

Dokumentation WECollab

WS 21/22

Hochschule: Hochschule der Medien

Modul: 119450a Interdisziplinäres Projekt

Projektbetreuer: Prof. Walter Kriha

Vorgelegt von:

Christoph Amann
Mobile Medien im 7. Semester
Martikelnummer: 36990

Mohamad Fadel Kaadan
Medieninformatik im 7. Semester
Martikelnummer: 37021

Idee

Bei WECollab handelt es sich um eine Collaboration Webanwendung. Die Idee hinter WECollab ist, dass man eine Projektidee hat, aber einem die Mitglieder dazu fehlen. Bereits im Studium ist uns aufgefallen, dass man nach Mitgliedern sucht, weil einem selbst die benötigten Fähigkeiten fehlen. Sei es einen Programmierer für das Entwickeln einer Webanwendung, einen UX/UI Designer für eine Mobile App oder einen Level Designer beim Game Design. Ein weiterer Anwendungsbereich ist außerhalb des Studiums oder dem Beruf. Mit WECollab hat man auch hier die Möglichkeit Mitglieder zu finden, wenn man eine Idee hat. Allerdings benötigt man nicht zwingend eine Idee, um WECollab zu nutzen. Ebenso hat man die Option, ohne eine Idee Teil eines Projekts zu werden. Die Nutzer, welche nach einem Projekt suchen, werden Projekte nach dessen Interessen angezeigt, können dann ein Projekt anschauen und schauen, ob ihnen das Projekt zusagt und es zu ihnen passt. Wenn es ihnen zusagt, können diese sich auf eine der ausgeschriebenen Stellen, welche der Projektersteller erstellt hat, bewerben. Der Projektersteller kann dann entscheiden, ob er die Person in seinem Projekt haben will. Bei einer Annahme wird diese Person dann dem Gruppenchat des Projekts hinzugefügt und die Gruppenmitglieder können miteinander kommunizieren.

Die Vorteile dieser Gruppenarbeit sind, dass man seine kommunikativen und sozialen Kompetenzen verbessert. Des Weiteren werden die Hard Skills verbessert oder neue dazugelernt, was auch für Unternehmen einen Vorteil hat. Ein weiterer Punkt ist, dass man auch Netzwerk, so neue Kontakte außerhalb seines gewohnten Umfelds knüpft und nebenbei ein Produkt komplett neu entwickelt oder es mit weiterentwickelt. Durch die Gruppenarbeit wird man motiviert etwas zu Lernen und weiterzumachen, da es uns persönlich auch schon aufgefallen ist, dass man einer Gruppenarbeit am meisten mitnimmt. Daher hat uns unser Projekt nochmals persönlich motiviert.

Ablauf

Bevor wir mit unserem Projekt beginnen konnten, mussten wir zunächst überlegen, was wir umsetzen. Dementsprechend haben wir mit einem Brainstorming Ideen gesammelt. Anschließend haben wir diskutiert was wir machen wollen und haben überlegt was wir umsetzen können. Dann musste noch geklärt werden, was unser USP (Unique Selling Point) ist, um unser Alleinstellungsmerkmal zu festzulegen. Daher haben wir eine Wettbewerbsanalyse mit anderen Softwareanbietern durchgeführt. Dort haben wir keine vergleichbare Anwendung entdeckt, bei welcher man Mitglieder für ein Projekt sucht und sich organisiert. Um uns neben den weiteren Vorlesungen und Projekten zu organisieren haben wir drei Boards in Jira erstellt. Eines war für die Entwicklung, ein weiteres für das Design und das letzte für das allgemeine Projektmanagement des Projekts. Diesen Boards haben wir dann Aufgaben und Deadlines hinzugefügt, um uns so besser organisiert zu sein und haben darauf geachtet genug Zeitpuffer einzuplanen, sollte etwas eintreten wie Krankheit oder ein anderes unerwartetes Ereignis. Des Weiteren haben wir wöchentliche Termine für Meetings gehabt, um uns abzusprechen. Bei diesen Meetings haben wir den aktuellen Stand gezeigt und haben geklärt was technisch möglich ist, wo man eventuell Abstriche macht und es zunächst einfacher umsetzt. Um uns unsere Zielgruppe besser vorstellen zu können, haben wir Persona erstellt. Somit hatten wir immer im Blick für wen wir die Webanwendung entwickeln. Dann haben wir uns überlegt, welche Navigationspunkte wir haben und wie diese Sinn ergeben. Mit Hilfe von möglichen Nutzern, haben wir dann eine Informationsarchitektur erstellt, um die Navigation der Webseite zu erstellen. Der nächste Schritt war es Skizzen der Webseite zu erstellen, dort haben wir auf Quantität und nicht auf Qualität gesetzt, um so viele mögliche Lösungen zu haben wie möglich. Aus diesen Skizzen haben wir dann gewählt und haben auf Basis dieser, Wireframes in der Anwendung „Figma“ erstellt. Darauffolgend haben wir ein Prototyp erstellt, um den User Flow zu sehen. Schließlich konnte bereits mit der Entwicklung und der Umsetzung begonnen werden, da man wusste was zu implementieren ist. Beim Design konnte dann mit einem Style Guide begonnen werden, damit das Design über die gesamte Webseite einheitlich ist. Nachdem dieser erstellt wurde, wurde der Style Guide auf den Screens angewendet. Dann haben wir das Design nochmals im Prototypen testen lassen und haben dafür einen Fragenbogen erstellt. Auf Basis dieser Ergebnisse, wurden dann weitere Änderungen

vorgenommen. Dann ging die UI in die Frontend Entwicklung. Nachdem das Back- und Frontend implementiert wurde, wurden Tests bezüglich Bugs gemacht und in einer Excelliste gesammelt. Die auftretenden Bugs werden in der Liste notiert und ständig verbessert.

Ziele

Aufgrund fehlender Kapazitäten haben wir für uns bei unserem Interdisziplinären Projekt für einen MVP entschieden, um so das Risiko zu verringern. Bereits beim Brainstorming hatten wir viele Überlegungen, mussten es aber auf das minimalste herunterbrechen und haben bereits dort überlegt, was alles möglich ist, in der vorhandenen Zeit umzusetzen.

Für die Zukunft planen wir an dem Projekt weiterzuarbeiten. Außerdem werden wir weitere Funktionen einbauen, welches aufgrund der fehlenden Zeit nicht funktionierte. Geplante Funktionen sind, dass man noch den Darkmode korrekt umsetzt, da dort noch nicht die korrekten Farben implementiert sind. Außerdem möchten wir ein Kanban Board hinzufügen, damit sich die Gruppen organisieren können.

Eventuell möchten wir auch ein Bewertungssystem für Personen einfügen, damit man auf Basis dieses auch entscheiden kann, ob die Person zu einem Projekt passt und wie die Erfahrungen anderer Gruppen mit der Person sind.

Weitere Überlegungen waren, dass man Events organisiert. Wie z.B. Hackathons, wo dann Gruppen organisiert werden und man dort Preise gewinnen kann. Außerdem möchten wir noch den Chat weiterausbauen und Untergruppen innerhalb der Gruppe haben, welche die Gruppen selbst festlegen können.

Code

Backend

Für unser Backend haben wir uns für Nest.js entschieden, weil im Vergleich zu express.js, ist NestJS ein hoch strukturiertes Framework. Es folgt dem Design-Paradigma "convention over configuration".

In Nest.js sind verschiedene sinnvolle Ansätze schon integriert:

- Der Controller ist mit einem "try-catch" Block umschlossen

- Der Request Body wird bereits geparsed
- Ein ErrorHandler ist integriert
- Middlewares und Logger sind vorhanden

Diese standardmäßigen “Probleme” müssen nicht vollkommen neu in jedem Projekt gelöst werden. Entwickler müssen mit Nest.js nur noch Controller, Services an der richtigen Stelle anlegen und sich um die Businesslogik kümmern.

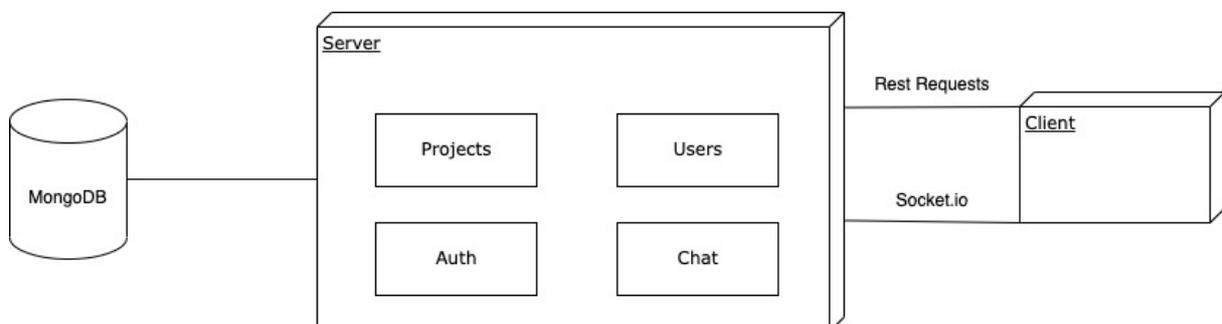
Als Datenbank haben wir MongoDB verwendet, weil das eine dokumentenorientierte Datenbank ist, die das Entwickeln von Webapplikationen vereinfacht, da man auch mit REST über eine dokumentenorientierten Struktur Daten austauscht.

Für die Authentifizierung haben wir die Passport Bibliothek verwendet, die in Nest.js schon eingebaut ist.

Die Chat-Funktion wurde mit Hilfe von socket.io implementiert und jedes Projekt hat einen eigenen Chat, wo alle Mitglieder miteinander kommunizieren können.

Wir haben in der Zukunft auch vor, die socket.io Bibliothek für ein “Real-Time Notification System” zu Nutzen zu bringen.

Architektur



Frontend

Mit React haben wir uns im Frontend die Arbeit vereinfacht, da wir bereits einige und gute Erfahrungen mit React haben. Der einzige Unterschied zu vorherigen Projekten war nun der Einsatz von Typescript, da wir zuvor nur mit Vanilla Javascript gearbeitet hatten. Besonders im static typing von Typescript haben wir einen großen Vorteil im Vergleich zu Javascript gesehen, da Typings bereits zur Compile Zeit feststehen und bei den Fehlersuchen einiges an Zeit sparen kann.

Das Styling wurde mithilfe von styled-components gemacht, da es generell mehr Funktionen und Flexibilität hat als normalen CSS.

Probleme

Durch den Fakt, dass wir uns bereits zu Beginn gut abgesprochen haben, wie unser Zeitplan ist und wir uns organisiert haben, hatten wir nur wenige Probleme. Dennoch sind wenige Probleme aufgetreten.

Da wir mit Nest.js und socket.io sehr wenig Erfahrung hatten, haben wir viel länger als erwartet gebraucht, um das Backend und den Chat zu implementieren.

Ein weiteres Problem war trotz der Planung die Zeit. Da wir zu zweit waren, war der Aufwand im Allgemeinen sehr hoch.

Das konnten wir dann zum Ende des Semesters spüren, da es zu Beginn sehr gut mit der Zeit lief, man allerdings gegen Ende mit weiteren Abgaben und Prüfungen noch weniger Zeit hatte und immer wieder Bugs aufgetreten sind, was weiterhin der Fall ist, und diese noch gelöst werden müssen.