

Beschte Onboarding

Julian Thiele

14. November 2023

Inhaltsverzeichnis

1	Einleitung	3
1.1	Ausbildungsbetrieb	3
1.2	Projektumfeld	3
1.3	Projektabgrenzung	3
1.4	Abweichungen vom Projektantrag	3
2	Projektplanung	4
2.1	Identifizierung der Arbeitspakete	4
2.2	Zeitplan	4
2.3	Ressourcenplanung	5
2.4	Kostenplanung	5
2.5	Nutzwertanalyse	5
3	Analysephase	5
3.1	Anforderungsgespräch	5
3.2	IST-Zustand Analyse	6
3.3	SOLL-Konzept	6
4	Designphase	7
4.1	Benutzeroberfläche	7
4.2	Datenbank	7
5	Implementierungsphase	8
5.1	Frontend	8
5.2	Backend	10
6	Testphase	10
7	Fazit	10

1 Einleitung

1.1 Ausbildungsbetrieb

Die BREDEX GmbH wurde im Jahr 1987 in Braunschweig gegründet. Den Schwerpunkt bildet die individuelle Softwareentwicklung, die BREDEX führt jedoch auch Beratungen zu Datenschutz, Datensicherheit, Qualitätssicherung von Software und Schulungen durch. Die BREDEX GmbH besitzt, mit seiner Tochterfirma BREDEX HUNGARY KFT. zusammen, ca. 200 Mitarbeiter.

1.2 Projektumfeld

Das Projekt wird unter der Aufsicht von Herr Christian Rucinski in den Geschäftsräumen der BREDEX GmbH entwickelt. Bei der Anwendung handelt es sich um ein BREDEX internes Tool. Sie soll für den Onboardingprozess neuer Mitarbeiter eingesetzt werden und den sogenannten BREDEX Buddy, einen Ansprechpartner neuer Mitarbeiter, bei seiner Arbeit unterstützen.

Das Frontend wird mit dem Framework Vue, bei welchem die Scriptsprache TypeScript Anwendung findet, geschrieben. Als Backend wird Firebase als "Backend As a Service" genutzt, in welche die NoSQL-Datenbank Firestore integriert ist.

1.3 Projektabgrenzung

Durch die Fachinformatikerausbildungsverordnung wird das Projekt zeitlich auf 80 Stunden begrenzt.

Die Anwendung ist ein alleinstehendes Tool und wird daher in kein Projekt eingegliedert. Daher gibt es keine weiteren Beschränkungen, auf die geachtet werden muss.

1.4 Abweichungen vom Projektantrag

Es gibt keine Abweichungen vom Projektantrag.

2 Projektplanung

2.1 Identifizierung der Arbeitspakete

Zu Beginn findet ein Anforderungsgespräch statt, bei dem die Projektziele definiert werden und es keine Differenzen im Verständnis dieser gibt und es definiert ist, welche Anforderungen erfüllt werden müssen.

Im Anschluss kommt die Analysephase, bei der Soll-Zustand niedergeschrieben wird.

Das dritte Paket bildet die Designphase, hierbei werden Benutzeroberfläche und Datenmodell geplant.

Daraufhin folgt die Entwicklung der Anwendung, bei der die Ergebnisse aus den vorherigen Arbeitspaketen umgesetzt werden. Um mögliche Fehler während der Implementierung frühzeitig zu erkennen und beheben zu können, finden Tests statt.

Im letzten Arbeitsschritt wird die Dokumentation des Projektes erstellt.

2.2 Zeitplan

Das Projekt wird im Zeitraum vom 22.09.2023 bis zum 22.11.2023 bearbeitet. Die im vorherigen Abschnitt erwähnten Arbeitspakete wurden in folgenden Zeitplan (Tabelle 1) eingeteilt.

Arbeitspaket	Dauer (in h)
Anforderungsgespräch	3
Analysephase	6
Planungsphase	6
Designphase	6
Umsetzung	40
Test und Abnahme	7
Dokumentation	12
Gesamt	80

Tabelle 1: Zeitplan

2.3 Ressourcenplanung

Um die Kosten so gering wie möglich zu halten, wurde nach Möglichkeit nur kostenlose oder bereits vorhandene Software, sowie Hardware genutzt. Als Arbeitsgerät wurde von der BREDEX GmbH ein Laptop mit Windows 10 und der Entwicklungsumgebung VisualStudio Code bereitgestellt. Des Weiteren wurde Firebase mit Firestore als Datenbank und Git als Versionsverwaltung eingesetzt.

2.4 Kostenplanung

Für den Prüfungsteilnehmer wird ein Stundenlohn von 9 EUR angesetzt. Ihm stehen für die Bearbeitung 80 Stunden zur Verfügung. Der Projektbetreuer führt das Anforderungsgespräch durch und steht dazu noch für weitere Fragen während der Bearbeitung zur Verfügung. Daher werden für ihn 6 Stunden angesetzt, bei einem geschätzten Stundenlohn von 45 EUR. Da nur bereits vorhandene oder kostenlose Ressourcen genutzt werden, ergeben sich keine weiteren Lizenz- oder Nutzungskosten.

Als Gesamtkosten berechnen sich hiermit 990 EUR (Tabelle 2).

Akteur	Stundensatz (in EUR)	Dauer (in h)	Kosten (in EUR)
Prüfungsteilnehmer	9	80	720
Projektbetreuer	45	6	270
Gesamtkosten			990

Tabelle 2: Kostenplan

2.5 Nutzwertanalyse

Kommt noch :D

3 Analysephase

3.1 Anforderungsgespräch

Als Erstes wurde ein Anforderungsgespräch mit dem Kunden durchgeführt. In dem Gespräch wird der IST-Zustand ermittelt und das SOLL-Konzept

definiert. Außerdem wurden die vom Kunden gewünschten Features der Anwendung übermittelt. Die wichtigsten Features der Anwendung sind:

- Achievements, die abgehakt werden können
- Rangliste, um sich mit Mitarbeiter*innen vergleichen zu können
- Als Administrator Achievementliste bearbeiten können

Die Punkte wurden in dem Gespräch noch weiter konkretisiert, um die Anwendung im Anschluss möglichst nach den Wünschen des Kunden implementieren zu können.

3.2 IST-Zustand Analyse

Damit neue Mitarbeiter*innen der BREDEX GmbH besser in das Unternehmen finden, werden ihnen ein Ansprechpartner, den sogenannten BREDEX-Buddy an die Seite gestellt. Dieser soll dem*der Mitarbeiter*in das Unternehmen näherbringen. Allerdings ist nicht genau klar, was die Aufgaben des Buddys alles beinhalten. Es gibt unterschiedlichen Orten findet man Listen mit verschiedenen Inhalten. Dadurch führen die Buddys ihre Aufgabe unterschiedlich aus.

3.3 SOLL-Konzept

Um in Zukunft dafür zu sorgen, dass alle Buddys ihre Aufgaben auf die gleiche Art und Weise zu erledigen soll es eine Webanwendung geben, bei der es eine Checkliste gibt, die neue Mitarbeiter*innen mit ihren Buddys abarbeiten können.

Zudem soll es eine Rangliste geben, in der sich Mitarbeiter*innen gegenseitig sehen können, die zur selben Zeit angefangen haben. Diese Gamification soll die Motivation erhöhen, die Achievements zu erledigen, wodurch die neuen Mitarbeiter*innen sich mehr in das Unternehmen eingliedern.

4 Designphase

4.1 Benutzeroberfläche

Bei dem Design der Benutzeroberfläche (im Folgenden GUI genannt) wird besonders darauf geachtet, dass alle Daten möglichst übersichtlich dargestellt werden. Dazu werden für jede der großen Anforderungen jeweils einzelne Unterseiten (im folgenden View genannt) unterteilt.

Hierbei gibt es eine View, um alle Achievements einsehen zu können und diese abhaken zu können. Achievements werden zu verschiedenen Zeitpunkten freigeschaltet. Manche sind direkt verfügbar, manche nach einer Woche, einem Monat oder erst nach einem halben Jahr. Entsprechend dieser Zeitstempel sind die Achievements in Akkordeons gruppiert. Dies führt dazu, dass Achievements übersichtlich einsehbar sind.

Eine weitere View gibt es für die Ansicht der Rangliste. Hier wird der derzeit eingeloggten Person eine Rangliste mit den Personen die zu einer ähnlichen Zeit angefangen haben angezeigt, mit Name, Email und Punktzahl.

Eine letzte View gibt es, um die Achievements verwalten zu können. Die View ist ähnlich aufgebaut, wie die Achievements-View. Die Achievements werden in Akkordeons gruppiert, jedoch befindet sich neben jedem Titel der Akkordeons ein weiterer Button, um neue Achievements für den Zeitpunkt hinzufügen zu können. Außerdem ist der Button um Achievements abzuhaken ausgetauscht durch jeweils einen Button um das Achievement löschen oder bearbeiten zu können.

Beim Hinzufügen oder Bearbeiten eines Achievements öffnet sich ein Modaldialog, bei dem sich die Daten bearbeiten lassen und dann der Vorgang abgeschlossen werden kann. Hierbei werden fehlende Eingaben durch eine rote Umrandung des Eingabefeldes vermittelt.

4.2 Datenbank

Bei der Datenbank handelt es sich um die NoSQL Datenbank Firestore von Firebase. Die Daten werden in JSON-Format gespeichert. Hierbei wird von Sammlungen und Attributen gesprochen. Sammlungen JSON-Objekte, mit einer ID und einem Datensatz. Die Daten können entweder aus weiteren Sammlungen oder Attributen wie Integern, Arrays oder ähnlichem bestehen.

Um die Achievements und die Daten der Nutzer der Anwendung speichern zu können wurden in der Datenbank jeweils Sammlungen angelegt.

5 Implementierungsphase

5.1 Frontend

Im Frontend wird das Webframework Vue eingesetzt. Dieses baut auf den Standard Websprachen HTML, JavaScript und CSS auf. Vue nutzt komponentenbasierte Programmierung, indem es HTML um die Möglichkeit erweitert, Templates zu erzeugen, welche dann entsprechend dem Status des JavaScript Datei die Benutzeroberfläche erzeugen. In dieser Anwendung wird Vue-Routing genutzt um eine Single-Page Anwendung zu erstellen. Dafür wurde sich entschieden, da Single-Page Anwendungen eine

Standardmäßig wird in Vue die Scriptsprache JavaScript genutzt. Das Framework kann auch mit der Sprache TypeScript genutzt werden, wofür sich in diesem Projekt entschieden wurde, um die Vorteile von Typisierung nutzen zu können. Dadurch können gewisse Fehler bereits zur Kompilierzeit vom Compiler entdeckt werden, wodurch die Anwendung weniger Fehleranfällig wird.

Vue stellt seit Version 3.0 auch eine neue API zur Verfügung: die Composition API. Die Vorteile der in Vue Standardmäßig genutzten Options API, ist die Möglichkeit den Code variabler und besser zu strukturieren, was die Lesbarkeit und Qualität des Codes verbessert. Außerdem erleichtert die Composition API die Wiederverwendbarkeit von Code in verschiedenen Komponenten, was eine große Auswirkung auf Codequalität und Fehleranfällig hat.

Für Styling wird das CSS-Framework Bootstrap genutzt, um einfacher ein modernes Design für die Anwendung erstellen zu können und sich auf die Programmierung des Projektes kümmern zu können.

Vue organisiert die Website über mehrere Single-Page-Komponenten. Hierbei handelt es sich um eine Dateistruktur, bei der jede Datei seine eigene Komponente beinhaltet. Die Wurzelkomponente einer jeden Vue-Anwendung stellt die App-Component dar. In dieser wird der `<vue-router>` Tag genutzt. Dadurch lässt sich ein Teil des Inhaltes der App-Komponente dynamisch mit anderen Komponenten austauschen.

Als erstes wurde die Achievement-View erstellt, da es sich bei der View um die Hauptanforderung der Website handelt. Hier werden alle Achievements angezeigt, die der Benutzer erledigen kann und ob diese bereits erledigt wurden. Im oberen Teil der Website gibt es eine kurze Einführung und eine Fortschrittsleiste zeigt an, wie viel der Nutzer bereits geschafft hat. Jede Vue-Komponente hat seinen eigenen Lifecycle. Hierzu gehören beispielsweise das Erstellen, Mounten, Aktualisieren oder Unmounten der Komponente. Der OnMounted-Lifecycle Hook wird hier genutzt, um kurz vor dem Anzweigen der Komponente die Daten aus der Datenbank zu laden. Diese Vorgehensweise wird in anderen Komponenten ebenfalls genutzt, um Daten zu laden.

Als nächstes wird die Login-View erstellt. Da in der BREDEX GmbH Microsoft Accounts genutzt werden, kann sich der Nutzer hier mit diesem Anmelden. Außerdem werden neue Nutzer erkannt, indem UUIDs verglichen werden und für jeweilige Nutzer neue Einträge in der Datenbank erstellt.

Im Anschluss wurde die Rangliste erstellt. Beim Laden der Nutzer aus der Datenbank können zusätzliche Bedingungen angegeben werden. Dadurch werden nur benötigte Daten aus der Datenbank geladen, um weniger Speicherplatz im lokalen Speicher des Clients speichern zu benötigen.

Die letzte View ist zum Verwalten der Achievements. Auf diese View sollen nur Personen mit bestimmten Gruppen Zugriff haben, wie beispielsweise die Personalabteilung. Dieser Zugriff kann mit dem Vue-Router implementiert werden, welcher eine Möglichkeit hat, eine Bedingung an die Weiterleitung an die Adresse zu binden. In dieser Bedingung wird die Rolle des Nutzers abgefragt. Da diese bei der Anmeldung über Firebase nicht mit dem Access-Token übergeben wird, muss sich separat mit dem Azure Portal verbunden werden, um diese abzufragen.

Um zuletzt die Mehrsprachigkeit umzusetzen, wird sich das Vue Plugin Vue I18N zunutze gemacht. In einer separaten Datei wird zu jedem Text der Anwendung im JSON-Format ein Key gespeichert und die jeweiligen Übersetzungen. In dem HTML-Template wird nun anstelle des Textes der Key eingetragen, welcher beim Erstellen der Komponente, je nach aktuell eingestellter Sprache, durch die Übersetzung ersetzt wird.

5.2 Backend

Für das Backend wird das Backend-As-A-Service Firebase genutzt. Darin sind verschiedene Optionen zur Benutzeranmeldung, wovon nur die Anmeldung über Microsoft genutzt wird, sowie die NoSQL Datenbank Firestore. Als erstes muss das Backend mit der Software verbunden werden. Dafür muss als erstes die Firebase SDK mit dem Befehl

```
npm install firebase
```

über Node.js installiert werden und im Anschluss in den Abhängigkeiten der Vue-Anwendung hinzugefügt werden. Nun muss nur noch die Vue-App mit den Firebase Einstellungen konfiguriert werden.

Als nächstes wird die Datenbank erstellt. Die NoSQL Datenbank basiert auf JSON-ähnlichen Dateistrukturen. Man kann Sammlungen erstellen, mit verschiedenen Elementen. Es wird eine Sammlung erstellt für die Nutzer und eine Sammlung für die Achievements.

6 Testphase

7 Fazit