

MODUL 295 – Backend für Applikationen realisieren

Projekt: Ski Service

Inhaltsverzeichnis

1 Vorwort.....	3
2 Informieren.....	3
2.1 Projektumfang	3
2.2 Anforderungsanalyse.....	3
3 Planen	4
3.1 PSP	4
3.2 Datenbankdesign.....	5
3.3 API Endpoints	5
3.4 Zusätzliche Frontend-Arbeit planen	6
4 Entscheiden	6
4.1 DB-First VS Code-First	6
5 Realisieren	7
5.1 Web API Projekt erstellen	7
5.1.1 Frontend integrieren	7
5.2 GitHub Repo erstellen	7
5.3 Installation von Xampp und Konfiguration des MySQL Servers	7
5.4 Frontend fertig entwickeln	7
5.5 Backend entwickeln	8
5.6 Tests entwickeln.....	8
6 Kontrollieren	9
6.1 Postman Tests durchführen	9
6.2 Unit Tests durchführen	9
7 Auswerten	10
7.1 Fazit	10
7.2 Lessons Learned	11

1 Vorwort

Das vorliegende Dokument beschreibt die Entwicklung eines Backend-Systems, das in enger Zusammenarbeit mit einem bereitgestellten Frontend realisiert wurde. Ziel war es, eine robuste Backend-Struktur zu schaffen, die eine sichere Authentifizierung mittels JWT bietet, strukturiert durch zwei dedizierte Controller für API-Endpoints, unterstützt durch eine Vielzahl von Modellen und Data Transfer Objects (DTOs) zur Optimierung des Datenaustausches. Die Implementierung nutzt MySQL als Datenbank und ist vollständig in der GitHub-Repository dokumentiert, inklusive SQL-Statements, einer Postman Collection und dem integrierten Frontend. Das Projekt stellt ein solides Backend dar und erfüllt die gestellten Anforderungen in vollem Umfang.

2 Informieren

2.1 Projektumfang

Das Projekt für Jetstream-Service, ein KMU in der Wintersportbranche, hatte einen beachtlichen Umfang und zielte darauf ab, die Verwaltung von Ski-Serviceaufträgen zu digitalisieren. Das Backend-Teilprojekt beinhaltete die Entwicklung einer Web-API mit Authentifizierung, die Erstellung eines Datenbankdesigns, die Implementierung von Tests und die Realisierung der Applikation gemäß den Spezifikationen.

Die Anforderungen an das Backend umfassten Authentifizierungsmöglichkeiten für Mitarbeiter, die Abfrage von Serviceaufträgen und deren Bearbeitung, inklusive Statusänderungen und Markierung als 'gelöscht' für interne Zwecke, ohne die Daten tatsächlich zu entfernen. Kunden sollten Aufträge mit persönlichen Informationen und gewünschten Dienstleistungen einreichen können.

Das Projekt sah auch die Entwicklung eines Testplans und die Durchführung von Tests mittels Postman sowie die Dokumentation aller Endpunkte in Swagger vor.

2.2 Anforderungsanalyse

Technische Anforderungen:

- Entwicklung eines ASP.NET Web API Projekts
- Erstellung eines Datenbank-Designs in der 3. Normalform (3.NF) mit Beachtung der referenziellen Integrität, implementiert in MySQL

Funktionale Anforderungen:

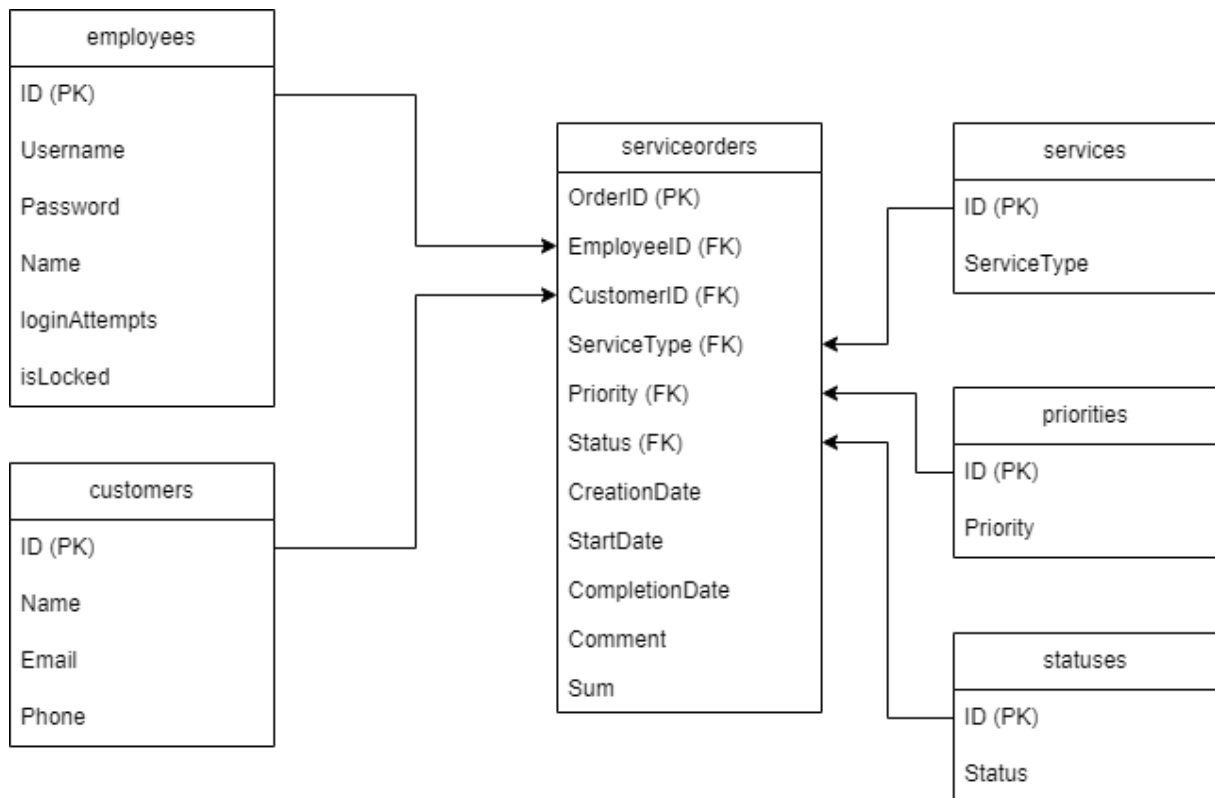
- Implementierung eines Mitarbeiter-Logins
- Mutation (Änderung) von Aufträgen
- Abruf und Anzeige der Serviceaufträge
- Möglichkeiten zum Löschen und Erstellen von Aufträgen
- Logging von wichtigen Aktivitäten und Fehlern in eine lokale Datei mit Serilog
- Filterung von Aufträgen nach Priorität

3 Planen

3.1 PSP

Nr.	Beschreibung	SOLL-Zeit	IST-Zeit
1	Informieren	2.1	2.1
1.1	Ausgangslage analysieren	1	1
1.2	Anforderungen analysieren	1	1
1.3	Plugin Installieren (MySQL)	0.1	0.1
2	Planen	5	6
2.1	Erstellung eines Zeitplans	0.5	1.0
2.2	Datenbankdesign entwerfen	2.5	3
2.3	API Endpoints planen	1	1
2.4	Zusätzliche Frontend-Arbeit planen	1	1
3	Entscheiden	0.2	0.2
3.1	DB-First/Code-First Entscheid treffen	0.2	0.2
4	Realisieren	26.1	37.6
4.1	Web API Projekt erstellen und Frontend integrieren	1	1
4.2	GitHub-Repo erstellen	0.1	0.1
4.3	Installation Xampp und Konfiguration des MySQL Servers	1	1
4.4	Frontend fertig entwickeln	1	2.5
4.5	Backend entwickeln (API Endpoints, Logger usw.)	20	30
4.6	Unit Tests und Postman Tests entwickeln/erarbeiten	3	3
5	Kontrollieren	1	1
5.1	Postman Tests ausführen	0.5	0.5
5.2	Unit Tests ausführen	0.5	0.5
6	Auswerten	3.6	4.6
6.1	Dokumentation nach IPERKA fertigstellen	3	4.6
6.2	Präsentation (Video)	0.5	0.5
6.3	Abgabe des GitHub Repo Links	0.1	0.1
Gesamt		31.8	50.3

3.2 Datenbankdesign



3.3 API Endpoints

GET /api/service:

Abrufen aller Serviceaufträge ohne Authentifizierung.

GET /api/service/priority/{priority}:

Abrufen aller Serviceaufträge, gefiltert nach Priorität.

PATCH /api/service/delete/{orderId}:

Markiert einen Auftrag als gelöscht. Benötigt Authentifizierung.

PATCH /api/service/assign/{orderId}:

Weist einen Auftrag einem Mitarbeiter zu. Benötigt Authentifizierung.

POST /api/auth/login:

Authentifizierung eines Mitarbeiters und Generierung eines JWT.

POST /api/service:

Erstellen eines neuen Serviceauftrags. Benötigt Authentifizierung.

PATCH /api/service/{orderId}:

Aktualisiert den Status eines Serviceauftrags. Benötigt Authentifizierung.

Diese Endpoints decken die Grundfunktionalitäten des Backend-Systems ab, einschließlich Authentifizierung, Datenmanipulation und Abrufen von Informationen.

3.4 Zusätzliche Frontend-Arbeit planen

Das verwendete Frontend wurde von Mahir und Bobby im Frontend-Modul erarbeitet. Netterweise haben sie mir gestattet, dass ich ihr Frontend für dieses Backend-Projekt verwenden darf. Da dies nur für das Frontend-Modul entwickelt wurde, fehlten noch die UI Aspekte, die für das Backend wichtig waren.

Solche Sachen wie den Login-Screen oder das Mitarbeiter-Panel mussten nachträglich noch programmiert werden. Somit habe ich diese Sachen geplant und bereitete mich auf emotionale Schmerzen vor, sobald ich das entwickeln musste.

4 Entscheiden

4.1 DB-First VS Code-First

Ich weiss nicht ob es daran lag, dass ich von Anfang an mich dazu entschieden habe das Projekt mit einer MySQL Datenbank zu machen, anstelle einer MSSQL. Vielleicht dachte ich, beeinflusst von dieser Entscheidung, dass Code-First die bessere Variante wäre.

Ob Code-First besser geeignet gewesen wäre, weiss ich im Endeffekt nicht zu 100%. Dennoch muss ich zugeben, dass DB-First wahrlich eine Herausforderung war. Ganz zu Beginn habe ich zwar gelesen, dass die DB in der 3.NF sein muss, damit sie eine referenzielle Integrität gewährleistet – irgendwie habe ich diesen Punkt jedoch dann ausgeblendet und die Datenbank erstmal ohne diesen Aspekt entwickelt. Dies war, wie sich herausstellte, ein gewaltiger Fehler.

Das nachträgliche Umbauen zu einer DB in der 3.NF gestaltete sich schwierig und erwies sich als sehr zeitaufwendig. Ich habe es irgendwie hinbekommen, aber das war nicht einfach.

5 Realisieren

5.1 Web API Projekt erstellen

Damit ich mit dem Projekt beginnen konnte, musste ich erstmal ein Web API Projekt in Visual Studio erstellen. Ich weiss, es gehört nicht hier her, jedoch muss ich sagen, ich mag diesen Editor einfach nicht besonders.

5.1.1 Frontend integrieren

Nun musste ich das Frontend integrieren. So wie wir es im Unterricht gelernt haben. Den wwwroot Ordner in das Projekt integrieren. In diesem Schritt habe ich auch direkt das Test-Projekt erstellt. Die Solution musste ich natürlich in den Root verschieben, damit das ganze Konstrukt nachher auch noch reibungslos läuft.

5.2 GitHub Repo erstellen

Ein sehr kleiner Punkt, dennoch erwähnenswert. Auf meinem privaten GitHub Account habe ich eine neue Repository erstellt. Dann Git im Projektordner initialisiert, die Repo als Origin hinzugefügt und meinen ersten, initialen Commit getätigt.

Damit konnte es nun endlich richtig losgehen.

5.3 Installation von Xampp und Konfiguration des MySQL Servers

Damit ich bei der Entwicklung das ganze lokal testen konnte, musste ich das alt bekannte Xampp herunterladen, installieren und die MySQL Konfigurationen darin vornehmen.

Beim Start wurde ich direkt von etlichen Errors überrascht. Wie es sich herausgestellt hat, lief im Hintergrund bereits ein anderer MySQL Service (ich habe keine Ahnung welcher und wie der dort hin kam, da kein anderer MySQL Server lief) und dieser blockierte den 3306 Port.

Kurz über den Task Manager diesen Schlingel beendet und dann lief alles wie Butter – wie gewohnt, bei Xampp.

5.4 Frontend fertig entwickeln

Nun war der Zeitpunkt gekommen.. Ich musste das Frontend zu ende entwickeln und Gott, die Kopfschmerzen gingen direkt los.

Naja, Kurzfassung: Ich habe einen Login-Screen entwickelt, welcher nicht direkt in der Navigation oder im Footer verlinkt ist, da dieser nicht so präsent für die gewöhnlichen Nutzer der Webseite sein soll. Diesen erreicht man mit dem Link /login.html.

Dahinter ist dann das Panel (/orders.html). Dort können eingeloggte Mitarbeiter alle Aufträge ansehen, den Status mutieren, sich einem Auftrag zuweisen oder diesen löschen.

5.5 Backend entwickeln

Nun der Hauptpart – das Backend.

Die Backend-Entwicklung für das Skiservice-Projekt war eine anspruchsvolle Aufgabe, die sorgfältige Planung und Durchführung erforderte. Der Prozess begann mit der Einrichtung des ASP.NET Core WebAPI-Projekts, das als Grundlage für die gesamte Anwendung diente. Besonders wichtig war die Einbindung einer MySQL-Datenbank, die durch das Entity Framework Core als ORM-Tool verwaltet wurde. Diese Entscheidung ermöglichte eine effiziente und flexible Handhabung der Daten, wobei das Design der Datenbanktabellen sorgfältig auf die Bedürfnisse des Projekts abgestimmt wurde.

Ein kritischer Aspekt der Backend-Entwicklung war die Implementierung eines sicheren Authentifizierungssystems mittels JWT (JSON Web Tokens). Dies war entscheidend, um den Mitarbeitern einen sicheren und autorisierten Zugriff auf die Serviceaufträge zu ermöglichen. Die Konfiguration und Verwaltung der Token war eine interessante Aufgabe.

Die Erstellung der verschiedenen API-Endpunkte war ein weiterer wichtiger Schritt. Jeder Endpunkt wurde sorgfältig entwickelt, um spezifische Funktionen wie das Anlegen, Abrufen, Ändern und Löschen von Serviceaufträgen sowie das Filtern der Aufträge nach Priorität zu ermöglichen. Hierbei wurde besonderes Augenmerk auf die Business-Logik gelegt, um sicherzustellen, dass die Endpunkte die Anforderungen des Projekts effizient erfüllen.

Um die Stabilität und Zuverlässigkeit der API zu gewährleisten, wurden robuste Fehlerbehandlungssysteme und Logging-Mechanismen implementiert. Das Logging erfolgte in lokale Dateien, was eine wichtige Rolle bei der Fehlerbehebung und der Überwachung der Systemaktivitäten spielte. Die Fähigkeit, Ereignisse zu protokollieren und aufzuzeichnen, war entscheidend für die Wartung und den Support der Anwendung.

5.6 Tests entwickeln

Schließlich wurden umfassende Tests entwickelt, um die Funktionalität und Zuverlässigkeit der API-Endpunkte zu gewährleisten. Hierbei kamen Integrationstests mit Postman zum Einsatz, die es ermöglichten, jeden Aspekt der API gründlich zu überprüfen. Diese Tests zu definieren war von entscheidender Bedeutung, um sicherzustellen, dass das Backend den spezifischen Anforderungen des Projekts gerecht wird und die Endbenutzer eine reibungslose Erfahrung haben.

Zusätzlich zur Entwicklung wurde eine ausführliche Dokumentation der API-Endpunkte mit Swagger/OpenAPI erstellt. Diese Dokumentation bot eine klare und nutzerfreundliche Beschreibung der Schnittstellen, was sowohl für die Entwickler als auch für die Endbenutzer von großem Vorteil war. Die Dokumentation diente nicht nur als Leitfaden für die Verwendung der API, sondern auch als wertvolles Werkzeug für zukünftige Erweiterungen und Wartungsarbeiten.

Insgesamt war die Backend-Entwicklung ein zentraler Bestandteil des Projekts, der eine hohe technische Kompetenz und ein tiefes Verständnis für die Anforderungen des Skiservice-Geschäfts erforderte.

6 Kontrollieren

6.1 Postman Tests durchführen

Ein wesentlicher Bestandteil der Qualitätssicherung des Backend-Projekts waren die umfangreichen Tests mit Postman. Jeder API-Endpunkt wurde rigoros getestet, um sicherzustellen, dass er gemäß den Spezifikationen funktioniert. Diese Tests umfassten das Senden von Anfragen für verschiedene Szenarien, einschließlich Authentifizierung, Abrufen und Mutation von Daten, sowie das Überprüfen der korrekten Reaktionen und Statuscodes des Servers. Fehlerhafte Anfragen wurden ebenfalls getestet, um sicherzustellen, dass das System angemessen reagiert. Durch diese Tests konnte ich die Zuverlässigkeit und Stabilität der API signifikant erhöhen.

6.2 Unit Tests durchführen

Unit Tests dienen dazu, individuelle Teile des Codes isoliert zu testen, was hilft, Fehler frühzeitig zu identifizieren und die Wartung zu vereinfachen. Diese Tests sind besonders wertvoll, da sie bei zukünftigen Änderungen am Code sicherstellen, dass keine bestehenden Funktionen unbeabsichtigt beeinträchtigt werden.

Insgesamt war die Testphase ein entscheidender Schritt im Entwicklungsprozess. Durch sorgfältiges Testen und Überprüfen jeder Komponente konnte ich ein robustes und zuverlässiges Backend-System erstellen, das den Bedürfnissen und Anforderungen des Skiservice-Projekts gerecht wird. Diese Phase hat maßgeblich zur Qualität und Stabilität des fertigen Produkts beigetragen.

7 Auswerten

7.1 Fazit

Dieses Projekt war eine herausfordernde und zugleich aufschlussreiche Erfahrung in meiner beruflichen Laufbahn. Die Auseinandersetzung mit verschiedenen Aspekten der Backend-Entwicklung, insbesondere der Implementierung von Authentifizierungsmethoden wie JWT, hat mein Wissen und meine Fähigkeiten in der Softwareentwicklung wesentlich erweitert. Dabei wurden meine Problemlösungskompetenzen, vor allem im Umgang mit den Schnittstellen zwischen Frontend, Backend und Datenbank, intensiv gefordert und gefördert.

Obwohl ich zu Beginn des Projekts den Umfang und die Komplexität der Aufgabenstellung unterschätzt hatte, bot mir das Projekt die Chance, meine Fähigkeiten in der praktischen Anwendung zu testen und zu vertiefen. Die Überwindung von Schwierigkeiten und die Lösung von Problemen haben nicht nur meine technischen Fähigkeiten, sondern auch mein Durchhaltevermögen und meine Flexibilität gestärkt.

Die intensive Beschäftigung mit dem Entity Framework und die Erweiterung meiner Kenntnisse im Bereich der Unit-Tests waren besonders bereichernd. Ich habe gelernt, wie wichtig gründliche Tests und eine umfassende Dokumentation für die Entwicklung robuster und wartbarer Software sind. Rückblickend betrachtet war das Projekt trotz seines anspruchsvollen Umfangs eine wertvolle Erfahrung. Es hat mir nicht nur ermöglicht, meine fachlichen Kenntnisse zu vertiefen, sondern auch meine persönlichen Kompetenzen weiterzuentwickeln. Ich bin dankbar für die vielfältigen Lernmöglichkeiten, die das Projekt geboten hat, und fühle mich nun besser gerüstet für zukünftige Herausforderungen in der Welt der Softwareentwicklung.

7.2 Lessons Learned

In diesem Projekt habe ich eine Fülle an neuen Kenntnissen und Fähigkeiten erworben, die mein Verständnis und meine Kompetenzen in der Softwareentwicklung nachhaltig geprägt haben. Besonders hervorzuheben ist mein vertieftes Wissen im Bereich der Authentifizierung mit JWT. Diese Technologie war für mich anfangs ein komplexes Feld, doch durch die praktische Anwendung konnte ich ein fundiertes Verständnis entwickeln.

Die Backend-Entwicklung mit C# stellte sich als ein faszinierendes und zugleich herausforderndes Unterfangen dar. Jede Phase des Projekts bot neue Lernmöglichkeiten, sei es bei der Gestaltung der API-Endpunkte, der Datenbankintegration oder der Fehlerbehandlung. Die Erfahrungen, die ich dabei sammeln konnte, waren ungemein wertvoll und haben meine Fähigkeiten als Entwickler gestärkt. Die Zusammenarbeit zwischen Frontend, Backend und Datenbank war eine weitere wichtige Lernerfahrung. Hierbei stieß ich auf Herausforderungen, die nicht nur technisches Know-how erforderten, sondern auch kreatives Problemlösen und eine flexible Herangehensweise. Diese Erfahrungen haben mir gezeigt, wie wichtig eine ganzheitliche Betrachtung von Softwareprojekten ist und wie entscheidend die nahtlose Integration aller Komponenten für den Erfolg eines Projekts ist. Ein weiterer zentraler Aspekt meiner Lernkurve war das vertiefte Verständnis für das Entity Framework. Die Arbeit mit diesem Framework hat mir ermöglicht, effizienter mit Datenbanken zu interagieren und komplexe Abfragen zu meistern.

Besonders bereichernd war mein Lernfortschritt im Bereich der Softwaretests. Das Erstellen und Durchführen von Unit Tests sowie das Testen mit Postman waren für mich zunächst Neuland. Diese Erfahrungen haben mir jedoch gezeigt, wie essenziell gründliches Testen für die Entwicklung robuster und zuverlässiger Anwendungen ist.

Insgesamt betrachtet, war dieses Projekt eine enorm bereichernde Erfahrung. Ich habe nicht nur mein technisches Wissen erweitert, sondern auch wichtige Erkenntnisse in Bezug auf Projektmanagement und Teamarbeit gewonnen. Diese Erfahrungen werden mir in meiner weiteren beruflichen Laufbahn von großem Nutzen sein und haben meine Leidenschaft für die Softwareentwicklung weiter entfacht.