



AutoXpert Systemtestspezifikation

Mücahid Eren KOÇ

e190503044@stud.tau.edu.tr

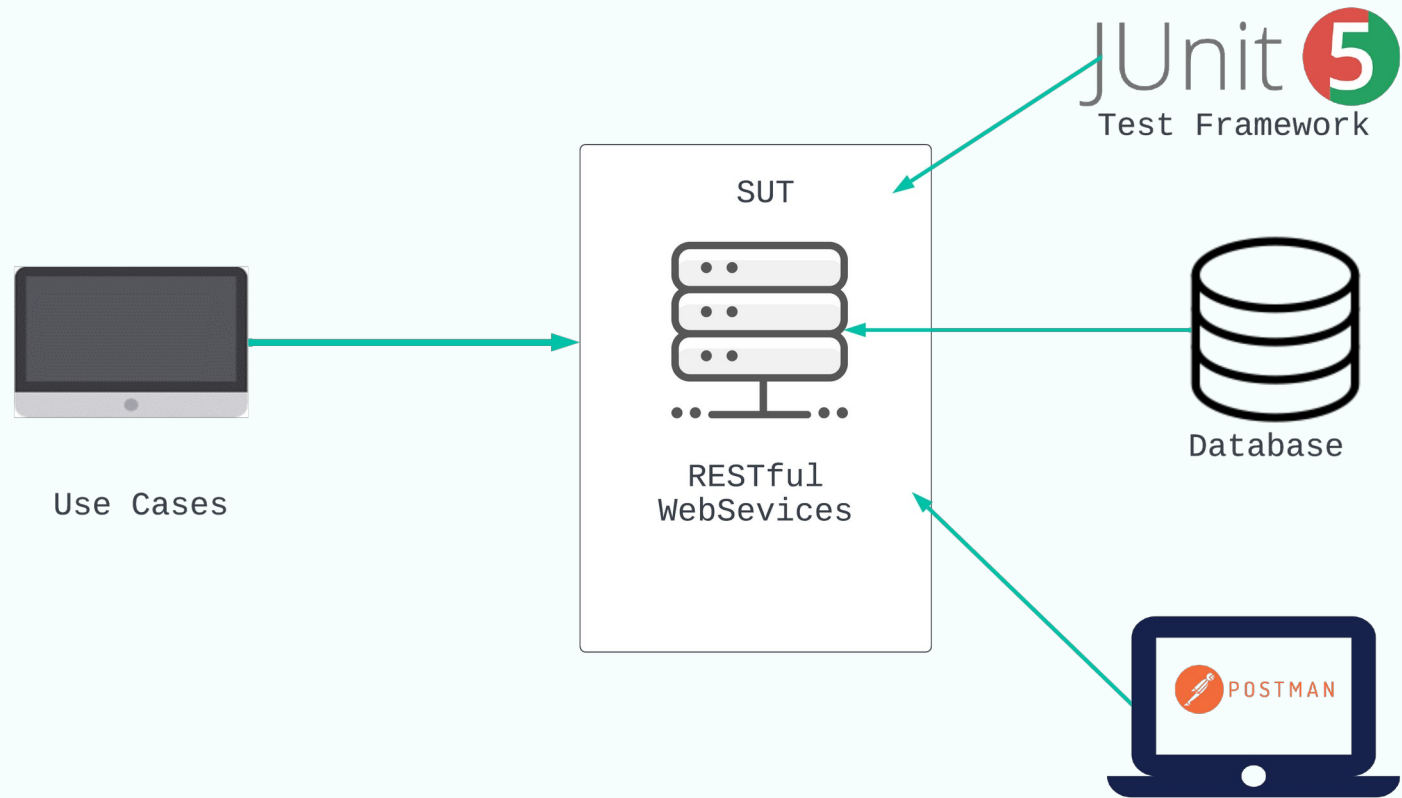
İbrahim Alper ÖLÜÇ

e200503016@stud.tau.edu.tr

Stakeholder: DI Ömer KARACAN

omer.karacan@tau.edu.tr

Systemüberblick



SYSTEMTESTFÄLLE – Standart Benutzer Erstellen

System Test Case ID Name	S_TC_1 Test Standart Benutzer erstellen via POST - Create_StandartBenutzer_by_Admin Service
Preconditions	<p>Benutzer Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: POST Create_StandartBenutzer_by_Admin (Benutzer standartbenutzer),• Response: Ein Benutzer-Objekt mit Benutzer Attribute<ul style="list-style-type: none">• Int BenutzerID<generiert>• String ID*• String Password (Verschlüsselt)• String name• String nachname <p>*Dieses id ist ein username aber vom Unternehmen gegeben und verschieden von BenutzerID</p> <ul style="list-style-type: none">• App Port Nummer: 8080(default)• Der Endpunkt "Benutzer/UserCreate" ist erreichbar.• Passwort wurde richtig verschlüsselt.• Der Benutzer Service ist richtig konfiguriert und wurde mit der Datenbank verbunden• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine POST-Anfrage an den Endpunkt "Benutzer/UserCreate" mit einem gültigen StandartBenutzer-Objekt in dem Anfragetext.2. Überprüfe, ob der Antwortstatuscode 201(CREATED) ist.3. Überprüfe, ob der Antworttext das gespeicherte Benutzerobjekt enthält.4. Sende noch eine POST-Anfrage an den Endpunkt "Benutzer/UserCreate" mit dasselbe StandartBenutzer-Objekt in dem Anfragetext.5. Überprüfe, ob der Antwortstatuscode 409(CONFLICT) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) "Der Benutzer ist bereits vorhanden" enthält.
Post-Conditions	Das Benutzer-Objekt wird in der Datenbank gespeichert.

SYSTEMTESTFÄLLE – Standart Benutzer Erstellen

Test Data	Barry, Allen, @XSB#23001, autoXpert/sb/1
Expected Result	Benutzer StandartBenutzer <ul style="list-style-type: none">• BenutzerID<generiert>• ID = "@XSB#23001"• Password = autoXpert/sb/1(Verschlüsselte Form)• name = "Barry"• nachname = "Allen"
Actual Result	Benutzer StandartBenutzer <ul style="list-style-type: none">• BenutzerID<generiert>• ID = "@XSB#23001"• Password = autoXpert/sb/1(Verschlüsselte Form)• name = "Barry"• nachname = "Allen"
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case UI-0 Create_StandartBenutzer_by_Admin

SYSTEMTESTFÄLLE – Login

System Test Case ID Name	S_TC_2 Test Login via POST - Login Service
Preconditions	<p>Benutzer WebService ist gegeben:</p> <ul style="list-style-type: none">• Request: POST Login*(String id**, String password),• Response: Ein Benutzer-Objekt mit einigen Benutzer Attribute<ul style="list-style-type: none">• Int BenutzerID// eindeutig <generiert>• String ID• String Password(verschlüsselt) <p>*Login → Login_as_Admin und Login_as_StandartBenutzer</p> <p>** Dieses id ist ein username aber vom Unternehmen gegeben und verschieden von BenutzerID</p> <ul style="list-style-type: none">• App Port Nummer: 8080(default)• Der Endpunkt “Benutzer/Login” ist erreichbar.• Der Benutzer Service ist richtig konfiguriert und wurde mit der Datenbank verbunden• Es gibt einen Benutzer in der Datenbank mit gültigen ID und Passwort• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine POST-Anfrage an den Endpunkt “Benutzer/Login” mit gültigen ID und Passwort Anmeldeinformationen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das vorhandene Benutzerobjekt enthält.4. Sende noch eine POST-Anfrage an den Endpunkt “Benutzer/Login” mit ungültigen ID und Passwort Anmeldeinformationen als path Variable.5. Überprüfe, ob der Antwortstatuscode 401(UNAUTHORIZED) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) “Ungültige ID oder Passwort” enthält.
Post-Conditions	<p>Der Benutzer-Objekt wird durch den ID aus der Datenbank abgerufen.</p>

SYSTEMTESTFÄLLE – Login

Test Data	@XSB#23001, autoXpert/sb/1 (Standart Benutzer), @XAB#1&1, @dmin-The1&only (Admin Benutzer)
Expected Result	Benutzer StandartBenutzer <ul style="list-style-type: none">• BenutzerID <generiert>• ID = @XSB#23001• Passwort = Verschlüsselte form von Passwort Benutzer AdminBenutzer <ul style="list-style-type: none">• BenutzerID <generiert>• ID = @XAB#1&1• Passwort = Verschlüsselte form von Passwort
Actual Result	Benutzer StandartBenutzer <ul style="list-style-type: none">• BenutzerID <generiert>• ID = @XSB#23001• Passwort = Verschlüsselte form von Passwort Benutzer AdminBenutzer <ul style="list-style-type: none">• BenutzerID <generiert>• ID = @XAB#1&1• Passwort = Verschlüsselte form von Passwort
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case UI-0 Login (Login_as_StandartBenutzer & Login_as_Admin)

SYSTEMTESTFÄLLE – Kunde Erstellen

System Test Case ID Name	S_TC_3 Test Kunde erstellen via POST - KundeErstellen Service
Preconditions	<p>Kunde Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: POST KundeErstellen*(Kunde kunde, Fahrzeug fahrzeug, Schade schade),• Response: Ein Kunde-Objekt mit;<ul style="list-style-type: none">• Int KundenID<generiert>• String name• String nachname• String Tel• String Adresse• String Kfz-Kennzeichen (von Fahrzeug-Objekt im Kunde-Objekt) <p>diese Attribute</p> <p>*Kunden werden mit ihren Fahrzeuge und Schaden von ihren Fahrzeuge eingefügt.</p> <ul style="list-style-type: none">• App Port Nummer: 8080(default)• Der Endpunkt “Kunde/einfügen” ist erreichbar.• Der Kunde Service ist richtig konfiguriert und wurde mit der Datenbank verbunden• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine POST-Anfrage an den Endpunkt “Kunde/einfügen” mit einem gültigen Kunde-Objekt in dem Anfragetext.2. Überprüfe, ob der Antwortstatuscode 201(CREATED) ist.3. Überprüfe, ob der Antworttext das gespeicherte Kunde-Objekt enthält.4. Sende noch eine POST-Anfrage an den Endpunkt “Kunde/einfügen” mit dasselbe Kunde-Objekt in dem Anfragetext.5. Überprüfe, ob der Antwortstatuscode 409(CONFLICT) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) “Der Kunde ist bereits vorhanden” enthält.
Post-Conditions	<p>Das Kunde-Objekt wird in der Datenbank gespeichert.</p>

SYSTEMTESTFÄLLE – Kunde Erstellen

Test Data	Bruce, Wayne,Tel, Adresse, Fahrzeug, Schaden
Expected Result	Kunde kunde <ul style="list-style-type: none">• KundenID<generiert>• name = "Bruce"• nachname = "Wayne"• Tel = "555 - 01051939"• Adresse = "Wayne Manor, Gotham"• Kfz-Kennzeichen = "135 8024 Gotham"
Actual Result	Kunde kunde <ul style="list-style-type: none">• KundenID<generiert>• name = "Bruce"• nachname = "Wayne"• Tel = "555 - 01051939"• Adresse = "Wayne Manor, Gotham"• Kfz-Kennzeichen = "135 8024 Gotham"
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case UI-1 & UI-2 Kunde Erstellen

SYSTEMTESTFÄLLE – SucheFahrzeug

System Test Case ID Name	S_TC_4 Test Fahrzeug Suche via GET-SucheFahrzeug Service
Preconditions	<p>Fahrzeug Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: GET SucheFahrzeug(String Kfz-Kennzeichen),• Response: Ein Fahrzeug Objekt mit einige Fahrzeug Attribute<ul style="list-style-type: none">• Int FahrzeugID // eindeutig <generiert>• String Kfz-Kennzeichen• App Port Nummer: 8080(default)• Der Endpunkt «Fahrzeug/Fahrzeugen/{Kfz-Kennzeichen}» ist erreichbar.• Der Fahrzeug Service ist richtig konfiguriert und wurde mit der Datenbank verbunden• Es gibt einen Fahrzeug in der Datenbank mit einem gültigen Kfz-Kennzeichen• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine GET-Anfrage an den Endpunkt «Fahrzeugen/{Kfz-Kennzeichen}» mit einem gültigen Kfz-Kennzeichen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das Fahrzeugobjekt mit dem angegebenen Kfz-Kennzeichen enthält.4. Sende noch eine GET-Anfrage an den Endpunkt «Fahrzeugen/{Kfz-Kennzeichen}» mit einem ungültigen Kfz-Kennzeichen als path Variable.5. Überprüfe, ob der Antwortstatuscode 404(NOT FOUND) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldng) «Fahrzeug nicht gefunden!» enthält.
Post-Conditions	<p>Der Fahrzeug-Objekt wird durch den Kfz-Kennzeichen aus der Datenbank abgerufen.</p>

SYSTEMTESTFÄLLE – SucheFahrzeug

Test Data	nichts
Expected Result	Fahrzeug Objekt <ul style="list-style-type: none">• FahrzeugID = <generiert>• Kfz-Kennzeichen = "135 8024 Gotham"
Actual Result	Fahrzeug Objekt <ul style="list-style-type: none">• FahrzeugID = <generiert>• Kfz-Kennzeichen = "135 8024 Gotham"
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case API-1 SucheFahrzeug

SYSTEMTESTFÄLLE – FahrzeugInformationen präsentieren

System Test Case ID Name	S_TC_5 Test Fahrzeug informationen via GET Fahrzeuginformationen-Service
Preconditions	<p>Fahrzeug Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: GET FahrzeugInformationen (String Kfz-Kennzeichen),• Response:<ul style="list-style-type: none">• Ein Fahrzeug Objekt mit alle Fahrzeug Attribute außer KundelID, SchadelID und FahrzeugID<ul style="list-style-type: none">• Int FahrzeugID // eindeutig <generiert>• String Kfz-Kennzeichen• ...• Besitzer des Autos mit Attiributes<ul style="list-style-type: none">• String Name• String Nachname• String Telefonnummer• String Adresse• App Port Nummer: 8080(default)• Der Endpunkt «Fahrzeug/Fahrzeugen/ansehen/{Kfz-Kennzeichen}» ist erreichbar.• Der Fahrzeug Service ist richtig konfiguriert und wurde mit der Datenbank verbunden.• Es gibt einen Fahrzeug in der Datenbank mit einem gültigen Kfz-Kennzeichen• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine GET-Anfrage an den Endpunkt «Fahrzeug/Fahrzeugen/ansehen/{Kfz-Kennzeichen}» mit einem gültigen Kfz-Kennzeichen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das Fahrzeugobjekt mit dem angegebenen Kfz-Kennzeichen und Besitzer mit angegebenen Attiributes enthält.4. Sende noch eine GET-Anfrage an den Endpunkt «Fahrzeug/Fahrzeugen/ansehen/{Kfz-Kennzeichen}» mit einem ungültigen Kfz-Kennzeichen als path Variable.5. Überprüfe, ob der Antwortstatuscode 404(NOT FOUND) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) «Fahrzeug nicht gefunden!» enthält.
Post-Conditions	<p>Der Fahrzeug-Objekt und Besitzer wird durch den Kfz-Kennzeichen aus der Datenbank abgerufen.</p>

SYSTEMTESTFÄLLE – FahrzeugInformationen präsentieren

Test Data	nichts
Expected Result	<p>Fahrzeug Objekt</p> <ul style="list-style-type: none">• Int FahrzeugID : <generiert>• Kfz-Kennzeichen: «“135 8024 Gotham”• ... <p>Kunde kunde</p> <ul style="list-style-type: none">• KundenID<generiert>• name = “Bruce”• nachname = “Wayne”• Tel = “555 - 01051939”• Adresse = “Wayne Manor, Gotham”• Kfz-Kennzeichen = “135 8024 Gotham”
Actual Result	<p>Fahrzeug Objekt</p> <ul style="list-style-type: none">• Int FahrzeugID : <generiert>• Kfz-Kennzeichen: «“135 8024 Gotham”• ... <p>Kunde kunde</p> <ul style="list-style-type: none">• KundenID<generiert>• name = “Bruce”• nachname = “Wayne”• Tel = “555 - 01051939”• Adresse = “Wayne Manor, Gotham”• Kfz-Kennzeichen = “135 8024 Gotham”
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case API-1 Fahrzeug informationen präsentieren

SYSTEMTESTFÄLLE – Wartungszustand aktualisieren

System Test Case ID Name	S_TC_6 Test Wartungszustand aktualisieren via PUT - Wartungszustand aktualisieren Service
Preconditions	<p>Fahrzeug Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: PUT WartungszustandAktualisieren(String KFZ-Kennzeichen, String Wartungszustand)• Response: Ein Fahrzeug-Objekt mit einigen Attribute<ul style="list-style-type: none">• Int FahrzeugID// eindeutig <generiert>• String KFZ-Kennzeichen• String Wartungszustand• App Port Nummer: 8080(default)• Der Endpunkt “Fahrzeugen /aktualisieren / {Kfz-Kennzeichen}” ist erreichbar.• Der FahrzeugService ist richtig konfiguriert und wurde mit der Datenbank verbunden• Es gibt einen Fahrzeug in der Datenbank mit eingegebene Kennzeichen.• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine PUT-Anfrage an den Endpunkt «Fahrzeugen /aktualisieren / {Kfz-Kennzeichen}» mit einem gültigen Kfz-Kennzeichen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das Fahrzeugobjekt mit dem angegebenen Kfz-Kennzeichen und für gegebene Attribute Änderungen enthält.4. Sende noch eine PUT-Anfrage an den Endpunkt «Schaden /aktualisieren / {Kfz-Kennzeichen}” mit einem ungültigen Kfz-Kennzeichen(die nicht existiert) als path Variable.5. Überprüfe, ob der Antwortstatuscode 404(NOT FOUND) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) «Fahrzeug nicht gefunden!» enthält.
Post-Conditions	<p>Der Fahrzeug-Objekt mit gegebene KFZ-Kennzeichen wird in der Datenbank aktualisiert.</p>

SYSTEMTESTFÄLLE – Wartungszustand aktualisieren

Test Data	Aktuelle Wartungszustand von Fahrzeug
Expected Result	Fahrzeug fahrzeug <ul style="list-style-type: none">• FahrzeugID: <generiert>• KFZ-Kennzeichen: "135 8024 Gotham"• Wartungszustand: «Repariert»
Actual Result	Fahrzeug fahrzeug <ul style="list-style-type: none">• FahrzeugID: <generiert>• KFZ-Kennzeichen: "135 8024 Gotham"• Wartungszustand: «Repariert»
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case UI-3 Wartungszustand aktualisieren

SYSTEMTESTFÄLLE – SchadenInformationen Aktualisieren

System Test Case ID Name	S_TC_7 Test SchadenInfoAktualieseren via PUT - SchadenInfoAktualieseren Service
Preconditions	<p>Schaden WebService ist gegeben:</p> <ul style="list-style-type: none">• Request: PUT SchadenInfoAktualisieren(String KFZ-Kennzeichen, String Chassis_und_Karossiere, String Motor,, String Getriebe, String Reifen_und_Bremsen)• Response: Ein Schade-Objekt mit einigen Schade Attribute<ul style="list-style-type: none">• Schade SchadeID// eindeutig <generiert>• String KFZ-Kennzeichen• String Chassis_und_Karossiere• String Motor• String Getriebe• String Reifen_und_Bremsen• App Port Nummer: 8080(default)• Der Endpunkt “Schaden /aktualisieren / {Kfz-Kennzeichen}” ist erreichbar.• Der SchadenService ist richtig konfiguriert und wurde mit der Datenbank verbunden• Es gibt einen Schade in der Datenbank mit eingegebene Kennzeichen.• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine PUT-Anfrage an den Endpunkt «Schaden /aktualisieren / {Kfz-Kennzeichen}» mit einem gültigen Kfz-Kennzeichen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das Schadeobjekt mit dem angegebenen Kfz-Kennzeichen und Änderungen für gegebene Attributen enthält.4. Sende noch eine PUT-Anfrage an den Endpunkt «Schaden /aktualisieren / {Kfz-Kennzeichen}” mit einem ungültigen Kfz-Kennzeichen als path Variable.5. Überprüfe, ob der Antwortstatuscode 404(NOT FOUND) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldng) «Schade nicht gefunden!» enthält.
Post-Conditions	<p>Der Schade-Objekt mit gegebene KFZ-Kennzeichen wird in der Datenbank aktualisiert.</p>

SYSTEMTESTFÄLLE – SchadenInformationen Aktualisieren

Test Data	«NULL», «NULL», «Beschädigte Kolben», NULL, «Beläge abgenutzt»
Expected Result	Schade schade <ul style="list-style-type: none">• SchadelID: <generiert>• KFZ-Kennzeichen: “135 8024 Gotham”• Chassis_und_Karosserie: NULL• String Motor: Beschädigte Kolben• String Getriebe: NULL• String Reifen_und_Bremsen: Beläge abgenutzt
Actual Result	Schade schade <ul style="list-style-type: none">• SchadelID: <generiert>• KFZ-Kennzeichen: “135 8024 Gotham”• Chassis_und_Karosserie: NULL• String Motor: Beschädigte Kolben• String Getriebe: NULL• String Reifen_und_Bremsen: Beläge abgenutzt
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case API-1 Schade Informationen Aktualisieren

SYSTEMTESTFÄLLE – Reparatur&Kosten aktualisieren

System Test Case ID Name	S_TC_8 Test Reparatur&KostenAktualisieren via PUT - Reparatur&Kosten Aktualisieren Service
Preconditions	<p>Fahrzeug Webservice ist gegeben:</p> <ul style="list-style-type: none">• Request: PUT Reparatur-KostenAktualisieren(String KFZ-Kennzeichen, String Reparatur, String Kosten)• Response: Ein Fahrzeug-Objekt mit einigen Attribute<ul style="list-style-type: none">• Int FahrzeugID// eindeutig <generiert>• String KFZ-Kennzeichen• String Reparatur• String Kosten• App Port Nummer: 8080(default)• Der Endpunkt "Fahrzeugen /reparaturen / {Kfz-Kennzeichen}" ist erreichbar.• Der FahrzeugService ist richtig konfiguriert und wurde mit der Datenbank verbunden• Es gibt einen Fahrzeug in der Datenbank mit eingegebene Kennzeichen.• Das System ist aktiv und läuft gerade!
Test Steps	<ol style="list-style-type: none">1. Sende eine PUT-Anfrage an den Endpunkt «Fahrzeugen /reparaturen / {Kfz-Kennzeichen}» mit einem gültigen Kfz-Kennzeichen als path Variable.2. Überprüfe, ob der Antwortstatuscode 200(OK) ist.3. Überprüfe, ob der Antworttext das Fahrzeugobjekt mit dem angegebenen Kfz-Kennzeichen und für gegebene Attribute Änderungen enthält.4. Sende noch eine PUT-Anfrage an den Endpunkt «Schaden /aktualisieren / {Kfz-Kennzeichen}" mit einem ungültigen Kfz-Kennzeichen als path Variable.5. Überprüfe, ob der Antwortstatuscode 404(NOT FOUND) ist.6. Überprüfe, ob der Antworttext die Nachricht(oder die Meldung) «Fahrzeug nicht gefunden!» enthält.
Post-Conditions	Der Fahrzeug-Objekt mit gegebene KFZ-Kennzeichen wird in der Datenbank aktualisiert.

SYSTEMTESTFÄLLE – Reparatur&Kosten aktualisieren

Test Data	«Beläge wurden gewechselt», «100 Euro»
Expected Result	Fahrzeug fahrzeug <ul style="list-style-type: none">• FahrzeugID <generiert>• KFZ-Kennzeichen: “135 8024 Gotham”• Reparatur: Beläge wurden gewechselt• Kosten: 100 Euro
Actual Result	Fahrzeug fahrzeug <ul style="list-style-type: none">• FahrzeugID <generiert>• KFZ-Kennzeichen: “135 8024 Gotham”• Reparatur: Beläge wurden gewechselt• Kosten: 100 Euro
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	Use Case UI-4 Reparatur&Kosten aktualisieren

Component Integration Test Case ID NAME	CI_TC_Benutzer_1 Test Benutzer Database Interface
Pre-Conditions	<ul style="list-style-type: none"> Die Datenbanktabelle „User“ enthält eine Zeile für: <ul style="list-style-type: none"> BenutzerID = <generiert> ID = “@XSB#23001” Password = autoXpert/sb/1(Verschlüsselte Form) name = “Barry” nachname = “Allen Benutzer Webservice ist mocked
Test Steps	<ul style="list-style-type: none"> Versuchen Sie, den Benutzer zu erstellen mit der Methode Create_StandartBenutzer_by_Admin (Benutzer standartbenutzer), der in der Datenbank vorhanden ist. Bestätigen Sie, dass der HTTP 409-Code (Konflikt) gesendet und eine Ausnahme(Exception) dafür ausgelöst wird. Erstellen Sie mit der Methode Create_StandartBenutzer_by_Admin (Benutzer standartbenutzer) einen neuen Benutzer in der Datenbank. Authentifizieren Sie einen Benutzer mit der Methode Login() (Login_as_Admin und Login_as_StandartBenutzer). Versuchen Sie beide Login Methoden mit ungültige IDs und Passwords.
Post Conditions	nichts

Test Data	«@XSB#23001», «autoXpert/sb/1», «Barry», «Allen»
Expected Result	<ul style="list-style-type: none"> • BenutzerID<generiert> • ID = «@XSB#23001» • Password = «autoXpert/sb/1(Verschlüsselte Form)» • name = «Barry» • nachname = «Allen»
Actual Result	<ul style="list-style-type: none"> • BenutzerID<generiert> • ID = «@XSB#23001» • Password = autoXpert/sb/1(Verschlüsselte Form) • name = «Barry» • nachname = «Allen»
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	/UI-0/

Component Integration Test Case ID NAME	CI_TC_Kunde_1 Test Kunden Database Interface
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle „Kunden“ enthält eine Zeile für: KundeID = <generated> Name = «Hans» Nachname = «Klaus» Adresse = «Musterstraße / 17» Tel = «5555555555»Kunde Web Service call ist mocked
Test Steps	<ul style="list-style-type: none">Versuchen Sie, den Kunde zu erstellen mit der Methode KundeErstellen(Name, Nachname, Tel, Adresse) , der in der Datenbank vorhanden istBestätigen Sie, dass der HTTP 409-Code (Konflikt) gesendet und eine Ausnahme(Exception) dafür ausgelöst wird.Erstellen Sie mit der Methode KundeErstellen(Name, Nachname, Tel, Adresse) einen neuen Kunde in der Datenbank.Bestätigen Sie, dass der HTTP 201-Code(Created) gesendet.Versuchen Sie KundeInformation zuzubekommen mit FahrzeugInformationpräsentieren(String KFZ-Kennzeichen) und bestätigen HTTP 200-Code(Ok) und ausgegebene KundeInformationen
Post-Conditions	nichts

Test Data	«Hans», «Klaus», Adresse: «Musterstraße / 17», Tel: «5555555555»
Expected Result	KundeID = <generated> Name = «Hans» Nachname = «Klaus» Adresse = «Musterstraße / 17» Tel = «5555555555»
Actual Result	KundeID = <generated> Name = «Hans» Nachname = «Klaus» Adresse = «Musterstraße / 17» Tel = «5555555555»
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	/API-1/, /UI-1/

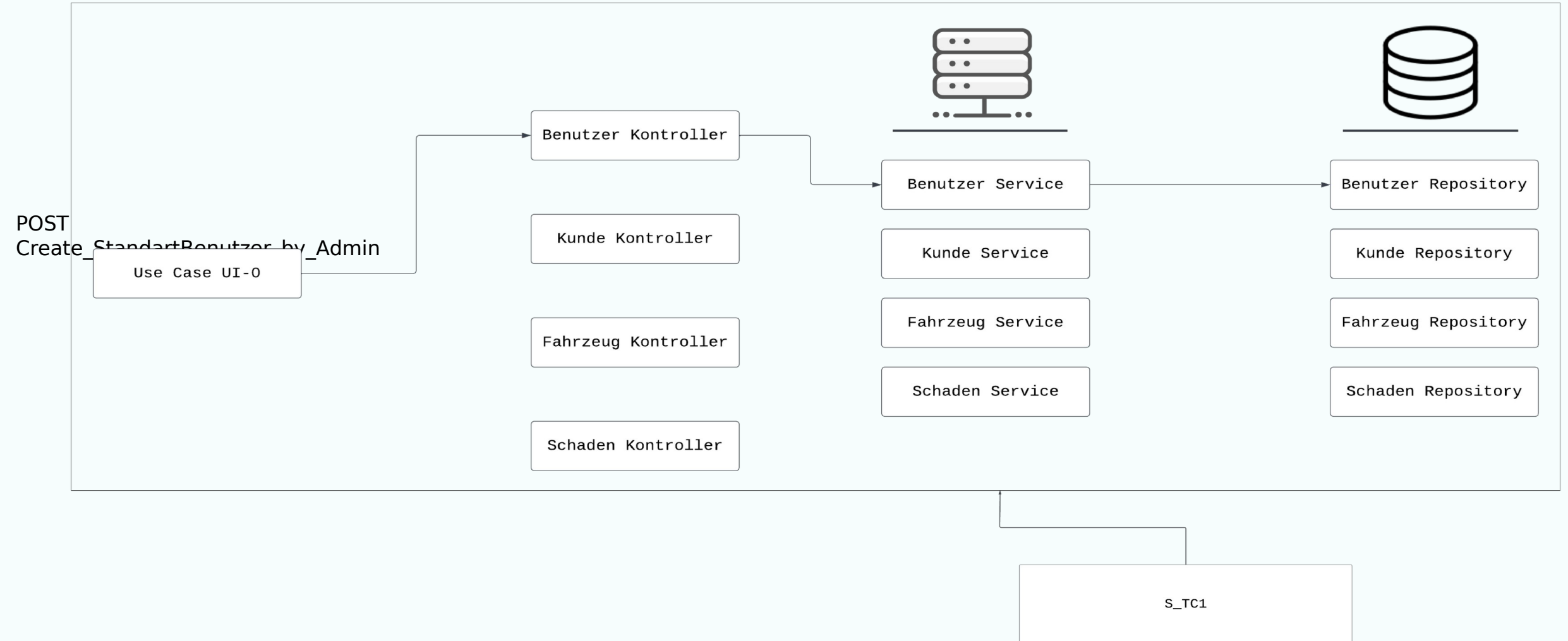
Component Integration Test Case ID NAME	CI_TC_Fahrzeug_1 Test Fahrzeug Database Interface
Preconditions	<ul style="list-style-type: none"> Die Datenbanktabelle „Fahrzeug“ enthält eine Zeile für: <ul style="list-style-type: none"> FahrzeugID = <generated> KFZ-Kennzeichen = «34 MEK 34» Mark,model,Baujahr,Farbe,KmStand,Ps,KraftStoff,Getriebe=NULL Wartezustand = «Reparatur» Reparaturen = NULL Kosten = NULL Fahrzeug Web Service call ist mocked
Test Steps	<ul style="list-style-type: none"> Rufen Sie die Methode SucheFahrzeug(KFZ-Kennzeichen) und FahrzeugInfopräsentieren(KFZ-Kennzeichen) und Bestätigen Sie, dass der HTTP 201-Code(Found) und ResponseEntity. Rufen Sie Reparatur-Kostenaktualisieren(KFZ-Kennzeichen) mit gegebene Test Data. Dann bestätigen Sie HTTP 200-Code und bestimmen die Änderungen von gegebene Attiributen. Rufen Sie WartungzustandAktualisieren(KFZ-Kennzeichen) mit gegebene Wartunzustand in Test Data. Bestatigen Sie HTTP 200-Code und bestimmen Sie Wartungzustand.
Post-Conditions	nichts

Test Data	«Beläge wurden gewechselt», «100 Euro», «Fertig»
Expected Result	<ul style="list-style-type: none"> • FahrzeugID = <generated> • KFZ-Kennzeichen = «34 MEK 34» • Mark,model,Baujahr,Farbe,KmStand,Ps,KraftStoff,Getriebe=NULL • Wartezustand = «Beläge wurden gewechselt» • Reparaturen = «Fertig» • Kosten = «100 Euro»
Actual Result	<ul style="list-style-type: none"> • FahrzeugID = <generated> • KFZ-Kennzeichen = «34 MEK 34» • Mark,model,Baujahr,Farbe,KmStand,Ps,KraftStoff,Getriebe=NULL • Wartezustand = «Beläge wurden gewechselt» • Reparaturen = «Fertig» • Kosten = «100 Euro»
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	/UI-3/, /UI-4/, /API-1/

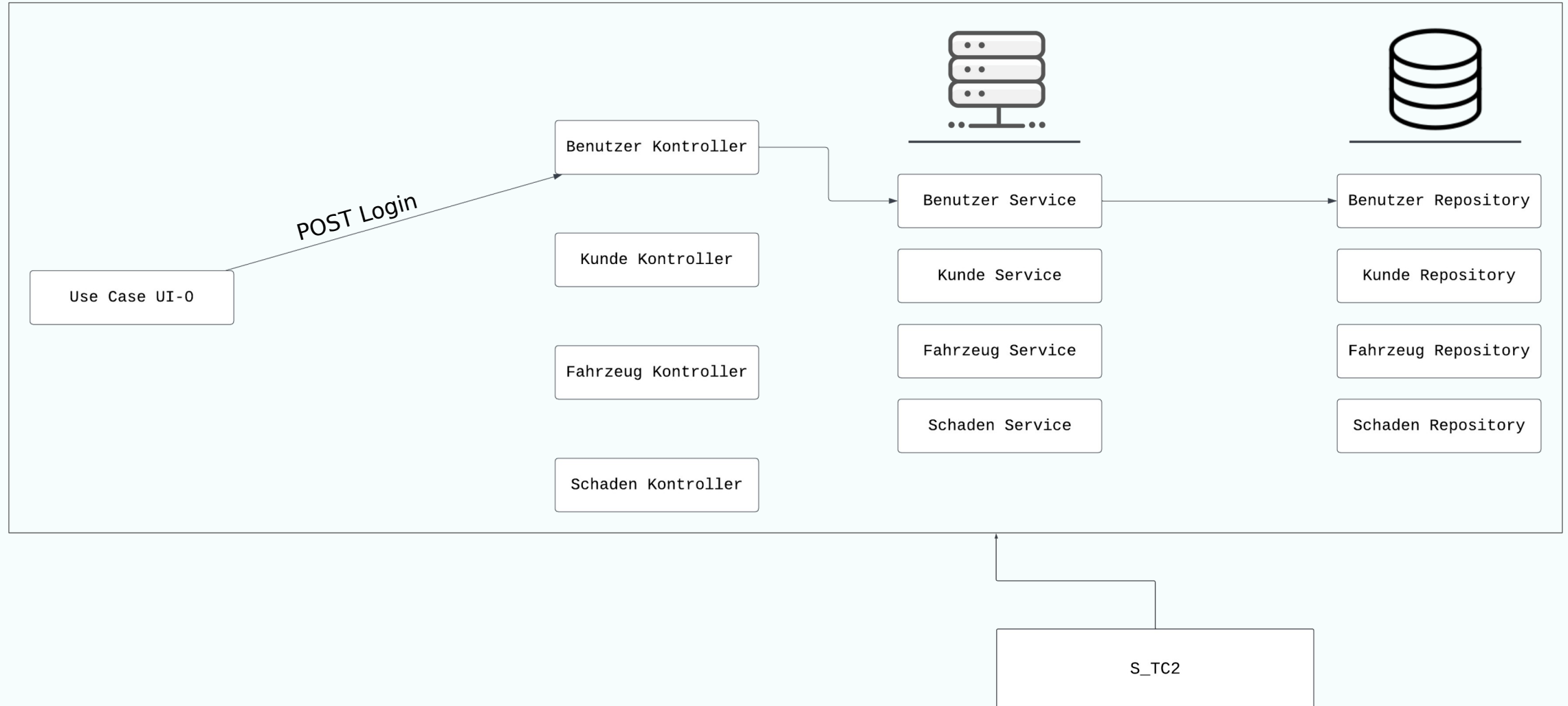
Component Integration Test Case ID NAME	CI_TC_Schade_1 Test Schade Database Interface
Preconditions	<ul style="list-style-type: none"> Die Datenbanktabelle „Benutzer“ enthält eine Zeile für: <ul style="list-style-type: none"> SchadeID:<generiert> FahrzeugID = 1 Motor= NULL Chassis_und_Karosserie = NULL Getriebe = NULL Reifen_und_Bremsen = NULL Schade Webservice ist mocked
Test Steps	<ul style="list-style-type: none"> Rufen Sie SchadeInformationenAktualisieren(KFZ-Kennzeichen) mit gegebene Test Data. Bestätigen Sie HTTP 200-Code und bestimmen die Änderungen von gegebene Attributen.
Post-Conditions	nichts

Test Data	«Beschädigte Kolben», «Beläge abgenutzt»
Expected Result	<ul style="list-style-type: none">• SchadelID:<generiert>• FahrzeugID = 1• Motor= «Beschädigte Kolben»,• Chassis_und_Karosserie = NULL• Getriebe = NULL• Reifen_und_Bremsen = «Beläge abgenutzt»
Actual Result	<ul style="list-style-type: none">• SchadelID:<generiert>• FahrzeugID = 1• Motor= «Beschädigte Kolben»,• Chassis_und_Karosserie = NULL• Getriebe = NULL• Reifen_und_Bremsen = «Beläge abgenutzt»
Verdict(Pass/Fail)	Pass
Verified Use Case & Req ID	/API-1/

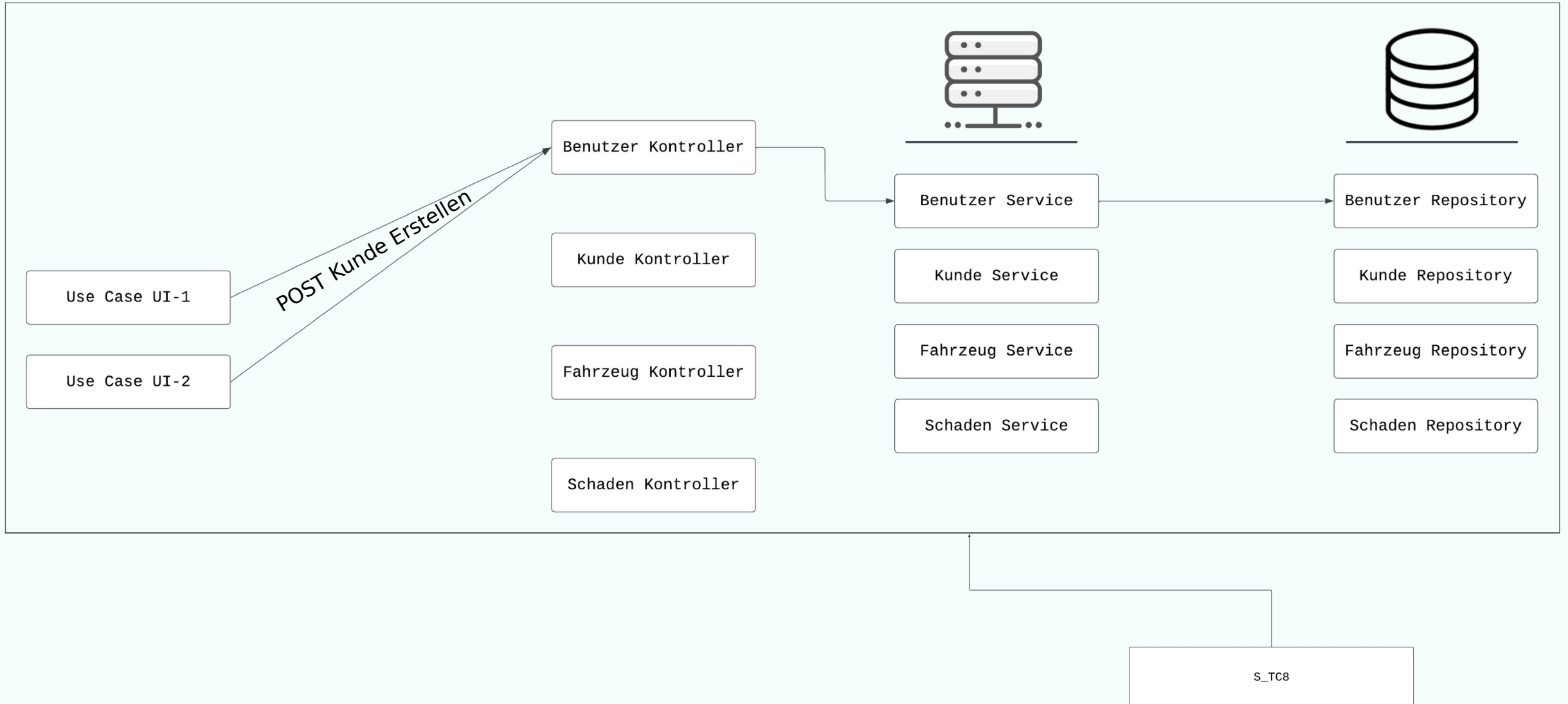
Rückverfolgbarkeit von Anforderungen



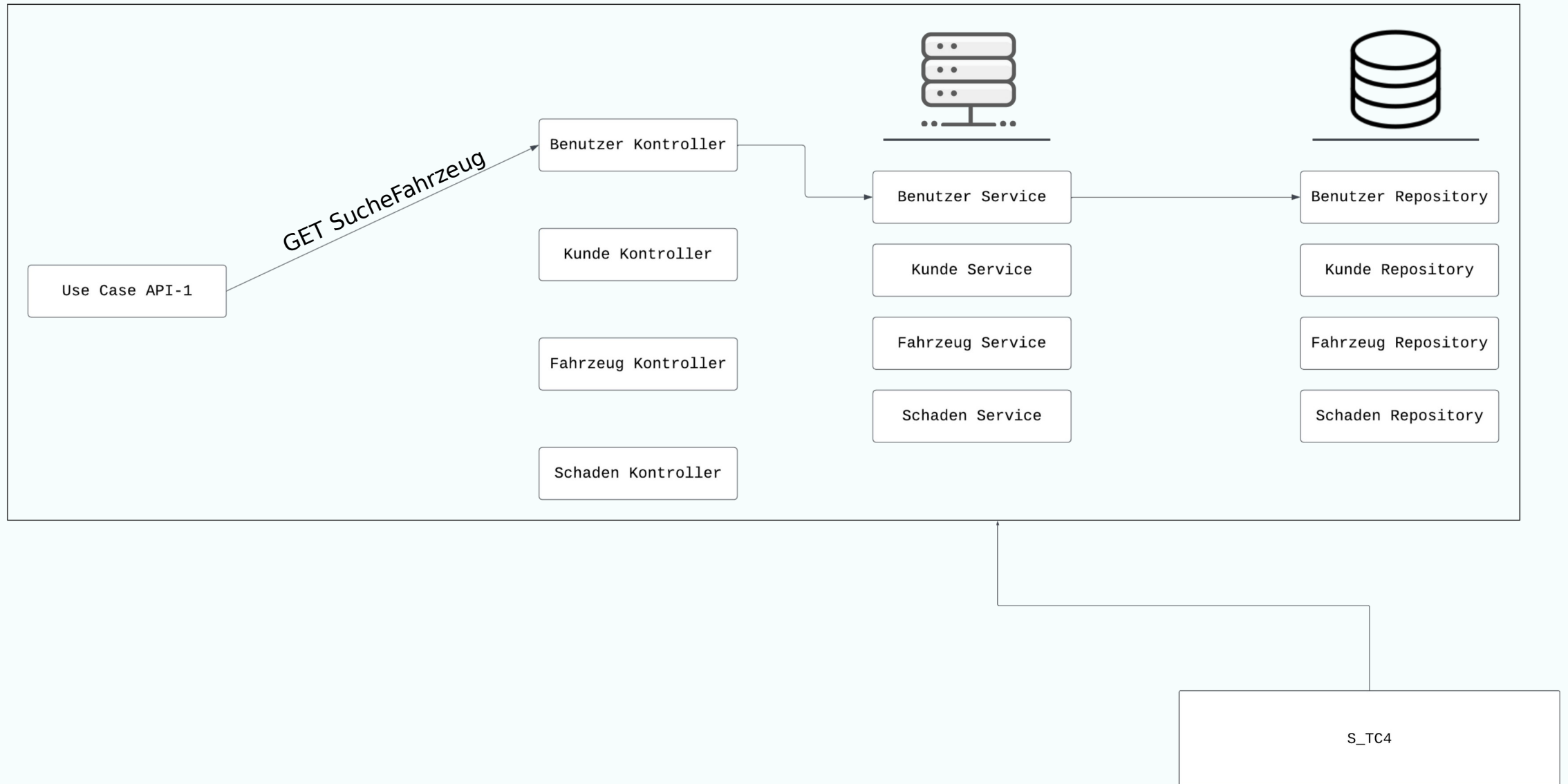
Rückverfolgbarkeit von Anforderungen



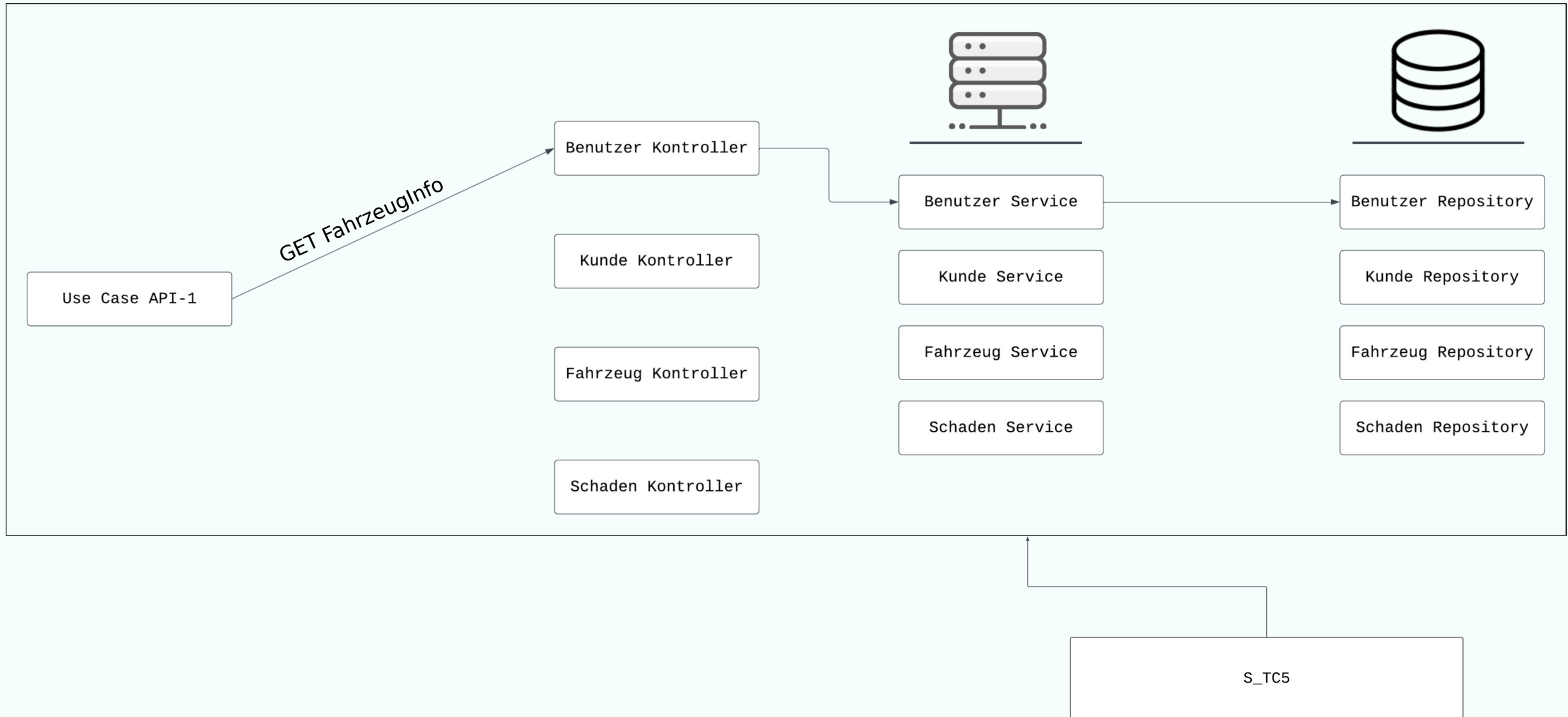
Rückverfolgbarkeit von Anforderungen



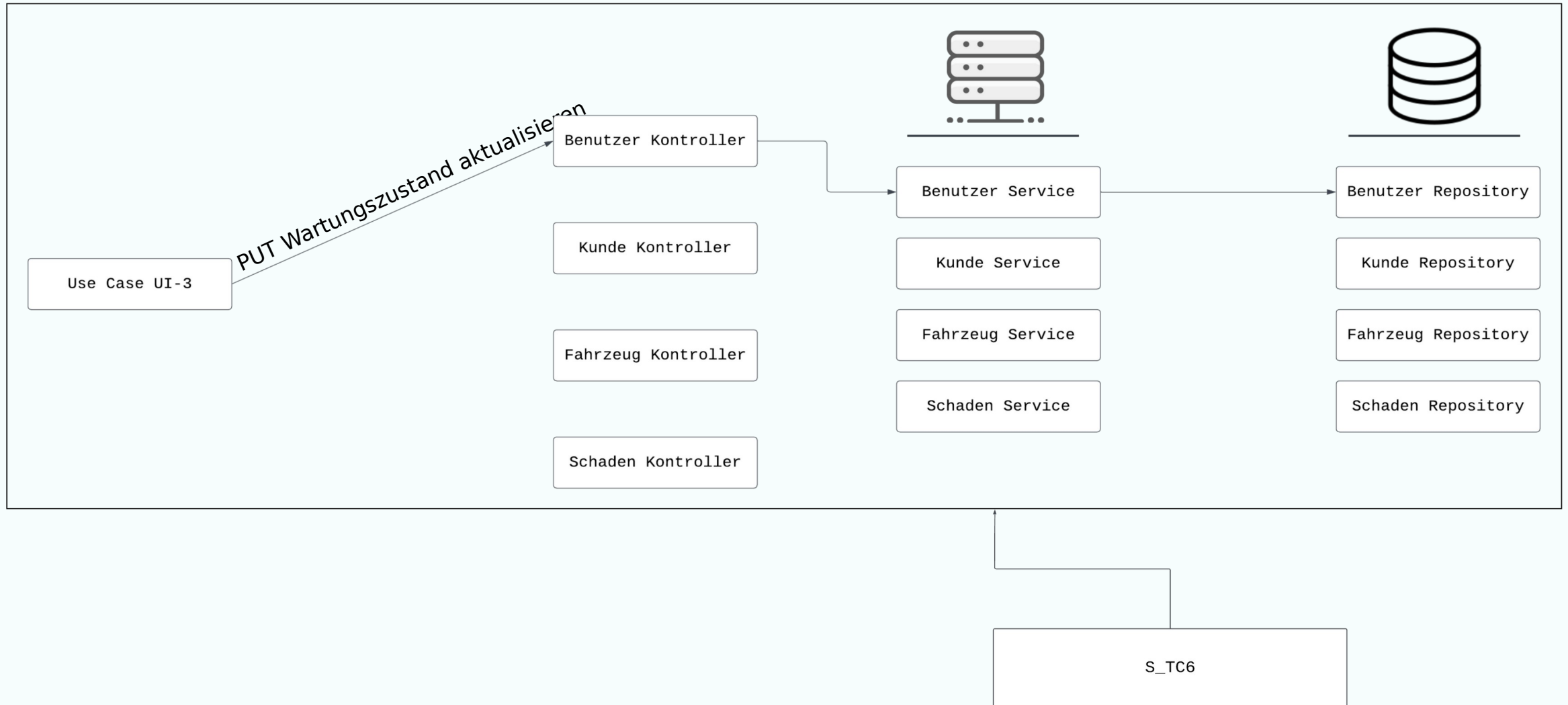
Rückverfolgbarkeit von Anforderungen



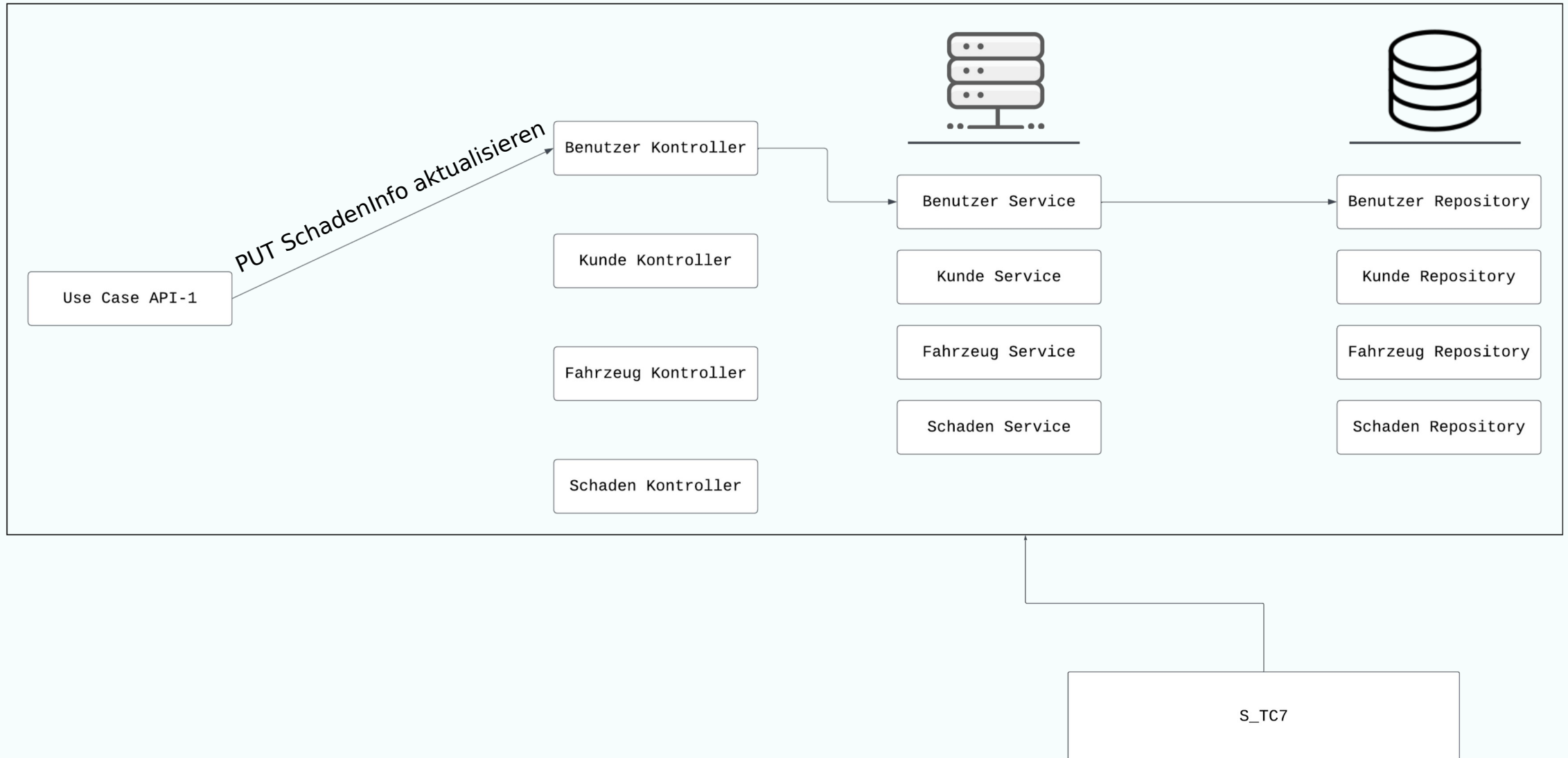
Rückverfolgbarkeit von Anforderungen



Rückverfolgbarkeit von Anforderungen



Rückverfolgbarkeit von Anforderungen



Rückverfolgbarkeit von Anforderungen

