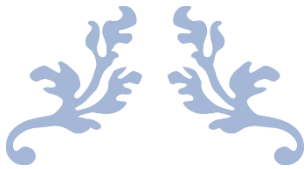

PROJEKTARBEIT MODUL 295

Aimo Altorfer

Inhaltsverzeichnis

Einleitung ... 3
1.1 Projektübersicht ... 3
Kurzbeschreibung: ... 3
1.2 Zielsetzung ... 3
Hauptziel: ... 3
Teilziele: ... 3
1.3 Projektumfang ... 3
Backend-Entwicklung: ... 3
Test und Dokumentation: ... 3
Zusätzliche Merkmale: ... 3
Informieren ... 4
2.1 Ausgangslage und Kontext ... 4
2.2 Anforderungsanalyse ... 4
Mitarbeiter-Login: ... 4
Auftragsanzeige: ... 4
Auftragsmutation: ... 4
Auftragslöschung: ... 4
Erweiterung der Online-Anmeldung: ... 4
Planung ... 5
3.1 Projektplan ... 5
Schritt 1 – Verständnis und Konzeptentwicklung: ... 5
Schritt 2 – Technische Planung: ... 5
Entscheiden ... 6
4.1 Tool Wahl ... 6
4.2 Begründung der getroffenen Entscheidungen ... 6
Realisieren ... 8
5.1 Umsetzungsstrategie ... 8
Modularer Ansatz: ... 8
Iterative Entwicklung: ... 8
5.2 Entwicklungsumgebung und Werkzeuge ... 8
5.3 Implementierungsphasen ... 8
5.4 Testing und Qualitätssicherung ... 9
6. Kontrollieren ... 10
6.1 Monitoring ... 10
7.1 Projektsteuerung ... 10

Auswerten 11
7.1 Projektabschlussbericht..... 11
7.2 Lessons Learned 11
Database-First-Ansatz:..... 11
Wiederholung und Vertiefung von Vorwissen: 11
Praktische Anwendung von Theorien:..... 11
Bedeutung der Flexibilität: 11

Einleitung

1.1 Projektübersicht

Kurzbeschreibung:

Das Projekt zielt auf die Entwicklung eines Backend-Systems für das Ski-Service Management. Es soll die Verwaltung von Serviceaufträgen für die Firma Jetstream-Service digitalisieren und optimieren.

1.2 Zielsetzung

Hauptziel:

Schaffung einer effizienten, webbasierten Lösung zur Verwaltung von Ski-Service-Aufträgen, um die internen Abläufe des Unternehmens zu verbessern.

Teilziele:

- Entwicklung einer sicheren und benutzerfreundlichen Web-API.
- Implementierung eines robusten Datenbanksystems für die Speicherung und Verwaltung von Auftragsdaten.
- Bereitstellung eines Systems zur Statusänderung und Verfolgung von Serviceaufträgen.
- Integration einer Authentifizierungsfunktion zur Sicherstellung des autorisierten Zugriffs.

1.3 Projektumfang

Backend-Entwicklung:

- Konzeption und Entwicklung der Backend-Logik.
- Gestaltung des Datenbankdesigns unter Einsatz von Entwurfsmustern.
- Einrichtung der Authentifizierung für Mitarbeiter.

Test und Dokumentation:

- Entwicklung von Unit-Tests zur Überprüfung der Funktionalitäten.
- Nutzung von Postman für die API-Tests.
- Erstellung einer umfassenden Dokumentation nach OpenAPI-Standard.

Zusätzliche Merkmale:

- Einbindung von Funktionalitäten für die Aktualisierung und Löschung von Serviceaufträgen.
- Bereitstellung einer nutzerfreundlichen Schnittstelle für Mitarbeiter zur Auftragsbearbeitung.

Informieren

2.1 Ausgangslage und Kontext

Die Firma Jetstream-Service, ein mittelständisches Unternehmen im Bereich Wintersport, steht vor der Herausforderung, ihre Skiservice-Prozesse zu optimieren. Bisher erfolgte die Verwaltung von Skiservice-Aufträgen manuell und zeitaufwendig, was besonders in der Hauptsaison zu Engpässen führte. Die Notwendigkeit einer digitalen Lösung zur Effizienzsteigerung und besseren Auftragsverwaltung ist offensichtlich. Die Integration einer web- und datenbankbasierten Anwendung soll die interne Verwaltung revolutionieren, indem sie eine schnelle und effiziente Bearbeitung von Serviceaufträgen ermöglicht. Dies wird nicht nur die Arbeitsprozesse erleichtern, sondern auch die Kundenzufriedenheit durch schnellere Servicezeiten erhöhen.

2.2 Anforderungsanalyse

Die Anforderungen für das Ski-Service Management-System wurden in Zusammenarbeit mit den Mitarbeitern und dem Management von Jetstream-Service definiert. Das System muss in der Lage sein:

Mitarbeiter-Login: Einen sicheren, passwortgeschützten Zugang für Mitarbeiter bieten, um Auftragsdaten einzusehen und zu bearbeiten.

Auftragsanzeige: Eine Liste aller anstehenden Serviceaufträge anzeigen, um Mitarbeitern einen schnellen Überblick über ihre Arbeitslast zu geben.

Auftragsmutation: Mitarbeitern ermöglichen, Änderungen an bestehenden Aufträgen vorzunehmen. Dies umfasst die Aktualisierung des Status von 'Offen' über 'In Arbeit' bis 'Abgeschlossen'.

Auftragslöschung: Die Option, Aufträge bei Bedarf zu löschen, insbesondere im Falle einer Stornierung.

Erweiterung der Online-Anmeldung: Ergänzung der bestehenden Online-Anmeldung um zusätzliche Kundeninformationen wie Name, E-Mail, Telefonnummer, Priorität des Auftrags und ausgewählte Dienstleistung.

Die Implementierung dieser Anforderungen soll die Effizienz steigern und eine solide Basis für die zukünftige Skalierung und Erweiterung des Unternehmens bilden.

Planung

3.1 Projektplan

Das Projekt "Ski-Service Management" ist eine Einzelarbeit, die in mehreren Schritten umgesetzt wird, um ein Backend-System für Skiservice-Aufträge zu entwickeln.

Schritt 1 – Verständnis und Konzeptentwicklung:

Zu Beginn des Projekts wird ein klares Verständnis der grundlegenden Anforderungen und Funktionalitäten des Systems entwickelt. In diesem Schritt wird ein einfaches Konzept für die Anwendung entworfen.

Schritt 2 – Technische Planung:

Auswahl der zu verwendenden Technologien und Tools, einschliesslich Programmiersprache, Frameworks und Datenbank. In diesem Schritt wird auch die Architektur der Anwendung festgelegt.

Schritt 3 – Implementierung:

Entwicklung des Backends, einschliesslich der Erstellung von API-Endpunkten, Datenbankintegration und Authentifizierung.

Schritt 4 – Testen:

Das entwickelte System wird durch einfache requests mit Postman überprüft, um sicherzustellen, dass alle Funktionen wie erwartet arbeiten.

Schritt 5 – Überprüfung und Anpassung:

Eventuelle Anpassungen und Verbesserungen basierend auf den Testergebnissen werden vorgenommen, um die Funktionalität und Leistung zu optimieren.

Entscheiden

4.1 Tool Wahl

Für das Projekt "Ski-Service Management" wurden folgende Tools und Technologien gewählt:

API Entwicklung: C# mit Visual Studio

Frontend Entwicklung: JavaScript, HTML, CSS mit Visual Studio Code

Datenbankmanagement: Microsoft SQL Server

API Testing: Postman

4.2 Begründung der getroffenen Entscheidungen

Die Entscheidung für die genannten Tools und Technologien basiert auf ihrer Verbreitung im Unterricht und ihrer Eignung für die Projektanforderungen:

C# mit Visual Studio für die API:

Vertrautheit: C# ist eine bekannte Sprache im Unterricht, was die Entwicklung effizienter macht.

Stärke von C#: C# ist eine robuste, objektorientierte Sprache, ideal für die Entwicklung komplexer Backend-Systeme.

Integration mit .NET: Die Nutzung des .NET-Frameworks für die Backend-Entwicklung bietet eine starke Unterstützung für Web-APIs.

JavaScript, HTML, CSS mit Visual Studio Code für das Frontend:

Standard-Webtechnologien: Diese Technologien sind der Standard für die Webentwicklung und ermöglichen die Erstellung eines responsiven und benutzerfreundlichen Frontends.

Flexibilität: JavaScript bietet Flexibilität und ist weit verbreitet, was die Integration mit verschiedenen Backend-Systemen erleichtert.

Microsoft SQL Server für die Datenbank:

Zuverlässigkeit: SQL Server ist ein bewährtes, leistungsfähiges Datenbanksystem, das sich für komplexe Datenmanagement-Aufgaben eignet.

Integration: Nahtlose Integration mit C# und dem .NET-Framework.

Postman für API Testing:

Benutzerfreundlichkeit: Postman ist ein intuitives Tool zum Testen von Web-APIs, das es ermöglicht, Anfragen einfach zu konfigurieren und Antworten zu analysieren.

Vielseitigkeit: Unterstützung für verschiedene HTTP-Anfragen und die Möglichkeit, Tests zu automatisieren.

Diese Auswahl an Tools und Technologien stellt sicher, dass die Entwicklung effizient und effektiv durchgeführt werden kann, während gleichzeitig eine solide Basis für das Ski-Service Management-System geschaffen wird.

Realisieren

5.1 Umsetzungsstrategie

Die Umsetzungsstrategie für das Ski-Service Management-System konzentriert sich auf eine schrittweise, modulare Entwicklung, die es ermöglicht, einzelne Komponenten unabhängig voneinander zu entwickeln und zu testen.

Modularer Ansatz:

Jeder Teil der Anwendung (API, Frontend, Datenbank) wird separat entwickelt, um die Komplexität zu reduzieren und die Wartbarkeit zu verbessern.

Iterative Entwicklung:

Die Entwicklung erfolgt in iterativen Zyklen, wobei jede Iteration spezifische Funktionen oder Verbesserungen umfasst. Dies ermöglicht eine kontinuierliche Überprüfung und Anpassung.

5.2 Entwicklungsumgebung und Werkzeuge

Visual Studio: Hauptentwicklungsumgebung für die API in C#.

Visual Studio Code: Für die Frontend-Entwicklung mit JavaScript, HTML, CSS.

Microsoft SQL Server: Datenbankmanagement und -design.

Postman: Zum Testen und Validieren der API.

5.3 Implementierungsphasen

Frontend-Entwicklung:

Da das Frontend bereits in einem vorherigem Modul zusammen mit Ilia implementiert wurde wird dieses als erstes ergänzt.

Datenbank-Design und -Einrichtung:

Aufbau der Datenbankstrukturen.

API-Entwicklung:

Erstellung der Endpunkte und Logik für das Backend.

Integration:

Zusammenführen von Frontend und Backend zu einem funktionierenden System.

5.4 Testing und Qualitätssicherung

End-to-End Testing:

Verwendung von Postman zur Simulation von API-Anfragen und Überprüfung der Integration zwischen Frontend und Backend.

5.5 Dokumentation der Entwicklungsarbeit

Code-Kommentare:

Klare Kommentare und Beschreibungen im Quellcode.

Entwicklungsdokumentation:

Erstellung einer Dokumentation, die den Entwicklungsprozess, die Entscheidungen und die Architektur der Anwendung erläutert.

Diese Realisierungsphase zielt darauf ab, ein robustes, gut getestetes und gut dokumentiertes System zu schaffen, das die Anforderungen von Jetstream-Service effektiv erfüllt.

6. Kontrollieren

6.1 Monitoring

Das Monitoring des Projekts basierte auf der direkten Beobachtung und Überwachung während der Entwicklungsphase:

Browser-Konsole: Regelmäßige Überprüfung der Browser-Konsole auf JavaScript-Fehler oder Warnungen während der Frontend-Entwicklung.

Visual Studio Fehlermeldungen: Aufmerksame Beachtung von Fehlermeldungen und Warnungen in Visual Studio während der Backend-Entwicklung, um Probleme frühzeitig zu identifizieren und zu beheben.

Manuelles Testen: Durchführen manueller Tests, um das Verhalten der Anwendung zu überprüfen und sicherzustellen, dass alle Funktionen wie erwartet arbeiten.

Diese Ansätze ermöglichen eine einfache und effektive Überwachung des Projektfortschritts und der Systemstabilität ohne den Einsatz komplexer Monitoring-Tools.

7.1 Projektsteuerung

Die Projektsteuerung für das Ski-Service Management-Projekt konzentrierte sich auf einfache, aber effektive Methoden, um den Fortschritt zu überwachen und sicherzustellen, dass die Projektziele erreicht werden:

Fortlaufende Selbstüberprüfung: Regelmäßige Überprüfung des Fortschritts im Hinblick auf die gesetzten Ziele und Anpassung der Planung bei Bedarf.

Zeitmanagement: Verwendung eines einfachen Zeitplans oder Kalenders, um Deadlines zu setzen und den Fortschritt im Auge zu behalten. Dies half, den Überblick über die verschiedenen Phasen des Projekts zu behalten.

Anpassungsfähigkeit: Flexibilität bei der Anpassung des Projektplans auf Basis von Herausforderungen oder unerwarteten Problemen, die während der Entwicklung auftraten.

Einfache Problembehandlung: Bei auftretenden Problemen wurden diese direkt und ohne Umwege angegangen, um Verzögerungen im Projektverlauf zu vermeiden.

Auswerten

7.1 Projektabschlussbericht

Im gross und ganzen sehe ich das Projekt als einen Erfolg, auch wenn ein- zwei Dinge fehlen.

Durch ein Unterschätzten des Projektumfanges habe ich mich mit der Zeitplanung definitiv etwas vertan, was zu Lücken in meinem Projekt führe. Allerdings funktioniert das System weil alle wichtigen Punkte umgesetzt wurden. Auch bei Punkten die nun schlussendlich fehlen habe ich das Projekt so erbaut als währen diese Punkte voll implementiert. Beispielsweise auch wenn die Authentifizierung nicht vollständig ist habe im Frontend das Login Fenster nicht weggelassen, oder die Tabelle in der Datenbank für die Mitarbeiter.

7.2 Lessons Learned

In diesem umfangreichen Projekt habe ich eine Vielzahl von wertvollen Erfahrungen gesammelt, von denen ich nicht alle einzeln aufzählen kann. Einige der wichtigsten Erkenntnisse sind jedoch besonders hervorzuheben:

Database-First-Ansatz:

Die Entscheidung für den Database-First-Ansatz erwies sich als weniger effizient als erwartet. Es kostete zusätzliche Zeit, vor allem beim Anpassen der Datenbank an die sich ändernden Anforderungen der Anwendung. Diese Erfahrung hat gezeigt, dass ein flexiblerer Ansatz, wie Code-First, in einigen Fällen vorteilhafter sein könnte.

Wiederholung und Vertiefung von Vorwissen:

Das Projekt bot eine hervorragende Gelegenheit, die in vorherigen Modulen erworbenen Kenntnisse anzuwenden und zu vertiefen. Insbesondere konnte ich mein Wissen im Bereich SQL und der Nutzung von Fetch für HTTP-Anfragen in JavaScript praktisch einsetzen und festigen.

Praktische Anwendung von Theorien:

Die direkte Umsetzung theoretischer Konzepte in die Praxis war eine bereichernde Erfahrung. Durch das eigenständige Arbeiten an diesem Projekt konnte ich ein besseres Verständnis für die praktische Anwendung von Programmier- und Datenbankkenntnissen entwickeln.

Bedeutung der Flexibilität:

Die Notwendigkeit, flexibel auf Herausforderungen und unerwartete Probleme zu reagieren, war eine der wichtigsten Lektionen. Diese Flexibilität ermöglichte es mir, effektiv auf Veränderungen zu reagieren und das Projekt erfolgreich voranzutreiben.

Insgesamt war dieses Projekt eine umfassende Lernerfahrung, die mir nicht nur vertiefte technische Kenntnisse, sondern auch wertvolle Einblicke in die Projektarbeit und -steuerung vermittelt hat.