

TAU INF202 Software Engineering

Projektarbeit

Architekturspezifikation

SoftwareUnternehmen Informationssystem

Verantwortliche:

Hilal Ebru Duran, e200503037@stud.tau.edu.tr

Zeynep Örengül, e200503058@stud.tau.edu.tr

Beraterin:

Fulya Yenilmez, fulya.yenilmez@stud.tau.edu.tr

Stakeholder:

DI. Ömer Karacan, omer.karacan@stud.tau.edu.tr

Inhaltsverzeichnis

1. Einleitung.....	3
2. Architekturüberblick	4
3. Beschreibung der „Controller“ Klassen.....	5
4. Rückverfolgbarkeit der Anforderungen	6
5. Beschreibung der DB-Zugriffsschicht	7

1. Einleitung

Dieses Dokument wurde erstellt, um das Design des SoftwareUnternehmen Informationssystems, seine Komponenten, Funktionen und Abhängigkeiten zwischen den Komponenten zu beschreiben. Es besteht aus 5 Teilen, einschließlich diesem Teil. Nachfolgend finden Sie kurze Informationen zu anderen Untertiteln.

- 2) **Arcitekturüberblick:** Dieses Kapitel beschreibt die allgemeine Architektur des Systems, seine Komponenten, die Beziehungen zwischen ihnen und die verwendeten Technologien. So erhalten Sie einen Überblick über die Funktionsweise und Konfiguration der Software.
- 3) **Beschreibung der „Contoller“ Klassen:** In diesem Kapitel werden die in den Web-Services der Software verwendeten Controller-Klassen erläutert. Diese Klassen wurden entwickelt, um die Anforderungen der Benutzer zu erfüllen und angemessene Antworten zu geben.
- 4) **Rückverfolgbarkeit der Anforderungen:** Dieses Kapitel erläutert die Beziehung zwischen Benutzeranforderungen und dem Design der Software. Auf diese Weise wird darüber informiert, wie die Anforderungen erfüllt und umgesetzt werden.
- 5) **Beschreibung der DB-Zugriffsschicht:** Dieses Kapitel beschreibt die Datenbankzugriffsschicht der Software. Diese Schicht enthält die Repository-Klassen, die für den Zugriff auf die Datenbank und das Lesen/Schreiben von Daten verwendet werden.

2. Architekturüberblick

Während des Designprozesses des Systems wird die RESTful-API als Webservice-Architektur und Spring Boot als Framework verwendet.

Als architektonische Struktur wird Controller/ Service/ Repository Schichtenarchitektur verwendet. Anfragen des Benutzers werden vom Controller empfangen. Die Dienstschicht ermöglicht die Kommunikation zwischen dem Controller und dem Repository. Die Repository-Schicht verwaltet Datenbankoperationen. Ein einfaches Diagramm, das diese Struktur beschreibt, ist in *Abbildung 1* gegeben.

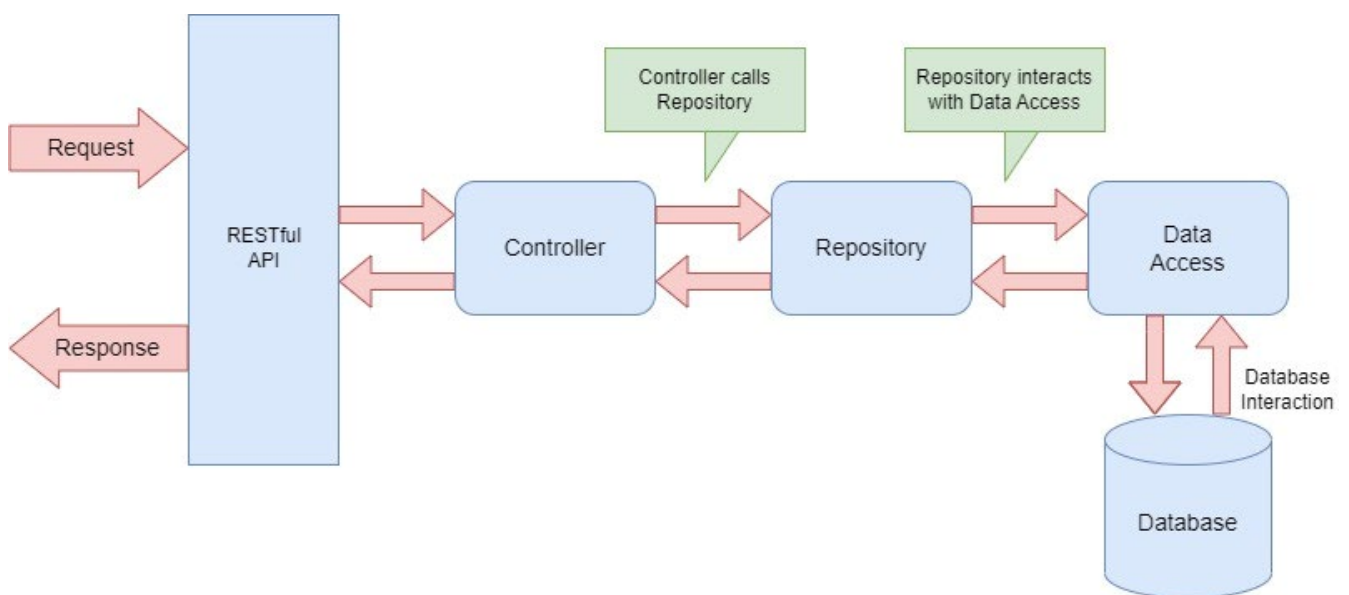


Abbildung 1

3. Beschreibung der „Controller“ Klassen

Zuerst haben wir einen Controller namens ProjectController erstellt und ihn dem Pfad „/project“ zugeordnet. In der Klasse haben wir eine Methode erstellt, die ein Objekt zurückgibt und einer POST- und einer DELETE-Anfrage zugeordnet ist. Daher wird jeder URL-Aufruf, der mit „project“ endet, an die postProjectID- und deleteProject-Methoden im ProjectController weitergeleitet. Wir verwenden diesen Controller, um das Projekt zu erstellen, seine Informationen zu speichern und bei Bedarf zu löschen.

Wir werden den EmployeeController verwenden, um das Kundenunternehmen zum System hinzuzufügen und bei Bedarf zu löschen. Für diesen Controller haben wir wie zuvor eine Methode erstellt, die ein Objekt in der Klasse zurückgibt und auf eine POST- und DELETE-Anforderung abbildet.

Schließlich verwenden wir den ReportController, um die dem System hinzugefügten Kundenunternehmen und Projekte anzuzeigen. Wir haben eine Methode für diesen Controller erstellt, die der GET-Anfrage zugeordnet ist.

Anhand von Anwendungsfällen erstellte Methoden und Controller-Anbindungen sind in Abbildung 2 dargestellt.

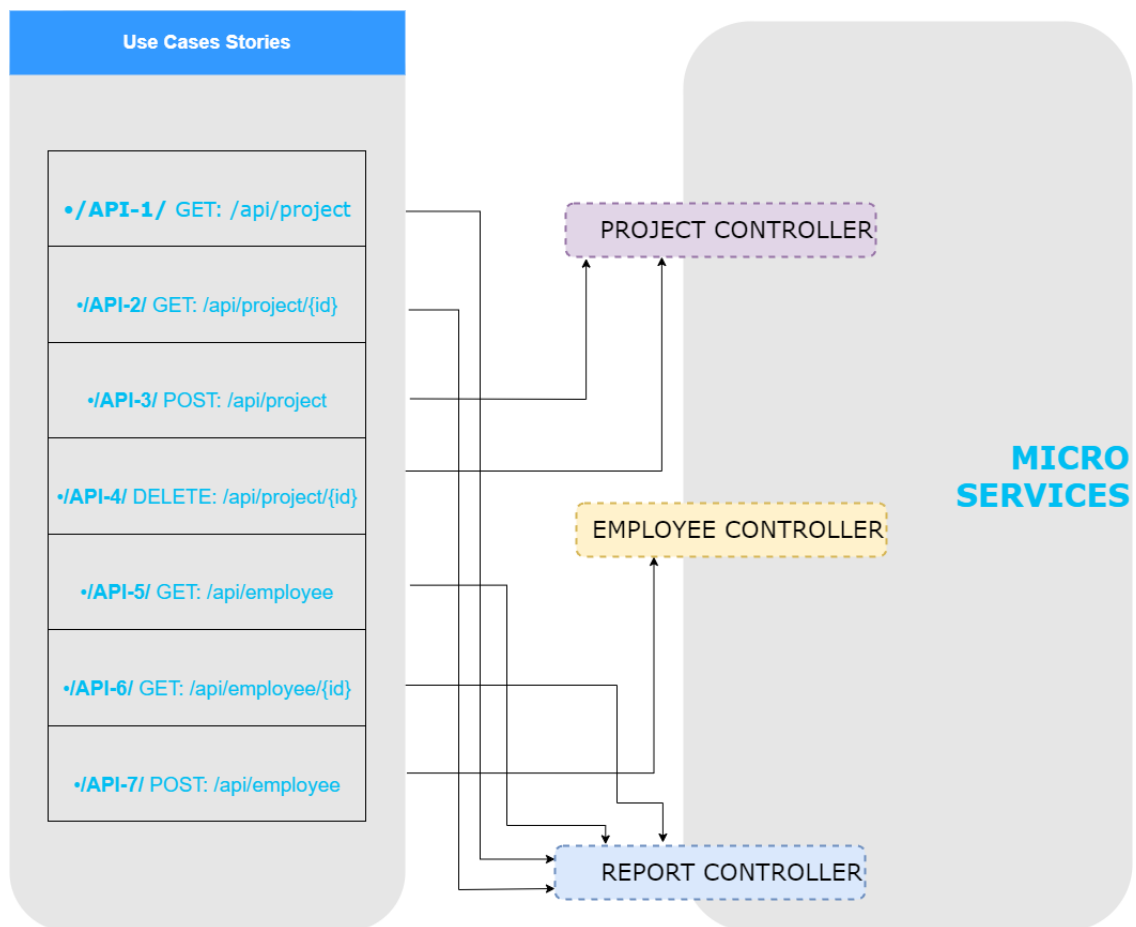


Abbildung 2

4. Rückverfolgbarkeit der Anforderungen

Unter dieser Überschrift werden die Verbindungen der im System definierten Use-Cases mit den zugehörigen Architekturkomponenten erläutert. Die Verbindungen wurden mithilfe ein einfaches Schemas erklärt und in der Abbildung 3 gegeben.

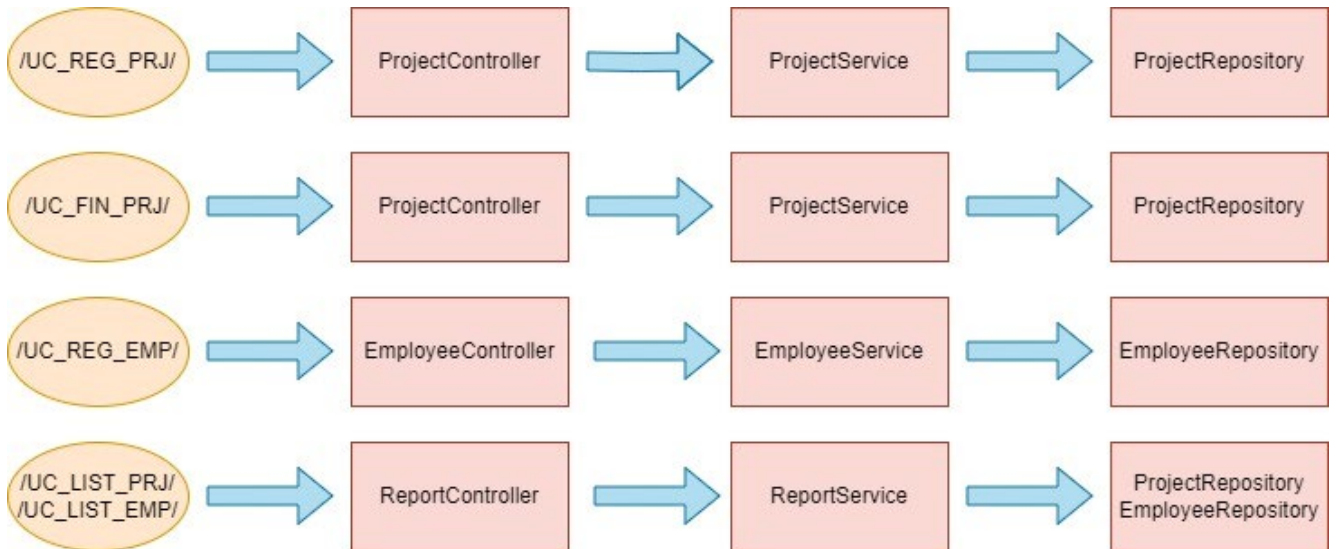


Abbildung 3

5. Beschreibung der DB-Zugriffsschicht

Dieser Abschnitt beschreibt den Datenbankzugriff der Software. Das System ist mit dem Datenbanksystem verbunden, um die Informationen von Projekten und Mitarbeitern zu speichern.

Verbindung und Zugriff zwischen Benutzer und System werden einfach in der Abbildung 4 dargestellt.

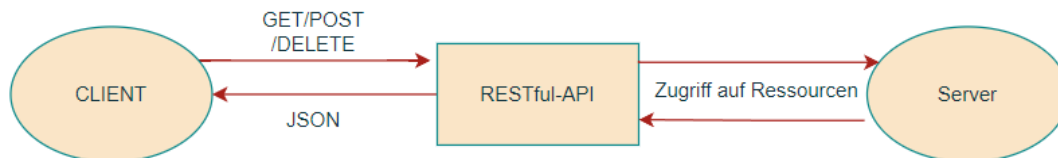


Abbildung 4

Zur Erstellung der Datenbank wird MySQL und für die Datenintegration wird RESTful API verwendet. Auch wurde JSON als Datenformat für den Datenaustausch festgelegt.

Die folgende Tabelle listet auf, welche Daten die Repository-Klasse in der Datenbank behält und mit welchen Datentypen dies benutzt werden.

ProjectRepository	Datentyp	EmployeeRepository	Datentyp
• projekt_id	• INT	• mitarbeiter_id	• INT
• firma	• VARCHAR (40)	• name	• VARCHAR(40)
• start_date	• DATE	• surname	• VARCHAR(40)
• end_date	• DATE	• isActive	• TINYINT
• managed_by	• VARCHAR (40)	• phone	• INT
• status	• VARCHAR (40)	• address	• VARCHAR(40)
• mitarbeiter_id	• INT	• salary	• INT

Tabelle 1