

Dokumentation Fullstack-Challenge

Thema: Webapplikation

Dokumentinformationen

Dateiname: Dokumentation
Speicherdatum: 26.06.2024

Autoreninformationen

Autor: Ramis, Jan, Lukas und Tahssin

Inhaltsverzeichnis

1	Einleitung.....	3
3	Informieren.....	4
3.1	Aufgabenstellung:	4
3.2	Lösung und Nutzer:.....	4
3.3	Anforderungen:	4
	Verfügbare Ressourcen:.....	4
3.4	Dokumentation und Kommunikation:	4
3.5	Belohnung:.....	4
4	Planung	5

1 Einleitung

Im Rahmen unserer Fullstack-Challenge haben wir eine Webapplikation für das Restaurant REST-Foods entwickelt, das bald an der Herostrasse 12 eröffnet. Diese Dokumentation beschreibt den gesamten Projektverlauf von der Aufgabenstellung bis zur finalen Auswertung. Unsere Webapplikation umfasst eine Menükarte und ein System für Tischreservierungen. Das Projektteam bestand aus Ramis, Jan, Lukas und Tahssin, die jeweils spezifische Aufgaben im Bereich Frontend und backend übernahmen.

Diese Dokumentation soll einen umfassenden Überblick über die verschiedenen Phasen unseres Projekts geben, inklusive Planung, Entscheidung, Realisierung, Kontrolle und Auswertung. Die Zusammenarbeit im Team, die verwendeten Methoden und die aufgetretenen Herausforderungen werden detailliert beschrieben. Am Ende der Dokumentation reflektieren die Teammitglieder ihre Erfahrungen und geben einen Ausblick auf mögliche Verbesserungen für zukünftige Projekte.

2 Informieren

2.1 Aufgabenstellung:

Wir entwickeln eine Webapplikation für das neue Restaurant REST-Foods, das bald an der Herostrasse 12 eröffnet. Die Applikation soll eine Menükarte und ein System für Tischreservierungen haben. Das Erstellen von Texten oder Bildern für die Webseite übernimmt das Restaurant selbst.

2.2 Lösung und Nutzer:

Die Applikation muss einfach zu bedienen sein, sodass Kunden leicht Menüs anschauen und Tische reservieren können. Das Restaurantpersonal sollte ebenfalls einfach die Daten in der App verwalten können.

2.3 Anforderungen:

Die fertige Webapplikation muss bis zum 26. Juni 2024 um 12:00 Uhr abgegeben (auf GitHub) werden. Die Präsentation der Applikation findet am selben Tag um 15:30 Uhr statt.

2.3.1 Verfügbare Ressourcen:

Wir können das Internet nutzen, um Informationen zu suchen, und haben Kurse, die uns helfen, mehr über die benötigten Themen mehr wissen zu erlangen. Bei Fragen können wir auch unsere Ausbilder fragen.

Verantwortlichkeiten:

- Backend: Tahssin (Menükarte), Lukas (Reservation)
- Frontend Ramis (Menükarte), Jan (Reservation)

2.4 Dokumentation und Kommunikation:

Am Ende unseres Projektes muss alles bis um 26.06 12:00 Uhr auf GitHub stellen, damit die Ausbilder sehen, an was wir gearbeitet haben.

2.5 Belohnung:

Als Dank für die Arbeit gibt es von REST-Foods Eiscreme für das ganze Team.

3 Planung

3.1 Meetings

Datum	Name	Tätigkeit	Probleme -> wie behoben	Ziel für den Tag/erreicht?
Mi 12.06	Ramis	Mockup erstellen	Figma Schwierigkeiten -> testen was geht	Mockup fertigen/nicht geschafft
	Jan	Mockup erstellen	-	Mockup fertigen/nicht geschafft
	Lukas	Docker aufsetzen	-	DB aufsetzen/erreicht
	Tahssin	Docker aufsetzen	Probleme mit Port	DB aufsetzen/erreicht
Do 13.06	Ramis	Mockup erstellen	-	Mockup zu Ende kriegen/erreicht
	Jan	Mockup erstellen	-	Mockup zu Ende kriegen/erreicht
	Lukas	Reservationslogik implementiert	-	Erreicht
	Tahssin	Menükarte Logik implementiert	Ständige Probleme mit Entities	Erreicht
Fr 14.06	Ramis	Menuseite implementieren	-	Menuseite ohne logik haben/erreicht
	Jan	Hauptpage anfangen	-	Hauptpage fertig kriegen/erreicht
	Lukas	Authority Erschaffen Mit Security	Frontend nicht genug weit (Security löschen)	nicht erreicht
	Tahssin	CRUD Elementente hinzugefügt	Fehler bei Endpoints	Crud implementiert, Postman Testing
Do 20.06	Ramis	Get auf menu im menupage	-	Get anfrage ok 200/erreicht
	Jan	-	-	-
	Lukas	Fehler Beheben Security neu Implementierung	Frontend muss daten schöpfen geht nicht mit Security (Wieder Gelöscht)	Erreicht

Webapplikation

	Tahssin	Swagger Implementation versucht	Falsche dependencies hinzugefügt	Swagger und Javadoc noch hinzugefügt
Fr 21.06	Ramis	Menupage design korrigieren	-	Schönes Design/erreicht
	Jan	-	-	-
	Lukas	Krank	Krank	Krank
	Tahssin	Postman Testing	Keine Probleme	Erfolgreich
Di 25.06	Ramis	Dashboard implementieren	Post conflict-> ük 294 kopiert	Dashboard Menu teil fertig implementieren/erreicht
	Jan	Reservation korrigieren	-	Reservation fertig/nicht erreicht
	Lukas	TBZ	TBZ	TBZ
	Tahssin	Swagger implementieren	Code im Fehler, welches Projekt nicht ausführen lässt	Projekt lässt sich ohne Probleme Ausführen

3.2 Zeitplan

Zeitplan												
Tag	Mittwoch 12.06.2024				Donnerstag 13.06.2024				Freitag 14.06.2024			
Datum												
Gruppenaufteilung	Frontend		Backend		Frontend		Backend		Frontend		Backend	
Personen	Jan	Ramis	Tahsinn	Lukas	Jan	Ramis	Tahsinn	Lukas	Jan	Ramis	Tahsinn	Lukas
Morgens (8 - 12 Uhr)	Zeitplan anfangen, Besprechung des Vorgehens, Zeitplan anfangen	Zeitplan anfangen, Besprechung des Vorgehens, repository erstellt, mockup beginnen	Zeitplan anfangen, Besprechung des Vorgehens	Zeitplan anfangen, Besprechung des Vorgehens	Mockup auf figma fertigen	Mockup auf figma fertigen	Menu(tem)logik implementieren	Reservationslogik implementieren	Pages ohne logik erstellen mit Design responsive	Pages ohne logik erstellen mit Design responsive	Role und user erstellen	Authorities erstellen
	Mockup auf figma erstellen	Mockup auf figma erstellen	Projekt Struktur erstellt und branch, Docker(DB) aufsetzen	Projekt Struktur erstellt und branch, Docker(DB) aufsetzen	Erste pages ohne logik erstellen, Footer	Erste pages ohne logik erstellen, Navibar	Menu(tem)logik implementieren	Reservationslogik implementieren	Pages ohne logik erstellen mit Design responsive	Pages ohne logik erstellen mit Design responsive	Role und user erstellen	Authorities erstellen
Nachmittag (13-16:45)												

Mittwoch (Infonachmittag) 19.06.2024				Donnerstag 20.06.2024				Freitag 21.06.2024				Dienstag 25.06.2024		Mittwoch 26.06.2024			
Frontend		Backend		Frontend		Backend		Frontend		Backend		Frontend		Backend		Frontend	
Jan	Ramis	Tahsinn	Lukas	Jan	Ramis	Tahsinn	Lukas	Jan	Ramis	Tahsinn	Lukas	Jan	Ramis	Tahsinn	Lukas	Jan	Ramis
Pages ohne logik Design und responsive korrigieren/hinz	ausfall	ausfall	ausfall	Routing implementieren für ganze pages mit Menu	Page mit logik (Anfrage) beginnen	Security beginnen	Security beginnen	Reservationspage beginnen	Menupage design verfeinern, responsive	Security beginnen	Security fertigstellen	Reservationsseite fertigstellen	Dashboard implementieren	Swagger implementieren und Javadoc Kommentieren	TBZ	letzte anfertigung im Web. Dokumentieren	Dokumentation fertigstellen
Pages ohne logik Design und responsive korrigieren/hinz	ausfall	ausfall	ausfall	Routing implementieren für ganze pages mit Menu	Page mit logik (Anfrage) fortfahren, pages mit Menu	Testdaten (sql) erstellen	Security beginnen	Reservationspage implementieren	Page mit logik -> Menuseite und Detailansicht	Security beginnen	Security fertigstellen	Reservationsseite fertigstellen, dashboard anfangen/h	Dashboard implementieren, Reflexion schreiben	Swagger implementieren und Javadoc Kommentieren	TBZ	Präsentation vorbereiten	Präsentation vorbereiten

Reflexion

Jan:

Ich habe das Team angenehm gefunden und wir hatten eine super Arbeitsaufteilung. Ramis und ich arbeiteten mit "branches" pro feature, weswegen wir uns fast nie in die Quere kamen.

Das Backend – Team war ebenfalls fleissig und schnell unterwegs. Das einzige Problem gab es bei dem Einfügen der Daten, da wir zuerst nicht wussten, was alles in die Datenbank eingefügt wird.

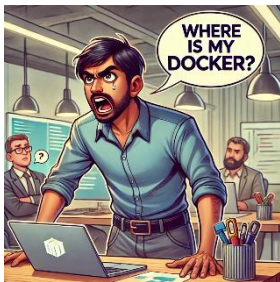
Ich hätte jedoch erwartet, dass wir alle 3 Prioritäten abschliessen können, jedoch konnten wir die 3. Priorität nicht vollständig abschliessen.

Lukas:



Meiner Ansicht nach hatten ich und mein Backend-Teammitglied Tahssin eine gute Synergie und konnten jegliche Hindernisse überwinden. Obwohl ich im Homeoffice war und einmal krank war, kam es immer zu einer stabilen Kommunikation. Der Fortschritt, den wir erreichen konnten, war sehr rasant, wurde jedoch schnell abgebremst.

Die Entwicklung des Frontendes kam mir zu Beginn eher langsam vor. Als das Frontend Daten verlangte, fragten sie ständig nach einem Docker. In unseren Augen ergab das jedoch keinen Sinn, da ein Docker nichts weiter als ein fester Datenbankersatz war. Aus diesem Grund zweifelte ich manchmal am Backend-Wissen des Frontend-Teams. Dieses unwissende Verhalten zog sich durch die gesamte Zeit, als sie das Backend erhielten, was unseren Entwicklungsfortschritt massiv zurückhielt. Wir mussten uns stundenlang mit offensichtlichen Fehlern und Erklärungen befassen, obwohl die Lösung meist an der Unfähigkeit lag, die Fehler zu verstehen.



Wir mussten vieles unserer Arbeit deaktivieren, da das Frontend einfach nicht weit genug war. Die ganze Arbeit, die wir in die Security gesteckt haben, war für nichts. Währenddessen hat das Frontend sich keine Mühe um die Security gemacht. Meine Projekterfahrung kann mit einem Wort beschrieben werden: FRUSTRIEREND.



Ramis:

Im Grossen und Ganzen bin ich zufrieden, aber es gibt immer Verbesserung Möglichkeiten. Meiner Meinung nach lief die Zusammenarbeit zwischen Jan und mir (beide Frontend) ziemlich gut, wir hatten eine gute Kommunikation miteinander und halfen uns gegenseitig bei Problemen bzw. Schwierigkeiten. Der Zeitplan wurde mittelmässig gehalten, da ich das Frontend erheblich unterschätzt habe bzw. das Programmieren in Typescript/React, da ich wenig Erfahrung hatte (max. 2 Woche) und somit etwas langsamer war als gedacht, aber mit der Zeit holte ich immer schneller auf, zudem noch die potenzielle Zeit, die wir verloren während des Infonachmittages. Die Arbeitsaufteilung war auch ebenfalls gut und fair, denn jeder hatte immer was zu tun und wusste auch was zu tun.

Tahssin:

Unser Backend-Team, das aus mir und Lukas bestand, hat gut zusammengearbeitet. Lukas hat sich besonders hervorgetan, auch wenn er leider einen Tag krank war. Trotzdem hat er hervorragende Arbeit geleistet und war eine grosse Hilfe.

Die Zusammenarbeit mit dem Frontend-Team war schwieriger. Sie haben langsam gestartet und oft grundlegende Dinge beim Backend nicht verstanden. Zum Beispiel haben sie immer wieder nach einem Docker gefragt. Das machte für uns wenig Sinn, weil ein Docker hier nur ein einfacher Ersatz für eine Datenbank ist. Diese Verwirrung über einfache Backend-Prinzipien führte zu vielen Verzögerungen. Wir mussten viel Zeit damit verbringen, einfache Fehler zu erklären und zu beheben, was zeigte, dass das Frontend-Team Grundlagen vergessen hatte. Das hat unseren Fortschritt sehr gestört. Es fühlte sich so an, als müssten wir nicht nur unsere Aufgaben erledigen, sondern auch ständig die Arbeit des Frontend-Teams korrigieren. Das war frustrierend und hat uns viel extra Arbeit gekostet.