



Paul Maier 15 Jahre
Neufraunhofen Winziger Straße 2 / Gymnasium Dorfen
Nadine Scheler und Christiane Drexler
Digitalisierung der Projektstage am Gymnasium Dorfen
Arbeitswelt
Jugend forscht
Bayern
2021

1 Kurzfassung

In meinem Projekt **diribit.io** geht es darum, die Organisation der Projektstage an unserer Schule durch eine Software so zu unterstützen, dass in erster Linie der administrative Arbeits- und Zeitaufwand gesenkt werden kann. Die Software soll vor allem die Zuordnung der Schüler zu den Projekten erleichtern, da diese nach wie vor manuell durch unsere Schulsozialarbeiterin Frau Vogler erfolgt. Weiter soll sie aber auch Probleme, wie eine unzureichende Kommunikation lösen.

Für einen Systementwurf untersuchte ich deshalb zuerst, welche Hürden im Ablauf auf organisatorischer- als auch auf der Schülerseite zu überwinden sind und welche Anforderungen an die Software für die Durchführung der Projektstage gestellt werden. Basierend auf den Ergebnissen erarbeitete ich zwei Lösungsansätze und wog anschließend ab, welchen ich umsetzen sollte.

Eine Umsetzung in Form einer Webapplikation erschien mir als die beste Herangehensweise, da sie alle Anforderungen erfüllt und kostengünstig und mit geringem Aufwand betrieben werden kann. Ein persönlicher Zugang für alle Beteiligten erlaubt dabei die Umsetzung eines Self-Service-Portals und damit eine gemeinsame Plattform für alle. So können zukünftig nicht nur lange Kommunikationswege gekürzt, sondern auch der manuelle Sortieraufwand vermieden werden.

Nach der Entwicklung der Webapplikation untersuchte ich, wie gut sie die Abläufe an den Projekttagen automatisieren kann und wertete mit Hilfe eines Fragebogens aus, ob die Webseite benutzerfreundlich und intuitiv bedienbar ist.

Inhaltsverzeichnis

1 Kurzfassung.....	1
2 Einleitung.....	3
3 Vorgehensweise, Materialien und Methode	3
3.1 Die Projektstage	3
3.2 Problemanalyse	3
3.3 Lösungsansätze.....	4
3.4 Systementwurf	4
3.4.1 Eine Webapplikation	5
3.4.2 Accountbasierter Zugriff	5
3.4.3 Synchronisation der Aktionen	5
3.4.4 Chat-Funktion.....	5
3.4 Algorithmus	5
3.4.1 Vorgehen.....	5
3.4.2 Sortieralgorithmus	6
3.5 Frontend	6
3.5.1 Angular und Ionic	6
3.5.2 Gestaltung.....	7
3.5.3 Dokumentation	7
3.6 Backend	7
3.6.1 Der JSON Testserver.....	7
3.6.2 Eine API Management Plattform.....	7
3.6.3 API der Marke Eigenbau.....	8
3.6.4 Das Laravel Framework.....	8
3.6.5 Portierung des Algorithmus nach PHP	8
3.6.6 Implementierung in der API	9
3.7 Migration auf einen öffentlichen Server	9
4 Ergebnis	9
4.1 Die Architektur	9
4.2 Die Funktionalität	10
4.3 Die Benutzeroberfläche.....	11
5 Ergebnisdiskussion	11
5.1 Theoretische Bewertung	12
5.2 Nutzererlebnis	13
5.3 Stabilität bei großen Datenmengen	15
5.4 Kontinuierliche Weiterentwicklung.....	16
6 Zusammenfassung	16
7 Quellen- und Literaturverzeichnis.....	17
8 Unterstützungsleistungen.....	17

2 Einleitung

Mein Projekt war zu Beginn nur eine spontane Idee, um die lästige Zuordnung von Schülern zu Projekten zu erleichtern. Ich war in der Nachmittagsbetreuung, als unsere Schulsozialpädagogin Frau Vogler den Raum betrat und begann, die Anmeldungen für die Projektstage in Briefumschläge zu sortieren. Sie war gerade dabei die Projektstage vorzubereiten, da merkte ich Spaßhaft an, dass ein Algorithmus eine solche Aufgabe schneller und zuverlässiger bewältigen könnte. Dem stimmte sie zu und konterte, ich solle dann doch einen programmieren und ihr die Arbeit abnehmen.

Mein Ehrgeiz war geweckt und ich stellte mir die Frage, wie ich die Organisation der Projektstage durch eine Software vereinfachen könnte, um diese für alle Beteiligten einfacher und bequemer zu gestalten.

3 Vorgehensweise, Materialien und Methode

3.1 Die Projektstage

Zuerst ist es für die Vorstellung meiner Arbeit nötig, die organisatorischen Abläufe der Projektstage meiner Schule zu erläutern. Diese finden jährlich an zwei Tagen kurz vor den Sommerferien statt. Die Projekte können von Lehrern, Eltern, weiteren Erwachsenen oder anderen Schülern angeboten werden. Damit es allerdings überhaupt soweit kommt, muss einiges an Vorbereitung im Hintergrund passieren. Als erstes muss ein Projektleiter sein Projekt anmelden, dies geschieht über ein Papierformular zur Projektregistrierung. Wenn der Registrierungszeitraum abgelaufen ist, müssen alle Projektvorschläge gelesen und überprüft werden.

Danach werden die Formulare in der Aula ausgehängt, so dass sie in den Pausen von den Schülern begutachtet werden können. Die eigentliche Anmeldung für ein Projekt erfolgt anschließend über ein weiteres Papierformular, auf welchem ein Schüler bis zu drei Wunschprojekte, an denen er gerne teilnehmen würde, eintragen kann.

Nun kommt für die Organisatoren der aufwändigste und zeitintensivste Teil: Die Schüler den Projekten zuzuordnen. Hierfür beschriftet unsere Schulsozialarbeiterin, welche die Projektstage maßgeblich organisiert, Briefumschläge, in die sie die Anmeldezettel der Schüler einsortiert. Nach langem manuellen Sortieren kommt jeder Schüler schließlich in eines seiner Wunschprojekte. Das Ergebnis hängt dann in Form von Teilnehmerlisten der einzelnen Projekte wieder in der Aula aus. In der Regel wollen aber einiger Schüler nachträglich noch Projekte tauschen. Hierfür gibt es nur die Möglichkeit mit einem Tauschpartner aus einem anderen Projekt persönlich zur Schulsozialarbeiterin zu gehen, um die Teilnehmerlisten anpassen zu lassen.

Jetzt stehen alle Teilnehmer fest, so dass die Projektleiter informiert werden können. Diese begleiten die Schüler durch die zwei Tage Projektdurchführung.

3.2 Problemanalyse

Um die Projektstage zu optimieren, musste ich zunächst die Probleme in den Abläufen ausmachen. Dabei standen Zeit- und Arbeitsaufwand und die Bequemlichkeit der Beteiligten im Fokus, dennoch betrachtete ich auch weitere Aspekte, wie zum Beispiel Umweltfreundlichkeit, sodass ich auf folgende maßgebliche Punkte kam:

- Das größte Problem ist der Zeitaufwand für die Organisatoren. Besonders unsere Schulsozialarbeiterin Frau Vogler muss jährlich viel Zeit in die Projektstage investieren. Dies ist vor allem der Tatsache geschuldet, dass die Projektstage sehr papierlastig sind. Sowohl die Registrierung der Projekte, als auch die Schüleranmeldungen erfolgen ausschließlich über Papierformulare.
- Ein weiterer negativer Effekt der Papierformulare ist natürlich, dass sehr viel Papier benötigt wird. Obwohl auf jedem Blatt zwei Anmeldungen Platz finden enden jährlich nach den Projekttagen umgerechnet ca. 400 DNA-4 Blätter nach einmaliger Benutzung im Müll!
- Ein weiteres sehr umfassendes Problem ist die Kommunikation, sie ist umständlich, langsam und nicht immer zuverlässig. Ein gutes Beispiel ist der Nachrichtenaustausch zwischen dem Projektleiter und seinen Teilnehmern. Falls dieser den Schülern noch etwas mitteilen, oder sie an etwas erinnern will, erreicht seine Nachricht oft nicht rechtzeitig die Empfänger. So haben die Projektleiter oft keine

andere Alternative, als die Schüler über eine Durchsage in einer der Pausen in einen Raum zu bitten und dort eine Besprechung abzuhalten. Das macht es wiederum für Eltern oder andere außenstehende Projektleiter noch schwieriger zu kommunizieren. Auch für die abwesenden Schüler ist diese Art der Kommunikation problematisch, da sie die Informationen nur über Freunde aus ihrem Projekt mitbekommen können.

- Zusätzlich zu dem bereits benannten Problem der Kommunikation zählt der erschwerende Umstand, dass das Tauschen von Projekten untereinander nur bedingt stattfinden kann. So bestehen beispielsweise keine offiziellen Formulare mit denen ein Projekttausch unter Schülern ermöglicht werden könnte. Ebenso gibt es keine offizielle Möglichkeit zur Angabe von Freunden, mit denen Schüler bevorzugt in einem Projekt teilnehmen möchten. Insofern müssen derartige Anliegen direkt bei den Organisatoren geklärt werden. Dies verstärkt das Problem der Kommunikation zusätzlich, da jetzt auch Änderungen bei den Teilnehmern mündlich kommuniziert werden müssen. Zusätzlich generiert es weitere Arbeit für die Schulsozialarbeiterin und verkompliziert die Leitung und Koordination eines Projektes.
- Das letzte große Problem ist, dass die Aushänge in der Aula schwer einsehbar sind, da sich dort fast immer eine große Traube an Schülern bildet, die die Sicht versperrt. Möchte ein Schüler also in Ruhe die zahlreichen Projekte begutachten, muss dieser entweder zu einem anderen Zeitpunkt wiederkehren oder sich dem Gedränge anschließen. Zusätzlich ist es teilweise schwierig die Handschrift auf manchen Zetteln überhaupt entziffern zu können.

3.3 Lösungsansätze

Aus der Menge der oben genannten Probleme habe ich mir phasenweise mehrere Lösungsansätze überlegt. Dabei war mein erster Gedanke, wie in der Einleitung beschrieben, nur einen Algorithmus für die Zuordnung von Schülern zu Projekten zu entwerfen. Dieser würde das Problem des hohen Zeitaufwands lösen, doch als einziges digitales Element der Organisation, hätte er einen Medienbruch zwischen Computer und Papierformular zur Folge. Es müssten also aufwendig die auf Papier gesammelten Daten für den Algorithmus digitalisiert und anschließend wieder auf Papier ausgedruckt werden. Für die Digitalisierung der auf den Formularen enthaltenen Informationen könnten entweder die Daten von den Zetteln abgeschrieben werden, was allerdings den Geschwindigkeitsvorteil der Sortierung durch einen Algorithmus zunichtemachen würde. Jedoch wäre auch beim Einscannen von den ca. 700 Anmeldungen immer noch ein signifikanter Arbeitsaufwand gefordert. Zusätzlich ergäbe sich das Problem, Daten von einem handschriftlich ausgefüllten Formular zuverlässig und fehlerfrei einzulesen und weiterzuverarbeiten. Hier ist vor allem die Vielfalt der Schriftfarben, -größen und -arten ein Problem, welche das Unterfangen in Anbetracht des gegebenen Zeitrahmens nahezu unmöglich macht. Abgesehen von der schwierigen Umsetzung muss auch das Ergebnis als fragwürdig betrachtet werden, da ein Großteil der oben genannten Probleme unberührt blieben. Es widerspricht außerdem in gewisser Weise dem Gedanken der Digitalisierung, da nur einer der Arbeitsschritte digitalisiert wird, die anderen aber immer noch papierbasiert blieben.

Ich kam dahingehend zu dem Schluss, dass es nur sinnvoll ist, das gesamte System zu digitalisieren. Ein solches System beschleunigt nicht nur den Sortierprozess durch einen Algorithmus, sondern hat auch das Potenzial, die sich daran anschließenden Probleme, wie den hohen Papierverbrauch, die schlechte Kommunikation, etc. zu lösen. Diese Herangehensweise ist zwar mit erheblich mehr Entwicklungsaufwand verbunden, hat aber den Vorteil, tatsächlich Anwendung finden zu können und sich auch langfristig als papierloses System zu etablieren.

3.4 Systementwurf

Um einen Plan für das System zu entwerfen, begann ich zuerst damit, mir die Anforderungen und genauen Ziele des Systems zu überlegen. Auf Basis dessen könnte ich entscheiden, in welcher Art und Weise ich das System schließlich umsetzen würde.

3.4.1 Eine Webapplikation

Ein erster wichtiger Punkt ist hier die Zugänglichkeit. Mein System soll möglichst einfach von jedem Standort aus abrufbar sein. So können auch abwesende Schüler und Außenstehende einfach darauf zugreifen und sich beteiligen. Der beste Weg dies umzusetzen erscheint mir mit einer Webapplikation. Sie ist von überall erreichbar und zudem plattformunabhängig. Vorteile der Unabhängigkeit sind, dass ich dadurch nicht verschiedene Versionen für unterschiedliche Betriebssysteme entwickeln muss, was in Konsequenz die Entwicklung und Wartung vereinfachen wird.

3.4.2 Accountbasierter Zugriff

Bei dem Aufbau des Systems entschied ich mich für ein Account-System. Jeder Schüler hat also einen persönlichen Account, auf den er jederzeit zugreifen kann und mit dem er alle seine Aufgaben wahrnimmt. Auch Administratoren und Projektleiter sollen jeweils einen Account haben, wodurch es insgesamt drei Accountrollen gibt. Für den Registrierungsvorgang entschied ich mich bei den Administratoren und Projektleitern für eine Anmeldung per E-Mail. Für Schüler konnte ich so etwas allerdings nicht voraussetzen, da die Schüler gerade in den niedrigeren Klassen noch nicht alt genug sind, um die AGBs für beispielsweise einen Gmail-Account zu erfüllen. Ich überlegte mir diverse Systeme für die Anmeldung der Schüler, aber in allen müssten entweder Anmeldedaten oder anderweitige Tokens verteilt werden. Doch dann kam die Coronapandemie und mit ihr das Homeschooling. Dieses brachte schließlich Microsoft Teams an die Schulen und damit eindeutige E-Mail-Adressen für jeden Schüler. Damit wurde schließlich eine Registration der Schüler über ihre E-Mail möglich, bei der ich sogar anhand der E-Mail-Adresse, welche hinter dem @-Zeichen in der Regel den Schulnamen beinhaltet, überprüfen könnte, ob derjenige Schüler überhaupt berechtigt ist sich anzumelden.

3.4.3 Synchronisation der Aktionen

Für einen reibungslosen und schnellen Ablauf der Projektstage ist eine Art der Koordinierung unerlässlich. Ich entschied mich hier für eine chronologische Ablauforganisation, die jederzeit vom Admin modifiziert werden kann. Diese weist eine ähnliche Struktur wie die der ursprünglichen Projektstage auf. Das erleichtert zum einen den Umstieg auf das digitale System, da ich die Änderungen zum vorherigen System so relativ klein halten könnte. Der Zeitplan gestattet dem Admin zum anderen aber auch alles so flexibel zu verwalten, dass selbst bei unvorhergesehenen Verzögerungen der Ablauf der Projektstage nicht in Gefahr gerät.

3.4.4 Chat-Funktion

Für eine gute Kommunikation zwischen den Projektleitern und den Teilnehmern ist es unerlässlich, eine Art Chat-Funktion in das System zu integrieren. Dabei entschied ich mich, die Möglichkeit Nachrichten zu senden auf die Projektleiter und die Schüler die Projekte leiten zu begrenzen. Dies ist insofern relevant, um den Chatverlauf übersichtlich und unter Kontrolle zu halten und dadurch eine möglichst direkte und eindeutige Kommunikation zu ermöglichen. Die Chatnachrichten sind außerdem auch für die Admins sichtbar, sodass diese bei Fehlverhalten eingreifen können.

3.4 Algorithmus

Da nun der Projektinhalt nicht mehr nur den Algorithmus, sondern das ganze System umfasst, kann ich infolgedessen nicht mehr so viel Zeit für ihn aufwenden wie ich ursprünglich geplant hatte. Deswegen blieb es bei einem eher simplen Algorithmus, der seine Aufgabe zwar zuverlässig, aber nicht unbedingt auf die effizienteste Weise löst.

3.4.1 Vorgehen

Mein Plan für den Algorithmus war, einen Prototypen in [Python](#) zu programmieren und ihn dann später in eine andere Sprache zu übersetzen. Die spätere Portierung war zwar um einiges schwieriger als ich erwartet hatte, doch so konnte ich in der Nachmittagsbetreuung an dem Algorithmus arbeiten und somit viel wertvolle Zeit für mein Projekt gewinnen.

3.4.2 Sortieralgorithmus

Bei der Funktionsweise des Algorithmus stand nicht unbedingt die Effizienz, sondern eine zuverlässige und einfache Art der Sortierung im Vordergrund. Also orientierte ich mich beim Entwerfen des Algorithmus sehr stark daran, wie ich das Problem per Hand lösen würde. Der erste Schritt ist, dass jeder Schüler zuerst in seinen Erstwunsch, also in das Projekt einsortiert wird, das er sich am meisten wünscht. Danach werden in weiteren Durchläufen für eine gleichmäßige Verteilung der Schüler über alle Projekte Teilnehmer aus ausreichend gefüllten Projekten in zu leere und Schüler aus überfüllten in ausreichend gefüllte Projekte verschoben. Am Ende jedes Durchgangs muss dann überprüft werden, ob alle Projekte entweder eine Teilnehmerzahl zwischen dem vom Projektleiter gesetzten Mini- und Maximum haben oder gar keine Teilnehmer mehr haben. In diesem Fall muss das Projekt aussortiert und falls es einen Schüler als Projektleiter hatte, dieser wieder in die Liste der zu sortierenden Schüler eingefügt werden. Das muss solange wiederholt werden, bis schließlich alle Schüler einsortiert sind.

Dieser einfache Ansatz funktionierte zwar als anfänglicher Prototyp, hatte aber immer noch einige Schwächen. Zum einen hing sich der Algorithmus manchmal in einer Endlos-Schleife auf, was einen Absturz des Programms zur Folge hatte. Zum anderen bestand ein weiteres Problem darin, dass der Algorithmus viel zu spät erkannte, wann ein Projekt schlicht nicht stattfinden konnte, weil es insgesamt nicht genug Teilnehmer mit diesem Projekt als Wunsch gab. Und zu guter Letzt gab es immer einen Fehler, sollten alle Projekte eines Schülers nicht stattfinden.

Das erste Problem war schnell durch eine Begrenzung der Durchläufe gelöst. Um das zweite Problem zu lösen, entfernte ich die Projekte, die nicht stattfinden konnten, schon vor der eigentlichen Sortierung, was auch dieses Problem erfolgreich eliminierte. Für das letzte Problem beschloss ich eine Liste einzuführen, welche die Schüler aufführt, die nicht zugeordnet werden konnten.

3.5 Frontend

Für das Frontend war es mir wichtig, Zuverlässigkeit und Sicherheit zu vereinen ohne dabei auf eine optisch ansprechende und einfach zu bedienende Benutzeroberfläche verzichten zu wollen. Diese sollte vor allem plattformübergreifend sein, um die Vorzüge einer plattformunabhängigen Webapplikation nutzen zu können. Letzteres würde vor allem die Nutzung vereinfachen, da sowohl Handys als auch Tablets unter Schülern mit Abstand am weitesten verbreitet und für diese auch praktischer im Umgang sind, als stationäre Computer.

3.5.1 Angular und Ionic

Bei der Suche nach einem Framework stieß ich auf [Angular](#), ein TypeScript-basiertes Webapplikationsframework, welches maßgeblich durch die Google LLC gefördert eine passende Open-Source-Lösung für mein Projekt darstellte, und mit dessen Hilfe ich schnell und einfach ein leistungsfähiges performantes Frontend entwickeln konnte.

Mich auf Angular zu beschränken, hätte immer noch eine paar zeitaufwändige Hürden, insbesondere die plattformübergreifende Gestaltung, die für das Projekt essenziell sein sollte, zu überwinden bedeutet. Daher entschied ich mich, Angular in Kombination mit [Ionic](#), einem weiteren Open-Source-Webframework einzusetzen. Ionic vereinfachte die Gestaltung der Seiten mit vorgefertigten Komponenten und Styles und verkürzte so insgesamt auch die Entwicklungsdauer drastisch.

Um die Liste verwendeter Programme zu vervollständigen, müssen allerdings noch [Node.js](#) und [Npm](#) Packet Manager sowie [Visual Studio Code](#) genannt werden. Dabei sind die ersten beiden Voraussetzungen für sowohl Angular wie auch Ionic, nahmen aber auch noch weitere Nebenrollen, wie zum Beispiel die des ersten Testservers ein. Visual Studio Code ist hingegen der Code-Editor, den ich für die Dauer des Projektes verwendete. Als Browser kam [Mozilla Firefox](#) zum Einsatz, mit dem ich das Frontend testete und debuggte. Im späteren Verlauf des Projektes kam außerdem noch die CSS-Bibliothek [Animate.css](#) hinzu, welche ich zum Animieren der HTML-Komponenten nutzte.

3.5.2 Gestaltung

Bei der Gestaltung der Seiten versuchte ich stets nur die nötigsten Inputfelder anzuzeigen. Um diesem Anspruch gerecht zu werden, ist die Seite zunächst in vier Hauptbereiche: Home, Anmeldung, Hilfe und Einstellungen unterteilt. Nach einer erfolgreichen Anmeldung gelangt der Nutzer zum Dashboard. Dort werden Funktionen, abhängig von seiner Rolle, angezeigt, die für den Nutzer derzeit anstehen. Das hat zwar einerseits den Nachteil, dass dieser keinen umfassenden Überblick über seine Funktionen bekommt, andererseits ist dies aber auch nicht nötig, da ein das System so in gewisser Weise durch die nötigen Schritte leitet. Damit verhält sich die Oberfläche ähnlich, wie ein Tutorial aus einem Computerspiel, bei dem alles außer der kennenzulernenden Aktion ausgegraut ist. Der Vorteil ist, dass der Nutzer nur machen kann, was für ihn vorgesehen ist. Dadurch kann jeder Schüler auch ohne Hintergrundwissen und aufwendiger Orientierung auf der Webseite die gewünschte Aktion durchführen.

3.5.3 Dokumentation

Ursprünglich hatte ich geplant mein Frontend nur durch Kommentare im Code zu dokumentieren, dabei stieß ich allerdings zufälligerweise auf [Compodoc](#), ein Dokumentations-Tool für Angular. Schon nach kurzer Zeit wurde mir klar, dass dieses Tool nicht nur übersichtlich, schnell und praktisch ist, sondern auch perfekt mit der Idee der Kommentierung im Sourcecode harmoniert, da es die Kommentare als Beschreibungen der Attribute und Funktionen einbindet. Ich war in der Lage, das Tool schnell einzubinden und hatte so schon nach kurzer Zeit eine übersichtliche Dokumentation meines Codes. Nun galt es lediglich die weiteren Kommentare zu schreiben, die diese Dokumentation schließlich erst ausführlich machen würden. Die noch unvollständige Dokumentation ist derzeit unter <http://192.248.186.231:8080/> verfügbar und enthält bereits die allgemeine Beschreibung, sowie eine Dokumentierung der einzelnen Module.

3.6 Backend

Mein Backend nahm im Verlauf der Entwicklung verschiedenste Formen an, und entwickelte sich von einem JSON-Testserver über ein vollständig selbst geschriebenes PHP-Backend zu einer sichereren und effizienteren Laravel API. Ich werde hier auch auf meine Fehlschläge eingehen, da ich aus ihnen am meisten gelernt habe. Das war für mein Projekt insgesamt sehr gewinnbringend und gab mir die Möglichkeit, unterschiedliche Perspektiven auf ein und dasselbe Problem einzunehmen.

3.6.1 Der JSON Testserver

Mein Backend startete als JSON Testserver. Dieser bot zu Beginn einige Vorteile, einer davon war, dass er leicht zu erstellen und zu starten war. Ein weiterer Vorteil war, dass ich die JSON-Struktur sehr einfach verändern konnte, was in den ersten Schritten der Entwicklung auch oft genug passierte. Außerdem sind die Zugriffe auf den Inhalt nur „simuliert“, so dass die Nutzdaten vor Änderungen geschützt waren. Ich brauchte also nicht den Datenbestand nach Tests jedes Mal wieder in den Ursprungszustand zu versetzen.

3.6.2 Eine API Management Plattform

Der JSON-Testserver hatte schon nach kurzer ausgedient. In der Applikation ist ein Login-Bereich vorgesehen, wo nach einer erfolgreichen Anmeldung unterschiedliche benutzerabhängige Funktionen zur Verfügung stehen sollen, der Testserver konnte diese Funktion allerdings nicht bieten. Ich versuchte mich deshalb sowohl an der API Management Plattform Fusio, wie auch DreamFactory, konnte aber beide aus verschiedenen Gründen nicht für meine Applikation einsetzen. Bei DreamFactory war es hauptsächlich eine zu stark eingeschränkte kostenlose Version in Verbindung mit einem astronomischen Kostenaufwand von mehreren hundert Euro für die erweiterte Version, während ich bei Fusio durch die schlechte Dokumentation schon an der Installation scheiterte.

3.6.3 API der Marke Eigenbau

Ich beschloss schlussendlich, ein eigenes Backend zu entwickeln. Es kostete zwar einiges an Zeit, hatte bei seiner Fertigstellung aber auch bereits einen Großteil der geplanten Funktionalität. Darunter zum einen die grundlegenden Dinge wie das Accountmanagement, Registrierung und Projekterstellung, zum anderen aber ebenso etwas komplexere Features wie den Zeitplan und den Sortieralgorithmus. Hier möchte ich vor allem kurz auf die Authentifizierung eingehen, da ich diese später für die Realisierung des Projekts weiterverwende.

Die Authentifizierung verwendet sogenannte [JWTs](#) (JSON Web Tokens), das sind in JSON formatierte gehashte Access-Tokens in Form eines Strings, über die die Authentifizierung abläuft. Dafür erstellt der Server bei richtiger Eingabe der Anmeldedaten ein solches Token und sendet es an den Client zurück. Dieser schickt es dann bei allen geschützten Anfragen mit, um sich so zu authentifizieren. Auf dem Server kann dieses Token dann nicht nur schnell überprüft werden, sondern auch Informationen, zum Beispiel über die Art des Accounts, enthalten. Dieses System der Stateless Session ist wesentlich einfacher zu implementieren, als herkömmliche Authentifizierungssysteme und bietet trotzdem einen starken Schutz. Die Vorteile der JWTs überzeugten mich schon für meine erste API, sodass ich das System auch für meine zweite API verwende.

Trotz dieses Fortschritts und seiner Erfolge hatte die API einige schwerwiegende Fehler, die, wie sich im Nachhinein herausstellte, durch frühere Fehlentscheidungen zu Stande gekommen waren. Zum einen war die Verbindung zu der [MySQL](#)-Datenbank nicht ausreichend vor SQL-Injektionen geschützt und auch die Verifizierung der Anfragen war nicht ausreichend umgesetzt. Es gab zudem weitere Probleme, die sich vor allem auf die Gesamtstruktur des Ganzen bezogen, die es aufwendig machten, neue Routen zu implementieren bzw. bestehende zu modifizieren. Schlussendlich wurde mir bewusst, dass dieses Backend seinen Ansprüchen nicht gerecht werden konnte, und ich beschloss, diese Schwächen durch eine komplette Neuimplementierung auszubessern. Bei der Entwicklung und auch aus den gemachten Fehlern habe ich nichtsdestoweniger viel gelernt und möchte die Wichtigkeit dieses Lernprozesses auf keinen Fall abstreiten, denn erst dieses Wissen verhalf mir dann im nächsten Versuch, bessere Entscheidungen zu treffen und diese besser umzusetzen.

3.6.4 Das Laravel Framework

Nun war ich erneut auf der Suche nach einer Möglichkeit mein Backend zu gestalten, doch diesmal suchte ich kein API Management System, sondern ein wirkliches Framework. Vor diesem Hintergrund entschied ich mich für das weitverbreitetste PHP Framework [Laravel](#), da ich bereits meine alte API in PHP geschrieben hatte. Da ich bereits mit PHP vertraut war, hoffte ich teilweise alte Funktionalität recyceln zu können. Ich nahm an, ein gewisser Rahmen, der mir bestimmte Abläufe vereinfachen und sich größtenteils um die Sicherheit kümmern könnte, dürfte für mein Vorhaben genau die richtigen Voraussetzungen bieten, um eine leistungsfähigere und sicherere API zu entwickeln.

Die Entwicklung war auch dank der Vorzüge des Frameworks um ein Vielfaches schneller und einfacher als die gänzlich eigene Umsetzung. Ich konnte mich schnell mit der Funktionsweise arrangieren und machte von den gebotenen Features, wie den automatisierten Datenbankabfragen und der Verwaltung der Routen durch die Middlewares gebrauch. Auch die Implementierung der JWTs war dank einer Bibliothek sehr einfach umzusetzen. Eine Folge dieser einfachen Benutzung ist allerdings, dass man relativ wenig über die tatsächliche Funktionsweise einer API lernt. Insofern kam mir mein vorheriger Blick hinter die Kulissen einer API wieder zugute, da ich ein Verständnis entwickelt hatte, mit dessen Hilfe ich Fehler auch durch den Sourcecode hindurch ausfindig machen konnte.

3.6.5 Portierung des Algorithmus in PHP

Als es dazu kam, den im Voraus entwickelten Algorithmus in das Backend zu portieren, boten sich mir einige Probleme. Das Übersetzen einer Programmiersprache in eine andere stellte sich als weitaus komplizierter heraus als ich anfangs angenommen hatte. Das lag vor allem an kleinen Bedeutungsunterschieden scheinbar ähnlicher Begriffe. Dieses Problem wurde außerdem durch die speziellen Objektstrukturen des Laravel-Frameworks verschärft. Da Klassen und Objekte des Laravel-Frameworks im Vergleich zu PHP anders aufgebaut sind, lassen sich diese nur eingeschränkt in Kombination mit Standard-PHP-Befehlen nutzen. Trotz dieser Schwierigkeiten und nach einigen Verbesserungen war es mir schließlich möglich, den Algorithmus zu übersetzen und seine Funktionalität zu gewährleisten.

3.6.6 Implementierung in der API

Mit der Übersetzung des Algorithmus, war der Algorithmus aber noch nicht vollständig in die Laravel API implementiert. Um ihn vollständig zu integrieren, musste ich ihn noch durch Anfragen zugänglich machen und mir außerdem überlegen, wie ich das von ihm gelieferte Ergebnis auf die Datenbank anwenden könnte. Dabei war mein größtes Problem dem Admin die Möglichkeit zu geben, an dem Sortiervorschlag auch Änderungen vorzunehmen. Zudem hatte ich Sorge, dass die Sortierung und Anwendung in kombinierter Form zu lange dauern würde und somit einen Timeout-Error hervorrufen könnten.

Als Lösung für beide Probleme teilte ich die Sortierung und Berechnung auf. Der geplante Algorithmus würde demnach ein Ergebnis liefern, welches zwischengespeichert werden könnte. Diese zwischengespeicherten Daten können dann vom Admin nachträglich noch bearbeitet werden. Anschließend könnte der Admin diesen Sortiervorschlag dann anwenden und somit die Änderungen auf die Datenbank übertragen. Dank dieses Ansatzes konnte ich das Problem einer zu langen Bearbeitungszeit weitgehendst umgehen und außerdem die Bearbeitung durch Administratoren ermöglichen. Ein Nachteil ist allerdings die umständliche Formatierung und das Auslesen der Daten, welche geringfügig fehleranfällig sind und außerdem ein Sicherheitsrisiko bergen, da der Zugriff auf die Datei nicht zusätzlich abgesichert ist, und somit eine Manipulation bzw. Verfälschung der Daten theoretisch möglich ist.

3.7 Migration auf einen öffentlichen Server

Da mein System nun größtenteils lauffähig ist, folgt der Arbeitsschritt, es auf einen öffentlich zugänglichen Server zu migrieren. Dies würde auch später wichtig werden, um das System vorzuführen bzw. mit mehreren Teilnehmern gleichzeitig testen zu können. Ich mietete mir einen Server bei [VUTR](#), auf dem ich alle notwendigen Pakete mit [Composer](#) installierte und verlagerte mein Projekt dann auf diesen Server. Als Betriebssystem nutze ich das Linux Derivat [Ubuntu](#) und [Nginx](#) als Webserver. Das Projekt wird in einem [git](#)-Repository für das Frontend und einem für das Backend verwaltet. Auf dem Server ist außerdem noch die Dokumentation für das Angular-Ionic Frontend einsehbar.

Leider war es mir noch nicht möglich, die Server-Client-Verbindung zwischen Browser und Webserver mit Hilfe von Zertifikaten zu sichern, da mein Server hierfür über einen Domainnamen erreichbar sein muss, was weitere Kosten generieren würde, welche ich nicht noch zusätzlich zur eigentlichen Miete des Servers tragen wollte. Daher ist auch noch kein Mail Service angeschlossen, der die E-Mails tatsächlich verschickt, sondern nur Mail Trap, ein kostenloser Fake SMTP, der die E-Mails abfängt. Grundsätzlich stellt in einer Testphase keiner dieser beiden Umstände ein Problem dar. Damit das System allerdings produktiv genutzt werden kann, sind diese weiteren Schritte noch notwendig. Die Seite in ihrem aktuellen Zustand ist unter <http://192.248.186.231/> zu finden.

4 Ergebnis

Das Ergebnis von Planung und Entwicklung werde ich hier nun kurz vorstellen. Dabei möchte ich auf die grobe Architektur, die implementierte Funktionalität, aber auch auf die Oberflächengestaltung eingehen. Das Präsentierte ist trotzdem nur für dieses Projekt als Endergebnis anzusehen, auf dessen Basis ich dann die Ergebnisdiskussion führe.

4.1 Die Architektur

Die Architektur der Applikation ist zweiteilig, getrennt in ein Frontend, welches das User-Interface darstellt und einem Backend, welches die Applikationslogik und die Datenspeicherung in einer MySQL-Datenbank übernimmt. Zur Authentifizierung verwende ich JWTs Access-Tokens, die nach einer gültigen Log-In Anfrage vom Client an diesen zurückgegeben werden und mit deren Hilfe sich dieser für alle weiteren Anfragen ausweisen kann.

4.2 Die Funktionalität

Die Funktionalität der App umfasst erst einmal alle Features, die zur Organisierung und Durchführung der Projektstage benötigt werden. Das ist der Mindestmaßstab, damit die Anwendung sinnvoll und produktiv genutzt werden kann. Darunter zum einen das Erstellen, Verwalten und Löschen der drei verschiedenen Accountrollen, das Zurücksetzen des Passwortes, die Erstellung und Verwaltung der Projekte, sowie die Anmeldung für diese Projekte. Darüber hinaus können die Schüler untereinander tauschen, aber auch zusammen ein Projekt leiten. Es ist außerdem für den Admin möglich, jegliche Aktivitäten zu beaufsichtigen und bei Fehlverhalten durchzugreifen. In der folgenden Tabelle wird spaltenweise dargestellt, welche Benutzerrollen mit welchen Funktionalitäten einhergeht.

Schüler	Projektleiter	Admin
Accounterstellung/-verwaltung		
Projekterstellung/ -bearbeitung		Zulassung weiterer Admin Accounts
Nachrichten schreiben/löschen		Zulassung/Löschung der Projekte
Projektassistenten zuweisen	Löschen des eigenen Accounts	Einsehen aller Projekte/Schüler/Projektleiter
Einsehen der Projekte		Löschung der Projektleiter
Anmeldung für die Projekte		Erstellung/Änderung/Anwendung eines Sortiervorschlages
Tauschanfragen stellen/annehmen		Annehmen/Verwerfen einer Tauschanfrage

Die Aktionen werden außerdem durch einen Zeitplan verwaltet, dieser kann durch den Admin festgelegt und nachträglich noch geändert werden. Der Zeitplan orientiert sich maßgeblich an dem Ablauf der Projektstage, enthält aber auch neue Phasen, wie zum Beispiel einen Zeitraum zum Tauschen von Projekten für Schüler. Die folgende Tabelle gibt einen Überblick zum Zeitstrahl:

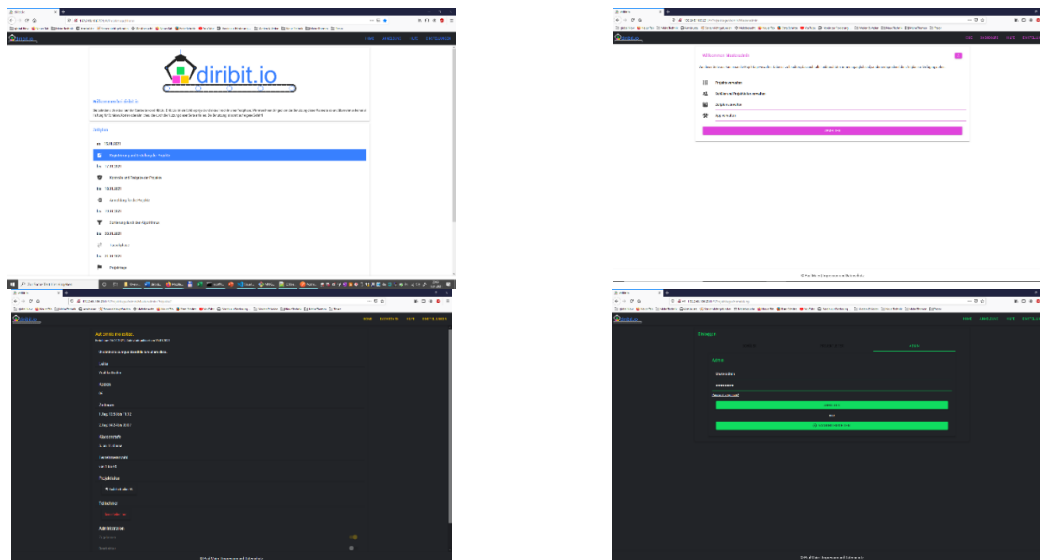
Zeitraum	1	2	3	4	5	6
Schüler	Accounterstellung					
	Projekterstellung					
	Projektbearbeitung					
		Anmeldung für die Projekte				
					Tauschen	
	Nachrichten schreiben					
Projektleiter	Accounterstellung					
	Projekterstellung					
	Projektbearbeitung					
	Nachrichten schreiben					
Admins	Accounterstellung					
		Kontrolle der Projekte				
			Sortierung			
					Bestätigung der Tauschanfragen	
	Freigabe neuer Admin Accounts					

Die App kann außerdem sehr einfach bereitgestellt werden und bietet eine gute Konfigurierbarkeit. So können zum Beispiel Synonyme für Projekte und Projektstage vergeben werden, womit die App flexibel einsetzbar ist und beispielsweise auch für Seminare oder Ausflüge verwendet werden könnte. Leider wird aktuell nur Deutsch als Sprache unterstützt, allerdings ist auch hier mein weiterführendes Ziel, zukünftig eine große Bandbreite an Sprachen zu unterstützen. Dafür werden jedoch noch weitreichende Änderungen im Frontend notwendig sein.

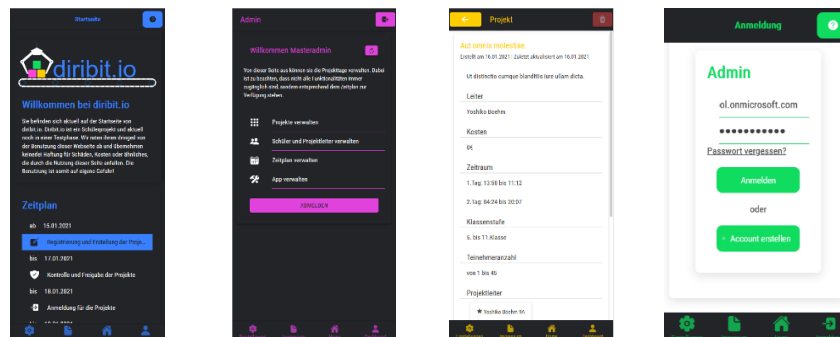
4.3 Die Benutzeroberfläche

Bei der Gestaltung der Benutzeroberfläche achtete ich, wie im Abschnitt Frontend dazu schon beschrieben, zum einen darauf nicht zu viel Funktionalität auf einer Seite zur Verfügung zu stellen, zum anderen bemühte ich mich, mit einer plattformübergreifenden Gestaltung die Nutzung auf allen Endgeräten zu ermöglichen.

In den folgenden Abbildungen sind repräsentativ vier verschiedene Seiten der Desktop Versionen in vier der acht möglichen Designs zu sehen, welche die Applikation unterstützt:



Im Folgenden sind dieselben Seiten, dieses Mal allerdings in der mobilen Version auf verschiedenen durch den Browser simulierten Endgeräten mit geänderter Hintergrundfarbe zu sehen:



5 Ergebnisdiskussion

Um zu bewerten, wie gut die Webapplikation ihren Zweck erfüllt, musste ich zum einen untersuchen, wie viele und wie effektiv die in der Problemstellung genannten Punkte gelöst werden. Zusätzlich muss ich untersuchen, wie intuitiv und schnell die Applikation zu bedienen ist und ob sie auch mit einer großen Anzahl an Benutzern stabil und zuverlässig benutzt werden kann. Das ist sehr wichtig, da sie sie im Einsatz mit Größenordnungen von 700-1500 Benutzern gleichzeitig umgehen können muss. Es sollte aber beachtet werden, dass im Rahmen dieses Projekts nur eine frühe Beta-Version zustande kam, welche immer noch der Weiterentwicklung bedarf. Diese Ergebnisdiskussion ist also für mich die Möglichkeit, meinen bisherigen Fortschritt und die getroffenen Entscheidungen zu überprüfen und offene Punkte wie auch eventuelle Fehler zu ermitteln. Nur so wird es mir möglich sein, eine stabile Version zu entwickeln.

Sehr gerne hätte ich Ergebnisse aus dem praktischen Einsatz, in die Ergebnisdiskussion aufgenommen. Leider sind diese erst kurz vor den Sommerferien und finden somit nach dem Wettbewerb statt. Dennoch plant die

Schule, falls die Situation vor den Sommerferien es zulässt, meine Software einzusetzen. Hierfür habe ich mich auch schon mit den zuständigen Organisatoren in Verbindung gesetzt, denen ich meine volle Unterstützung zugesichert habe.

5.1 Theoretische Bewertung

In der theoretischen Bewertung rekapituliere ich, ob die Funktionalität ausreicht, um die eingangs genannten Probleme zu lösen. Hierbei spielt die tatsächliche Nutzbarkeit bzw. der Aufwand für die Nutzung erstmal noch keine Rolle.

Als größtes Problem hatte ich den Zeitaufwand für die Projektorganisatorin angeführt, der vor allem durch die Menge der Schüler und die manuelle Sortierung begründet ist. Das digitale System hingegen ist komplett papierlos, womit der Zeitaufwand erheblich gesenkt wird, da weder das Drucken noch Sortieren oder Lesen der handschriftlichen Formulare nötig ist. Auch der Zeitaufwand für die Kommunikation kann durch das Schreiben von E-Mails gesenkt werden. Ein Großteil der Zeitersparnis ist aber auch dem Algorithmus und seiner Fähigkeit die Schüler schnell und zuverlässig ihren Projekten zuzuordnen geschuldet.

Die digitale Natur des Systems macht es zudem umweltfreundlicher in Bezug auf den Papierverbrauch. Allerdings wird dieser Vorteil durch die Notwendigkeit eines Servers, welcher selbst die Umwelt belastet, etwas getrübt. Die Belastung kann trotzdem gering gehalten werden, indem der Server, wenn die App nicht benötigt wird, heruntergefahren wird.

Das Problem der Kommunikation zwischen Projektleitern und Teilnehmern hingegen konnte durch die integrierte Chat-Funktion des Systems gut bewältigt werden. So können die Projektleiter über den Chat Erinnerungen oder Planänderungen schnell und zuverlässig kommunizieren, ohne die Teilnehmer persönlich ansprechen zu müssen. Hierdurch können auch Außenstehende effizient mit den Teilnehmern kommunizieren und sogar zeitweise abwesende Schüler bekommen zuverlässig alle Nachrichten.

Die Funktionalität umfasst außerdem Funktionen, die vorher nicht bei der papierbasierten Organisation gegeben waren. So können Schüler Freunde angeben und Tauschanfragen stellen, wobei allerdings die Bestätigung eines Admins nötig ist, um übermäßige Tauschanliegen zu vermeiden. Dies vereinfacht die Aufgabe der Organisatoren, da nun nicht mehr ausführlich und organisatorisch aufwendig mit Schülern und Projektleitern gesprochen werden muss. Stattdessen bekommen diese alle nötigen Informationen über die Plattform.

Das System macht außerdem Aushänge in der Aula überflüssig, da alle Projekte von den Schülern auf der Seite eingesehen werden können. Diese sind dank ihrer digitalen Form sehr einheitlich und immer gut lesbar. Es muss nun also nicht mehr gedrängelt oder früh angestanden werden, um sich die Projektmöglichkeiten in Ruhe durchlesen zu können, was zusammen mit den anderen Vorteilen zu einem insgesamt angenehmeren Erlebnis führt.

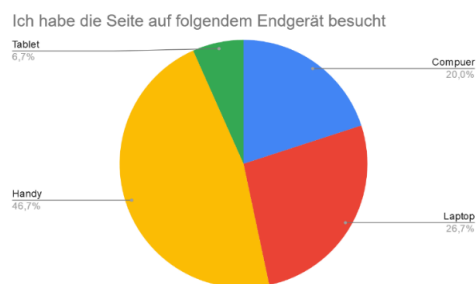
Weiterhin konnte ich die Koordination durch den integrierten Zeitplan effektiv umsetzen. Dieser ist stets gut für alle Teilnehmer sichtbar, sodass alles übersichtlich und verständlich bleibt. Die einzige Limitierung, die dem Zeitplan geschuldet ist, besteht darin, dass eine Phase nicht kürzer sein kann als einen Tag, womit sich ein minimaler Zyklus von fast einer Woche ergibt. Diese minimale Zyklusdauer stellt aber meiner Meinung nach kein wirkliches Problem dar, da bei einem so strikt getakteten Plan nicht das Zeitlimit, sondern schlicht die Nutzer und ihre Zuverlässigkeit die Hürde darstellen. Unter Berücksichtigung dieser Faktoren ist der Zeitrahmen von circa einer Woche schon sehr kurz getaktet, um in der Praxis tatsächlich Anwendung finden zu können.

Schließlich möchte ich noch etwas auf den Algorithmus eingehen. Dieser kam durch den eminenten Zeitaufwand für die Entwicklung des restlichen Systemrahmens leider deutlich zu kurz, da das System für den praktischen Einsatz relevanter war als der reine Algorithmus. Er ist zwar zuverlässig und kann relativ große Datenmengen handhaben, ist aber nicht unbedingt der effizienteste und liefert auch nicht immer die optimalste Lösung. Mein Plan besteht darin, dies in einem weiterführenden Projekt im Bereich der Informatik zu ändern, indem ich ein ganzes Projekt der Entwicklung eines verbesserten Algorithmus widme.

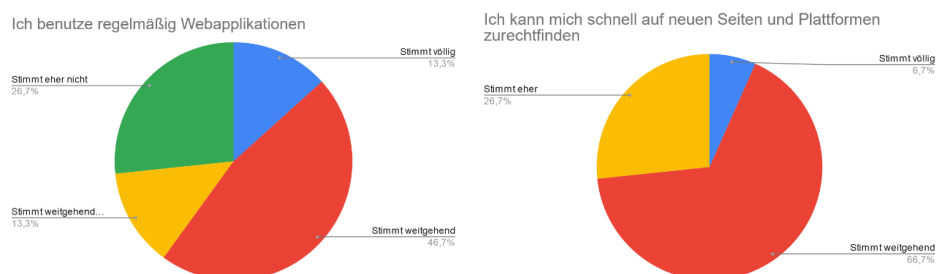
5.2 Nutzererlebnis

Die theoretische Betrachtung der Funktionalität reicht allerdings nicht aus, um ein solches System umfangreich bewerten zu können, da hierfür das Nutzererlebnis und subjektive Empfinden eine große Rolle spielt. Ein gutes Nutzererlebnis ist wichtig, um einen effizienten und reibungslosen Einsatz des Systems garantieren zu können, aber auch um Akzeptanz und Zusprache für dieses neue System zu gewinnen. Für die Bewertung des Nutzererlebnisses beschloss ich, die Projektstage mit einer möglichst großen Gruppe an Freunden und Klassenkameraden durchzuspielen. Anschließend befragte ich sie mithilfe eines Formulars zu ihrer Nutzung.

Für die Erstellung des Befragungsbogens und die anschließende Umwandlung zu Diagrammen verwendete ich [Google Forms](#) und [Google Sheets](#). In dem Formular stellte ich Fragen dazu, wie schnell und intuitiv Aufgaben erledigt werden konnten und wie ansprechend die Seite aufgefasst wurde. Für die Auswertung war mir wichtig, die Erfahrung und somit die Repräsentativität der Nutzergruppe einschätzen zu können, weshalb ich auch Fragen einbaute, die auf die Vertrautheit mit Webapplikationen hinweisen. Das Ergebnis ist in Anbetracht der erhobenen Daten vorerst nur für Schüler repräsentativ, was angesichts der Tatsache, dass diese auch den Großteil der Nutzerschaft ausmachen werden, ein zu vernachlässigendes Problem darstellt. Ein großes Problem hingegen ist die geringe Teilnehmerzahl (15), die die Repräsentativität meiner Umfrage erheblich beeinträchtigt. Dennoch bin ich für das Feedback sehr dankbar, da ich so schon etwas einschätzen kann, wie es um das Nutzererlebnis steht und was ich verbessern kann.

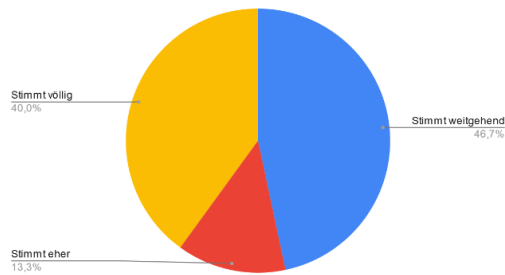


Als erstes möchte ich auf die Verteilung der benutzten Endgeräte eingehen. Im Diagramm ist zu erkennen, dass über die Hälfte der befragten Personen die Seite auf einem mobilen Endgerät wie Handy oder Tablet bedient haben. Dieser Anteil könnte im tatsächlichen Einsatz sogar noch höher liegen, da die Umfrage im Lockdown gemacht wurde, also als die meisten zu Hause waren. In Anbetracht dieser Tatsache ist die Entscheidung und die Bemühungen für eine plattformübergreifende Applikation mehr als gerechtfertigt, weshalb ich in der weiteren Entwicklung verstärkt darauf achten werde, die App plattformübergreifen zu halten. Um die App für Handys und Tablets noch zugänglicher zu machen, könnte ich sie beispielsweise auch als App im Play Store zum Download bereitstellen. Dieses Kompilieren zu einer Applikation wird auch durch Ionic unterstützt, wenngleich ich noch nicht die Zeit fand es zu testen.

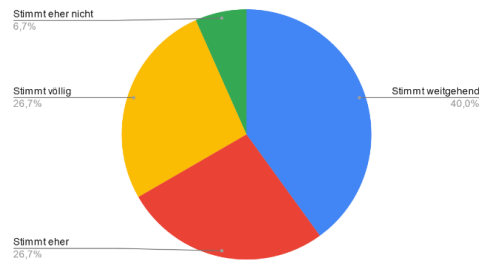


Als nächstes möchte ich die Erfahrung und Eignung der Befragten beurteilen. Dies ist wichtig, um die Ergebnisse der folgenden Fragen besser einschätzen und deuten zu können. Ich habe für diese persönliche Einschätzung zum einen nach der Häufigkeit der Benutzung, und zum anderen nach dem Zurechtfinden auf neuen Plattformen gefragt. Den Antworten zufolge qualifizieren sich die meisten als etwa durchschnittliche tendenziell leicht unterdurchschnittliche Benutzer. Die Beurteilung geht dafür von einer leichten Überschätzung der Teilnehmer aus, was den Punkt des Zurechtfindens auf neuen Plattformen betrifft.

Ich konnte meine Aufgaben schnell erledigen

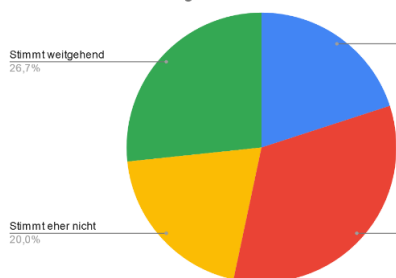


Es fiel mir einfach meine Aufgaben zu erledigen

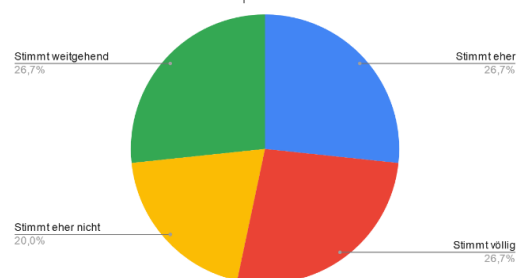


Die nächsten beiden Fragen beziehen sich auf die Erledigung der Aufgaben. Sie lassen vermuten, dass die Benutzung relativ schnell und einfach ist, womit zwei wichtige Punkte für ein gutes Nutzererlebnis abgehakt werden können. Ein wenig zu irritieren scheint dennoch, dass die Einfachheit der Erledigung schlechter bewertet wurde als die Geschwindigkeit mit der diese erledigt wurden. Diese müssten eigentlich einen engeren Zusammenhang zeigen, da man bei Aufgaben, die einem schwerer fallen, mehr Zeit benötigt, um diese zu erledigen. Ein mögliche Erklärung für die Differenzen zwischen den beiden Diagrammen könnte sein, dass die benötigte Zeit allgemein so gering war, dass selbst schwierige Aufgaben verhältnismäßig schnell zu erledigen waren. Eine andere Möglichkeit der Erklärung wäre, dass die Erledigung kurzweilig ist und somit die Wahrnehmung der Zeit etwas verzerrt. In beiden Fällen ist das Ergebnis allerdings positiv, da sowohl wenig Zeitaufwand als auch eine kurzweilige Benutzung der App auf ein gutes Nutzererlebnis deuten lassen. Die Schwierigkeit der Aufgaben ist akzeptabel zu werten, wenngleich es mein Ziel sein wird, diese durch Verbesserungen in der UI weiter zu senken.

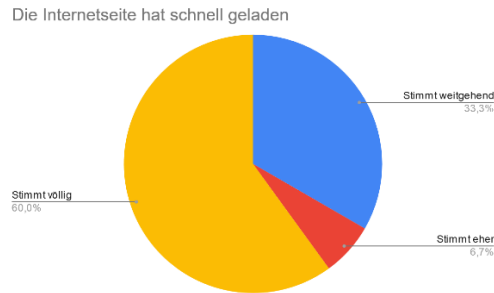
Ich fand die Gestaltung übersichtlich



Ich fand die Internetseite ansprechend



Weiter war mir natürlich wichtig, wie die Gestaltung der Benutzeroberfläche bei den Befragten ankam. Hierbei interessierte mich, wie übersichtlich und ansprechend die Gestaltung aufgefasst wurde. Das Ergebnis ist allerdings alles andere als homogen. Das Feedback kann als viergeteilt zwischen den vier besten der sechs Möglichkeiten betrachtet werden, was zwar zum einen darauf hindeutet, dass die Gestaltung nicht völlig fehlgeschlagen ist, aber anscheinend dennoch nicht ausreicht, um von sich zu überzeugen. Es ist zwar zu erkennen, dass gut drei viertel die Gestaltung mindestens akzeptabel finden, aber dennoch schneidet sie von allen Kategorien am schlechtesten ab. Hinzu kommt, dass auch hier die Bewertung wahrscheinlich aufgrund zwischenmenschlicher Beziehungen leicht positiv verzerrt ist, womit anzunehmen ist, dass die tatsächliche Auffassung des Designs wahrscheinlich noch etwas schlechter einzuordnen ist. Meine Vermutung ist, dass das für eine Webseite eher untypische Design zusammen mit den Neonfarben auf gespaltene Meinungen trifft. Diese anhand der erhobenen Daten ersichtlich gewordene gespaltene Meinung ist natürlich negativ zu werten, da meine Seite auf möglichst viele Menschen ästhetisch ansprechend wirken soll. Mein Fazit hieraus ist, dass ich sowohl das Farbschema, als auch die Übersichtlichkeit ändern bzw. verbessern muss.



Schließlich war für mich die Performance der Seite noch von Interesse. Diese war erfreulicherweise bei den Befragten sehr gut. Allerdings ist diese gute Statistik mit Vorsicht zu genießen, da die Seite relativ klein ist und der Aufgabenbereich für Schüler und Projektleiter nicht unbedingt die umfangreicher aufgebauten Seiten umfasst. Diese aufwändigeren Seiten finden sich ausschließlich bei den Admins und da ich die Testphase selbst administriert habe, liegen keine Umfrageergebnisse dazu vor. Trotzdem ist davon auszugehen, dass Performance kein wirkliches Problem darstellt oder darstellen wird, obwohl ich dennoch versuchen werde, vor allem die leistungsintensiveren Seiten noch zu verbessern. Solche Verbesserungen habe ich aber auch teilweise schon vorgenommen. Ein gutes Beispiel ist die Seite, auf der der Admin die Schüler und Projektleiter einsehen kann. Diese hatte zu Beginn große Schwierigkeiten, da alle Schüler auf einmal geladen wurden, was bei großen Schülerzahlen zu einigen Problemen führte. Die Umstellung, die Schüler nur noch in Paketen mit fester Größe zu laden, löste dieses Problem, ohne die Übersichtlichkeit zu beeinträchtigen.

5.3 Stabilität bei großen Datenmengen

Da für das Testen der Stabilität bei großen Datenmengen eine enorme Gruppe an Freiwilligen nötig gewesen wäre, die ich nicht aufbringen konnte, musste ich auf generierte Datensätze zurückgreifen. Mit ihrer Hilfe konnte ich vor allem den Algorithmus und die Ansichten der Administratoren sowohl auf Richtigkeit, als auch auf Performance überprüfen. Ein Nachteil dieser Methode war allerdings, dass ich mögliche Überlastungen zum Beispiel bei der Anmeldung nicht simulieren konnte, womit ein wichtiger Teil des Belastungschecks leider nicht möglich war.

Für die Tests generierte ich verschiedene Mengen an Schülern, Projektleitern und Projekten. Dabei war es wichtig, ein gleichbleibendes Verhältnis von Schülern zu Projekten beizubehalten, da dieses die Geschwindigkeit des Algorithmus stark beeinflusst. Ich wählte ein Verhältnis von 1 zu 20 und verteilte die Projekte dann gleichmäßig auf Projektleiter der Klasse Schüler und der Klasse Projektleiter. Die Schüler, die an Projekten teilnehmen sollen, wählen dann für jeweils Erst-, Zweit- und Drittwunsch ein zufälliges Projekt. Ich erstellte dann jeweils fünfmal über die Webseite und über das Tool Postman einen Sortiervorschlag und notierte die Zeit, die der Request gebraucht hatte. Dabei kam ich zu folgenden Ergebnissen:

Anzahl Schüler	100	200	500	1000	2500	5000
Anzahl Schüler mit Projekt	2	5	12	25	62	125
Anzahl Projektleiter	3	5	13	25	63	125
Anzahl Projekte gesamt	5	10	25	50	125	250
Gemessene Zeit in Millisekunden (Postman)	154	267	1106	4140	27550	> 30s
	168	275	1331	4660	27410	> 30s
	125	278	1343	4160	29330	> 30s
	127	281	1286	4170	26960	> 30s
	133	286	1185	4400	27720	> 30s
Durchschnitt	141	277	1250	4306	27794	> 30s
Gemessene Zeit in Millisekunden (Browser)	117	259	1179	4360	28620	> 30s
	221	304	1156	4250	28150	> 30s
	132	259	1213	4450	26950	> 30s
	119	253	1192	4420	27650	> 30s
	119	239	1119	4220	27840	> 30s
Durchschnitt	141	263	1172	4340	27842	> 30s

Der Belastungstest zeigt, dass die zum Sortieren benötigte Zeit rasch ansteigt. Während sich beispielsweise die Anzahl der Schüler von 100 auf 500 verfünffacht, ist die benötigte Zeit mehr als das Achtfache der ursprünglich benötigten Zeit. Wir sehen außerdem, dass zwischen der im Browser gemessenen Zeit und der von Postman kaum ein Unterschied besteht und die minimalen Unterschiede, die es gibt, sind durch das Caching in der DNS Auflösung und Messungenauigkeiten im Browser zu erklären.

Außerdem können wir erkennen, dass der Algorithmus 2500 Schüler gerade noch handhaben kann, aber bei 5000 Schülern versagt. Der Fehler, der dabei auftrat war jedoch wider meiner Erwartung keine Zeitüberschreitung der http-Anfrage, sondern eine beim Ausführen des PHP Skripts. Hier gibt PHP ein Limit von 30 Sekunden vor, welches durch die große Anzahl an auszulesenden und zu sortierenden Schülern nicht eingehalten werden kann. In zukünftigen Versionen wird es also notwendig sein, die Daten geteilt zu laden und effizienter zu verarbeiten oder dieses Limit zu erhöhen, wobei ersteres natürlich erstrebenswerter wäre. Weiterhin wäre es ideal, wenn die Sortierung in mehrere Prozesse aufgeteilt werden könnte. Zusätzlich könnte die Sortierung beschleunigt werden, indem die Effizienz des Algorithmus weiter verbessert wird. Dafür wäre allerdings ein komplett neuer Ansatz nötig, welchen ich in diesem Projekt aus zeitlichen Gründen nicht mehr ausarbeiten konnte. Die angesprochenen Punkte werden allerdings in mein nächstes Projekt zur Verbesserung dieses Sortieralgorithmus mit einfließen.

Abschließend würde ich den Algorithmus als ausreichend einstufen, da dieser seine Aufgabe weitgehend zuverlässig und fehlerfrei erledigen kann, wenngleich die Effizienz und Geschwindigkeit nicht hervorragend sind. Trotzdem ist die maximal benötigte Zeit von wenigen Sekunden der manuellen Bearbeitungszeit von mehreren Tagen weit überlegen, weshalb ich den Algorithmus zwar durchaus als Erfolg und Verbesserung zur vorherigen Arbeitsweise verbuchen würde, in Anbetracht der erörterten Punkte jedoch weitere Verbesserungen in einem zukünftigen Projekt anstreben werde.

5.4 Kontinuierliche Weiterentwicklung

Die genaue Untersuchung und Bewertung meiner bisherigen Ergebnisse zeigen, dass die App schon durchaus einsatzfähig ist und ihren Zweck erfüllt. Sie hat aber auch Schwachpunkte enthüllt und mögliche Ansätze zur Verbesserung einiger Feinheiten aufgezeigt. Mein Ansatz ist deshalb eine kontinuierliche Weiterentwicklung der App. Dafür werde ich zum einen nächstes Jahr ein Projekt der Steigerung der Effizienz des Algorithmus widmen, zum anderen werde ich, sobald die App für die Projektstage praktisch angewendet wurde, wieder Feedback sammeln, um dieses dann in die Version für das folgende Jahr einfließen zu lassen. Mit dieser Feedback-Schleife kann ich die App immer auf einem aktuellen Stand halten und sie speziell auf ihre Aufgabe ausrichten.

6 Zusammenfassung

Der Einsatz einer Webapplikationen zur Vereinfachung und Optimierung zeitaufwändiger und unpraktischer Arbeitsabläufe stellte sich als wesentlich schwieriger heraus als ich anfangs angenommen hatte. Sowohl die Planung und langwierige Entwicklung, wie auch das Inbetriebnehmen eines solchen Systems, stellen hier große Hürden für die Digitalisierung dar. Dabei bleibt auch die Frage der Kosten meist ungeklärt, da keine wirklichen Budgets für die Entwicklung und das Betreiben von digitaler Software an Schulen existieren, welche nicht unmittelbar für den Unterricht verwendet wird.

Dabei ist es meiner Meinung nach sehr wichtig, die Digitalisierung voranzutreiben und Systeme zu fördern, die eine umweltfreundliche und einfache Alternative gegenüber dem papierbasierten Systemen darstellen. Diesen Gedanken setzt mein Projekt sowohl konzeptuell als auch anwendungsbezogen schon sehr gut um, da man mit dem von mir entwickelten System die Projektstage sowohl papierlos, effizienter als auch schneller durchführen kann. Somit gibt es trotz einiger Hürden, wie der Entwicklung und Finanzierung, eine Aussicht auf die digitale Zukunft an Schulen. Aus diesem Grund möchte ich mein System, sobald es sich bewährt hat, auch an anderen Schulen etablieren und somit einen weiteren Schritt in Richtung einer ökonomischen und nachhaltigen Zukunft machen, da selbst kleine Schritte der Digitalisierung langfristig einen großen Vorteil bei der Gestaltung einer nachhaltigen Umwelt bieten.

7 Quellen- und Literaturverzeichnis

Im Folgenden sind alle Tools und Bibliotheken aufgelistet, die ich für die technische Umsetzung meiner Webapplikation im Laufe ihrer Entwicklung verwendet habe.

- Angular <https://angular.io/>
- Animate.css <https://animate.style/>
- Compodoc <https://compodoc.app/>
- Ionic <https://ionicframework.com/>
- JWT <https://jwt.io/>
- Laravel <https://laravel.com/>
- Mozilla Firefox <https://www.mozilla.org/de/firefox/>
- MySQL <https://www.mysql.com/de/>
- Node.js <https://nodejs.org/en/>
- Npm <https://www.npmjs.com/>
- PHP <https://www.php.net/>
- Postman <https://www.postman.com/>
- Python <https://www.python.org/>
- Visual Studio Code <https://code.visualstudio.com/>

Die weitere Auflistung ist den bis jetzt ungenannten Betriebssystemen, Services und Tools gewidmet, die ich für den Einsatz oder die Verwaltung der Serverversion genutzt habe.

- Composer <https://getcomposer.org/>
- Git <https://git-scm.com/>
- Nginx <https://www.nginx.com/>
- Ubuntu <https://ubuntu.com/>
- VULTR <https://www.vultr.com/>

Hier sind nun alle Programme und Seiten aufgelistet, die ich für sonstige Tätigkeiten verwendet habe, die mit meinem Projekt bzw. der schriftlichen Arbeit darüber zu tun hatten.

- Google Forms <https://www.google.com/forms/about/>
- Google Sheets <https://www.google.com/sheets/about/>
- Microsoft Word <https://www.microsoft.com/de-de/microsoft-365/word>

8 Unterstützungsleistungen

Zuerst möchte ich natürlich meiner Familie danken, die mein hohes Zeitinvestment für dieses Projekt toleriert und mich auch sonst immer unterstützt hat. Ein besonderer Dank gilt hier meinem Vater, der die nötigen technischen Voraussetzungen schuf, der aber auch immer ein offenes Ohr für meine Probleme und Sorgen hatte.

Mein Dank gilt außerdem den Jugend forscht-Alumni, die mir mit ihrem fantastischen Mentoring Programm dieses Jahr wichtige Impulse für mein Projekt gegeben haben und durch die mein Projekt schließlich in der richtigen Sparte gelandet ist.

Zudem möchte ich den Testlesern meiner schriftlichen Arbeit, die diese vor allem um Tippfehler und unnötige Kommas erleichtert haben, für ihre Zeit und ihr Engagement danken.

Auch bei meiner Schule, dem Gymnasium Dorfen, möchte ich mich für die tolle Kooperation in diesem Projekt danken. Hier ist natürlich vor allem unsere Schulsozialarbeiterin Frau Vogler hervorzuheben, da sie mich nicht nur auf die Idee gebracht, sondern mich zudem durch Lob an meiner Arbeit motiviert hat.

Ein großer Dank geht weiterhin an alle, die mir geholfen haben mein Projekt zu testen, da nur durch ihre intensive Nutzung und dem daraus gewonnenen Feedback eine abschließende Bewertung meiner Arbeit möglich wurde.