SEP Projektmappe

**Projektmappe des Projektes**

Supreme Eating Program

**Dokumentation des Projektes**

Gruppe 12:

Nicklas Czorny

Leon Gashi

Oguzhan Kaluk

Alexandra Klein

Dursun Berat Saral

Nick Waldschmidt

**Hinweis**

An vielen Stellen findet Ihr im Dokument folgendes Kästchen:

*Dies ist eine Hilfestellung.*

Diese Kästen dienen dazu, Euch kurze Informationen über Ziele und Inhalte der jeweiligen Abschnitte zu geben. Auch die Beispiele und Templates dienen dazu, euch bei der Dokumentation eures Projektes zu unterstützen. **Sowohl die Kästchen als auch die Beispiele und Templates sind spätestens zur finalen Abgabe der Projektmappe vollständig zu entfernen.** Betrachtet dieses Dokument bitte nicht als Aufgabe, die man von oben nach unten abarbeiten soll; es soll vielmehr als durchgängige Dokumentation eurer Projektarbeit dienen und fortlaufend erweitert bzw. angepasst werden, sodass am Ende des SEPs der Entwicklungsprozess Eurer Software vollständig dokumentiert ist.

Das SEP-Team wünscht Euch  
**Viel Erfolg**

Inhalt

[Projektbeschreibung 5](#_Toc93312813)

[Supreme Eating Program 5](#_Toc93312814)

[Einleitung 5](#_Toc93312815)

[Zyklus 1 5](#_Toc93312816)

[Zyklus 2 6](#_Toc93312817)

[Zyklus 3 8](#_Toc93312818)

[Anmerkungen 9](#_Toc93312819)

[Zyklus I 10](#_Toc93312820)

[Spezifikationsplanung 10](#_Toc93312821)

[User-Stories 11](#_Toc93312822)

[User Stories Zyklus 1 12](#_Toc93312823)

[Papierprototypen 17](#_Toc93312824)

[Szenarien (MSCs) 21](#_Toc93312825)

[Strukturdiagramm (Klassendiagramm) 28](#_Toc93312826)

[Funktionalitätsplanung 29](#_Toc93312827)

[Systemtests 32](#_Toc93312828)

[Zyklus II 39](#_Toc93312829)

[Spezifikationsplanung 39](#_Toc93312830)

[User-Stories 41](#_Toc93312831)

[Papierprototypen 49](#_Toc93312832)

[Kundenkonto 49](#_Toc93312833)

[Restaurant 52](#_Toc93312834)

[Essensbestellung 52](#_Toc93312835)

[Bewertung 54](#_Toc93312836)

[Szenarien (MSCs) 56](#_Toc93312837)

[Strukturdiagramm (Klassendiagramm) 64](#_Toc93312838)

[Funktionalitätsplanung 66](#_Toc93312839)

[Unittests 69](#_Toc93312840)

[Systemtests 70](#_Toc93312841)

[Zyklus III 73](#_Toc93312842)

[Spezifikationsplanung 73](#_Toc93312843)

[User-Stories 75](#_Toc93312844)

[Papierprototypen 80](#_Toc93312845)

[Aktionen 80](#_Toc93312846)

[Sonderwünsche 80](#_Toc93312847)

[Statistik 82](#_Toc93312848)

[Lieferzeit 82](#_Toc93312849)

[Gericht Suchen 83](#_Toc93312850)

[Szenarien (MSCs) 84](#_Toc93312851)

[Strukturdiagramm (Klassendiagramm) 90](#_Toc93312852)

[Funktionalitätsplanung 92](#_Toc93312853)

[Unittests 95](#_Toc93312854)

[Systemtests 96](#_Toc93312855)

[Nutzerhandbuch 97](#_Toc93312856)

[Technische Anforderungen 97](#_Toc93312857)

[Installationsanleitung 97](#_Toc93312858)

[Bedienungsanleitung 97](#_Toc93312859)

# Projektbeschreibung

## Supreme Eating Program

### Einleitung

Die Supreme Eating Program soll ein Programm werden, das Essensbestellungen für

Restaurantbesitzer und deren Kunden abwickelt. Restaurantbesitzer können ihr Restaurant

registrieren, ihre Speisekarten veröffentlichen und die Bewertungen einsehen, die deren Kunden

abgegeben haben. Kunden können Restaurants in ihrer Nähe suchen, ihr Lieblingsessen bestellen

und anschließend bewerten. Das Programm soll auf einer Client-Server-Architektur aufbauen; der

Server soll von mehreren Clients gleichzeitig erreichbar sein.

### Zyklus 1

Im ersten Zyklus soll zunächst die Grundstruktur des Programms, bestehend aus einem Client und

einem Server, implementiert werden. Sämtliche Daten des Programms (Daten der Restaurantbesitzer

und deren Restaurants inklusive der Speisekarten sowie im weiteren Verlauf die der Kunden) sollen

persistent gespeichert werden.

(Registrierung der Restaurantbesitzer)

Es soll Nutzern möglich sein, sich als Restaurantbesitzer zu registrieren und somit ihr Profil zu

erstellen. Diese Profile sollen persistent gespeichert werden und aus dem Vor- und Nachnamen,

einer E-Mailadresse und einem Passwort bestehen.

Der Login-Vorgang soll durch eine Zwei-Faktor-Authentisierung (auch Zwei-Faktor-

Authentifizierung genannt) realisiert werden, bei der, neben dem Passwort, ein Sicherheitscode

eingegeben werden muss, der per E-Mail an den Restaurantbesitzer versendet wird. Die Zwei-Faktor-

Authentisierung sollte über den Server realisiert werden.

(Restaurant anlegen)

Nach der Erstellung seines Profils sollte der Restaurantbesitzer sein Restaurant erstellen. Ein

Restaurant besteht aus einem Namen, einer Adresse und einer Kategorie, in die dessen Speisen

fallen. Mögliche Kategorien sind beispielsweise Pizza, Burger, Schnitzel, Sushi und Döner. Darüber

hinaus sollte er einen Lieferbereich für sein Restaurant definieren. Der Lieferbereich gibt die

maximale Entfernung in Kilometern ausgehend von der Adresse des Restaurants an, die durch das

Restaurant beliefert wird. Das Restaurant soll auf einer Karte markiert und der entsprechende

Lieferbereich angezeigt werden. Außerdem gehören die Angabe eines Mindestbestellwertes sowie

der Lieferkosten zu der Erstellung des Restaurants.

Alle diese Restaurant-Daten sollen im Nachhinein geändert werden können.

(Speisekarte anlegen)

Der nächste Schritt ist die Erstellung der Speisekarte. Eine Speisekarte ist ebenfalls in Kategorien

unterteilt (z.B. Schnitzel, Nudelgerichte, Salate, etc). Für die Erstellung der Speisekarte gibt es zwei

Möglichkeiten. Zum einen kann die Speisekarte aus einer vordefinierten xml-Datei1 ausgelesen und

integriert werden, zum anderen soll diese manuell (durch Eingabe der einzelnen Gerichte) erstellt

werden können. Jedes Gericht besteht aus einem Namen, einer Beschreibung und einem Preis. Dazu

kann optional ein Bild des Gerichts eingefügt werden. Die Speisekarte sollte auch im Nachhinein

bearbeitet werden können (Gerichte hinzufügen oder löschen, Namen anpassen, Preis ändern).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 Eine xml-Beispieldatei wird in Moodle bereitgestellt

### Zyklus 2

(Kundenkonto anlegen)

Im zweiten Zyklus soll das Programm durch die Kundenkonten erweitert werden. Ein Kundenkonto

besteht aus einem Vor- und Nachnamen, dem Geburtsdatum, einer Mailadresse, einem Passwort

und einer Adresse (Standard-Adresse). Außerdem enthält das Kundenkonto ein Guthaben für

Essensbestellungen, das bei der Erstellung des Kontos 0 Euro beträgt.

Zusätzlich gibt es eine Bestellübersicht über vergangene Bestellungen im Kundenkonto, die zu Beginn

ebenfalls leer ist.

Außerdem sollte der Kunde eine Möglichkeit haben, eine alternative Adresse einzugeben, wenn er

die Bestellung nicht nach Hause geliefert haben möchte.

(Auswahl von Restaurants anzeigen)

Damit der Kunde eine Bestellung abgeben kann, muss er zunächst ein passendes Restaurant suchen.

Restaurants werden ihm in einer Liste angezeigt, die den Namen des Restaurants, die Kategorie

sowie die durchschnittliche Bewertung (siehe unten) anzeigt. Alternativ kann der Kunde sich die

Restaurants in seiner Nähe (zu seiner Standard-Adresse oder zu seiner alternativen Adresse) auf der

Karte anzeigen lassen. Sowohl in der Liste als auch auf der Karte erscheinen nur diejenigen

Restaurants, in deren Lieferbereich sich der Kunde befindet. Dazu muss der Kunde auswählen, ob er

zu seiner Standard-Adresse oder zu der alternativen Adresse liefern lassen möchte. Da der Kunde nur

bei den Restaurants in seiner Umgebung (in deren Lieferbereich er sich befindet) bestellen kann,

werden die übrigen ausgeblendet.

Bei der Suche nach einem bestimmten Restaurant kann der Kunde die Angebote filtern. Er kann

entweder einen bestimmten Restaurantnamen eingeben, um ein Restaurant zu finden oder

alternativ kann er nach Kategorien oder der Entfernung (zu seiner Standard-Adresse oder zu seiner

alternativen Adresse) filtern.

Jedes Suchergebnis kann als Favorit gespeichert werden. Hat der Kunde ein oder mehrere

Restaurants als Favoriten markiert, so kann er auch nach Favoriten filtern. Favoriten werden immer

angezeigt, auch wenn der Kunde sich nicht in dem Lieferbereich befindet.

Der Kunde soll die Möglichkeit haben, sich die Speisekarte des jeweiligen Restaurants anzusehen.

(Essensbestellungen)

Bevor der Kunde sein Essen bestellen kann, muss er sein Guthaben aufladen. Dazu gibt er einen

beliebigen Betrag ein, der anschließend seinem Konto gutgeschrieben wird. Anschließend kann er die

Gerichte, die er auf der Speisekarte eines Restaurants gefunden hat, zu seiner Bestellung hinzufügen.

Um die Bestellung abschließen zu können, muss der Mindestbestellwert erreicht werden und das

Guthaben ausreichend sein. Nachdem der Kunde seine Bestellung abgeschlossen hat, erhält er eine

Bestätigung per E-Mail und seine Bestellung wird in dem Kundenkonto in die Bestellhistorie

eingetragen. Zu jedem dieser Einträge gehört ein Datum, der Name des Restaurants, die bestellten

Gerichte sowie der Gesamtpreis der Bestellung.

(Bewertung)

Nachdem der Kunde ein Essen bestellt hat, kann er seine Bestellung bewerten. Zu diesem Zweck

kann er Sterne (1-5) in den Kategorien „Essen“ und „Lieferung“ vergeben. Außerdem kann er einen

Kommentar zu seiner Bewertung hinzufügen. Diese Bewertungen sind anschließend für die anderen

Kunden sichtbar. In der Liste der Restaurants wird die durchschnittliche Bewertung der Sterne für die

beiden Kategorien angezeigt. Zusätzlich können sich Restaurantbesitzer und Kunden durch die

Einzelbewertungen klicken.

(Treuepunkte)

Für jede Bestellung werden dem Kundenkonto Treuepunkte gutgeschrieben (5 Euro Bestellwert = 1

Treuepunkt). Für das Erreichen von 10 Treuepunkten wird dem Kunden automatisch ein Coupon

zugeschickt. Ein Coupon besteht aus einem 12-stelligen Code, der per Mail zugeschickt wird. Dieser

Code kann schließlich für Rabatte eingesetzt werden. Die Form der Rabatte kann frei gewählt werden

### Zyklus 3

(Aktionen)

Um Werbung für ihr Restaurant zu machen, können Restaurantbesitzer zeitlich begrenzte Rabatte

definieren. Zum einen sollen sie einen Tag in der Woche angeben können, an dem die Kunden

automatisch 20 Prozent Rabatt auf ihre Bestellungen erhalten. Alternativ können sie auch einen

bestimmten Zeitraum (in Tagen) angeben, in dem der Rabatt von 20 Prozent angeboten wird.

Außerdem erhalten Kunden an ihrem Geburtstag automatisch 20 Prozent Rabatt. Rabattaktionen

sollen in der Übersicht aller Restaurants angezeigt werden. Kunden, die das Restaurant als Favorit

markiert haben, sollen eine Mail mit der Rabattaktion erhalten.

Um diese Funktion effektiv testen zu können, soll das Datum im Programm verändert werden

können.

(Statistik)

Um den Effekt ihrer Maßnahmen/Aktionen beobachten zu können, sollen Restaurantbesitzer die

Statistik über ihr Restaurant einsehen können. Dazu können sie einsehen, welches Gericht in einem

bestimmten Monat bzw. einer Woche (Zeitraum frei wählbar) besonders häufig bestellt wurde2.

Außerdem sollen sie die Entwicklung der Bewertungen, die sich auf ihr Restaurant beziehen einsehen

können. Diese Daten sollen sowohl textuell als auch grafisch aufbereitet werden.

(Sonderwünsche)

Natürlich kann es passieren, dass die Kunden Sonderwünsche äußern möchten, die nicht in der

Speisekarte enthalten sind. Zu diesem Zweck soll dem Kunden ein Textfeld vor dem Abschluss der

Bestellung zur Verfügung stehen. Der Restaurantbesitzer kann diese Sonderwünsche anschließend

annehmen oder ablehnen. Im Falle einer Annahme muss die Rechnung entsprechend angepasst

werden. Der Kunde soll eine Benachrichtigung über die Annahme/Ablehnung erhalten, bevor er die

Bestellung abschickt.

(Lieferzeit)

Nachdem der Kunde eine Bestellung abgeschickt hat, soll ihm die voraussichtliche Lieferzeit

angezeigt werden. Diese berechnet sich aus der Dauer der Zubereitung (10 Minuten), der aktuellen

Auslastung des Restaurants (bei mehr als 2 aktuellen Bestellungen kommen weitere 10 Minuten

hinzu) und der Fahrzeit (10 Minuten je 5 Kilometer).

(Suche nach bestimmten Gericht)

Wenn der Kunde sich bei der Auswahl des Restaurants unsicher ist, soll er die Möglichkeit haben, auf

allen im Programm hinterlegten Speisekarten (in deren Lieferbereich sich der Kunde befindet) nach

einem bestimmten Gericht zu suchen. Nach der Eingabe des Namens des Gerichts sollen ihm alle

Restaurants angezeigt werden, die dieses Gericht anbieten.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 Um diese Funktion testen zu können, sollten mindestens 10 Bestellungen für die Abnahme vorbereitet und in der Datenbank hinterlegt werden.

### Anmerkungen

* Das Programm ist in der Programmiersprache Java zu entwickeln.
* Die Datenbank ist als lokale Datenbank anzulegen (z.B. via XAMPP

https://www.apachefriends.org/de/index.html). Bei Datenbanken, die online bereitgestellt

werden, kann es passieren, dass diese bei der Abnahme nicht zur Verfügung stehen.

* Wir stellen im Laufe der Veranstaltung Technologien vor, mit denen sich diese

Aufgabenstellung realisieren lässt. Sollten Sie alternative Technologien verwenden wollen, so

können Sie dies gerne tun. Allerdings können wir in diesem Fall eine Hilfestellung bei

Problemen nicht garantieren.

* Wir veröffentlichen besonders gelungene Software auf unserer SEP-Webseite. Hierzu ist es

unbedingt erforderlich, dass das System keine urheberrechtlich geschützten Inhalte (Bilder,

Musik, etc.) enthält.

# Zyklus I

## Spezifikationsplanung

Jedes Artefakt, das im Rahmen des SEP erstellt wird, muss mit dem Namen genau einer Verantwortlichen/eines Verantwortlichen versehen werden. Das bedeutet, dass jede User Story, jeder Papierprototyp, jedes MSC, das Klassendiagramm und sämtliche Tests mit dem Namen der/des Verantwortlichen versehen und hier in der untenstehenden Tabelle entsprechend eingetragen werden muss. Natürlich kann die Gruppe gemeinsam an einem Artefakt arbeiten, als Verantwortlicher sollte aber genau eine Person eingetragen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | Registrierung der Nutzer |  |  |  |
| 1.1 | Abschnitt: Registrierung der Restaurantbesitzer  Abschnitt: Restaurant anlegen  Abschnitt: Speisekarte anlegen | User Story | Nick Waldschmidt  Dursun Berat Saral  Nick Waldschmidt und Dursun Berat Saral | In Bearbeitung |
| 1.2 | * Login-Fenster * Registrierungs-Fenster * Zwei-Faktor-Authentifizierungs-Fenster * Restaurant Daten erstellen * Speisekarte erstellen aus xml-Datei * Eingabefelder aus der xml-Datei * Speisekarte manuell erstellen | Papierprototyp | Alexandra Klein | In Bearbeitung |
| 1.2.1 | Nutzerkontoerstellung  Login und Zwei-Faktor-Authentifizierung  Restaurantdaten erstellen  Speisekarte erstellen aus vordefinierter xml-Datei  Speisekarte manuell erstellen | MSC | Leon Gashi  Leon Gashi  Oguzhan Kaluk  Oguzhan Kaluk  Oguzhan Kaluk | In Bearbeitung |
| **2.** | Klassendiagramm Client  Klassendiagramm Server | Klassendiagramm | Leon Gashi  Oguzhan Kaluk  Alexandra Klein  Dursun Berat Saral  Nick Waldschmidt  Alexandra Klein | In Bearbeitung |

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

**Schlechtes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** | Ich möchte ich rechtzeitig informiert werden, wenn ein Patient einen Termin nicht wahrnimmt. |
| **Geschätzter Realisierungsaufwand** | 1337 |
| **Priorität** | - |
| **Autor** | Emmett Brown, Rick Sanchez, Amelia Pond |
| **Abhängigkeiten zu anderen User Stories** |  |

**Gutes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** | 1.6 |
| **User Story-Beschreibung** | Als Arzt möchte ich mindesten fünf Minuten vor dem Termin informiert werden, wenn ein Patient einen Termin nicht wahrnimmt, sodass ich andere Patienten vorziehen kann. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Emmett Brown |
| **Abhängigkeiten zu anderen User Stories** | 1.3, 1.5 |

### User Stories Zyklus 1

|  |  |
| --- | --- |
| **User Story-ID** | 1.1 |
| **User Story-Beschreibung** | Als Restaurantbesitzer will ich mich registrieren können, um mein Profil zu erstellen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 2,3 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich das mein Profil persistent gespeichert werden soll, damit ich mich nicht erneut registrieren muss. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 1,3 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich eine Zwei-Faktor-Authentisierung per E-Mail, um einen sicheren Login zu garantieren. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | mittel |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 1,2 |

|  |  |
| --- | --- |
| **User Story-ID** | 2.1 |
| **User Story-Beschreibung** | Als Restaurant Besitzer will ich ein Restaurant erstellen können, damit meine Kunden auch über meine Webseite bestellen können. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | Hoch |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 2.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer sollen Kontaktdaten, wie Name, Adresse und Kategorien mit den Speisen, eingefügt werden. Dadurch sollen die Kunden den Standort Restaurants kennen und Speisen nach Kategorien sehen können. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 2.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll ein Lieferbereich angezeigt werden, damit wir nicht außerhalb dieses Bereichs liefern. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 2.4 |

|  |  |
| --- | --- |
| **User Story-ID** | 2.4 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll mein Restaurant auf der Karte markiert werden, damit Kunden mich sehen können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 2.3 |

|  |  |
| --- | --- |
| **User Story-ID** | 2.5 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll beim Bestellen ein Mindestbestellwert angezeigt werden, damit Bestellungen unter diesem Wert nicht betätigt werden können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 2.6 |
| **User Story-Beschreibung** | Als Restaurantbesitzer sollen die Lieferkosten angezeigt werden, damit man ein Überblick über die Ausgabe hat beim Bestellen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | mittel |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 2.7 |
| **User Story-Beschreibung** | Als Restaurantbesitzer sollen alle Restaurant-Daten jederzeit änderbar sein, falls sich z. B Kosten oder Kontaktdaten ändern, diese direkt umsetzen zu können. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 3.1 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll man eine Speisekarte erstellen können, um die Gerichte des Restaurants zu repräsentieren. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 3.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll die Speisekarte in mehrere Kategorien unterteil werden wie z.B. Nudelgerichte und Salate, damit das Gewünschte Produkt schneller gefunden werden kann. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 3.3 |

|  |  |
| --- | --- |
| **User Story-ID** | 3.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll die Speisekarte über eine xml.-Datei erstellt werden, um Aufwand zu sparen. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 3.4 |
| **User Story-Beschreibung** | Als Restaurantbesitzer sollen Gerichte Namen, Beschreibung und ein Preis bekommen (optional auch noch ein Bild des Gerichts). Diese sollen sich die Kunden dann anschauen können und darunter eins auswählen. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | hoch |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 3.1, 3.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 3.5 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll die Speisekarte änderbar sein, falls Gerichte dazu kommen oder nicht mehr angeboten werden oder Preise sich ändern. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** |  |

Papierprototypen

Das Erstellen eines Papierprototypen dient als Methode des Brainstormings, Designs, Herstellens, Testens und des Kommunizierens von Benutzer Interfaces.

**Autor: Alexandra Klein**

Startseite:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 1: Startseite

Login-Seite:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 2: Login-Seite

Registrierungsseite:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 3: Registrierungsseite

Nach dem Einloggen oder Anmelden wird auf die Seite der Zwei-Faktor-Authentifizierung weitergeleitet:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 4: 2FA

Nach der Zwei-Faktor-Authentifizierung erscheint kann das Restaurant angelegt werden.



Abbildung 5: Restaurant anlegen

Die Kategorie kann per Drop-Down und der Lieferbereich durch Eingabe in das Kästchen oder durch Verschieben des Reglers ausgewählt werden.



Abbildung 6: Restaurant anlegen

Der Button „Zur Speisekarte“ führt zur Seite, wo eine Speisekarte manuell angelegt oder hochgeladen werden kann.



Abbildung 7: Speisen hinzufügen

## Szenarien (MSCs)

MSC-Diagramme inkl. Bezeichner

**Hauptszenario:** Typische Folge von Interaktionsschritten zur Erreichung des Ziels bzw. der Ziele einer oder mehrerer User Stories

**Alternativszenario:** Alternative Interaktionsschritte, die das Hauptszenario (oder Teile davon) ersetzen und ebenfalls zur Erfüllung der Ziele hinter den User Stories führen

**Ausnahmeszenario:** Interaktionen in Ausnahmefällen, die dazu führen, dass nicht alle Ziele hinter den abgebildeten User Stories erreicht werden

Szenarien dienen als Grundlage zur Definition von Testfällen

**Autor: Leon Gashi**

Szenario: Registrierung

**Restaurantbesitzer**

**System**

**Registrierungswunsch**

**Registrierungsformular**

**Name,Vorname,E**

**-**

**Mail,Passwort**

**when**

Restaurantbesitzer

E

-

Mail

bereits

vergeben

**e**

**xc**

**Registrierung fehlgeschlagen**

**Bestätigung**

**Konto**

**fertigstellen**

**Hauptszenario:** Der Restaurantbesitzer möchte sich anmelden. In dem Anmeldefenster gibt der Restaurantbesitzer seine E-Mail und sein Passwort ein. Der Restaurantbesitzer erhält einen PIN per E-Mail zugeschickt den er zusätzlich eingeben muss, um sich erfolgreich anzumelden.

**Ausnahmeszenario 1:** Wenn die E-Mail und das Passwort nicht übereinstimmen, dann ist die Anmeldung fehlgeschlagen.

**Ausnahmeszenario 2:** Wenn die PIN, welche per E-Mail an den Restaurantbesitzer gesendet wurde, falsch eingegeben wird, so ist auch in diesem Fall die Anmeldung fehlgeschlagen.

Szenario: Login

**Restaurantbesitzer**

**System**

**Anmeldeversuch**

**Anmeldefenster**

**E**

**-**

**Mail,**

**Passwort**

**when**

Restaurantbesitzer

E

-

Mail

und Passwort

stimmen nicht überein

**e**

**xc**

**Anmeldung**

**fehlgeschlagen**

**PIN eingeben**

**Schicke PIN**

**PIN erhalten**

**exc**

**when**

Restaurantbesitzer

PIN stimmt nicht überein

**Anmeldung fehlgeschlagen**

**Anmeldung erfolgreich**

**Hauptszenario:** Der Restaurantbesitzer möchte ein Konto anlegen. Nachdem er das vom System gegebene Registrierungsformular mit persönlichen Daten ausgefüllt hat, wird sein Konto im System erstellt und der Restaurantbesitzer erhält eine Bestätigung.

**Ausnahmeszenario:** Wenn der Restaurantbesitzer eine E-Mail angibt, welche bereits vergeben ist, so ist die Registrierung fehlgeschlagen.

**Autor: Oguzhan Kaluk**

Szenario: Restaurantdaten erstellen



**Hauptszenario:** Der Restaurantbesitzer möchte sein Restaurant im System anlegen. Das System fordert die Eingabe der folgenden Daten: Name, Adresse, Kategorie, MBW, Lieferbereich, Lieferkosten. Diese werden in der Datenbank hinterlegt und dem Restaurantbesitzer wird die Durchführung bestätigt.

**Autor: Oguzhan Kaluk**

Szenario: Restaurantdaten ändern



**Hauptszenario:** Der Restaurantbesitzer möchte das im System hinterlegte Restaurant bearbeiten. Hierzu kann er angeben, was verändert werden soll und den neuen Inhalt bestimmen. Dies wird in der Datenbank aktualisiert und dem Restaurantbesitzer wird die Durchführung bestätigt.

**Autor: Oguzhan Kaluk**

Szenario: Speisekarte anlegen



**Hauptszenario:** Der Restaurantbesitzer möchte eine Speisekarte im System anlegen. Diese kann per .xml-Datei angelegt werden. Dazu wird die Datei mit den Spalten Kategorie, Name, Beschreibung, Preis und URL hochgeladen. Die Daten werden dann in der Datenbank hinterlegt und die Änderung wird den Restaurantbesitzer bestätigt. Alternativ ist das Manuelle Erstellen möglich, dazu wird jeder Eintrag einzeln durchgeführt. Jeweils wird eine Änderung in der Datenbank durchgeführt und der Restaurantbesitzer bekommt eine Bestätigung der Änderung.

**Alternativszenario:** Alternativ ist das Manuelle Erstellen möglich, dazu wird jeder Eintrag einzeln durchgeführt. Jeweils wird eine Änderung in der Datenbank durchgeführt und der Restaurantbesitzer bekommt eine Bestätigung der Änderung.

**Autor: Oguzhan Kaluk**

Szenario: Speisekarte ändern



**Hauptszenario:** Der Restaurantbesitzer möchte die im System hinterlegte Speisekarte verändern. Er wählt den zu ändernden Eintrag aus und gibt die neuen Werte ein. Diese werden dann im System aktualisiert und die Änderung wird dem Restaurantbesitzer bestätigt.

**Alternativszenario:** Alternativ ist das Löschen einzelner Einträge möglich. Dazu wird der zu löschende Eintrag ausgewählt und bestätigt. Die Datenbank wird aktualisiert und der Restaurantbesitzer bekommt eine Bestätigung.

## Strukturdiagramm (Klassendiagramm)

Ein Strukturdiagramm dient der grafischen Darstellung von Klassen, Schnittstellen und deren Beziehungen. Es hilft dabei, Quellcode und Implementierungsarbeiten zu strukturieren, bevor diese starten und ermöglicht eine Aufteilung der Programmieraufgaben.

Das UML Klassendiagramm wurde in Front-End und Back-End unterteilt. Diese werden aufgrund der Lesbarkeit noch als Anhang beigefügt.

Front-End:



Back-End:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Funktionalitätsplanung

Anhand der Funktionalitätsplanung werdet Ihr geprüft. Diese sollte dementsprechend **immer** aktuell gehalten werden und pro Funktionalität **nur einen** Verantwortlichen enthalten. Als Quellcodereferenz solltet Ihr immer euer Package, eure Klasse und die dazugehörige Methode angeben. ggf. könnt Ihr auch Zeilenangaben machen.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Login** |  |  |  |  |  |  |
| 1.1 | Login anlegen | 1 Tag | Leon Gashi | - |  | LoginController | fertig |
| 1.2 | Validierung der Login Daten | 2 Tage | Nick Waldschmidt | - |  | LoginController | fertig |
| 1.2.1 | Erfolgreiche Anmeldung, zwei-Faktor-Authentifizierung per E-Mail | 3 Tage | Nick Waldschmidt | - | 1.3 | VerificationController | fertig |
| 1.2.2 | Verifizierungscode korrekt, Weiterleitung zur Startseite | 2 Tage | Nick Waldschmidt | - | 1.3 | VerificationController | fertig |
| 1.2.3 | Pop-Up Fehlermeldung bei falschen Anmeldedaten | 1 Tag | Nick Waldschmidt | - | - | LoginController | fertig |
| **2.** | **Profil** |  |  |  |  |  |  |
| 2.1 | Registrierungs-GUI | 1 Tag | Leon Gashi | - | - |  | fertig |
| 2.1.1 | Registrierungsdaten in Datenbank speichern | 1 Tag | Leon Gashi | - | 1.1 | RegistrationController | fertig |
| 2.2 | Gespeicherte Daten ändern | 1 Tag | Leon Gashi | - |  | EditProfileController | fertig |
| 3 | Startseite |  |  |  |  |  |  |
| 3.1 | GUI Anlegen | 1 Tage | Leon Gashi | - | - |  | fertig |
| 3.2 | Startseite | 1 Tag | Alexandra Klein |  |  | StartseiteController | fertig |
| 4 | Restaurant |  |  |  |  |  |  |
| 4.1 | GUI Anlegen | 1 Tag | Alexandra Klein | - | - |  | fertig |
| 4.2 | Restaurant Anlegen | 2 Tage | Alexandra Klein | - | 2.1, 2.2, 2.5 und 2.6 | AddRestaurantController | fertig |
| 4.3 | Lieferbereich anlegen | 1 Tag | Alexandra Klein | - | 2.3 | AddRestaurantController | fertig |
| 4.4 | Karten Markierung Anzeigen | 1 Tag | Oguzhan Kaluk | - | 2.4 | MapController | fertig |
| 4.5 | Restaurantdaten sind bearbeitbar | 2 Tage | Alexandra Klein | - | 2.7 | EditRestaurantController | fertig |
| 5 | Speisekarte |  |  |  |  |  |  |
| 5.1 | GUI Anlegen | 1 Tag | Dursun Berat Saral | - | - |  | fertig |
| 5.2 | Speisekarte Anlegen | 2 Tage | Dursun Berat Saral | - | 3.1 | AddFoodController | fertig |
| 5.2.1 | Kategorien anlegen | 1 Tag | Dursun Berat Saral | 5.6 | 3.2 | AddFoodController | fertig |
| 5.3 | Speisen Hinzufügen | 2 Tage | Dursun Berat Saral | - | 3.4 | AddFoodController | fertig |
| 5.4 | Speisen Bearbeiten | 2 Tage | Dursun Berat Saral | - | 3.5 | EditFood2Controller  EditFoodController | fertig |
| 6 | Datenbank |  |  |  |  |  |  |
| 6.1 | Datenbank initialisieren | 10 std | Nick Waldschmidt  Oguzhan Kaluk | - | - | Model:  -> Food  -> Menu  Model:  -> Restaurant  -> User | fertig |
| 6.2 | Daten persistent speichern | 8 std | Nick Waldschmidt  Oguzhan Kaluk | - | 1.2 | Repo:  -> FoodRepo  -> MenuRepo  Repo:  -> RestaurantRepo  -> UserRepo | fertig |
| 7 | Server-Client-Netzwerk |  |  |  |  |  |  |
| 7.1 | Verbindung zum Server aufbauen | 15 std | Oguzhan Kaluk | - | - | ServerApplication | fertig |
| 7.2 | XML-Umwandlung in JSON | 15 std | Nick Waldschmidt |  |  | xmlController | fertig |
| 7.2.1 | XML-Speicherung in Datenbank | 1 Tag | Nick Waldschmidt |  | 3.3 | xmlController | fertig |
| 7.3 | HTTP Anbindung | 2 std | Oguzhan Kaluk | - | - | FoodController  MenuController  RestaurantController  UserController | fertig |
| 7.4 | ConnectionController dauerhafte Verbindung zum Server | 3 Tage | Nick Waldschmidt | - | - | ConnectionController. | fertig |
| 7.5 | Service implementation | 3 Tage | Oguzhan Kaluk | - | - | FoodService  MenuService  RestaurantService  UserService | fertig |
| 7.5.1 | Klassen (Entity) in Datenbanken einbinden | 1 Tag | Oguzhan Kaluk | - | - | Model  -> Food  -> Menu  -> User  -> Restaurant  Repo  -> FoodRepo  -> MenuRepo  -> RestaurantRepo  -> UserRepo | fertig |

## Systemtests

Systemtests sind Tests des Gesamtsystems gegen die Anforderungen nach erfolgreicher Integration. Eingaben und Sollverhalten werden dabei aus der Anforderungs-spezifikation abgeleitet.

Die Systemtests werden von Eurer Parallelgruppe spezifiziert und durchgeführt, daher ist dieser Bereich von den Mitgliedern der Parallelgruppe auszufüllen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | Test nicht bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 15.11.2021 | | |
| **Tester** | Oguzhan Kaluk | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Der Nutzer „Max Mustermann“ ist noch nicht im System registriert und befindet sich auf der Registrierungsseite. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt seinen Vor- und Nachnamen in die Felder „Vorname“ und Nachname“ ein. | Der Vorname „Max“ und der Nachname „Mustermann“ werden in dem Feld angezeigt. | **√** |
| 2 | Der Benutzer gibt die E-Mail-Adresse „maxmustermann@sep.de“ im Feld „E-Mail“ ein | Die E-Mail-Adresse wird in dem Feld angezeigt. | **√** |
| 3 | Der Benutzer gibt das Passwort „1234“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 4 | Der Benutzer klickt auf „Registrieren“. | Das System zeigt die Meldung „Registrierung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Der Nutzer ist im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 15.11.2021 | | |
| **Tester** | Oguzhan Kaluk | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedingung(en)** | Der Nutzer „Max Mustermann“ ist im System registriert und befindet sich auf der Login-Seite. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt die E-Mail-Adresse „maxmustermann@sep.de“ im Feld „E-Mail“ ein | Die E-Mail-Adresse wird in dem Feld angezeigt. | **√** |
| 2 | Der Benutzer gibt das Passwort „1234“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Der Benutzer wurde erfolgreich angemeldet und wird auf die Seite der 2-Faktor-Authentifizierung weitergeleitet. | **√** |
| 4 | Der Benutzer gibt den ihm übersendeten Code für die 2-Faktor-Authentifizierung im Feld ein. | Der Code wird in dem Feld angezeigt. | **√** |
| 5 | Der Benutzer klickt auf „Anmelden“. | Der Benutzer wurde erfolgreich angemeldet und befindet sich auf der Startseite des Menüs. | **√** |
| (Ausnahmeszenario)  2a | Der Benutzer gibt das Passwort „abcd“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 2a1 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt eine Fehlermeldung "Error: Daten nicht Korrekt" an und es wird nicht auf die Seite der 2-Faktor-Authentifizierung weitergeleitet. | **√** |
| **Nachbe-dingung(en)** | Der Nutzer ist im System angemeldet. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 15.11.2021 | | |
| **Tester** | Oguzhan Kaluk | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Der Nutzer „Max Mustermann“ ist im System registriert und befindet sich im Fenster „Restaurant anlegen“. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt in das Feld „Name“ den Restaurantnamen „Pizzahouse“ ein. | Der Name „Pizzahouse“ wird in dem Feld „Name“ angezeigt. | **√** |
| 2 | Der Benutzer gibt in das Feld „Straße“ die Straße „Schützenbahn 70“ ein. | Die Straße „Schützenbahn 70“ wird in dem Feld „Straße“ angezeigt. | **√** |
| 3 | Der Benutzer gibt in das Feld „PLZ“ die PLZ „45127“ ein. | Die PLZ „45127“ wird in dem Feld „PLZ“ angezeigt. | **√** |
| 4 | Der Benutzer gibt in das Feld „Stadt“ die Stadt „Essen“ ein. | Die Stadt „Essen“ wird in dem Feld „Stadt“ angezeigt. | **√** |
| 5 | Der Benutzer gibt in das Feld „Mindestbestellwert“ den Mindestbestellwert „10.60“ ein. | Der Mindestbestellwert „10.60“ wird in dem Feld „Mindestbestellwert“ angezeigt. | **√** |
| 6 | Der Benutzer gibt in das Feld „Lieferkosten“ den Lieferkosten „1.50“ ein. | Die Lieferkosten „1.50“ werden in dem Feld „Lieferkosten“ angezeigt. | **√** |
| 7 | Der Benutzer gibt in das Feld „Lieferbereich“ den Lieferbereich „5“ ein. | Der Lieferbereich „5“ wird in dem Feld „Lieferbereich angezeigt. | **√** |
| 8 | Der Benutzer wählt in dem Drop-Down Menü die Kategorie „Deutsch“ aus. | Die Kategorie „Deutsch“ wird in dem Feld „Kategorie“ angezeigt. | **√** |
| 9 | Der Benutzer klickt auf „Speichern“. | Die Daten wurden gespeichert. Es erscheint keine Meldung und es wird zurück auf die Startseite des Menüs weitergeleitet. | **√** |
| **Nachbe-dingung(en)** | Das Restaurant „Pizzahouse“ wurde angelegt. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 15.11.2021 | | |
| **Tester** | Oguzhan Kaluk | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Der Nutzer „Max Mustermann“ ist im System registriert und befindet sich im Fenster „Speise hinzufügen“. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer wählt in dem Drop-Down Menü die Kategorie „Pizza“ aus. | Die Kategorie „Pizza“ wird in dem Feld „Kategorie“ angezeigt. | **√** |
| 2 | Der Benutzer gibt in dem Feld „Name“ „Pizza Tonno“ ein. | Der Name „Pizza Tonno“ wird in dem Feld „Name“ angezeigt. | **√** |
| 3 | Der Benutzer gibt in dem Feld „Beschreibung“ „Pizza mit Thunfisch“ ein. | Die Beschreibung „Pizza mit Thunfisch“ wird in dem Feld „Beschreibung angezeigt. | **√** |
| 4 | Der Benutzer gibt in das Feld „Preis“ „5.50“ ein. | Der Preis „5.50“ wird in dem Feld „Preis“ angezeigt. | **√** |
| 5 | Der Benutzer klickt auf „Hinzufügen“. | Die Speise wird hinzugefügt. | **√** |
| 6 | Der Benutzer klickt auf „fertig“. | Die Speisen werden zur Speisekarte hinzugefügt,  Weiterleitung zur Startseite | **√** |
| **Nachbe-dingung(en)** | Die Speise „Pizza Tonno“ wurde hinzugefügt. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 15.11.2021 | | |
| **Tester** | Oguzhan Kaluk | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Der Nutzer „Max Mustermann“ ist im System registriert und hat die Speise „Pizza Tonno“ hinzugefügt. Er befindet sich in dem Fenster „Speise bearbeiten“. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer trägt die SpeisenID ein und klickt auf „Bearbeiten“. | Die Speise kann bearbeitet werden. | **√** |
| 2 | Der Benutzer wählt in dem Drop-Down Menü die Kategorie „Pizza“ aus. | Die Kategorie „Pizza“ wird in dem Feld „Kategorie“ angezeigt. | **√** |
| 3 | Der Benutzer gibt in dem Feld „Name“ „Pizza Funghi“ ein. | Der Name „Pizza Funghi“ wird in dem Feld „Name“ angezeigt. | **√** |
| 4 | Der Benutzer gibt in dem Feld „Beschreibung“ „Pizza mit Pilzen“ ein. | Die Beschreibung „Pizza mit Pilzen“ wird in dem Feld „Beschreibung angezeigt. | **√** |
| 5 | Der Benutzer gibt in das Feld „Preis“ „5.00“ ein. | Der Preis „5.00“ wird in dem Feld „Preis“ angezeigt. | **√** |
| 6 | Der Benutzer klickt auf „Hinzufügen“. | Die Speise wird hinzugefügt. | **√** |
| 7 | Der Benutzer klickt auf „fertig“. | Die Speisen werden zur Speisekarte hinzugefügt,  Weiterleitung zur Startseite | **√** |
| **Nachbe-dingung(en)** | Die Speise wurde von „Pizza Tonno“ auf „Pizza Funghi“ geändert. | | **√** |
| **Testurteil** | Test bestanden. | | |

# Zyklus II

## Spezifikationsplanung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | Kundenkonto anlegen |  |  |  |
| 1.1 | * Konto erstellen * Guthaben für Essensbestellungen anzeigen * Alternativadresse * Bestellhistorie | User Story | Alexandra Klein | Fertig |
| 1.2 | * Konto erstellen * Guthaben/Treuepunkte für Essensbestellungen anzeigen * Alternativadresse * Bestellhistorie * Profil bearbeiten * Startseite | Papierprototyp | Alexandra Klein | Fertig |
| 1.3 | * Registrierung | MSC | Leon Gashi | Fertig |
| **2.** | Auswahl von Restaurants anzeigen |  |  |  |
| 2.1 | * Restaurant suchen * Restaurant in der Nähe (Map) * Restaurant filtern * Restaurant als Favorit markieren | User Story | Oguzhan Kaluk | Fertig |
| 2.2 | * Restaurant suchen * Restaurant in der Nähe (Map) * Restaurant filtern * Restaurant als Favorit markieren | Papierprototyp | Oguzhan Kaluk | Fertig |
| 2.3 | * Restaurant als Favorit markieren * Alternativadresse hinzufügen | MSC | Oguzhan Kaluk | Fertig |
| **3.** | Essensbestellungen |  |  |  |
| 3.1 | * Guthaben aufladen * Speisen zur Bestellung hinzufügen * Bestätigung per E-Mail * Bestellung der Bestellliste hinzufügen * Bestellübersicht | User Story | Nick Waldschmidt  Dursun Berat Saral | Fertig |
| 3.2 | * Guthaben aufladen * Warenkorb * Bestellübersicht * Bestellung der Bestellliste hinzufügen | Papierprototyp | Nick Waldschmidt | Fertig |
| 3.3 | * Guthaben aufladen * Bestellung hinzufügen * Bestellhistorie anzeigen | MSC | Dursun Berat Saral | Fertig |
| **4.** | Bewertung |  |  |  |
| 4.1 | * Bewertung hinzufügen * Bewertungsliste * Restaurantliste | User Story | Alexandra Klein | Fertig |
| 4.2 | * Bewertung hinzufügen * Bewertungsliste * Restaurantliste | Papierprototyp | Alexandra Klein | Fertig |
| 4.3 | * Bewertung hinzufügen | MSC | Alexandra Klein | Fertig |
| **5.** | Treuepunkte |  |  |  |
| 5.1 | * Treuepunkte anzeigen * Coupon per E-Mail * Rabattcode hinzufügen | User Story | Leon Gashi | Fertig |
| 5.2 | * Treuepunkte anzeigen * Rabattcode hinzufügen | Papierprototyp | Alexandra Klein | Fertig |
| 5.3 | * Treuepunkte | MSC | Leon Gashi | Fertig |
| **6.** | UML-Klassendiagramm |  |  |  |
| 6.1 | Klassendiagramm | Front-End | Nicklas Czorny | Fertig |
| 6.2 | Klassendiagramm | Back-End | Nicklas Czorny | Fertig |

## User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | 4.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich ein Kundenkonto anlegen, damit ich bei der Webseite Speisen bestellen kann. |
| **Geschätzter Realisierungsaufwand** | 10 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 4.2 |
| **User Story-Beschreibung** | Als Kunde möchte ich mein Guthaben angezeigt bekommen, damit ich weiß wie viel auf dem Guthabenkonto zur Verfügung steht. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 4.3 |
| **User Story-Beschreibung** | Als Kunde möchte ich eine Bestellhistorie angezeigt bekommen, damit ich einen Überblick über meine Bestellungen und Restaurants habe. |
| **Geschätzter Realisierungsaufwand** | 10 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 4.4 |
| **User Story-Beschreibung** | Als Kunde möchte ich eine Alternativadresse eingeben können, wenn ich nicht zu Hause bin, damit das Essen an einen anderen Ort geliefert werden kann. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 2.3 |

|  |  |
| --- | --- |
| **User Story-ID** | 5.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich in der Lage sein, ein mir passendes Restaurant finden zu können, um eine Bestellung tätigen zu können. |
| **Geschätzter Realisierungsaufwand** | 10 Std. |
| **Priorität** | hoch |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 5.2 |
| **User Story-Beschreibung** | Als Kunde möchte ich die Namen, Kategorien und die durchschnittliche Bewertung der Restaurants sehen, um ein mir passendes Restaurant finden zu können. |
| **Geschätzter Realisierungsaufwand** | 10 Std. |
| **Priorität** | hoch |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 5.3 |
| **User Story-Beschreibung** | Als Kunde möchte ich in der Lage sein, eine alternative Lieferadresse auswählen zu können, damit ich unabhängig meines Aufenthalts Essen bestellen kann. |
| **Geschätzter Realisierungsaufwand** | 5 Std. |
| **Priorität** | niedrig |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** | 1.4 |

|  |  |
| --- | --- |
| **User Story-ID** | 5.4 |
| **User Story-Beschreibung** | Als Kunde möchte ich nur die Restaurants sehen, in deren Lieferbereich ich mich befinde, damit ich nicht negative Rückmeldungen bekomme bei der Essensbestellung. |
| **Geschätzter Realisierungsaufwand** | 15 Std. |
| **Priorität** | mittel |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 5.5 |
| **User Story-Beschreibung** | Als Kunde möchte ich die Möglichkeit haben, die Speisekarten der jeweiligen Restaurants anzusehen, damit ich im Vorfeld entscheiden kann, ob dieses Restaurant zu mir passt oder nicht. |
| **Geschätzter Realisierungsaufwand** | 5 Std. |
| **Priorität** | mittel |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 5.6 |
| **User Story-Beschreibung** | Als Kunde möchte ich eine Karte sehen, in der meine Lieferadresse im Mittelpunkt ist. Damit kann ich Restaurants in meiner unmittelbaren Umgebung sehen. |
| **Geschätzter Realisierungsaufwand** | 15 Std. |
| **Priorität** | niedrig |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 5.6.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich auf der Karte nur die Restaurants sehen, in deren Lieferbereich ich mich befinde. |
| **Geschätzter Realisierungsaufwand** | 5 Std. |
| **Priorität** | niedrig |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** | 2.6 |

|  |  |
| --- | --- |
| **User Story-ID** | 5.7 |
| **User Story-Beschreibung** | Als Kunde möchte ich Restaurants als Favoriten setzen können, um diese schneller bei weiteren Essensbestellungen finden zu können. |
| **Geschätzter Realisierungsaufwand** | 5 Std. |
| **Priorität** | niedrig |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 6.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich mein Guthaben aufladen können, um eine Bestellung abzuschließen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 3.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 6.2 |
| **User Story-Beschreibung** | Als Kunde will ich einen beliebigen Betrag auf mein Konto gutschreiben können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | mittel |
| **Autor(en)** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 3.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 6.3 |
| **User Story-Beschreibung** | Als Kunde möchte ich Gerichte aus der Speisekarte zu meiner Bestellung hinzufügen, damit ich ein überblick im Warenkorb habe. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 6.4 |
| **User Story-Beschreibung** | Als Kunde muss ich ausreichend Guthaben besitzen, um die Bestellung abschließen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | mittel |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 3.4 |

|  |  |
| --- | --- |
| **User Story-ID** | 6.5 |
| **User Story-Beschreibung** | Als Kunde möchte ich nach dem Bestellvorgang eine Bestätigungsmail erhalten, um sicher zu gehen, dass meine Bestellung eingegangen ist. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 3.7 |

|  |  |
| --- | --- |
| **User Story-ID** | 6.6 |
| **User Story-Beschreibung** | Als Kunde soll meine Bestellung in einer Bestellhistorie aufzufinden sein, um meine letzten Bestellungen nachzuverfolgen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | mittel |
| **Autor(en)** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 3.6, 3.8 |

|  |  |
| --- | --- |
| **User Story-ID** | 6.7 |
| **User Story-Beschreibung** | Als Kunde will ich in der Bestellhistorie Name des Restaurants, die Bestellten Gerichte, deren Bestelldatum und Preise angezeigt bekommen, um mein Konsumverhalten im Überblick zu haben. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | niedrig |
| **Autor(en)** | Dursun Berat Saral |
| **Abhängigkeiten zu anderen User Stories** | 3.7 |

|  |  |
| --- | --- |
| **User Story-ID** | 7.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich mein Essen und meine Lieferung mit 1-5 Sternen bewerten, damit ich ein Feedback geben kann. |
| **Geschätzter Realisierungsaufwand** | 15 Std. |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 4.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 7.2 |
| **User Story-Beschreibung** | Als Kunde möchte ich einen Kommentar zu meiner Bewertung hinzufügen können, damit ich ein detailliertes Feedback zu meiner Bewertung geben kann. |
| **Geschätzter Realisierungsaufwand** | 15 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 4.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 7.3 |
| **User Story-Beschreibung** | Als Kunde möchte ich alle Bewertungen des Restaurants lesen können, wenn ich dort bestelle, damit ich mir ein Bild von dem Restaurant mache kann. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 2.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 7.4 |
| **User Story-Beschreibung** | Als Kunde möchte ich mich durch die einzelnen Bewertungen klicken, damit ich diese lesen kann. |
| **Geschätzter Realisierungsaufwand** | 10 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 7.5 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich mich durch die Bewertungen klicken, damit ich das Feedback lesen kann. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 8.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich einen Coupon per Mail zugeschickt bekommen, welchen ich für Rabatte nutzen kann, nachdem ich 10 Treuepunkte gesammelt habe. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Niedrig |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 8.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass der Kunde bei einer Bestellung für jede fünf Euro, die er ausgibt, einen Treuepunkt bekommt. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 8.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass der Kunde einen Coupon per Mail zugeschickt bekommt, nachdem er 10 Treuepunkte gesammelt hat. Dieser soll explizit für Rabatte eingesetzt werden. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Niedrig |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** |  |

Papierprototypen

### Kundenkonto

Autor: Alexandra Klein

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 8: Kundenkonto erstellen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 9: Treuepunkte/Guthaben anzeigen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 10: Bestellhistorie über alle Restaurants



Abbildung 11: Profil bearbeiten

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 11: Alternativadresse hinzufügen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 12: Startseite

### Restaurant

Autor: Oguzhan Kaluk

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 13: Restaurants anzeigen

### Essensbestellung

Autor: Nick Waldschmidt

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 14: Guthaben aufladen

Ein Bild, das Platz enthält.

Automatisch generierte Beschreibung

Abbildung 15: Warenkorb

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 16: Bestellüberblick

### Bewertung

Autor: Alexandra Klein



Abbildung 17: Bewertung hinzufügen



Abbildung 18: Bewertungen anzeigen



Abbildung 19: Restaurantliste mit durchschnittlicher Bewertung

## Szenarien (MSCs)

Autor: Leon Gashi

Registrierung Kunde:



**Hauptszenario:** Der Kunde möchte ein Konto anlegen. Nachdem er das vom System gegebene Registrierungsformular mit persönlichen Daten ausgefüllt hat, wird sein Konto im System erstellt und der Kunde erhält eine Bestätigung.

**Ausnahmeszenario:** Wenn der Kunde eine E-Mail angibt, welche bereits vergeben ist, so ist die Registrierung fehlgeschlagen.

Autor: Oguzhan Kaluk

Restaurant als Favorit markieren:



**Hauptszenario:** Der Kunde möchte ein Restaurant als Favorit markieren. Dafür möchte das System wissen, welches Restaurant als Favorit markiert werden soll. Das vom Kunden übermittelte Restaurant wird dem Kunden als Favorit zugewiesen und der Kunde erhält eine Bestätigung.

Autor: Oguzhan Kaluk

Alternativadresse eingeben:



**Hauptszenario:** Der Kunde möchte bei der Essensbestellung die alternative Adresse nutzen. Das System überprüft, ob diese vorhanden ist. Das System ruft diese Daten auf und der Kunde wird zurück zur Liste der Restaurants weitergeleitet.

**Ausnahmeszenario:** Bei Nicht-Vorhanden-Sein der alternativen Adresse, wird der Kunde ein Formular zur Eingabe der alternativen Adresse gegeben. Diese werden dann genutzt und der Kunde wird zur Liste der Restaurants weitergeleitet.

Autor: Dursun Berat Saral

Guthaben aufladen:

Kunde

Server

Guthaben Aufladungswunsch

Guthaben Formular

Anzahl von Guthaben (int wert)

Aufladung Erfolgreich

Füge Guthaben hinzu

**Hauptszenario:** Der Kunde möchte sein Guthaben aufladen, wenn er vom System das Auflade Formular mit einem bestimmten Guthaben gefüllt hat, wird sein Kontostand im System gespeichert. Dann bekommt der Kunde eine Bestätigung, dass die Aufladung erfolgreich war.

Autor: Dursun Berat Saral

Bestellung hinzufügen:

Sende E-Mail

Wunsch, Gerichte in Bestellung aufnehmen

Kunde Server

Bestellung Abschließen

WHEN

Zu wenig Guthaben

Oder

Mindestbestellwert nicht erreicht

Mindestbestellwert nicht erreicht

exc

Bestellung fehlgeschlagen

Bestätigung der Bestellung

**Hauptszenario:** Der Kunde will eine Speise in die Bestellung aufnehmen und die Bestellung abschließen. Das System schickt dem Kunden eine Bestätigung der Bestellung. Zusätzlich schickt das System eine Bestätigungs-E-Mail an den Kunden.

**Ausnahmeszenario:** Wenn der Kunde zu wenig Guthaben auf seinem Konto hat oder den mindestbestellwert nicht erreicht, dann ist die Bestätigung fehlgeschlagen.

Autor: Dursun Berat Saral

Bestellübersicht anzeigen:

Kunde Server

Bestellhistorie anzeigen wunsch

Bestellhistorie anzeigen

Daten schicken

**Hauptszenario:** Der Kunde will seine Bestellhistorie aufrufen. Das System schickt dann dem Kunden die Daten und zeigt dem Kunden die Bestellhistorie.

Autor: Alexandra Klein

Bewertung hinzufügen:



**Hauptszenario:** Der Kunde möchte eine Bewertung zu dem Essen und der Lieferung hinzufügen. Dieser kann dabei eine Sterne-Bewertung mit eins bis fünf Sternen geben und einen Kommentar hinzufügen. Das System speichert die Daten und diese werden in der Bewertungsliste des Restaurants angezeigt.

Autor: Leon Gashi

Treuepunkte:



**Hauptszenario:** Der Kunde bekommt für jede 5 Euro, die er in seiner Bestellung ausgibt, einen Treuepunkt.

**Alternativ:** Wenn der Kunde 10 Treuepunkte hat, bekommt er einen Coupon per Mail zugesandt, der als Rabatt dient.

## Strukturdiagramm (Klassendiagramm)

Autor: Nicklas Czorny

Das UML Klassendiagramm wurde in Front-End und Back-End unterteilt. Diese werden aufgrund der Lesbarkeit noch als Anhang beigefügt.

Front-End

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Back-End

Klassendiagramm aus dem ersten Zyklus:



## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **8.** | **Kundenkonto anlegen** |  |  |  |  |  |  |
| 8.1 | Kundenkonto erstellen | 4 h | Nicklas Czorny |  | 4.1 | KundeRegistrationController | Fertig |
| 8.1.2 | Kundendaten ändern | 3 h | Nicklas Czorny |  |  | EditProfileController | Fertig |
| 8.1.3 | Guthaben für das Kundenkonto erstellen | 6 h | Dursun Berat Saral |  | 4.2 | GuthabenAufladenController | Fertig |
| 8.2 | Bestellhistorie erstellen | 6 h | Nick Waldschmidt |  | 4.3 | bestellHistorieController | Fertig |
| 8.3 | Alternativadresse erstellen | 4 h | Nicklas Czorny |  | 4.4 | EditAltProfileController | Fertig |
| 8.4 | Startseite | 2 h | Nicklas Czorny |  |  | KundeStartseiteController | Fertig |
| **9.** | **Auswahl von Restaurants** |  |  |  |  |  |  |
| 9.1. | Restaurants filtern | 5 h | Oguzhan Kaluk |  |  | RestaurantsController | Fertig |
| 9.1.2 | Restaurants als Favorit markieren | 3 h | Oguzhan Kaluk |  | 5.7 | RestaurantsController | Fertig |
| 9.2 | Restaurants auf Karte im Lieferbereich anzeigen | 8 h | Oguzhan Kaluk |  | 5.6.1 | RestaurantsController | Fertig |
| 9.3 | Lieferbereich zu der Kundenadresse anzeigen lassen | 4 h | Oguzhan Kaluk |  | 5.4, 5.6 | RestaurantsController | Fertig |
| 9.4 | Speisekarte zum Restaurant anzeigen | 5 h | Dursun Berat Saral |  | 5.5 | MenuSpeisekarteeController | Fertig |
| 9.5 | HTTP Anbindung | 2 h | Oguzhan Kaluk |  |  | FavController | Fertig |
| 9.6 | Klassen in Datenbank einbinden | 2 h | Oguzhan Kaluk |  |  | Model: Fav  Repo: FavRepo | Fertig |
| 9.7 | Service Implementation | 2 h | Oguzhan Kaluk |  |  | FavService | Fertig |
| 9.8 | JUNIT-Test Distanzerrechnung | 2 h | Oguzhan Kaluk |  |  | DistanceTest | Fertig |
| **10.** | **Essensbestellung** |  |  |  |  |  |  |
| 10.1 | Guthaben aufladen | 5 h | Dursun Berat Saral |  | 6.1 | GuthabenAufladenController | Fertig |
| 10.2 | Gerichte zum Warenkorb hinzufügen | 5 h | Nick Waldschmidt |  | 6.3 | WarenkorbController | Fertig |
| 10.3 | Bestätigungsmail versenden | 6 h | Nick Waldschmidt |  | 6.6 | WarenkorbController | Fertig |
| 10.3.1 | E-Mail-Server | 4 h | Nick |  |  | UserController | Fertig |
| 10.4 | Gericht der Bestellhistorie hinzufügen | 3 h | Dursun Berat Saral |  | 6.7, 6.8 | bestellHistorieController | Fertig |
| 10.5 | Klassen in Datenbank einbinden | 5 h | Dursun Berat Saral |  |  | Model:   * Bestellung * BestellFood * BestellHistorie * Warenkorb * WarenFood   Repo:   * BestellRepo * BestellfoodRepo * BestellHisRepo * WarenkorbRepo * WarenFoodRepo | Fertig |
| 10.6 | HTTP Anbindung | 2 h | Nick Waldschmidt |  |  | BestellungController  WarenkorbController | Fertig |
| 10.7 | Service Implementation | 5 h | Nick Waldschmidt |  |  | * BestellungService * BestellfoodServie * BestellHisService * WarenkorbService * WarenFoodService | Fertig |
| 10.8 | JUNIT-Test Bestellhistorie | 3 h | Nicklas Czorny |  |  | BestellHisRepoTest | Fertig |
| **11.** | **Bewertung** |  |  |  |  |  |  |
| 11.1 | Essensbewertung hinzufügen | 6 h | Alexandra Klein |  | 7.1, 7.2 | BewertungController | Fertig |
| 11.1.2 | Lieferbewertung hinzufügen | 3 h | Alexandra Klein |  | 7.1, 7.2 | BewertungController | Fertig |
| 11.1.3 | Bewertungsliste | 4 h | Alexandra Klein |  | 7.3, 7.4, 7.5 | BewertungListController | Fertig |
| 11.2 | Durchschnittliche Bewertungen anzeigen | 4 h | Alexandra Klein |  | 5.2 | BewertungController | Fertig |
| 11.3 | Klassen in Datenbank einbinden | 5 h | Oguzhan Kaluk |  |  | Model: Rating  Repo: RatingRepo | Fertig |
| 11.4 | HTTP Anbindung | 2 h | Oguzhan Kaluk |  |  | RatingController | Fertig |
| 11.5 | Service Implementation | 5 h | Oguzhan Kaluk |  |  | RatingService | Fertig |
| 11.6 | JUNIT-Test zur durchschnittlichen Bewertung anzeigen | 2 h | Alexandra Klein |  |  | AvgBewertungTest | Fertig |
| **12.** | **Treuepunkte** |  |  |  |  |  |  |
| 12.1 | Treuepunkte zu dem Kundenkonto hinzufügen | 6 h | Leon Gashi |  | 8.2, 8.2 | TreuepunkteController | Fertig |
| 12.2 | Coupon per E-Mail versenden | 4 h | Leon Gashi |  | 8.3 | TreuepunkteController | Fertig |
| 12.2.1 | Coupon hinzufügen und speichern | 2 h | Leon Gashi |  | 8.3 | TreuepunkteController | Fertig |
| 12.3 | Klassen in Datenbank einbinden | 5 h | Oguzhan Kaluk |  |  | Model: Coupon  Repo: CouponService | Fertig |
| 12.4 | HTTP Anbindung | 2 h | Oguzhan Kaluk |  |  | CouponController | Fertig |
| 12.5 | Service Implementation | 5 h | Oguzhan Kaluk |  |  | CouponService | Fertig |

## Unittests

Unittest, auch Modultests, sind Komponententests. Diese werden in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Units) von Computerprogrammen zu testen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **1** | Es wird geprüft, ob das Datum der Bestellhistorie mit dem tatsächlichen Datum übereinstimmt. | Server/src/test/java/ude/sep/repo/BestellHisRepoTest  derTest1() Zeile:64-69 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |
| **1.1** | Es wird geprüft, ob der Name des Gerichts mit dem tatsächlichen Gericht übereinstimmt. | Server/src/test/java/ude/sep/repo/BestellHisRepoTest  derTest2() Zeile:71-75 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |
| **1.2** | Es wird geprüft, ob das bestellte Gericht mit dem tatsächlichen Gericht übereinstimmt. | Server/src/test/java/ude/sep/repo/BestellHisRepoTest  derTest3() Zeile:78-82 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |
| **1.3** | Es wird geprüft, ob der Betrag mit dem tatsächlichen Betrag übereinstimmt. | Server/src/test/java/ude/sep/repo/BestellHisRepoTest  derTest4() Zeile: 84-88 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |
| **2** | Der Test überprüft, ob ein Restaurant korrekt mit den richtigen Daten in die Datenbank geschrieben wurde. | Testen/src/main/java/AvgBewertungTest  restaurantTest() Zeile: 161-178 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |
| **2.1** | Der Test überprüft, ob die durchschnittlichen Bewertungen eines Gerichts und einer Lieferung korrekt berechnet wurden. | Testen/src/main/java/AvgBewertungTest  avgTest() Zeile: 182-191 | Muss für die Endabnahme wiederholt werden. Der Meilenstein am 16.12.2021 wurde aber bestanden. |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 16.12.2021 | | |
| **Tester** | Leon | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Es wurden Restaurantbesitzer, Restaurants und Speisekarten erstellt. Der Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert und eingeloggt. Der Benutzer befindet sich auf der Kundenstartseite. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer klickt auf „Restaurant suchen“. | Der Benutzer wird auf die View der Restaurantliste weitergeleitet. | **√** |
| 2 | Der Benutzer klickt auf „Alternativadresse“. | Der Radiobutton ist ausgewählt. | **√** |
| 3 | Der Benutzer klickt auf den Button „aendern“. | Der Benutzer wird auf die View der Alternativadresse weitergeleitet. | **√** |
| 4 | Der Benutzer gibt die Attribute „Alternative Straße“, „Alternative Nummer“, „Alternative PLZ“ und „Alternative Stadt“ ein. | Die Textfelder sind gefüllt. | **√** |
| 5 | Der Benutzer klickt auf den Button „Speichern“. | Die Daten wurden gespeichert und werden an die Datenbank übermittelt. Der Benutzer wird wieder auf die View der Restaurantliste weitergeleitet. | **√** |
| **Nachbe-dingung(en)** | Die Alternativadresse wurde im System gespeichert und dem Nutzer werden die Restaurants in seiner Nähe angezeigt. | | **√** |
| **Testurteil** | Test bestanden | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 16.12.2021 | | |
| **Tester** | Leon | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Es wurden ein Restaurantbesitzer, Restaurant und eine Speisekarte erstellt. Der Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert und eingeloggt. Der Benutzer befindet sich auf der Restaurantliste. Wichtig: Der Benutzer hat Guthaben auf sein Guthabenkonto geladen. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer sucht sich ein Restaurant aus und klickt auf den Button „Bestellen“. | Der Benutzer wird auf die Speisekarte des Restaurants weitergeleitet. | **√** |
| 2 | Der Benutzer klickt in einer Zeile auf ein Gericht und klickt auf „Hinzufügen“. | Das Gericht wurde in seinen Warenkorb hinzugefügt. | **√** |
| 3 | Der Benutzer klickt auf den Button „fertig“. | Der Benutzer wird auf den Warenkorb weitergeleitet. | **√** |
| 4 | In dem Warenkorb wird die Gesamtsumme (ohne Lieferkosten und die Lieferkosten angezeigt.  Der Benutzer klickt auf den Button „bestellen“. | Die Bestellung wurde durchgeführt und der Benutzer wird auf die Bewertungsview weitergeleitet. | **√** |
| **Nachbe-dingung(en)** | Der Benutzer konnte bei einem Restaurant ein Gericht bestellen. | | **√** |
| **Testurteil** | Test bestanden | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 16.12.2021 | | |
| **Tester** | Leon | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Es wurden ein Restaurantbesitzer, Restaurant und eine Speisekarte erstellt. Der Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert und eingeloggt. Der Benutzer hat eine Bestellung ausgelöst und befindet sich in der Bewertungsview. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer wählt in der Gericht-ChoiceBox die „5“ aus. | Die Ziffer 5 wurde ausgewählt. | **√** |
| 2 | Der Benutzer wählt in der Lieferung-ChoiceBox die „3“ aus. | Die Ziffer 3 wurde ausgewählt. | **√** |
| 3 | Der Benutzer füllt das Kommentarfeld mit dem Kommentar „Essen war super“. | Das Kommentarfeld wurde gefüllt. | **√** |
| 4 | Der Benutzer gibt den Namen „Hans“ in das Namenstextfeld ein. | Das Namenstextfeld wurde gefüllt. | **√** |
| 5 | Der Benutzer klickt auf den Button „Speichern“. | Die Bewertung wird in die Datenbank geschrieben und der Benutzer wird auf die Startseite weitergeleitet. | **√** |
| 6 | Der Benutzer klickt auf den Button „Restaurants suchen“. | Der Benutzer befindet sich in der Restaurantliste. | **√** |
| 7 | Der Benutzer sieht die durchschnittliche Bewertung. Der Benutzer klickt auf „Bewertungen“. | Der Benutzer sieht seine Bewertung für das Restaurant. | **√** |
| **Nachbe-dingung(en)** | Der Benutzer konnte ein Restaurant bewerten. | | **√** |
| **Testurteil** | Test bestanden | | |

# Zyklus III

## Spezifikationsplanung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | **Aktionen** |  |  |  |
| 1.1 | * Rabatt an einem Tag * Rabatt in einem bestimmten Zeitraum (in Tagen) * Kunde hat Rabatt an Geburtstagen * Rabattaktionen werden in der Restaurantübersicht angezeigt * Kunden erhalten Mails von ihren favorisierten Restaurants | User Story | Leon Gashi | Fertig |
| 1.2 | * Rabatt an einem Tag * Rabatt in einem bestimmten Zeitraum (in Tagen) | Papierprototyp | Leon Gashi | Fertig |
| 1.3 | * Kunde hat Rabatt an Geburtstagen * Kunden erhalten Mails bei Rabattaktionen von favorisierten Restaurants | MSC | Dursun Berat Saral | Fertig |
| **2.** | **Statistik** |  |  |  |
| 2.1 | * Statistik über Restaurant einsehen können * Welches Gericht wird besonders häufig bestellt im Monat/Woche (Zeitraum muss wählbar sein) * Entwicklung der Bewertungen einsehen können * Statistiken sollen textuell und grafisch einsehbar sein | User Story | Nicklas Czorny | Fertig |
| 2.2 | * Statistik über Restaurant einsehen können * Welches Gericht wird besonders häufig bestellt im Monat/Woche (Zeitraum muss wählbar sein) * Entwicklung der Bewertungen einsehen können * Statistiken sollen textuell und grafisch einsehbar sein | Papierprototyp | Nicklas Czorny | Fertig |
| 2.3 | * Statistik über Restaurant einsehen können * Statistiken sollen textuell und grafisch einsehbar sein | MSC | Nicklas Czorny | Fertig |
| **3.** | **Sonderwünsche** |  |  |  |
| 3.1 | * Sonderwünsche für Kunden * Restaurantbesitzer kann Rechnung annehmen/ablehnen (Benachrichtigung für Kunden) * Anpassung der Rechnung | User Story | Nick Waldschmidt | Fertig |
| 3.2 | * Sonderwünsche für Kunden * Restaurantbesitzer kann Rechnung annehmen/ablehnen (Benachrichtigung für Kunden) | Papierprototyp | Nick Waldschmidt | Fertig |
| 3.3 | * Sonderwünsche für Kunden * Restaurantbesitzer kann Rechnung annehmen/ablehnen (Benachrichtigung für Kunden) * Anpassung der Rechnung | MSC | Dursun Berat Saral | Fertig |
| **4.** | **Lieferzeit anzeigen** |  |  |  |
| 4.1 | * Lieferzeit anzeigen | User Story | Oguzhan Kaluk | Fertig |
| 4.2 | * Lieferzeit anzeigen | Papierprototyp | Oguzhan Kaluk | Fertig |
| 4.3 | * Lieferzeit anzeigen | MSC | Oguzhan Kaluk | Fertig |
| **5.** | **Suche nach bestimmtem Gericht** |  |  |  |
| 5.1 | * Kunde kann bestimmtes Gericht in allen Speisekarten der Restaurants suchen * Suchfunktion für Name des Gerichts | User Story | Alexandra Klein | Fertig |
| 5.2 | * Kunde kann bestimmtes Gericht in allen Speisekarten der Restaurants suchen * Suchfunktion für Name des Gerichts | Papierprototyp | Alexandra Klein | Fertig |
| 5.3 | * Kunde kann bestimmtes Gericht in allen Speisekarten der Restaurants suchen * Suchfunktion für Name des Gerichts | MSC | Alexandra Klein | Fertig |
| **6.** | **UML-Klassendiagramm** |  |  |  |
| 6.1 | Front-End | UML-Klassendiagramm | Alexandra Klein | Fertig |
| 6.2 | Back-End | UML-Klassendiagramm | Alexandra Klein | Fertig |

## User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | 9.1 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich an einem bestimmten Tag in der Woche oder in bestimmten Zeiträumen (in Tagen) einen Rabatt von 20% festlegen können. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 9.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass der Kunde an seinem Geburtstag automatisch einen Rabatt von 20% erhält. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Niedrig |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 9.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass die Rabattaktionen in der Übersicht aller Restaurants angezeigt werden können. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Hoch |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** | 9.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 9.4 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass der Kunde Mails von Rabattaktionen von seinen favorisierten Restaurants bekommt. |
| **Geschätzter Realisierungsaufwand** | 5 Std |
| **Priorität** | Mittel |
| **Autor** | Leon Gashi |
| **Abhängigkeiten zu anderen User Stories** | 9.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 10.1 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich Statistiken zu meinem Restaurant einsehen können, damit die Effektivität von Maßnahmen besser bewerten werden können. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | Mittel |
| **Autor** | Nicklas Czorny |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 10.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich in der Statistik einsehen können, welches Gericht im Monat besonders häufig bestellt wurde, damit ich die Zutaten dafür einplanen kann. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | Mittel |
| **Autor** | Nicklas Czorny |
| **Abhängigkeiten zu anderen User Stories** | 10.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 10.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich in der Statistik einsehen können, wie sich die Bewertungen über einen Zeitraum hinweg entwickeln, damit ich erkennen kann, ob die Kundenzufriedenheit zunimmt oder abnimmt. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | Hoch |
| **Autor** | Nicklas Czorny |
| **Abhängigkeiten zu anderen User Stories** | 10.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 10.4 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich die Statistik textuell und graphisch angezeigt bekommen, damit Veränderungen verdeutlicht und wichtige Kennzahlen direkt hervorgehoben sind. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | Mittel |
| **Autor** | Nicklas Czorny |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 11.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich einen Sonderwunsch äußern können, um meine offen Wünsche zu erfüllen. |
| **Geschätzter Realisierungsaufwand** | 1 Stunde |
| **Priorität** | Mittel |
| **Autor** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 11.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer soll ich den Sonderwunsch annehmen oder ablehnen können, um einen vielleicht nicht möglichen Sonderwunsch abzulehnen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 11.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 11.3 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich den Preis ändern, wenn ich einen Sonderwunsch annehme, um die Mehrkosten zu decken. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | mittel |
| **Autor** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 11.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 11.4 |
| **User Story-Beschreibung** | Als Kunde möchte ich benachrichtig werden, wenn meine Speise angenommen oder abgelehnt wird, um zu wissen, ob mein Sonderwunsch akzeptiert wurde oder nicht. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor** | Nick Waldschmidt |
| **Abhängigkeiten zu anderen User Stories** | 11.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 12.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich die voraussichtliche Lieferzeit sehen können, damit ich abwägen kann, wann die Lieferung eintrifft. |
| **Geschätzter Realisierungsaufwand** | 6 Std. |
| **Priorität** | hoch |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 12.2 |
| **User Story-Beschreibung** | Als Restaurantbesitzer möchte ich, dass die Lieferzeit automatisch berechnet wird, je nachdem, wie viele Bestellungen eingegangen sind, um unsere Kunden möglichst zufrieden stellen zu können. |
| **Geschätzter Realisierungsaufwand** | 6 Std. |
| **Priorität** | hoch |
| **Autor** | Oguzhan Kaluk |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 13.1 |
| **User Story-Beschreibung** | Als Kunde möchte ich bestimmte Speisen suchen können, wenn ich mir unschlüssig bin. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | hoch |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 13.2 |

|  |  |
| --- | --- |
| **User Story-ID** | 13.2 |
| **User Story-Beschreibung** | Als Kunde möchte ich alle Restaurants angezeigt bekommen, die meine gesuchte Speise anbieten. |
| **Geschätzter Realisierungsaufwand** | 1 Std |
| **Priorität** | hoch |
| **Autor** | Alexandra Klein |
| **Abhängigkeiten zu anderen User Stories** | 13.1 |

## Papierprototypen

### Aktionen

Autor: Leon Gashi

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 20: Zeitraum für Rabattaktionen

### Sonderwünsche

Autor: Nick Waldschmidt

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 21: Tableview mit Sonderwünschen

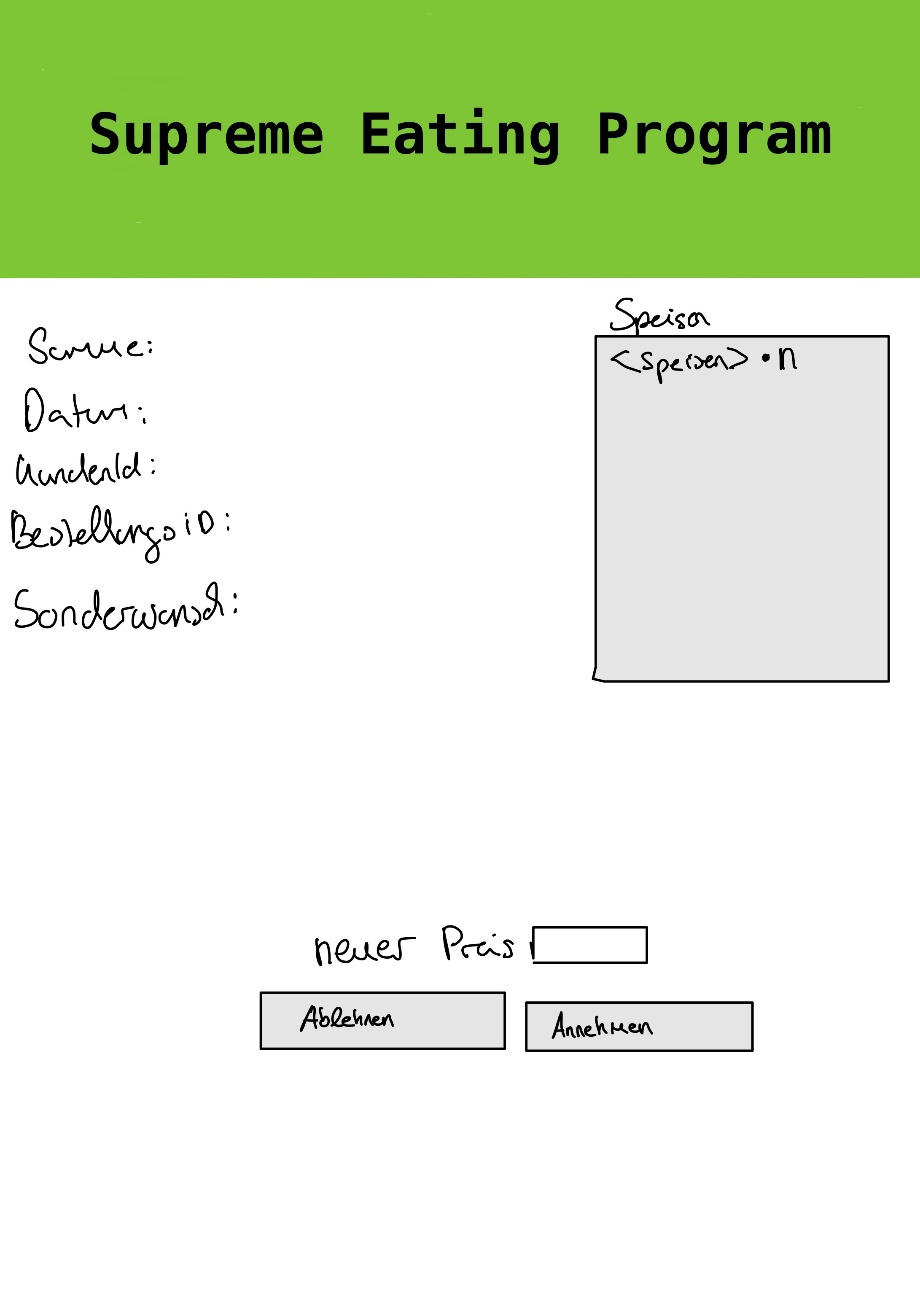


Abbildung 22: Ablehnen/Annehmen für Restaurantbesitzer

### Statistik

Autor: Nicklas Czorny



Abbildung 23: Restaurantstatistik

### Lieferzeit

Autor: Oguzhan Kaluk





Abbildung 24: Lieferzeit

### Gericht Suchen

Autor: Alexandra Klein

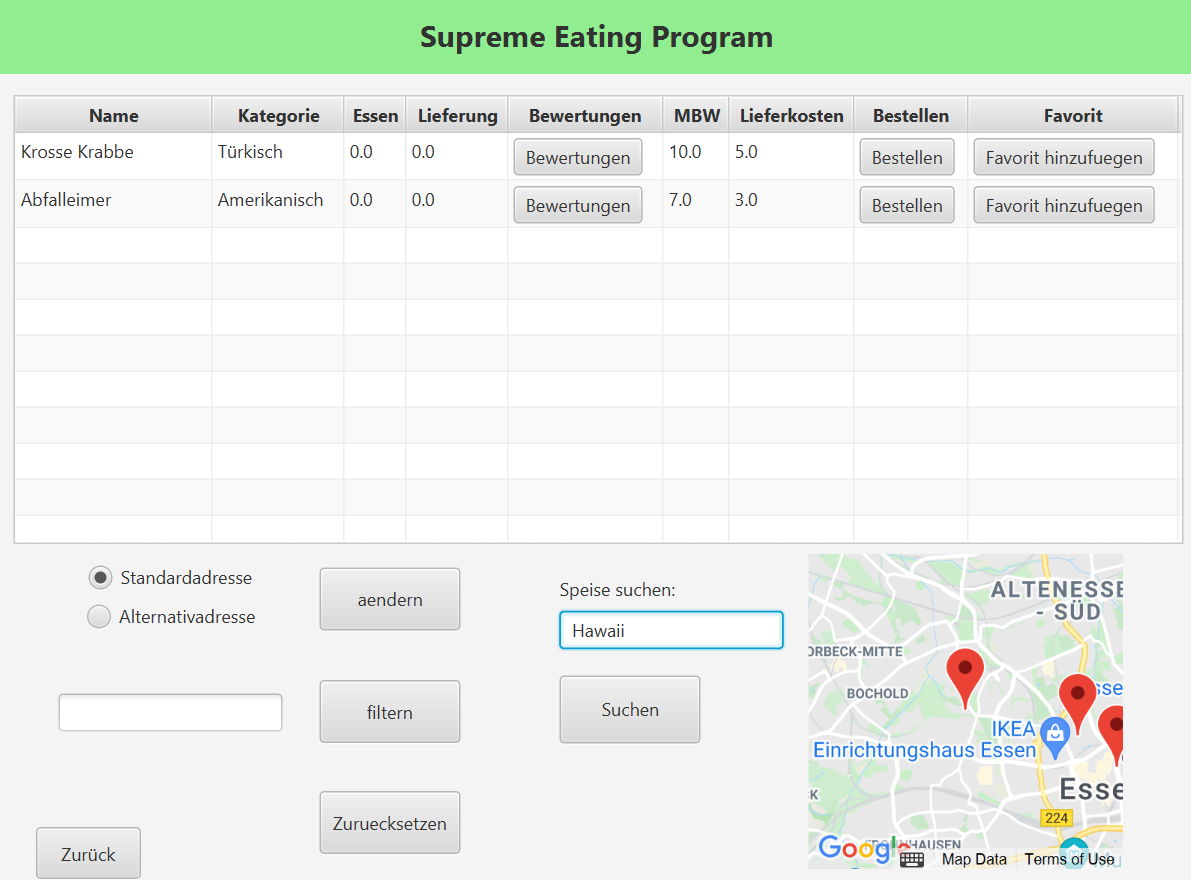
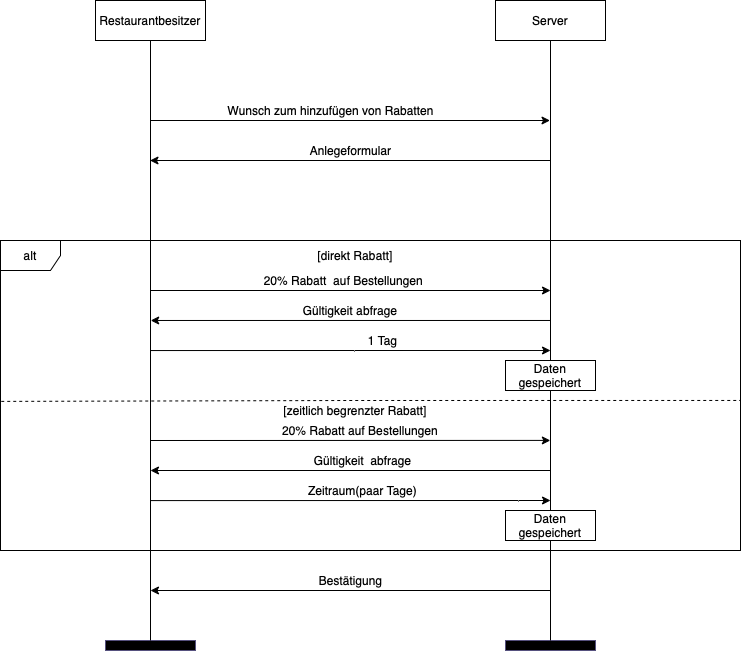


Abbildung 25: Suche nach bestimmtem Gericht

## Szenarien (MSCs)

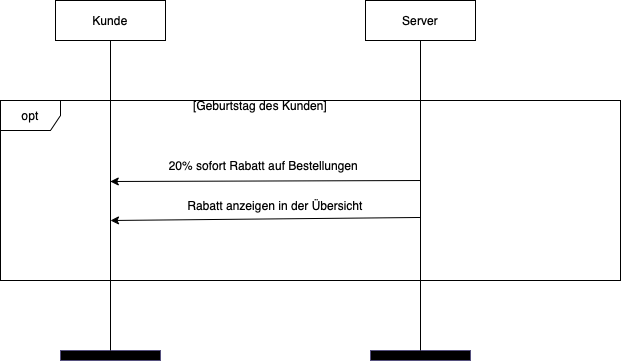
Autor: Dursun

Aktionen

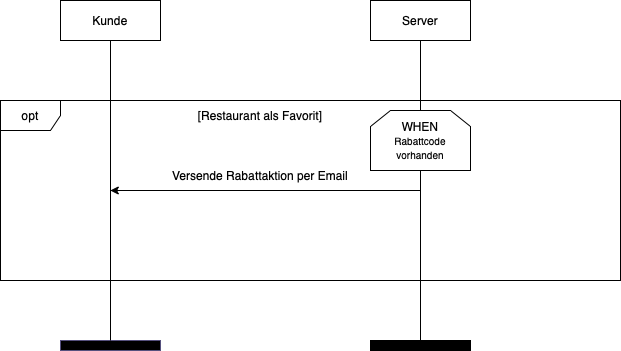


**Hauptszenario** Der Restaurantbesitzer hat den Wunsch Rabatte hinzuzufügen. Das System sendet dem Besitzer das Formular dazu den Rabatt einzugeben. Der Restaurantbesitzer entscheidet sich für einen Direktrabatt der automatisch abgezogen wird. Das System fragt an, wie lange der Rabatt gültig sein soll. Der Restaurantbesitzer gibt seinen Wunschzeitraum an. Das System speichert die Daten und sendet dem Besitzer eine Bestätigung.

**Alternativszenario:** Der Restaurantbesitzer hat den Wunsch Rabatte hinzuzufügen. Das System sendet dem Besitzer das Formular dazu den Rabatt einzugeben. Der Restaurantbesitzer entscheidet sich für einen zeitlich begrenzten Rabatt. Das System fragt an, wie lange der Rabatt gültig sein soll. Der Restaurantbesitzer gibt seinen Wunschzeitraum an. Das System speichert die Daten und sendet dem Besitzer eine Bestätigung.



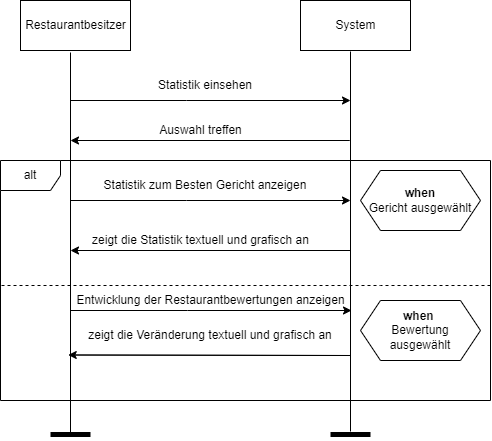
**Hauptszenario:** Der Kunde hat Geburtstag, daraufhin ruft der Server die Daten auf und schickt den sofort Rabatt an den Kunden. Außerdem zeigt der Server den Rabatt in der Übersicht für den Kunden.



**Hauptszenario:** Der Kunde hat ein Restaurant als Favorit. Wenn das geschieht und ein Rabattcode für dieses Restaurant vorhanden ist, schickt der Server diesen Rabattcode an den Kunden als E-Mail.

Autor: Nicklas Czorny

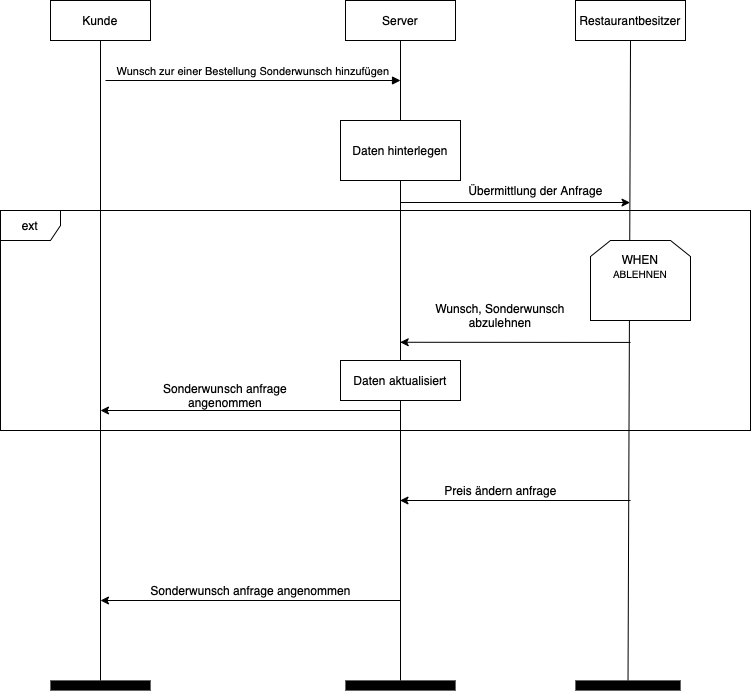
Statistik



**Hauptszenario** Der Restaurantbesitzer will die Statistik zu seinem Restaurant einsehen. Das System fordert den Restaurantbesitzer auf seine Auswahl zu treffen. Dabei kann der Restaurantbesitzer zwischen den Entwicklungen der Restaurantbewertungen und der Statistik zum besten Gericht entscheiden. Nach der Auswahl zeigt das System dem Restaurantbesitzer entweder die Veränderungen der Bewertungen über einen Zeitraum grafisch und textuell an oder die Statistik zu dem am meisten verkauften Gericht in einem bestimmten Zeitraum grafisch und textuell an.

Autor: Dursun

Sonderwünsche

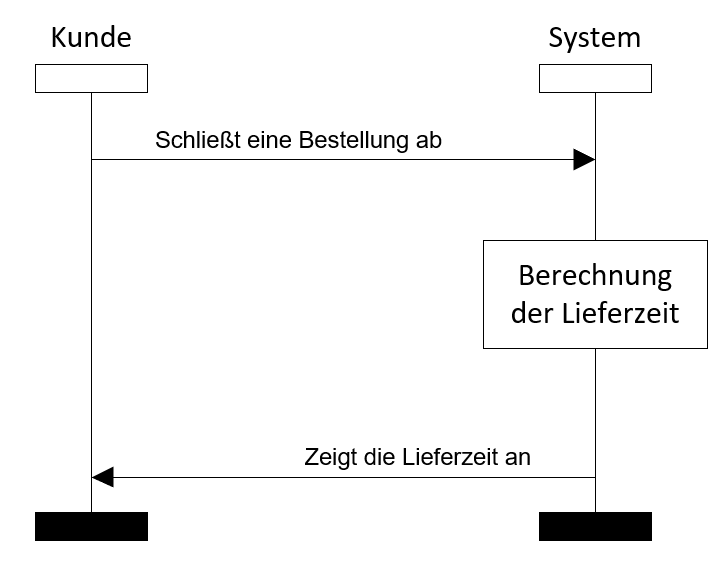


**Hauptszenario:** Der Kunde will einen Sonderwunsch äußern und die Bestellung abschließen. Das System speichert die Daten und übermittelt die Anfrage an den Restaurantbesitzer. Der Restaurantbesitzer nimmt diesen Sonderwunsch an und passt den Preis dementsprechend an. Das System aktualisiert die Daten und übermittelt den bestätigten Sonderwunsch an den Kunden.

**Ausnahmeszenario:** Wenn der Restaurantbesitzer die Anfrage ablehnt, werden die Daten im System aktualisiert und der Kunde bekommt vom System den abgelehnten Sonderwunsch als Nachricht zugeschickt.

Autor: Oguzhan Kaluk

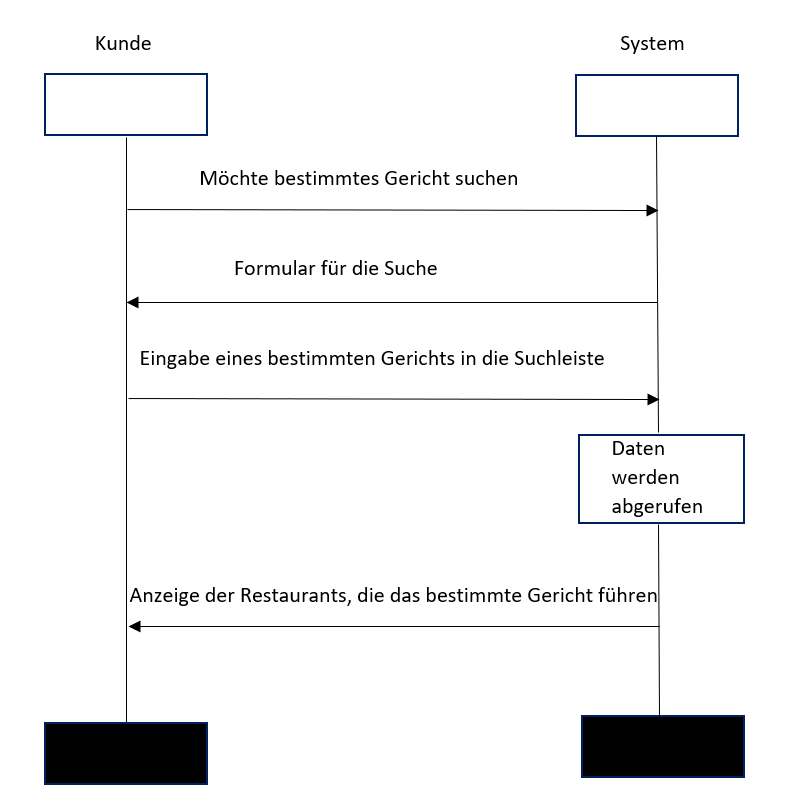
Lieferzeit:



**Hauptszenario:** Der Kunde möchte nach Abschluss seiner Bestellung die Lieferzeit sehen. Diese wird zunächst systemseitig berechnet. Nachdem dies abgeschlossen ist, wird die Lieferzeit dem Kunden angezeigt.

Autor: Alexandra Klein

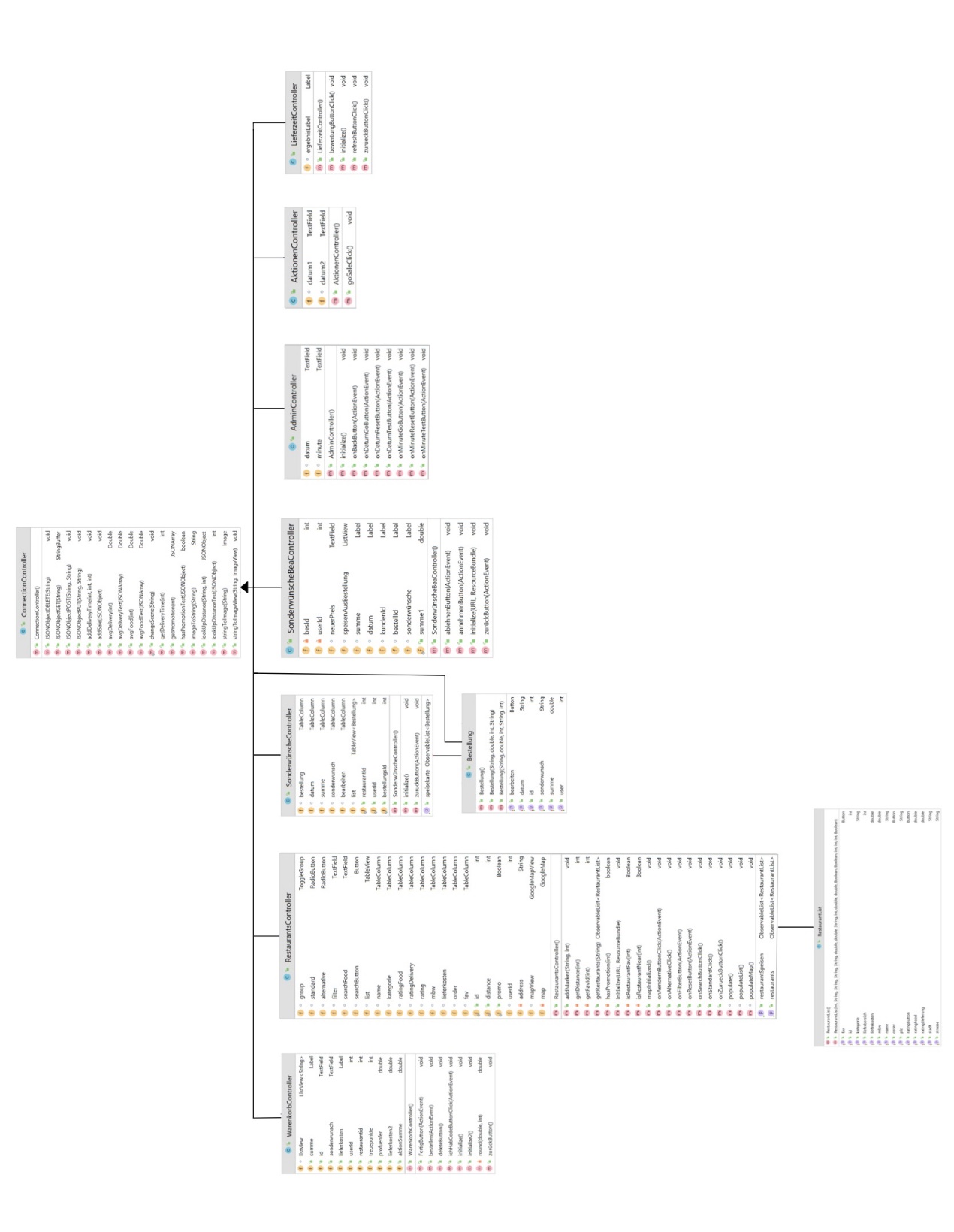
Suche nach einem bestimmten Gericht



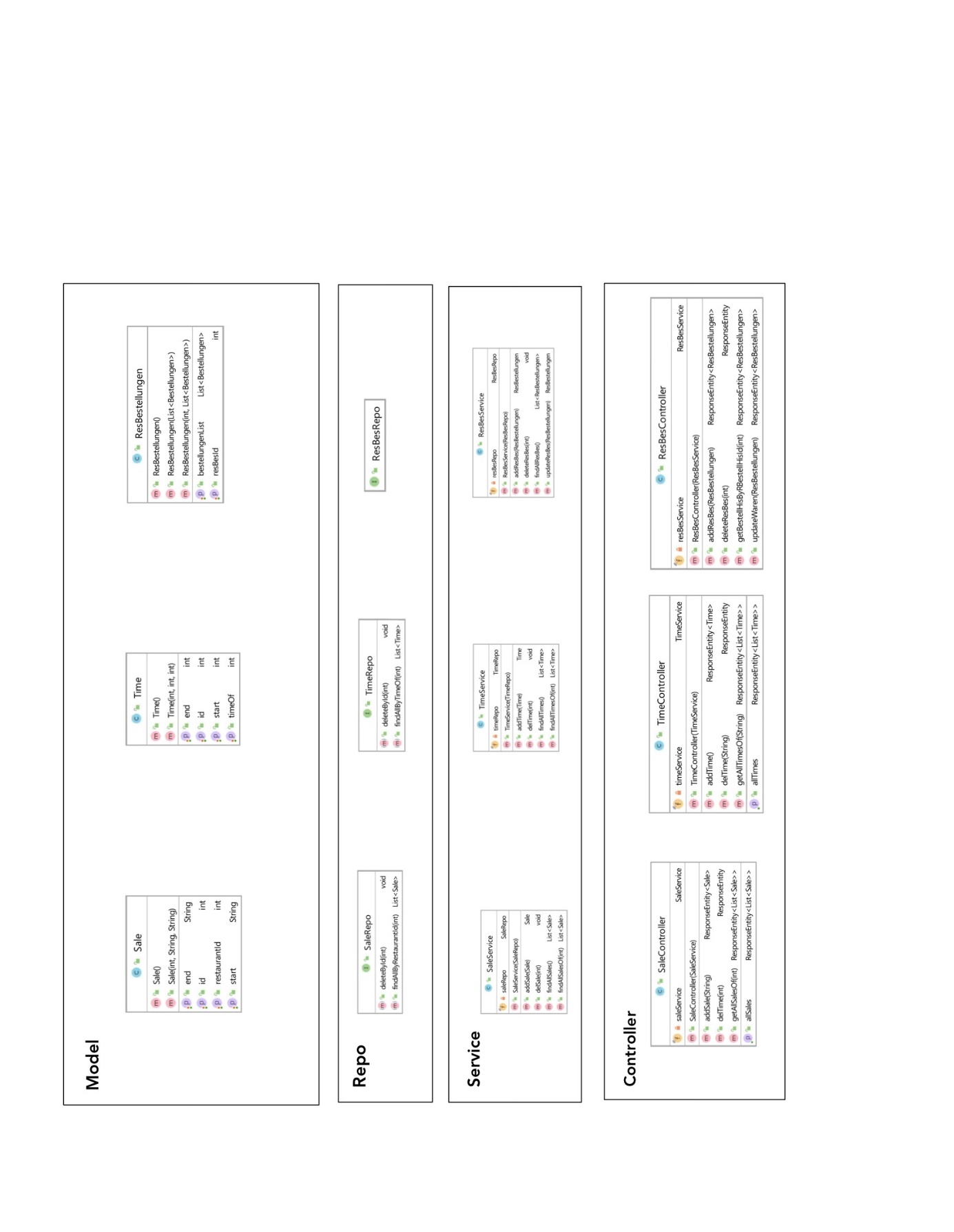
**Hauptszenario:** Der Kunde möchte die angezeigten Restaurants nach einem bestimmten Gericht durchsuchen. Dabei gibt dieser in der Suchleiste den Namen des Gerichts ein. Systemseitig werden die Daten abgerufen und die Restaurants gefiltert, die das bestimmte Gericht führen. Dem Kunden werden schließlich alle Restaurants angezeigt, die das gesuchte Gericht in ihrer Speisekarte enthalten haben.

## Strukturdiagramm (Klassendiagramm)

Autor: Alexandra Klein

Front-End:  


Back-End:



## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **13.** | **Aktionen** |  |  |  |  |  |  |
| 13.1 | Rabatt an einem Tag oder Zeitraum setzen | 6h | Leon Gashi |  | 9.1 | AktionenController | Fertig |
| 13.2 | Rabatt an Geburtstagen von Kunden automatisch setzen | 4h | Leon Gashi | 13.3 | 9.2 | RestaurantsController   * hasPromotion(int restaurantId) | Fertig |
| 13.3 | Rabattaktionen werden in der Übersicht aller Restaurants angezeigt | 6h | Leon Gashi | 13.2 | 9.3 | RestaurantsController   * hasPromotion(int restaurantId) | Fertig |
| 13.4 | Mails verschicken an Kunden, dessen Restaurants Angebote haben | 3h | Leon Gashi |  | 9.4 | <Folgt> | In Bearbeitung |
| 13.5 | Klassen in Datenbank einbinden | 1 h | Leon Gashi |  |  | Model: Sale  Repo: SaleRepo | Fertig |
| 13.6 | Service Implementation | 1 h | Leon Gashi |  |  | SaleService | Fertig |
| 13.7 | HTTP-Anbindung | 1 h | Leon Gashi |  |  | SaleController | Fertig |
| 13.8 | JUNIT-Test Rabattaktionen | 1 h | Oguzhan Kaluk |  |  | SaleTest | Fertig |
| **14.** | **Statistik** |  |  |  |  |  |  |
| 14.1 | Zeitraum der Beobachtung einstellen | 4 h | Nicklas Czorny | 14.2 | 10.3 | RestaurantbesitzerStatistikController | In Bearbeitung |
| 14.2 | Gerichte nach Häufigkeit ausgeben | 4h | Dursun Berat Saral | 14.2.1, 14.1 | 10.2 | RestaurantbesitzerStatistikController | In Bearbeitung |
| 14.2.1 | Grafische Darstellung (Balkendiagramm) | 3 h | Dursun Berat Saral | 14.2 | 10.4 | RestaurantbesitzerStatistikController | In Bearbeitung |
| 14.3 | Zeitraum für Beobachtungen (Bewertung) | 3 h | Dursun Berat Saral | 14.1, 14.4 | 10.3 | <Folgt> | In Bearbeitung |
| 14.4 | Entwicklung der Bewertungen | 3 h | Nicklas Czorny | 14.3 | 10.4 | <Folgt> | In Bearbeitung |
| 14.4.1 | Grafische Darstellung (Balkendiagramm) | 3 h | Nicklas Czorny | 14.2.1 | 10.4 | <Folgt> | In Bearbeitung |
| 14.5 | JUNIT-Test Statistik | 2 h | Dursun Berat Saral |  |  | StatistikTest | In Bearbeitung |
| **15.** | **Sonderwünsche** |  |  |  |  |  |  |
| 15.1 | Sonderwünsche hinzufügen | 4 h | Nick Waldschmidt |  | 11.1 | SonderwünscheController | Fertig |
| 15.2 | Sonderwünsche vom Restaurantbesitzer annehmen/ablehnen/bearbeiten | 4 h | Nick Waldschmidt | 15.3 | 11.2, 11.3 | SonderwünscheBeaController | Fertig |
| 15.3 | Sonderwunsch Annahme/Ablehnung als Nachricht erhalten | 2 h | Nick Waldschmidt | 15.2 | 11.4 | SonderwünscheBeaController | Fertig |
| **16.** | **Lieferzeit** |  |  |  |  |  |  |
| 16.1 | Admin Ansicht für Änderung der Zeiträume zum Testen | 4 h | Oguzhan Kaluk |  |  | AdminController | Fertig |
| 16.2 | Funktionen für Lieferzeit, Distanz, Rabattaktionen, durchschnittliche Bewertungen | 1 h | Oguzhan Kaluk |  |  | ConnectionController   * addDeliveryTime(int timeFor, int timeOf, int distance) * getDeliveryTime(int timeFor) * lookUpDistance(String address, int restaurantId) * addSale(JSONObject input) * getPromotion(int restaurantId) * avgFood(int id) * avgDelivery(int id) * avgFoodTest(JSONArray j) * avgDeliveryTest(JSONArray j) * lookUpDistanceTest(JSONObject j) * hasPromotionTest(JSONObject cur) | Fertig |
| 16.3 | Lieferzeit berechnen | 3 h | Oguzhan Kaluk |  | 12.1, 12.2 | LieferzeitController | Fertig |
| 16.4 | Integrierung der Lieferzeit | 1 h | Oguzhan Kaluk |  |  | WarenkorbController   * Zeile 305-311 | Fertig |
| 16.5 | Klassen in Datenbank einbinden | 1 h | Oguzhan Kaluk |  |  | Model: Time  Repo: TimeRepo | Fertig |
| 16.6 | Service Implementation | 1 h | Oguzhan Kaluk |  |  | TimeService | Fertig |
| 16.7 | HTTP Anbindung | 1 h | Oguzhan Kaluk |  |  | TimeController | Fertig |
| 16.8 | JUNIT-Test Lieferzeit | 2 h | Oguzhan Kaluk |  |  | TimeTest | In Bearbeitung |
| **17.** | **Nach bestimmtem Gericht suchen** |  |  |  |  |  |  |
| 17.1 | Suchfunktion für ein bestimmtes Gericht | 4 h | Alexandra Klein | 17.2 | 13.1 | RestaurantsController   * onSearchButtonClick() * getRestaurantSpeisen() | Fertig |
| 17.2 | Suchfunktion wird in der Übersicht aller Restaurants angezeigt | 3 h | Alexandra Klein | 17.1 | 13.2 | RestaurantsController   * onSearchButtonClick() * getRestaurantSpeisen() | Fertig |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **3** | Der gesetzte Zeitraum soll mit dem tatsächlichen Zeitraum übereinstimmen | Gruppe-12>testen>src>main>java>SaleTest | Bestanden |
|  |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Nutzerhandbuch

## Technische Anforderungen

Technische Mindestanforderungen, welche das Programm benötigt, um wie gewünscht bedienbar zu sein.

MySQL mit dem Schema "SEP" und als Passwort: 12345678 und Java Version 11 sind erforderlich.

Der Ordner „Executable“ ist erforderlich. Darin enthalten sind die Ordner „Client“ und „Server“.

Die client.jar muss in dem Ordner „Client“ enthalten sein, damit diese gestartet werden kann. Der Ordner „Client“ enthält die auszuführbare client.jar, den „classes“ Ordner und den „lib“ Ordner.

Die server.jar muss in dem Ordner „Server“ enthalten sein, damit diese gestartet werden kann. Der Ordner „Server“ beinhaltet die auszuführbare server.jar und den Ordner „classes“.

## Installationsanleitung

Genaue Erläuterung, wie das entwickelte Programm vollkommen funktionsfähig auf einem Rechner in Betrieb genommen werden kann.

Zuerst wird die **server.jar** ausgeführt. Danach wird die **client.jar** ausgeführt.  
Das Programm öffnet sich.

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.

Registrierung

Der Bediener registriert sich als „Ich bin Business User“ mit seinen Daten in der Registrierungs-View.

Login und 2-Faktor-Authentifizierung

Der Bediener befindet sich in der Login-View und gibt seine Daten ein. Danach wird dieser auf die View der 2-Faktor-Authentifizierung weitergeleitet. Hier wird der per E-Mail gesendete Code eingegeben. Beim Klicken auf „Anmelden“ wird auf die Home-View weitergeleitet.

Restaurant anlegen

In dieser View kann der Bediener die Restaurantdaten wie „Name“, Straße“, „Lieferkosten“, „Lieferbereich“, usw. pflegen. Mit Klick auf „Speichern“ gelangt der Bediener wieder in die Home-View.

Restaurant anlegen

In dieser View kann der Bediener die Restaurantdaten, die zuvor in der View „Restaurant anlegen“ gepflegt wurden, ändern. Mit Klick auf „Speichern“ werden die neuen Daten übernommen und es wird auf die Home-View weitergeleitet.

Map anzeigen

In dieser View kann der Bediener die Karte anzeigen. Der Standort des Restaurants, sowie der Radius des Lieferbereichs werden hier angezeigt. Durch das Scrollen kann der Bediener in die Karte rein- und rauszoomen.

Speisekarte

In dieser View kann der Bediener durch Klicken auf die Buttons „Speise Hinzufügen“, „Speise Bearbeiten“ und „xml“, diverse Aktionen im Zusammenhang mit der Speisekarte ausführen.

So können in der View „Speise Hinzufügen“ verschiedene Speisen mit diversen Kategorien angelegt werden.

Durch die SpeiseIDs können in der View „Speise Bearbeiten“ die Speisen bearbeitet oder gelöscht werden durch Klick auf „Delete“.

In der View „xml“ können Speisen durch eine xml.-Datei hochgeladen werden.

Kundenkonto anlegen

Der Bediener registriert sich als ,,Ich bin Kunde“ mit seinen Daten in der Registrierungs-View.

Login und 2-Faktor-Authentifizierung

Der Bediener befindet sich auf der Login-View und gibt seine Daten ein. Danach wird dieser auf die View der 2-Faktor-Authentifizierung weitergeleitet. Hier wird der per E-Mail gesendete Code eingegeben. Beim Klicken auf ,,Anmelden“ wird auf die KStartseite-View weitergeleitet.

Guthaben hinzufügen

Mit dem Klick auf den „Guthaben aufladen“-Button wird der Kunde auf die Guthaben -View weitergeleitet. Dort kann der Kunde mit der Eingabe der Anzahl der Guthaben mit einem Aufladen-Button Guthaben hinzufügen. Mit dem Fertig-Button und dem Zurück-Button wird auf die KStartseite-View weitergeleitet.

Auswahl von Restaurants anzeigen

Mit Klick auf die „Restaurant suchen“-Button wird der Kunde auf die Restaurants-View weitergeleitet. Dort hat der Kunde die Möglichkeit ein bestimmtes Restaurant in der Liste zu bewerten, zu Favoriten hinzuzufügen und zu Bestellen. Außerdem kann der Kunde sich dafür entscheiden auf den Standardadresse-Button oder Alternativadresse-Button zu klicken und bei der Alternativadresse, diese zu ändern. Beim Klick auf den „aendern“-Button wird der Kunde zur „EditAltProfile“-View weitergeleitet und kann dort seine Alternative Adresse mit einem Klick auf den „Speichern“-Button sichern oder mit einem Klick auf den „Zurück“-Button zurück zur Restaurants-View gehen. Mit Klick auf den „Bestellen“-Button in der Restaurantliste wird man auf die Menu-View weitergeleitet. Mit dem Klick auf den „Favoriten“-Button wird das Restaurant als Favorit hinzugefügt. Mit Klick auf den „Bewertungen“-Button wird auf die Bewertung-View des Restaurants weitergeleitet.

Essensbestellungen

In der View ,,Menu“ kann der Kunde sich die Produkte des Restaurants ansehen und mit einem Klick auf „hinzufügen“-Button die gewünschten Gerichte zum Warenkorb hinzufügen. Durch Klicken auf den „Fertig“-Button wird man zu der Warenkorb-View weitergeleitet. Dort hat man die Möglichkeit noch Gerichte aus dem Warenkorb zu entfernen mit dem Klick auf „Aus Warenkorb entfernen“. Abschließend kann man die Bestellung abschließen und hat die Möglichkeit eine Bewertung hinzuzufügen, dabei wird der Kunde automatisch auf die Bewertung-View weitergeleitet und kann dort eine Bewertung zu seiner Bestellung hinzufügen.

Bewertung

In der Bewertung-View hat der Kunde die Möglichkeit in den Kategorien Essen und Lieferung jeweils eine Bewertung von 1-5 Punkten abzugeben. Außerdem hat der Kunde die Möglichkeit einen Kommentar zu verfassen und seinen Namen einzugeben. Mit Klick auf den „Zurück“-Button, kann der Kunde die Bewertung abbrechen und mit dem Klick auf den „Speichern“-Button kann der Kunde die Bewertung speichern.

Treuepunkte

Der Kunde bekommt pro 5 Euro Bestellwert einen Treuepunkt. Ab 10 Treuepunkt bekommt der Nutzer einen Coupon an seine E-Mail gesendet und hat die Möglichkeit den Code bei seinem Warenkorb auf „Ich habe einen Rabattcode !“ zu klicken und wird auf die Treuepunkte-View weitergeleitet. Dort hat der Kunde die Möglichkeit den Rabattcode einzugeben und mit einem Klick auf den „Weiter“-Button auf die Warenkorb-View zurückgeleitet zu werden.

Bestellhistorie anzeigen

Alle abgeschlossenen Bestellungen können in der bestellHistorie-View vom Kunden angesehen werden. Mit Klick auf den „Zurück“-Button kann der Kunde zur KStartseite-View weitergeleitet werden.