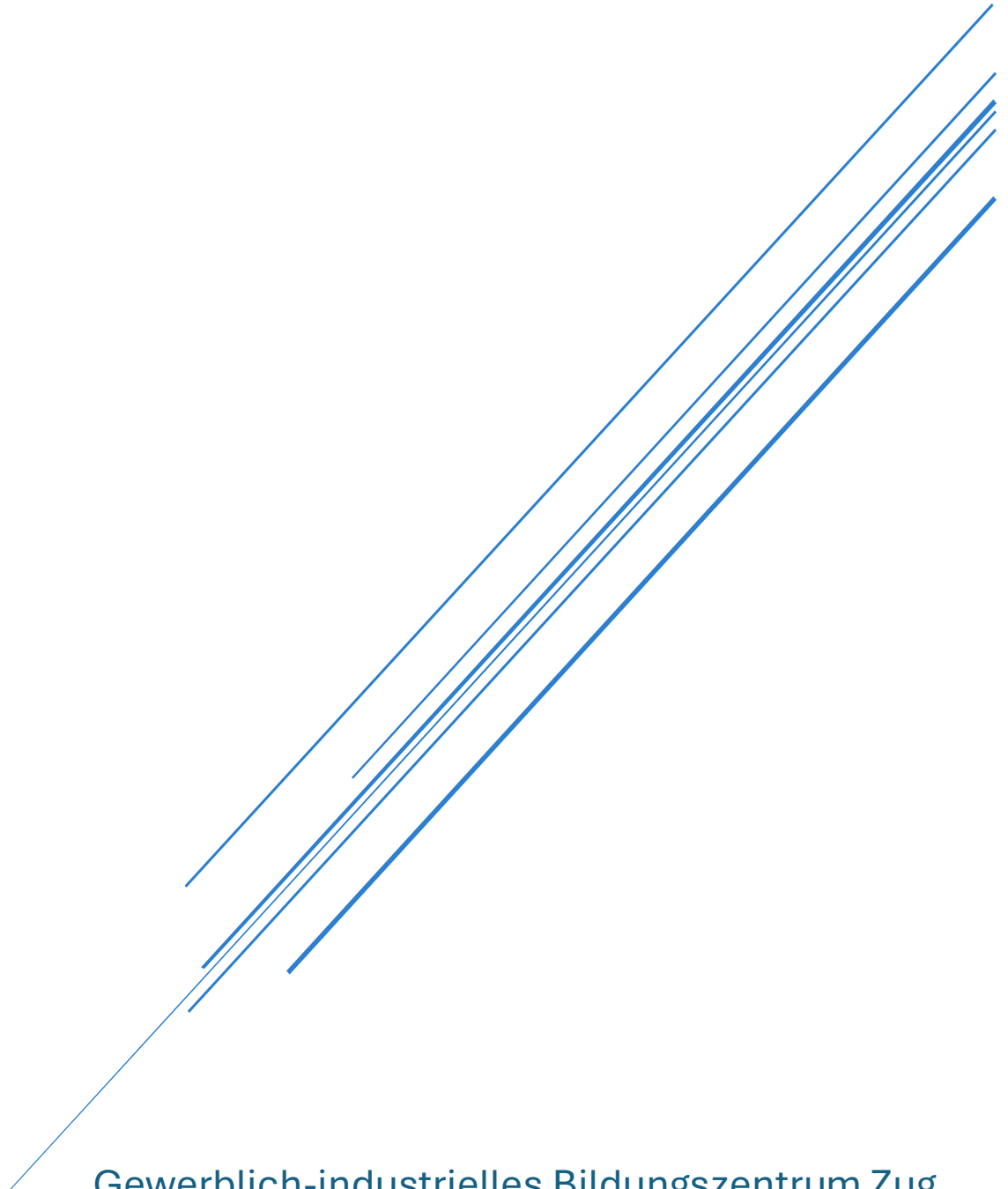


ZEUGNISNOTENANPASSUNG

M431 Projektdokumentation

Aufträge im eigenen Berufsumfeld selbstständig durchführen



Gewerblich-industrielles Bildungszentrum Zug
Modul 431

Inhaltsverzeichnis

1. Einleitung	2
1.1 Einleitung ins Projekt.....	2
1.2 Anleitung für Inbetriebnahme	2
2. Überblick	2
2.1 Beschreibung	2
2.2 Rollenverteilung.....	3
2.3 Rollenverteilung in unserem Team.....	3
2.4 Praktische Anwendbarkeit	3
3. Entscheidungen	4
3.1 Entscheidungen der Struktur	4
3.2 Entscheidungsprozess	4
3.3 Entscheidungen mit Prorektor (C. Lindauer)	5
4. Projektrisiken.....	6
4.1 Risikomatrix	6
5. Testing	6
5.1 Blackbox-Testing.....	6
6. Auswertung	6
6.1 Ziele	6
6.2 Stärken und Schwächen des Projekts	7
6.3 Fazit/Reflexion.....	8

1. Einleitung

1.1 Einleitung ins Projekt

In der Schule passiert es immer wieder, dass Zeugnisnoten erst spät oder nachträglich korrigiert werden müssen. Oft ist der Grund, dass eine Prüfung kurz vor Semesterende noch bewertet werden muss oder etwas vergessen ging. Das Problem dabei: Wenn das Zeugnis schon fast fertig ist, kann die Lehrperson die Note nicht einfach selber ändern. Dafür braucht es eine Anpassung and Zeugnisnoten, die über das Prorektorat geprüft und auch freigegeben werden muss. Momentan läuft das je nach Lehrperson unterschiedlich und es gibt keinen wirklich standardisierten Ablauf. Damit der Prorektor für die IT einen besseren Prozess für eben genau diese Notenabgabe hat, bekamen wir die Aufgabe, eine gutes Projekt dafür umzusetzen.

In diesem Projekt erstellen wir als Gruppe eine praktikable Lösung damit Anfragen für Notenänderungen einheitlich erfasst, klar dokumentiert und vom Prorektorat einfach geprüft und freigegeben werden können. Ziel ist, dass alle Beteiligten schneller arbeiten können und Änderungen einfacher laufen.

1.2 Anleitung für Inbetriebnahme

Um unser Projekt zu starten, muss unser GitHub-Repository lokal geklont werden. Als nächstes muss das .sln-file gestartet werden. Danach muss kontrolliert werden, ob “New Profile” als Startprojekt ausgewählt ist, damit die API und das Web gleichzeitig laufen können. Dann kann man das Projekt starten. Eigentlich sollte sich dann der Swagger und die Blazor-Webseite laufen. Danach kann man sich mit dem E-Mail und Passwort: **julia.abaecherli@gibz.ch** und **Init1234** Anmelden. Es wird zum Zeitpunkt der Abgabe aber nicht viel funktionieren und wir können nicht versprechen, dass die Webseite startet.

2. Überblick

2.1 Beschreibung

Bei diesem Projekt können Lehrer mit wenigen Klicks eine neue Note erfassen, ob es jetzt verspätet oder pünktlich ist. Im jetzigen Zustand unseres Projekts können wir nichts zu 100% versprechen das es laufen wird. Die App ist noch inmitten der Produktion und es folgen weitere Features. Nach dem Anmelden können neue Noten über den Tab “Neue Note hinzufügen” hinzugefügt werden. Wenn neue Noten hinzugefügt wurden, können diese bestätigt werden, wenn man sich als Prorektor angemeldet hat.

2.2 Rollenverteilung

- **Lehrpersonen**
 - Tragen Noten von Schüler:innen ein
 - Beim Eintragen von Verspäteten Noten wird eine Nachricht an den passenden Prorektor gesendet.
- **Prorektoren**
 - Der Prorektor des passenden Bereichs wird benachrichtigt, wenn eine Anfrage von einer verspäteten Note reinkommt.
 - Entscheidet und kontrolliert, ob die Daten korrekt sind, und nimmt die Noten an oder lehnt sie ab.

2.3 Rollenverteilung in unserem Team

- **Nelio**
 - Nelio war zuständig für die Datenbank und das Backend. Er arbeitete mit Cyril zusammen und sorgte dafür, dass die Verbindung zwischen diesen beiden lief.
- **Nevio**
 - Nevio war für das Mockup und schlussendlich auch für die Dokumentation zuständig. Er machte kleinere Aufgaben im Frontend wie z.B. die 2-Faktor-Authentifizierung.
- **Cyril**
 - Cyrils Arbeitsbereich war auch im Backend. Er sorgte hauptsächlich dafür, dass die meisten CRUD-Operationen funktionierten.
- **Emanuel**
 - Für das Frontend war Emanuel zuständig. Er schaute, dass die Controller auch vom Frontend verwendet werden und dass die GUI gut aussieht.

2.4 Praktische Anwendbarkeit

Die neue Funktion dieser Notenverwaltung soll das eintragen von Noten nicht nur einfacher machen, aber auch besser, um Stress bei der Zeugnisabgabe zu vermeiden. Das nervige Nachfragen beim Prorektor fällt weg, und der ganze Prozess wird vereinfacht. Das macht es effizienter und verhindert auch Fehler, die in Zeugnissen vorkommen können. Die Benutzeroberfläche soll einfach verwendbar und zugänglich sein. Mit einer 2-Faktor-Authentifizierung verhindert man, dass sich ein Schlingel mit nicht allzu guten Noten, in die App einloggen kann und sich nur Lehrer, die sich in der Datenbank befinden auf die App gelangen. Es profitieren nicht nur die Lehrer und Prorektoren, mit einem schnelleren, effizienteren und fehlerfreieren Prozess, sondern auch die Schüler, die den Schulabschluss ohne Fehler in ihrem Zeugnis, geniessen können.

3. Entscheidungen

3.1 Entscheidungen der Struktur

Wir entschieden uns, das ganze Projekt über GitHub zu hosten, da wir alle wussten, wie wir uns dort etwa orientieren konnten. Wir wussten aber auch, dass es dadurch viele nervige Konflikte (Mergeconflicts) geben wird. Das nahmen wir aber in Kauf und blieben bei GitHub.

Für die App selbst entschieden wir uns, eine Nutzwertanalyse zu machen, da jeder verschiedene Meinungen hatte und es eine gute Methode ist, die verschiedenen Optionen auszuwerten und sich für eine passende zu entscheiden.

Bei der Entscheidung der Datenbank sah es anders aus. Wir wollten es zuerst die Datenbank mit MySQL aufsetzen, wir hatten aber Diskussionen da nicht jeder MySQL so effizient fand. Da wir genau in dieser Zeit im ÜK Supabase bearbeiteten, es einfach zu benutzen ist und damit sie nicht jeder lokal installieren muss, entschieden wir uns für eine Supabase Datenbank mit PostgreSQL.

3.2 Entscheidungsprozess

Viele Entscheidungen waren logisch und es gab gar nichts zu diskutieren oder auszuwählen (DB, GitHub). Bei der Entscheidung, mit welcher Technologie wir das Hauptprojekt machen, hatten wir verschiedene Optionen zur Verfügung. Diese Entscheidungen trafen und hielten wir anhand einer Nutzwertanalyse fest. Die Kriterien für diese Nutzwertanalyse erarbeiteten wir mit einem kurzen Brainstorming am Start des Projekts.

Wir endeten mit acht passenden Kriterien. Darunter fallen Wissen und Arbeitsaufwand, was natürlich die wichtigsten Kriterien sind.

Nutzwertanalyse								
Projekt: M431 Notenverwaltung			Datum: 14.01.2026					
Nr	Kriterium	Faktor	HTML + CSS + JS		Blazor		MS Forms	
			Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte
1	Wissen	0.24137931	4.5	1.086207	5	1.206897	5.5	1.327586
2	Arbeitsaufwand	0.155172414	5	0.775862	4	0.62069	5.5	0.853448
3	Funktionalität	0.189655172	5	0.948276	5.5	1.043103	3.5	0.663793
4	Sendungsformat	0.068965517	4.5	0.310345	5	0.344828	5.5	0.37931
5	Gestaltung	0.103448276	4	0.413793	5	0.517241	4	0.413793
6	Erweiterbarkeit	0.068965517	4.5	0.310345	5.5	0.37931	4	0.275862
7	Sicherheit	0.137931034	5	0.689655	6	0.827586	4.5	0.62069
8	Integration zu Systemen	0.034482759	5.5	0.189655	5.5	0.189655	4	0.137931
Nutzwert			1	4.724138		5.12931		4.672414

Abbildung 1: Entscheidung für die Technologie anhand einer Nutzwertanalyse

Wie man im Bild erkennen kann, viel Blazor mit Abstand als beste Technologie auf. Das kann gut daran liegen da wir Blazor intensiv im ÜK angeschaut haben. Somit war es also fest. Die Technologie, die wir verwenden werden, ist Blazor.

3.3 Entscheidungen mit Prorektor (C. Lindauer)

- **Künstliche Intelligenz für die Hilfe beim Entscheiden von Zeugnisnoten**
 - Das widersprach unseren Vorstellungen. Wir hatten nicht vor und wussten auch nicht, wie wir das in unser Projekt einbauen sollten. Wir gingen aber auf die Idee ein. Wir schlugen vor, dass wir einen Zähler einbauen, der zählt wie viele Noten schon aufgerundet oder abgerundet wurden.
- **Sichere Authentifizierung der Lehrpersonen und Prorektoren**
 - Dies war eine Änderung, die gut möglich für uns war. Wir wägen es aus, ob wir überhaupt gut genug sind und ob wir es noch einbauen können. Wir entschieden uns, 2-Faktor-Authentifizierung zu verwenden mit einem Code, der auf das E-Mail vom angemeldeten Lehrer geschickt wird.
- **Turbo Blockchain Modus, kryptografische Verschlüsselung**
 - Bei dieser Anforderung war die Entscheidung schnell getroffen. Nicht nur, würde diese Umwandlung eine Menge Geld kosten, aber es entspricht auch nicht unseren Kompetenzen. Zudem ist **nachträgliches** anpassen gar nicht möglich, da Blockchain das nicht zulässt.

4. Projektrisiken

4.1 Risikomatrix

Die Risikoanalyse sollte man grundsätzlich für jedes Projekt machen. Dort legt man vor und während des Projekts Risiken fest, die wirkliche Folgen haben könnte auf das Projekt. Man bewertet die Risiken anhand Auswirkung, also wie sehr schadet es dem Projekt und wie hoch ist die Wahrscheinlichkeit, dass dieses Risiko eintreten wird.

Risiko						Analyse		
Nr.	Thema / Kategorie	Kurzbezeichnung	Risikobeschreibung	Erfasst am	Erfasser/-in	Risikostatus	Eintrittswahrscheinlichkeit	Auswirkung
1	Technik	Server Ausfall	Die Microsoft Server fallen aus	06.11.2025	Emanuel Wildi	aktiv	30%	4
2	Technik	Wasserschaden	Laptop wird mit Wasser überschüttet	06.11.2025	Emanuel Wildi	aktiv	10%	3
3	Organisation	Feuer	Die Schule fängt an zu brennen	06.11.2025	Emanuel Wildi	aktiv	10%	5
4	Technik	Kein Backup	Wenn Server/Computer abstürzt ohne Backup	06.11.2025	Emanuel Wildi	aktiv	50%	2
5	Technik	WLAN	Unbefugter Zugriff aufs WLAN	06.11.2025	Lindauer	aktiv	50%	4
6	Technik	Datenverlust	Dateien Lokal speichern statt auf dem Server	06.11.2025	Lindauer	aktiv	30%	3
7	Organisation	Phishing Email	Eine Phishing Email öffnen	06.11.2025	Lindauer	aktiv	30%	3
8	Organisation	Zeit	Nicht genug Zeit, die Lösung zu beenden...	14.01.2026	Nevio Gallo	aktiv	70%	4
9	Organisation	Streit	Das Team möchte nicht mehr zusammen arbeiten	14.01.2026	Nevio Gallo	aktiv	10%	5
10	Fachlichkeit	Erfahrung	Unsere Kompetenzen fürs Arbeiten sind nicht gut genug.	14.01.2026	Nevio Gallo	aktiv	30%	3
11	Organisation	Abgabetermin	Wir verpassen den Abgabetermin	14.01.2026	Nevio Gallo	aktiv	10%	2
12	Organisation	Fehlende Kommunikation	Die Produktion dauert länger und kann Streit auslösen.	15.01.2026	Nello Schnieper	aktiv	90%	2
13	Technik	Cloud Datenverlust	Daten in der Cloud/Git werden beschädigt oder gelöscht.	15.01.2026	Nello Schnieper	aktiv	10%	4
13	Vertragsmanagement	Anforderungsänderungen	Der Auftraggeber hat neue Anforderungen.	15.01.2026	Nello Schnieper	aktiv	50%	3

Wie verhindern wir das Eintreten der von uns verhinderbaren Risiken? Wir sorgen dafür, dass jeder klare Aufgaben hat und ein klarer Plan (Codecks) verfügbar ist. Zudem war uns wichtig, dass jeder auch weiss, was seine Aufgaben sind und dass diese Person diese Einhält. Wir erinnerten uns gegenseitig, wenn jemand etwas bei seiner Aufgabe vergessen hat oder noch machen soll. Wir schauten, dass jedes Teammitglied weiss, wie man richtig Committed und pusht auf GitHub, um Datenverlust zu vermeiden.

5. Testing

5.1 Blackbox-Testing

(Noch nicht durchgeführt)

Wir planen, Blackbox-Tests durchzuführen, wenn unsere App vollständig läuft. So können wir versteckte Fehler oder Schwachstellen finden und diese beheben. So fördern wir Benutzerfreundlichkeit und testen somit auch, wie einfach es ist, sich in unserer App zu orientieren.

6. Auswertung

6.1 Ziele

Mit dem jetzigen stand, finden wir, dass wir die Ziele erreicht haben. Wir arbeiten gut als Team zusammen und es fühlt sich wie ein richtiges Projekt an welchen Spass macht im Team zu bearbeiten. Bis jetzt sind wir sehr zufrieden, was wir erreicht haben. Wir haben

es geschafft, ohne viel Stress ein stabiles Projekt, bei dem wir einen Plan befolgen können (Codecks), ein Frontend, ein Backend, eine Datenbank, mehrere Analysen und eine umfangreiche Dokumentation. Wir sind froh, dass jeder seinen Bereich ausarbeiten kann und die Teamarbeit so gut funktioniert. Wir hatten interessante Diskussionen aber keinen einzigen Streit. Das Projekt geht gut voran und jeder zeigt Interesse und Spass, was das Wichtigste ist.

6.2 Stärken und Schwächen des Projekts

Stärken:

Was sehr gut lief, ist die Rollenverteilung. Jeder bekam seine Aufgaben und arbeitete auch selbständig daran. Was sehr gut lief ist der Start. Das Mockup war schnell, umfangreich und verständlich. Das UI dazu war schnell auf die Beine gestellt. Die Atmosphäre war sehr angenehm, wenn wir z.B. allein in einem Raum daran arbeiteten. Wir hatten sehr interessante Diskussionen und dann auch Entscheidungen, die getroffen werden mussten, welche das Projekt nur noch besser machten.

Schwächen oder Herausforderungen:

Wo wir am meisten Probleme hatten, war die Zeit. Es ist nicht so, dass wir Stress hatten, sondern es war einfach eine Herausforderung Zeit dafür zu finden. Unter all den anderen Tests in der Schule und den ÜKs war es schwierig, Zeit in seine Aufgaben investieren zu können. Auch wenn wir die Aufgaben gut erledigten, mussten wir uns viel Zeit am Abend oder an den Wochenenden Zeitnehmen, um diese zu bearbeiten.

6.3 Fazit/Reflexion

Die frühe Rollenverteilung war für uns sehr hilfreich, weil dadurch von Anfang an klar war, wer was macht und man nicht ständig doppelt arbeitet. Cyril achtete darauf, dass die Aufgaben im Code vorwärtsgehen und fertig werden. Nevio übernahm eher das Organisatorische und schaute, dass Unterlagen und Abgaben rechtzeitig bereit sind. So konnten wir generell konzentrierter arbeiten.

Bei der Aufgabenbewältigung half uns vor allem, dass wir einen Plan hatten (Codecks) und die Aufgaben sauber aufgeteilt waren. Entscheidungen im Team konnten wir meistens schnell treffen, weil wir vieles logisch klären konnten. Bei grösseren Fragen, wie der Technologie, war es hilfreich, dass wir die Nutzwertanalyse gemacht haben, damit nicht einfach die lauteste Meinung gewinnt.

Was die Arbeit manchmal gestört hat, war vor allem das Thema Zeit und Überschneidungen mit Schule und ÜK. Dadurch war es schwieriger, im gleichen Tempo zu bleiben, und manchmal musste man abends oder am Wochenende noch etwas erledigen. Eine weitere Herausforderung war, dass Frontend, Backend und Datenbank voneinander abhängig sind. Wenn irgendwo etwas fehlt oder sich ändert, steht ein anderer schnell mal an.

Positiv war für uns, dass die Zusammenarbeit angenehm war und wir trotz Diskussionen keinen Streit hatten. Man hatte das Gefühl, dass alle wirklich etwas beitragen. Negativ war, dass man bei einem Projekt schnell merkt, dass "noch schnell etwas einbauen" meistens doch länger dauert als gedacht.

Wenn ich das nächste Mal etwas anders machen würde, dann würden wir früher festlegen, welche Funktionen am Schluss wirklich zwingend fertig sein müssen und welche nur "nice to have" sind.

Insgesamt haben wir aus dem Projekt mitgenommen, wie wichtig klare Zuständigkeiten, ein realistischer Plan und saubere Absprachen sind. Das Projekt ist zwar noch nicht komplett fertig, aber wir wissen jetzt besser, wie man ein Projekt strukturiert startet, wie man Entscheidungen im Team sinnvoll trifft und was als nächstes nötig ist, damit es am Ende auch wirklich einsatzbereit wird.

-Eine detailliertere Ansicht von Arbeitsschritten von jedem Teammitglied befindet sich im Readme auf unserem GitHub, dieses wird in Zukunft noch erweitert.