Hotelreservirungssystem Architekturspezifikation

Umut Yalçın – 200503021

Ali Cemal Deniz – 200503036

Kapitel 1 ……………..Architekturüberblick

Kapitel 2 ……………..API Kontrolleklassen und Rückverfolgbarkeit der Anforderungen

Kapitel 3 ……………..DB-Zugriffschicht

Kapitel 4 ……………..DB-Entitäten

**Kapitel 1 – Architekturüberblick**

**Systemarchitektur:**

Request (GET, POST, PUT, DELETE)

Client

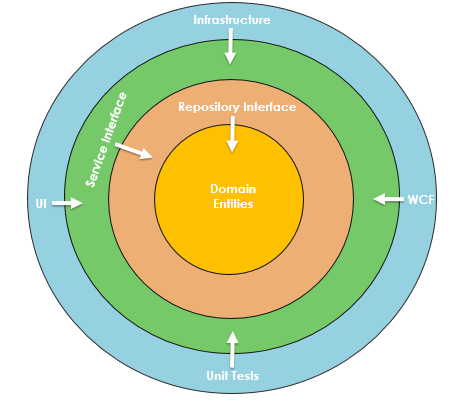
(React JS App)

Web Service

(ASP.NET Core Web API)

Response (JSON)

**Onionarchitektur:** Bei Web-API Service wird Onionarchitektur verwendet.



Schichte von Onionarchitektur

Onionarchitekture wird auf “Umkehrung der Kontroller” Prinzip basiert. Es löst “tight coupling” und “separation of concerns” Probleme. Jede Schichte werden durch “Service Interface” Schicht abstrahiert. Diese Schichte sind flexibel, ersetzbar, testbar und zeverlässig.

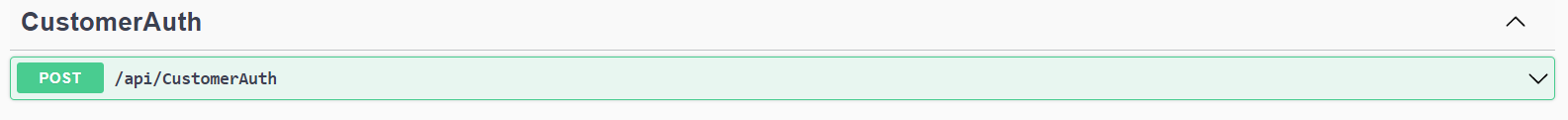
**Kapitel 2 – API Kontrollerklassen und Rückverfolgbarkeit der Anforderungen**

**Customer Controller:** Listen, Addieren, Löschen, Aktualieseren von Kunden

**A picture containing text, line, font, number

Description automatically generated**

**CustomerAuthController: „**Authentication“ ve „Authorization“ von Kunden

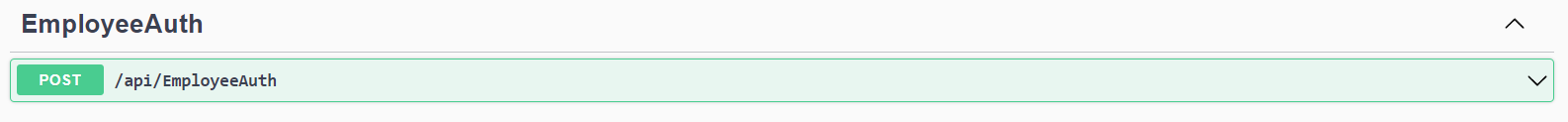
****

**EmployeeController:** Listen, Addieren, Löschen, Aktualieseren von Mitearbeitern

**A picture containing text, line, font, number

Description automatically generated**

**EmployeeAuthController: „**Authentication“ ve „Authorization“ von Mitearbeitern

****

**FloorController: „**Authentication“ ve „Authorization“ von Boden

**A picture containing text, screenshot, line, font

Description automatically generated**

**RezervationController:** Rezervationen kontrollieren, rezervieren, Rezervierung löshen und aktualisieren

**A picture containing text, line, font, number

Description automatically generated**

**Room Controller:** Listen, Addieren, Löschen, Aktualieseren von Räumen

**A picture containing text, line, font, number

Description automatically generated**

**Kapitel 3 – DB-Zugriffschicht**

**Entity Framework Core:** Objekt basiert ZB-Zugriffgerät (ORM) für C# und .NET Core Laufzeit.

DB-Zugriffe werden durch Lesen- und Schreiben- generische Repositoryschnittstelle abstrahiert.

A screen shot of a computer program

Description automatically generated with low confidence

Write Repository

A screen shot of a computer code

Description automatically generated with low confidence

Read Repository

Für jede Objekt diese Repositoryklassen werden implementiert.

A screenshot of a computer program

Description automatically generated with medium confidence

Repository Klassen

**Kapitel 4 – DB-Entitäten**

Jede Entität erbt von BaseEntity Klasse

A screen shot of a computer code

Description automatically generated with low confidence

BaseEntity Klasse

**Kunden**

**A picture containing text, screenshot, software, multimedia software

Description automatically generated**

Customer Klasse

**Mitarbeitern**

**A picture containing text, screenshot, software, font

Description automatically generated**

BaseEntity Klasse

**Boden**

**A picture containing text, screenshot, software, multimedia software

Description automatically generated**

Floor Klasse

**Rezervierung** **A screen shot of a computer program

Description automatically generated with low confidence**

Rezervation Klasse

**Räume** **A screen shot of a computer program

Description automatically generated with low confidence**

Room Klasse