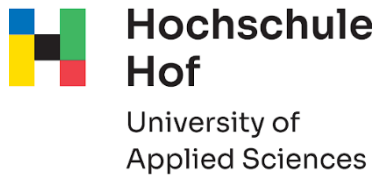


Studienarbeit
im Fach Praktikum Softwareentwicklung

*Task Management-System –
mit integriertem Raumbuchungssystem*



Verfasser: ***Dejan Fraas***
00481613
Gruppe: "Bottlenecks"
Sommersemester 2022
Hochschule Hof

Dozent: Dipl.-Inf. *Stefan Müller*

Abgabetermin: *08.07.2022*

Inhaltsverzeichnis

1. Vorwort	3
2. Vorbereitungen	3
2.1 Gruppenbildung	3
2.2 Themenauswahl	3
2.3 Auswahl der Technologien und Frameworks.....	4
2.4 Aufgabenaufteilung der jeweiligen Gruppenmitglieder	4
3. Organisatorisches	4
3.1 Projekt Zusammenführung	4
3.2 Einweisung des neuen Mitglieds.....	5
4. Design.....	5
4.1 Gestaltung der Wireframes	5
5. Frontend Struktur.....	5
5.1 Allgemein	5
5.2 User-Dashboard	5
5.3 Projekt-Dashboard	6
5.4 Weitere Elemente	6
6. Frontend Implementierung	6
6.1 Allgemein	6
6.2 User-Dashboard – offene und abgeschlossene Tasks	7
6.3 User-Dashboard – Ankündigungen	7
6.4 User-Dashboard – Account Verwalten	7
6.5 User-Dashboard – Raumbuchungen	8
6.6 User-Dashboard – Infoseite	8
6.7 User-Dashboard – Account Verwalten	8
6.8 Projekt-Dashboard – Räume.....	8
6.9 User-Dashboard & Projekt-Dashboard - PopUps	9
6.10 User-Dashboard & Projekt-Dashboard - Funktionalität Suchleiste und Sortierpfeile	9
6.11 User-Dashboard & Projekt-Dashboard - Funktionalität Suchleiste und Sortierpfeile . Fehler! Textmarke nicht definiert.	
7. Testen & Errorhandling	10
7.1 Errorhandling.....	10
7.2 Testen	10
8. Komponenten für Stiva Web-Applikation	Fehler! Textmarke nicht definiert.

1. Vorwort

Für den Kurs Software-Praktikum bestand die Aufgabe ein Software-Projekt im Rahmen von festgelegten Richtlinien innerhalb des Sommersemesters 2022 zu entwickeln. Dabei soll in einem Team eine arbeitsnahe Umgebung geschaffen werden. Darunter fällt die Zusammenarbeit mit „Arbeitskollegen“ sowie die Aufgabenaufteilung und auch organisatorische Aspekte. Dabei soll auf zukünftige Geschehnisse vorbereitet werden.

2. Vorbereitungen

2.1 Gruppenbildung

Zu Beginn bestand unsere Gruppe aus 3 Mitgliedern Johannes Matus, Daniel Vogel und Eugen Kudraschow. Dabei haben wir uns auf „Bottlenecks“ für den Gruppennamen geeinigt. Zum späteren Verlauf musste ein Mitglied der Gruppe „Bottlejobs“ krankheitsbedingt ausfallen, weshalb diese dann nur noch aus 2 Mitgliedern bestanden. Aufgrund dessen haben sich das Team „Bottlenecks“ und „Bottlejobs“ zusammengetan. Somit haben wir eine endgültige Gruppengröße von 5 Mitgliedern.

2.2 Themenauswahl

Nach der ersten Veranstaltung und Gruppenbildung, haben wir uns zum Brainstorming getroffen, um geeignete Themen für unsere Studienarbeit zu bekommen. Bei dem Team „Bottlejobs“ fiel die Themenwahl sehr schnell auf einen Raumplaner. Ursprünglich geknüpft an die Hochschule. Dies stellte sich allerdings, aufgrund der Zusammenführung der Teams als unpraktisch heraus, weswegen die Raumplanungsfunktion generell für Räume aller Art in das bereits vorhandene Task-Management-System integriert wurde.

2.3 Auswahl der Technologien und Frameworks

Da parallel zum Software-Praktikum eine weitere Studienarbeit erstellt werden sollte, fiel die Entscheidung einfach uns auf eine weitere Web-Applikation zu einigen. So kann gelerntes aus beiden Kursen effektiver eingesetzt werden und man erhält mehr Erfahrung im Bereich der Web-Entwicklung, welches ein großes Themengebiet in der Informatik ist.

Bei der Auswahl der Frameworks entschieden wir uns für Laravel, welches im Hintergrund (Backend) läuft und die Datenbank sowie Struktur des Datenmodelles enthält. Für die grafische Darstellung und Benutzereingaben haben wir uns im Frontend für das Framework React entschieden. Als Schnittstelle wurde eine Rest-API von seitens Laravel entwickelt.

2.4 Aufgabenaufteilung der jeweiligen Gruppenmitglieder

Durch unsere Gruppengröße von anfangs 3 Leuten war es einfach die Aufgaben innerhalb des Projektes aufzuteilen. Johannes Matus hat sich mit dem Framework Laravel beschäftigt und war somit für das Datenmodell und die Datenbank zuständig. Diese werden bei jeder Web-Applikation benötigt. Daniel Vogel agierte als Bindeglied zwischen uns und hat beide Aufgaben teilweise übernommen. Eugen Kudraschow hat sich mit dem Frontend auseinandergesetzt. Hierzu war ein Einfinden in React notwendig, um grafische Darstellungen von Komponenten und Daten zu gewährleisten. Wie unter Punkt 2.1 bereits beschrieben, wurden Sebastian Bär und ich dem Team „Bottlenecks“ im weiteren Verlauf zugeordnet. Sebastian wurde dem Backend und ich dem Frontend zugeordnet.

3. Organisatorisches

3.1 Projekt Zusammenführung

Wie bereits erwähnt mussten die Teams „Bottlenecks“ und „Bottlejobs“ kooperieren und beide Themen zusammenführen. Die Aufgabe bestand, das

Raumplanungssystem sinnvoll in ein Task-Management-System zu integrieren. Sobald wir jedoch die Raumplanung von der Hochschule getrennt hatten, stellte sich diese Aufgabe als konstruktiv und machbar dar.

3.2 Einweisung des neuen Mitglieds

Eugen Kudraschow hat mich in das bereits bestehende Projekt eingewiesen. Hierzu haben wir uns an einem Vormittag getroffen und alle bereits erledigten Punkte im Frontend besprochen. Ebenso eine erste grobe Aufgabenverteilung vorgenommen.

4. Design

4.1 Gestaltung der Wireframes

Grundlegend ist das Programm in ein User-Dashboard und ein Projekt-Dashboard aufgebaut. Die Implementierung der Raumbuchung hat sich in beiden Dashboards ausgewirkt. Im Projekt können Räume erstellt und gebucht werden, worauf diese dann in dem User-Dashboard angezeigt werden. Nach Zusammenschluss galt es für mich, die PopUps und die Raumseiten zusammen mit den anderen an die Wireframes anzupassen. Das Design ging aus den vorher erstellten Wireframes hervor.

5. Frontend Struktur

5.1 Allgemein

Grundsätzlich wird zwischen dem User-Dashboard und dem Projekt-Dashboard unterschieden. In dem Projekt-Dashboard ist auch das Admin-Dashboard zu finden. Da diese verschiedene Ansichtsmöglichkeiten und Funktionen haben. Zusätzlich gibt es für die Anmeldung und Registrierung einen eigenen Ordner.

5.2 User-Dashboard

Bei der Struktur im User-Dashboard wurde zwischen den einzelnen Reitern in der Navigationsleiste unterschieden. Diese sind:

Projekte	Zeigt die aktuellen Projekte an, in denen man sich befindet
Ankündigungen	Zeigt Ankündigungen von den jeweiligen Projekten an
Abgeschlossene Tasks	Zeigt abgeschlossenen Tasks innerhalb eines Projekts an
Offene Tasks	Zeigt offenen Tasks innerhalb eines Projektes an
Account verwalten	Damit lassen sich benutzerdefinierte Einstellungen ändern
Raumbuchungen	Zeigt die Raumbuchungen an
Info	Informationen über die Entwickler

5.3 Projekt-Dashboard

Bei der Struktur im Projekt-Dashboard wurde zwischen den einzelnen Reitern in der Navigationsleiste unterschieden. Diese sind:

Tasks	Zeigt alle Tasks innerhalb eines Projektes an
Bevorstehende Deadline	Zeigt Tasks an, die innerhalb einer Woche bzw. innerhalb eines Monats zu erledigen sind
Übersicht	Gibt eine Übersicht des aktuellen Projektes an
Ankündigungen	Zeigt Ankündigungen innerhalb eines Projektes
Tags	Zeigt Tags an, die zur Verfügung stehen
Räume	Gibt die Räume wieder die für das Projekt zur Verfügung stehen
Info	Informationen über die Entwickler

5.4 Weitere Elemente

Zusätzlich werden noch weitere Elemente benötigt, die dafür sorgen, dass die gesamte Seite gerendert wird. Diese beinhalten den Main-Bereich inkl. Der Suchleiste und auch der Navigationsleiste.

6. Frontend Implementierung

6.1 Allgemein

Wie ursprünglich angedacht, wollten wir eine klare Grenze ziehen, wer welche Komponenten bearbeitet. Im weiteren Verlauf haben sich diese Grenzen immer mehr überschritten. Zum Schluss hat jeder an allen Komponenten gearbeitet. Nachfolgend werden die Komponenten erwähnt, an denen ich hauptsächlich beteiligt war.

Die Views in den einzelnen Komponenten wurden meist durch eine minimale Ansicht ermöglicht, die immer zu sehen ist. Diese kann man durch Klicken eines Pfeiles maximieren, wodurch weitreichendere Informationen zu finden sind.

6.2 User-Dashboard – offene und abgeschlossene Tasks

Die Useransicht der offenen und abgeschlossenen Tasks beinhalten alle Tasks, an denen der User beteiligt ist. Bei den offenen Tasks werden die Priorität, die Deadline und die Beschreibung der Aufgabe angezeigt. Auch in welchem Raum die Tasks ausgeführt werden soll, wird angezeigt. Der Zustand des Tasks wird farblich hervorgehoben, hierunter fallen „in Arbeit“, „Abgeschlossen“ und „Abgebrochen“. Hier kann man einen Task auch abschließen, ohne direkt auf die Projektansicht zugreifen zu müssen.

Bei den abgeschlossenen Tasks sieht man selbstverständlich alle beendeten Tasks, zusätzlich zum Status, in dem der Task zum Abschluss stand, erhält man Informationen über den Abschluss.

6.3 User-Dashboard – Ankündigungen

Hier findet man alle Ankündigungen, welche grundlegend von dem Admin im Admin-Dashboard für ein gewisses Projekt erstellt werden, aller Projekte, in denen der User aktuell arbeitet. Die View entspricht wieder einer minimalen Ansicht und einer optionalen, maximalen Ansicht.

6.4 User-Dashboard – Account Verwalten

Hier bieten wir dem User die Möglichkeit sowohl seinen Vornamen und seinen Nachnamen, also auch seine E-Mail und sein Passwort zu ändern. Für die Änderung der Namen haben wir direkt ein Eingabefeld verwendet, um mehr Sicherheit zu bieten, haben wir uns bei dem Passwort und der E-Mail für ein PopUp entschieden, auf das ich im Punkt 6.9 genauer eingehe.

6.5 User-Dashboard – Raumbuchungen

Hier werden alle Raumbuchungen eines Users angezeigt. Die Buchungen generiert man grundlegend in der Projektansicht. Es lassen sich Informationen zu dem Raum, Zeitangaben und den Mitarbeitern finden. Natürlich kann eine Buchung wieder gelöscht werden, welches in der maximierten View ersichtlich ist und ein automatisches Rendern initiiert.

6.6 User-Dashboard – Infoseite

Diese kleine Seite dient als Impressum und Informationen zu den Entwicklern.

6.7 User-Dashboard – Account Verwalten

Hier wird dem User ermöglicht sowohl seinen Vor- und Nachnamen als auch seine E-Mail und sein Passwort zu ändern. Aus Sicherheitsgründen kann der User direkt in die Namensfelder schreiben und auf Knopfdruck speichern, für das Passwort und die E-Mail muss er jedoch ein extra PopUp öffnen, auf das ich bei 6.9 genauer eingehe.

6.8 Projekt-Dashboard – Räume

Da die Raumbuchungen aus meiner alten Gruppe hinzukamen, habe ich mich auch für die Projektseite darum gekümmert. Hier können vom Admin Räume erstellt werden, in dem durch Knopfdruck über ein PopUp verschiedene Daten wie Raumnummer, Kapazität, Öffnungszeiten, Adresse, Titel, Beschreibung und die Ausstattung eingegeben werden. Sind Räume erstellt, sind diese für alle User, die in diesem Projekt sind, sichtbar und auch von ihnen zur Buchung auswählbar.

Mit der minimierten Sicht der Räume in diesem Reiter lässt sich über einen weiteren Knopf das PopUp zur Raumbuchung öffnen. Hier stehen alle Buchungen, die diesen Raum betreffen. Mit Angabe des gewünschten Zeitraumes kann die Buchung abgeschlossen werden.

Die Buchung wird dann wie oben beschrieben im User-Dashboard unter Raumbuchungen angezeigt.

6.9 User-Dashboard & Projekt-Dashboard - PopUps

Einer der größten Bereiche galt den PopUps, da diese in sehr vielen Komponenten unterkommen mussten. Jeweils beim Erstellen von: Projekten, Tasks, Räumen und Raumbuchungen, sowie dem Bearbeiten der Accountdaten wird ein PopUp verwendet. Das PopUp liegt in der Elternkomponente und wird abhängig von einer booleschen Variablen angezeigt. Die Schwierigkeit hierbei war, dass die Daten aus der Elternkomponente nicht beim Laden der PopUp-Komponente gezogen werden, da diese zur selben Zeit geladen werden, was zu Datenverlust führen kann und auf alle Daten bezogen ist, sondern erst bei Betätigung der jeweiligen Knöpfe.

6.10 User-Dashboard & Projekt-Dashboard - Funktionalität Suchleiste und Sortierpfeile

Wir haben uns dafür entschieden eine Gruppe von Komponenten für die Such- und Sortierleiste zu nehmen, bestehend aus Searchbar, Sortierfeld und einzelnen Sortierelementen. Die Funktionalität habe ich für alle Komponenten hinzugefügt. Gesucht wird immer nach Schlagwörtern abhängig von der Komponente, in der sich der User befindet (z.B. Projekttitel, Tasktitle, Raumtitel). Die Suchfunktion wird bei jeder Eingabe im Suchfeld getriggert, das Wort wird mit der in JS eingebauten Filter-Funktion und dem gewünschten Parameter verglichen. Hierbei wird case-sensitivity missachtet wird, da das dem User eine angenehmere Sucherfahrung bietet. Alle Datenblöcke, die dem Parameter entsprechen werden in einem React-State gespeichert und die Seite wird abhängig davon gerendert. Das Rendern auf alle Daten geschieht, sobald die Suchleiste wieder leer ist.

Die Sortierfunktion beinhaltet etwas mehr Aufwand, da die Identifizierung des richtigen Pfeiles notwendig war. Dies geschieht über eine nach oben gereichte ID.

Des Weiteren wird eine boolesche Variable mitgegeben, mit der Ausfindig gemacht wird, ob ab- oder aufsteigend sortiert werden soll. Das Sortieren an sich geschieht dann nur noch mit der eingebauten sort-Funktion.

6.12 API-Requests

Die meisten API-Requests bestanden aus GET beim Laden der Komponente oder aus POST beim Erstellen gewisser Datensätze über PopUps wie bei 6.9 beschrieben. Besonders bei den POST-Requests war es wichtig, die Daten aus den verschiedenen Eingabefeldern in das richtige Format zu bekommen. Vereinzelt kamen auch DELETE und PUT auf, dies wurde aber hauptsächlich von Daniel übernommen.

7. Testen & Error Handling

7.1 Error Handling

Beim Error Handling sind wir agil vorgegangen. Bedeutet, wenn ein Fehler beim Programmieren festgestellt wurde, wurde er sofort behoben. Bei Fehlermeldungen die später festgestellt wurden, wurde der Fehler zuerst analysiert, in welcher Komponente es auftritt, danach verteilt an die zuständige Person und behoben. Typische Fehler waren z.B. falsch aufgerufene Routen von der API oder Listen von Objekten wurden falsch geladen. Diese sind sogar gravierend aufgefallen, da es zum Systemabsturz der gesamten Seite gekommen ist.

7.2 Testen

Beim Testen wurden die verschiedenen Funktionen der Web-Applikation mit Benutzer Interaktionen getestet. Diese wurde parallel zur Programmierung durchgeführt. Zum Ende der Programmierphase wurde gemeinsam jede Funktion getestet.