Gut Oriol

oriol.gut@noser.com

Exposee

Eine einfache Web Applikation die es einem oder mehrere Restaurants ermöglich ihre Menüs, Gerichte und Getränke digital zu erfassen. Ausserdem bietet es die Möglichkeit Tageskarten nach einem spezifischen Menü auszuwählen.

Dokumentation Restaurant WebAPP

Oriol Gut AP15b

Inhalt

[1. Einführung 2](#_Toc498688786)

[2. Technologie 2](#_Toc498688787)

[3. Use Cases 3](#_Toc498688788)

[3.1. Akteur 3](#_Toc498688789)

[3.2. UseCase Diagram 3](#_Toc498688790)

[3.3. Gericht abspeichern 3](#_Toc498688791)

[3.4. Getränk abspeichern 5](#_Toc498688792)

[3.5. Tageskarte erstellen 6](#_Toc498688793)

[4. Testprotokoll 6](#_Toc498688794)

[4.1. Menü erstellen 6](#_Toc498688795)

[4.2. Gericht abspeichern 6](#_Toc498688796)

[4.3. Getränk abspeichern 7](#_Toc498688797)

[4.4. Tageskarte erstellen 7](#_Toc498688798)

[5. ERD 7](#_Toc498688799)

[6. Abbildungsverzeichnis 8](#_Toc498688800)

1. Einführung

Meine Idee ist es eine ASP.Net Webapplikation für Restaurants zu erstellen. Was braucht ein Restaurant alles? Ein Restaurant hat meistens eine Menükarte in welcher das Gericht mit einer kleinen Beschreibung und einen Preis hinterlegt wird. Zusätzlich kann man Getränke in verschiedenen Kategorien eintragen, wie zum Beispiel alkoholfreie heisse oder kalte Getränke oder Cocktails. Der Aspekt der das Programm jedoch ausmacht ist die Tageskarte. Die Tageskarte wird abgespeichert mit einem Zeitstempel der das Ablaufdatum der Tageskarte abgespeichert wird. Läuft dieser ab wird die Tageskarte archiviert jedoch nicht ganz gelöscht. Schlussendlich kann also ein Restaurant alles abspeichern und auch eine Liste ihrer Gerichte und Getränke anzeigen lassen. Als Erweiterungsmöglichkeit sehe ich eine Funktion die ein Word Dokument generiert sodass man das gespeicherte Menü auch direkt ausdrucken kann.

1. Technologie

Das Projekt wurde in der .Net Umgebung entwickelt mit ASP.Net. Zusätzlich wurde das MVC Pattern angewendet und als Unterstützung Data Transfer Object benutzt. Als Datenbanksystem verwende ich SQL Server 2016 (MSSQL). Für den Data Abstraction Layer wird Entity Framework als OR Mapper[[1]](#footnote-1) benutzt.

1. Use Cases
   1. Akteur

Dieses Programm beinhaltet nicht verschiede Akteure. Die einzige Voraussetzung ist das man eingeloggt ist. Daher ist der Akteur ein normaler Benutzer.

* 1. UseCase Diagram

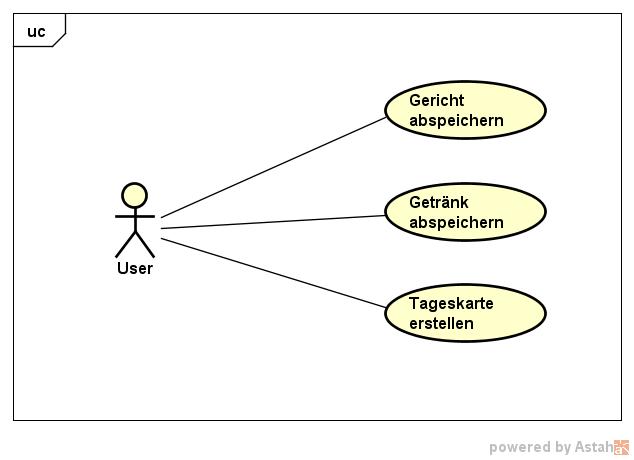


Abbildung Use Case Diagram

|  |  |
| --- | --- |
| * 1. Gericht abspeichern | |
| Motivation / Ziel | Gericht in Datenbank abspeichern |
| Vorbedingung | Eingeloggt |
| Nachbedingung bei Erfolg | Bestätigung das Gericht erfasst und gespeichert wurde |
| Nachbedingungen bei Fehlschlag | Fehlermeldung da nicht alle Pflichtfelder ausgefüllt sind. |
| Akteur | Benutzer |
| Auslösendes Ereignis | Erfolgreiches Einloggen |
| Strukturierte Beschreibung des Standardfalls | 1. Pflichtfelder für Gerichtserfassung ausgefüllt 2. Speichern gedrückt |
| Strukturierte Beschreibung der alternativen Abläufe | 1. Pflichtfelder für Gerichtserfassung ausgefüllt 2. Optionale Felder ausgefüllt 3. Speichern gedrückt |
| Strukturierte Beschreibung der Fehlerfälle | Fehlerfall 1: Name leer   * 1. Name leer gelassen   2. Speichern gedrückt   3. Fehlermeldung: für Pflichtfelder.   Fehlerfall 2: Preis leer   1. Preis leer gelassen 2. Speichern gedrückt 3. Fehlermeldung: für Pflichtfelder.   Fehlerfall 3: Menu Zugehörigkeit leer   1. Menu Zugehörigkeit leer gelassen 2. Speichern gedrückt 3. Fehlermeldung: für Pflichtfelder. |

|  |  |
| --- | --- |
| * 1. Getränk abspeichern | |
| Motivation / Ziel | Getränk in Datenbank abspeichern |
| Vorbedingung | Eingeloggt |
| Nachbedingung bei Erfolg | Bestätigung das Getränk erfasst und gespeichert wurde |
| Nachbedingungen bei Fehlschlag | Fehlermeldung da nicht alle Pflichtfelder ausgefüllt sind. |
| Akteur | Benutzer |
| Auslösendes Ereignis | Erfolgreiches Einloggen |
| Strukturierte Beschreibung des Standardfalls | 1. Pflichtfelder für Getränkeerfassung ausgefüllt 2. Speichern gedrückt |
| Strukturierte Beschreibung der alternativen Abläufe | 1. Pflichtfelder für Getränkeerfassung ausgefüllt 2. Optionale Felder ausgefüllt 3. Speichern gedrückt |
| Strukturierte Beschreibung der Fehlerfälle | Fehlerfall 1: Name leer   1. Name leer gelassen 2. Speichern gedrückt 3. Fehlermeldung: für Pflichtfelder.   Fehlerfall 2: Preis leer   1. Preis leer gelassen 2. Speichern gedrückt 3. Fehlermeldung: für Pflichtfelder.   Fehlerfall 3: Menu Zugehörigkeit leer   1. Menu Zugehörigkeit leer gelassen 2. Speichern gedrückt 3. Fehlermeldung: für Pflichtfelder. |

|  |  |
| --- | --- |
| * 1. Tageskarte erstellen | |
| Motivation / Ziel | Tageskarte erstellen |
| Vorbedingung | Eingeloggt |
| Nachbedingung bei Erfolg | Bestätigung das Tageskarte erfasst und gespeichert wurde |
| Nachbedingungen bei Fehlschlag | Fehlermeldung da keine Getränke und Gerichte ausgewählt wurden |
| Akteur | Benutzer |
| Auslösendes Ereignis | Erfolgreiches Einloggen |
| Strukturierte Beschreibung des Standardfalls | 1. Menü auswählen 2. Felder ausgewählt 3. Speichern gedrückt |
| Strukturierte Beschreibung der alternativen Abläufe |  |
| Strukturierte Beschreibung der Fehlerfälle | Fehlerfall 1: Keine Felder ausgewählt   1. Keine Felder ausgewählt 2. Speichern gedrückt 3. Fehlermeldung: |

1. Testprotokoll
   1. Menü erstellen

|  |  |
| --- | --- |
| Vorbedingung |  |
| Nachbedingung bei Erfolg | Übersicht aller Menüs |
| Nachbedingungen bei Fehlschlag | Fehlermeldung des Menüname-Felds das nicht oder falsch ausgefüllt worden ist |
| Akteur | Benutzer |
| Auslösendes Ereignis | „Create new“ auf Menü Controller (Index Methode) geklickt |
| Getestet | 17.11.2017 |

* 1. Gericht abspeichern

|  |  |
| --- | --- |
| Vorbedingung | Menü erstellt |
| Nachbedingung bei Erfolg | Übersicht aller Gerichte |
| Nachbedingungen bei Fehlschlag | Fehlermeldung der Felder die nicht oder falsch ausgefüllt worden sind |
| Akteur | Benutzer |
| Auslösendes Ereignis | „Create new“ auf Gericht Controller (Index Methode) geklickt |
| Getestet | 17.11.2017 |

* 1. Getränk abspeichern

|  |  |
| --- | --- |
| Vorbedingung | Menü erstellt |
| Nachbedingung bei Erfolg | Übersicht aller Getränke |
| Nachbedingungen bei Fehlschlag | Fehlermeldung der Felder die nicht oder falsch ausgefüllt worden sind |
| Akteur | Benutzer |
| Auslösendes Ereignis | „Create new“ auf Getränk Controller (Index Methode) geklickt |
| Getestet | 17.11.2017 |

* 1. Tageskarte erstellen

|  |  |
| --- | --- |
| Vorbedingung | Menü erstellt |
| Nachbedingung bei Erfolg | Übersicht aller Tageskarten |
| Nachbedingungen bei Fehlschlag | Fehlermeldung der Felder die nicht oder falsch ausgefüllt worden sind |
| Akteur | Benutzer |
| Auslösendes Ereignis | „Create new“ auf Tageskarten Controller (Index Methode) geklickt |
| Getestet | 17.11.2017 |

1. ERD

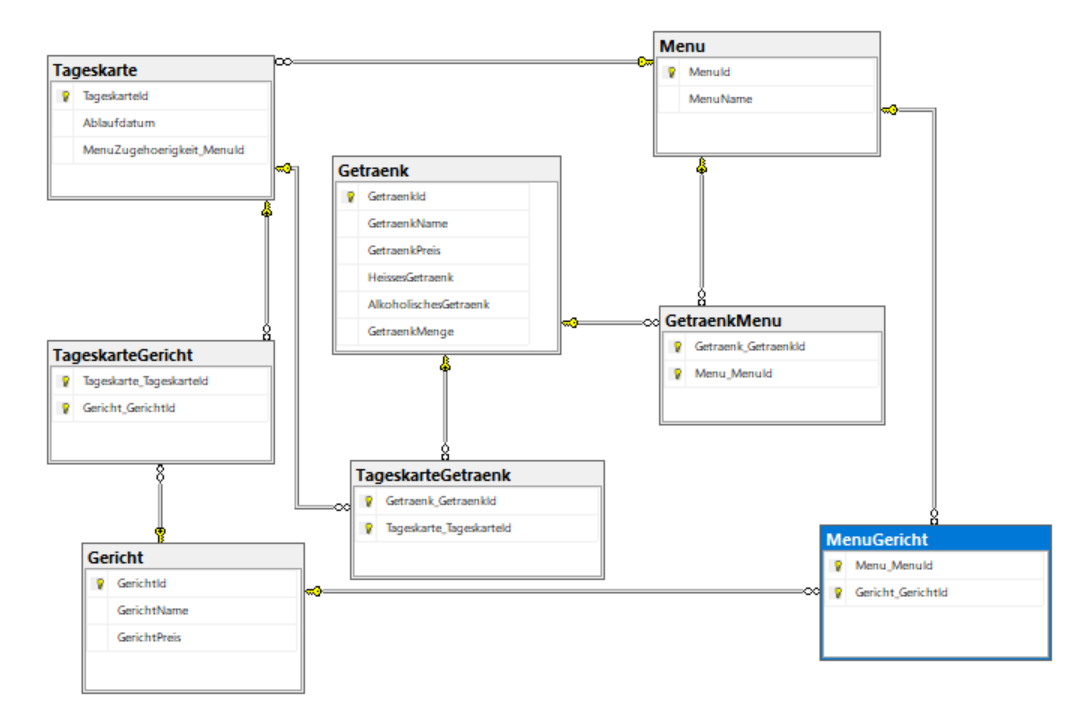


Abbildung Datenbank ERD

1. Abbildungsverzeichnis

[Abbildung 1 Use Case Diagram 2](#_Toc498688521)

[Abbildung 2 Datenbank ERD 6](#_Toc498688522)

1. Object Relation Mapper [↑](#footnote-ref-1)