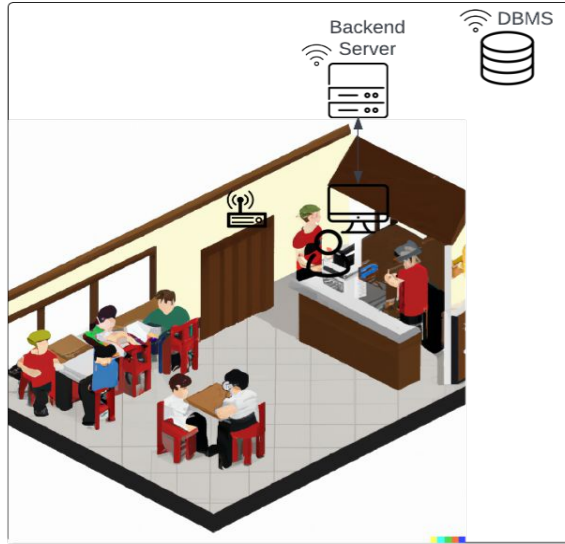

Order Tracking System (SiparisTakip)

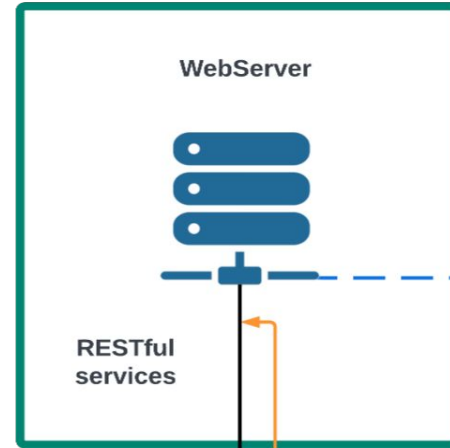
Softwaretest

Kerem ŞAKİR 200503038
Yasin AYDIN 220503068

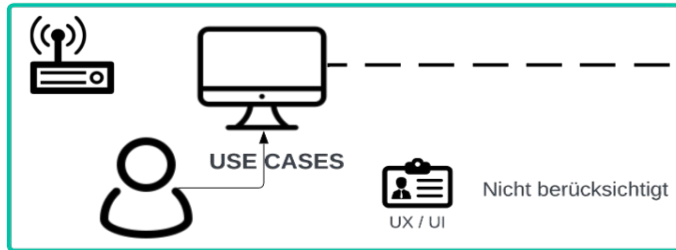
Systemüberblick



Physikalische Systemarchitektur

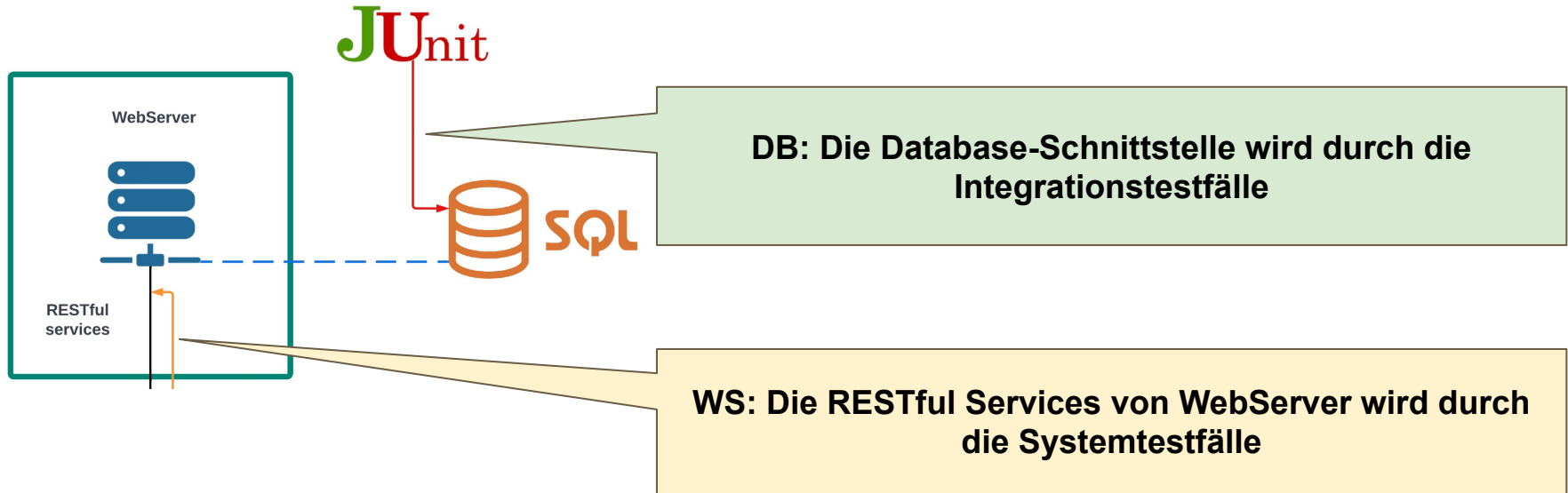


JUnit



Systemtestfälle

Das System wird über zwei Systemschnittstellen getestet:



Postman Webservice Testen

Die Testmethoden(GET, POST) sind in <https://www.postman.com/supply-technologist-1170177/workspace/siparistakipwstests> verfügbar.

System Test Case ID Name	S_TC_1 Test, dass sich der registrierte Benutzer mit GET-girisYap am System anmelden kann.
Preconditions	<ul style="list-style-type: none">• Der Webservice ist gegeben:<ul style="list-style-type: none">◦ Request: GET girisYap(String sifreHashed),◦ Response: Informationen des Personals, dessen Passwort als Parameter angegeben ist<ul style="list-style-type: none">■ Data "Personel"<ul style="list-style-type: none">• Long id // unique• String personelAdi• long izinSeviyeID• String personelSifreHashed■ Where personelSifreHashed = SHA256(personelSifre)■ Where izinSeviyeID<ul style="list-style-type: none">• =1 => Garson(Kellner)• =2 => Kasiyer(Kassierer)• Der Server ist konfiguriert:<ul style="list-style-type: none">◦ Personel<ul style="list-style-type: none">■ id = 2■ personelAdi = "yasin"■ izinSeviyeID = 1■ personelSifreHashed = a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3◦ Personel<ul style="list-style-type: none">■ id = 1■ personelAdi = "kerem"■ izinSeviyeID = 2■ personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72• App port number = (default)8080• Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	1. GET girisYap(String sifreHashed) wird mit dem per SHA256 kodierten Personalpasswort aufgerufen.
Post-Condition	null
Test Data	GET http://localhost:8080/girisYap?sifreHashed=a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3
Expected Result	{ "id": 2, "personelAdi": "yasin", "izinSeviyeID": 1, "personelSifreHashed": "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3" }
Actual Result	{ "id": 2, "personelAdi": "yasin", "izinSeviyeID": 1, "personelSifreHashed": "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3" }
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req1

System Test Case ID Name	S_TC_2 Test der Möglichkeit, die Bestellungen für den ausgewählten Tisch zu bearbeiten.
Preconditions	<ul style="list-style-type: none"> Der Webservice ist gegeben: <ul style="list-style-type: none"> Request: GET masaSiparisi(String sifreHashed, String masaAdi), Response: Aktuelle Bestellinformationen für den angegebenen Tischnamen <ul style="list-style-type: none"> Data “Siparis” <ul style="list-style-type: none"> Long id // unique long personelID String masaAdi long siparisZamani String urunler double toplamTutar Where urunler = {(Urun.id)urunID: (int)urunAdet} Where siparisZamani = (long)UNIX Time Request: POST siparisGuncelle(String sifreHashed, long siparisID, long urunID, int miktar), Response: Informationen zum Hinzufügen des Produkts mit der angegebenen Produkt-ID zur Bestellung mit der angegebenen Bestell-ID in der angegebenen Menge <ul style="list-style-type: none"> String Antwortnachricht Der Server ist konfiguriert: <ul style="list-style-type: none"> Masa <ul style="list-style-type: none"> id=32 masaAdi="A-04" siparisID=161 Urun <ul style="list-style-type: none"> id=34 urunAdi="Patates Kızartması" urunFiyati=32.00 urunTuruID=2 UrunTuru <ul style="list-style-type: none"> id=2 urunTuruAdi="Ara Sıcak" Siparis <ul style="list-style-type: none"> id = 161 personelID = 1 masaAdi = "A-04" siparisZamani = 1684941603616 urunler= {"34":4} toplamTutar= 128.0 Personel <ul style="list-style-type: none"> id = 1 personelAdi = "kerem" izinSeviyeID = 2 personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72 App port number = (default)8080 Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	<ol style="list-style-type: none">1. GET masaSiparisi(String sifreHashed, String masaAdi) wird mit dem Tischnamen aufgerufen.2. POST siparisGuncelle(String sifreHashed, long siparisID, long urunID, int miktar) wird mit der siparisID in der vorherigen Antwort, der ID des Produkts, das der Bestellung hinzugefügt werden soll, der Menge, und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	<p>GET http://localhost:8080/masaSiparisi?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&masaAdi=A-04</p> <p>POST http://localhost:8080/siparisGuncelle?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&siparisID=161&urunID=34&miktar=2</p>
Expected Result	<pre>{ "id": 161, "personelID": 1, "masaAdi": "A-04", "siparisZamani": 1684941603616, "urunler": "{\"34\":4}", "toplamTutar": 128.0 }</pre> <p>"İşlem tamamlandı! "</p>
Actual Result	<pre>{ "id": 161, "personelID": 1, "masaAdi": "A-04", "siparisZamani": 1684941603616, "urunler": "{\"34\":4}", "toplamTutar": 128.0 }</pre> <p>"İşlem tamamlandı!</p>
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req2, WS_req3

System Test Case ID Name	S_TC_3 Ruftest der Zahlungshistorien
Preconditions	<ul style="list-style-type: none">Der Webservice ist gegeben:<ul style="list-style-type: none">Request: GET odemeGecmisi(String sifreHashed),Response: Liste vergangener Zahlungen<ul style="list-style-type: none">Data "JSONArray":<ul style="list-style-type: none">[[Long id, long siparisID, long odemeYontemiID, long personelID, double bahsis, double toplam, String odemeNotu, long odemeZamani, String masaAdi, String odemeYontemiAdi, String personelAdi]]Where odemeZamani = (long)UNIX TimeDer Server ist konfiguriert:<ul style="list-style-type: none">OdemeYontemi<ul style="list-style-type: none">id=2odemeYontemiAdi="Kredi Kartı"Urun<ul style="list-style-type: none">id=34urunAdi="Patates Kızartması"urunFiyati=32.00urunTuruID=2UrunTuru<ul style="list-style-type: none">id=2urunTuruAdi="Ara Sıcak"Siparis<ul style="list-style-type: none">id = 161personelID = 1masaAdi = "A-04"siparisZamani = 1684941603616urunler= {"34":6}toplamTutar= 192.0Odeme<ul style="list-style-type: none">id=101siparisID=161odemeYontemiID=2personelID=1bahsis=8.00toplam=200.00odemeNotu="ZiraatBankası"odemeZamani=1684944823200Personel<ul style="list-style-type: none">id = 1personelAdi = "kerem"izinSeviyeID = 2personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72App port number = (default)8080Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	1. GET odemeGecmisi(String sifreHashed) wird mit Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	GET http://localhost:8080/odemeGecmisi?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72
Expected Result	[{"odemeYontemiID":2,"siparisID":161,"toplam":200,"odemeNotu":"ZiraatBankasi","odemeYontemiAdi":"Kredi Karti", "odemeZamani":1684944823200,"bahsis":8,"personelAdi":"kerem","masaAdi":"A-04","id":101,"personelID":1}]
Actual Result	[{"odemeYontemiID":2,"siparisID":161,"toplam":200,"odemeNotu":"ZiraatBankasi","odemeYontemiAdi":"Kredi Karti", "odemeZamani":1684944823200,"bahsis":8,"personelAdi":"kerem","masaAdi":"A-04","id":101,"personelID":1}]
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req4

System Test Case ID Name	S_TC_4 Test der Auflistung von Produkten, die für einen Produkttyp registriert sind.
Preconditions	<ul style="list-style-type: none">• Der Webservice ist gegeben:<ul style="list-style-type: none">◦ Request: GET urunTurleri(String sifreHashed),◦ Response: Liste der in der Datenbank registrierten Produkttypen<ul style="list-style-type: none">■ Data "List<UrunTuru>":<ul style="list-style-type: none">• Long id //unique• String urunTuruAdi◦ Request: GET urunler(String sifreHashed, long urunTuruID),◦ Response: Liste aller Produkte, bei denen die urunTuruID als Parameter angegeben ist<ul style="list-style-type: none">■ Data "List<Urun>":<ul style="list-style-type: none">• Long id //unique• String urunAdi• double urunFiyati• long urunTuruID• Der Server ist konfiguriert:<ul style="list-style-type: none">◦ Urun<ul style="list-style-type: none">■ id=34■ urunAdi="Patates Kizartmasi"■ urunFiyati=32.00■ urunTuruID=2◦ UrunTuru<ul style="list-style-type: none">■ id=2■ urunTuruAdi="Ara Sıcak"◦ Personel<ul style="list-style-type: none">■ id = 1■ personelAdi = "kerem"■ izinSeviyeID = 2■ personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72• App port number = (default)8080• Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	<ol style="list-style-type: none"> GET urunTurleri(String sifreHashed) wird mit Mitarbeiterpasswort(SHA256 Hashed) aufgerufen. GET urunler(String sifreHashed, long urunTuruID) wird mit der urunTuruID in der vorherigen Antwort, und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	<p>GET http://localhost:8080/urunTurleri?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72</p> <p>GET http://localhost:8080/urunler?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&urunTuruID=2</p>
Expected Result	<pre>[{ "id": 2, "urunTuruAdi": "Ara Sıcak" }] [{ "id": 34, "urunAdi": " Patates kızartması", "urunFiyati": 32.0, "urunTuruID": 2 }]</pre>
Actual Result	<pre>[{ "id": 2, "urunTuruAdi": "Ara Sıcak" }] [{ "id": 34, "urunAdi": " Patates kızartması", "urunFiyati": 32.0, "urunTuruID": 2 }]</pre>
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req5, WS_req6

System Test Case ID Name	S_TC_5 Test des Hinzufügens eines neuen Elements zum Menü
Preconditions	<ul style="list-style-type: none">• Der Webservice ist gegeben:<ul style="list-style-type: none">◦ Request: GET urunTurleri(String sifreHashed),◦ Response: Liste der in der Datenbank registrierten Produkttypen<ul style="list-style-type: none">■ Data "List<UrunTuru>":<ul style="list-style-type: none">• Long id //unique• String urunTuruAdi◦ Request: POST urunEkle(String sifreHashed, String urunAdi, double urunFiyati, long urunTurulD),◦ Response: Informationen zum Hinzufügen des Produkts zum Menü<ul style="list-style-type: none">■ String Antwortnachricht• Der Server ist konfiguriert:<ul style="list-style-type: none">◦ Urun<ul style="list-style-type: none">■ id=34■ urunAdi="Patates Kızartması"■ urunFiyati=32.00■ urunTurulD=2◦ UrunTuru<ul style="list-style-type: none">■ id=1■ urunTuruAdi="Ana Yemek"◦ UrunTuru<ul style="list-style-type: none">■ id=2■ urunTuruAdi="Ara Sıcak"◦ Personel<ul style="list-style-type: none">■ id = 1■ personelAdi = "kerem"■ izinSeviyeID = 2■ personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72• App port number = (default)8080• Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	<ol style="list-style-type: none"> GET urunTurleri(String sifreHashed) wird mit Mitarbeiterpasswort(SHA256 Hashed) aufgerufen. POST urunEkle(String sifreHashed, long urunAdi, double urunFiyati, long urunTurulD) wird mit der urunTurulD in der vorherigen Antwort, Produktname, Produktpreis und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	<p>GET http://localhost:8080/urunTurleri?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72</p> <p>POST http://localhost:8080/urunEkle?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&urunAdi=lzgara%20Kalamar&urunFiyati=75.00&urunTurulD=1</p>
Expected Result	<pre>[{ "id": 1, "urunTuruAdi": "Ana Yemek" }, { "id": 2, "urunTuruAdi": "Ara Sıcak" }]</pre> <p>“İşlem Tamamlandı! ”</p>
Actual Result	<pre>[{ "id": 1, "urunTuruAdi": "Ana Yemek" }, { "id": 2, "urunTuruAdi": "Ara Sıcak" }]</pre> <p>“İşlem Tamamlandı! ”</p>
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req5, WS_req7

System Test Case ID Name	S_TC_6 Test des Löschen eines vorhandenen Menüpunkts aus dem Menü.
Preconditions	<ul style="list-style-type: none">Der Webservice ist gegeben:<ul style="list-style-type: none">Request: GET urunTurleri(String sifreHashed),Response: Liste der in der Datenbank registrierten Produkttypen<ul style="list-style-type: none">Data "List<UrunTuru>":<ul style="list-style-type: none">Long id //uniqueString urunTuruAdiRequest: GET urunler(String sifreHashed, long urunTurulID),Response: Liste aller Produkte, bei denen die urunTurulID als Parameter angegeben ist<ul style="list-style-type: none">Data "List<Urun>":<ul style="list-style-type: none">Long id //uniqueString urunAdidouble urunFiyatiRequest: POST urunCikar(String sifreHashed, String urunAdi, double urunFiyati, long urunTurulID),Response: Informationen zum Löschen des Produkts zum Menü<ul style="list-style-type: none">String AntwortnachrichtDer Server ist konfiguriert:<ul style="list-style-type: none">Urun<ul style="list-style-type: none">id=34urunAdi="Patates Kızartması"urunFiyati=32.00urunTurulID=2Urun<ul style="list-style-type: none">id=65urunAdi="Izgara Kalamar"urunFiyati=75.00urunTurulID=1UrunTuru<ul style="list-style-type: none">id=1urunTuruAdi="Ana Yemek"UrunTuru<ul style="list-style-type: none">id=2urunTuruAdi="Ara Sıcak"Personel<ul style="list-style-type: none">id = 1personelAdi = "kerem"izinSeviyeID = 2personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72App port number = (default)8080Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	<ol style="list-style-type: none">1. GET urunTurleri(String sifreHashed) wird mit Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.2. GET urunler(String sifreHashed, long urunTuruID) wird mit der urunTuruID in der vorherigen Antwort, und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.3. POST urunCikar(String sifreHashed, long urunID) wird mit der urunID in der vorherigen Antwort und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	GET http://localhost:8080/urunTurleri?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72 GET http://localhost:8080/urunler?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&urunTuruID=1 POST http://localhost:8080/urunCikar?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&urunID=34
Expected Result	<pre>[{"id": 1, "urunTuruAdi": "Ana Yemek"}, {"id": 2, "urunTuruAdi": "Ara Sıcak"}]</pre> <pre>[{"id": 65, "urunAdi": "Izgara Kalamar", "urunFiyati": 75.0, "urunTuruID": 1}]</pre> "İşlem Tamamlandı! "
Actual Result	<pre>[{"id": 1, "urunTuruAdi": "Ana Yemek"}, {"id": 2, "urunTuruAdi": "Ara Sıcak"}]</pre> <pre>[{"id": 65, "urunAdi": "Izgara Kalamar", "urunFiyati": 75.0, "urunTuruID": 1}]</pre> "İşlem Tamamlandı! "
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req5, WS_req6, WS_req8

System Test Case ID Name	S_TC_7 Test, die Zahlung der Tischbestellungen mit dem angegebenen Trinkgeldbetrag, der Zahlungsart und dem Zahlungsvermerk zu erhalten und in der Datenbank zu erfassen.
Preconditions	<ul style="list-style-type: none"> Der Webservice ist gegeben: <ul style="list-style-type: none"> Request: GET masaSiparisi(String sifreHashed, String masaAdi), Response: Aktuelle Bestellinformationen für den angegebenen Tischnamen <ul style="list-style-type: none"> Data "Siparis" <ul style="list-style-type: none"> Long id // unique, long personelID, String masaAdi, long siparisZamani, String urunler, double toplamTutar Where urunler = {(Urun.id)urunID: (int)urunAdet} Where siparisZamani = (long)UNIX Time Request: GET odemeYontemleri(String sifreHashed), Response: Liste aller in der Datenbank registrierten Zahlungsarten <ul style="list-style-type: none"> Data "List<OdemeYontemi>": <ul style="list-style-type: none"> Long id //unique String odemeYontemiAdi Request: POST odeme(String sifreHashed, long siparisID, long odemeYontemiID, double bahsis, String odemeNotu), Response: Informationen zur Erfassung der Zahlung der Bestellung <ul style="list-style-type: none"> String Antwortnachricht Der Server ist konfiguriert: <ul style="list-style-type: none"> Masa <ul style="list-style-type: none"> id=30 masaAdi="A-05" siparisID=163 OdemeYontemi <ul style="list-style-type: none"> id=1 odemeYontemiAdi="Nakit" Urun <ul style="list-style-type: none"> id=65 urunAdi="Izgara Kalamar" urunFiyati=75.00 urunTuruID=1 UrunTuru <ul style="list-style-type: none"> id=1 urunTuruAdi="Ana Yemek" Siparis <ul style="list-style-type: none"> id = 163 personelID = 1 masaAdi = "A-05" siparisZamani = 1684948698338 urunler= {{{"65":1}} toplamTutar= 75.00 Personel <ul style="list-style-type: none"> id = 1 personelAdi = "kerem" izinSeviyeID = 2 personelSifreHashed = 8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72 App port number = (default)8080 Das System ist betriebsbereit!
Test Steps	

Preconditions	
Test Steps	<ol style="list-style-type: none"> 1. GET masaSiparisi(String sifreHashed, String masaAdi) wird mit der masaAdi und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen. 2. GET odemeYontemleri(String sifreHashed) wird mit Mitarbeiterpasswort(SHA256 Hashed) aufgerufen. 3. POST odeme(String sifreHashed, long siparisID, long odemeYontemilID, double bahsis, String odemeNotu) wird mit der urunID in der ersten Antwort, odemeYontemilID in zweiten Antwort, der Zahlungsart, dem Zahlungsvermerk, und Mitarbeiterpasswort(SHA256 Hashed) aufgerufen.
Post-Condition	null
Test Data	<p>GET http://localhost:8080/masaSiparisi?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&masaAdi=A-05</p> <p>GET http://localhost:8080/odemeYontemleri?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72</p> <p>POST http://localhost:8080/odeme?sifreHashed=8d23cf6c86e834a7aa6eded54c26ce2bb2e74903538c61bdd5d2197997ab2f72&siparisID=163&odemeYontemilID=1&bahsis=5&odemeNotu=İşBankas I</p>
Expected Result	<pre>{ "id": 163, "personelID": 1, "masaAdi": "A-05", "siparisZamani": 1684948698338, "urunler": [{ "id": 1, "odemeYontemiAdi": "Nakit" }] }</pre> <p>“İşlem tamamlandı! “</p>
Actual Result	<pre>{ "id": 163, "personelID": 1, "masaAdi": "A-05", "siparisZamani": 1684948698338, "urunler": [{ "id": 1, "odemeYontemiAdi": "Nakit" }] }</pre> <p>“İşlem tamamlandı! “</p>
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	WS_req2, WS_req9, WS_req10

JUnit Datenbank Testen

Die Testfunktionen sind in Klasse `javamudavimleri.siparistakip.spring.veritabani.test.Testler` verfügbar.

Component Integration Test Case ID Name	CI_TC_1 Testen des Hinzufügens einer neuen Berechtigungsstufe zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “IzinSeviyeDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>izinSeviyeAdi = “”izinSeviyesi = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft izinSeviyeTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String izinSeviyeAdi = “IzinSeviyeTest”; int izinSeviyesi = 1;
Expected Result	DB IzinSeviye.izinSeviyesi = 1
Actual Result	DB IzinSeviye.izinSeviyesi = 1
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req1

Component Integration Test Case ID Name	CI_TC_2 Test zum Hinzufügen eines neuen Mitarbeiters zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “PersonelDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>personelAdi = “”izinSeviyeID = “”personelSifreHashed = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft personelTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String personelAdi = “PersonelTest”; long izinSeviyeID = 1L; String personelSifreHashed = SHA256(“SifreTest”);
Expected Result	DB Personel.personelSifreHashed = SHA256(“SifreTest”)
Actual Result	DB Personel.personelSifreHashed = SHA256(“SifreTest”)
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req2

Component Integration Test Case ID Name	CI_TC_3 Test zum Hinzufügen eines neuen Produkttyps zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “UrunTuruDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>urunTuruAdi = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft urunTuruTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String urunTuruAdi = “UrunTuruTest”;
Expected Result	DB UrunTuru.urunTuruAdi = “UrunTuruTest”
Actual Result	DB UrunTuru.urunTuruAdi = “UrunTuruTest”
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req3

Component Integration Test Case ID Name	CI_TC_4 Test zum Hinzufügen eines neuen Produkts zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “UrunDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>urunAdi = “”urunFiyati = “”urunTurulID = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft urunTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String urunAdi = “UrunTest”; double urunFiyati = 1.25; long urunTurulID = 2L;
Expected Result	DB Urun.urunFiyati = 1.25
Actual Result	DB Urun.urunTurulID = 1.25
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req4

Component Integration Test Case ID Name	CI_TC_5 Test zum Hinzufügen eines neuen Bestellung zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle "SiparisDB" enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>personellID = ""siaprisZamani = ""masaAdi = ""
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft siparisTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String masaAdi = "MasaTest"; long personellID = 1L; long siparisZamani = System.currentTimeMillis();
Expected Result	DB Siparis.siparisZamani = System.currentTimeMillis()
Actual Result	DB Siparis.siparisZamani = System.currentTimeMillis()
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req5

Component Integration Test Case ID Name	CI_TC_6 Test zum Hinzufügen eines neuen Tisch zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “MasaDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>masaAdi = “”siparisID = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft masaTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String masaAdi = "MasaTest"; long siparisID = 163L;
Expected Result	DB Masa.siparisID = 163L
Actual Result	DB Masa.siparisID = 163L
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req6

Component Integration Test Case ID Name	CI_TC_7 Test des Hinzufügens einer neuen Zahlungsmethode zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “OdemeYontemiDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>odemeYontemiAdi = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft odemeYontemiTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	String odemeYontemiAdi = "OdemeYontemiTest"
Expected Result	DB OdemeYontemi.odemeYontemiAdi = “OdemeYontemiTest”
Actual Result	DB OdemeYontemi.odemeYontemiAdi = “OdemeYontemiTest”
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req7

Component Integration Test Case ID Name	CI_TC_8 Test des Hinzufügens einer neuen Zahlung zur Datenbank.
Preconditions	<ul style="list-style-type: none">Die Datenbanktabelle “OdemeDB” enthält diese Zeilen:<ul style="list-style-type: none">id = <generated>siparisID = “”odemeYontemiID = “”personelID = “”bahsis = “”toplamlam = “”odemeNotu = “”odemeZamani = “”
Test Steps	<ol style="list-style-type: none">JUnit-Testfall ruft odemeTest() auf,Die Antwortdaten werden analysiert und die erwarteten Daten werden ausgewählt.
Post-Conditions	null
Test Data	long siparisID = 163L; long odemeYontemiID = 1L; long personelID = 1L; double bahsis = 0.75; double araToplam = 75.00; String odemeNotu = "OdemeNotuTest"; long odemeZamani = System.currentTimeMillis();
Expected Result	DB Odeme.odemeNotu = “OdemeNotuTest”
Actual Result	DB Masa.siparisID = “OdemeNotuTest”
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	DB_req8

Rückverfolgbarkeit

Anforderungen Use Cases Stories

- **UI-1/** Auf dem Anmeldebildschirm kann sich jeder Kellner mit seinem eigenen Passwort anmelden.
- **UI-2/** In der Tische-Oberfläche werden alle Tische im System angezeigt, der zu bearbeitende ausgewählt und die Bestellungen können bearbeitet werden.
- **UI-3/** Die Voucher-Schnittstelle hat eine Historie von zuvor erbrachten Dienstleistungen; Tischname, Ticketnummer, Zahlungsbetrag, Zahltag usw. kann betrachtet werden.
- **UI-4/** Das vorhandene Menü wird von der Menüoberfläche angezeigt, es bietet die Möglichkeit zu bearbeiten und hinzuzufügen und zu entfernen.
- **UI-5/** Die Zwischensumme des Benutzers wird auf der Zahlungsschnittstelle angezeigt und er/sie kann ein Trinkgeld hinterlassen, wenn er/sie möchte. Nach Auswahl der Zahlungsmethode zahlt der Nutzer.



- Unser Kunde möchte die Auftragsverfolgung der Tische und Kellner am Arbeitsplatz überwachen.
- Dafür wünscht er sich eine benutzerfreundliche, einfache Oberfläche und eine effektive Anwendung. Die Interaktion findet statt, wenn der Kunde die in der Benutzeroberfläche verfügbaren Tabellen auswählt.
- Die Produkte und Dienstleistungen, die aus der prominenten Liste angefordert werden, nachdem der Benutzer den Tisch ausgewählt hat, den er handeln möchte.

