

TAU INF202 Software Engineering

Projektarbeit

Systemtest Spezifikation

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. Systemüberblick	3
2. Systemtestfälle	4
3. Rückverfolgbarkeit	18

1. Systemüberblick

Um sicherzustellen, dass jede Komponente unseres Systems korrekt funktioniert, werden mithilfe von JUnit Unit-Tests durchgeführt. Der Zweck von Unit-Tests besteht darin, die Anwendung in die kleinsten testbaren Teile zu zerlegen und ihre Funktionalität zu überprüfen. In der Abbildung 1 wurde dazu ein einfaches Schema gegeben.

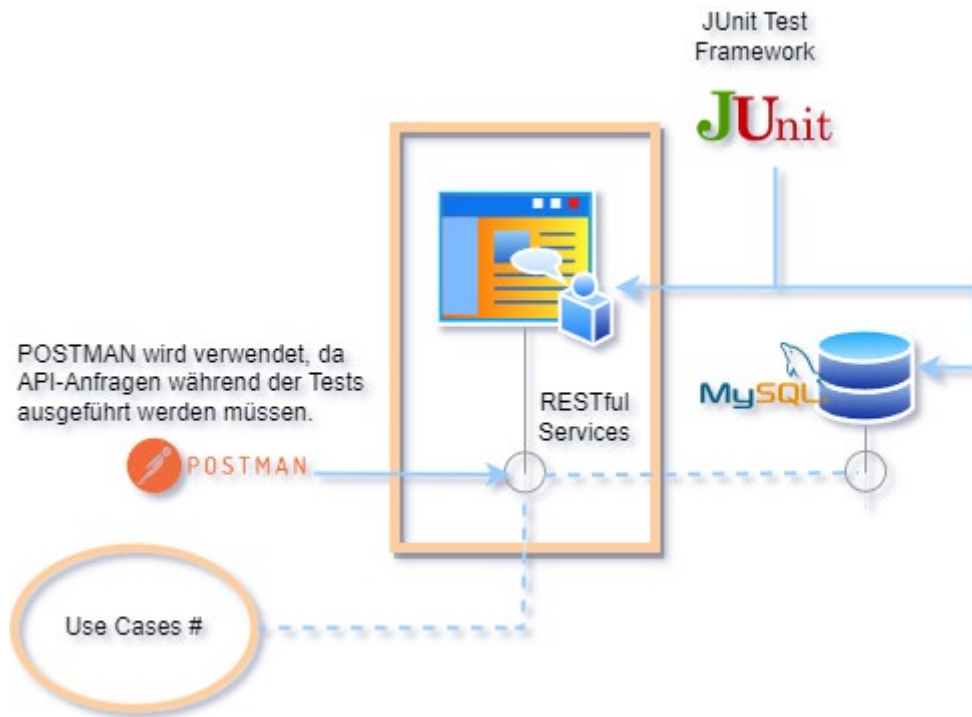


Abbildung 1

2. Systemtestfälle

In diesem Abschnitt des Berichts finden Sie Informationen zum Testen von Systemschnittstellen und Systemtestfällen.

Im ersten Teil müssen wir die API-Funktionalität testen, indem wir Anfragen an den Server senden und die Antworten analysieren. Hierzu müssen Integrationstests für Anwendungsfälle durchgeführt werden. Für diese Tests wurde der Postman API Client aufgrund seiner Kompatibilität mit dem REST-Architekturprotokoll und den zu verwendenden Programmiersprachen als Testtool ausgewählt.

Erinnern wir uns an die für unser Projekt ermittelten Anwendungsfälle:

Use Case#1 Name:	Register Project
Summary:	Der Benutzer notiert die Einzelheiten über das Projekt.
Pre-Condition:	-
Basic Flow:	<ol style="list-style-type: none">1. Use Case#1 beginnt, wenn der Benutzer ein Projekt hinzufügen möchte.2. Das System fragt nach dem Projektnamen, dem Start- und Enddatum des Projekts, der Anzahl der zu arbeitenden Personen nach Abteilungen und weitere Informationen. Einige der Informationen sind erforderlich.3. Der Benutzer gibt die Informationen ein.4. Je nach Personenzahl und Abteilung ordnet das System dem Projekt die passenden Mitarbeiter zu und druckt deren Namen und dann fordert den Benutzer zur Bestätigung auf.5. Nach der Genehmigung speichert das System Informationen.
Alternative Flows:	<ol style="list-style-type: none">3. Wenn der Benutzer ein erforderliches Feld nicht ausfüllt, wird eine Meldung angezeigt und Use Case wiederholt Schritt 2.5. Wenn der Benutzer nicht zustimmt, dann Use Case wiederholt Schritt 2.

Tabelle1: Use Case#1

Use Case#2 Name:	Finish Project
Summary:	Der Benutzer aktualisiert den aktuellen Status des Projekts als „Abgeschlossen“.
Pre-Condition:	Das Projekt muss gespeichert und mit seiner eindeutigen ID gefunden werden.
Basic Flow:	<ol style="list-style-type: none">1. Use Case#2 beginnt, wenn der Benutzer den Status des abgeschlossenen Projekts aktualisieren möchte.2. Das System fragt nach einer Bestätigung des Projekts, das der Benutzer abschließen möchte.3. Nach der Genehmigung markiert das System das Projekt als „Abgeschlossen“.4. Das System aktualisiert auch den Status der Mitarbeiter, die am Projekt arbeiten.
Alternative Flows:	<ol style="list-style-type: none">2. Wenn der Benutzer nicht zustimmt, wird Use Case#2 abgebrochen.

Tabelle2: Use Case#2

Use Case#3 Name:	Register Employee
Summary:	Der Benutzer möchte einen neuen Mitarbeiter hinzufügen.
Pre-Condition:	-
Basic Flow:	<ol style="list-style-type: none">1. Use Case#3 beginnt, wenn ein neuer Mitarbeiter eingestellt wird.2. Das System fordert zur Eingabe der erforderlichen Informationen über den Mitarbeiter auf.3. Der Benutzer gibt die Informationen ein.4. Je nach Personenzahl und Abteilung ordnet das System dem Projekt die passenden Mitarbeiter zu und druckt deren Namen und dann fordert den Benutzer zur Bestätigung auf.5. Das System speichert die Informationen.
Alternative Flows:	<ol style="list-style-type: none">3. Wenn der Benutzer ein erforderliches Feld nicht ausfüllt, wird eine Meldung angezeigt und Use Case wiederholt Schritt 2.

Tabelle3: Use Case#3

Use Case#4 Name:	Projekt/ Mitarbeiter Anzeigen
Summary:	Der Benutzer möchte Informationen zu einem registrierten Projekt oder einem Mitarbeiter anzeigen.
Pre-Condition:	Das Projekt oder der Mitarbeiter mit der eindeutigen ID gefunden werden.
Basic Flow:	<ol style="list-style-type: none"> 1. Use Case#4 beginnt, wenn der Benutzer Projekt- oder Benutzerinformationen anzeigen möchte. 2. Das System fragt nach der ID des Projekts oder des Mitarbeiters. 3. Der Benutzer gibt die Informationen ein. 4. Informationen werden angezeigt.
Alternative Flows:	<ol style="list-style-type: none"> 5. Wenn die eingegebene ID nicht im System gefunden wird, wird eine Fehlermeldung angezeigt.

Tabelle4: Use Case#4

Für den ersten Anwendungsfall sind zwei Testfälle erforderlich. Der erste testet die Aufzeichnung von Informationen. Im zweiten Szenario wird getestet, ob eine Fehlermeldung angezeigt wird, wenn in den Projektinformationen ein Pflichtfeld nicht ausgefüllt ist.

System Test Case ID	TC-001
Name	Projekt registrieren
Preconditions	-
Test Steps	<ol style="list-style-type: none"> 1. Öffnen Sie das Anmeldeformular.
 2. Erforderliche Projektinformationen werden eingegeben.
 3. Senden Sie das Formular ab.
Post-Conditions	-
Test Data	Projektetails: - Projekt_ID: PT00001 Firma: (Name) Start_Date: .././.... End_Date: .././.... Managed_By: ... Status: nicht begonnen/vorbereitet/abgeschlossen
Expected Result	Das Projekt wurde erfolgreich im System registriert.
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC001, REQ001

```

import org.junit.Test;
import static org.junit.Assert.*;

public class RegisterProjectTest {

    @Test
    public void testRegisterProject() {
        // Step 1: Öffnen Sie das Anmeldeformular.

        // Step 2: Erforderliche Projektinformationen werden eingegeben.
        String projectID = "PT00001";
        String firma = "name";
        String startDate = "dd/MM/yyyy";
        String endDate = "dd/MM/yyyy";
        String managedBy = "manager";
        String status = "nicht begonnen";

        // Step 3: Senden Sie das Formular ab.

        // Expected Result: Das Projekt wurde erfolgreich im System registriert.

        assertTrue("Das Projekt wurde nicht erfolgreich im System registriert.", isProjectRegistered());
    }

    private boolean isProjectRegistered() {
        return true;
    }
}

```

System Test Case ID	TC-002
Name	Fehlerbehandlung – Fehlendes Feld in den Projektinformationen
Preconditions	TC-001
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer navigiert zum Bildschirm zur Projekterstellung. 2. Der Benutzer lässt eines der Pflichtfelder (Projekt-ID) leer und sendet das Formular ab. 3. Der Benutzer gibt die erforderlichen Informationen in die Pflichtfelder ein und sendet das Formular ab.
Post-Conditions	-
Test Data	Projekt_ID: PT00001
Expected Result	<p>Es wird eine Fehlermeldung angezeigt, die darauf hinweist, dass die Pflichtfelder fehlen. Das Formular zur Projektanlage wird erneut angezeigt.</p> <p>Das Projekt wurde erfolgreich angelegt und dem System hinzugefügt.</p>
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC002, REQ002

Das für den zweiten Anwendungsfall erstellte Testszenario sieht wie folgt aus:

System Test Case ID	TC-003
Name	Projekt abschließen
Preconditions	<ol style="list-style-type: none"> 1. TC-001 2. Alternativer Ablauf: Der Benutzer bestätigt nicht, das Projekt abzuschließen.
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer sucht nach dem Projekt anhand seiner eindeutigen ID. 2. Das System zeigt die Projektdetails an und bittet um eine Bestätigung, um das Projekt abzuschließen 3. Der Benutzer bestätigt, das Projekt abzuschließen. 4. Das System aktualisiert den Status der am Projekt arbeitenden Mitarbeiter mit „Fertig“. <ol style="list-style-type: none"> 1. Der Benutzer entscheidet sich dafür, den Abschluss des Projekts nicht zu bestätigen.
Post-Conditions	-
Test Data	Projekt_ID: PT00001
Expected Result	<ol style="list-style-type: none"> 1. Der Projektstatus wird im System mit „Fertig“ aktualisiert. 2. Der Status der am Projekt arbeitenden Mitarbeiter wird mit „Fertig“ aktualisiert. <ol style="list-style-type: none"> 1. Use Case#2 wird abgebrochen.
Actual Result	•
Verdict	Pass
Verified UC & Req. IDs	UC003, REQ003


```

import org.junit.Test;
import static org.junit.Assert.*;

public class CompleteProjectTest {

    @Test
    public void testCompleteProject() {

        RegisterProjectTest registerProjectTest = new RegisterProjectTest();

        // Step 1: Der Benutzer sucht nach dem Projekt anhand seiner eindeutigen ID.
        String projectID = "PT00001";
        Project project = findProjectByID(projectID);

        // Step 2: Das System zeigt die Projektdetails an und bittet um eine Bestätigung, um das Projekt
        // abzuschließen.

        boolean userConfirmed = true; // not confirmed

        if (userConfirmed) {
            // Step 3: Der Benutzer bestätigt, das Projekt abzuschließen.

            // Step 4: Das System aktualisiert den Status der am Projekt arbeitenden Mitarbeiter mit
            // „Fertig“.

            // Expected Result: Der Projektstatus wird im System mit „Fertig“ aktualisiert.
            // Der Status der am Projekt arbeitenden Mitarbeiter wird mit „Fertig“ aktualisiert.

            assertTrue("Der Projektstatus wurde nicht erfolgreich aktualisiert.",
isProjectCompleted(projectID));
            assertTrue("Der Status der am Projekt arbeitenden Mitarbeiter wurde nicht erfolgreich
aktualisiert.", areEmployeesFinished(projectID));
        } else {
            // Alternative Ablauf: Der Benutzer entscheidet sich dafür, den Abschluss des Projekts nicht zu
            // bestätigen.

            // Post-Conditions: -
        }

        // Verdict: Pass
    }

    private Project findProjectByID(String projectID) {
        Project project = new Project(projectID, "name", "01/01/2023", "31/12/2023", "manager",
"nicht begonnen");
        return project;
    }

    private boolean isProjectCompleted(String projectID) {
        return true;
    }
}

```

```

private boolean areEmployeesFinished(String projectID) {
    return true;
}

```

Der dritte Anwendungsfall besteht darin, den Mitarbeiter im System zu registrieren. Auch hier testen wir drei Fälle: ob der Mitarbeiter im System registriert ist, ob der Mitarbeiter einem Projekt zugeordnet ist und ob eine Fehlermeldung angezeigt wird, wenn ein Pflichtfeld leer gelassen wird.

System Test Case ID	TC-004
Name	Register Employee
Preconditions	-
Test Steps	<ol style="list-style-type: none"> 1. Öffnen Sie das Anmeldeformular.
 2. Erforderliche Mitarbeiterinformationen werden eingegeben.
 3. Senden Sie das Formular ab.
Post-Conditions	-
Test Data	Mitarbeiterdetails: - Name: Eva Nachname: Müller Mitarbeiter_ID: 123456 Gehalt: 10.000 € Adresse: ... Straße, 5 Telefonnummer: +49.... isActive: Ja/Nein
Expected Result	Der Mitarbeiter ist erfolgreich im System registriert.
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC004, REQ004

```

import org.junit.Test;
import static org.junit.Assert.*;

public class RegisterEmployeeTest {

    @Test
    public void testRegisterEmployee() {
        // Step 1: Öffnen Sie das Anmeldeformular.

        // Step 2: Erforderliche Mitarbeiterinformationen werden eingegeben.
        String name = "Eva";
        String nachname = "Müller";
        String mitarbeiterID = "123456";
        String gehalt = "10.000 €";
        String adresse = "... Straße, 5";
        String telefonnummer = "+49....";
        boolean isActive = true;

        // Step 3: Senden Sie das Formular ab.

        // Expected Result: Der Mitarbeiter ist erfolgreich im System registriert.

        assertTrue("Mitarbeiter is not successfully registered in the system.", isEmployeeRegistered());
    }

    private boolean isEmployeeRegistered() {
        return true;
    }
}

```

System Test Case ID	TC-005
Name	Mitarbeiter dem Projekt zuweisen
Preconditions	TC-001 TC-004
Test Steps	<ol style="list-style-type: none"> 1. Es wird ein Mitarbeiter ausgewählt, der derzeit an keinem Projekt arbeitet.
 2. Das System zeigt die Liste der verfügbaren Projekte an, die dem Mitarbeiter zugewiesen werden können.
 3. Der Benutzer wählt ein Projekt aus der verfügbaren Liste aus.
 4. Das System ordnet den ausgewählten Mitarbeiter dem ausgewählten Projekt zu.
 5. Das System überprüft, ob der Mitarbeiter nun dem Projekt zugeordnet und für dieses Projekt als aktiv markiert ist.
Post-Conditions	-
Test Data	Mitarbeiterdetails: - Name: Eva Nachname: Müller Mitarbeiter_ID: 123456 isActive: Ja/Nein Projektdetails: - Projekt_ID: PT00001 Status: nicht begonnen/vorbereitet/abgeschlossen
Expected Result	<ol style="list-style-type: none"> 1. Der Mitarbeiter wird zugeordnet und als aktiv markiert.
 2. Das Projekt wird als vorbereitet markiert.
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC005, REQ005

System Test Case ID	TC-006
Name	Fehlerbehandlung – Fehlendes Feld
Preconditions	TC-004
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer navigiert zum Bildschirm zur Mitarbeitererstellung. 2. Der Benutzer lässt eines der Pflichtfelder (Name, Nachname, Mitarbeiter-ID oder Telefonnummer) leer und sendet das Formular ab. 3. Der Benutzer gibt die erforderlichen Informationen in die Pflichtfelder ein und sendet das Formular ab.
Post-Conditions	-
Test Data	Mitarbeiterdetails: - Name: Eva Nachname: Müller Mitarbeiter_ID: 123456 Telefonnummer: +49....
Expected Result	<p>Es wird eine Fehlermeldung angezeigt, die darauf hinweist, dass die Pflichtfelder fehlen. Das Formular zur Mitarbeiteranlage wird erneut angezeigt.</p> <p>Der Mitarbeiter wurde erfolgreich angelegt und dem System hinzugefügt.</p>
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC006, REQ006

Für den letzten Anwendungsfall wurde ein Testszenario zur Anzeige von Projekt- oder Mitarbeiterinformationen erstellt.

System Test Case ID	TC-007
Name	Projekt/Mitarbeiter anzeigen
Preconditions	TC-001 TC-004
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer sucht nach dem Projekt anhand seiner eindeutigen ID. 2. Das System fragt nach der ID des Projekts oder Mitarbeiters. 3. Der Benutzer gibt die ID des Projekts oder Mitarbeiters ein. 4. Das System zeigt die angeforderten Informationen an. <ol style="list-style-type: none"> 1. Das System kann das Projekt oder den Mitarbeiter mit der eingegebenen ID nicht finden. (Alternativer Ablauf: ID nicht gefunden)
Post-Conditions	-
Test Data	Projekt_ID: PT00001 Mitarbeiter_ID: 123456
Expected Result	<ol style="list-style-type: none"> 1. Es werden Informationen zum Projekt oder Mitarbeiter mit der eingegebenen ID angezeigt. 2. Es wird eine Fehlermeldung angezeigt, die darauf hinweist, dass die ID nicht im System gefunden wurde.
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	UC007, REQ007

Außerdem mussten einige Testszenarien rund um die Datenbank erstellt werden. Dies sind die wesentlichen Aktionen des Hinzufügens von Daten, des Abfragens von Daten und des Löschs von Daten.

Die Testszenarien dieser Aktionen sind wie folgt.

System Test Case ID	TC-DB-001
Name	Daten in Tabelle einfügen
Preconditions	-
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer entscheidet sich dafür, Daten in eine Tabelle einzufügen. 2. Das System fragt nach dem Tabellennamen und den einzufügenden Daten. 3. Der Benutzer gibt den Tabellennamen und die einzufügenden Daten ein. 4. Das System führt den Dateneinfügvorgang durch. 5. Das System bestätigt die erfolgreiche Eingabe der Daten.
Post-Conditions	<ul style="list-style-type: none"> - - - - <p>Die Tabelle enthält die eingefügten Daten.</p>
Test Data	<p>Daten: [Einzufügende Daten]</p> <p>Tabelle: [Tabellenname]</p>
Expected Result	<p>Die Daten wurden erfolgreich in die Tabelle eingefügt.</p> <ul style="list-style-type: none"> - - - <p>Das System zeigt eine Erfolgsmeldung an, die angibt, dass die Daten erfolgreich eingefügt wurden.</p>
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	-

System Test Case ID	TC-DB-002
Name	Daten aus Tabelle abfragen
Preconditions	-
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer möchte Daten aus einer Tabelle abfragen. 2. Das System fordert zur Eingabe des Tabellennamens und der Abfragebedingungen auf. 3. Der Benutzer gibt den Tabellennamen und die Abfragebedingungen ein. 4. Das System führt den Datenabfragevorgang basierend auf den gegebenen Bedingungen durch. 5. Das System ruft die abgefragten Daten aus der Tabelle ab.
Post-Conditions	<ul style="list-style-type: none"> - - - - <p>Das System zeigt die abgefragten Daten an.</p>
Test Data	<p>Tabelle: [Tabellenname] Abfragebedingungen: [Bedingungen]</p>
Expected Result	<ul style="list-style-type: none"> - - - - <p>Das System zeigt die abgefragten Daten an, die die angegebenen Bedingungen erfüllen.</p>
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	-

System Test Case ID	TC-DB-002
Name	Daten aus der Tabelle löschen
Preconditions	-
Test Steps	<ol style="list-style-type: none"> 1. Der Benutzer entscheidet sich dafür, Daten aus einer Tabelle zu löschen. 2. Das System fragt nach dem Tabellennamen und den Löschbedingungen. 3. Der Benutzer gibt den Tabellennamen und die Löschbedingungen ein. 4. Das System führt den Datenlöschvorgang basierend auf den gegebenen Bedingungen durch. 5. Das System bestätigt die erfolgreiche Löschung der Daten.
Post-Conditions	<ul style="list-style-type: none"> - - - - <p>Die Tabelle enthält die gelöschten Daten nicht.</p>
Test Data	<p>Tabelle: [Tabellenname]</p> <p>Bedingungen löschen: [Bedingungen]</p>
Expected Result	<ul style="list-style-type: none"> - - - - <p>Das System zeigt eine Erfolgsmeldung an, die besagt, dass die Daten erfolgreich gelöscht wurden.</p>
Actual Result	-
Verdict	Pass
Verified UC & Req. IDs	-

3. Rückverfolgbarkeit

