SEP Projektmappe

**Projektmappe des Projektes**

Supreme Eating Program

**Dokumentation des Projektes**

Gruppe 12:

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 4](#_Toc85022905)

[Zyklus I 5](#_Toc85022906)

[Spezifikationsplanung 5](#_Toc85022907)

[User-Stories 5](#_Toc85022908)

[Papierprototypen 6](#_Toc85022909)

[Szenarien (MSCs) 6](#_Toc85022910)

[Strukturdiagramm (Klassendiagramm) 7](#_Toc85022911)

[Funktionalitätsplanung 7](#_Toc85022912)

[Systemtests 7](#_Toc85022913)

[Zyklus II 10](#_Toc85022914)

[Spezifikationsplanung 10](#_Toc85022915)

[User-Stories 10](#_Toc85022916)

[Papierprototypen 10](#_Toc85022917)

[Szenarien (MSCs) 10](#_Toc85022918)

[Strukturdiagramm (Klassendiagramm) 10](#_Toc85022919)

[Funktionalitätsplanung 10](#_Toc85022920)

[Unittests 11](#_Toc85022921)

[Systemtests 11](#_Toc85022922)

[Zyklus III 14](#_Toc85022923)

[Spezifikationsplanung 14](#_Toc85022924)

[User-Stories 14](#_Toc85022925)

[Papierprototypen 14](#_Toc85022926)

[Szenarien (MSCs) 14](#_Toc85022927)

[Strukturdiagramm (Klassendiagramm) 14](#_Toc85022928)

[Funktionalitätsplanung 15](#_Toc85022929)

[Unittests 15](#_Toc85022930)

[Systemtests 15](#_Toc85022931)

[Nutzerhandbuch 16](#_Toc85022932)

[Technische Anforderungen 16](#_Toc85022933)

[Installationsanleitung 16](#_Toc85022934)

[Bedienungsanleitung 16](#_Toc85022935)

# 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

Login-Seite:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Registrierungs-Seite:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

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

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

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



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



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



## 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.

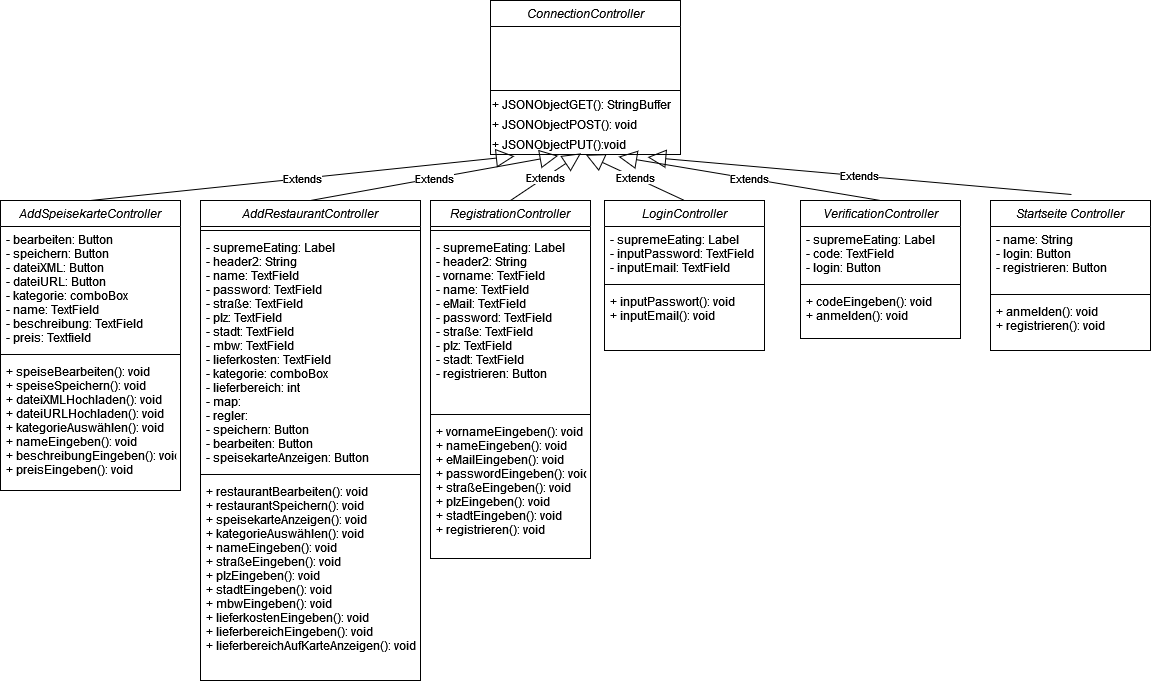
**Alternativszenario:** Alternativ ist das Löschen einzelner Einträge möglich. Dazu wird der zu löschende Eintrag ausgewählt und bestätigt. Die Datenbank wird aktualisiert und der Restaurantbesitzer bekommt eine Bestätigung.

## 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 Tisch 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 |
| 4.6 | Alle Daten 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 | AddSpeisekarteController. | fertig |
| 5.2.1 | Kategorien anlegen | 1 Tag | Dursun Berat Saral | 5.6 | 3.2 | AddSpeisekarteController | fertig |
| 5.3 | Speisen Hinzufügen | 2 Tage | Dursun Berat Saral | - | 3.4 | AddSpeisekarteController | fertig |
| 5.4 | Speisen Bearbeiten | 2 Tage | Dursun Berat Saral | - | 3.5 | 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 |  |  |  | fertig |
| 7.2.1 | XML-Speicherung in Datenbank | 1 Tag | Nick Waldschmidt |  | 3.3 |  | fertig |
| 7.3 | HTTP Anbindung | 2 std | Oguzhan Kaluk | - | - | FodController  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  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. | | |

# Zyklus II

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |

Papierprototypen

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** |  |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |

## 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** |
| **…** |  |  |  |

## Systemtests

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

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

# Zyklus III

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** |  |

Papierprototypen

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Login** |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## 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.

## Installationsanleitung

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

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.