



# Projektarbeit Ski-Service

## Auftragsverwaltung

By Arda D. | Modul 165

02.02.2024

# Inhaltsverzeichnis

01

Einleitung

02

Technologieüberblick

03

Anforderungsüberblick  
und Risikoanalyse

04

Planungsphase +  
Realisierung

05

Live Demo

06

Projektbewertung +  
Fazit

# Einleitung

---

# Einleitung

## Die Aufgabe>

Die Aufgabe war es anhand der vorherigen MySQL Datenbankmodel eine NoSQL Datenbank zu erstellen.

### Zielgruppe:

- Skiverleihgeschäfte
- Servicezentren
- IBZ

### Zweck:

- Datenverteilung und Skalierung
- Lizenzkosten einsparen
- Effizienz

# SQL

Employees	
Id	
Username	
Password	
IsLocked	
FailedLoginAttempts	

ServiceOrders	
Id	
CustomerName	
Email	
PhoneNumber	
Priority	
ServiceTypeId	
CreationDate	
PickupDate	
Comments	
Status	

ServiceTypes	
Id	
Name	
Cost	

# Technologie- überblick

---

# Genutzte Tools



mongoDB



autoMapper



# Anforderungsüberblick und Risikoanalyse

---



# Zusätzliche Anforderungen

Nr.	Beschreibung
AO1	Automatisiertes Backup-Konzept durchgeführt u. implementiert.
AO2	Komplexe Schema Validierungen umgesetzt (Referenzen, enum, min, max. usw)
AO3	Datenmigrationsskripte zu den RDBMS nach NoSQL realisiert
AO4	Komplexes Datenmodell mit mehr als 6 Grundtypen (Collection / Labels) implementiert
AO5	Komplexe statistische Auswertungsabfragen realisiert





# Risikoanalyse

## Risikoanalyse NoSQL Ski Service Manager

Risikotyp	Nr.	Wahr- sch.	Aus- wirk.	Ampel	Beschreibung	Behandlung und Kontrolle
<b>Risikomanagement NoSQL Projekt</b>						
Datenverlust	1	3	4	12	Kann durch unzureichende Backup-Verfahren oder Fehler beim Handling der Datenbank entstehen. Schaden wäre der Verlust von wertvollen Daten und Zeit für die Wiederherstellung oder Rekonstruktion.	Regelmässige Backups, Datenvalidierung
Sicherheitsrisiken	2	2	4	8	Mögliche Verletzungen der Datensicherheit durch Hacking oder interne Leaks.	Sicherheitsprotokolle, Zugriffsrechteverwaltung
Technische Defekte	3	2	3	6	Risiko von Ausfällen durch Hardware- oder Softwarefehler.	Redundanzplan, regelmässige Wartung
Performance-Probleme	4	3	3	9	Risiko von Leistungsengpässen, die Benutzererfahrungen beeinträchtigen können.	Performance-Monitoring, Skalierungsoptionen
Fehlende Dokumentation	5	3	2	6	Das Risiko, dass mangelnde Dokumentation zu Verwirrung und Fehlern führt.	Dokumentationsrichtlinien, regelmässige Reviews
Projektverzögerungen	6	3	3	9	Risiken, die zu Verzögerungen im Projektzeitplan führen.	Zeitmanagement, Meilenstein-Planung
Veraltete Technologie	7	1	2	2	Die Gefahr, dass veraltete Technologien die Effizienz und Sicherheit beeinträchtigen	Technologie-Review-Zyklen, Weiterbildung
Unzureichendes Testing	8	3	3	9	Das Risiko, dass durch unzureichende Tests Fehler in der Produktion auftreten.	Umfassende Testpläne, automatisierte Tests
Anforderungsänderungen	9	0	2	0	Risiko von Projektkomplikationen durch sich ändernde Anforderungen.	Change-Management-Prozesse, agile Methodik
Kommunikationsprobleme	10	0	3	0	Risiko von Missverständnissen und Fehlern aufgrund schlechter Kommunikation.	Klare Kommunikationskanäle, regelmässige Meetings (Trello, Miro usw.)

# Planungsphase + Realisierung

---

# Planungsphase mit WBSTool>



## PSP

Es wurde ein  
Projektstrukturplan erstellt



## GANTT

Der Zeitablauf wurde  
visuell nach  
Arbeitspaketen dargestellt



## Kostenübersicht (Soll / Ist)

Es wurde davon  
ausgegangen das jede  
Stunde 80CHF kosten würde  
die der Mitarbeiter bräuchte

# Planungsphase mit WBSTool>



## NoSQL Ski Service Manager

### 1 Information

- 1.1 Studium des Auftrags und Festlegung der Anforderungen
- 1.2 Auswahl des Technologiestacks und der Tools
- 1.3 Erstellung der Projektumgebung
- 1.4 Anforderungen definiert und Tools eingerichtet.

### 2 Planung

- 2.1 Detaillierte Projektplanung und Erstellung des Zeitplans
- 2.2 Festlegung der Meilensteine und Deadlines
- 2.3 Definition der Datenbank- und Backend-Architektur
- 2.4 Auswahl der notwendigen Tools und Bibliotheken
- 2.5 Festlegen der Backup-Strategie und Sicherheitskonzepte
- 2.6 Projektplan und Architektur festgelegt

### 3 Entscheidung

- 3.1 Entscheidungen über die Authentifizierungsmethoden und das Datenmodell
- 3.2 Abschluss der Architekturentscheidung
- 3.3 Abschluss der Datenmodellentscheidung
- 3.4 Technologische Entscheidungen getroffen

### 4 Realisierung

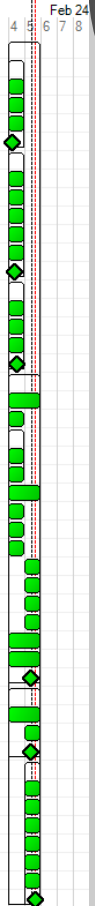
- 4.1 Projektdokumentation erstellen und führen
- 4.2 Initial Commit und Setup der Projektstruktur
- 4.3 Implementierung der Authentifizierung und des Logins
  - 4.3.1 Implementierung, Beginn
  - 4.3.2 Implementierung, Fortsetzung
- 4.4 Allgemeine Fehlerbehebung
- 4.5 Login-Funktionalität fertigstellen
- 4.6 Backup-Prozedur implementieren, die alle 24 Stunden läuft
- 4.7 Erstellung und Anpassung der Postman Tests
- 4.8 Implementierung von Enums für Statuswerte
- 4.9 Skripte für Backup und Postman hinzufügen
- 4.10 Schemas hinzufügen
- 4.11 README anlegen und wichtige Informationen festhalten
- 4.12 Code Dokumentation
- 4.13 Powerpoint erstellen und Inhalt auswendig lernen
- 4.14 Kernfunktionalitäten implementiert und getestet.

### 5 Kontrolle

- 5.1 Überprüfung der Funktionalität (Nach jedem Commit)
- 5.2 Überprüfung und Anpassung der Dokumentation
- 5.3 Code und Dokumentation finalisiert

### 6 Auswertung

- 6.1 Dokumentation der Lessons Learned
- 6.2 Finalisierung der Projektdokumentation
- 6.3 Durchführung der Präsentation vor dem Kurs
- 6.4 Bewertung vom Erfolg des Projekts
- 6.5 Abgabe der finalen Präsentation und Dokumentation im Git-Repo
- 6.6 Übermittlung des GitHub-Links via OneNote
- 6.7 Projekt erfolgreich beendet



# Planungsphase mit WBSTool >

	PSP-Nr.	Beschreibung	Verantwortlich	Plan Arbeit h	Ist-Arbeit bisher h	Rest-Arbeit h	Ausblick Arbeit h	Abw. %	Plan ext. Kosten CHF	Plan Gesamtkosten CHF
▶		<b>NoSQL Ski Service Manager</b>		<b>70.25</b>	<b>82.50</b>	<b>0.00</b>	<b>82.50</b>	<b>+17%</b>	<b>0</b>	<b>5'620</b>
	<b>1</b>	<b>Information</b>		<b>3.00</b>	<b>3.50</b>	<b>0.00</b>	<b>3.50</b>	<b>+17%</b>	<b>0</b>	<b>240</b>
	1.1	Studium des Auftrags und Festlegung der Anforderungen		1.00	1.00	0.00	1.00	+0%	0	80
	1.2	Auswahl des Technologiestacks und der Tools		1.00	1.00	0.00	1.00	+0%	0	80
	1.3	Erstellung der Projektumgebung		1.00	1.50	0.00	1.50	+50%	0	80
	1.4	Anforderungen definiert und Tools eingerichtet.		0.00	0.00	0.00	0.00	+0%	0	0
	<b>2</b>	<b>Planung</b>		<b>8.50</b>	<b>8.75</b>	<b>0.00</b>	<b>8.75</b>	<b>+3%</b>	<b>0</b>	<b>680</b>
	2.1	Detaillierte Projektplanung und Erstellung des Zeitplans		1.00	3.00	0.00	3.00	+200%	0	80
	2.2	Festlegung der Meilensteine und Deadlines		1.00	1.00	0.00	1.00	+0%	0	80
	2.3	Definition der Datenbank- und Backend-Architektur		2.00	3.00	0.00	3.00	+50%	0	160
	2.4	Auswahl der notwendigen Tools und Bibliotheken		2.50	0.75	0.00	0.75	-70%	0	200
	2.5	Festlegen der Backup-Strategie und Sicherheitskonzepte		2.00	1.00	0.00	1.00	-50%	0	160
	2.6	Projektplan und Architektur festgelegt		0.00	0.00	0.00	0.00	+0%	0	0
	<b>3</b>	<b>Entscheidung</b>		<b>4.00</b>	<b>4.50</b>	<b>0.00</b>	<b>4.50</b>	<b>+13%</b>	<b>0</b>	<b>320</b>
	3.1	Entscheidungen über die Authentifizierungsmethoden und das Datenmodell		2.00	2.00	0.00	2.00	+0%	0	160
	3.2	Abschluss der Architekturentscheidung		1.00	0.50	0.00	0.50	-50%	0	80
	3.3	Abschluss der Datenmodellentscheidung		1.00	2.00	0.00	2.00	+100%	0	80
	3.4	Technologische Entscheidungen getroffen		0.00	0.00	0.00	0.00	+0%	0	0
	<b>4</b>	<b>Realisierung</b>		<b>37.50</b>	<b>46.25</b>	<b>0.00</b>	<b>46.25</b>	<b>+23%</b>	<b>0</b>	<b>3'000</b>
	4.1	Projektdokumentation erstellen und führen		8.00	7.00	0.00	7.00	-13%	0	640
	4.2	Initial Commit und Setup der Projektstruktur		2.00	3.00	0.00	3.00	+50%	0	160
	4.3	Implementierung der Authentifizierung und des Logins		3.00	6.00	0.00	6.00	+100%	0	240
	4.3.1	Implementierung, Beginn		1.00	3.00	0.00	3.00	+200%	0	80
	4.3.2	Implementierung, Fortsetzung		2.00	3.00	0.00	3.00	+50%	0	160
	4.4	Allgemeine Fehlerbehebung		5.00	9.00	0.00	9.00	+80%	0	400
	4.5	Login-Funktionalität fertigstellen		5.00	5.00	0.00	5.00	+0%	0	400
	4.6	Backup-Prozedur implementieren, die alle 24 Stunden läuft		1.00	1.00	0.00	1.00	+0%	0	80
	4.7	Erstellung und Anpassung der Postman Tests		1.00	1.00	0.00	1.00	+0%	0	80
	4.8	Implementierung von Enums für Statuswerte		2.00	2.00	0.00	2.00	+0%	0	160
	4.9	Skripte für Backup und Postman hinzufügen		1.00	1.00	0.00	1.00	+0%	0	80
	4.10	Schemas hinzufügen		3.00	5.00	0.00	5.00	+67%	0	240
	4.11	README anlegen und wichtige Informationen festhalten		3.00	2.75	0.00	2.75	-8%	0	240
	4.12	Code Dokumentation		2.50	2.50	0.00	2.50	+0%	0	200
	4.13	Powerpoint erstellen und Inhalt auswendig lernen		1.00	1.00	0.00	1.00	+0%	0	80
	4.14	Kernfunktionalitäten implementiert und getestet.		0.00	0.00	0.00	0.00	+0%	0	0
	<b>5</b>	<b>Kontrolle</b>		<b>12.00</b>	<b>13.00</b>	<b>0.00</b>	<b>13.00</b>	<b>+8%</b>	<b>0</b>	<b>960</b>
	5.1	Überprüfung der Funktionalität (Nach jedem Commit)		10.00	9.00	0.00	9.00	-10%	0	800
	5.2	Überprüfung und Anpassung der Dokumentation		2.00	4.00	0.00	4.00	+100%	0	160
	5.3	Code und Dokumentation finalisiert		0.00	0.00	0.00	0.00	+0%	0	0
	<b>6</b>	<b>Auswertung</b>		<b>5.25</b>	<b>6.50</b>	<b>0.00</b>	<b>6.50</b>	<b>+24%</b>	<b>0</b>	<b>420</b>
	6.1	Dokumentation der Lessons Learned		1.00	1.50	0.00	1.50	+50%	0	80
	6.2	Finalisierung der Projektdokumentation		2.00	3.00	0.00	3.00	+50%	0	160
	6.3	Durchführung der Präsentation vor dem Kurs		0.50	0.50	0.00	0.50	+0%	0	40
	6.4	Bewertung vom Erfolg des Projekts		1.00	1.00	0.00	1.00	+0%	0	80
	6.5	Abgabe der finalen Präsentation und Dokumentation in Git-Repo		0.50	0.25	0.00	0.25	-50%	0	40
	6.6	Übermittlung des GitHub-Links via OneNote		0.25	0.25	0.00	0.25	+0%	0	20
	6.7	Projekt erfolgreich beendet		0.00	0.00	0.00	0.00	+0%	0	0

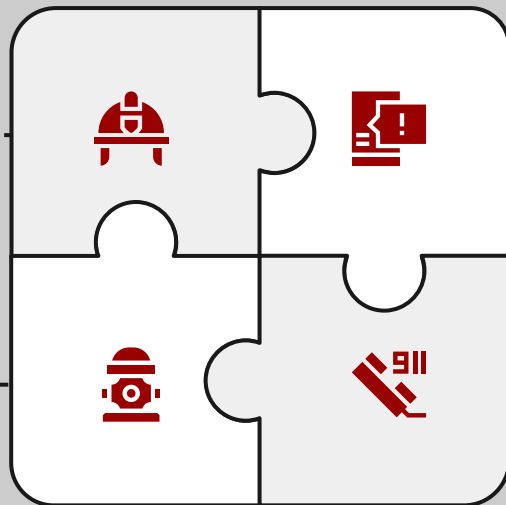
# Realisierung>

## Generische Klassen

Es wurden generische Controller und Services erstellt, um Code Redundanz zu verhindern

## Data Seeding

Beim ersten Mal, starten des Backends werden die Schemas, Indexes und Daten automatisch gesetzt



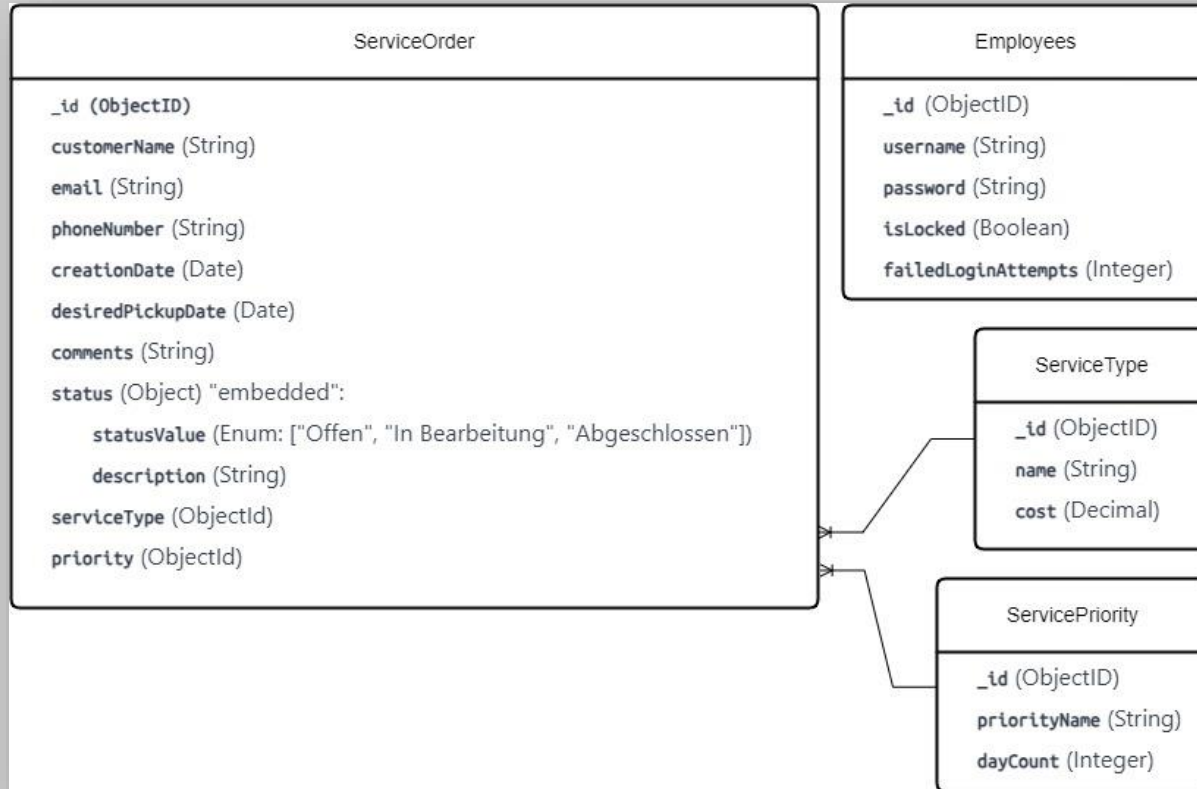
## Automapper

Darüber hinaus wurde Automapper eingesetzt, um auch viel Code zu sparen

## Middleware

Es wurde auch eine Exception-Middleware erstellt, die verschiedene Fehler aufhängt und somit Crashes verhindert

# Datenbankmodell >

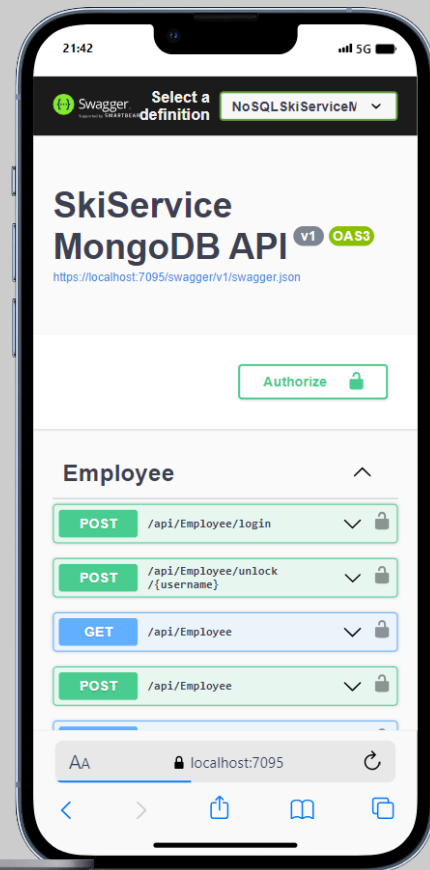
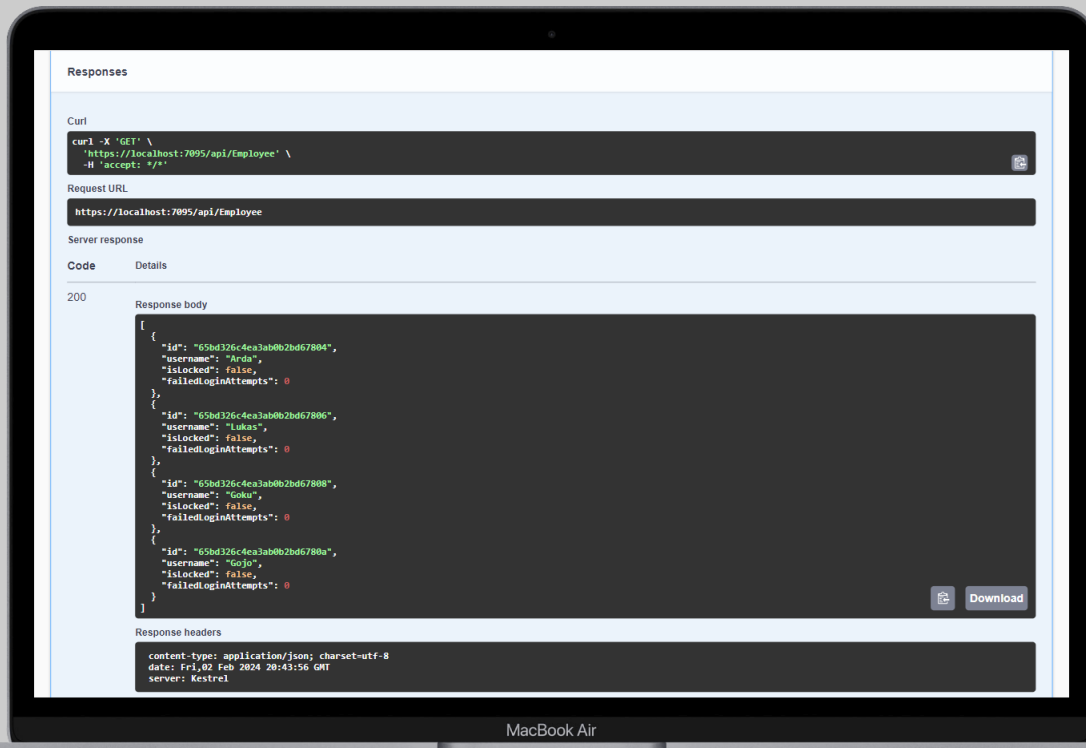


# Live Demo

---



# Live Demonstration



# Projektbewertung + **Fazit**

---

# Projekt **Bewertung**>



## **Erfolg / Misserfolg?**

Meiner Meinung nach  
war das ein guter  
Erfolg, hat aber noch  
Luft nach oben.



## **Verbesserungen?**

Performance und Sicherheit  
wird in Zukunft noch verbessert



## **Soll / Ist eingehalten?**

Leider konnten die 70h  
nicht ganz eingehalten  
werden

A decorative graphic of stylized clouds in the bottom-left corner, rendered in shades of gray and white.

# Fazit

---

# Fazit >



## Meine Meinung

- Einer meiner Erfolgreichsten Projekten bisher
- Hat wirklich spass gemacht!
- Aus Interesse neues zu lernen mit generischen Klassen gearbeitet



## Danke fürs zuhören!

Falls ihr Fragen habt, könnt ihr  
sie gerne stellen