

# Klinikverwaltungssystem

İpek Yılmaz 200503052  
Haluk Harun Gündoğan 200503031

- **Wer wird die Applikation verwenden?**

Diese Applikation wird bei den Sekretärinnen in der Klinik benutzt.

- **Wann wird die Applikation verwendet?**

Wenn ein neuer Patient kommt oder wenn ein Patient einen Termin erstellen möchte, wird die Applikation verwendet. Wenn ein Patient nicht mehr der Patient der Klinik ist, wird dieser Patient aus dem System gelöscht.

- **Warum wird die Applikation verwendet?**

Um die Verwaltung von Patientendaten und Terminen zu erleichtern, sie digital und sicher zu speichern, wird die Applikation verwendet.

- **Was macht die Applikation?**

Diese Applikation fügt die Patienten ein, setzt die Termine auf, listet die Patienten und Termine. Wenn nötig, löscht die Applikation die Patienteninformationen.

- **Wo wird die Applikation eingesetzt?**

Die Applikation wird in den Computern der Sekretärinnen eingesetzt.

- **Wie wird die Applikation verwendet?**

Der Benutzer kann die gewünschte Operation aus dem Systeminterface auswählen.

- **Welche Programmiersprachen werden benutzt?**

React und Spring Boot werden benutzt.

Presentation Frontend  
Integration

POST <http://localhost:8080/Termine/neuerTermin>

Webserver

Microservices

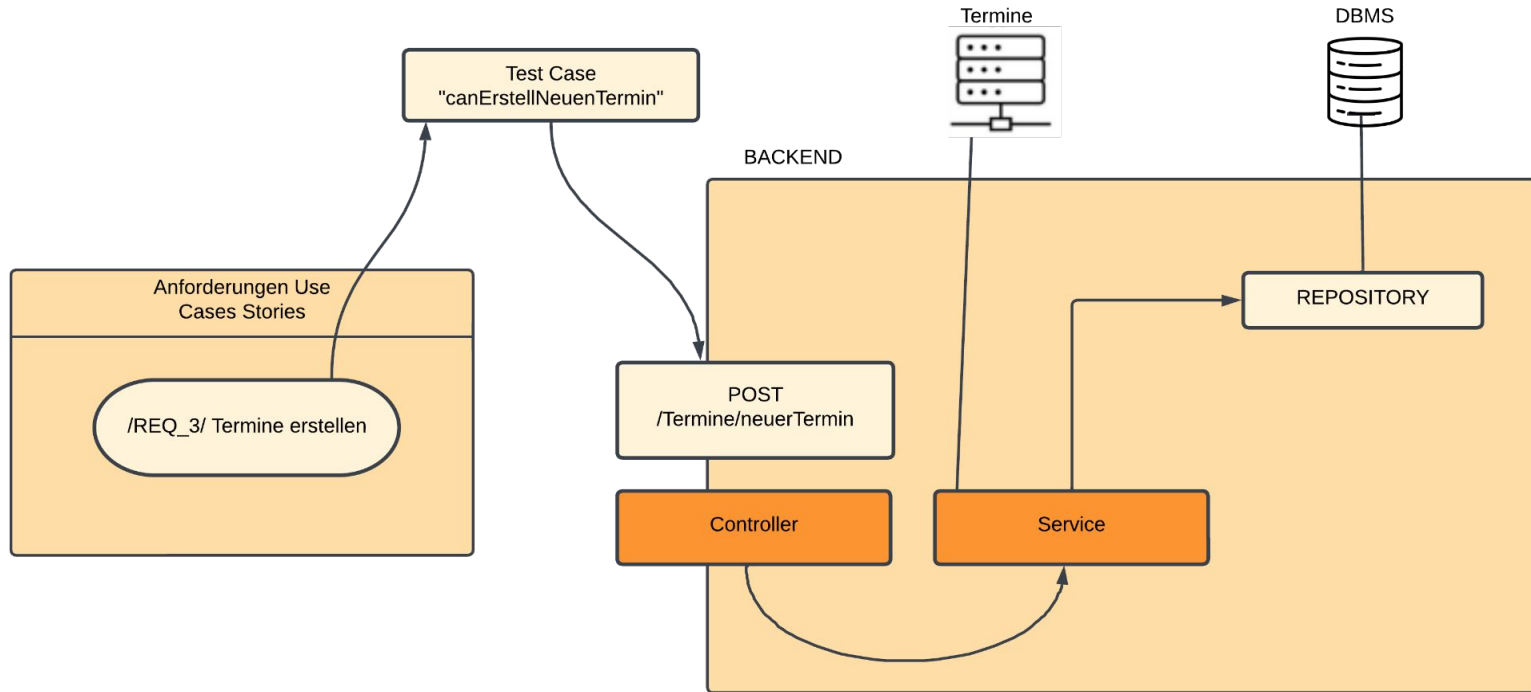
TerminController

TerminService

TerminRepository

Database





The screenshot shows an IDE with a project structure on the left, a code editor in the center, and a run console at the bottom.

**Project Structure:**

- Inf202BackendKlinikApplication
  - resources
  - test
    - java
      - com.example.inf204backendklinik
        - patient
        - termine
          - TerminRepositoryTest (selected)
          - TerminServiceTest

**Code Editor:**

```
harungundogan +1 *  
@Test  
void checksWhetherDatumExists() {  
  
    Patient patient = new Patient( ausweisnummer: 39722222222L);  
    Doktor doktor = new Doktor( doktor_id: 1L);  
    LocalDate now = LocalDate.parse( text: "2019-12-15");  
    Termine termin = new Termine(now, zeit: "9.35");  
  
    underTest.save(termin);  
  
    Boolean expected = underTest.existsTerminByDatum(termin.getTerminDatum());  
  
    assertThat(expected).isTrue();  
}
```

**Run Console:**

Run: Inf202BackendKlinikApplication x TermineRepositoryTest.checksWhetherDatumExists x

Tests passed: 1 of 1 test - 403 ms

TermineRepositoryTest (com.e 403 ms)  
checksWhetherDatumExist: 403 ms

C:\Users\HaLuk\.jdk\openjdk-19.0.2\bin\java.exe ...

16:14:34.798 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Neither  
16:14:34.802 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default  
16:14:34.803 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not det  
16:14:34.863 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using  
16:14:35.023 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Ident

The screenshot displays an IDE with a project structure on the left, a code editor in the center, and a run console at the bottom.

**Project Structure (Left):**

- PatientService
  - termine
    - Termin
    - TerminController
    - TerminRepository
    - TerminService
  - Inf202BackendKlinikApplication
  - resources
  - test
    - java
      - com.example.inf204backendklinik
        - patient
        - termine
          - TerminRepositoryTest (selected)
          - TerminServiceTest

**Code Editor (Center):**

```
46  
47 @Test  
48 void checkWhetherTerminZeitExists() {  
49     //given  
50     Patient patient = new Patient( ausweisnummer: 39722222222L);  
51     Doktor doktor = new Doktor( doktor_id: 1L);  
52     LocalDate now = LocalDate.now();  
53     Termine termin = new Termine(now, zeit: "9.40");  
54     underTest.save(termin);  
55     // when  
56     Optional<Termine> expected2 = underTest.findTerminByZeit("9.40");  
57     // then  
58     assertNotNull(expected2);  
59 }
```

**Run Console (Bottom):**

Run: Inf202BackendKlinikApplication x TerminRepositoryTest.checkWhetherTerminZeitExists x

Tests passed: 1 of 1 test - 344 ms

TerminRepositoryTest (com.e 344 ms)  
checkWhetherTerminZeitEx 344 ms

C:\Users\HaluK\.jdk\openjdk-19.0.2\bin\java.exe ...

16:15:01.058 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Neither  
16:15:01.063 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default  
16:15:01.064 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not det  
16:15:01.127 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using  
16:15:01.274 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Ident

The screenshot shows an IDE with a project structure on the left, a code editor in the center, and a run console at the bottom.

**Project Structure (Left):**

- TerminRepository
- TerminService
- Inf202BackendKlinikApplication
  - resources
  - test
    - java
      - com.example.inf204backendklinik
        - patient
        - termine
          - TerminRepositoryTest
          - TerminServiceTest
- Inf202BackendKlinikApplicationTests
  - resources
  - target
  - classes
  - generated-sources
  - generated-test-sources
  - test-classes

**Code Editor (Center):**

```
51 @Test
52 void canErstellNeuenTermin() {
53     //given
54     Patient patient = new Patient( ausweisnummer: 39722222222L);
55     Doktor doktor = new Doktor( doktor_id: 1L);
56     LocalDate now = LocalDate.now();
57     Termine termin = new Termine(now, zeit: "9.40");
58     // when
59     underTest.erstellNeuenTermin(termin);
60     //then
61     ArgumentCaptor<Termine> termineArgumentCaptor = ArgumentCaptor.forClass(Termine.class);
62     Mockito.verify(terminRepository).save(terminArgumentCaptor.capture());
63     Termine capturedTermin = termineArgumentCaptor.getValue();
64     assertThat(capturedTermin).isEqualTo(termin);
65 }
66
```

**Run Console (Bottom):**

Run: Inf202BackendKlinikApplication x TerminServiceTest.canErstellNeuenTermin x

Tests passed: 1 of 1 test - 889 ms

TerminServiceTest (com.exarr 889 ms)

- canErstellNeuenTermin() 889 ms

C:\Users\Haluk\.jdk\openjdk-19.0.2\bin\java.exe ...

Process finished with exit code 0



http://localhost:8080/Termine/neuerTermin



Save



POST



http://localhost:8080/Termine/neuerTermin

Send



Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON



Beautify

```
1  {}
2  ... "terminDatum": "2021-06-24",
3  ... "zeit": "9.35",
4  ... "patient": {
5  ... | ... "ausweisnummer": 39271172908
6  ... | },
7  ... "dokter": {
8  ... | ... "dokter_id": 1
9  ... | }
10 {}
```

Body

Cookies

Headers (4)

Test Results



200 OK

379 ms

123 B



Save as Example



Pretty

Raw

Preview

Visualize

Text



1



```
linik=# SELECT * FROM Termine;
```

termin_id	termin_datum	zeit	doktor_id	ausweisnummer
1	2021-06-24	9.35	1	39271172908

```
(1 row)
```

## M#5

Github: <https://github.com/Gruppe1-Fulya/Klinikverwaltungssystem>

Video: [https://drive.google.com/file/d/1OqH4u2Y99IHWNVFCqVfUVldYqnoC51O4/view?usp=drive\\_link](https://drive.google.com/file/d/1OqH4u2Y99IHWNVFCqVfUVldYqnoC51O4/view?usp=drive_link)