

Timely



Entwicklung eines Stundenplantools



Maturitätsarbeit Kantonsschule Büelrain Winterthur

Verfasser: Andrin Blass, 4eW
Betreuung: Sebastian Schiendorfer
Abgabedatum: 02.12.2024

Abstract

Automatische Information über Stundenplanänderungen der nächsten sieben Tage, Vorschläge zur Stundenverschiebung und eine einfache Suche nach freien Zimmern – all das bietet Timely. Das von mir entwickelte Tool tauscht automatisiert Daten zwischen dem Intranet, einer selber programmierten Webapp sowie Teams aus, verarbeitet die Daten und hält so die Schülerinnen und Schüler über verschobene oder ausgefallene Stunden auf dem Laufenden, liefert Vorschläge für Verschiebungsmöglichkeiten samt vorgefertigtem Nachrichtentext an die Lehrperson und zeigt die Zimmerbelegung an. Die Verschiebungsvorschläge werden dabei mit Künstlicher Intelligenz generiert. Link zur Webapp: timely.school

Inhaltsverzeichnis

1. Ausgangslage	1
2. Aufgabe und Ziele	1
3. Umfrage und Ergebnisse	3
4. Das Tool Timely und seine Funktionalitäten	9
4.1. Timely Design-Guidelines	
5. Funktionsweise des Tools	11
5.1. Stundenplan-Benachrichtigungen und Verschiebungsvorschläge	
5.2. Webapplikation	
6. Entwicklungsprozess und Programmierung	17
6.1. Automatische Verschiebungsvorschläge und Teams-Benachrichtigung	
6.2. Timely API Server	
6.3. Webapplikation	
7. Verwendung von KI	37
8. Schlusswort und Ausblick	39
9. Abbildungsverzeichnis	III
10. Quellenverzeichnis	IV

1. Ausgangslage

Fast jede Schülerin und jeder Schüler kennt die Situation: eine Stunde wird im Stundenplan gestrichen, aber man merkt es erst am Morgen des betreffenden Schultages. Meist ist es dann zu spät, um eine Randlektion an die Stelle der gestrichenen Lektion zu verschieben. Und selbst wenn man es frühzeitig merkt, ist es oft aufwendig, die Stundenpläne der in Frage kommenden Lehrerinnen und Lehrer zu vergleichen, um jemanden zu finden, der Zeit hätte, seine Stunde zu verschieben. Aber auch dann muss man, nachdem man sich in der Klasse auf eine Lösung geeinigt hat, die Lehrerin oder den Lehrer anfragen, ob die Verschiebung auch wirklich möglich ist.

Ziel meiner Maturitätsarbeit war es, den Prozess des Stundenverschiebens und die Funktionen des Stundenplans generell mit Hilfe selbstentwickelter digitaler Tools für alle zu erleichtern und zu verbessern. Besonders wichtig war mir dabei, die Lösung so benutzerfreundlich und benutzerorientiert wie möglich zu gestalten.

2. Aufgabe und Ziele

Aufgrund dieser Ausgangslage habe ich zuerst eine Umfrage durchgeführt, um meine Erfahrungen aus dem Stundenverschiebungsprozess bei der Zielgruppe zu überprüfen sowie die potenziellen künftigen Nutzerinnen und Nutzer zu fragen, was ihnen bei einem solchen Tool wichtig wäre.

Die grundsätzlichen Ziele meines Tools sind folgende:

- Die automatisierte Überprüfung des Stundenplans bezüglich Änderungen wie Verschiebungen, Raumänderungen und Ausfällen.
- Basierend auf den vom Tool gefundenen Änderungen sollen möglichst effiziente Lösungen vorgeschlagen werden, um Stunden zu verschieben und unnötige Freilektionen zu vermeiden.
- Die Verschiebemöglichkeiten sollen den Schülerinnen und Schülern vorgeschlagen werden, damit diese in der Klasse besprochen und die betroffenen Lehrerinnen oder Lehrer angefragt werden können.
- Allfällige andere Features, die dem Stundenplan fehlen, sollen nach Möglichkeit ebenfalls ergänzt werden.

3. Umfrage und Ergebnisse

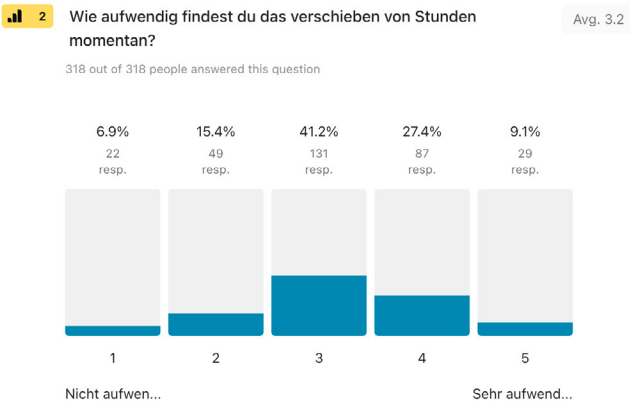
Als ersten Schritt habe ich eine Umfrage durchgeführt. Ziel war es, die Zufriedenheit mit der aktuellen Situation bezüglich Stundenplanänderungen sowie die Bedürfnisse meiner potenziellen Nutzerinnen und Nutzer zu evaluieren. Die online durchgeführte Umfrage wurde von 318 Schülerinnen und Schülern der gesamten KBW ausgefüllt – das entspricht mehr als einem Drittel der Schülerschaft und damit der potenziellen Nutzerinnen und Nutzern. Dadurch erhielt ich wertvolle Informationen für die Entwicklung meines Tools. Die Auswertung erfolgte je nach Frage nach Punkten (1 bis 5) oder in Prozenten.

Im ersten Teil der Umfrage ging es darum, die aktuelle Situation zu evaluieren und das grundsätzliche Bedürfnis für eine Verbesserung zu ermitteln. Dazu habe ich folgende beiden Fragen genutzt, die auf einer Skala von 1 bis 5 Punkten beantwortet werden mussten.

Abb. 1: Umfrageergebnis Frage 1



Abb. 2: Umfrageergebnis Frage 2



1. Wie wichtig ist es dir bei ausgefallenen Stunden, diese zu verschieben um unnötige Freilektionen zu vermeiden? (vgl. Abb. 1)

Für mehr als 68% der Befragten ist es wichtig oder sehr wichtig Lektionen auf ausgefallene Stunden zu verschieben um Freilektionen zu vermeiden.

Dieses Ergebnis deckt sich mit meinen Erwartungen und bestätigt meine anfängliche Vermutung, dass es wichtig ist Vorschläge zur Vermeidung unnötiger Freilektionen in einer Änderungsbenachrichtigung zu inkludieren.

2. Wie aufwendig findest du das Verschieben von Stunden momentan? (vgl. Abb. 2)

Mehr als 36% der Schüler und Schülerinnen empfanden zum Zeitpunkt der Umfrage das Verschieben von Lektionen aufwendig bis sehr aufwendig. Die Mehrheit (41%) bewertete das Verschieben neutral.

Dieses Resultat zeigt, dass es definitiv noch Verbesserungspotenzial gibt, die momentane Situation jedoch etwas weniger schlecht beurteilt wird, als ich es erwartet hatte. Dass eine Mehrheit diese Frage mit «neutral» beantwortet hat, könnte daran liegen, dass in vielen Klassen die Klassenchefs und einige wenige andere Schülerinnen und Schüler diese Aufgabe übernehmen und ein Grossteil der Klasse nur über die möglichen Lösungen abstimmen muss. Somit sind auch für eine Mehrheit der Klasse Verschiebungen nicht besonders aufwendig.

Im zweiten Teil meiner Umfrage ging es darum, Daten zu sammeln um das Tool möglichst benutzerfreundlich und benutzerorientiert zu gestalten. Zusätzlich zur Frage für welche Änderungen im Stundenplan Push-Benachrichtigungen gewünscht würden, habe ich sechs Fragen zur Bewertung von verschiedenen möglichen Features gestellt. Dazu kamen noch eine offene Frage zu weiteren gewünschten Funktionen und eine Frage dazu, wie weit im Voraus die Änderungen kommuniziert werden sollten. Die Bewertung basierte dabei auf der Skala: «Nicht nötig» (1), «Nice to have» (3), «Zwingend nötig» (5). Auf der folgenden Seite sind diese Fragen aufgelistet.

Abb. 3: Umfrageergebnis Frage 3

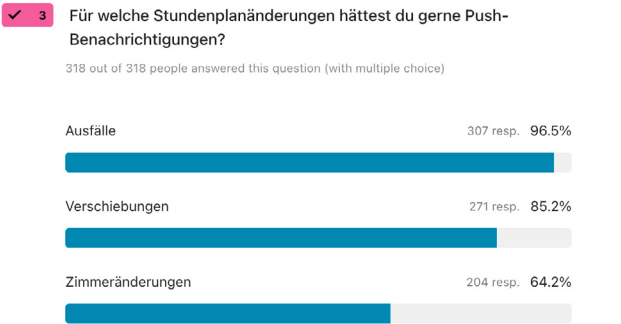


Abb. 4: Umfrageergebnis Frage 4



3. Für welche Stundenplanänderungen hättest du gerne Push-Benachrichtigungen? (vgl. Abb. 3)

Wie in der Abbildung 3 ersichtlich, waren alle Änderungstypen ziemlich beliebt. Bei einer genaueren Auswertung der Umfragedaten mit Excel wurde ersichtlich, dass mehr als 55% alle Änderungstypen ausgewählt haben. Sehr klar war das Resultat bezüglich Infos zu Stundenausfällen: Hier wünschen sich 96% der Befragten eine Benachrichtigung auf ihrem Handy. Dieses Resultat entsprach auch meinen Erwartungen.

Bitte bewerte, wie wichtig dir die folgenden Features wären:

4. Info, wann die Stundenplanänderung ist (vgl. Abb. 4)

Erwartungsgemäss findet die Mehrheit der Befragten dieses Feature nötig bis zwingend nötig. Erstaunlich war jedoch, dass von rund 40% der Befragten dieses Feature «nur» als «Nice to have» markiert wurde. Im Nachhinein hätte ich diese Frage besser formulieren müssen, um klarzumachen, dass es dabei um eine Push-Benachrichtigung geht. Denn bereits jetzt werden Änderungen im Stundenplan im Intranet klar ersichtlich dargestellt, jedoch aber ohne Verschieben einer Benachrichtigung.

Abb. 5: Umfrageergebnis Frage 5

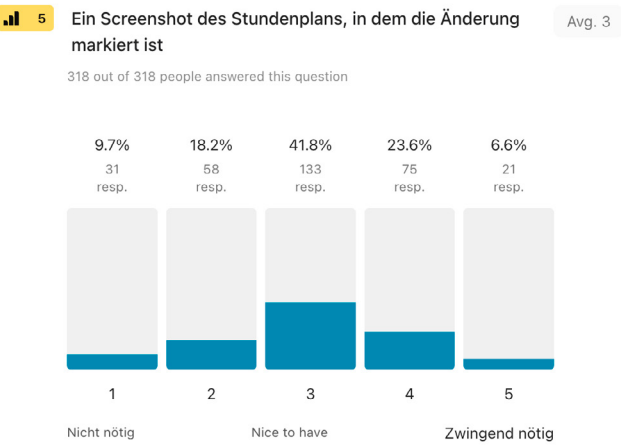
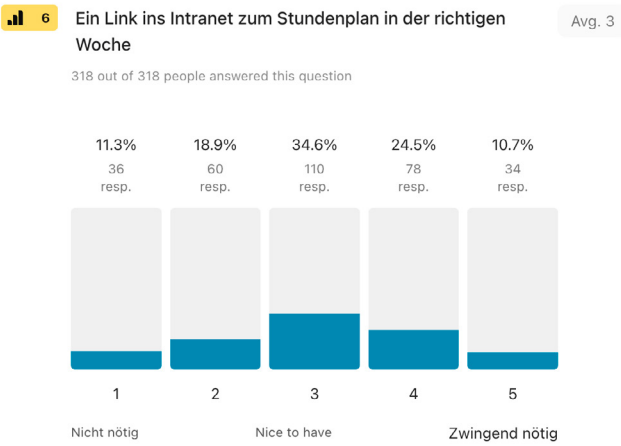


Abb. 6: Umfrageergebnis Frage 6



5. Ein Screenshot des Stundenplans, in dem die Änderung markiert ist (vgl. Abb. 5)

Dieses Feature wurde im Durchschnitt 0.6 Punkte schlechter als jenes von Frage vier und von mehr als 41% der Befragten als «Nice to have» bewertet. Dies entspricht meinen Erwartungen. Dieses Feature habe ich allerdings schlussendlich nicht umsetzen können.

6. Einen Link ins Intranet zum Stundenplan der richtigen Woche (vgl. Abb. 6)

Im Durchschnitt wurde dieses Feature mit 3 von 5 Punkten bewertet und deshalb ebenfalls als «Nice to have» eingestuft. Allerdings waren die Meinungen breiter verteilt und dadurch stuften es auch von 35% der potenziellen Nutzerinnen und Nutzer als wichtig oder sehr wichtig ein. Diese Resultate entsprachen ebenfalls meinen Erwartungen.

Abb. 7: Umfrageergebnis Frage 7

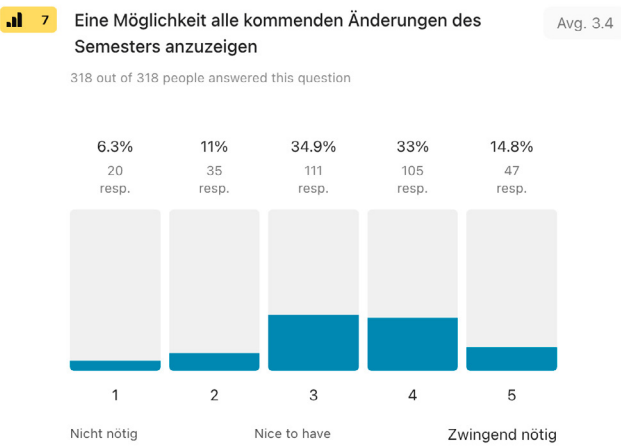
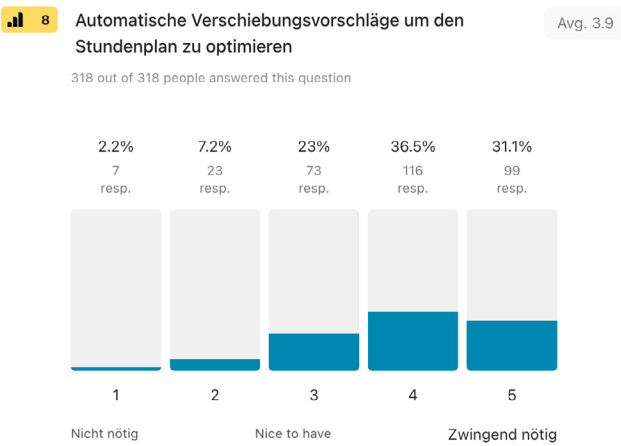


Abb. 8: Umfrageergebnis Frage 8



7. Eine Möglichkeit alle kommenden Änderungen des Semesters anzuzeigen (vgl. Abb. 7)

Dieses Feature wurde im Durchschnitt mit 3.4 Punkten bewertet und über 47.8% der Befragten finden es nötig bis zwingend nötig. Dieses Resultat hat mich am meisten erstaunt, da bei Frage zehn die Mehrheit nur Benachrichtigungen für die nächsten sieben Tage wollte.

8. Automatische Verschiebungsvorschläge, um den Stundenplan zu optimieren (vgl. Abb. 8)

Erfreulicherweise wurde dieses Feature im Durchschnitt mit 3.9 Punkten bewertet und über 67% finden es wichtig bis sehr wichtig. So ein klares Ergebnis hätte ich nicht erwartet. Eine Mehrheit der Befragten fand das Verschieben von Stunden weder als speziell aufwändig noch als nicht aufwändig. Die Gründe dafür sind vermutlich – wie weiter oben schon dargelegt, dass sich meist die Klassenchefin oder der Klassenchef darum kümmern. Trotzdem ist davon auszugehen, dass viele potenzielle Nutzer – allen voran die Klassenchefinnen und -chefs – ein solches Feature be-

Abb. 9: Umfrageergebnis Frage 9

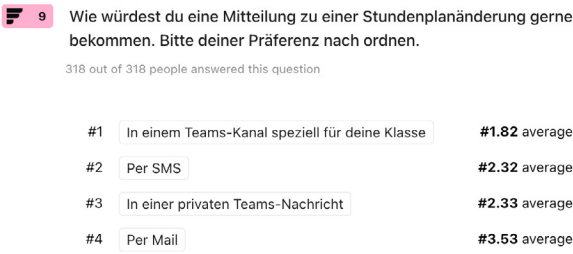
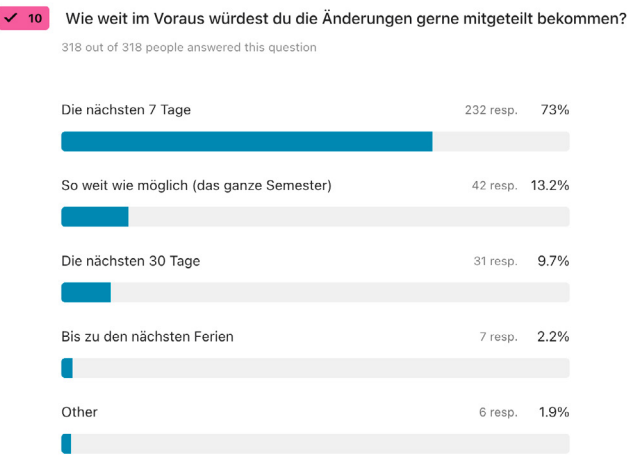


Abb. 10: Umfrageergebnis Frage 10



grüssen würden, um den Prozess des Verschiebens erleichtern zu können.

9. Wie würdest du eine Mitteilung zu einer Stundenplanänderung gerne bekommen? (vgl. Abb. 9)

Sehr klar wurde bei dieser Frage Teams als bester Informationskanal ausgewählt, um Benachrichtigungen zum Stundenplan zu erhalten. Deshalb habe ich mich dazu entschieden ebenfalls auf Teams zu setzen.

10. Wie weit im Voraus würdest du die Änderungen gerne mitgeteilt bekommen? (vgl. Abb. 10)

Über 73% der Schülerinnen und Schüler wollen die Änderungen nicht mehr als sieben Tage im Voraus mitgeteilt bekommen. Für weiter in der Zukunft liegende Änderungen bevorzugen viele der Befragten (siehe Frage sieben) lieber eine Möglichkeit selbst nachschauen, was für Änderungen es in Zukunft gibt.

11. Hast du noch weitere Features, die dir wichtig wären?

Total wurden 21 zusätzliche Features vorgeschlagen. Die wichtigsten sind Folgende:

- **Freie Zimmer anzeigen**
Fünf Personen haben dies vorgeschlagen. Deshalb habe ich dieses Feature aufgenommen und eine Lösung dafür entwickelt.
- **KBW Stundenplan in App**
Drei Personen wünschten sich eine App um den Stundenplan aus dem Intranet anzuzeigen. Allerdings gibt es bereits mehrere Apps, die das können wie z.B. die offizielle Intranet App¹. Deshalb habe ich dies auch nicht weiterverfolgt in meiner Arbeit.
- **Notenbüchlein Benachrichtigung**
Jemand wünschte sich ein Feature für eine Push-Benachrichtigung, sobald eine Note ins Notenbüchlein eingetragen wird. Persönlich hätte ich auch Interesse an einem solchen Feature, jedoch ist die Umsetzung technisch relativ schwierig. Kommt dazu: Nur wenige Lehrpersonen benutzen das Online-Notenbüchlein regelmässig, beziehungsweise informieren oftmals so wieso nach dem Hochladen der Noten die Klasse über Teams. Aus diesen Gründen habe ich mich dazu entschieden dieses Feature nicht weiter zu verfolgen.
- **Integration mit IMS Stundenplan BBW**
Eine Schülerin oder ein Schüler aus der IMS wünschte sich eine Integration des Stundenplans der Berufsfachschule. Da dieses Feature nur wenige Klassen betreffen würde, habe ich mich ebenfalls entschieden es nicht in die Programmierung aufzunehmen.

¹ Vgl. Apple Inc., IN2 Mobile, Internet.

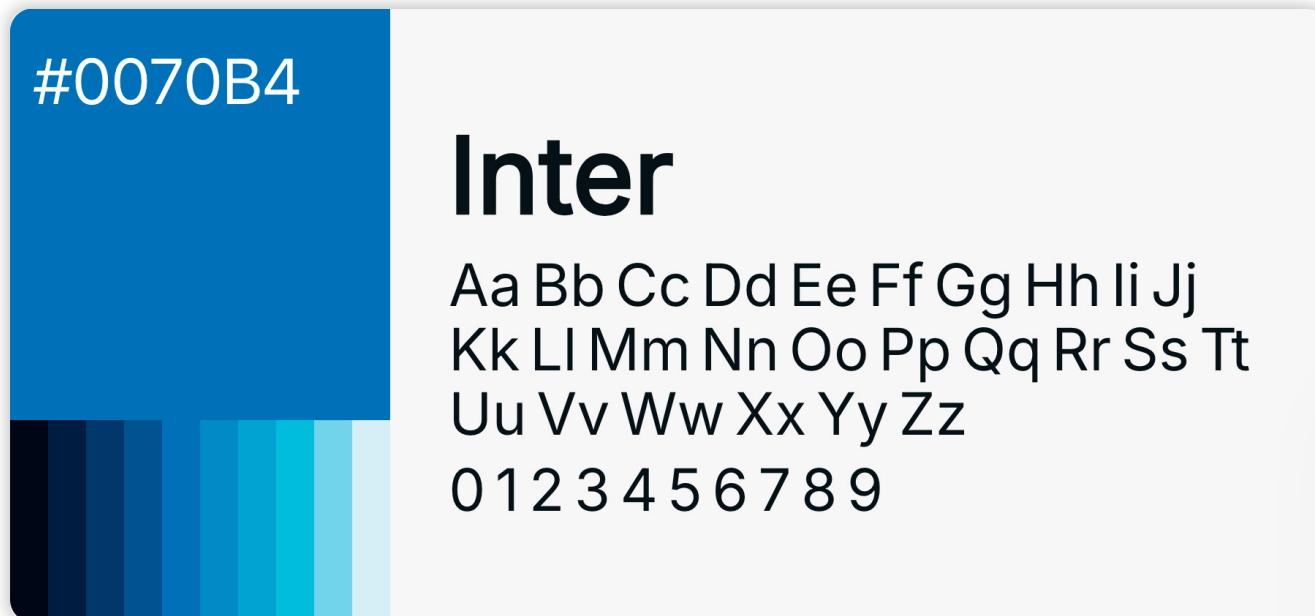
Abb. 11: Timely Logo



Abb. 12: Timely App-Icon



Abb. 13: Timely Farbpalette und Schriftart



4. Das Tool Timely und seine Funktionalitäten

4.1. Name und Logo

In der Umfrage hatte ich auch um Namensvorschläge für das Tool gebeten. Die Ideen waren als Inspiration sehr hilfreich. Schlussendlich habe ich mich aber, auch mit Blick auf die Zielsetzungen, für den von mir selbst entwickelten Namen «Timely» entschieden. Kriterien für die Suche nach dem Namen waren unter anderem: Kurzes Wort, damit der Domainname der Webapplikation einfach zu merken ist; einprägsam, modern und einfach zu merken; intuitiver Hinweis auf die Funktion des Tools sowie die Verfügbarkeit der Domain timely.school.

Der Name Timely erfüllt all diese Kriterien, spielt auf Timetable (Stundenplan) an und hat zugleich die Bedeutung «pünktlich».

Timely umfasst kurz zusammengefasst folgende Elemente und Funktionalitäten:

- Einen Benachrichtigungsdienst, der laufend Stundenplanänderungen erfasst und die Nutzerinnen sowie Nutzer darüber informiert. Der Dienst schlägt zudem auch vor, wie durch Verschiebungen entstandene Lücken im Stundenplan geschlossen werden könnten.
- Eine Webapplikation, auf der alle kommenden Stundenplanänderungen ersichtlich sind und mit der nach freien Zimmern gesucht werden kann.

Das Logo (vgl. Abb. 11) umfasst den Schriftzug und ein abstrahiertes Kalenderblatt. Das App-Icon (vgl. Abb. 12) besteht nur aus dem Kalender-Icon kombiniert mit einem feinen, dunklen Farbverlauf als Hintergrund. Das Häkchen unterstreicht die Einfachheit, Anwenderfreundlichkeit sowie Vertrauenswürdigkeit der App und vermittelt eine positive Ausstrahlung.

4.2. Timely Design-Guidelines

Damit Timely einheitlich daherkommt, entschied ich mich für das Festlegen einiger simpler Design-Guidelines. Als Schriftart wählte ich «Inter», da sie der Schriftart von Apple sehr ähnlich sieht, aber dennoch ohne Restriktionen verwendet werden darf. Als Primär- und Akzentfarbe habe ich mich für ein ruhiges Blau entschieden, bei der Hintergrundfarbe für ein dunkles Anthrazit um einen professionellen Eindruck zu erreichen. Diese Design-Guidelines (vgl. Abb. 13) kommen insbesondere im visuellen Design der Webapplikation und für die Gestaltung des Werkberichts zur Anwendung.

Abb. 14: Beispiel einer Raumänderungsnachricht

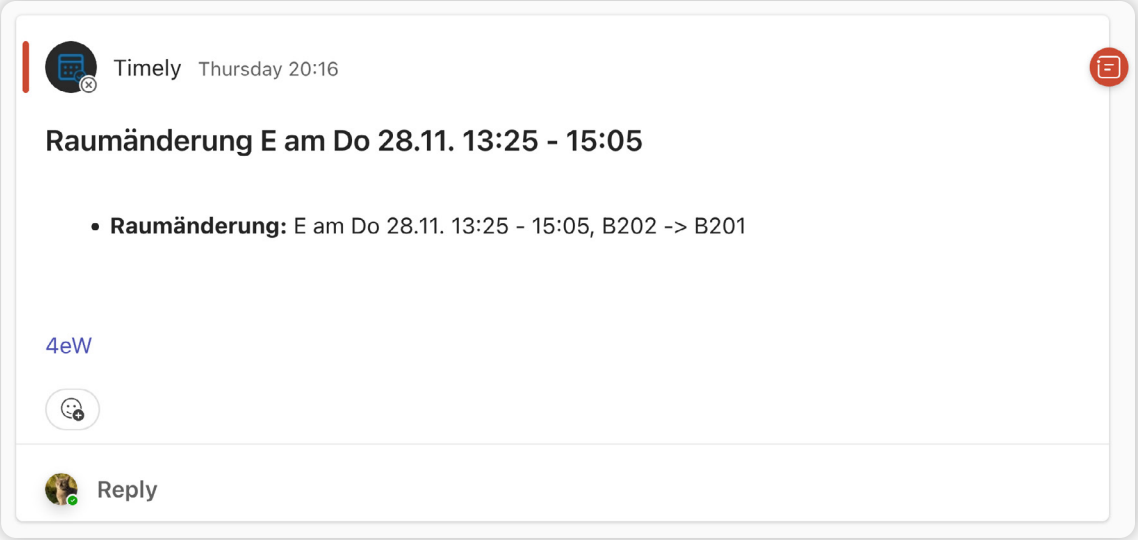


Abb. 15: Beispiel eines automatischen Verschiebungsvorschlages

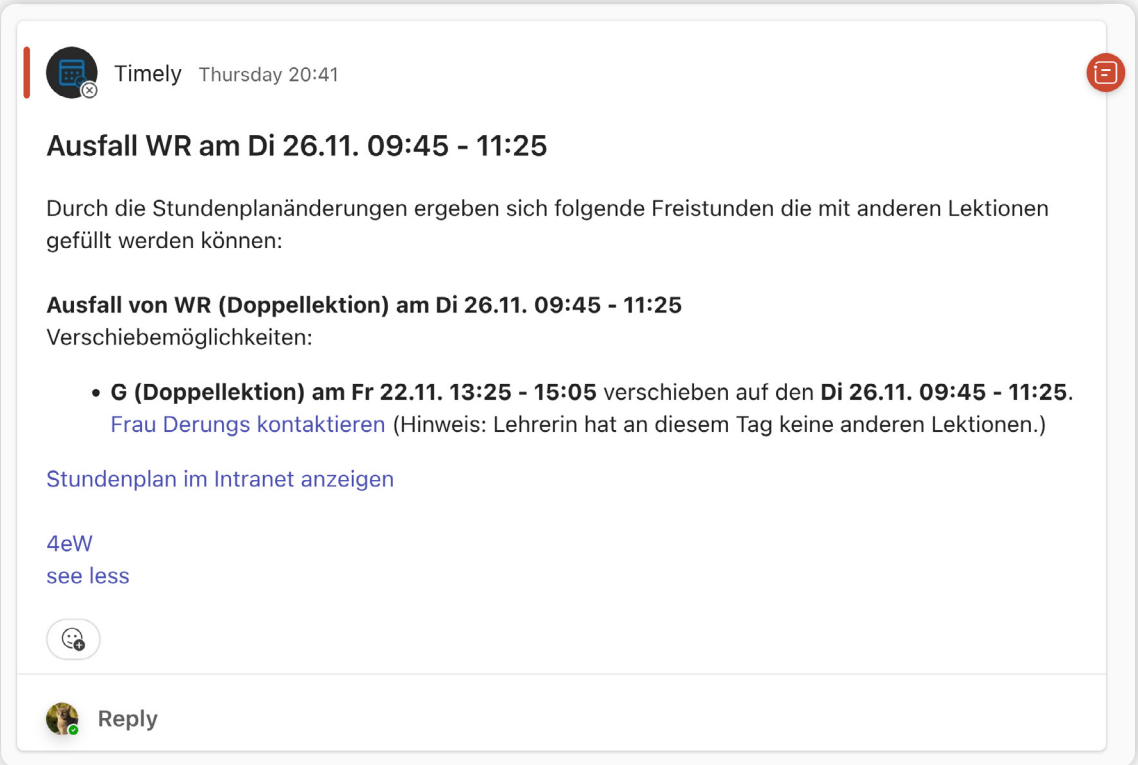
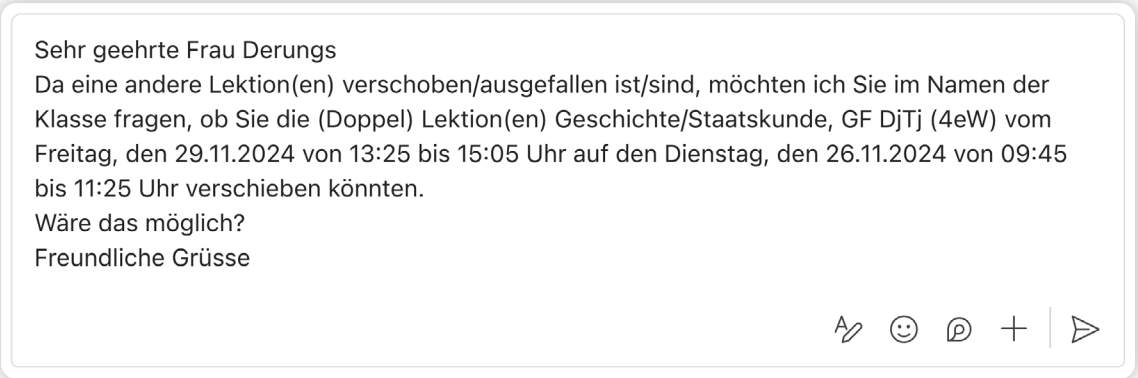


Abb. 16: Beispiel einer automatisch generierten Verschiebungsanfrage



5. Funktionsweise des Tools

In diesem Kapitel erkläre ich die Funktionsweise aus Sicht der Nutzenden - in der Fachsprache auch «Front-End» genannt. Dabei zeige ich sowohl, wie das Tool generell funktioniert, als auch wie es bedient wird. Die Abläufe im Hintergrund, in der Fachsprache genannt «Back-End», werden im Kapitel 6 beschrieben. Aktuell ist Timely als Probetrieb für drei Klassen der KBW implementiert (4eW, 4bW, 3bW), eine Erweiterung auf die ganze Schule ist aber problemlos möglich und als nächster Schritt auch vorgesehen.

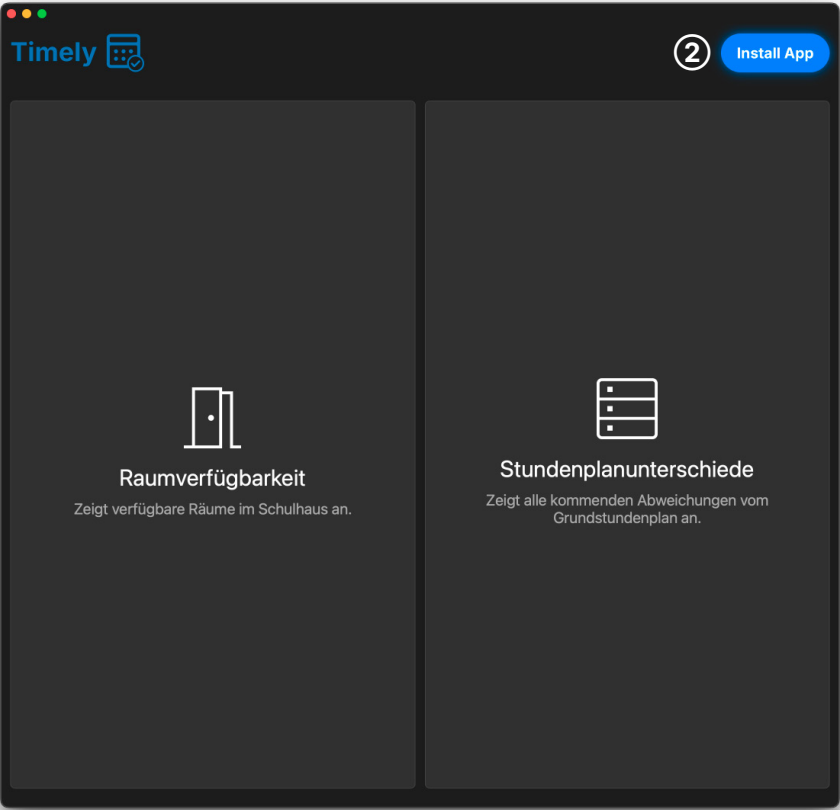
5.1. Stundenplan-Benachrichtigungen und Verschiebungsvorschläge

Timely kommuniziert jede Stundenplanänderung, sofern diese die nächsten sieben Tage betrifft. Für Änderungen, die ausserhalb des Betrachtungszeitraums von Timely von sieben Tagen erstellt wurden, verschickt das Tool keine Mitteilungen. Solche längerfristigen Stundenplanänderung können die Schülerinnen und Schülern gut auch auf der Stundenplan-Webseite nachschauen und haben genügend Zeit sich um Verschiebungen von Stunden zu kümmern.

Der Stundenplan wird stündlich zwischen 06:00 und 22:00 Uhr vom Tool analysiert. Entdeckt es eine Änderung, wird die betroffene Klasse über den dafür vorgesehenen Teams-Kanal im Timely-Teams benachrichtigt. Diese Nachrichten sind so konfiguriert, dass jede Schülerin und jeder Schüler der Klasse eine Push-Benachrichtigung erhält. Diese Benachrichtigungen beinhalten folgende Punkte:

- **Welche Lektion(en) geändert wurde(n):** Das Fach und der Zeitraum der geänderten Lektion(en).
 - **Was geändert wurde:** Es gibt verschiedene Änderungstypen:
 - **Raumänderung:** Es wird gezeigt, welche Lektion(en) von der Raumänderung betroffen ist/sind sowie die Nummer des neuen und ursprünglichen Raums kommuniziert. (vgl. Abb. 14)
 - **Ausfall:** Es wird erwähnt, wann die Lektion stattgefunden hätte und ob eine oder mehrere Freilektionen dadurch entstanden sind. (vgl. Abb. 15)
 - **Nachrichtenänderung:** Wenn von der Schule eine Nachricht zur Änderung der Lektion hinzugefügt wurde (z.B. «LP krank, siehe Auftrag Teams»), wird diese von Timely ebenfalls mit einer kurzen Änderungsmitteilung kommuniziert.
 - **Verschiebungsvorschläge:** Falls Lektionen verschoben wurden und dadurch Freilektionen entstanden sind, sucht Timely automatisch nach Verschiebungsmöglichkeiten. Dabei prüft es alle Randlektionen der nächsten sieben Tage auf ihr Verschiebungspotenzial. Die Resultate werden dann ebenfalls in der Teams-Nachricht mitgeteilt (vgl. Abb. 15). Ausserdem ist jede Verschiebungsmöglichkeit mit einem Link verknüpft. Dieser öffnet eine vorgefertigte Anfrage zur Verschiebung der Lektion(en) in einem Teams-Chat mit der Lehrperson deren Stunden verschoben werden könnten. Somit können die Schülerinnen oder Schüler einfach auf den Link klicken, bei Bedarf die Nachricht anpassen und die Lehrpersonen unkompliziert anfragen. (vgl. Abb. 16)
- Sollte eine Verschiebungsmöglichkeit auf einen Tag fallen, an dem die betroffene Lehrperson keine Lektionen hält, wird darauf ebenfalls in der Teams-Benachrichtigung von Timely hingewiesen.

Abb. 17: Startbildschirm Timely Webapp



timely.school



wiesen. In seltenen Fällen oder bei komplexen Stundenplanänderungen kann es vorkommen, dass Timely keinen Verschiebungsvorschlag findet. Dann enthält die Nachricht von Timely einen Link ins Intranet zum Stundenplan. Dort können die Schülerinnen und Schüler selbst nach einer Lösung suchen.

Im seltenen Fall von mehreren gleichzeitigen Verschiebungen versucht Timely eine möglichst übersichtliche Zusammenfassung der Änderungen zu erstellen, inklusive allfälliger automatisch generierter Verschiebungsmöglichkeiten.

5.2. Webapplikation

Nebst den Teams-Kanälen für Stundenplanänderungen, gibt es auch eine Webanwendung (vgl. Abb. 17). Sie kann über die URL <https://timely.school> erreicht werden. Diese Webanwendung können Schülerinnen und Schüler nutzen, um einerseits die Verfügbarkeit von Räumen im KBW zu überprüfen oder andererseits, um alle bevorstehenden Abweichungen vom Grundstundenplan anzusehen.

Startbildschirm

Beim Öffnen der Webapplikation hat man die Möglichkeit zwischen den zwei Teilapplikationen zu wählen. Wenn man später wechseln möchte, kann dies über den Button ① mit dem jeweiligen Icon oben links gemacht werden.

Der Installations-Button ② hilft dabei die Webapplikation auf dem Home-Bildschirm des Smartphones, Tablet oder Laptops zu installieren. Nach dem Anklicken des Buttons erhält man eine gerätespezifische Anleitung dafür.

Abb. 18: Raumverfügbarkeitsanzeige Timely Webapp

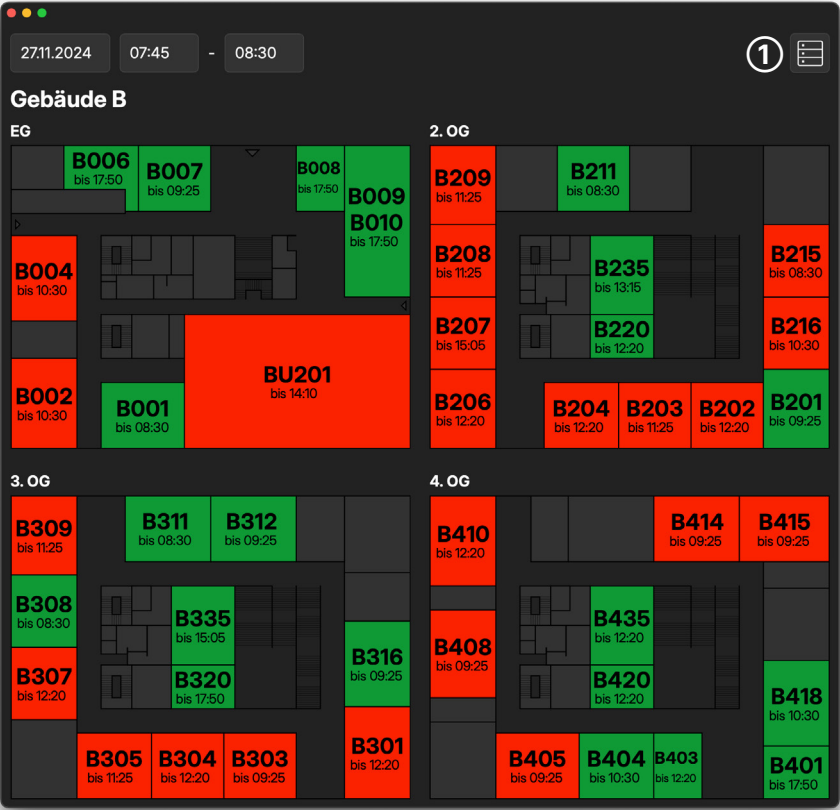
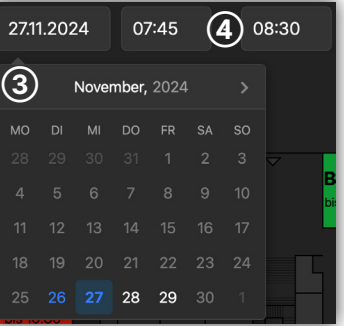


Abb. 19: Dropdown-Menü



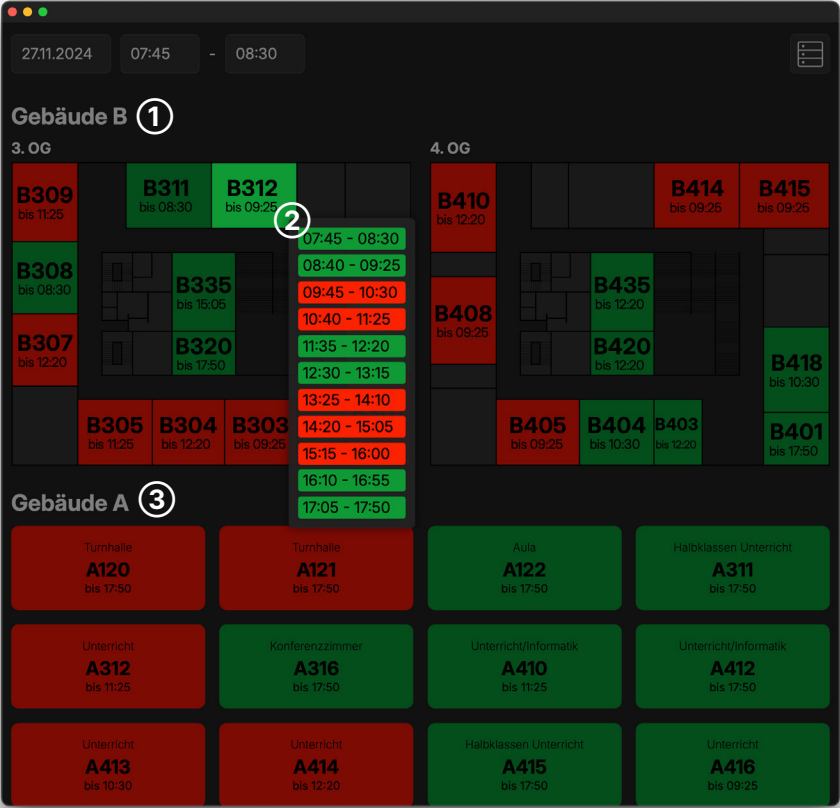
Raumverfügbarkeitsanzeige

Die Raumverfügbarkeitsanzeige ist sehr nützlich, wenn man z.B. als Schülerin oder Schüler am Mittag oder während Freilektionen ein freies Zimmer sucht, um dort zu essen oder zu arbeiten.

Um ein Datum auszuwählen, an welchem man die Raumverfügbarkeit überprüfen möchte, muss man die Datumsauswahl öffnen ③. Standardmässig wird der aktuelle Tag als Datum ausgewählt, ausser wenn der Schulunterricht bereits beendet ist. In diesem Fall wird das Datum des Folgetages ausgewählt.

Um eine oder mehrere Lektionen auszuwählen, für die man eine Raumverfügbarkeit überprüfen möchte, muss man den Anfang und das Ende der Lektionen im jeweiligen Dropdown-Menü ④ auswählen. Standardmässig wird die aktuelle Lektion ausgewählt oder – falls momentan keine stattfindet – die nächste Lektion. Wenn man einen anderen Lektionsbeginn im Dropdown-Menü auswählt, passt sich das Lektionsende automatisch an. Um mehrere Lektionen auszuwählen, kann das Lektionsende beliebig angepasst werden, beispielsweise um einen Raum für eine Doppelktion zu suchen. Bei jeglicher Datums- oder Zeitänderung werden die Verfügbarkeitsdaten aktualisiert und die Farbkodierung der Räume aktualisiert.

Abb. 20: Darstellung der Räume und Raumstundenplan Timely Webapp



Darstellung der Räume

Aufgrund der unterschiedlichen Struktur des A- und B-Gebäudes habe ich eine jeweils angepasste Darstellung gewählt.

Da das B-Gebäude ① eine Vielzahl unterschiedlicher Räume mit verschiedenen Funktionen hat (z.B. Chemielabor, Musikzimmer, etc.), die stark genutzt werden, gibt es in Timely interaktive Grundrisse jedes Geschosses. Ist ein Raum grün markiert, ist er frei – wenn er rot markiert ist, besetzt. Die Uhrzeitangaben unter dem Raumnamen zeigt jeweils an, wie lange der aktuelle Status gültig ist. Der Raum B312 ② zum Beispiel ist gerade frei, aber nur bis 09:25.

Im A-Gebäude ③ sind nur wenige Räume für Schülerinnen und Schüler relevant und diese sind auf mehreren Stockwerken verteilt. Daher wurde hier auf einen Grundrissplan verzichtet. Die zur Verfügung stehenden Räume können über interaktive Schaltflächen angewählt werden, die über die gleichen Funktionalitäten verfügen, wie bei den Räumen im Grundriss des B-Gebäudes. Damit man weiss um was für einen Raum es sich handelt, steht über jeder Raumnummer die Bezeichnung des Raumes.

Jeder Raum im Grundriss des B-Gebäudes sowie die Schaltflächen für die Räume im A-Gebäude können angeklickt werden, um den Raumverfügbarkeitsplan des ganzen Tages in einem farb-kodierten Stundenplan ② anzuzeigen.

Abb. 21: Abweichungen vom Grundstundenplan Timely Webapp

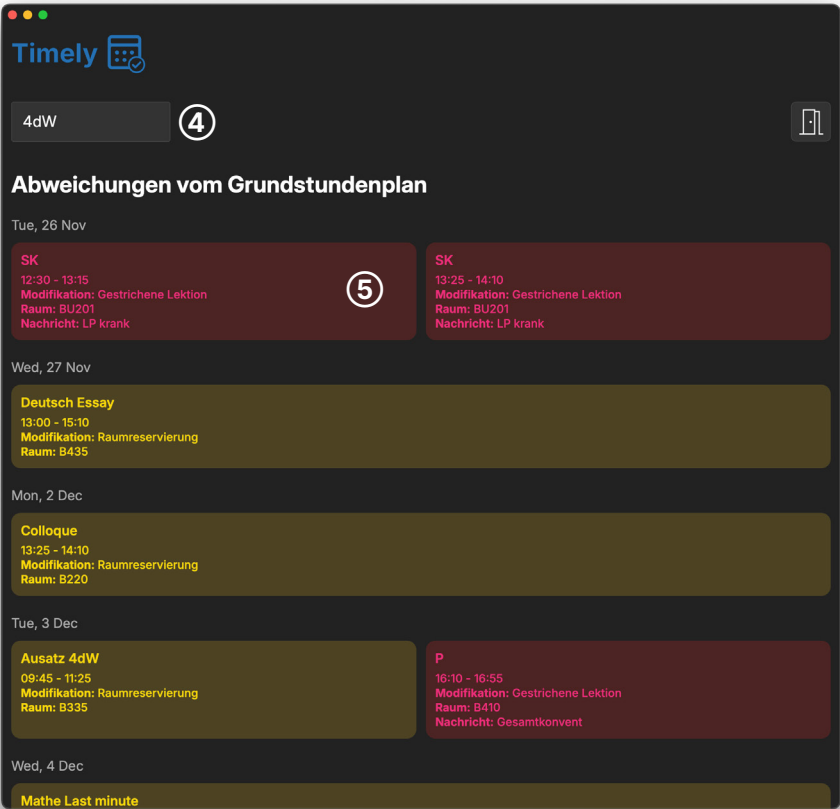
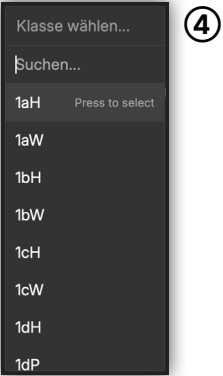


Abb. 22: Dropdown-Menü



Abweichungen vom Grundstundenplan

Die zweite Teilbereich der Webapplikation dient dazu den Schülerinnen und Schülern einen Überblick über bevorstehende Änderungen im Stundenplan zu verschaffen.

Mit dem Dropdown-Menü und dem Suchfeld ④ kann man rasch die gewünschte Klasse finden und sich Abweichungen vom Grundstundenplan anzeigen lassen.

Die angezeigten Abweichungen ⑤ vom Grundstundenplan liefern alle wichtigen Informationen dazu: Name der Unterrichtslektion, Zeitspanne der Abweichung, Modifikationstyp und – je nach Änderung – noch eine Nachricht der Lehrperson, wie z.B. «LP krank, siehe Auftrag Teams»,. Ausserdem sind sie je nach Modifikationstyp verschieden eingefärbt, um den Überblick zu erleichtern.

6. Entwicklungsprozess und Programmierung

In diesem Kapitel erläutere ich den Prozess der eigentlichen Entwicklung des Tools nach den Umfrageergebnissen und die genaue Funktionsweise der Serverseite (also dem Backend) und der Webapplikation.

Zur besseren Verständlichkeit folgt hier zuerst eine Erklärung der Entwicklungsgrundsätze und den später verwendeten Fachbegriffen. Erklärt wird die Programmierung in diesem Kapitel bewusst nur mithilfe von Flowcharts. Diese illustrieren die Abläufe und Prozesse einfach verständlich. Dabei zeigen die Beschriftungen der Pfeile jeweils welche Daten ausgetauscht werden.

Wahl der Programmiersprache

Das ganze Backend wurde mit der Programmiersprache NodeJS entwickelt. NodeJS ist eine Version von JavaScript und eine Programmiersprache, die eigentlich nur für die Verwendung in Webbrowsern gedacht ist, aber speziell für die Benutzung für Server angepasst wurde. Ich habe mich für diese Programmiersprache entschieden, da ich bereits gute Vorkenntnisse hatte und sie sich für dieses Tool perfekt eignete.

Hosting

Gehostet werden die verschiedenen Programme und Server von Timely aktuell bei mir Zuhause auf einem Mac Mini. Dies ermöglicht ein kostengünstiges Hosting, ohne dabei auf Drittanbieter zurückgreifen zu müssen. Die verschiedenen Programme und Server von Timely laufen isoliert in sogenannten «Containern» und können aber, sofern von mir konfiguriert, miteinander kommunizieren. Dafür verwende ich die Software «Docker». Sie hat den Vorteil, dass ich die Container später auch ohne Probleme bei externen Anbietern hosten könnte. Möglich ist dies, weil jeder Container sein eigenes kleines Betriebssystem hat. Einer der Container betreibt dabei einen Statusmonitor, der jede Minute überprüft, ob alle anderen Container korrekt laufen. Dafür habe ich das Programm Kuma Status Monitor benutzt (vgl. Abb. 23).

Erklärung Fachbegriffe

JSON

«JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben sowie für Maschinen einfach zu parsen (analysieren von Datenstrukturen) sowie zu generieren ist.»¹

Bei JSON können Daten entweder als Objekte mit verschiedenen Parametern und Werten oder als Array, also einer Liste von Objekten und/oder Parametern aufgebaut werden.² Der Stundenplan im Intranet der KBW z.B. ist eine Liste mit Objekten (den Lektionen), diese Objekte enthalten wiederum Parameter (z.B. Name der Lektion, Raumname, Start-Datum, End-Datum, usw.) (vgl. Abb. 24).

¹ JSON.org, Einführung in JSON, Internet.

² Vgl. JSON.org, Einführung in JSON, Internet.

Abb. 23: Timely Status Website

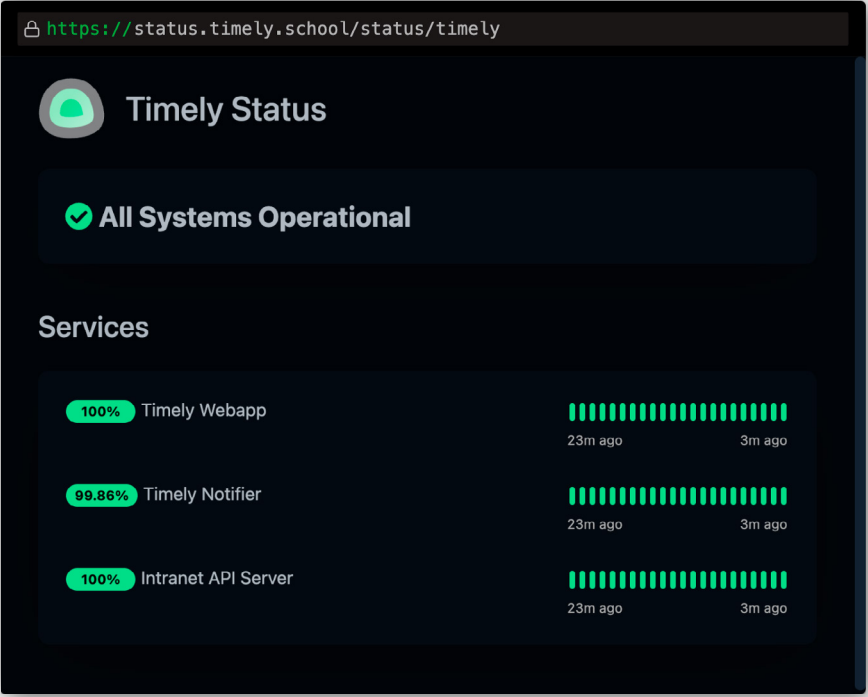


Abb. 24: Beispiel eines Stundenplans im Format JSON

```
[
  {
    "id": 1206136, //Lektions ID (für jede Lektion anders)
    "title": "WR", //Lektionstitel (kurz)

    "lessonStart": "2024-11-26T09:45:00", //Lektionsstart
    "lessonEnd": "2024-11-26T10:30:00", //Lektionsende
    "timetableEntryTypeLong": "canceled lesson", //Lektionstyp
    "message": "LP Krank", //Nachricht zur Lektion (optional)
    "room": "B303", //Raumname
    "teacher": [ //Liste von Lehrpersonen
      4001306 //Lehrpersons ID
    ],
    "course": "Wirtschaft und Recht, SP (4eW)", //Lektionstitel (lang)
    "elementId": 22005 //Element ID (bei Doppellektionen gleich)
  },
  ... //Hier wären weitere Lektionen
]
```

Asynchrone Ausführung von Code

Asynchrone Ausführung von Code bedeutet, dass mehrere Unterprogramme eines Hauptprogrammes unabhängig voneinander ausgeführt werden können. Dies kann für jede Funktion in einem Programm aktiviert werden, dabei ist aber darauf zu achten, dass das Programm nicht eine Funktion ausführt, ohne dass die dafür benötigten Daten, die z.B. von einer anderen Funktion generiert werden, vorhanden sind.¹

In meinem Tool verwende ich die asynchrone Ausführung z.B. beim Abrufen einer grösseren Zahl von Stundenplänen oder den Raumverfügbarkeiten. Die asynchrone Ausführung erlaubt in diesem Fall eine effizientere und schnellere Ausführung des Programmes, da mehrere Anfragen an den Intranet Server gleichzeitig ausgeführt werden können.

API

«APIs (Application Programming Interfaces oder Programmierschnittstellen) bestehen aus mehreren Definitionen und Protokollen zur Entwicklung und Integration von Anwendungssoftware. Eine API ist eine Schnittstelle, die es unabhängigen Anwendungen ermöglicht, miteinander zu kommunizieren und Daten auszutauschen.»²

PWA

Eine PWA (Progressive Web App) ist eine Webapplikation, die als App auf vielen Geräten installiert werden kann und so ausserhalb eines Browsers ohne Adresszeile und andere Bedienelemente dargestellt werden kann. Ausserdem erleichtert sie den Zugriff auf die Webapplikation, da diese vom Home-Bildschirm des Mobiltelefons, Tablets oder Laptops aus geöffnet werden kann.³

Webhook

Ein Webhook ist eine Anfrage an einen Server, die eine Aktion auslöst, beispielsweise die Automation in Microsoft PowerAutomate, die später erläutert wird.

Library

Eine Library ist eine Sammlung von vordefiniertem Code, der einfach wiederverwendet werden kann. Meistens sind diese für spezifische Anwendungen entwickelt worden.⁴ Der Grossteil der Libraries stammt aus frei verfügbaren Open-Source Projekten im Internet oder den jeweiligen Software-Anbietern. Die wichtigsten in meinem Tool verwendeten Libraries sind im folgenden Abschnitt angegeben.

¹ UX-REPUBLIC, Async/Await anhand von Diagrammen und Beispielen erklärt, Internet

² Red Hat Inc., Was ist eine API?, Internet

³ Vgl. Sana Commerce, PWA (Progressive Web App), Internet

⁴ Vgl. Meltzer, What is a Programming Library?, Internet

Benutzte Libraries

Intranet Library

Um die Authentifikation und den Datenzugriff aufs Intranet einfacher zu gestalten, habe ich eine Library programmiert, die diesen Prozess in meinen anderen Programmen vereinfacht. Diese wurde mithilfe von Reverse Engineering entwickelt. Das heisst, ich habe die Intranet-Website genau analysiert, um herauszufinden, wie mein Webbrowser die Daten beim Intranet-Server abfragt. Dies brauchte ich später in fast allen Programmen des Backends. Im Arbeitsprozess war die Entwicklung dieser Library einer der ersten Schritte, da ich mein Tool ohne die Abfrage von Intranet-Daten nicht hätte programmieren können.

Microsoft Teams/Power Automate Library

Für das einfache Senden von Teams-Nachrichten mit Microsoft PowerAutomate (vgl. Abb. 26: Flowchart in Kap. 6.1.) habe ich eine kleine Library programmiert.

Google AI Generative Language API Library

Diese Library habe ich verwendet, um die generative KI in meinem Programm benutzen zu können. Eine generative KI ist eine KI die mit einem Text Input, einem sogenannten «Prompt» einen Output erzeugen kann, wie dies z.B. ChatGPT tut.

Dotenv

Diese Library ermöglicht es Zugangsdaten oder andere sensitive Daten aus Sicherheitsgründen in einem separaten File zu speichern und im Programm abzurufen. Die Daten sind in einer Datei mit dem Kürzel .env abgelegt.

Lodash

Diese Library ermöglicht es einfacher mit JSON-Dateien zu arbeiten, diese beispielsweise zu bearbeiten, anzupassen, zu vergleichen oder zu filtern.

Express

Diese Library ermöglicht es auf einfache Weise einen Server in NodeJS zu erstellen. Das habe ich bei dem weiter hinten beschriebenen Timely-API-Server benutzt (vgl. Kap 6.2.).

Air Datepicker

Diese Library ermöglicht es auf einfache Weise eine benutzerdefinierte Auswahlfunktion für das Datum in einer Website zu integrieren.

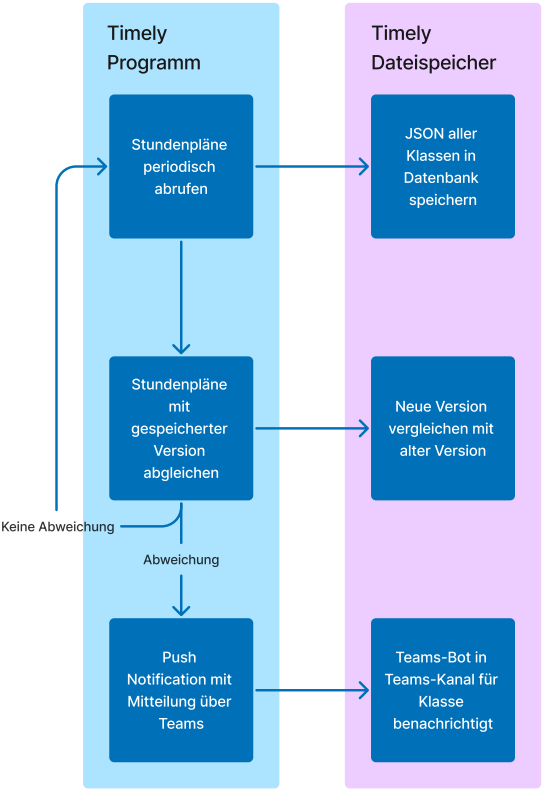
Choice.js

Diese Library ermöglicht es auf einfache Weise eine Dropdownliste mit Suchfunktion in einer Webapplikation zu integrieren.

Add-to-homescreen

Diese Library ermöglicht es gerätespezifische Anleitungen zur Installation einer PWA bereitzustellen. Dies ist nötig, da das Installationsprozedere in jedem Webbrowser anders funktioniert.

Abb. 25: Erste Version des Flowcharts für die automatischen Verschiebungsvorschläge



6.1. Automatische Verschiebungsvorschläge und Teams-Benachrichtigung

Als erstes erstellte ich einen Flowchart, um zu visualisieren, wie dieser Teil des Tools grundsätzlich funktionieren soll (vgl. Abb. 25). Im Verlauf der Entwicklung des Tools waren aber einige Änderungen notwendig. Diese Anpassungen und die Gründe dafür erkläre ich im Folgenden mit Hilfe des finalen Flowcharts. (vgl. Abb. 26 nächste Seite)

Ursprünglicher Flowchart

In der ersten Version des Flowcharts (vgl. Abb. 25) kann man sehen, wie ich mir zuerst vorstellte, dass das Programm später funktionieren wird. Die Stundenpläne sollen periodisch abgerufen und mit einer alten Version verglichen werden. Falls Änderungen festgestellt werden, soll die betroffene Klasse im jeweiligen Teams-Kanal benachrichtigt werden. Bei dieser Version fehlen die automatischen Verschiebungsvorschläge noch vollständig.

Finaler Flowchart

In diesem Abschnitt wird der finale Flowchart erklärt für die automatischen Verschiebungsvorschläge und die Änderungsbenachrichtigungen. Der Flowchart (vgl. Abb. 26 nächste Seite) befindet sich aus Platzgründen auf der folgenden Doppelseite.

Intranet Login ①

Um Zugriff auf die Stundenpläne im Intranet zu erhalten, muss sich das Programm einmalig eine Session-ID mit meinen persönlichen Zugangsdaten erstellen. Eine solche ID ist ein Zugangsschlüssel, der vom Intranet-Server generiert wird, um mich zu identifizieren. Dieser Loginprozess wird mit meiner eigens dafür entwickelten Intranet-Library getätigt und benutzt meine persönlichen Intranet-Zugangsdaten aus der .env-Datei.

Stundenplan abrufen ②

Nachdem der Zugangsschlüssel für das Intranet angefragt und bereitgestellt wurde, ruft mein Programm jede Stunde von 6 Uhr bis 22 Uhr die aktuellen Stundenpläne der nächsten sieben Tage für die von mir definierten Klassen ab. Die so generierten Daten werden dann im Format JSON gespeichert.

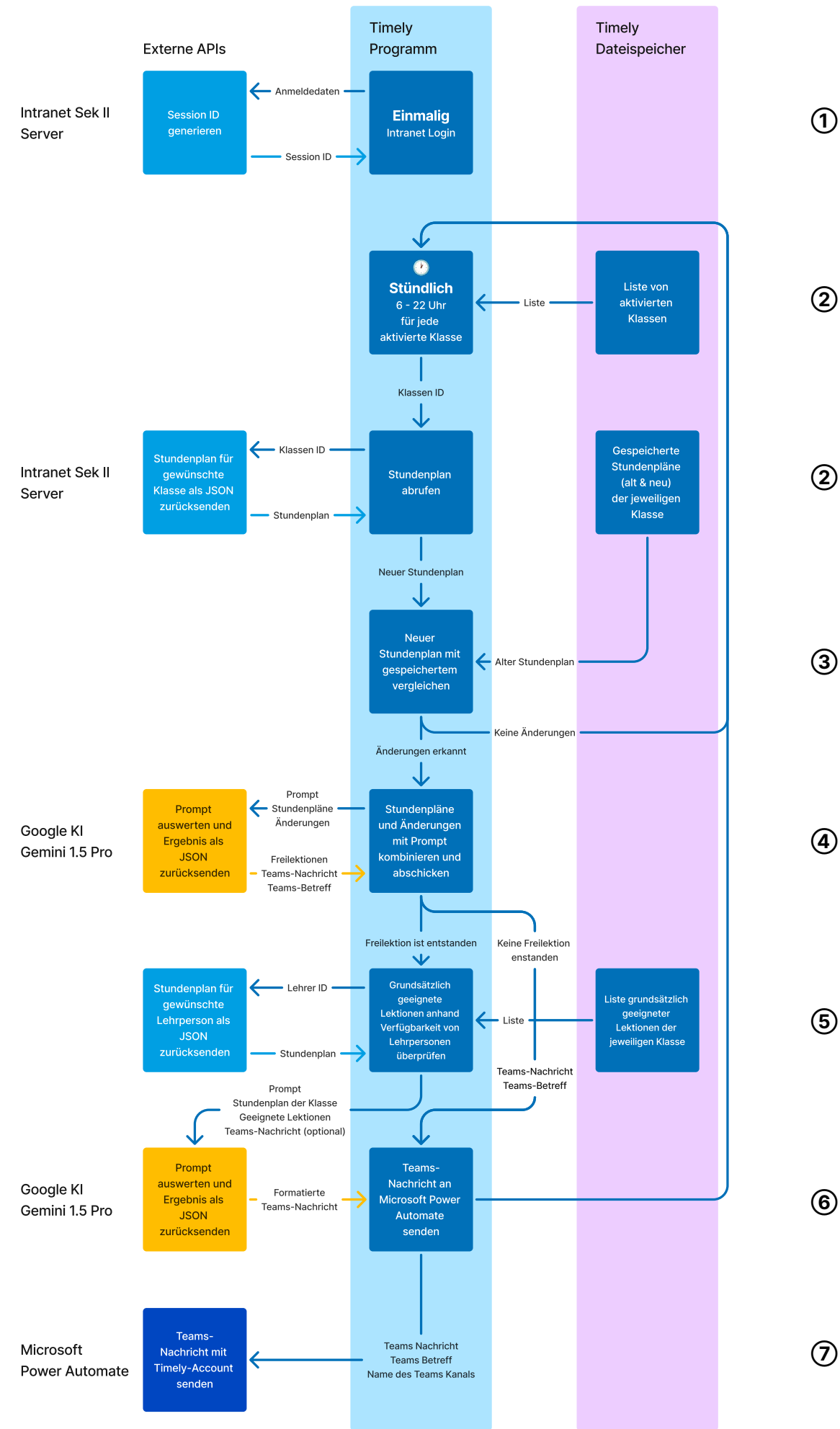
Neuer Stundenplan mit gespeichertem vergleichen ③

Sobald die aktuellen Stundenpläne abgefragt wurden, erfolgt der Abgleich mit denjenigen von einer Stunde zuvor. Dazu wird die Library «LoDash» verwendet, in der die Daten der vorherigen Abfrage abgelegt sind. Falls eine neue Lektion hinzugefügt oder eine bestehende modifiziert wurde, wird dies in einer temporären Konstante gespeichert.

Stundenpläne und Änderungen mit Prompt kombinieren und abschicken ④

Hat das Tool Änderungen erkannt, wird ein Prompt, also die Anweisung an die KI zusammen

Abb. 26: Flowchart des Programms für die automatischen Verschiebungsvorschläge



mit den Veränderungen, sowie mit der alten und mit der neuen Version des Stundenplans einer generativen KI (Gemini 1.5 Pro, siehe Kap. 7) geschickt. Der Prompt enthält Instruktionen, um die Änderungen zu analysieren und herauszufinden, ob dadurch neue Freilektionen entstanden sind. Ist dies der Fall, werden alle Freilektionen als JSON-Objekt gelistet. Gibt es keine Freilektion, wird bereits in diesem Schritt eine Teams-Nachricht generiert und anschliessend an die Schülerinnen und Schüler der betroffenen Klasse gesendet. Im seltenen Fall von mehreren Modifikationen auf einmal, die nicht alle eine Freilektionen generieren, werden beide Resultate ausgegeben. Dadurch ist sichergestellt, dass später die Modifikation, die keine Freilektionen generiert hat, nicht in der Teams-Mitteilung fehlt.

Grundsätzlich geeignete Lektionen anhand Verfügbarkeit von Lehrpersonen überprüfen ⑤

Falls Freilektionen entstanden sind, wird anhand einer Liste der grundsätzlich geeigneten Lektionen überprüft, welche davon sich verschieben lassen würden. Diese Liste von grundsätzlich geeigneten Lektionen wurde von mir für jede Klasse separat definiert. In einer früheren Version versuchte ich diese Liste zuerst mit einer programmatischen Logik zu generieren. Dies funktionierte allerdings nur suboptimal, da es schwierig ist für jede Klasse zu erkennen, welche Halbklassen-Lektionen, Ergänzungsfächer und Freifächer es gibt und welche davon nicht zu den geeigneten Randlektionen zählen. Ebenso war es nicht einfach Doppellectionen als solche zu erkennen, damit diese nicht fälschlicherweise als einzelne Lektionen verschoben werden.

In einem nächsten Schritt probierte ich die Aufgabe mithilfe generativer KI zu lösen, dies erzielte jedoch keine zuverlässigen Resultate für Lektionsverschiebungen, die wirklich geeignet waren. Schlussendlich entschied ich mich für jede Klasse, für die das Tool aktiviert ist, selbst eine Liste von grundsätzlich geeigneten Lektionen zu erstellen. Dabei achtete ich darauf, dass diese Schulstunden Randlektionen sind und dass auch Randlektionen inkludiert sind, auf die Freifächer folgen. In der Liste sind Einzel- und Doppellectionen separat aufgeführt, damit Letztere beim Verschieben nicht aufgesplittet werden. Durch diese von Hand erstellten Listen, konnte ich schlussendlich Daten bereitstellen, mit denen sich die Lektionen zuverlässig verschieben lassen.

Um die grundsätzlich geeigneten Lektionen nun zu überprüfen, wird die Liste zuerst gefiltert, um Lektionen temporär von der Liste zu entfernen, die in den nächsten sieben Tagen nicht stattfinden oder bereits verschoben wurden. Als nächstes wird für jede Lehrperson der geeigneten Lektionen der Lehrerstundenplan für die nächsten sieben Tage vom Intranet abgefragt. Danach prüft das Tool für jede entstandene Freilektion, bei welchen von den geeigneten Lektionen die Lehrperson keinen Unterricht hat. Ist dies der Fall, wird die geeignete Lektion einer Liste hinzugefügt. Gleichzeitig vermerkt das Tool, ob die Lehrperson an diesem Tag andere Lektionen hat oder ob es ein freier Tag wäre. Dies wird später dann in der Teams-Nachricht an die Schülerinnen und Schüler als Hinweis vermerkt.

Teams-Nachricht an Microsoft Power Automate senden ⑥

Nachdem das Tool all diese Schritte ausgeführt hat, benutzt es erneut die generative KI um eine Teams-Nachricht mit den Verschiebungsvorschlägen zu erstellen. Der Prompt enthält die ursprüngliche Stundenplanänderung, die Liste der Freilektionen, die Liste der geeigneten Lektionen und eine Liste aller E-Mail-Adressen der Lehrpersonen, die die geeigneten Lektionen

unterrichten. Optional erhält der Prompt auch die ursprünglich generierte Teams-Nachricht aus Schritt ④, damit Änderungen, die keine Freilektionen generiert haben, ebenfalls in der finalen Teams-Nachricht erwähnt werden. Zusätzlich zu diesen Daten wird die generative KI im Prompt instruiert, wie die Teams-Nachricht formuliert und formatiert werden soll.

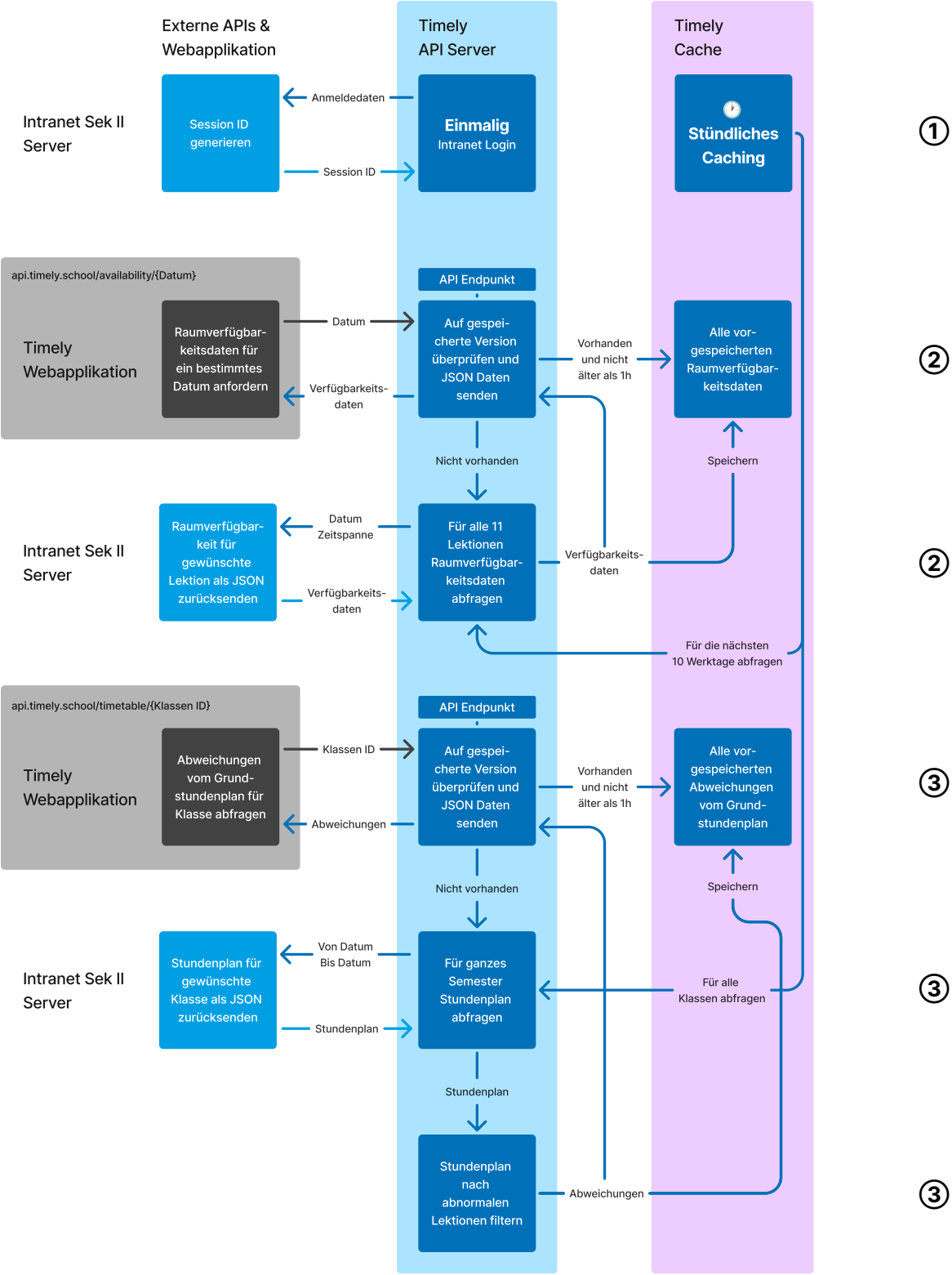
Teams-Nachricht mit Timely-Account senden ⑦

Um die finale Teams-Nachricht zu verschicken, oder diejenige aus Schritt ④, falls es keine Freilektionen gibt, benutze ich ein Tool von Microsoft namens PowerAutomate. Dabei handelt es sich um eine Plattform, die dazu benutzt werden kann, um Automationen für Microsoft Produkte zu erstellen. Dort habe ich eine Automation erstellt, die, wenn sie Daten über einen Webhook empfängt, eine Teams-Nachricht mit dem Teams-Account von Timely im Kanal der jeweiligen Klasse versendet - mit dem Text und dem Betreff, die in den Daten spezifiziert wurden. Dieser Webhook wird von meinem Programm ausgelöst und enthält die notwendigen Daten zur Erstellung der Teams-Nachricht.

In einer früheren Version des Codes, hatte ich versucht zuerst anstatt eines normalen Teams-Account, einen Bot zu benutzen, um Nachrichten in die Kanäle der jeweiligen Klasse zu schicken. Ein Teams-Bot ist ein Weg, um Nachrichten programmatisch zu versenden. Dies stellte sich jedoch nach einiger Zeit als nicht geeignet heraus, da diese Bots zum einen keine Teams-Kanäle markieren und somit schlussendlich auch keine Benachrichtigungen in privaten Kanälen hätten schicken können.

Die Folge wäre gewesen, dass ich für jede Klasse ein einzelnes Team hätte erstellen müssen. Mit dem «Workaround» den ich gefunden habe, der mithilfe von Microsoft PowerAutomate Nachrichten verschicken kann, war es auf viel einfachere Weise möglich, alle von mir gewünschten Anforderungen für den Versand der Benachrichtigungen in Teams zu erfüllen.

Abb. 27: Flowchart des Timely-API-Servers



6.2. Timely API Server

Damit die Webapplikation `timely.school` die Daten des Intranets effizient und ohne zusätzliches Login auf der Nutzerseite verwenden kann, braucht es einen Server, der die benötigten Daten bereitstellt. Dieser Server ist also ein API-Server, der es der Webapplikation ermöglicht indirekt mit dem Intranet zu kommunizieren. Dieser Server ist über die URL `api.timely.school` erreichbar. Der Flowchart für diesen Server ist in Abb. 27 zu sehen. Er verfügt über zwei sogenannte Endpunkte, also die Punkte, die meine Webapplikation verwendet um die Daten des Intranets indirekt abzurufen.

Intranet Login ①

Wie beim Programm für die automatischen Verschiebungsvorschläge, braucht dieser Server ebenfalls eine Session ID um sich gegenüber dem Intranet-Server identifizieren zu können. Dies wird mit derselben Intranet Library getätigt, die ich bereits beim Programm für die automatischen Verschiebungsvorschläge benutzt habe.

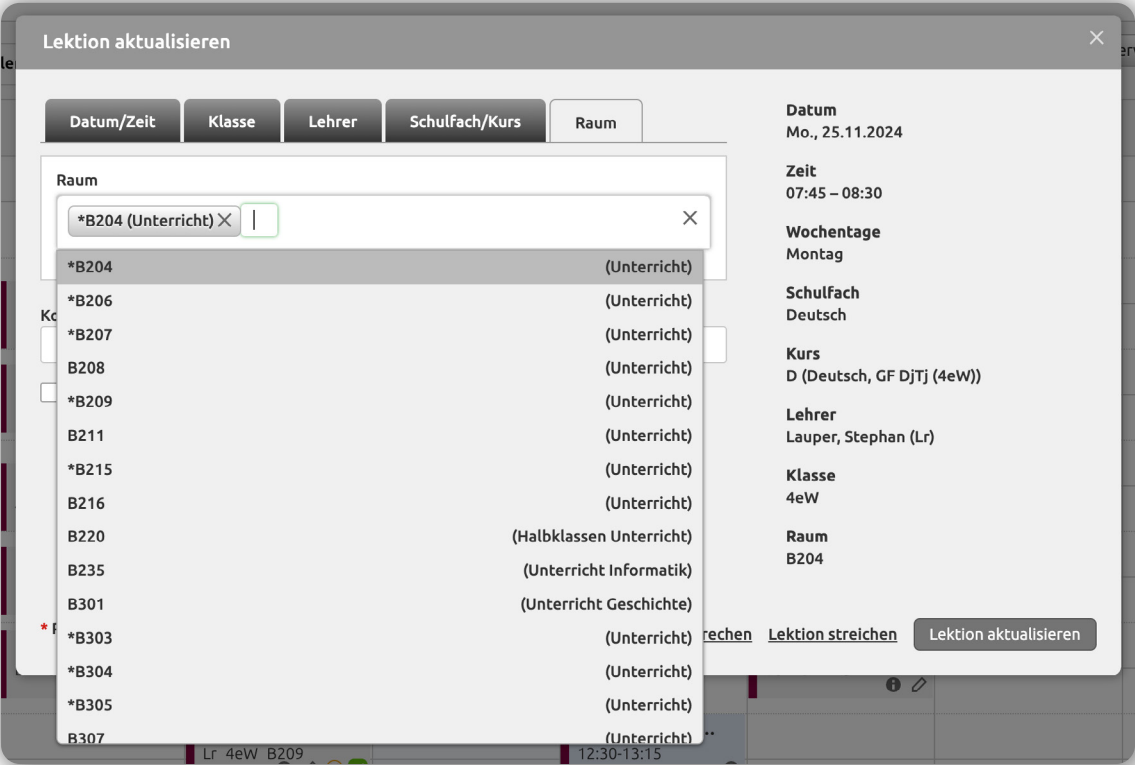
Raumverfügbarkeitsdaten für ein bestimmtes Datum anfordern ②

Dieser Endpunkt meldet die Verfügbarkeitsdaten aller Räume im KBW für ein bestimmtes Datum als JSON-Datei zurück.

Um die Raumverfügbarkeitsdaten vom Intranet-Server abzufragen, musste ich zuerst mithilfe von Reverse Engineering herausfinden, wie ich diese Daten vom Intranet-Server abfragen kann. Beim Untersuchen der Intranet-Website ist mir aufgefallen, dass es eine versteckte Einstellung gibt, die regelt, welche Funktionen für den momentanen Benutzer verfügbar sind. Diese habe ich so modifiziert, dass mir die Optionen zum Verschieben von Lektionen angezeigt wurde - also die gleiche Ansicht, die auch eine Stundenplanerin oder ein Stundenplaner hätte (vgl. Abb. 28 nächste Seite). Wenn sie oder er eine Lektion verschieben möchte, öffnet sich ein Dialogfenster, in dem man die Änderungen eintragen kann. Teil dieses Dialogfensters ist eine Liste für die Raumauswahl. Wenn man diese Liste öffnet, werden alle Räume angezeigt. Diejenigen, die zur Zeit der zu modifizierende Lektion nicht verfügbar sind, sind dabei mit einem «*» markiert. Als ich dann geprüft habe, mit welcher Anfrage (API-Request) diese Verfügbarkeit abgerufen wird, konnte ich dieselbe in meine Intranet Library einbauen und im Timely-API-Server verwenden.

Dieser API-Request hat zwei Parameter, einer für das Startdatum/Startzeit und einer für das Enddatum/Endzeit während derer die Verfügbarkeit überprüft werden soll. Dieser API-Request meldet dann für jeden Raum, ob er während des abgefragten Zeitraums verfügbar ist. Da ich aber die Verfügbarkeit für jede einzelne Lektion benötige, muss ich pro Tag für alle elf Lektionen einen separaten API-Request ausführen. Deshalb ist es bei der Funktion, die die Raumverfügbarkeiten abfragt, besonders wichtig, den Code asynchron auszuführen. Nach der erfolgreichen Abfrage für alle Lektionen werden die Ergebnisse in eine JSON-Datei zusammengeführt, gespeichert und der Webapp gesendet. Damit der API-Server nicht bei jeder Anfrage elf separate API-Requests an den Intranet-Server durchführen muss, werden die Raumverfügbarkeitsdaten «gecached», also es werden bereits im Vorhinein abgefragte Daten zurückgemeldet. Dies ist im Abschnitt «Stündliches Caching» auf der nächsten Seite beschrieben.

Abb. 28: Ansicht des Fensters zur Lektionsmodifikation



Abweichungen vom Grundstundenplan für Klasse abfragen ③

Dieser Endpunkt meldet alle Abweichungen vom Grundstundenplan für die jeweilige Klasse zurück. Die Klassenidentifikationsnummer ist dabei die gleiche, wie die vom Intranet vorgegebene und besteht bei allen Klassen aus vier Nummern. Zum Beispiel die Nummer 1477 für die Klasse 4eW.

Um die Abweichungen vom Grundstundenplan abzurufen, wird zuerst mit mithilfe der Intranet Library der Stundenplan, der jeweils von der Webapplikation abgefragten Klasse von heute bis Ende des Semesters abgerufen. Anschliessend filtert das Tool die Lektionen, welche jeweils als JSON-Objekte repräsentiert sind, sodass nur noch modifizierte Lektionen übrigbleiben. Diese erkennt man daran, dass der Parameter «TimetableEntryTypeLong» nicht «lesson» ist, sondern z.B. «canceled lesson». Diese gefilterte Version des Stundenplans wird anschliessend gespeichert und als JSON zurückgemeldet.

Stündliches Caching ①

Die Datenabfragen im Intranet beanspruchen zum Teil eine beträchtliche Zeit. Bei der Raumverfügbarkeit müssen, wie oben erwähnt, für jeden Tag elf API-Requests ausgeführt werden, was mehrere Sekunden dauern kann. Die Stundenplanabfragen brauchen auch relativ lange, da nicht nur für eine Woche, sondern für den ganzen Rest des Semesters die Daten abgefragt werden. Deshalb werden diese Daten «gecached», das heisst, einmal pro Stunde werden die Daten zur Raumverfügbarkeit für die nächsten zehn Schultage (also zwei Wochen) und die Abweichungsdaten für alle Klassen beim Intranet-Server abgefragt und auf dem Timely-API-Server gespeichert. Bei einer Anfrage an einem der zwei API-Endpunkte wird die gespeicherte Version der angefragten Daten zurückgesendet. Falls keine gespeicherte Version vorhanden, oder sie älter als eine Stunde ist, erfolgt die Abfrage regulär vom Intranet-Server mit der oben beschriebenen Angabe der zwei Endpunkte. Nur mithilfe von Caching ist es möglich, dass die Raumverfügbarkeiten und die Abweichungen vom Grundstundenplan innerhalb von weniger 100 Millisekunden abrufbar sind.

6.3. Webapplikation

Allgemein

Die eigentliche Webapplikation besteht aus zwei Teilapplikationen: Der Raumverfügbarkeitsanzeige und der Möglichkeit, die Abweichung vom Grundstundenplan anzuzeigen. Diese zwei Teilapplikationen sind aber in der gleichen HTML-Datei integriert – also derjenigen Datei, die dem Browser sagt, wie die Webseite dargestellt wird und wie sie ausgeführt werden muss. Das hat den Vorteil, dass sie sehr schnell geladen werden kann. Während das Menü mit der Auswahl zwischen den zwei Teilapplikationen den Nutzerinnen und Nutzern gezeigt wird, können im Hintergrund die Libraries Airdatepicker für das Datum-Auswahlfenster sowie «Choices.js» für das Klassen-Dropdown-Menü heruntergeladen und die Raumverfügbarkeit für das heutige Datum vom Timely-API-Server abgefragt werden. Nur so konnte die Website im «Lighthouse-Test» von Google ein nahezu perfektes Resultat erzielen. Dieser prüft verschiedene Aspekte – unter anderem die Ladegeschwindigkeit der Website, die auch Performance genannt wird. Dabei erzielte Timely in dieser Kategorie 99 von 100 Punkten. Bei den weiteren Optimierungs- und Qualitätsaspekten hat die Webapplikation dank vieler Optimierungen überall Bewertungen von mehr als 90 von 100 Punkten erreicht. Damit erfüllt sie das von mir gesteckte Ziel einer möglichst hohen Benutzerfreundlichkeit.

Programmiert wurde die Webapplikation mit der Sprache Javascript. Diese habe ich gewählt, da sie die Standardsprache für Webseiten und -apps ist und ich darin bereits Kenntnisse hatte.

Gehostet wird die Webapplikation beim Anbieter Cloudflare. Dieser ermöglicht ein kostenloses Hosting sowie aufgrund seiner weltweit verteilten Standorte – in der Schweiz sind dies Zürich und Genf – einen schnellen Zugriff auf die Applikation.¹

Bei der Darstellung und Farbwahl habe ich auf eine übersichtliche, einheitliche und ruhig wirkende Optik geachtet, die auf einen Blick die wichtigsten Informationen liefert. Dabei hielt ich mich an die von mir im definierten Design-Guidelines. Ein Beispiel dafür ist die Akzentfarbe im Fenster für die Datumsauswahl.

Die Webapplikation kann ebenfalls als PWA installiert und zum Homescreen der meisten Geräte hinzugefügt werden. Um dies zu ermöglichen, benutze ich die Library «Add-to-homescreen», welche es auf Knopfdruck erlaubt, einfach ein Fenster mit gerätespezifischen Installationsanweisungen darzustellen. Was, wie oben beschrieben, den Zugriff auf die Webapplikation erleichtert und verbessert. Ausserdem sollen so die Nutzerinnen und Nutzer animiert werden, die die Applikation regelmässig nutzen.

Raumverfügbarkeitsanzeige

Um ein auf allen Geräten ein einheitliches Auswahlfenster für das Datum zu haben, verwende ich die Library «Air Datepicker». Diese stellt ein Datum-Auswahlfenster bereit, in dem ich bei der Programmierung die Wochenenden deaktivieren konnte. Beim erstmaligen Aufrufen der Raumverfügbarkeit ermittelt das Skript für die Raumverfügbarkeitsanzeige das Datum der aktuellen oder nächsten Unterrichtslektion und wählt dieses aus. Anschliessend werden die Raumverfügbarkeitsdaten vom der Timely-API abgefragt. Um diese darzustellen, werden die Räume des B-Gebäudes in den von mir erstellten Grundrissen eingefärbt. Aufgrund der ausgewählten

¹ Vgl. Cloudflare Inc., Das globale Cloudflare-Netzwerk, Internet

Abb. 29: Datumsauswahl Dropdown-Menü

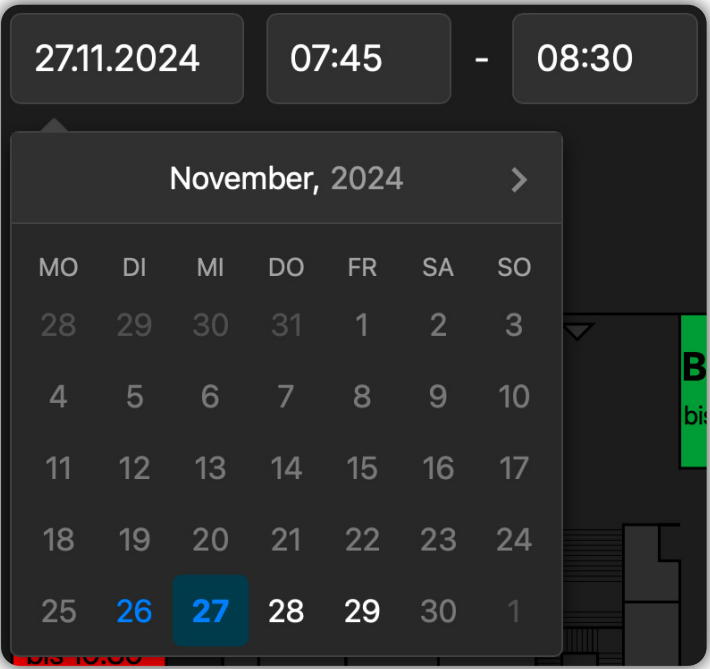
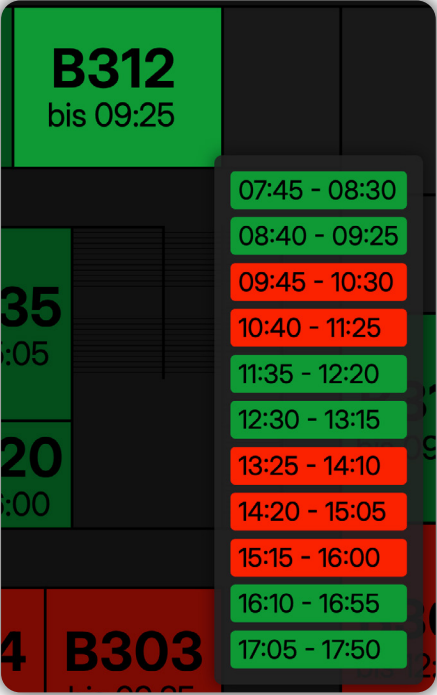


Abb. 30: Raumverfügbarkeitsanzeige im Detail



Lektion(en) wird zudem die «bis» Zeitangabe angezeigt. Dabei prüft das Skript, wie lange der Zustand (verfügbar oder besetzt) während den ausgewählten Lektion(en) bestehen bleibt (vgl. Abb. 30). Das gleiche gilt für die Räume im A-Gebäude – statt den Grundrissen erfolgt hier die Darstellung aber in Form von farbkodierten Feldern. Bei jeglicher Datums- oder Zeitänderung werden die Verfügbarkeitsdaten aktualisiert und die Farbkodierung der Räume aktualisiert.

Wenn auf einen Raum geklickt wird, egal ob auf dem Grundriss oder einem Feld, erscheint ein kleines Fenster mit einem farbkodierten Stundenplan für den ganzen Tag für diesen Raum. Dieser basiert auf den gleichen Raumverfügbarkeitsdaten. Der Rest der Webapplikation, ausser der ausgewählte Raum, wird bei der Anzeige des kleinen Stundenplanfensters verdunkelt, sodass er besser erkennbar ist.

Abweichungen vom Grundstundenplan

Um ein Dropdown-Menü mit einem Suchfeld für die Klassen darstellen zu können (vgl. Abb. 31), verwende ich die Library «Choices.js». Sie ermöglicht diese Funktionalität ohne aufwändige Zusatzprogrammierung. Sobald diese Teilapplikation zum ersten Mal geladen wird, öffnet ein Skript automatisch das Dropdown-Menü und man kann eine Klasse auswählen. Anschliessend wird eine Anfrage an den Timely-API-Server gesendet. Die empfangenen JSON-Daten werden dann so dargestellt, dass für jeden Tag ein Eintrag mit den Informationen zur Lektion erstellt wird. Zusätzlich teilt das Tool diesem Eintrag aufgrund seines Modifikationstyps eine Farbe zu (z.B. rot für eine gestrichene Lektion). Wählt die Nutzerin oder der Nutzer eine andere Klasse im Dropdown-Menü, werden wiederum automatisch die Daten für die gewünschte Klasse heruntergeladen und dargestellt.

Abb. 31: Klassenauswahl Dropdown-Menü



Abb. 32: Beispiel für eine Github Copilot Anfrage um Code zu generieren

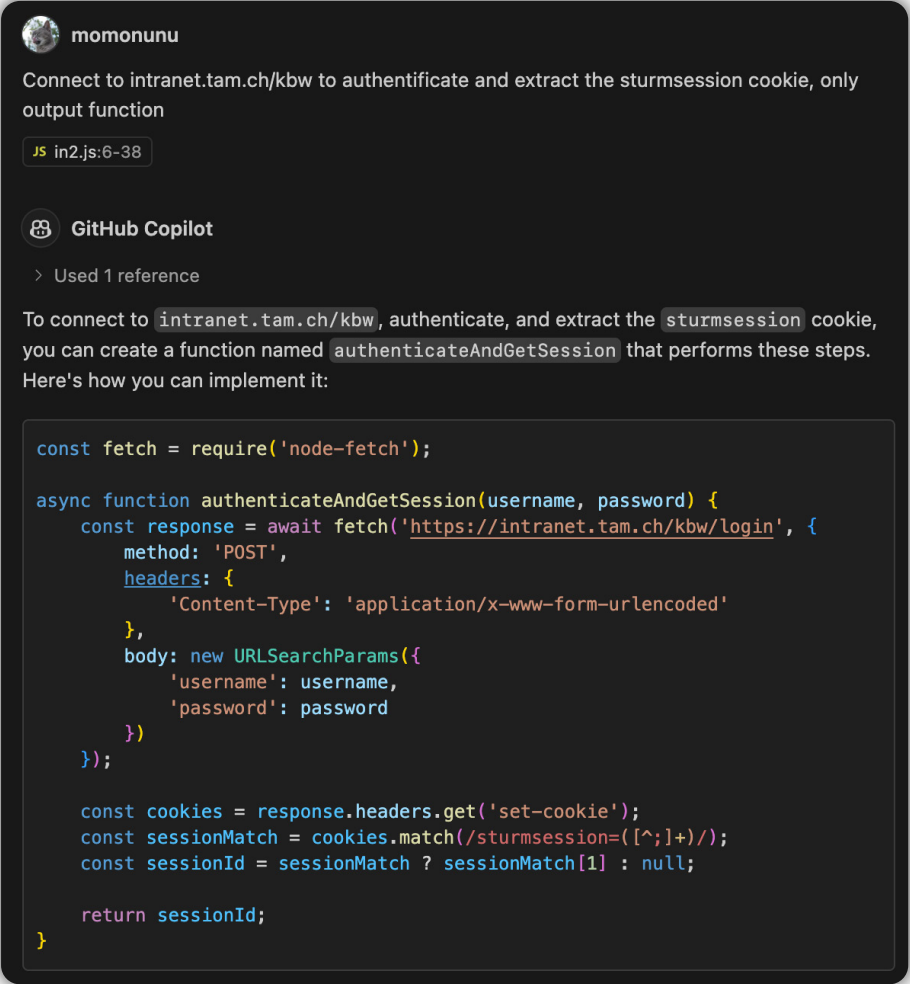
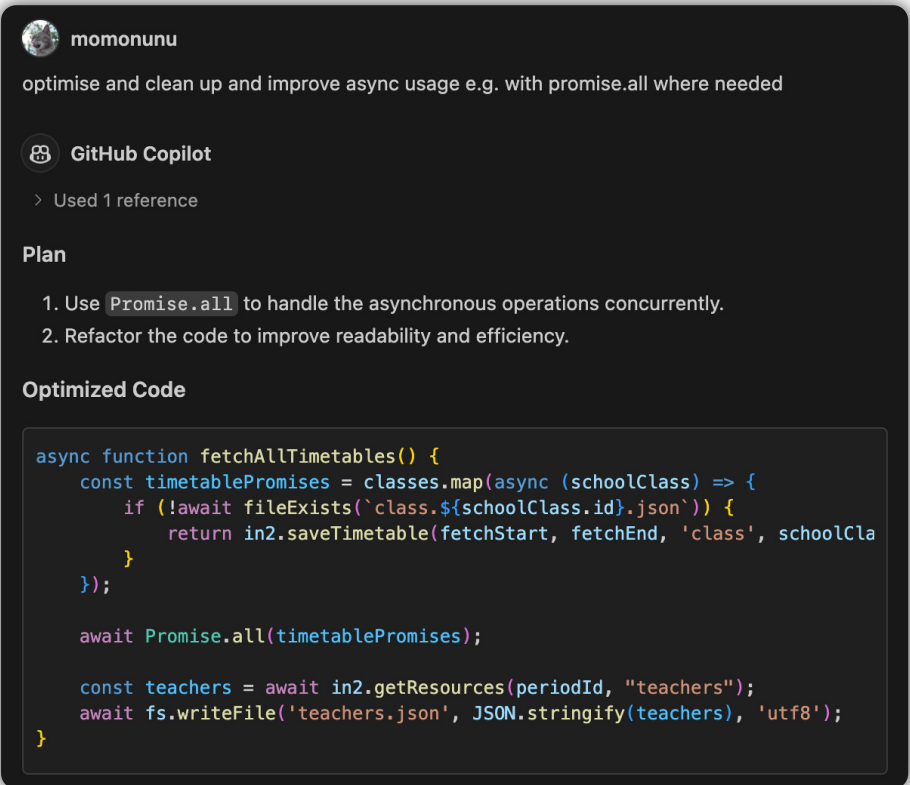


Abb. 33: Beispiel einer Github Copilot Anfrage um Code zu modifizieren/optimieren



7. Verwendung von KI

Im Timely-Tool

Für die Nutzung innerhalb des Leistungsumfangs von Timely verwende ich die Generative KI Gemini 1.5 Pro von Google. Sie ist eine der KIs mit dem bestem Kosten-Qualitäts-Verhältnis.¹ Zu Beginn der Arbeit benutzte ich Gemini 1.5 Flash, eine simplere und günstigere KI. Die durch sie erzeugten Resultate waren jedoch nicht einheitlich und zu wenig zuverlässig. Gemini 1.5 Pro kostet pro 1 Million Tokens (1 Wort entspricht ca. 0.75 Tokens) ca. 2 Franken. Eine Stundenplanverschiebungsanfrage kommt damit auf ca. 5 Rappen zu stehen. Ausserdem ist die Gemini KI äusserst datenschutzfreundlich, da die gesendeten Prompts und Daten nicht zum Training der KI verwendet werden. Die generative KI von Google ermöglicht es zudem die Antworten auf speziell vordefinierte JSON-Datenstrukturen zu begrenzen.²

Für die Entwicklung von Timely

Für die Entwicklung von Timely selbst habe ich das KI-Tool Github Copilot verwendet. Diese habe ich konkret benutzt um Teile des Code zu generieren, den Code zu kommentieren, mir bei Fehlern erklären zu lassen, was dessen Ursprung sein könnte oder den Code zu optimieren,. Github Copilot eignet sich besonders gut um kleinere Code-Abschnitte zu generieren und so einen Startpunkt für die Entwicklung zu haben. Der so generierte Code-Abschnitt kann anschliessend selbst angepasst werden. Die Unterstützung von Github Copilot hat allerdings auch seine Grenzen, manchmal funktionieren die Anpassungen am Code nicht auf Anhieb oder erzeugen nicht das gewünschte Resultat. Deshalb war das KI-Tool Github Copilot für die Entwicklung zwar äusserts hilfreich, dennoch stoss das Tool aber häufig an seine Grenzen, vor allem bei komplexeren Programmierungen. Einige Beispiele von Prompts und Antworten sind links zu finden (vgl. Abb. 32, 33).

Für den Werkbericht

Für die Optimierung der Formulierung einzelner Sätze und Paragraphen des Werkberichtes kam ChatGPT zum Einsatz.

¹ Vgl. Artifical Analysis, Quality vs. Price, Internet

² Vgl. Google Cloud, So verwendet Gemini für Google Cloud Ihre Daten, Internet

8. Schlusswort und Ausblick

Zu Beginn meiner Maturitätsarbeit hätte ich mir nicht vorstellen können, welche Komplexität das fertige Tool schliesslich haben würde. So hat sich beispielsweise, die am Anfang als Nebenfeature angedachte Webapp im Verlauf der Arbeit zu einem Hauptfeature entwickelt. Überraschend ist für mich auch, dass das fertige Tool und das, was die Nutzerin oder Nutzer sieht, sehr einfach daherkommt und man auf den ersten Blick nicht erahnt, welcher Aufwand dahinter steckt.

Timely bietet viel Potenzial zur Weiterentwicklung. So beschränken sich die automatischen Verschiebevorschläge momentan auf Lektionen, deren Anpassung den Zeitraum der jeweils nächsten sieben Tage betrifft und innerhalb dieses Zeitraumes auch modifiziert wurden. Hier wären erweiterte Funktionalitäten denkbar: So würden beispielsweise Verschiebungsvorschläge für Lektionen, die ausserhalb der Frist von sieben Tagen liegen oder schon vor mehr als sieben Tagen modifiziert wurden, das Tool perfekt ergänzen. Ein interessantes Feature wäre auch eine Nachricht, die am Sonntagabend eine Übersicht der Änderungen der kommenden Woche liefert. Ausserdem wäre eine automatische Ermittlung der grundsätzlich für eine Verschiebung geeigneten Stunden pro Klasse wünschenswert. In der aktuellen Version habe ich diese manuell herausgesucht. Der Programmieraufwand dafür würde aber wahrscheinlich in einem schlechten Verhältnis zur Zeit stehen, die für die einmal pro Jahr nötige manuelle Erstellung gebraucht wird.

Kostenmässig und im Sinne von noch besserem Datenschutz, wäre es zudem interessant die Generative KI auf meinem eigenen Server zu hosten. Das würde die Betriebskosten senken und den Datenschutz nochmals verbessern. Diese Generative KI könnte mit Daten der Schule gefüttert werden um auf Fragen von Schülerinnen und Schülern über die KBW antworten zu können.

Bei der Webapplikation sehe ich ebenfalls Weiterentwicklungspotenzial: Einerseits wäre es schön, wenn die aktuellen Raumverfügbarkeiten auch auf den Stundenplandisplays der KBW angezeigt würden. Andererseits wäre der Schritt weg von einer Webapplikation in Richtung einer App, die fest auf dem Gerät installiert werden kann, sinnvoll. Damit könnte die Benutzerfreundlichkeit von Timely noch weiter verbessert werden. Eine solche App würde ebenfalls Widgets und Push-Benachrichtigungen ermöglichen.

Timely auf andere Kantonsschulen im Kanton Zürich auszuweiten wäre einfach, da diese alle das «Intranet Sek II» benutzen. Einen gewissen Aufwand würde aber die Implementierung der jeweiligen Gebäudegrundrisse erfordern, sofern man sich nicht auf die Anzeige mit Feldern beschränkt, wie ich sie beim Gebäude A der KBW verwendet habe.

Als nächsten Schritt der Weiterentwicklung möchte ich, nach der Testphase der drei ausgewählten Klassen, das automatische Verschiebungstool für weitere Klassen im KBW aktivieren.

Danksagung

Ein grosses Dankeschön geht an meinen Betreuer Sebastian Schiendorfer für die immer schnellen Antworten auf Fragen und das entgegengebrachte Vertrauen. Auch bei meinen Eltern möchte ich mich herzlich bedanken für das Gegenlesen der Arbeit. Danke an Dominic Kleiner und Kalina Bagreva vom IT-Departament für das Einrichten des Timely-Teams-Account und des Timely-Teams. Ein grosser Dank gilt auch allen Schülerinnen und Schülern der KBW, die meine Umfrage ausgefüllt haben und sowie meinen Kolleginnen und Kollegen für ihr wertvolles Feedback während der Entwicklung von Timely.

9. Abbildungsverzeichnis

Quellen: Abb. 1 - 33: Eigene Darstellung

Abb. 1: Umfrageergebnis Frage 1	2
Abb. 2: Umfrageergebnis Frage 2	2
Abb. 3: Umfrageergebnis Frage 3	4
Abb. 5: Umfrageergebnis Frage 5	4
Abb. 7: Umfrageergebnis Frage 7	4
Abb. 4: Umfrageergebnis Frage 4	4
Abb. 6: Umfrageergebnis Frage 6	4
Abb. 8: Umfrageergebnis Frage 8	4
Abb. 9: Umfrageergebnis Frage 9	6
Abb. 10: Umfrageergebnis Frage 10	6
Abb. 11: Timely Logo	8
Abb. 12: Timely App-Icon	8
Abb. 13: Timely Farbpalette und Schriftart	8
Abb. 14: Beispiel einer Raumänderungsnachricht	10
Abb. 15: Beispiel eines automatischen Verschiebungsvorschlages	10
Abb. 16: Beispiel einer automatisch generierten Verschiebungsanfrage	10
Abb. 17: Startbildschirm Timely Webapp	12
Abb. 18: Raumverfügbarkeitsanzeige Timely Webapp	12
Abb. 19: Dropdown-Menü	12
Abb. 20: Darstellung der Räume und Raumstundenplan Timely Webapp	14
Abb. 21: Abweichungen vom Grundstundenplan Timely Webapp	14
Abb. 22: Dropdown-Menü	14
Abb. 23: Timely Status Website	16
Abb. 24: Beispiel eines Stundenplans im Format JSON	16
Abb. 25: Erste Version des Flowcharts für die automatischen Verschiebungsvorschläge	22
Abb. 26: Flowchart des Programms für die automatischen Verschiebungsvorschläge	24
Abb. 27: Flowchart des Timely-API-Servers	28
Abb. 28: Ansicht des Fensters zur Lektionsmodifikation	30
Abb. 29: Datumsauswahl Dropdown-Menü	32
Abb. 30: Raumverfügbarkeitsanzeige im Detail	34
Abb. 31: Klassenauswahl Dropdown-Menü	34
Abb. 32: Beispiel für eine Github Copilot Anfrage um Code zu generieren	36
Abb. 33: Beispiel einer Github Copilot Anfrage um Code zu modifizieren/optimieren	36

10. Quellenverzeichnis

Apple Inc.; «IN2 Mobile». Zugriffen 27. Nov 2024, <https://apps.apple.com/ch/app/in2-mobile/id1560963697>.

Artificial Analysis; «Quality vs. Price». Zugriffen 27. Nov 2024, <https://artificialanalysis.ai/#:~:text=support%20each%20model.,Quality%20vs.%20Price,-15%20of%2080>.

Cloudflare Inc.; «Das globale Cloudflare-Netzwerk». Zugriffen 27. Nov 2024, <https://www.cloudflare.com/de-de/network/>.

Google Cloud; «So verwendet Gemini für Google Cloud Ihre Daten». Zugriffen 27. Nov 2024, <https://cloud.google.com/gemini/docs/discover/data-governance?hl=de>.

JSON.org; «Einführung in JSON». Zugriffen 27. Nov 2024, <https://www.json.org/json-de.html>.

Meltzer, Rachel; «What is a Programming Library?». Januar 2023. Zugriffen 27. Nov 2024, <https://careerfoundry.com/en/blog/web-development/programming-library-guide/>.

Red Hat Inc.; «Was ist eine API?». Zugriffen 27. Nov 2024, <https://www.redhat.com/de/topics/api/what-are-application-programming-interfaces>.

Sana Commerce; «PWA (Progressive Web App)». Zugriffen 27. Nov 2024, <https://www.sana-commerce.com/de/ecommerce-erklaert/was-ist-pwa/>.

UX-Republic; «Async/Await anhand von Diagrammen und Beispielen erklärt». Januar 2018. Zugriffen 27. Nov 2024, <https://www.ux-republic.com/de/async-await-wird-anhand-von-Diagrammen-und-Beispielen-erkl%C3%A4rt/>.