

1. Auftraggeber

Im Auftrag von der Codire GmbH war es im Sommersemester 2022 unsere Aufgabe, ein Zeitbuchungssystem in Form einer Web-Applikation zu erstellen. Codire ist ein Softwaredienstleister und bietet CTO as a Service an. Das bedeutet, Codire bietet die Dienstleistung an, dass Kunden deren gesamte IT auslagern können und diese dann von Codire verwaltet werden kann. Da Projektstunden der Mitarbeiter von Codire direkt beim Kunden abgerechnet werden, ist es wichtig ein gutes und detailliertes Zeiterfassungssystem zu haben. Zudem arbeiten die meisten Mitarbeiter von Codire an vielen verschiedenen Projekten für verschiedene Kunden gleichzeitig, sprich die Projektstunden sollten einfach und zentral dokumentiert werden können.

2. Ist-Zustand

Die Zeiterfassung wird aktuell funktional durch eine GraphQL-API durchgeführt, welche über Queries und mutations mit dem ERP System

von Codire kommuniziert. Aktuell müssen die Mitarbeiter ihre Zeiten über eine Browser-Extension namens Altair-GraphQL-Client durch manuell eingegebene Queries bzw. Mutations buchen (siehe Abbildungen). Diese können zwar gespeichert werden, es ist dennoch eine schlechte User Experience und dauert zu lang.

```
(Run mutation)
mutation {addTimesheetEntry(data: {task_id:27 description:"meine Buchung"
break_duration:50 start:"2022-04-07 14:30" end:"2022-04-07 17:40"})
{DESCRIPTION START END BREAK_DURATION}}
```

```
{
  "DESCRIPTION": "meine Buchung",
  "START": "2022-04-07T14:30:00.000Z",
  "END": "2022-04-07T17:40:00.000Z",
  "BREAK_DURATION": 50
}
```

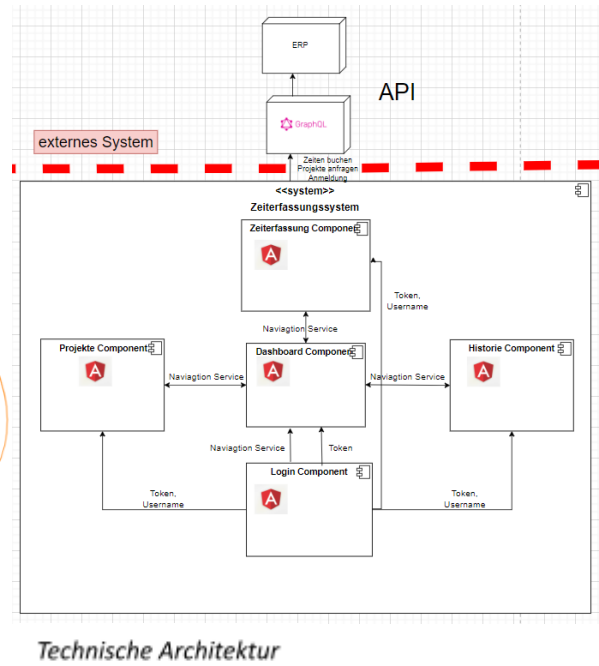
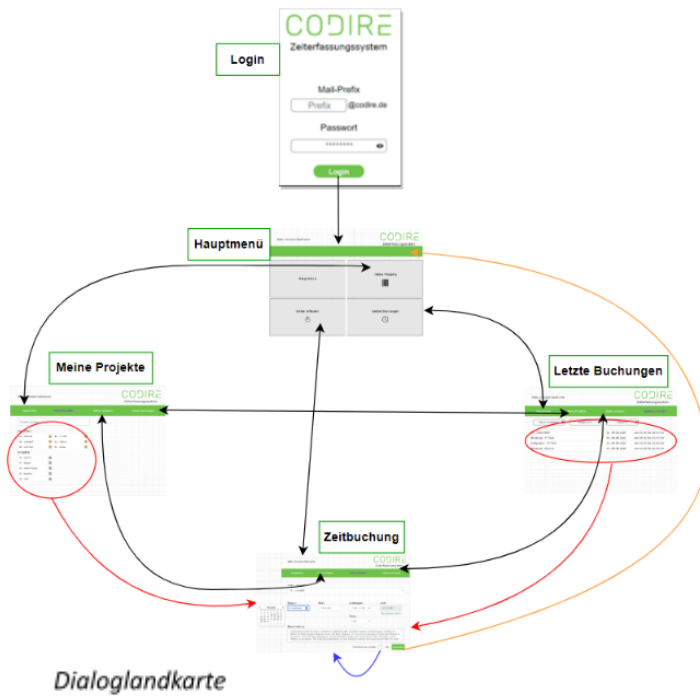
3. Soll-Zustand

Aufgrund der oben genannten Sachlage war es in unserem Projekt das Ziel, die User-Experience bei der Zeitbuchung für Mitarbeiter von Codire zu verbessern. Aufgabe war es, ein Frontend für die GraphQL-API entwickeln, dass die Zeitbuchung für die operativen Mitarbeiter maximal einfach und schnell gestaltet. Nach intensiver Absprache mit Codire haben wir uns für eine Webapplikation in Angular entschieden.

4. Ergebnis

Das System besteht aus fünf Angular Komponenten. Login, Dashboard, Zeiterfassung, Historie und Projekte. Die Die Landing-Page wird durch die Login-Komponente abgebildet. Dort kann sich der Benutzer mit seiner Firmenkennung anmelden. Im Hintergrund wird ein Token bei erfolgreicher Anmeldung generiert und auf das Dashboard weitergeleitet. Danach ist es möglich, eine neue Zeitbuchung anzulegen, Projekte einzusehen oder die eigenen Buchungen zu verwalten.

Bei der Verwaltung der Buchungen ist es möglich, alte Buchungen, die noch nicht abgerechnet wurden, zu bearbeiten oder zu löschen. In der linken Abbildung werden die beschriebenen Prozesse noch einmal als Dialoglandkarte veranschaulicht. In der rechten Abbildung ist die technische Architektur der Angular Komponenten modelliert.



5. Erkenntnisse

Da vier aus unserem Team bis vor dem Projekt mit Angular, Typescript, node.js und allgemein mit Webentwicklung bzw. Frontend-Entwicklung noch kaum Berührungspunkte hatten, ging die Lernkurve zu Beginn des Projekts steil nach oben. Der Einstieg in das Typescript-Framework Angular fiel uns leicht, da wir bereits grundlegende Programmierparadigmen durch Java und C# kannten. Wir haben die Komponentenarchitektur von Angular kennengelernt und verinnerlicht. Auch allgemein die Entwicklung im Frontend war sehr interessant, da man direkt sichtbar die Software und User Experience beeinflusst.

Aber auch im Projektmanagement-Teil konnten wir vieles mitnehmen. Wir lernten, wie wichtig aktive Kommunikation mit dem Auftraggeber und regelmäßiges Feedback ist. Zudem wurde die Wichtigkeit Scrum Artefakte, wie Dailys und Product Backlog im Laufe des Projekts immer deutlicher erkennbar.

Team

