

STAFFBITE

Dario Schaffner, Jordan Dainese, Mario Kopp



Dokumentation M-326 Objektorientiert entwerfen und implementieren

Abgabe: 16.12.2022

Klasse: INFA 3a

Modullehrer: Peter Gisler

Repository: <https://github.com/dar1000/staffbite>

Inhalt

Vorwort	3
Aufbau und Ablauf (Teil 1)	3
Aufgabenstellung	3
Ausgangslage.....	3
Aufgabenstellung im Projektbeschreib.....	3
Vorgegebene Funktionale Anforderungen	4
Mögliche Erweiterte Funktionale Anforderungen	4
Vorgegebene Nicht-funktionale Anforderungen	4
Technologien.....	5
Projektaufbauorganisation	5
Projektmitglieder	5
Projektmethode	5
Zeitplan	6
Vorgegebene Termine	6
Geplante Arbeiten.....	6
Arbeitsjournal	8
Projekt (Teil 2).....	21
Kurzzusammenfassung	21
Ausgangslage.....	21
Umsetzung	21
Ergebnis.....	21
Informieren	22
Anforderungen.....	22
Planen	23
Sitemap	24
Entscheiden.....	25
Varianten.....	25
Risiken	25
Entscheid.....	25
Realisieren.....	26
Frontend.....	26
Backend.....	28
Klassendiagramm	30
Kontrollieren	32

Testkonzept.....	32
Unit Tests & UI Tests	36
Manuelle Tests	37
Benutzerhandbuch.....	50
Erstellen von Kompetenzen (Nur als Admin möglich)	50
Editieren von Kompetenzen (Nur als Admin möglich).....	51
Neue Subkompetenzen hinzufügen	51
Editieren einer bestehenden Subkompetenz	52
Subkompetenzen auf den Status erledigt setzen	52
Erstellen von zufälligen Gruppen	53
Auswerten.....	54
Reflexion	54
Erfolge	54
Misserfolge.....	54
Verbesserungsmöglichkeiten.....	55
Fazit	55
Quellenverzeichnis.....	57
Glossar	57
Abbildungsverzeichnis	58

Vorwort

Diese Dokumentation gehört zum Modul-326 und wurde für das GIBZ von Dario Schaffner, Jordan Dainese und Mario Kopp umgesetzt. In dieser Dokumentation ist das Vorgehen und der Ablauf der Arbeit an der Staffbite Applikation beschrieben.

Die Dokumentation ist in 2 Teile gegliedert.

Im ersten Teil werden die ausgearbeitete Aufgabenstellung und der Ablauf der Arbeit aufgezeigt. Ebenfalls ist ersichtlich, mit welchen Mitteln gearbeitet wurde und welche Vorkenntnisse in unserem Team bereits vorhanden sind.

Im zweiten Teil ist die Dokumentation der eigentlichen Arbeit niedergeschrieben, sowie die aufgetretenen Hindernisse und das Testing der Applikation.

Die Arbeit wurde von Anfang bis Schluss mit der Projektplanungsmethode IPERKA geplant.

Die Auswertungsphase wird mit einer Reflexion und Fazit abgeschlossen, welche am Schluss stehen.

Aufbau und Ablauf (Teil 1)

Aufgabenstellung

Inhaltlich geht es in der Projektarbeit des Moduls 326 um die Erstellung einer Software, welche von den Lehrpersonen des GIBZ zur längerfristigen Evaluation ihrer digitalen, unterrichtsbezogenen Anwendungskompetenzen genutzt werden kann: Aus einem Set definierter Kompetenzen sollen Lehrpersonen per Selbstdeklaration die individuelle Erreichung unterschiedlicher Kompetenzen erfassen und planen. Zur Unterstützung der Lehrpersonen können für die Kompetenzen unterschiedliche Ressourcen bereitgestellt werden.

Ausgangslage

Die Projektarbeit im Modul 326 wird dafür genutzt, die Prozesse, Formalitäten, Beurteilungsmodalitäten und Rahmenbedingungen dieser Projektarbeit im schulischen Kontext zu erfahren bzw. zu üben. Sie sollen durch eine möglichst praxisnahe Umsetzung einen ersten Einblick in die Durchführung dieser PA erhalten und daraus – neben den fachlichen Inhalten – auch wichtige Erkenntnisse hinsichtlich der Planung und Durchführung der PA erhalten.

Aufgabenstellung im Projektbeschreib

Die Mindestanforderung für die Projektarbeit umfasst die Konzeption, Realisierung, Testung und Dokumentation eines Tools für die Erfassung von unterrichtsbezogenen Informatik-Anwendungskompetenzen.

Vorgegebene Funktionale Anforderungen

- Benutzerinnen und Benutzer mit spezieller Berechtigung können Anwendungskompetenzen definieren. Dabei wird jede Kompetenz einem frei wählbaren Kompetenzbereich und einer Kompetenzstufe innerhalb dieses Bereichs zugewiesen. So könnte beispielsweise die Kompetenz ein Inhaltsverzeichnis erstellen, zum Kompetenzbereich Textverarbeitung gehören und auf der zweiten von insgesamt drei Kompetenzstufen eingeordnet sein.
- Jeder Kompetenz können beliebig viele Ressourcen (URLs) zugeordnet werden. Die Inhalte der Ressourcen unterstützen die Lehrpersonen bei der Erreichung der jeweiligen Kompetenz. Die Ressourcen werden durch die gleichen Personen verwaltet, welche die Kompetenzen und Kompetenzbereiche verwalten.
- Lehrpersonen können die zu entwickelnde Applikation nutzen, um ihre persönlichen Anwendungskompetenzen zu erfassen und weiterzuentwickeln. Dazu können Sie alle Kompetenzen (inkl. Ressourcen) einsehen, die individuell vorhandenen Kompetenzen in anonymisierter Form erfassen und persistieren. Zu einem späteren Zeitpunkt soll es möglich sein, den Kompetenzzuwachs abzubilden, indem die zusätzlich erworbenen Anwendungskompetenzen ergänzt werden.

Mögliche Erweiterte Funktionale Anforderungen

- Ein Tool, welches verwendet werden kann, um zufällig Gruppen zu erstellen. Der Benutzer kann angeben, wie viele Schüler- / Innen in einer Gruppe sein könne und welche Themen es gibt. Das Tool wird automatisch die Schüler einem Thema zuordnen und einer Gruppe hinzufügen.
- Ein Timer, welcher die Prüfungsdauer anzeigt und nach Durchlaufen der Zeit den User informiert.
- Ein Testkalender, in welchem die Lehrer die anstehenden Tests einschreiben können.

Vorgegebene Nicht-funktionale Anforderungen

- Das Projekt muss in einer objektorientierten Sprache umgesetzt werden.
 - Sprache und das Framework sind frei wählbar
- Die Entwicklung eines Testkonzepts inklusive Beschreibung aller Testfälle ist erforderlich
- Teil 2 der Dokumentation enthält ein vollständiges Testprotokoll
- Eine Testabdeckung durch automatisierte Tests (Unit-Tests) von 60 % ist anzustreben.

Technologien

- Angular
 - Front-End-Webapplikationsframework
 - Basiert auf Typescript
 - Cross-Plattform
 - Entwickelt von Google
- .NET Core (C#)
 - Software-Plattform, für die Entwicklung und Ausführung von Anwendungsprogrammen
 - Open-Source
- MongoDB
 - No-SQL Datenbankmanagementsystem
 - Daten werden in JSON-ähnlichen Dokumenten verwaltet
 - Flexibel und skalierbar

Projektaufbauorganisation

Projektmitglieder

Am Projekt Staffbite sind folgende Personen beteiligt

Name	Rolle
Dario Schaffner	Projektmitglied
Jordan Dainese	Projektmitglied
Mario Kopp	Projektmitglied
Peter Gisler	Experte, Fachvorgesetzte Person

Projektmethode

Als Projektmethode kommt IPERKA zum Einsatz.

Die Stärke der IPERKA Methode liegt darin, einen Auftrag Punkt für Punkt zu analysieren und auszuführen. Ausserdem lässt sich damit die Projektphasen klar voneinander trennen.

Aufgrund dieser Stärken und persönlichen Erfahrungen aus früheren Modulen haben wir uns somit klar für diese Projektmethode entschieden.

Zeitplan

Vorgegebene Termine

Datum	Termin
26.08.2022	Modulstart; Evaluation des Projektgegenstandes
02.09.2022	Abgabe des Projektauftrags
09.09.2022	Beginn der Umsetzungsarbeiten
04.11.2022	Zwischengespräch
16.12.2022	Abgabetermin Projektarbeit und Projektbericht
20.01.2023	Fachgespräche

Geplante Arbeiten

Da wir im Oktober 2 Wochen Ferien hatten, und wir geplant haben in dieser Zeit nicht am Projekt zu arbeiten, jedoch manche Sachen doch innerhalb dieser Ferien erledigt wurden, haben gewisse Aufgaben im Zeitplan keine Ist-Zeit markiert.

Auf der nächsten Seite wird der Zeitplan in der Dokumentation abgelegt. Um eine bessere Qualität zu ermöglichen, kann der Zeitplan auch unter folgendem Link angesehen werden.

Zeitplan Staffbite

<https://imgur.com/a/6Us9Rrf>

Zeitplan Staffliste			Aufwand (Lektionen)														IPERKA Phase	
	Sollzeit	Istzeit																
Kriterien im Team besprechen	1.00	1.00			September												Informieren	
Verbindlichen Projektauftrag definieren	1.00	2.00			18.09.2022	23.09.2022	30.09.2022	07.10.2022	Oktober									
Abschließen Journal	20.00	20.00			1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022	1.10.2022		
Zeitplan Layout erstellen	1.00	1.00															Planen	
Zeitplan planen	3.00	3.00																
Aufbau der Dokumentation	2.00	2.00															Einschleiden	
Dokumentation	16.00	15.00																
Lösungsalternative festlegen	1.00	1.00															Realisieren	
Einrichten der Projektumgebung	2.00	1.00																
Pages für Login und Dashboard	1.00	1.00															Kontrollieren	
Datenbank MongoDB mit Backend verlinken	1.00	1.00																
Services für Login	1.00	1.00															Auswerten	
Dashboard	2.00	1.00																
404 Not found page	1.00	1.00															Auswerten	
Usermanagementpage	2.00	2.00																
Komponenten für Admins und Lehrer	8.00	11.00															Auswerten	
Settingspage	2.00	1.00																
Teacher / Admin different views / navigation	2.00	1.00															Auswerten	
Dashboard Random Groups Funktion	2.00	3.00																
Dashboard Timer Funktion	2.00	3.00															Auswerten	
Session / Cookie Handling	4.00	4.00																
Testkonzept definieren	2.00	2.00															Auswerten	
Manuelle Tests durchführen	3.00	2.00																
Unit tests implementieren	4.00	4.00															Auswerten	
Refaktorieren & Fazit	2.00	2.00																
Total	58.00	57.00															Auswerten	

Mellertstein

Schäfer

Abbildung 1 - Zeitplan

Arbeitsjournal**Woche 4 (16.09.2022)**

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Einzelheiten des Projektauftrages im Team besprochen und notiert.	Analysieren der Kriterien des Projektauftrags. Verbindlicher Projektauftrag definiert. Einzelheiten des Projektauftrages im Team besprochen und notiert.	Einzelheiten des Projektauftrages im Team besprochen und notiert.
Erfolge & Misserfolge	Das Erstellen und definieren des Projektauftrags	Das Erstellen und definieren des Projektauftrags	Das Erstellen und definieren des Projektauftrags
Hilfestellung			

Woche 5 (23.09.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Aufsetzen der Basisstruktur. Angular Applikation erstellt sowie ein .NET Core Projekt für das Backend. Github Repository erstellt, mit korrekten gitignore Files. Github Actions hinzugefügt mit .NET und CodeQL Analysis Workflows.	Verfeinern und Überarbeiten des Projektauftrags. Erstellen des Layouts für den Zeitplan in der Form der Projektplanungsmethode IPERKA. Planen der allerersten Arbeitsschritte, um dem Team den Startschuss der Realisierung zu ermöglichen.	Einarbeiten in das für mich unbekannte Framework .NET Core. Vorlage mit Dario angeschaut und Tutorial zum Aufsetzen eines Webservers mit .NET Core angeschaut.
Erfolge & Misserfolge	Das Erstellen der Basisstruktur sowie des Repositories war leicht, da ich dies schon öfters erledigt habe. Das Erstellen der einzelnen Workflows war jedoch eher kompliziert, da ich noch keine Ahnung hatte, wie man einen Workflow verwendet / erstellt	Da wir die Projektplanungsmethode IPERKA bereits in früheren Modulen verwendete haben, war der Aufbau für mich klar. Die Erstellung des Zeitplans ging auch relativ schnell, da ich das ungefähre Layout bereits im Kopf hatte.	Schwieriger Einstieg, da ich in einer neuen Umgebung gearbeitet habe und das Objektorientierte programmieren schon lange nicht mehr gemacht habe.
Hilfestellung			

Woche 6 (30.09.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Arbeitsjournal in einem separaten Dokument erstellt und mit Herr Gisler besprochen. Basis Frontend Struktur erstellt mit Pages für Login und Dashboard. Solution Folders unbenannt und Branches damit sie dem "staffbite" Theme passen. Login Page und Service erstellt für das Anmelden. Der Service schickt ein Post Request an das backend.	Die Abschnitte Informieren, Planen und Entscheiden im Zeitplan eingeplant. Mit dem Team die genauen Arbeiten besprochen und Notizen zu den erwarteten Zeitaufwänden gemacht. Aufgrund der angenommenen Zeitaufwänden die Abschnitte Realisieren und Kontrollieren geplant. Mit dem Aufbau der Dokumentation begonnen.	Krank
Erfolge & Misserfolge	Das Erstellen des Services ging schnell, jedoch habe ich lange gebraucht für das Styling der Login Page, da «Design» eher eine meiner Schwächen ist.	Die 4 Abschnitte Informieren, Planen und Entscheiden und Kontrollieren konnte ich schnell einplanen. Der Realisierungsteil war einiges schwerer. Ich hatte etwas Schwierigkeiten abzuschätzen, wie lange für eine Arbeit braucht, da die Fähigkeiten & Stärken bei jeder Person unterschiedlich ist.	
Hilfestellung			

Woche 7 (07.10.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	<p>Dashboard Komponenten erstellt für Administratoren und Lehrpersonen.</p> <p>Eine Vertikale Navigation erstellt für das Navigieren innerhalb des Dashboards.</p> <p>Page Not Found Seite erstellt, damit bei invaliden Links der Benutzer auf eine schöne Seite gelangt und somit weiss der Link war ungültig.</p> <p>UI für die Einstellungen Seite.</p> <p>Ein simples form erstellt, damit der Benutzer seine «Profile» Einstellungen ändern kann wie z.B. : E-Mail und Password.</p>	<p>Zeitplan nochmals etwas revidiert.</p> <p>Die Inhalte der Dokumentation bestimmt, diese wurden in Aufbau und Ablauf (Teil 1), dem Projekt (Teil 2) und dem Vorwort, unterteilt, um den Kriterien zu entsprechen. Der nächste Schritt war, das Vorwort zu verfassen.</p>	<p>Aufsetzen des Webserver, erste Testroutes erstellt.</p> <p>Filestruktur aufgebaut mit Controllern, Services und Models.</p> <p>MongoDB Verbindung mit dem Backend hergestellt und ausprobiert.</p>
Erfolge & Misserfolge	<p>Die Dashboard-Navigation sowie auch die Page Not Found Seite hatte ich schnell erledigt. Bei der vertikalen Navbar hatte ich jedoch Schwierigkeiten die svg's richtig anzuzeigen und das alle die gleiche Grösse und Farbe hatten. Dank Stack Overflow habe ich dies dann doch auch gut hingbracht.</p> <p>In dieser Woche war leider die Lehrperson nicht vor Ort und allfällige Fragen mussten wir auf die Woche darauf verschieben.</p>	<p>Zeitplan abgeschlossen. Zuerst war mir etwas unklar, welche Teile alle in der Dokumentation vorhanden sein müssen. Dazu habe ich mir den Teil B im Kriterienkatalog nochmals durchgeschaut.</p>	<p>Erste Erfolge im Aufsetzen des Web Servers konnten erzielt werden.</p>
Hilfestellung			

Woche 8 (14.10.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	In dieser Woche habe ich mit der Implementation der Kompetenzen Seite angefangen und das Feature implementiert, dass Lehrpersonen die Kompetenzen als «erledigt» markieren können. Zusätzlich habe ich das Feature erstellt, dass Admins neue Benutzer erfassen können.	Diese Woche habe ich mit dem Teil 1 (Aufbau und Ablauf) begonnen. Dabei habe ich die Ausgangslage, die Aufgabenstellung, verwendeten Technologie und die vorgegebenen funktionalen, möglichen erweiterten und die nicht funktionalen Anforderungen vom Projektantrag übernommen und eingepflegt. Als Nächstes habe ich die Projektaufbauorganisation und die geplanten Termine in die Dokumentation übernommen.	Ich habe mit der Funktionalität angefangen, dass ein User erstellt werden kann und in der Datenbank gespeichert wird. Dazu habe ich auch das erste Model für den User erstellt, um es in der Datenbank speichern zu können. Dies hat schliesslich auch funktioniert, jedoch war die Response, die ans Frontend zurückgegeben wird, noch nicht richtig.
Erfolge & Misserfolge	Das feature eine Kompetenz als «erledigt» zu markieren war ziemlich einfach. Hier war mir jedoch noch gar nicht bewusst, dass man pro «Level» mehrere Sub-Kompetenzen haben kann. Dies habe ich erst später im Verlauf des Projektes bemerkt.	Teil 1 ist nun abgeschlossen. Der Zeitplan wird jedoch erst eingefügt, wenn die «Ist-Zeit» ebenfalls ausgefüllt ist. Misserfolge gab es heute nicht, da das ganze simple war.	Erste Route, die fast richtig funktioniert, Daten können in der MongoDB gespeichert werden.
Hilfestellung			

Woche 9 (21.10.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	<p>In dieser Woche habe ich das Feature erstellt, dass die Schulleitung / Personen mit der Admin Rolle die Möglichkeit haben neue Kompetenzen zu erfassen.</p> <p>Ich habe auch die notwendigen Services erstellt, damit die richtigen Post und Get requests gemacht werden und die richtigen Daten mitgeschickt werden.</p>	<p>Heute habe ich mit dem Teil 2 (Projekt) und somit mit der Kurzzusammenfassung angefangen. Dabei habe ich die Ausgangslage des Projektauftrags beschrieben, eine kurze Zusammenfassung der geplanten Umsetzung verfasst und noch niedergeschrieben, was unsere Applikation am Schluss alles können soll.</p>	<p>Heute habe ich zwei neue Routes erstellt. Jetzt kann man einen User wieder löschen und auch authentifizieren für ein Login.</p>
Erfolge & Misserfolge	<p>Bei diesem Feature hatte ich immer wieder Probleme. Zuerst musste ich irgendwie herausfinden, wie ich das ganze überhaupt darstellen möchte und wie eine Lehrperson ein «Form» ausfüllen kann, welches die Kompetenzen erweitert. Ich hatte ziemlich viel Probleme mit Referenzen auf falsche Listen. Dies habe ich jedoch am Ende beheben können.</p>	<p>Die Kurzzusammenfassung für den Teil 2 (Projekt) verfasst. Ich hatte keine Schwierigkeiten beim Verfassen dieses Abschnitts.</p>	<p>2 neue Routes erstellt und ausprobiert. Diese funktionieren und ich konnte diese mit fast keinen Problemen erstellen.</p>
Hilfestellung			

Woche 10 (28.10.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	In dieser Woche habe ich ein neues Feature erstellt, damit Admin's eine Übersicht aller Benutzer sehen sowie auch die einzelnen Kompetenzen einer Lehrperson ansehen können. Dies ermöglicht es zu sehen, welche Kompetenzen eine Lehrperson hat und welchen Bereichen sie vielleicht noch Hilfe / Aufklärung braucht. Ich habe auch diverse Fehler behoben und UI-Optimierungen erledigt.	Militär Orientierungstag	Eine Funktionalität, die wir noch brauchen, die zusammenhängend mit der Loginfunktion, die ich letzte Woche erstellt habe, ist das Session-Handling. Da ich so etwas noch nie gemacht habe, musste ich mich zuerst darüber informieren, wie das genau funktioniert und wie ich das am besten mit meinem Framework umsetzen kann. Danach habe ich mit der Implementation angefangen, bin jedoch noch nicht fertig geworden.
Erfolge & Misserfolge	Diese Aufgaben waren eher leicht, da ich ähnliches in anderen Projekten schon erledigt hatte.		Session Handling als neues Thema, von dem ich keine Ahnung habe, ist am Anfang schwierig für mich
Hilfestellung			

Woche 11 (04.11.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	In dieser Woche habe ich gemeinsam mit Jordan das erste Mal das Backend angeschaut und getestet, ob dies funktioniert so wie wir es uns gedacht haben. Es hat sich herausgestellt, dass ziemlich viel am Backend und in den Services noch angepasst werden muss, damit es schlussendlich mit einer MongoDB funktionieren wird. Zusätzlich habe ich noch die Option erstellt, dass man pro «Level» noch Sub-Kompetenzen erfassen kann sowie auch pro Kompetenz ein Video-URL und eine PDF-URL hinzufügen kann.	Heute stand der Teil Informieren der IPERKA Methode an. Dabei habe ich die möglichen erweiterten Funktionen, welche nun fest implementiert werden, übernommen. Mir ist noch aufgefallen, dass es gewisse nicht funktionale Funktionalitäten gibt, welche nicht definiert wurden, allerdings essenziell sind für eine gute und zuverlässige Applikation. Ich habe dies mit dem Team besprochen und noch als erweiterte nicht funktionale Anforderungen niedergeschrieben.	Heute habe ich mit Dario zusammen die vorhandenen Funktionalitäten des Backends mit Dario getestet und dieses mit dem Frontend benutzt, dass man beides zusammen verwenden kann. Dies hat nach ein paar Anpassungen dann auch einigermaßen funktioniert, jedoch fehlten noch einige Funktionalitäten und man muss auch noch ein paar Anpassungen machen. Dann habe ich noch angefangen mehrere Models für den User zu erstellen, die wir für die einzelnen Requests brauchen wollen, um die Inhalte der Anfragen des Frontend's einfacher empfangen zu können.
Erfolge & Misserfolge	Da ich für das Frontend zuständig war und Jordan eigentlich für das Backend war es schwierig die Services auf den ersten versuch direkt zum Laufen zu bringen, da Jordan nicht wusste, was ich alles ihm von meinem Request mitschicke und was er zurückgeben sollte. Aber als wir zusammen hingesessen sind und es uns genauer angeschaut haben, war ziemlich schnell alles klar und die User Registrierung hat funktioniert.	IPERKA Teil Informieren ist nun verfasst. Durch das Projekt hindurch kommen immer neue Erkenntnisse welche, das ganze Projekt / die Applikation noch verbessern könnte. Die kleinen Änderungen (wie z.B. bei uns die erweiterten nicht-funktionalen Anforderungen) können einen massiven Einfluss auf die Qualität der Applikation haben, wenn man sich dies Explizit darauf achtet.	Erste Funktionalität des Registrierens hat in Kombination mit Backend und Frontend funktioniert. Weiter besprochen, was noch alles gemacht werden muss.
Hilfestellung			

Woche 12 (11.11.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	In dieser Woche habe ich viele Bugs behoben sowie auch neue Features implementiert. Eines davon war die Funktionalität für Admins Kompetenzen zu löschen. Für das Editieren war ich mir nicht sicher, ob dies auch eine Anforderung war und deshalb habe ich die Woche darauf dann nachgefragt, ob dies auch notwendig sei.	In dieser Woche ging es um das Erfassen der Planung. Dies ist der 2 Teil der IPERKA Projektplanungsmethode. Dabei ging es um die allgemeine Einsetzung der Projektplanungsmethode IPERKA und auch um die Zielsetzung bei den Tests, welche wir in Manuelle und Unit-Tests unterschieden haben. Danach ging es an den 3 Teil von IPERKA dem Entscheid. Die Varianten & Risiken haben wir am Anfang des Projekts gemeinsam besprochen und evaluiert.	Das Session Handling habe ich letzte Woche pausiert und an etwas anderem gearbeitet, da dies aber noch nicht funktioniert, habe ich mich heute wieder darangesetzt und konnte es auch umsetzen, dass ein Cookie generiert wird.
Erfolge & Misserfolge	Als ich weiter am Backend gearbeitet habe, habe ich bemerkt, dass unsere Requests von CORS geblockt werden. Dies ist kritisch, da sonst unser ganzes backend nicht funktionieren würde. Mithilfe von ein paar YouTube Videos konnte ich dies aber schnell beheben.	Der Teil Planen und der Entscheid sind nun abgeschlossen und wurden kurz im Team besprochen. Misserfolge gab es keine.	Session Handling hat so weit funktioniert. Danach habe ich noch angefangen mich zu informieren wie das Hashen eines Passworts funktioniert, da ich dies nicht wusste und wir dies auch noch umsetzen wollen.
Hilfestellung		Mit der Lehrperson besprochen, ob alle relevanten Punkte in der Dokumentation vorhanden sind.	

Woche 13 (18.11.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	<p>Ich habe angefangen, alles gründlich zu testen und geschaut, was alles noch angepasst werden muss. Ich habe auch genau geschaut, ob ich irgendwelche komische Fehler erzeugen könnte. Nach dem Testen habe ich alles Schritt für Schritt durcharbeitet und alle Fehler, die aufgetaucht sind, behoben.</p> <p>In dieser Woche konnte ich auch noch alles, was bei dem Zwischengespräch aufgetaucht ist beheben / implementieren. Die Möglichkeit bestehende Kompetenzen zu editieren sei anscheinend auch eine Anforderung und dies habe ich auch probiert zu implementieren.</p> <p>Ich habe auch noch die ganzen Workflows / Actions auf GitHub angepasst da ich bemerkt habe, dass diese seit dem Umbenennen der Ordner-Struktur gar nicht mehr funktionierten.</p> <p>Zusätzlich habe ich noch die Timer-Funktion für Lehrpersonen erstellt.</p>	<p>Heute habe ich mit dem Testkonzept begonnen. Dabei habe ich die Tabellen für das Konzept und das eigentliche Protokoll erstellt. Das Konzept ist lediglich die Beschreibung, dieses Konzept kann dann bei jeder Änderung getestet und protokolliert werden.</p>	<p>Heute konnte ich umsetzen, dass das eingegebene Passwort des Nutzers gehasht wird, bevor es in der Datenbank gespeichert wird. Zusätzlich habe ich dann eine weitere notwendige Funktionalität hinzugefügt, damit man den Hash des Passworts abgleichen kann, da sonst das Hashen an sich sinnlos wäre.</p>
Erfolge & Misserfolge	<p>Mit dem Editieren der bestehenden Kompetenzen sind ganz viele neue Fragen und Fehler aufgetaucht. Ich musste zuerst überlegen, wie ein Admin am liebsten Kompetenzen editieren will. Nach langem herumtüfteln hatte ich es jedoch geschafft, dass man bereits bestehende Kompetenzen editieren und verändern kann.</p>	<p>Manchmal ist es als Projektmitglied einer Software schwierig zu erkennen, welche «dummen» Eingaben/Klicks ein User eigentlich machen kann. So kam auch das Extremwert-Testing zum Einsatz, welches unsere Applikation noch etwas benutzerfreundlicher gemacht hat. Misserfolge dabei waren allerdings, dass wir auch noch ein paar Schwachstellen entdeckt haben, welche besser gehandelt werden können.</p>	<p>Passwort Hashing hat funktioniert und man kann nun den Hash auch mit dem eingegebenen Passwort vergleichen.</p>

Hilfestellung		<p>Mit der Lehrperson besprochen, ob wir als Projektmitglieder ebenfalls Blackboxtesting durchführen sollen.</p> <p>Mit der Lehrperson besprochen, dass wir nun nur Whiteboxtesting durchführen. Wir vermerken jedoch, dass es beide Methoden gibt.</p>	
---------------	--	---	--

Woche 14 (25.11.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Da nun das Frontend so weit fertig ist und im Backend noch so gut wie nichts gemacht wurde, habe ich angefangen Jordan zu helfen und mit der Logik für das Speichern der Kompetenzen angefangen.	Da das Backend noch nicht ready war, konnten die richtigen Tests noch nicht durchgeführt und protokolliert werden. Unser Frontend funktionierte bereits mit Runtime Data, weshalb ich mich dazu entschieden habe, heute die Benutzeranleitung zu verfassen. Heute habe ich mit dem Benutzerhandbuch angefangen. Dabei habe ich für diverse Szenarien Anleitungen verfasst, um das Ganze noch benutzerfreundlicher zu machen. Für die Funktionen wie erstellen, editieren einer Kompetenz, dem abschliessen einer Kompetenz und dem Erstellen von zufälligen Gruppen habe ich einen Abschnitt erstellt.	Die Hilfe von Dario kam mir gelegen, da ich oft Schwierigkeiten mit der MongoDB und .NET Core hatte und deshalb nicht so schnell vorangekommen bin wie gewünscht. Da er sich bereits damit auskannte, konnte er gut aushelfen. Ich habe zeitgleich damit angefangen mit anzuschauen, wie ich Unittests erstellen kann, da wir eine 60 % Coverage anstreben. Ich habe mich dazu entschieden das Testframework Xunit zu verwenden und habe angefangen damit herumzuprobieren und erste Unittests für den UserController zu erstellen.
Erfolge & Misserfolge	Frontend ist nun fertig.	Der Schwerpunkt hier war herauszufinden,	Neue Technologie für Unittests, diese ist

		was für den User schwierig sein könnte. Denn als Teammitglied ist es klar, dass alle Teile klar und verständlich sind. Mein Erfolg heute war, dass nun sicherlich jeder Benutzer mit der Applikation klarkommt.	jedoch einfach verständlich und ich konnte schnell damit Fortschritte erzielen.
Hilfestellung			

Woche 15 (02.12.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	In dieser Woche habe ich mich mit dem Thema UI Testing befasst. Ich habe schon ein bisschen Erfahrung, da ich UI-Tests mit Selenium im Geschäft mache. Jedoch wollte ich etwas anderes verwenden also habe ich mich für Cypress entschieden. Ich habe die Basis Grundstruktur für das Ausführen der notwendigen Cypress UI-Tests erstellt.	Heute habe ich mich mit dem eigentlichen Durchführen der manuellen Tests befasst.	In dieser Woche habe ich weiter an den Unittests gearbeitet und konnte nun mit insgesamt 13 Tests den Usercontroller abdecken. Zudem habe ich noch in der Dokumentation gearbeitet, da ich noch ein paar Sachen nachholen musste.
Erfolge & Misserfolge	Ich habe den ersten Test geschrieben, welcher direkt erfolgreich durchgeführt wurde. Das Aufsetzen von Cypress hat nicht lange gedauert und dank der ausführlichen Anleitung, war alles klar.	Da wir dank IPERKA das ganze Projekt gut durchgeplant hatten, verliefen die manuellen Tests sehr gut. Es mussten nur noch wenige Anpassungen vorgenommen werden.	Unittests für den Usercontroller sind fertig.
Hilfestellung			

Woche 16 (09.12.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Mit dem Team die letzten notwendigen Schritte & Anpassungen besprochen.	Mit dem Team die letzten notwendigen Schritte & Anpassungen besprochen. Die ganze Dokumentation und die Applikation zusammen reviewed. Bereits mit der letzten Phase begonnen, dem Auswerten.	Mit dem Team die letzten notwendigen Schritte & Anpassungen besprochen.
Erfolge & Misserfolge	Die intensive Projektplanung hat uns gut durch den Zeitplan hindurch gebracht. Somit sind nur noch wenige Anpassungen und Verbesserungen nötig.	Die intensive Projektplanung hat uns gut durch den Zeitplan hindurch gebracht. Somit sind nur noch wenige Anpassungen und Verbesserungen nötig.	Die intensive Projektplanung hat uns gut durch den Zeitplan hindurch gebracht. Somit sind nur noch wenige Anpassungen und Verbesserungen nötig.
Hilfestellung	<p>Mit der Lehrperson besprochen, ob wir ein Realisierungskonzept erstellen sollten und besprochen, wie wir unser Testkonzept aufbauen sollten.</p> <p>Mit der Lehrperson besprochen, ob wir jedes einzelne Feature im Testprotokoll einfügen sollten, was dazu führen würde, dass wir um die 60 Testfälle haben. Gemeinsam entschieden, nur gewisse Funktionalitäten im Testprotokoll einzufügen.</p>		

Woche 17 (16.12.2022)

	Dario	Mario	Jordan
Aktivität (inkl. Zeiten)	Bug fixes, Dokumentation erweitert, mit der Frontend Realisierung, Zeitplan überprüft. Dokumentation überflogen	Persönliches Fazit verfasst, unser Quellenverzeichnis abgeschlossen, wichtige Wörter im Glossar erklärt und das Abbildungsverzeichnis erstellt.	An der Dokumentation gearbeitet, Klassen Modell gemacht, Code kommentiert, Zeitplan überprüft, Fazit hinzugefügt, offene Sachen noch abgeschlossen.
Erfolge & Misserfolge	Unser Projekt haben wir erfolgreich abgeschlossen und abgegeben.	Unser Projekt haben wir erfolgreich abgeschlossen und abgegeben.	Unser Projekt haben wir erfolgreich abgeschlossen und abgegeben.
Hilfestellung			

Projekt (Teil 2)

Kurzzusammenfassung

Ausgangslage

Gegenwärtig werden die unterrichtsbezogenen Informatik-Anwendungskompetenzen von Hand erfasst. Es soll nun ein Webapplikation geschaffen werden, in welcher man all diese Informatik-Anwendungskompetenzen online erfassen kann.

Dabei sollen Anwendungskompetenzen definiert werden, welche einen frei wählbaren Kompetenzbereich und eine Kompetenzstufe enthalten. Jede Kompetenz soll beliebig viele Ressourcen zugeordnet werden können. Lehrpersonen sollen die Möglichkeit haben, selbst Kompetenzen zu erfassen und weiterzuentwickeln. Dazu sollen sie alle Kompetenzen inklusiv. Ressourcen einsehen können.

Umsetzung

Das Projekt wurde unter Verwendung der IPERKA-Projektmethodik durchgeführt.

Als Erstes stand die Informationsphase an, in dieser wurde gemäss der Aufgabestellung Anforderungen definiert.

Während der Planungsphase wurde ein Zeitplan, eine Grundstruktur für die Dokumentation erstellt.

In der Realisierungsphase wurde mithilfe des Angular-Webframework basierend auf Typescript, HTML und CSS das Frontend und mithilfe von .NET Core basierend auf C# das Backend entwickelt.

Die Kontrollphase bestand aus zahlreichen Unit-Tests und weiteren manuellen Tests.

Zum Schluss stand eine Auswertung an, in der Erfolge, Probleme und Verbesserungsmöglichkeiten analysiert und diskutiert wurden. Aufgrund der vorliegenden Fakten haben wir ein gemeinsames Fazit gezogen.

Ergebnis

Mit unserer Applikation kann die Schulleitung eine Übersicht über die Kompetenzen der Lehrer erhalten. Damit kann die Schulleitung ihre eigenen Kompetenzen, in Form eines Kompetenzrasters, definieren und die Lehrer können angeben, welcher Kompetenz sie mächtig sind. Zusätzlich können die Lehrpersonen auf weitere nützliche Features zugreifen, um den Lehreralltag zu vereinfachen. Darunter kann er unter seinen Schülern zufällige Gruppen in beliebiger Grösse erstellen und den jeweiligen Gruppen ein Thema zuzuweisen, welches sie bearbeiten sollen. Als weiteres Feature kann der Lehrer eine Stoppuhr einstellen und diese beispielsweise bei Prüfungen verwenden und so einen einfachen Überblick über die Zeit behalten.

Informieren

Das genaue Analysieren der Aufgabenstellung ist das Wichtigste in der ersten Phase der Projektplanungsmethode IPERKA.

Anforderungen

Wir haben hier alle funktionalen, nicht funktionalen und erweiterten Anforderungen basierend auf der Aufgabenstellung genau definiert. Basierend auf diesen Anforderungen wird im nächsten Schritt die Implementierungen geplant.

Funktionale Anforderungen

- Benutzerinnen und Benutzer mit spezieller Berechtigung können Anwendungskompetenzen definieren. Dabei wird jede Kompetenz einem frei wählbaren Kompetenz-bereich und einer Kompetenzstufe innerhalb dieses Bereichs zugewiesen. So könnte beispielsweise die Kompetenz ein Inhaltsverzeichnis erstellen, zum Kompetenzbereich Textverarbeitung gehören und auf der zweiten von insgesamt drei Kompetenzstufen eingeordnet sein.
- Jeder Kompetenz können beliebig viele Ressourcen (URLs) zugeordnet werden. Die Inhalte der Ressourcen unterstützen die Lehrpersonen bei der Erreichung der jeweiligen Kompetenz. Die Ressourcen werden durch die gleichen Personen verwaltet, welche die Kompetenzen und Kompetenzbereiche verwalten.
- Lehrpersonen können die zu entwickelnde Applikation nutzen, um ihre persönlichen Anwendungskompetenzen zu erfassen und weiterzuentwickeln. Dazu können Sie alle Kompetenzen (inkl. Ressourcen) einsehen, die individuell vorhandenen Kompetenzen in anonymisierter Form erfassen und persistieren. Zu einem späteren Zeitpunkt soll es möglich sein, den Kompetenzzuwachs abzubilden, indem die zusätzlich erworbenen Anwendungskompetenzen ergänzt werden.

Erweiterte Anforderungen

- Ein Tool, welches verwendet werden kann, um zufällig Gruppen zu erstellen. Der Benutzer kann angeben, wie viele Schüler- / Innen in einer Gruppe sein könne und welche Themen es gibt. Das Tool wird automatisch die Schüler einem Thema zuordnen und einer Gruppe hinzufügen.
- Timer, welcher die Prüfungsdauer anzeigt und nach Durchlaufen der Zeit den User informiert.

Nicht funktionale Anforderungen

- Das Projekt muss in einer objektorientierten Sprache umgesetzt werden.
 - Sprache und das Framework sind frei wählbar
- Die Entwicklung eines Testkonzepts inklusive Beschreibung aller Testfälle ist erforderlich
- Teil 2 der Dokumentation enthält ein vollständiges Testprotokoll
- Eine Testabdeckung durch automatisierte Tests (Unit-Tests) von 60 % ist anzustreben.

Erweiterte nicht funktionale Anforderungen

- Die Applikation kann Fehler tolerieren und stürzt bei Falscheingaben nicht gleich ab.
- Passwort Hashing für Benutzerpasswörter
- Vernünftige Ladezeiten
- Selbsterklärende Verständlichkeit der Bedienung
- Clean Code für einfache Wartbarkeit

Planen

In dieser Phase wird der Verlauf des Projektes bestimmt und ist somit ein essenzieller Teil in der Projektplanungsmethode IPERKA. In dieser Phase wird der zeitliche Ablauf bestimmt und in den Zeitplan eingefügt.

Der Zeitplan befindet sich beim Punkt «Geplante Arbeiten». Aus Redundanzgründen wird dieser hier nicht nochmals abgebildet.

Das Projekt wurden von Anfang bis Schluss mit der IPERKA Projektplanungsmethode geplant und durchgeführt.

Es ist eine Testabdeckung von rund 60 % geplant, welche die Funktionalitäten der Applikation überprüfen.

Unterschieden wird zwischen Unit-Tests und manuellen Tests. Wobei bei es zwei Arten von manuellen Tests gibt. Das sogenannte Whitebox Testing und das Blackbox Testing. Da wir die Entwickler der Applikation sind und somit eine Ahnung des Codes haben, ergibt es keinen Sinn das wir Blackbox Tests durchführen. Dies wurde mit der Fachvorgesetzten abgeklärt.

Bei den manuellen Tests wird ein Testkonzept verfasst.

Sitemap

Diese Sitemap soll beim Realisieren einen guten Überblick über die verschiedenen Seiten und deren Relationen verschaffen. Sie zeigt somit die logische Struktur der Webapplikation auf.

- **Login**
 - Login, Der Besucher kann sich hier in seinen Account einloggen
- **Home**
 - Hier sieht der Benutzer eine kleine Willkommens Nachricht
- **User Management** (Nur für User mit «Administratoren» Rechten zugänglich)
 - Erstellen eines Benutzers
 - Löschen eines Lehrers
 - Liste aller Lehrer
 - Kann die Kompetenzen eines einzelnen Lehrers anschauen. Damit sieht man welche Kompetenzen schon abgeschlossen sind, und welche noch offen sind.
- **Skill Management** (Nur für User mit «Administratoren» Rechten zugänglich)
 - Erstellen der Kompetenzen Liste
 - Löschen einzelner Kompetenzen
 - Editieren einzelner Kompetenzen
 - Hinzufügen eines Video-Links zu einer Sub-Kompetenz
 - Hinzufügen eines PDF-Links zu einer Sub-Kompetenz
- **Skill Management** (Nur für Lehrer)
 - Übersicht des Kompetenzen Rasters
 - Kompetenz als erledigt / abgeschlossen markieren
 - Verlinkte Videos öffnen
 - Verlinkte PDFs öffnen
 - Anzeige wie viele Kompetenzen erledigt sind in %
- **Timer** (Nur für Lehrer)
 - Erstellen eines Countdowns für Prüfungen
 - Eingabe in Minuten
 - Anzeige des Countdowns in Stunden, Minuten, Sekunden
 - Rote Anzeige wann der Countdown abgelaufen ist
- **Random Groups** (Nur für Lehrer)
 - Eingabe der Begriffe / Topics
 - Eingabe aller Schüler
 - Erstellen der Gruppen Zuteilung (Schüler werden zufällig einer Gruppe zugeteilt)
 - Übersicht der erstellten Gruppe mit den jeweiligen Mitgliedern / Schülern
- **Settings**
 - Möglichkeit den Benutzernamen zu ändern
 - Möglichkeit die E-Mail-Adresse zu ändern

Entscheiden

In der dritten Phase der IPERKA Projektplanungsmethode wird entschieden, welcher Lösungsweg verwendet wird. Dabei werden Lösungsvarianten verglichen und allfällige Risiken evaluiert.

Varianten

Wir mussten zuerst entscheiden, welches Framework und welche Datenbanken wir verwenden möchten. Entweder stellen wir uns der Herausforderung, ein komplett neues Framework und eine neue Datenbank zu verwenden, die niemand von uns kennt und keiner Erfahrung hat, oder nicht.

Für das Frontend hatten wir die Optionen Vue, React oder Angular zu verwenden. Für die Datenbank standen SQL und MongoDB infrage. Wir hätten auch die Option beide Datenbanken zu verwenden, falls wir merken, dass dies noch hilfreich sein könnte, wenn wir Relationen bräuchten.

Risiken

Das Risiko mit einer SQL-Datenbank war, dass wir erstens nicht viel Ahnung davon haben und wir nicht wissen, wie wir es in unser Projekt einbinden können und zweitens wussten wir, dass wir mit einer SQL-DB mit dem Speichern von bestimmten Daten, später Probleme haben werden. Es war klar das SQL jedoch eine bessere Variante für das Speichern von persönlichen Daten wäre.

Mit der MongoDB hatten wir im Team schon ein bisschen Erfahrung und wir wussten, dass persönliche Daten genauso gut in einer NoSQL-DB persönliche Daten speichern können. Wir wussten auch, dass uns das Speichern von Objekten einfacher fallen wird in einer MongoDB.

Das Risiko bei React und Vue waren, dass dies viel mehr Aufwand für uns sein wird, da noch niemand von dem Team damit gearbeitet hat. Das bedeutet, dass wir zuerst einmal diese Frameworks verstehen müssen und dann eine komplette funktionale Applikation damit gestalten sollten. Somit kann das dazu führen, dass wir theoretisch gar nicht fertig werden, da wir zuerst viel Zeit in das Lernen des Frameworks einsetzen müssten.

Entscheid

Aufgrund der Vorkenntnisse und der möglichen Risiken haben wir uns schlussendlich für die Kombination aus Angular, .NET Core (C#) und der MongoDB entschieden. Wie wir bereits bei den Varianten erwähnt haben, haben wir MongoDB und .NET Core (C#) am meisten bei unseren Schulprojekten verwendet und sind deshalb auch am fittesten damit. Die möglichen Risiken haben wir im Voraus abgeklärt und wirkliche Nachteile gegenüber der anderen Varianten gibt es nicht. Ebenfalls liegt bei uns in diesem Projekt der Fokus auf Qualität und Funktionalitäten der Applikation, sowie dem Erlernen der Dokumentations-Kompetenzen. Deshalb haben wir uns schlussendlich für diese Kombination entschieden.

Realisieren

In der vierten Phase der Projektmethode IPERKA geht es um die Umsetzung. Hier werden die komplexen Elemente erklärt und dargestellt. Im Code werden auch Abschnitte kommentiert, wenn dies bei der Verständlichkeit hilft.

Frontend

Wie bereits erwähnt haben wir das Frontend unserer Applikation mit Angular erstellt. Angular ist ein Front-End-JavaScript-Framework, das für die Erstellung von Webanwendungen verwendet wird. Es wurde von Google entwickelt und wird derzeit von einer Gemeinschaft von Entwicklern gepflegt.

Angular verwendet eine komponentenbasierte Architektur, das heisst es ermöglicht Entwicklern, Anwendungen zu erstellen, indem sie sie in kleinere, wiederverwendbare Komponenten aufteilen. Dadurch lassen sich komplexe Anwendungen leichter entwickeln und pflegen.

Angular wird mit TypeScript erstellt, einer Obermenge von JavaScript, die optionale statische Typisierung und andere Funktionen bietet. Dadurch sind Angular-Anwendungen einfacher zu schreiben und zu warten, insbesondere bei umfangreichen Projekten.

Angular-Anwendungen werden in der Regel mit einer Kombination aus HTML, CSS und JavaScript erstellt. Es umfasst eine Reihe von Tools und Bibliotheken, die das Erstellen und Testen von Webanwendungen erleichtern, sowie eine Vielzahl von Funktionen wie Routing, Formularvalidierung und Dependency Injection.

Einige Beispiele für Anwendungen, die mit Angular erstellt wurden, sind Google AdWords, Forbes und Upwork. Angular ist aufgrund seiner leistungsstarken Funktionen und der großen Entwicklergemeinschaft eine beliebte Wahl für die Erstellung von Webanwendungen.

Struktur:

Da wir eine übersichtliche Ordner Struktur anstreben haben wir verschiedene Stufen dafür erstellt. Die Services, Models und enums befinden sich auf der äussersten Ebene. Innerhalb des Components Ordners befinden sich alle Seiten wie Dashboard, Login, Navigation und unsere Page-Not-Found Seite. Innerhalb des Dashboard Ordners sind alle Sub-Pages, die für Lehrer und Admins wichtig sind.

Dank solch einer Struktur haben wir eine bessere Übersicht und man weiss immer, wo sich was befindet.

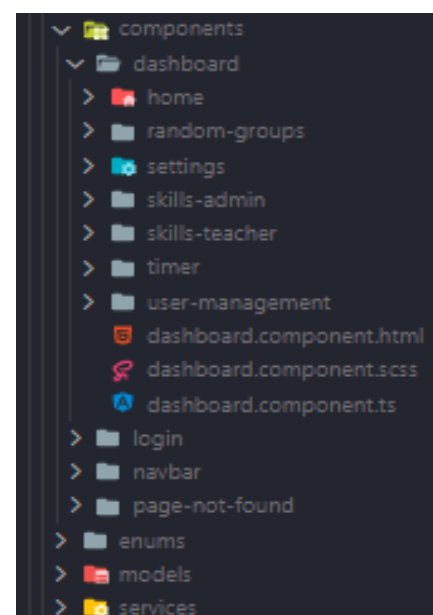


Abbildung 2 - Frontend Ordner Struktur

Design

Für das Design hatten wir nicht eine Idee. Wir haben keine Mockups erstellt, sondern einfach losprogrammiert. Da wir genaue Anforderungen hatten, wussten wir schon ziemlich gut, wie wir unsere Applikation gestalten wollen. Es war für uns direkt klar, dass wir für die Kompetenzen eine Tabelle erstellen werden und für das Editieren ein simples Formular.

Für die Farben haben wir uns an den Github Theme gewendet. Wir wollten etwas eher Dunkles mit knalligen Farben, damit es auch einfach zu erkennen ist, welche Kompetenzen bereits erledigt sind und welche noch abzuschliessen sind.

Random Groups Feature

Das Feature für zufällige Gruppeneinteilung haben wir realisiert, indem man eine Liste der Themen eingeben kann, und eine Liste der Schüler, welche zufällig zugeordnet werden sollen.

Damit diese Zuteilung auch immer zufällig passiert, haben wir folgende Methode erstellt.

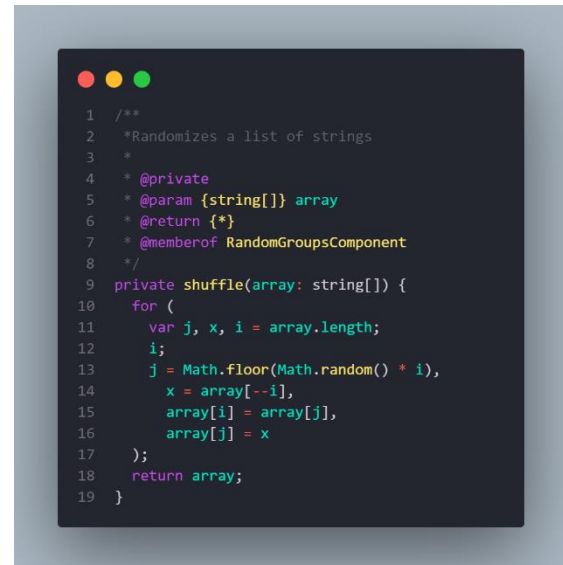
Die Funktion beginnt mit der Deklaration einer Schleife, die über die Elemente im Array iteriert. Die Schleife verwendet eine Zählvariable «i», die mit der Länge des Arrays initialisiert wird und bei jeder Iteration um eins dekrementiert wird. Die Schleife wird so lange ausgeführt, wie «i» wahr ist, daher, bis «i» 0 wird.

Innerhalb der Schleife deklariert die Funktion drei Variablen: «j», «x» und «i». «j» wird mit dem Ergebnis von `Math.floor(Math.random() * i)` initialisiert, dass eine zufällige ganze Zahl zwischen 0 und «i» (ohne «i») erzeugt. «x» wird mit dem Wert des Elements mit dem Index «i» - 1 im Array initialisiert (da «i» vor der Zuweisung um eins dekrementiert wird). Schließlich wird der Wert am Index «i» - 1 im Array durch den Wert am Index «j» ersetzt, und der Wert am Index «j» wird durch «x» ersetzt.

Nach Beendigung der Schleife wird das gemischte Array zurückgegeben. Das Mischen wird erreicht, indem man über das Array iteriert und den Wert bei jedem Index mit einem zufällig ausgewählten Wert aus dem Rest des Arrays vertauscht. Dieses Verfahren ist als Fisher-Yates-Shuffle-Algorithmus bekannt und ist eine beliebte und effiziente Methode, um ein Array zufällig zu mischen.

Die Liste der Themen und der Schüler wird beides durch diese Methode vermischt und neu zugeordnet. Nach dem Vermischen werden schrittweise alle Schüler einem Thema zugeteilt, bis die Liste der Schüler leer ist.

Am Ende wird das Eingabe Formular ausgeblendet und eine Tabelle mit den zugeteilten Schülern wird angezeigt.



```

1  /**
2   *Randomizes a list of strings
3   *
4   * @private
5   * @param {string[]} array
6   * @return {*}
7   * @memberof RandomGroupsComponent
8   */
9  private shuffle(array: string[]) {
10     for (
11         var j, x, i = array.length;
12         i;
13         j = Math.floor(Math.random() * i),
14         x = array[--i],
15         array[i] = array[j],
16         array[j] = x
17     );
18     return array;
19 }

```

Abbildung 3 - Shuffle Funktion

Timer Feature

Für das Timer Feature haben wir eine Library verwendet, welche uns die Stoppuhr zu Verfügung stellt. Die Library heisst «ngx-countdown» und wir verwenden die aktuelle Version.

Den Timer konnten wir ganz einfach im HTML code einbinden und im Typescript das Config initialisieren.

«leftTime» wird an die Input Box verlinkt. Damit kann nun ein Benutzer eine beliebige Zeit eingeben und mit dem Start Knopf den Timer starten.

```
<countdown
  #cd
  (event)="handleEvent($event)"
  [config]="config"
  class="custom-style"
>/countdown>
```

Abbildung 4 - Countdown HTML

```
public config: CountdownConfig = {
  leftTime: this.time,
  notify: [1],
  format: 'HH:mm:ss',
  prettyText: (text) => {
    return text
      .split(':')
      .map((v) => `<span class="item">${v}</span>`)
      .join('');
  },
};
```

Abbildung 5 - Countdown Config

Backend

Das Backend wurde mit C# und dem Framework ASP .NET CORE 6.0 erstellt. .NET Core ist eine plattformübergreifende, Open-Source- und modulare Version des .NET-Frameworks. Es wurde von Microsoft entwickelt, um Entwicklern die Erstellung von Anwendungen zu ermöglichen, die auf mehreren Plattformen ausgeführt werden können, darunter Windows, MacOS und Linux.

Eines der wichtigsten Merkmale von .NET Core ist, dass es modular ist, das heisst man kann wählen, welche Komponenten und Bibliotheken man in der Anwendung einbinden möchten. So kann man leichtgewichtige und effiziente Anwendungen erstellen, die nur die Funktionen enthalten, die man benötigt. .NET Core enthält auch eine Reihe von Leistungsverbesserungen und neuen Funktionen im Vergleich zum vollständigen .NET-Framework. Es hat eine schnellere Startzeit und einen geringeren Speicher-Overhead und unterstützt Funktionen wie die Side-by-Side-Installation, die es Ihnen ermöglicht, mehrere Versionen von .NET Core auf demselben Rechner ohne Konflikte zu installieren.

Struktur

Wie beim Frontend streben wir auch im Backend eine gute Ordnerstruktur an, für die leichte Erweiterbarkeit und bessere Organisation im Team. Im Backend sieht diese wie folgt aus:

Die hauptfunktionalitäten bestehen aus Controllern, Models und Services. In den Controllern werden die verschiedenen Routes definiert und die Antworten für die UI erstellt und gesendet. Dafür greift er auf die Funktionalitäten der Services zu und holt sich somit die Daten die er braucht für die Antworten der Requests. Die Models sind die Modellklassen, in denen wir die Datenspeichern und für Requests brauchen.

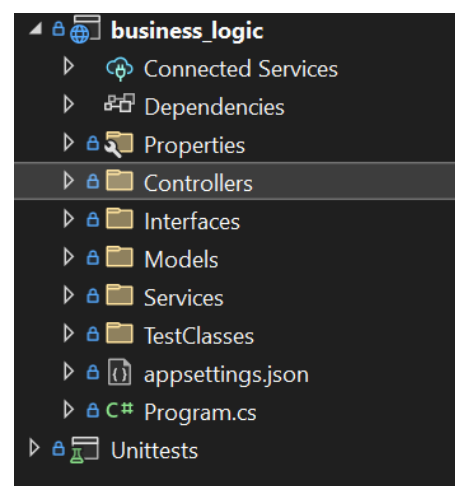


Abbildung 6 - Ordnerstruktur Backend

Controller

Der Controller ist für die Routes und Antworten der Requests verantwortlich. Ich habe mir dafür als Beispiel die Funktion um einen Benutzer zu löschen genommen.

Dieser versucht die DeleteUser Funktion des User Services aufzurufen und mit dem mitgesendeten User auszuführen. Wenn dies funktioniert, schickt es den Status 200 (Ok) ans Frontend zurück. Da, die Funktion einen Laufzeitfehler gibt, wenn es den User nicht findet, wird das in einem try {} catch{} ausgeführt. Wenn es den User also nicht findet, fängt es den Fehler ab und schickt ein 404 (Not Found) ans Frontend zurück.

```

/// <summary>
/// route to delete user in database
/// </summary>
/// <param name="user"></param>
/// <returns>status 200 or 404</returns>
[HttpPost]
[Route("delete")]
public IActionResult Delete(User user)
{
    try
    {
        _userService.DeleteUser(user);
        return Ok();
    }
    catch
    {
        return NotFound();
    }
}

```

Abbildung 7 - Delete User Controller Funktion

Service

Die Services sind die Klassen, die hinter dem Controller die Daten aus der Datenbank holen, hinzufügen oder auch editieren, sie aufbereitet und dann an den Controller schickt. Hierzu habe ich wieder die Funktion, um einen User zu erstellen genommen.

Innerhalb der Methode wird zunächst eine neue leere Liste von TeacherSkills-Objekten erstellt.

Danach wird eine Liste von TeacherSkillsMongoDb-Objekten von dem MongoCRUD abgerufen. Das MongoCrud beinhaltet die Basisfunktionen für die Interaktionen mit der Datenbank. In diesem Fall holt diese Funktion alle Skills die in der Datenbank sind.

Anschließend wird eine Schleife über die Liste der Fähigkeiten durchlaufen, und für jedes TeacherSkillsMongoDb-Objekt wird ein neues TeacherSkills-Objekt erstellt und mit den entsprechenden Eigenschaften initialisiert. Das neue TeacherSkills-Objekt wird dann der Liste von Fähigkeiten hinzugefügt.

Sobald die Schleife abgeschlossen ist, wird die Liste von Fähigkeiten dem User zugewiesen, und das Benutzerobjekt wird mit dem MongoCRUD in der Datenbank eingefügt. Die Methode gibt anschließend kein Ergebnis zurück.

Als zweites Beispiel werde ich noch die HashPassword Methode des UserService beschreiben. Die sieht wie folgt aus:

```

2 references
public void CreateUser(MongoDbUser user)
{
    //get given skills from db
    List<TeacherSkills> teacherSkills = new List<TeacherSkills>();
    List<TeacherSkillsMongoDb> skills = MongoCRUD.LoadRecords<TeacherSkillsMongoDb>("Skills");

    //iterate through skills and add to user
    foreach (TeacherSkillsMongoDb sk in skills)
    {
        TeacherSkills teacherSkill = new TeacherSkills();
        teacherSkill.SkillTopic = sk.SkillTopic;
        teacherSkill.SkillLevels = sk.SkillLevels;
        teacherSkills.Add(teacherSkill);
    }
    user.TeacherSkills = teacherSkills;
    //insert to db
    MongoCRUD.InsertRecord<MongoDbUser>(collection, user);
}

```

Abbildung 8 - Service für Nutzererstellung

Innerhalb der Methode werden zunächst zwei Byte-Arrays namens «salt» und «buffer2» erstellt. Dann wird ein Rfc2898DeriveBytes-Objekt erstellt und mit dem übergebenen Passwort, einem festgelegten Saltwert und einer festgelegten Iterationszahl initialisiert. Das Saltwert und die Iterationszahl werden verwendet, um das Hashing-Verfahren zu verstärken und zu erschweren.

Das Rfc2898DeriveBytes-Objekt wird dann verwendet, um den Saltwert und den gehashten Passwortwert in die entsprechenden Byte-Arrays zu schreiben.

```

/// <summary>
/// hash a password
/// </summary>
/// <param name="password"></param>
/// <returns>password hash</returns>
2 references
public string HashPassword(string password)
{
    //create salt for hash
    byte[] salt;
    byte[] buffer2;
    using (Rfc2898DeriveBytes bytes = new Rfc2898DeriveBytes(password, 0x10, 0x3e8))
    {
        salt = bytes.Salt;
        buffer2 = bytes.GetBytes(0x20);
    }
    byte[] dst = new byte[0x31];
    Buffer.BlockCopy(salt, 0, dst, 1, 0x10);
    Buffer.BlockCopy(buffer2, 0, dst, 0x11, 0x20);
    return Convert.ToBase64String(dst);
}

```

Abbildung 9 - HashPassword Service Funktion

Anschließend wird ein neues Byte-Array mit der Länge 0x31 erstellt und der Saltwert und der gehashte Passwortwert in das Array kopiert. Schließlich wird das Byte-Array in eine Zeichenfolge konvertiert und als Ergebnis der Funktion zurückgegeben.

Klassendiagramm

Ein User kann beliebig viele Teacherskills haben, die vom Admin definiert werden. Diese beinhalten 3 Skilllevel die dann aus weiteren Subskills bestehen, die der Lehrer dann als abgeschlossen markieren kann.

Unser Klassendiagramm sieht wie folgt aus:

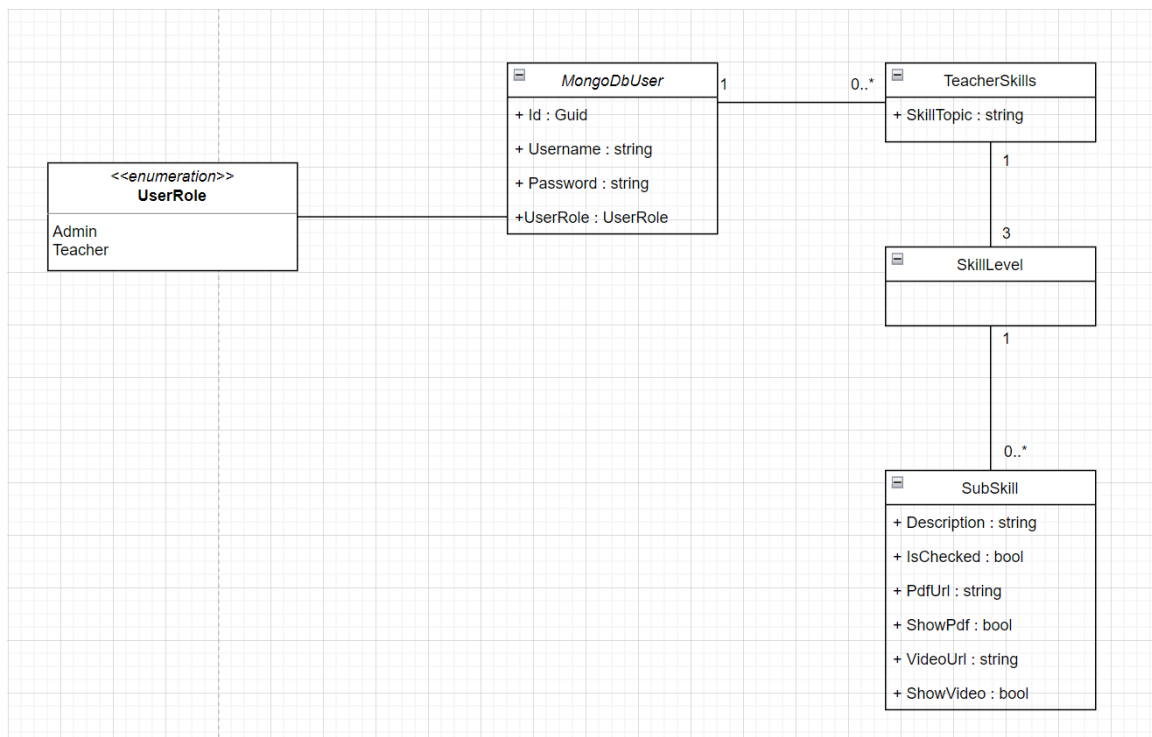


Abbildung 10 - Klassendiagramm

Sequenz Diagramm:

Unsere http Requests passieren folgendermassen. Von dem Client aus wird ein Request an die API / das Backend gemacht. Im Backend wird der Aufruf in einem Controller abgefangen und an den Service weitergeleitet. Der Service verarbeitet den Request und erledigt das nötige. Danach werden, falls nötig, die Daten in der Datenbank angepasst. Dieser Request wird dann zurück an den Controller geleitet, welcher einen http Response erstellt welcher entweder Daten, ein Success oder ein Fehler an den Client liefert. Der Client verarbeitet diese Infos und das UI wird entsprechend angepasst.

Hier ist ein Beispiel wie ein solcher Request aussehen kann.

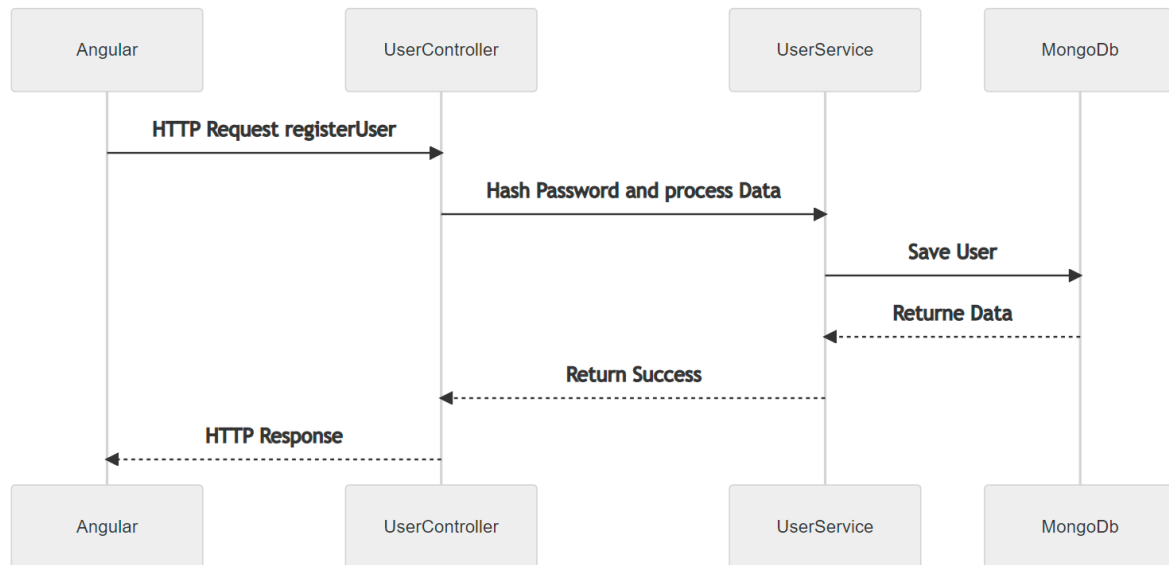


Abbildung 11 - Sequenz Diagramm

Kontrollieren

In der Phase Kontrollieren der Projektmethode IPERKA geht es darum, das Projekt auf die vorausgesetzte und erwartete Funktionalität zu testen. Hier werden Korrekturen, sowie gefundene Fehler beschrieben und die komplette Applikation aufgrund des Testprotokolls getestet.

Testkonzept

Das folgende Testkonzept soll beschreiben, wie getestet werden muss, sodass die Arbeit korrekt geprüft ist. Die Tests werden auf Home-Computern mit einem Chrome Browser (Version wird am Anfang der manuellen Tests angegeben) geprüft.

Testfall-Nr: 1	Zugriff auf Dashboard ohne einloggen
Beschreibung	Ein Benutzer versucht das Dashboard per URL zu öffnen, ohne sich vorher eingeloggt zu haben
Durchführung	<ul style="list-style-type: none"> Der Benutzer öffnet das Dashboard, in dem er direkt die URL: «"Host Domain"/Dashboard» eingibt
Erwartetes Resultat	Der Benutzer sieht einen Text, in dem steht, dass er sich zuerst anmelden soll.

Testfall-Nr: 2	Teacher Timer normal time
Beschreibung	Der Benutzer will die Funktion der Stoppuhr verwenden.
Durchführung	<ul style="list-style-type: none"> Der Benutzer meldet als Lehrer an. Über die Navigation navigiert der Benutzer auf die Stoppuhr / Timer Seite. Er gibt den Wert 10 ein und klickt den Start-Knopf.
Erwartetes Resultat	Ein Countdown von 10 Minuten beginnt

Testfall-Nr: 3	Teacher timer extreme time
Beschreibung	Der Benutzer gibt einen hohen Wert in das Eingabe Feld der Zeit ein.
Durchführung	<ul style="list-style-type: none"> Der Benutzer meldet sich über das Login an. Über die Navigation navigiert der Benutzer auf die Stoppuhr / Timer Seite. Er gibt den Wert 1'000'000 ein und klickt den Start-Knopf.
Erwartetes Resultat	Eine Fehlermeldung wird angezeigt, die hinweist, dass die Eingabe zu hoch sei

Testfall-Nr: 4	Teacher timer negative test
Beschreibung	Der Benutzer gibt einen negativen Wert in das Eingabefeld der Zeit ein.
Durchführung	<ul style="list-style-type: none"> Der Benutzer meldet als Lehrer an. Über die Navigation navigiert der Benutzer auf die Stoppuhr / Timer Seite. Er gibt den Wert -10 ein und klickt den Start-Knopf.
Erwartetes Resultat	Eine Fehlermeldung wird angezeigt, die hinweist, dass die Eingabe ungültig ist.

Testfall-Nr: 5	Admin Create a new Skill
Beschreibung	Ein Administrator erfasst eine neue Kompetenz.
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet sich als Administrator an und navigiert auf die Kompetenzen Seite. • In dem Add Skills Form erfasst er ein neues Topic: «Word». • Für die Beginner Stufe erfasst er eine Kompetenz «Kennt Word basis Funktionen». • Klickt add Subskill • Klickt auf Advanced. • Für die Advanced Stufe erfasst er eine Kompetenz «Kennt Word Keyboardshortcuts wie ctrl + s --> Save» • Klickt Add Subskill • Klickt auf Expert • Für Expert erfasst er eine neue Kompetenz «Kann Tabellen in Word erstellen und schön Formatieren» und fügt einen Video Link hinzu: «https://www.youtube.com/watch?v=FcVEVmY7tBk» • Klickt Add Subskill • Er Klickt auf Save
Erwartetes Resultat	Die erfassten Kompetenzen werden in der linken Tabelle richtig angezeigt und das Formular wird «zurück» gesetzt

Testfall-Nr: 6	Admin Edit a Skill
Beschreibung	Ein Administrator will eine bereits existierende Kompetenz anpassen, da sich die URL für ein Video geändert hat.
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet sich als Administrator an. • Er navigiert auf die Kompetenzen Seite. • Klickt auf Edit bei der Englisch Kompetenz. • In dem Edit Skills Form klickt er auf die Grammar Kompetenz. • Der Benutzer editiert die existierende Video URL auf: «https://www.youtube.com/watch?v=O9YRy8m1Rf8». • Klickt aus Save Subskill. • Klickt auf Save.
Erwartetes Resultat	Wenn der Benutzer nun auf den Videobutton klickt, bei der Grammar Kompetenz landet er auf der neuen Seite, die eingetragen wurde.

Testfall-Nr: 7	Admin delete a Skill
Beschreibung	Ein Administrator will eine bereits existierende Kompetenz löschen.
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet sich als Administrator an. • Er navigiert auf die Kompetenzen Seite. • Klickt auf delete bei der Englischkompetenz.
Erwartetes Resultat	Eine grüne Success Meldung erscheint, die das erfolgreiche Löschen bestätigt.

Testfall-Nr: 8	Admin delete User
Beschreibung	Ein Administrator will eine Lehrperson entfernen da diese nicht mehr an der Schule arbeitet.
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet sich als Administrator an. • Er navigiert auf die User Management Seite. • Klickt auf delete bei der gewünschten Lehrperson.
Erwartetes Resultat	Die Lehrerliste sollte sich anpassen und eine grüne Meldung erscheint, die das erfolgreiche Löschen bestätigt.

Testfall-Nr: 9	Admin create User
Beschreibung	Ein Administrator will eine neue Lehrperson hinzufügen, da diese neu an der Schule arbeitet.
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet sich als Administrator an. • Er navigiert auf die User Management Seite. • Klick auf Add User • Er füllt das Formular wie folgt aus: <ul style="list-style-type: none"> ○ Username: maxmustermann ○ Email address: maxmusterman@online.gibz.ch ○ Is Admin bleibt, wie es ist, da die Person ein Lehrer ist und nicht ein Administrator ○ Password: test123! • Klick auf Save
Erwartetes Resultat	Eine grüne Success Meldung erscheint, die das Erstellen der Lehrperson bestätigt. Der neue Benutzer wird nun in der Lehrerliste angezeigt.

Testfall-Nr: 10	Teacher create random Groups
Beschreibung	Der Benutzer will zufällig generierte Gruppen erstellen
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet als Lehrer an. • Über die Navigation navigiert der Benutzer auf die Current Skills / Gehirn Symbol Seite. • Er füllt das Form wie folgt aus <ul style="list-style-type: none"> ○ Topics: DoS and DDoS, MITM, Phishing, Ransomware, SQL Injection, XSS ○ Students: Student1, Student2, Student3, Student4, Student5, Student6 • Klick auf Generate
Erwartetes Resultat	<p>Eine Überschrift mit dem Inhalt «Number of Groups:» erscheint, welche den Wert 6 ausgeben soll.</p> <p>Eine Liste, welche auf der linken Seite die Topics und auf der rechten Seite die Students ausgibt, soll angezeigt werden.</p> <p>Bei erneutem Wiederholen des Tests soll die Reihenfolge der Topics und Students unterschiedlich sein</p>

Testfall-Nr: 11	Teacher check skill
Beschreibung	Der Benutzer will eine Kompetenz auf erledigt setzen
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet als Lehrer an. • Über die Navigation navigiert der Benutzer auf die Current Skills / Gehirn Symbol Seite. • Der Benutzer hovort nun über die Subkompetenz «Grammer» bis diese grün hervorgehoben ist. • Klickt nun auf die hervorgehobene Subkompetenz • Der Benutzer bewegt seine Maus von der Subkompetenz weg
Erwartetes Resultat	<p>Die Subkompetenz «Grammer» soll nun dauerhaft Grün markiert sein.</p> <p>Bei der Angabe «You have mastered % of all Skills» sollte sich jetzt die Prozentzahl erhöht haben.</p>

Testfall-Nr: 12	Teacher uncheck skill
Beschreibung	Der Benutzer will eine Kompetenz von erledigt auf nicht erledigt setzen
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet als Lehrer an. • Über die Navigation navigiert der Benutzer auf die Current Skills / Gehirn Symbol Seite. • Der Benutzer hovort nun über die Subkompetenz «Grammar» bis diese rot hervorgehoben ist. • Klickt nun auf die hervorgehobene Subkompetenz • Der Benutzer bewegt seine Maus der Subkompetenz weg
Erwartetes Resultat	<p>Die Markierung der Subkompetenz «Grammer» soll nun aufgehoben sein.</p> <p>Bei der Angabe «You have mastered % of all Skills» sollte sich nun die Prozentzahl verringert haben.</p>

Testfall-Nr: 13	Teacher view PDF and Video
Beschreibung	Der Benutzer will das zugehörige PDF und Video zu einer Kompetenz öffnen
Durchführung	<ul style="list-style-type: none"> • Der Benutzer meldet als Lehrer an. • Über die Navigation navigiert der Benutzer auf die Current Skills / Gehring Symbol Seite. • Der Benutzer hovert nun über den Video-Button der Subkompetenz «Grammer» • Klickt den Button • Der Benutzer navigiert in den Staffbite-Tab zurück • Der Benutzer hovert nun über den PDF-Button der Subkompetenz «Grammer» • Klickt den Button
Erwartetes Resultat	<p>Nach dem Klick auf den Video-Button der Subkompetenz «Grammer» soll sich das YouTube-Video in einem neuen Tab öffnen.</p> <p>Nach dem Klick auf den PDF-Button der Subkompetenz «Grammer» soll sich das PDF-File in einem neuen Tab öffnen.</p>

Unit Tests & UI Tests

Unit Tests

Die Unit Tests wurden mit dem Framework «xunit» erstellt und haben alle funktioniert.

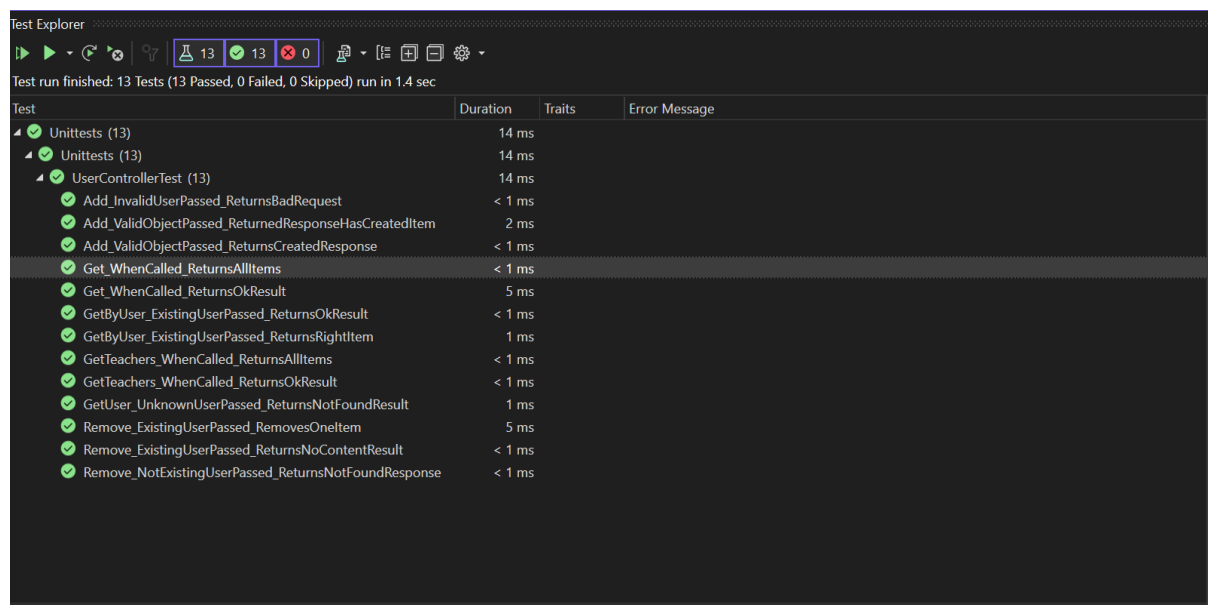


Abbildung 12 - Passing Unit Tests

UI Tests

UI-Tests wurden mit einem Framework namens Cypress erstellt

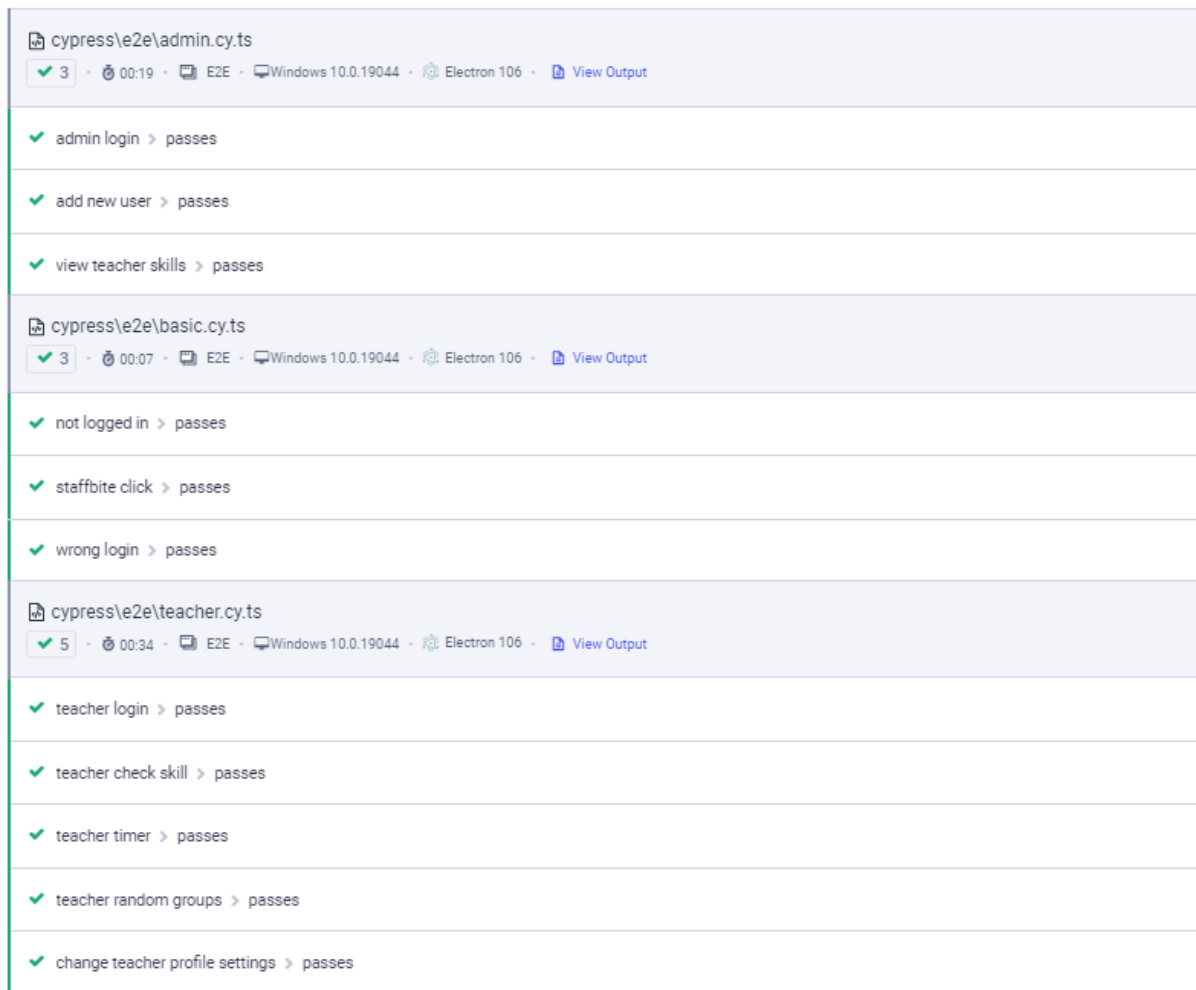


Abbildung 13 - UI Passing Tests

Manuelle Tests

Name	Dario PC
Prozessor	Ryzen 9 5950X
Ram	64 GB
Browser	Chrome Browser Version: 108
Betriebssystem	Windows 11 Pro 64-Bit (Version 21H2)
Grafikkarte	Nvidia RTX 4080

Name	Mario PC
Prozessor	Corie i9-9900K
Ram	16GB
Browser	Chrome Browser Version 108.0.5359.98 (Offizieller Build) (64-Bit)
Betriebssystem	Windows 11 Pro 64-Bit (Version 21H2)
Grafikkarte	GTX 1080

Manuelle Tests

Protokoll: 1 – Testfall-Nr: 1	Zugriff auf Dashboard ohne einloggen
Erhaltenes Resultat	<ul style="list-style-type: none"> • Das Dashboard direkt über die URL: «https://staffbite.vercel.app/dashboard» geöffnet • Es wird ein Text ausgegeben in dem steht, dass ich mich zuerst anmelden soll.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 2	Teacher timer normal time
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Stoppuhr / Timer Seite navigiert • Den Wert 10 in die Eingabebox eingegeben • Beginknopf gedrückt • Der Countdown läuft nun von 00 10 00 herunter
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 3	Teacher timer extreme time
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Stoppuhr / Timer Seite navigiert. • Dort den Wert 1'000'000 eingegeben • Auf den Begin Knopf geklickt • Es wird nun eine Fehlermeldung angezeigt, welche besagt, dass der eingegebene Wert zu hoch ist.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 4	Teacher timer negative test
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Stoppuhr / Timer Seite navigiert. • Dort den Wert -10 eingegeben • Auf den Beginknopf geklickt • Es wird nun eine Fehlermeldung angezeigt, welche besagt, dass der eingegebene Wert zu ungültig ist.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 5	Admin Create a new Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Administratorlogin angemeldet • Anschliessend über die Navigation auf die Kompetenzen / Gehirn Icon Seite navigiert. • In dem Add Skills Form ein neues Topic erfasst: «Word». • Für die Beginner Stufe eine Sub-Kompetenz erfasst: «Kennt Word basis Funktionen». • Auf Add subskill geklickt • Auf Advanced geklickt • Für die Advanced Stufe eine Sub-Kompetenz erfasst: «Kennt Word Keyboardshortcuts wie ctrl + s --> Save» • Auf Add subskill geklickt • Auf Expert geklickt • Für Expert eine neue Kompetenz erfasst: «Kann Tabellen in Word erstellen und schön Formatieren» und fügt einen Video Link hinzu: «https://www.youtube.com/watch?v=FcVEVmY7tBk» • Auf Add subskill geklickt • Auf Save geklickt • Die neu erfasste Komptenz: «Word» erscheint nun link in der Kompetenzenliste. • Unter dem Punkt Beginner ist eine Subkompetenz names «Kennt Word basis Funktionen» erfasst • Unter dem Punkt Advanced ist eine Subkompetenz names «Kennt Word Keyboardshortcuts wie ctrl + s --> Save» erfasst • Unter dem Punkt Expert ist eine Subkompetenz namens «Kann Tabellen in Word erstellen und schön Formatieren» erfasst, ebenso ist der Videobutton ersichtlich welcher auf den Link «https://www.youtube.com/watch?v=FcVEVmY7tBk» verweist.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 6	Admin Edit a Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Administratorlogin angemeldet • Anschliessend über die Navigation auf die Kompetenzen / Gehirn Icon Seite navigiert. • Auf edit bei der Englischkompetenz geklickt • In dem Edit Skills Form auf die Grammar Kompetenz geklickt • Die existierende Video URL editiert auf: «https://www.youtube.com/watch?v=O9YRy8m1Rf8». • Klickt aus Save subskill. • Klickt auf Save.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 7	Admin delete a Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Administratorlogin angemeldet • Anschliessend über die Navigation auf die Kompetenzen / Gehirn Icon Seite navigiert. • Auf delete bei der Englischkompetenz gedrückt • Success-Meldung erscheint in Grün mit der Nachricht, dass die Kompetenz gelöscht wurde • Seite neu geladen, der Skill wird nicht mehr in der Kompetenzen liste angezeigt • Kompetenz wurde von der DB gelöscht
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 8	Admin delete User
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Administratorlogin angemeldet • Anschliessend über die Navigation auf die Management / 3 Personen Icon Seite navigiert. • Auf delete bei der Person Alfred gedrückt <ul style="list-style-type: none"> ○ Username: Alfred • Success-Meldung erscheint in Grün mit der Nachricht, dass der User gelöscht wurde • Seite neu geladen, der User wird nicht mehr in der Liste angezeigt • User wurde von der DB gelöscht
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 9	Admin create User
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Administratorlogin angemeldet • Anschliessend über die Navigation auf die Management / 3 Personen Icon Seite navigiert. • Auf Add User geklickt • Das Form wie folgt ausgefüllt: <ul style="list-style-type: none"> ○ Username: maxmustermann ○ Email address: maxmusterman@online.gibz.ch ○ Is Admin bleibt wie, es ist, da die Person ein Lehrer ist und nicht ein Administrator ○ Password: test123! • Auf Save geklickt • Success-Meldung erscheint in Grün mit der Nachricht, dass der User erstellt wurde • Seite neu geladen, der User wird nun in der Liste angezeigt • User wurde in der DB erstellt
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 10	Teacher create random Groups
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Random Groups / 3 Personen Symbol Seite. • Das Form wie folgt ausgefüllt: <ul style="list-style-type: none"> ○ Topics: DoS and DDoS, MITM, Phishing, Ransomware, SQL Injection, XSS ○ Students: Student1, Student2, Student3, Student4, Student5, Student6 • Auf Generate geklickt • B der Überschrift «Number of Groups» wird der Wert 6 ausgegeben • Eine Tabelle mit 2 Listen erscheint, in der rechten Liste werden die Topics durchmischt angezeigt und in der linken List werden die Students durchmischt, angezeigt. • Ich habe die Schritte erneut ausgeführt, um zu sehen, ob sich die Reihenfolge der Students und Topcis ändert. • Die Reihenfolge der Topics und Students hat sich geändert.
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 11	Teacher check skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Current Skills / Gehirn Symbol Seite. • Über die Subkompetenz «Grammer» gehovert, diese ist dann grün hervorgehoben worden • Auf die hervorgehobene Subkompetenz geklickt • Meine Maus von der geklickten Subkompetenz wegbewegt • Die Subkompetenz «Grammer» ist nun grün markiert. • Bei der Angabe «You have mastered % of all Skills» hat sich der Wert von 0 % auf 14 % erhöht
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 12	Teacher uncheck skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Current Skills / Gehirn Symbol Seite. • Über die Subkompetenz «Grammer» gehovert, diese ist dann rot hervorgehoben worden. • Auf die hervorgehobene Subkompetenz geklickt • Meine Maus von der geklickten Subkompetenz wegbewegt • Die Subkompetenz «Grammer» ist nun nicht mehr markiert. • Bei der Angabe «You have mastered % of all Skills» hat sich der Wert von 14 % auf 0 % verringert
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 1 – Testfall-Nr: 13	Teacher view PDF and Video
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich mit meinem Lehrerlogin angemeldet • Anschliessend über die Navigation auf die Current Skills / Gehirn Symbol Seite. • Über den Video-Button der Subkompetenz «Grammer» gehovert • Auf den Button geklickt • Nach dem Klick auf den Video-Button hat sich der Video-Link in einem neuen Tab geöffnet • Auf den Staffbite-Tab zurück, navigiert • Über den PDF-Button der Subkompetenz «Grammer» gehovert • Auf den Button geklickt • Nach dem Klick auf den PDF-Button hat sich das PDF-File in einem neuen Tab geöffnet
Status	Bestanden
Datum	02.12.2022
Getestet von	Mario Kopp

Protokoll: 2 – Testfall-Nr: 1	Zugriff auf Dashboard ohne einloggen
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe im Edge Browser das Dashboard direkt über die URL: «https://staffbite.vercel.app/dashboard» geöffnet • Es wird ein Text angezeigt in dem steht, dass ich mich zuerst anmelden soll.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 2	Teacher timer normal time
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich als Lehrer angemeldet • Anschliessend über die Navigation auf die Stoppuhr / Timer Seite navigiert • Den Wert 10 in die Eingabebox eingegeben • Start Knopf gedrückt • Der Countdown läuft nun von 00 10 00 herunter
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 3	Teacher timer extreme time
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich als Lehrer angemeldet • Anschliessend über die Navigation auf die Stoppuhr / Timer Seite navigiert. • Den Wert 1'000'000 eingegeben • Start Knopf gedrückt • Es wird eine Fehlermeldung angezeigt, welche mich darauf hinweist, dass der eingegebene Wert zu hoch ist.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 4	Teacher timer negative test
Erhaltenes Resultat	<ul style="list-style-type: none"> • Ich habe mich als Lehrer angemeldet • Über die Navigation auf die Stoppuhr / Timer Seite navigiert. • Den Wert -10 eingegeben • Auf den Startknopf geklickt • Es wird eine Fehlermeldung angezeigt, auf der steht, dass der eingegebene Wert ungültig ist.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 5	Admin Create a new Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Admin angemeldet • Über die Navigation auf die Kompetenzen Seite navigiert. • In dem Add Skills Form ein neues Topic erfasst: «Word». • Für die Beginner Stufe eine Sub-Kompetenz erfasst: «Kennt Word basis Funktionen». • Auf Add subskill geklickt • Auf Advanced geklickt • Für die Advanced Stufe eine Sub-Kompetenz erfasst: «Kennt Word Keyboardshortcuts wie ctrl + s --> Save» • Auf Add subskill geklickt • Auf Expert geklickt • Für Expert eine neue Kompetenz erfasst: «Kann Tabellen in Word erstellen und schön Formatieren» und fügt einen Video-Link hinzu: «https://www.youtube.com/watch?v=FcVEVmY7tBk» • Auf Add subskill geklickt • Auf Save geklickt • Die erfasste Kompetenz: «Word» erscheint nun in der Kompetenzenliste.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 6	Admin Edit a Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Admin angemeldet • Über die Navigation auf die Kompetenzen Seite navigiert. • Auf edit bei der Englisch Kompetenz geklickt • In dem Edit Skills Form auf die Grammarkompetenz geklickt • Die existierende Video URL editiert auf: «https://www.youtube.com/watch?v=O9YRy8m1Rf8». • Auf Save subskill geklickt. • Auf Save geklickt
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 7	Admin delete a Skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Admin angemeldet • Anschliessend über die Navigation auf die Kompetenzen Seite navigiert. • Auf delete bei der Englischkompetenz gedrückt • Success-Meldung erscheint in Grün mit der Nachricht, dass die Kompetenz gelöscht wurde • Kompetenz wird nicht mehr in der Kompetenzenliste angezeigt.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 8	Admin delete User
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Admin angemeldet • Auf die User Management Seite navigiert. • Auf delete bei dem Lehrer Alfred gedrückt <ul style="list-style-type: none"> ○ Username: Alfred • Grüne Success-Meldung erscheint mit der Nachricht, dass der User gelöscht wurde • User wird nicht mehr in der Userliste angezeigt.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 9	Admin create User
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Admin angemeldet • Auf die User-Management Seite navigiert. • Auf Add User geklickt • Das Form wie folgt ausgefüllt: <ul style="list-style-type: none"> ○ Username: maxmustermann ○ Email address: maxmusterman@online.gibz.ch ○ Is Admin bleibt wie, es ist, da die Person ein Lehrer ist und nicht ein Administrator ○ Password: test123! • Auf Save geklickt • Grüne Success-Meldung erscheint mit der Nachricht, dass der User erstellt wurde • User wird nun in der Userliste angezeigt.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 10	Teacher create random Groups
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Lehrer angemeldet • Auf die Random Groups Seite navigiert. • Das Formular wie folgt ausgefüllt: <ul style="list-style-type: none"> ○ Topics: DoS and DDoS, MITM, Phishing, Ransomware, SQL Injection, XSS ○ Students: Student1, Student2, Student3, Student4, Student5, Student6 • Auf Generate geklickt • Bei der Überschrift «Number of Groups» wird der Wert 6 ausgegeben • Eine Tabelle mit 2 Listen erscheint. In der rechten Liste werden die Topics durchmischt angezeigt und in der linken Liste werden die Schüler durchmischt angezeigt. • Ich habe die Schritte erneut ausgeführt, um zu sehen, ob die Zuteilung auch wirklich zufällig ist. • Die Reihenfolge der Topics und Students war anders als bei Versuch Nr.1.
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 11	Teacher check skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Lehrer angemeldet • Über die Navigation auf die Current Skills Seite navigiert. • Über die Subkompetenz «Grammar» gehovert, diese ist dann grün hervorgehoben worden • Auf die hervorgehobene Subkompetenz geklickt • Meine Maus von der geklickten Subkompetenz wegbewegt • Die Subkompetenz «Grammar» ist nun grün markiert. • Bei der Angabe «You have mastered % of all Skills» hat sich der Wert von 0 % auf 14 % erhöht
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 12	Teacher uncheck skill
Erhaltenes Resultat	<ul style="list-style-type: none"> • Als Lehrer angemeldet • Anschliessend über die Navigation auf die Current Skills Seite navigiert. • Über die Subkompetenz «Grammar» gehovert. Diese ist dann Rot hervorgehoben worden. • Auf die hervorgehobene Subkompetenz geklickt • Meine Maus von der geklickten Subkompetenz wegbewegt • Die Subkompetenz «Grammar» ist nun nicht mehr markiert. • Bei der Angabe «You have mastered % of all Skills» hat sich der Wert von 14 % auf 0 % verringert
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Protokoll: 2 – Testfall-Nr: 13	Teacher view PDF and Video
Erhaltenes Resultat	<ul style="list-style-type: none">• Als Lehrer angemeldet• Auf die Current Skills Seite navigiert.• Über den Video-Button der Subkompetenz «Grammer» gehovort• Auf den Button geklickt• Nach dem Klick auf den Video-Button hat sich der Video-Link in einem neuen Tab geöffnet• Auf den Staffbite-Tab zurücknavigiert• Über den PDF-Button der Subkompetenz «Grammer» gehovort• Auf den Button geklickt• Nach dem Klick auf den PDF-Button hat sich das PDF-File in einem neuen Tab geöffnet
Status	Bestanden
Datum	02.12.2022
Getestet von	Dario Schaffner

Benutzerhandbuch

Das Benutzerhandbuch sollte alle Interaktionen und Funktionalitäten des UI verdeutlichen. Bei der Planung wurde auf ein benutzerfreundliches und selbsterklärendes UI geachtet, zu Vollständigkeit und im Falle von Unklarheiten steht jedoch dieses Benutzerhandbuch zur Verfügung.

Erstellen von Kompetenzen (Nur als Admin möglich)

Als Admin hat man die Möglichkeit Skills zu erstellen. Diese können dann individuell abgearbeitet werden.

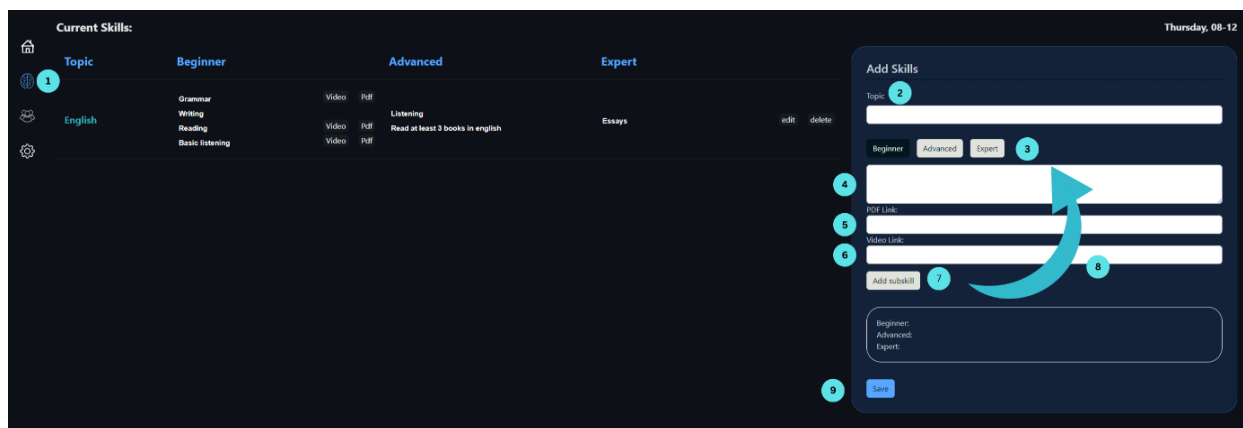


Abbildung 14 - Erstellen von Kompetenzen

1. Unter dem Menüpunkt «Gehirn» findet man als Admin die Übersicht der aktuellen Kompetenzen vor. Hier ist es möglich, neue Kompetenzen zu erstellen und bestehende Skills zu editieren oder löschen.
2. Hier wird der Name für die neue Kompetenz festgelegt, dieser wird zum Schluss unter dem Punkt Topic angezeigt.
3. Es gibt 3 verschiedene Schwierigkeitsstufen -> Anfänger, Fortgeschritten, Experte. Pro Stufe können mehrere Sub-Skills erfasst werden
4. In diesem Feld wird der Name der Subkompetenz erfasst
5. Hier wird das benötigte PDF, welches zum Erarbeiten dieser Subkompetenz benötigt wird, abgelegt
6. Hier wird der benötigte Videolink, welcher zum Erarbeiten dieser Subkompetenz benötigt wird, abgelegt
7. Mit diesem Button kann die Subkompetenz der Kompetenz hinzugefügt werden.
8. Um eine Kompetenz erfolgreich abzuspeichern, muss jede Schwierigkeitsstufe mindestens eine Subkompetenz enthalten. Dazu müssen die Schritte 3-7 mit jeder Schwierigkeitsstufe durchlaufen werden.
9. Mit diesem Button kann die Kompetenz und die damit verknüpften Subkompetenzen gespeichert werden

Editieren von Kompetenzen (Nur als Admin möglich)

Neue Subkompetenzen hinzufügen

Als Admin hat man die Möglichkeit neue Subkompetenzen zu bestehenden Kompetenzen hinzuzufügen

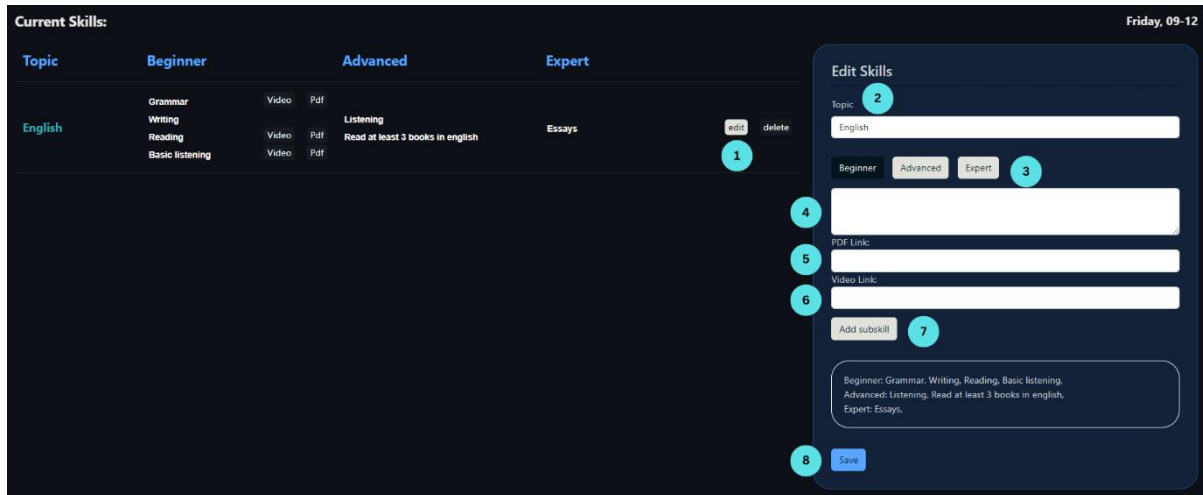


Abbildung 15 - Editieren von Kompetenzen

1. Bei der zu bearbeitenden Kompetenz den «Edit» Button klicken
2. Der Name der Kompetenz kann beliebig abgeändert werden
3. Die Schwierigkeitsstufe der neuen Subkompetenz wird hier festgelegt
4. In diesem Feld wird der Name der Subkompetenz erfasst
5. Hier wird das benötigte PDF, welches zum Erarbeiten dieser Subkompetenz benötigt wird, abgelegt
6. Hier wird der benötigte Videolink, welcher zum Erarbeiten dieser Subkompetenz benötigt wird, abgelegt
7. Mit diesem Button kann die Subkompetenz der Kompetenz hinzugefügt werden.
8. Mit diesem Button kann die Kompetenz und die damit verknüpften Subkompetenzen gespeichert werden

Editieren einer bestehenden Subkompetenz

Als Admin hat man die Möglichkeit bestehende Subkompetenzen abzuändern

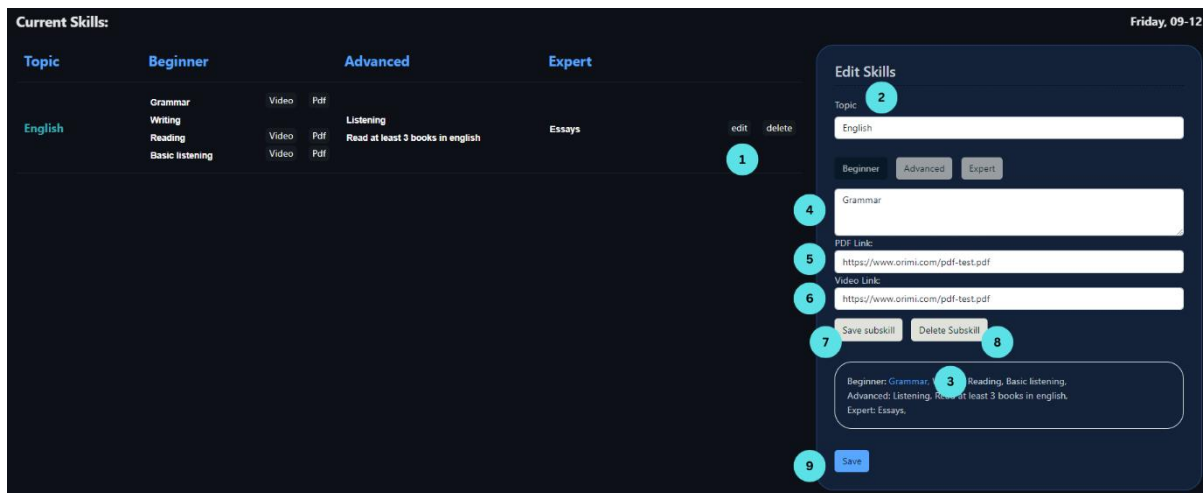


Abbildung 16 - Editieren einer bestehenden Subkompetenz

1. Bei der zu bearbeitenden Kompetenz den «Edit» Button klicken
2. Der Name der Kompetenz kann beliebig abgeändert werden
3. Hier wird eine bestehende Subkompetenz mit einem Buttonklick selektiert.
4. In diesem Feld kann der Name der Subkompetenz abgeändert werden
5. Hier kann das benötigte PDF, welches zum Erarbeiten dieser Subkompetenz benötigt wird, abgeändert werden
6. Hier kann der benötigte Videolink, welcher zum Erarbeiten dieser Subkompetenz benötigt wird, abgeändert werden
7. Mit diesem Button kann die Subkompetenz der Kompetenz gespeichert werden.
8. Mit diesem Button kann die Subkompetenz der Kompetenz gelöscht werden.
9. Mit diesem Button kann die Kompetenz und die damit verknüpften Subkompetenzen gespeichert werden

Subkompetenzen auf den Status erledigt setzen

Als User hat man die Möglichkeit Kompetenzen auf erledigt zu setzen.

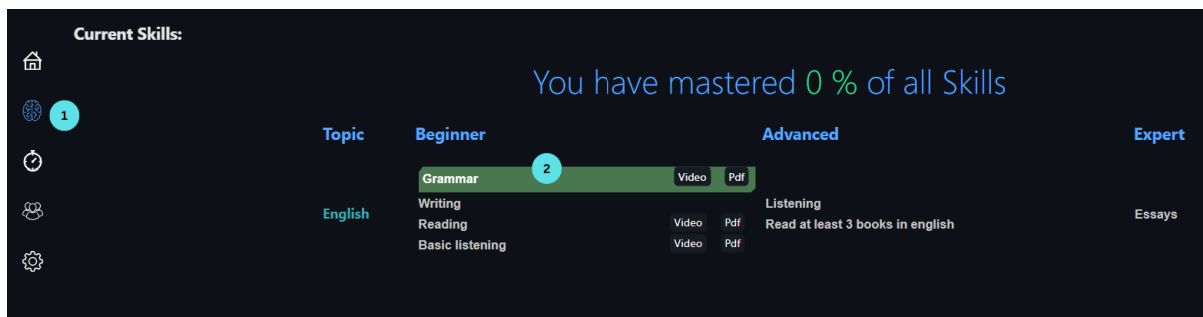


Abbildung 17 - Subkompetenzen auf den Status erledigt setzen

1. Unter dem Menüpunkt «Gehirn» findet man als Lehrer die Übersicht der aktuell zu erledigenden Kompetenzen vor. Hier ist es möglich, Subkompetenzen auf den Status erledigt gesetzt werden.
2. Durch einen Klick auf die Subkompetenz kann diese auf erledigt gesetzt werden, durch erneutes klicken kann diese auch wieder auf «To be done» gesetzt werden

Erstellen von zufälligen Gruppen

Als User hat man die Möglichkeit zufällige Gruppen zu erstellen

Random Groups

Generate new random Groups

Enter all topics and student names and separate them with a comma.

Example: DoS and DDoS, MITM, Phishing, Ransomware, SQL Injection, XSS

Topics **2**

DoS and DDoS, MITM, Phishing, Ransomware, SQL Injection, XSS

Students **3**

Schüler1, Schüler2, Schüler3, Schüler4, Schüler5, Schüler6

Generate **4**

Abbildung 18 - Erstellen von zufälligen Gruppen

1. Unter dem Menüpunkt 4 («3 Personen») können zufällige Gruppen erstellt werden
2. Hier können die Themen/Namen der Teams eingegeben werden. Es ist wichtig, dass die Themen/Namen mit einem «,» getrennt werden. Die Themen/Namen können mit oder ohne Abstand zwischen den Kommas eingegeben werden.
3. Hier können die Namen der Schüler eingegeben werden. Es ist wichtig, dass die Namen mit einem «,» getrennt werden. Die Namen können mit oder ohne Abstand zwischen den Kommas eingegeben werden.
4. Mit diesem Button können die Gruppen erstellt werden

Auswerten

Das Auswerten ist der letzte Schritt der IPERKA Projektplanungsmethode. In diesem Teil wird reflektiert, Erfolge und Misserfolge erkannt, sowie Verbesserungsmöglichkeiten gesucht und zum Schluss ein Fazit daraus gezogen.

Reflexion

Das Projekt hat im Grossen und Ganzen reibungslos geklappt, auf wirkliche Probleme sind wir zum Glück nicht gestossen. Ein grosser Teil hat sicherlich auch die ausführliche Planung dazu beigetragen, sowie auch die Kommunikation im Team.

Das Zusammenarbeiten im Team hat ebenfalls gut funktioniert, so gab es auch sehr wenige Unstimmigkeiten und Differenzen. Das hat sicherlich auch damit zu tun, dass wir ein gemeinsames Ziel hatten, welches wir anstrebten und auch durchgezogen haben.

Erfolge

Unser grösster Erfolg ist, dass wir all unsere Ziele erreicht haben. Wir konnten alle geplanten Features umsetzen und testen. Zudem konnten wir noch weitere Features / Optimierungen von Features umsetzen. Wir konnten alle offenen Fragen stets gegenseitig beantworten und falls wir Hilfe brauchten, war die Lehrperson stets zur Hilfe da. Der detaillierte Zeitplan hat uns geholfen strukturiert zu unserem Ziel zu kommen. Dieser war oft sehr akkurat und eine gute Orientierung um weitere Schritte anzugehen.

Misserfolge

Misserfolge gab es nicht viele, da wir sehr fleissig gearbeitet, geplant und kommuniziert haben. Dennoch gab es Punkte, bei welchen Verbesserungspunkte vorhanden sind. Eigentlich hatten wir uns vorgenommen, den letzten Tag ausschliesslich für das Review unserer Arbeit zu reservieren. Da wir noch ein paar kleinere Pendenzen offen hatten, hat dies nicht ganz funktioniert. Ein weiterer Punkt, bei welchem wir uns noch verbessern können, ist die Eintragung der Ist-Zeit im Zeitplan. Wir haben die Ist-Zeiten zum Schlussprojekt aufgrund des Journals eingetragen, was nicht ganz optimal ist.

Zum Schluss kommen wir noch zu einem Punkt, der unserer Meinung nach kein Misserfolg ist, jedoch noch eine weitere Möglichkeit der Ausführung dieses Projektes bietet. Wir haben die Funktionen der einzelnen Projektmitglieder in diesem Projekt konkret aufgeteilt. Jedes Projektmitglied war für einen spezifischen Teilbereich zuständig. Man könnte bei der nächsten Arbeit die Teilbereiche in einzelne Aufgaben unterteilen und anschliessend die Aufgaben eines Teilbereichs unter den Projektmitgliedern aufteilen.

Verbesserungsmöglichkeiten

Aufgrund der Punkte, welche wir bei den Misserfolgen aufgelistet hatten, haben wir nun eine Liste mit Verbesserungsmöglichkeiten erstellt.

- Projekt 1 Woche vor Schluss komplett abschliessen, damit wir den letzten Tag ausschliesslich zum Reviewen und Überprüfen der Kriterienlisten zur Verfügung haben
- Die Ist-Zeit im Zeitplan jeden Tag kurz nach der Lektion anpassen und nicht basierend auf dem Journal die Ist-Zeit ausfüllen.
- Zum Testen, ob dies eine höhere Effizienz bringt, könnte man beim nächsten Projekt, die Teilbereiche in verschiedene Aufgaben unterteilen und diese anschliessend zwischen den Projektmitgliedern aufteilen.

Fazit

Mario

Ziel dieses Produktes war es, der Schulleitung eine Übersicht über die Kompetenzen der Lehrer zugeben. Sowie der Schulleitung zu ermöglichen ihre eigenen Kompetenzen, in Form eines Kompetenzrasters zu definieren, welche die Lehrer dann abarbeiten können. Zusätzlich war es das Ziel den Lehrern eine Selektion von Features zu errichten, mit welchen sie ihren Lehreralltag vereinfachen können.

All diese Ziele haben wir erreicht, somit kann zusammenfassend gesagt werden, dass unser Endprodukt ein Erfolg ist. Das Ganze lief vom Projektantrag an bis zur Abgabe so weit reibungslos, was sicherlich darauf zurückzuführen ist, dass wir uns im Team gut verstanden und gut zusammen kommuniziert haben. Ich habe nun zum ersten Mal eine Praxis ähnliche Projekt-Arbeit (IPA) durchlaufen, fachlich habe ich sicherlich einiges in der Dokumentation solcher Arbeiten dazugelernt. Die Fähigkeit, ein Projekt korrekt zu dokumentieren, wird sicherlich noch einige Male von Nutzen sein.

Von der Testing Seite her, habe ich bis jetzt mehrheitlich nur Unit-Tests geschrieben. Im Verlaufe dieses Projektes habe ich ausführlich mit den Manuellen-Tests befasst, weshalb ich mich auf dieser Ebene jetzt fit fühle. Abschliessend kann gesagt werden, dass mir dieses Projekt einiges an neuem Wissen gebracht hat und dazu noch Spass gemacht hat.

Dario

Ich muss sagen, dass es eine sehr erfolgreiche und angenehme Erfahrung war. Einer der Schlüsselfaktoren für unseren Erfolg war die Art und Weise, wie wir als Team kommuniziert und zusammengearbeitet haben.

Von Anfang an stellten wir sicher, dass wir klare Kommunikationswege einrichten und uns regelmäßig absprechen, um alle auf dem Laufenden zu halten. Auf diese Weise konnten wir sicherstellen, dass alle auf derselben Seite standen und wir alle auf dieselben Ziele hinarbeiteten.

Was die eigentliche Projektarbeit anbelangt, so fand ich, dass die Aufteilung der Aufgaben in kleinere, leichter zu bewältigende Teile dazu beitrug, die Dinge zu organisieren und reibungslos voranzubringen. Wir haben auch darauf geachtet, die Aufgaben nach den Stärken und Fachkenntnissen der einzelnen Teammitglieder aufzuteilen, was dazu beigetragen hat, dass wir unsere Zeit und Ressourcen so effizient wie möglich genutzt haben.

Insgesamt bin ich der Meinung, dass die Kombination aus guter Kommunikation und effektivem Projektmanagement der Schlüssel zum Erfolg dieses Projekts war. Ich bin stolz auf das, was wir erreicht haben, und freue mich darauf, unsere nächste Herausforderung als Team in Angriff zu nehmen.

Jordan

Dieses Projekt war eine sehr gute Erfahrung für mich. Wir haben schon von Anfang an unser Projekt strukturiert durchführen können. Ich denke der Schlüssel dazu war, dass wir eine klare Aufteilung der Arbeiten hatten und stetig miteinander kommuniziert haben, um über die Arbeiten der Teammitglieder Bescheid zu wissen. Dadurch konnten wir uns gegenseitig unterstützen, wenn es nötig war. Wir sind während der gesamten Projektlaufzeit nie auf ein Problem gestossen, das wir nicht lösen konnten und haben deshalb effizient arbeiten können und konnten alles abschliessen, was wir uns vorgenommen haben.

Ich selbst konnte bei diesem Projekt viel lernen da ich vorher mit ASP .NET CORE noch nicht auseinandergesetzt habe und jetzt einen guten Einblick bekommen konnte. Des Weiteren habe ich bereits eine Weile nicht mehr Objektorientiert programmiert und konnte meine Kenntnisse in diesem Bereich auch wieder auffrischen.

Quellenverzeichnis

- Angular: <https://angular.io/>
- Canva: <https://www.canva.com/>
- Cypress: <https://www.cypress.io/>
- IPERKA: <https://www.bexio.com/de-CH/blog/view/iperka-methode>
- .Net Core (C#): <https://learn.microsoft.com/en-us/dotnet/core/introduction>
- Mongo DB: <https://www.mongodb.com/home>
- Sitemap: <https://de.wikipedia.org/wiki/Sitemap>
- Imgur: <https://imgur.com/>
- Ngx-countdown: <https://www.npmjs.com/package/ngx-countdown>

Glossar

API -> Eine API (Application Programming Interface) ist eine Schnittstelle, die ermöglicht, dass verschiedene Computerprogramme miteinander kommunizieren und Daten und Funktionen austauschen können.

Client -> Der Client ist ein Gerät, welches mit einem Netzwerk verbunden ist. Er sendet Anfragen an den Server und der Server liefert die gewünschten Daten oder Dienste zurück.

Fisher-Yates-Shuffle -> Das Fisher-Yates-Shuffle ist ein Algorithmus zum Zufälligen Durcheinanderbringen (Shufflen) von Daten in einem Array oder einer Liste.

Hash -> Ein Hash ist ein Wert, der aus einem Eingabewert, mithilfe einer Hash-Funktion erzeugt wird. Die Hash-Funktion nimmt den Eingabe-Wert als Input und gibt einen feste Länge besitzenden Hash-Wert als Output zurück. Dies erhöht die Sicherheit der Daten.

Rfc2898DeriveBytes-Objekt -> Rfc2898DeriveBytes ist eine Klasse in der .NET-Framework-Bibliothek, die verwendet wird, um sichere Schlüssel auf der Grundlage von Passwort-Phrasen zu erzeugen.

Salt -> Salt ist ein zufälliger Zeichenfolgen-Wert, der beim Erstellen von Hash-Werten verwendet wird. Es wird häufig verwendet, um die Sicherheit von Hash-Werten zu erhöhen, indem es verhindert, dass Angreifer Hash-Wert-Tabellen nutzen, um Passwörter zu entschlüsseln.

Sitemap -> Die Sitemap soll beim Realisieren einen guten Überblick über die verschiedenen Seiten und deren Relation verschaffen. Sie zeigt somit die logische Struktur der Webapplikation auf.

Abbildungsverzeichnis

Abbildung 1 - Zeitplan	7
Abbildung 2 - Frontend Ordner Struktur	26
Abbildung 3 - Shuffle Funktion	27
Abbildung 4 - Countdown HTML.....	28
Abbildung 5 - Countdown Config.....	28
Abbildung 6 - Ordnerstruktur Backend.....	28
Abbildung 7 - Delete User Controller Funktion.....	29
Abbildung 8 - Service für Nutzererstellung.....	29
Abbildung 9 - HashPassword Service Funktion.....	30
Abbildung 10 - Klassendiagramm	30
Abbildung 11 - Sequenz Diagramm.....	31
Abbildung 12 - Passing Unit Tests	36
Abbildung 13 - UI Passing Tests	37
Abbildung 14 - Erstellen von Kompetenzen.....	50
Abbildung 15 - Editieren von Kompetenzen	51
Abbildung 16 - Editieren einer bestehenden Subkompetenz.....	52
Abbildung 17 - Subkompetenzen auf den Status erledigt setzen.....	52
Abbildung 18 - Erstellen von zufälligen Gruppen	53