

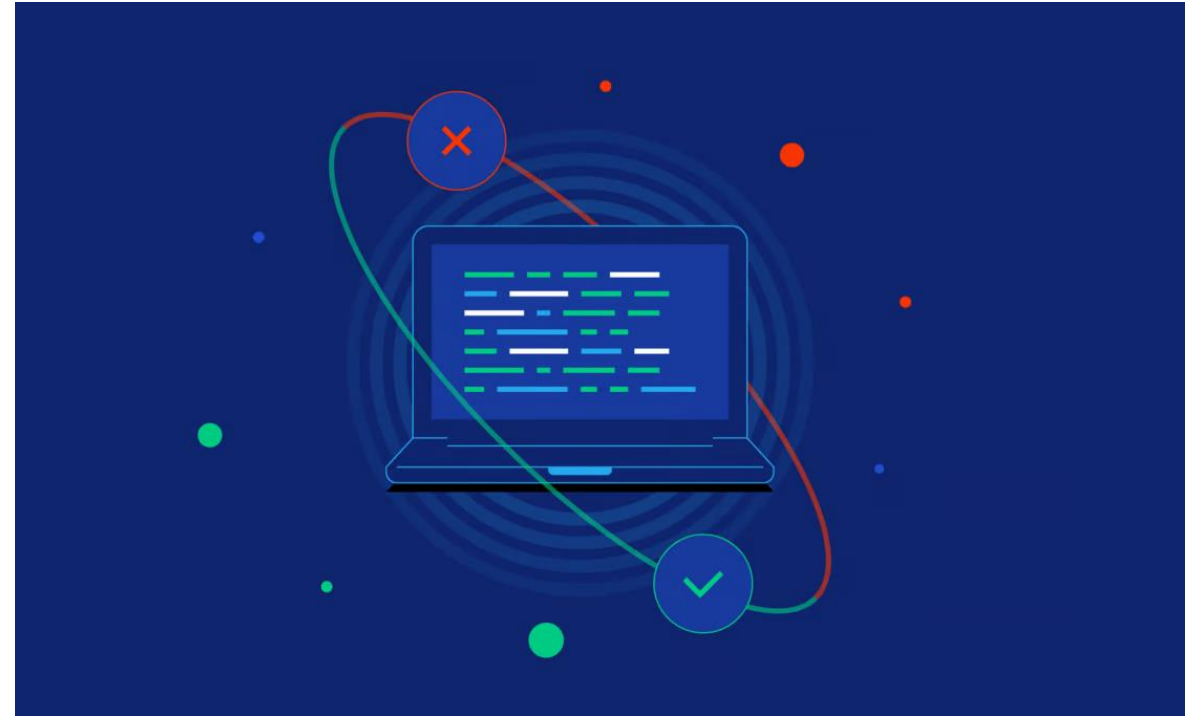
Softwaretest

FOKUS: Systemtest

INF202

Kaan İmamoğlu 200503011


Ahmet Oğuz Örsler 200503035



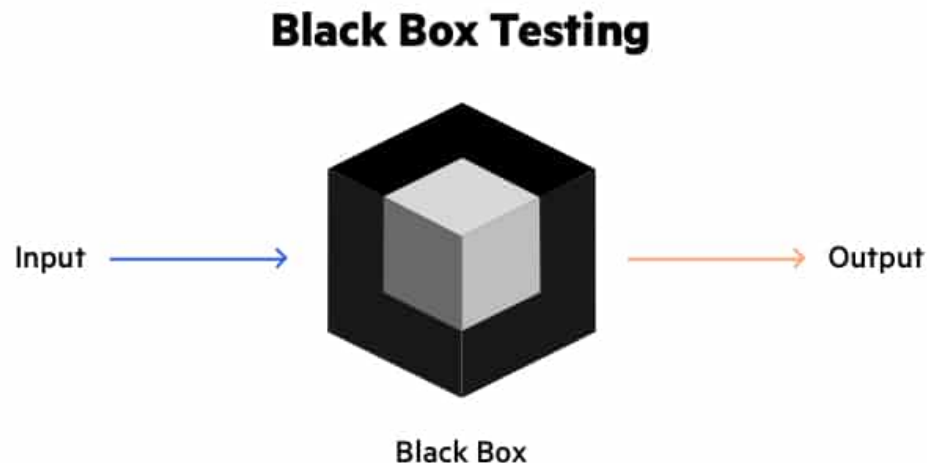
Was ist der Softwaretest?

In der Softwareentwicklung werden Systemtests durchgeführt, um ein vollständiges und vollständig integriertes Softwaresystem zu testen. Ein Systemtest wird durchgeführt, um die Konformität des Systems mit den festgelegten Anforderungen zu bewerten.

Systemtests stellen sicher, dass alle Softwaremodule oder Komponenten, wenn sie zusammengestellt werden, wie erwartet funktionieren und das System als Ganzes korrekt und effizient funktioniert. Sie wird normalerweise von einem Testerteam durchgeführt und findet vor dem Abnahmetest und nach dem Integrationstest statt.



Was ist der Softwaretest?

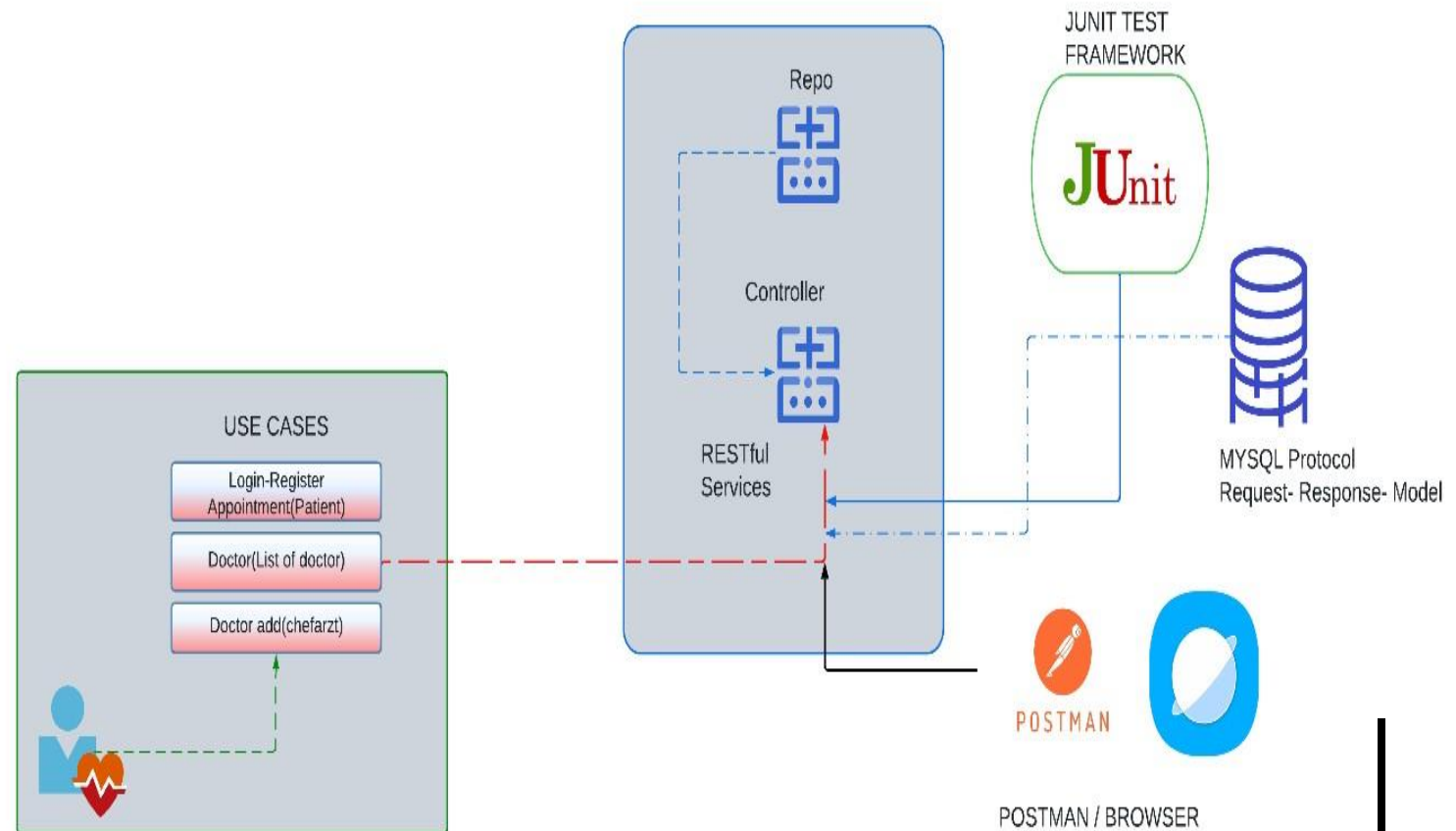


Systemtests stellen sicher, dass alle Softwaremodule oder Komponenten, wenn sie zusammengestellt werden, wie erwartet funktionieren und das System als Ganzes korrekt und effizient funktioniert. Sie wird normalerweise von einem Testerteam durchgeführt und findet vor dem Abnahmetest und nach dem Integrationstest statt.

Die Testphase dieses Projekts wurde mit Hilfe der Black-Box-Methode durchgeführt.

Systemüberblick

Benutzerregistrierungs- und Anmeldebildschirm:
Dieser Bildschirm ermöglicht den Benutzern die erste Erfahrung mit dem System. Im Registrierungsformular werden persönliche Daten des Benutzers wie Name und E-Mail abgefragt. Diese Informationen werden in der Datenbank gespeichert, um den Benutzer in späteren Sitzungen zu identifizieren. Auf dem Anmeldebildschirm können bereits registrierte Benutzer durch Eingabe ihres Namens und ihrer E-Mail-Informationen auf das System zugreifen.



Systemüberblick

Arztbildschirm: Auf diesem Bildschirm können Krankenhausadministratoren oder Benutzer mit bestimmten Berechtigungen Arztinformationen hinzufügen. Informationen wie Name des Arztes, Fachgebiet, Arbeitszeiten werden hinzugefügt und diese Informationen werden in der Datenbank gespeichert. Auf diesem Bildschirm werden auch die aus der Datenbank abgerufenen Arztinformationen angezeigt. Die Liste der verfügbaren Ärzte können Benutzer hier einsehen.

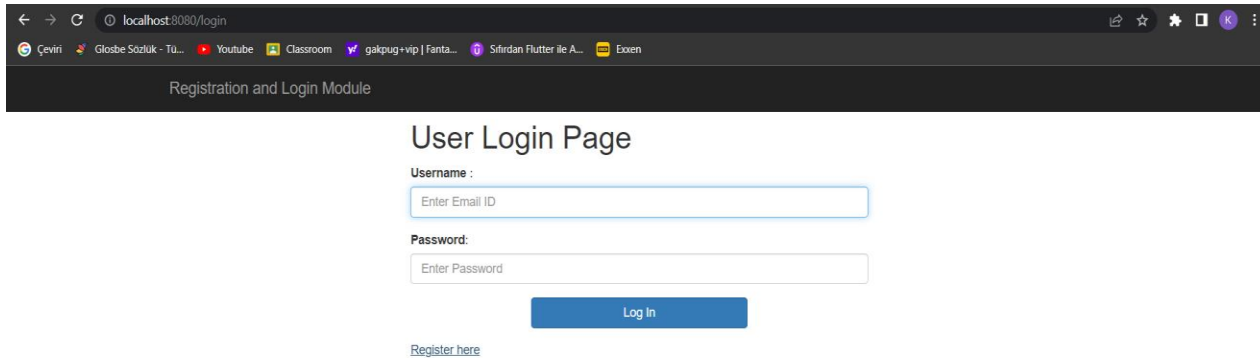
Terminbildschirm: Über diesen Bildschirm können Benutzer einen Arzt auswählen und einen Termin vereinbaren. Die erstellten Termininformationen werden in der Datenbank gespeichert und Benutzer können ihre eigenen Termininformationen anzeigen.

Test-Systemüberblick

Was machen wir?

Zuerst starten wir den Testprozess. Wir tun dies mithilfe der @Test-Annotation im JUnit-Framework. In diesem Prozess haben wir festgelegt, was der Test testen soll und wie die Testdaten aussehen werden. Anschließend simulieren wir die Abhängigkeiten des Tests, in diesem Fall den Krankenhaus-Webdienst. In diesem Schritt legen wir fest, welche Art von Antwort wir vom Webservice erwarten. Im Rahmen der Tests rufen wir Funktionen auf, die die gewünschten Informationen abrufen. Diese Funktionen verwenden als Parameter die ID des Arztes, den Termin und den Benutzer (abhängig vom Test). Wenn der zurückgegebene Wert die erwarteten Informationen enthält, bestehen wir den Test. Sollten nicht die erwarteten Informationen enthalten sein, erklären wir den Test für nicht erfolgreich.

Login-Register



Registration and Login Module

User Login Page

Username :

Password:

[Log In](#)

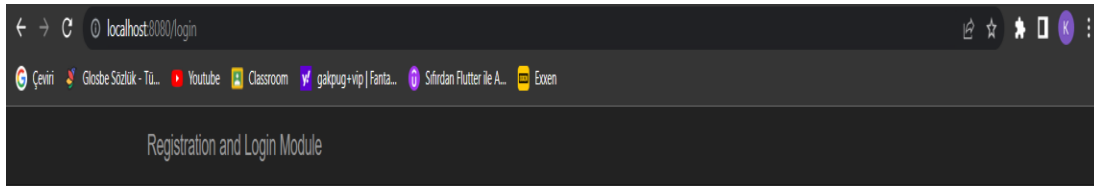
[Register here](#)

Nach dem Ausführen des Projekts wird bei Eingabe des Ports localhost:8080 vor dem Kunden eine Login-Registrierungsseite angezeigt. Wenn sich die Person noch nie zuvor angemeldet hat, sollte sich das System auf dieser Seite in der Datenbank registrieren. Es stellt dies auch im Registerbereich bereit

Login Interface

Test Name	Login database interface
Conditions	E-Mail-Adresse und Password, die bereits in der Datenbank registriert sind
Test Steps	Es sollten Informationen eingetragen werden, die bereits in der Datenbank registriert sind.
Expected Results	«You have succesfully logged in message» and Main (index) interface
Actual Results	«You have succesfully logged in message» and Main (index) interface
Verdict (Pass/Fail)	Pass

Login Postman- Browser



User Login Page

Username :

Password:

Log In

[Register here](#)

GET http://localhost:8080/login

Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Header	Value
Postman-Token	<calculated when request is sent>
Host	<calculated when request is sent>
User-Agent	PostmanRuntime/7.32.2
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

Body Cookies (1) Headers (12) Test Results

200 OK 16 ms 2.41 KB Save as Example

Pretty Raw Preview Visualize HTML

```
53 <div class="form-group">
54   <label for="password">Password</label>: <input type="password"
55   id="password" name="password" class="form-control"
56   placeholder="Enter Password" />
57 </div>
58
59 <div class="form-group">
60   <div class="row">
61     <div class="col-sm-6 col-sm-offset-3">
62       <input type="submit" name="login-submit" id="login-submit"
63       class="form-control btn btn-primary" value="Log In" />
64     </div>
65   </div>
66 </div>
```

Register Interface

Test Name	register database interface
Conditions	passende E-Mail, Vor- und Nachname, Passwort für die Anmeldung im System (Datenbank)
Test Steps	Der Kunde gibt die gewünschten Informationen in das System ein und verwendet diese für spätere Anmeldungen.
Expected Results	«You have succesfully registered message» Informationen zur Datenbank
Actual Results	«You have succesfully logged in message» Informationen zur Datenbank
Verdict (Pass/Fail)	Pass

Register Interface Database

✓ Gösterilen satır 0 - 3 (toplam 4, Sorgu 0,0006 saniye sürdü.)

```
SELECT * FROM `user`
```

☐ Profil çıkart [[Satır içi düzenle](#)] [[Düzenle](#)] [[SQL'i aç](#)] [[PHP kodu oluştur](#)] [[Yenile](#)]

☐ Tümünü göster

Satır sayısı: 25

Satırları süz: Bu tabloda ara

Anahtara göre sırala: Yok

Fazladan seçenekler

			id	first_name	last_name	email	password	
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1	Kaan	İmamoğlu	sadasd@gmail.com	\$2a\$10\$QyxQkp/kx0qxYvIwkHc9ze/kfFRznovdWz1ewTs2BMO...
<input type="checkbox"/>	Düzenle	Kopyala	Sil	2	Kaan	İmamoğlu	imamoglu.kaan2002@gmail.com	\$2a\$10\$O9.563w/dhu4BO9vPK9fN.5z/EA0AEtx.E9qslGsyZw...
<input type="checkbox"/>	Düzenle	Kopyala	Sil	3	oğuz	örşler	asdas	\$2a\$10\$Ic4mWkRHkYoc3GlmbU3E1uQ07.Xyc8770EUSTeNV5Dc...
<input type="checkbox"/>	Düzenle	Kopyala	Sil	4	selo	imamoglu	askjaskjd@gmail.com	\$2a\$10\$eAIEUF89v4CmSLdrXp2ON0Y7DiwJG6KltzjfuTXa8l...

„UserRegistration Controller“,
im Falle einer Registrierung
werden die mit Hilfe von GET
`http://localhost:8080/registratio`
`n` und `UserService` und
`UserRepository` eingegebenen
Informationen in die POST
`http://localhost:8080/registratio`
`n?success-Datenbank`
übertragen.

Doctor Management Interface

Test Name	doctor database interface
Conditions	Der Kunde muss im System eingetragen sein.
Test Steps	<ul style="list-style-type: none">- Arzt aus dem System (Datenbank) löschen- Arzt aus dem System (Datenbank) hinzufügen
Expected Results	<ul style="list-style-type: none">- Erhalten Sie Arztinformationen vom Service und sehen Sie sich die Liste an- Ändern der Informationen in der Datenbank mit der Schaltfläche „Update“.- Entfernen und Löschen des «id» mit Hilfe des Service mit der Schaltfläche „Delete“.- Öffnen eines Formulars zum Hinzufügen eines Arztes und Hinzufügen zur Datenbank mithilfe des Repository und des Controllers mit der Schaltfläche „Add Doctor“

Doctor Management Interface

Actual Results

- Erhalten Sie Arztinformationen vom Service und sehen Sie sich die Liste an
- Ändern der Informationen in der Datenbank mit der Schaltfläche „Update“.
- Entfernen und Löschen des «id» mit Hilfe des Service mit der Schaltfläche „Delete“.
- Öffnen eines Formulars zum Hinzufügen eines Arztes und Hinzufügen zur Datenbank mithilfe des Repositorys und des Controllers mit der Schaltfläche „Add Doctor“

Verdict(Pass/Fail)

Pass

Sunucu: 127.0.0.1 » Veritabanı: demo » Tablo: doctor

Gözet Yapı SQL Ara Ekle Dışa aktar İçe aktar Yetkiler

✓ Gösterilen satır 0 - 2 (toplam 3, Sorgu 0,0004 saniye sürdü.)

SELECT * FROM `doctor`

☐ Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıkla] [PHP kodu oluştur] [Yenile]

☐ Tümünü göster | Satır sayısı: 25 Satırları süz: Bu tabloda ara Anahtara göre

Fazladan seçenekler

				id	email	first_name	last_name
<input type="checkbox"/>	Düzenle	Kopyala	Sil	2	sakdjhk@gmail.com	Kaan	İmamson
<input type="checkbox"/>	Düzenle	Kopyala	Sil	3	xxx@gmail.com	Selo	İmamson
<input type="checkbox"/>	Düzenle	Kopyala	Sil	5	aaa@gmail.com	Melisa	Aykaç

☐ Tümünü işaretle Seçilileri: Düzenle Kopyala Sil Dışa aktar

localhost:8080/doctor

Çeviri Glosbe Sözlük - Tü... Youtube Classroom gakpug+vip | Fanta... Sıfırdan Flutter ile A... Exxen

Hospital Management System Doctor Management Appointment Management Logout

List of Doctors

Add Doctor

Doctor First Name	Doctor Last Name	Doctor Email	Actions
Kaan	İmamson	sakdjhk@gmail.com	<button>Update</button> <button>Delete</button>
Selo	İmamson	xxx@gmail.com	<button>Update</button> <button>Delete</button>
Melisa	Aykaç	aaa@gmail.com	<button>Update</button> <button>Delete</button>

Appointment Management Interface

Test Name	appointment database interface
Conditions	Der Kunde muss im System eingetragen sein.
Test Steps	<ul style="list-style-type: none">- Appointment aus dem System (Datenbank) löschen- Appointment aus dem System (Datenbank) hinzufügen
Expected Results	<ul style="list-style-type: none">- Erhalten Sie Appointmentinformationen vom Service und sehen Sie sich die Liste an- Ändern der Informationen in der Datenbank mit der Schaltfläche „Update“.- Entfernen und Löschen des «id» mit Hilfe des Service mit der Schaltfläche „Delete“.- Öffnen eines Formulars zum Hinzufügen eines Appointmentes und Hinzufügen zur Datenbank mithilfe des Repository und des Controllers mit der Schaltfläche „Add Appointment“

Appointment Management Interface

Actual Results

- Erhalten Sie Arztinformationen vom Service und sehen Sie sich die Liste an
- Ändern der Informationen in der Datenbank mit der Schaltfläche „Update“.
- Entfernen und Löschen des «id» mit Hilfe des Service mit der Schaltfläche „Delete“.
- Öffnen eines Formulars zum Hinzufügen eines Appointmentes und Hinzufügen zur Datenbank mithilfe des Repositorys und des Controllers mit der Schaltfläche „Add Appointment“

Verdict(Pass/Fail)

Pass

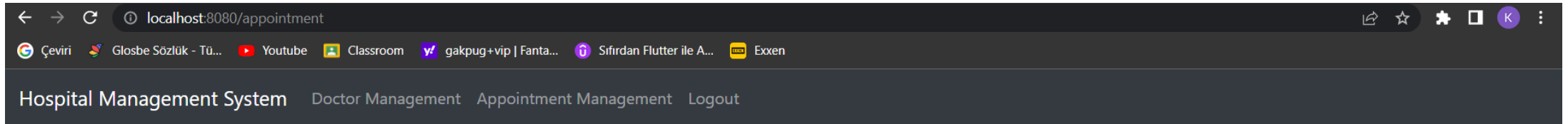
SELECT * FROM `appointment`

☐ Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıkla] [PHP kodu oluştur] [Yenile]

☐ Tümünü göster | Satır sayısı: 25 ▼ Satırları süz: Bu tabloda ara

Fazladan seçenekler

				id	doctor_name	patient_name	time
<input type="checkbox"/>		Düzenle		Kopyala		Sil	2 Selo Oğuz 21:00-22:00
<input type="checkbox"/>		Düzenle		Kopyala		Sil	3 Melisa Kaan 20:00-21:00



List of Appointments

Add Appointment

User Name	Doctor Name	Time(...)	Actions
Oğuz	Selo	21:00-22:00	<button>Update</button> <button>Delete</button>
Kaan	Melisa	20:00-21:00	<button>Update</button> <button>Delete</button>

KURZE BESCHREIBUNG DER INTEGRATIONSTESTFÄLLE (Database)

Test Name	ProvideReportofDoctor database
Preconditions	Database Table “Doctor” includes a row for: doctorID = <generated> doctorFirstname = "Kaan" doctorLastname = "Imamson" doctorEmail = "sakdjhk@gmail.com" Hospital Web service call is mocked!
Test Steps	1- Ensure that Hospital Web service call is mocked! 2- Junit Test Case calls provideReportOfDoctor(int doctorID) with the doctor's ID. 3- Response data is parsed, and the expected data is selected
Postconditions	none

KURZE BESCHREIBUNG DER INTEGRATIONSTESTFÄLLE (Database)

Test Data	"Kaan","İmamson","sakdjhk@gmail.com"
Expected Results	Doctor.doctorName = "Kaan"
Actual Results	Doctor.doctorName = "Kaan"
Verdict(Pass/Fail)	Pass

KURZE BESCHREIBUNG DER INTEGRATIONSTESTFÄLLE (Database)

In diesem Test zeigen wir, dass es in der Datenbank eine Tabelle mit dem Namen „Doctor“ gibt, die die Informationen eines Arztes enthält. Der Zweck des Tests besteht darin, zu überprüfen, ob die Informationen eines bestimmten Arztes korrekt aus der Datenbank übernommen wurden. In diesem Fall verwenden wir als Testdaten einen Arzt namens „Kaan“. Außerdem weisen wir darauf hin, dass der Webservice-Aufruf verspottet wird. Dies bedeutet, dass kein tatsächlicher Aufruf des Webdienstes erfolgt, sondern ein bestimmtes Ergebnis zurückgegeben wird.

JUnit Test

The screenshot shows the Eclipse IDE interface with the following components:

- Top Bar:** INF202-Krankenhaus-Terminsystem - KrankenhausTerminsystem/src/test/java/com/example/krankenhausterminsystem/KrankenhausTerminsystemTests.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Standard Eclipse development tools.
- Package Explorer:** Shows the project structure with 'KrankenhausTerminsystemTests.java' selected.
- JUnit Runner:** Shows 'Finished after 4,992 seconds', 'Runs: 2/2', 'Errors: 0', and 'Failures: 1'.
- Code Editor:** Displays the source code for 'KrankenhausTerminsystemTests.java' with the following imports and annotations:

```
10 import org.junit.jupiter.api.Test;
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
13 import org.springframework.boot.test.context.SpringBootTest;
14 import org.springframework.test.web.servlet.MockMvc;
15
16 import com.example.krankenhausterminsystem.web.MainController;
17
18 @SpringBootTest
19 @AutoConfigureMockMvc
```
- Outline:** Shows the class hierarchy for 'KrankenhausTerminsystemTests' with methods like 'controller : MainCc', 'mockMvc : MockM', 'contextLoads() : vc', and 'shouldReturnDefai'.
- Console:** Displays the output of the test run, including mock HTTP response and request details:

```
<terminated> KrankenhausTerminsystemTests [JUnit] C:\Users\imamo\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (1 Haz 2023 13:15:53 - 13:16:00) [pid: 210]

MockHttpServletResponse:
    Status = 302
    Error message = null
    Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-revalidate"]
    Content type = null
    Body =
    Forwarded URL = null
    Redirected URL = http://localhost/login
    Cookies = []

MockHttpServletRequest:
    HTTP Method = GET
    Request URI = /
    Parameters = {}
    Headers = []
    Body = null
    Session Attrs = {SPRING_SECURITY_SAVED_REQUEST=DefaultSavedRequest[http://localhost/]}
```
- Failure Trace:** Shows the details of the test failure.

Rückverfolgbarkeit

Die Rückverfolgbarkeit des von uns erstellten Systems wird wie folgt kurz dargestellt. Die gewünschten Daten im System werden einfach in die Datenbank verarbeitet.

