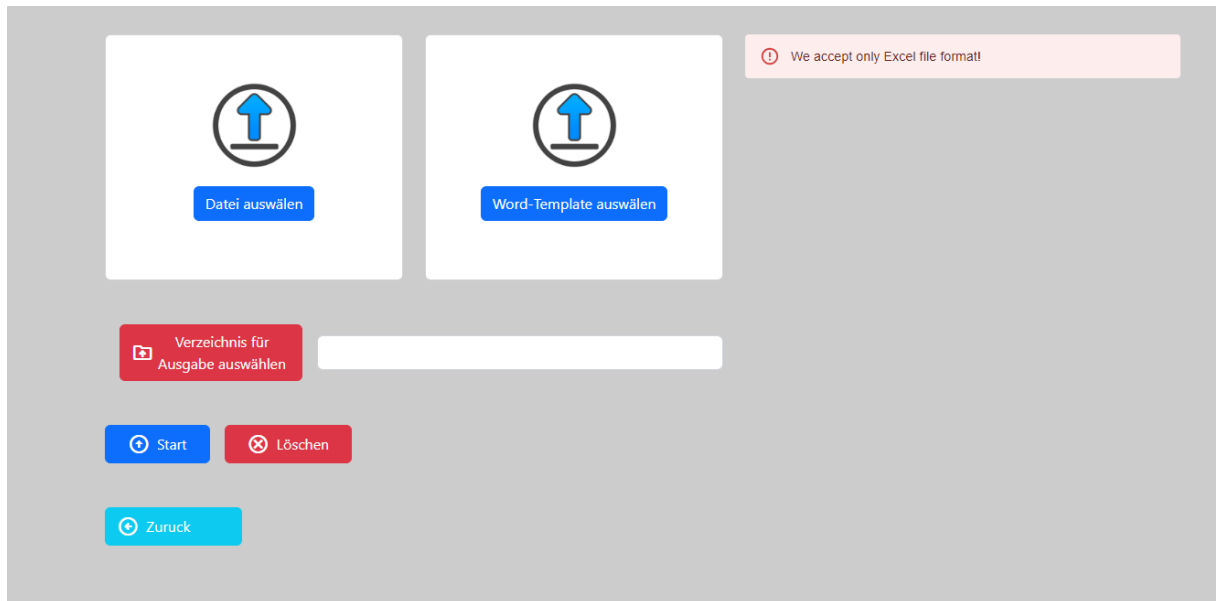


Bild. 7: Screenshot von Fehlermeldung : falscher Upload Format

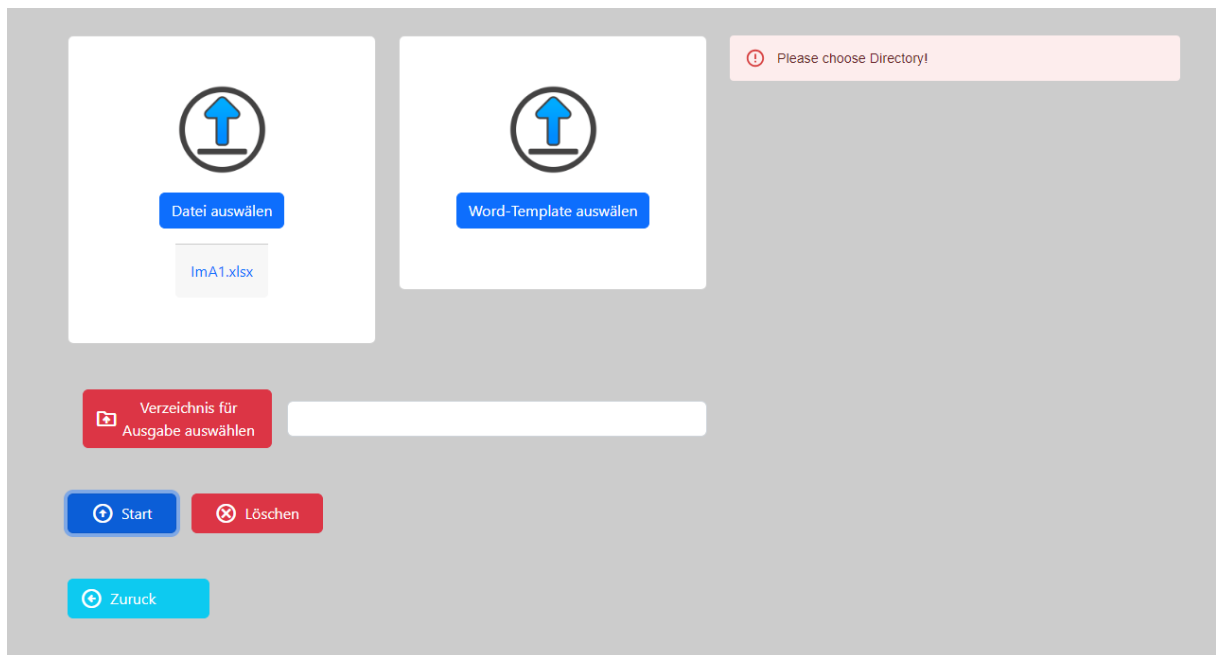


Bild. 8: Screenshot von Fehlermeldung: Wählen des Verzeichnisses

- Weitere Fehlermeldungen

Achten Sie darauf beim Hochladen der Templates 6 Word-Dateien auszuwählen, sonst erscheint eine Fehlermeldung die Sie darauf aufmerksam macht.

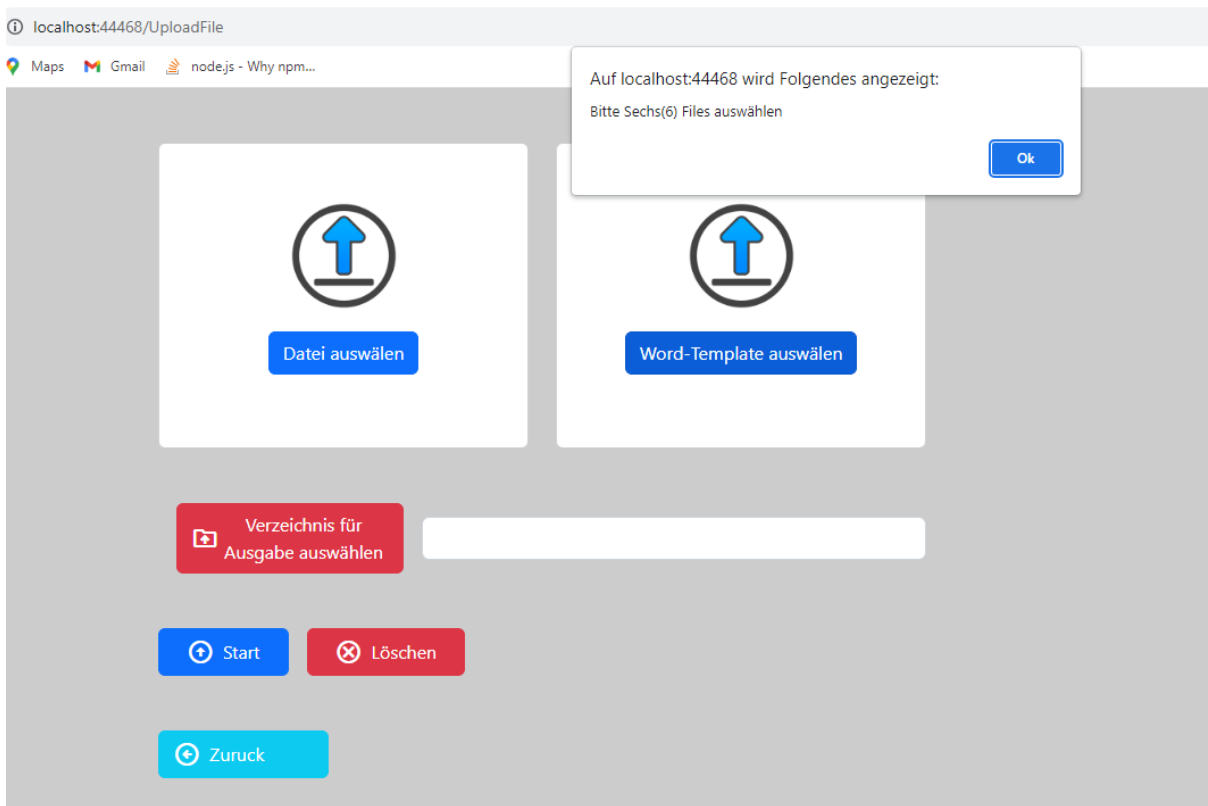


Bild. 9: Screenshot von Fehlermeldung: Wählen der Word Templates. Anzahl der Files muss 6 sein

- Templates hochladen

Beim Hochladen der Templates ist die Reihenfolge gegeben. Es ist unabhängig in welcher Reihenfolge Sie die Dateien hochladen. Sie erhalten eine Vorschau Ihrer Dateien und können durch "Submit" die Dateien in den Server hochladen. Dabei sehen Sie einen Fortschrittsbalken mit der Aufschrift "Completed" wenn die Dateien erfolgreich hochgeladen wurden.

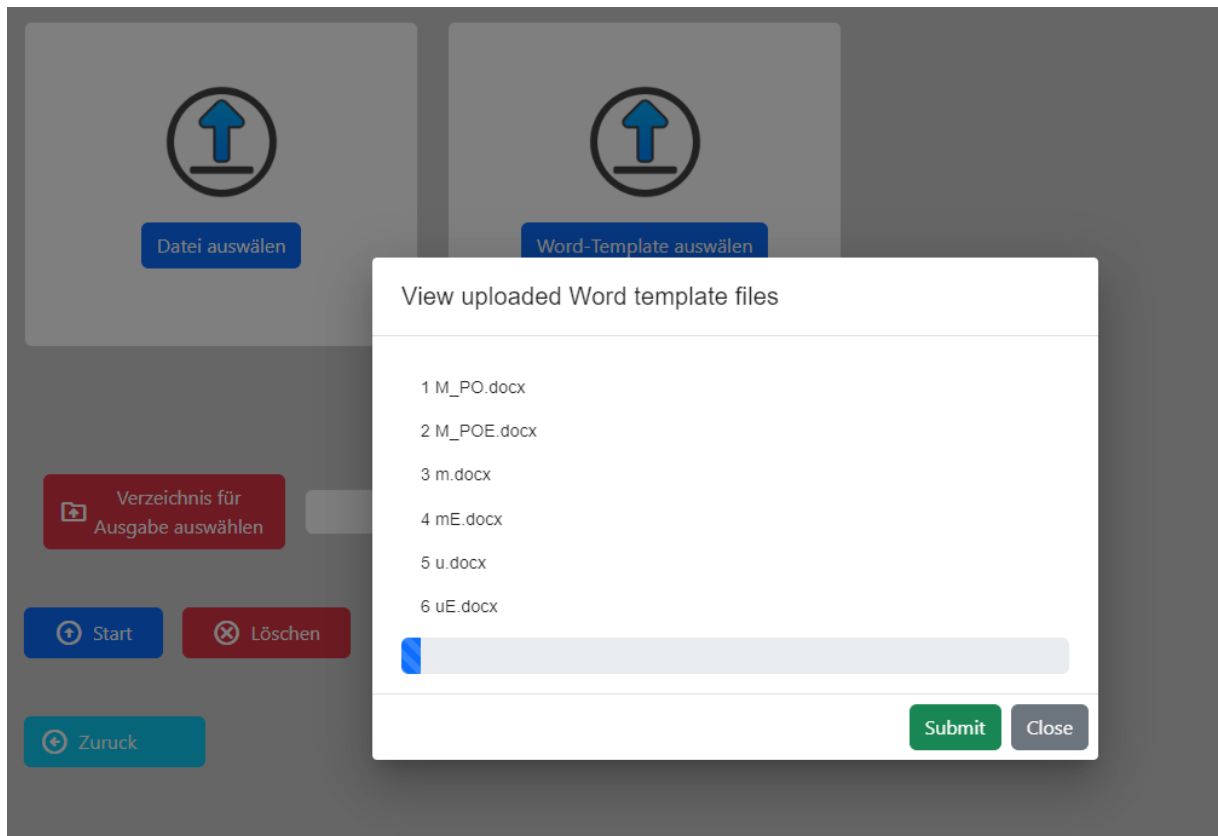


Bild. 10: Screenshot von Hochladen der Word Templates

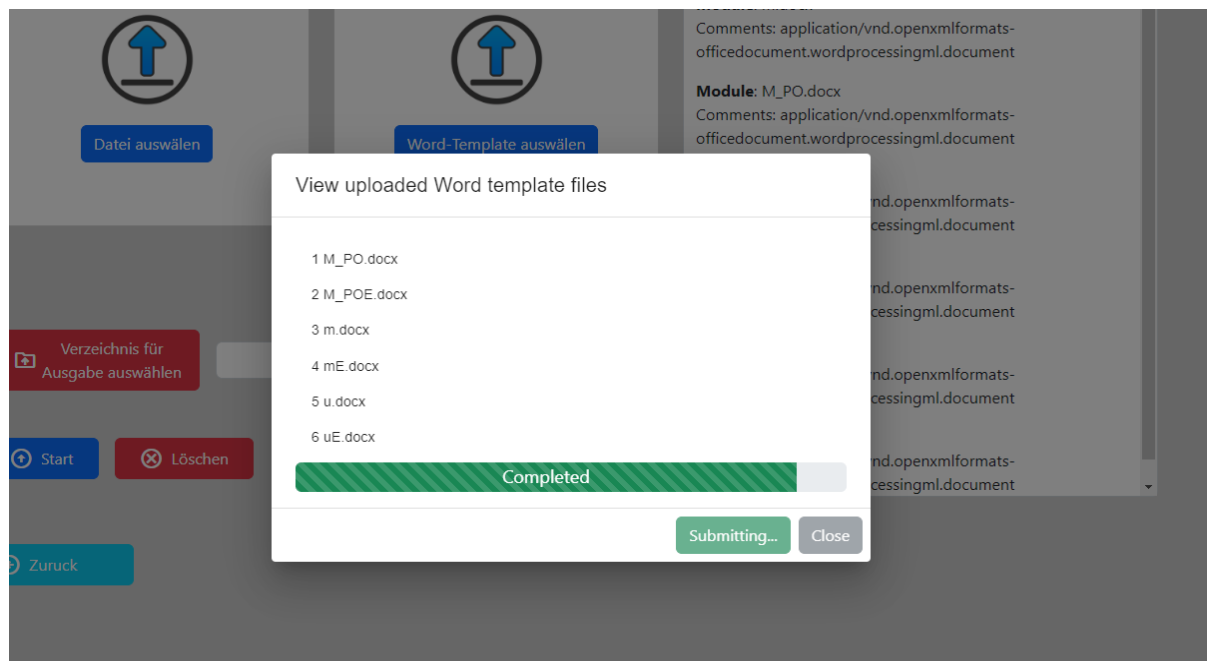


Bild. 11: Screenshot von Hochladen der Word Templates und zu backend schicken

- Verzeichnis auswählen

Hier sehen Sie das Fenster für den Verzeichnis der Ausgabe. Sie können wählen zwischen einem Excel-Ordner oder einem Word-Ordner in der das Modulhandbuch abgespeichert wird.

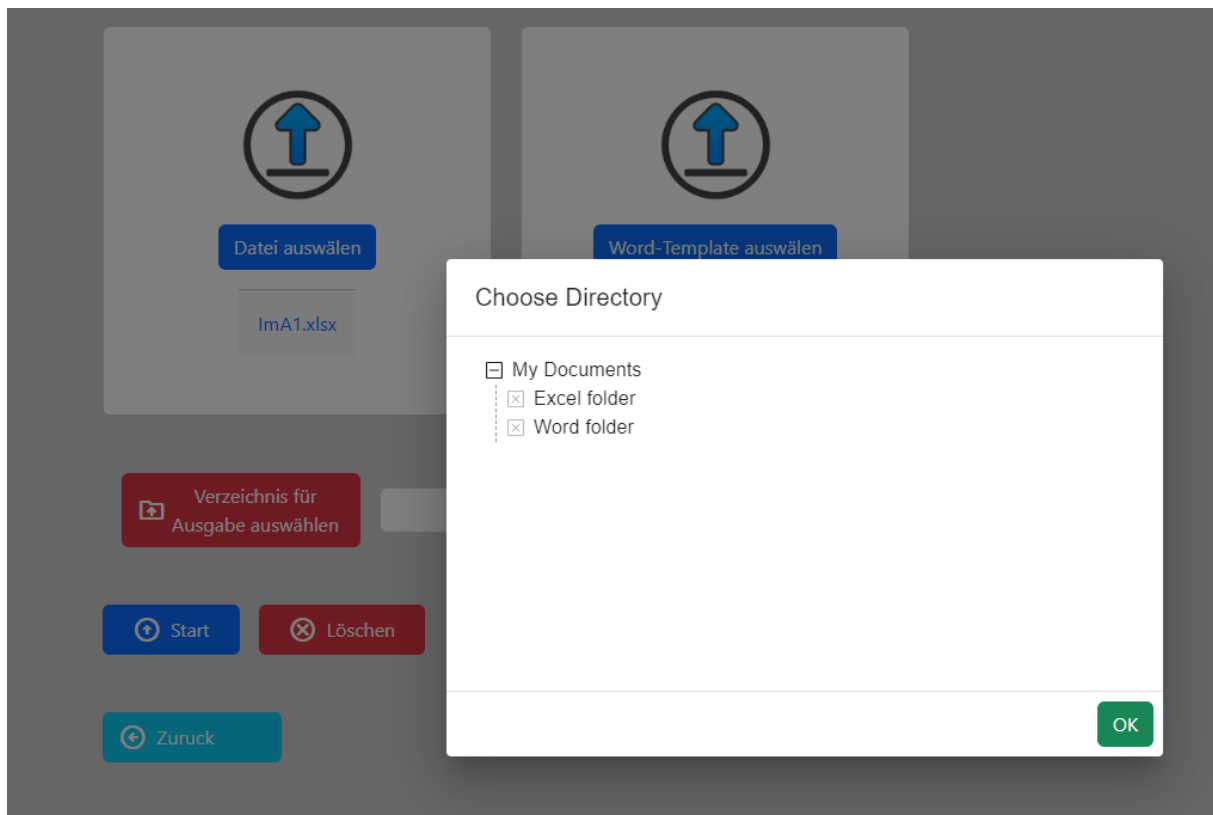


Bild. 12: Screenshot von wählen des Verzeichnisses

- Anwendung ausführen

Nachdem Sie alle benötigten Dateien hochgeladen haben und auch ein Verzeichnis für die Ausgabe ausgewählt haben, können Sie auf “Start” drücken. Rechts erscheint dann eine Fortschrittsanzeige, in der Sie sehen können, welche Module bearbeitet wurden. Darüber hinaus erscheint auch auf der Anzeige Fehlermeldungen, wenn welche auftauchen. Die Generierung des Modulhandbuchs wird mit einem weiteren Fortschrittsbalken begleitet. Wenn das Modulhandbuch erfolgreich generiert wurde, erscheint auf dem Fortschrittsbalken ein “Completed”.

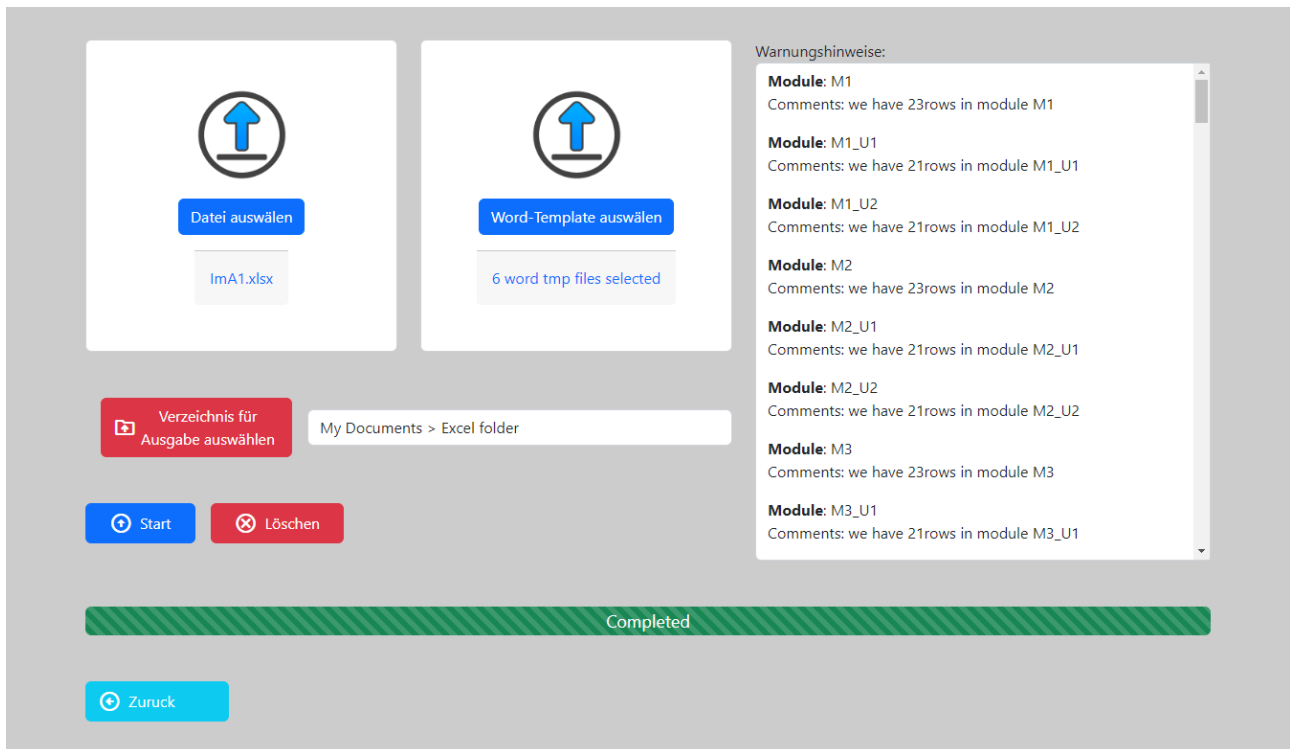


Bild. 13: Screenshot von Oberfläche, nachdem das hochladen und Bearbeitung fertig ist.

5. Statutory Declaration

Wir erklären hiermit, dass wir diese Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst haben. Wörtlich zitierte Sätze oder Satzteile sind als solche gekennzeichnet; sonstige Hinweise auf Aussage und Umfang sind durch vollständige Angabe der betreffenden Publikationen gekennzeichnet. Die Arbeit ist in gleicher oder ähnlicher Form noch bei keiner Prüfungsstelle eingereicht und nicht veröffentlicht worden. Die Arbeit wurde noch nicht, auch nicht teilweise, in einer anderen Prüfungsleistung oder als Studienarbeit verwendet.

Mohamed Benchellal:

William Iradukunda:

Unterschrift der Projektmitglieder