FoodMood

Anforderungsanalyse

|  |  |
| --- | --- |
| **Autor(en):** | Donato Wolfisberg, Jonas Koller |
| **Version:** | 1.0 |
| **Erstellt am:** | 28. November 2016 |
| **Letzte Änderung:** | 28. November 2016 |

Inhalt

[1 Revisionen 3](#_Toc467835242)

[2 Einleitung 3](#_Toc467835243)

[2.1 Systemidee 3](#_Toc467835244)

[2.2 Management Summary 3](#_Toc467835245)

[2.3 Projektteam 3](#_Toc467835246)

[2.4 Glossar 3](#_Toc467835247)

[3 Ausgangslage (IST), Problembereiche 3](#_Toc467835248)

[3.1 Beschreibung der Ausgangslage 3](#_Toc467835249)

[3.2 Problembereiche und Schwachstellen 3](#_Toc467835250)

[4 Ziele (SOLL) 3](#_Toc467835251)

[4.1 Beschreibung der Ziele 3](#_Toc467835252)

[4.2 Produktperspektive, Nutzen 3](#_Toc467835253)

[4.3 Zielkonflikte 3](#_Toc467835254)

[4.4 Abgrenzung 3](#_Toc467835255)

[5 Anforderungsanalyse 3](#_Toc467835256)

[5.1 Identifizierung der Akteure 3](#_Toc467835257)

[5.1.1 Kontextdiagramm 3](#_Toc467835258)

[5.2 Anforderungskatalog 3](#_Toc467835259)

[5.2.1 F.REQ: Funktionale Anforderungen 3](#_Toc467835260)

[5.2.2 NF.REQ: Nichtfunktionale Anforderungen 3](#_Toc467835261)

[6 Logisches Datenmodell 3](#_Toc467835262)

[6.1 Beschreibung der Entitäten 3](#_Toc467835263)

[6.1.1 Offene Fragen 3](#_Toc467835264)

[7 Systemablaufmodelle (Aktivitäten) 3](#_Toc467835265)

[7.1 Aktivitätsdiagramm "DVD ausleihen" 3](#_Toc467835266)

[7.1.1 Detaillierte Beschreibung der Aktivitäten 3](#_Toc467835267)

[7.2 Aktivität «XY» 3](#_Toc467835268)

[8 Systemschnittstellen 3](#_Toc467835269)

[8.1 Identifikation der Schnittstellen 3](#_Toc467835270)

[8.1.1 Schnittstellenbeschreibung "DVD ausleihen" 3](#_Toc467835271)

[8.2 Beschreibung der Schnittstellenelemente 3](#_Toc467835272)

[9 Risiko-Analyse 3](#_Toc467835273)

[9.1 Risikokatalog 3](#_Toc467835274)

[10 Anhang 3](#_Toc467835275)

[10.1 Termine 3](#_Toc467835276)

[10.2 Referenzen 3](#_Toc467835277)

# Revisionen

|  |  |  |
| --- | --- | --- |
| **Datum** | **Name** | **Kommentar** |
| 28.11.2016 | Jonas Koller | Dokument erstellt. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Einleitung

## Systemidee

* Menus eintragen (Name, Zutaten, Aufwand, Foto)
* Menüplan generieren für die kommende Woche
* Menüplan für kommende Woche anzeigen
* Rechnungen für Essen verwalten
* Einkaufsliste für Tag/Woche
* Gruppen erstellen
* Benutzer in Gruppen hinzufügen
* Anpassbares Layout (Farben)

## Management Summary

Mit diesem Projekt möchten wir eine Internetseite erstellen, auf welcher es möglich ist Menüs einzutragen und mit diesen einen Menüplan für die kommende Woche zu erstellen. Es sollte Gruppen geben mit je einem Gruppenleiter. Dieser kann neue Nutzer hinzufügen ein Grundeinstellungen vornehmen. In diesen Gruppen können das Menüs eingetragen werden und mit diesen wird ein Wochenplan erstellt. Dazu gibt es auch noch eine passende Einkaufsliste. Die Rechnungen für das Essen können auch direkt Online verwaltet werden.

## Projektteam

|  |  |  |
| --- | --- | --- |
| **Name** | **Funktion** | **Kontakt (Telefon, Mail...)** |
| Urs Nussbaumer | Auftraggeber | 041 371 24 28 urs.nussbaumer@ict-bz.ch |
| Donato Wolfisberg | Projektleiter | +41 77 497 01 04 |
| Jonas Koller | Projektleiter | +41 79 758 08 41 |

# Ausgangslage (IST), Problembereiche

In diesem Kapitel wird die Ausgangslage beschrieben und es werden die Problembereiche identifiziert, die sich in der heutigen Situation zeigen.

## Beschreibung der Ausgangslage

Momentan stellen wir unseren Menüplan von Hand zusammen und müssen dafür Zeit aufwenden. Es gibt immer wieder Unklarheiten, was es jetzt zu Essen gibt. Die Rechnungen können wir nicht online verwalten, sondern müssen sie immer in ein Excel Dokument eintragen und so eine Rechnung erstellen.

## Problembereiche und Schwachstellen

Mit der jetzigen Lösung geht viel Zeit verloren und die Abrechnung funktioniert nur mässig, da wir immer wieder alle Kassenzettel zusammensuchen müssen.

# Ziele (SOLL)

In diesem Kapitel werden die übergeordneten Ziele beschrieben, die mit dem zu entwickelnden System erreicht werden sollen.

## Beschreibung der Ziele

Ziel ist es, ein Online-Programm zu erstellen welches und die Arbeit abnimmt und für Übersicht bei den Rechnungen sorgt. Es soll schnell Arbeiten, ein ansprechendes Layout haben und leicht zu verstehen sein.

## Produktperspektive, Nutzen

Wenn dieses Projekt erfolgreich beendet wird, hat es den Nutzen das viel Zeit eingespart werden kann und eine übersichtliche Verwaltung der Rechnungen möglich ist.

# Anforderungsanalyse

Die Anforderungen an das zu entwickelnde System definieren alle zu erfüllenden Eigenschaften oder die zu erbringende Leistung, sowie allfällige technische Vorgaben und weitere Forderungen des Kunden im Zusammenhang mit dem (den) zu erstellenden Produkt(en).

## Identifizierung der Akteure

Ein Akteur bezeichnet eine Rolle, die mit dem System interagiert. Es wird zwischen 4 Akteuren unterschieden:

* **Administrator:** Administrator; Maximale Rechte, er darf alles. Verwaltet das System.
* **Group Owner:** Hat die vollen Berechtigungen in einer spezifischen Gruppe, verwaltet diese, kann neue Benutzer zur Gruppe hinzufügen.
* **User:** Ist in einer Gruppe, kann Menüs hinzufügen und Rechnungen erstellen, Zugriff auf Daten der Gruppe in welcher er ist.
* **Guest:** Kann die Landing Page sehen, kann nicht interagieren, keine Rechte.

## Anforderungskatalog

Der Anforderungskatalog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein soll. Es wird unterschieden zwischen funktionalen und nicht-funktionalen Anforderungen.

Jede Anforderung wird in Alltagssprache, in der Form einer "User-Story", formuliert, mit einer ID zur späteren Identifizierung versehen sowie mittels 3 verschiedener Kategorien priorisiert: 1 = hohe Prorität, 2 = mittlere Priorität, 3 = keine Priorität. Diese drei Prioritäten repräsentieren die Verpflichtungen "must", "should" und "nice to have".

Im Verlauf der Zeit können neue Anforderungen hinzukommen und/oder bestehende Anforderungen können wegfallen. Anforderungen, die wegfallen, sind im Dokument zu belassen und als gestrichen zu markieren.

Hinter einer Anforderung kann ihn eckigen Klammern in der Form [Z#] eine Zusatzinformation hinterlegt werden und mit [F#] bzw. [F#, F#, …] können eine oder mehrere offene Fragen referenziert werden.

### F.REQ: Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| F.REQ.001 | Als Verkäufer möchte ich neue DVDs erfassen können, damit diese von den Kunden im Shop ausgeliehen werden können. [F1] | 1 |
| F.REQ.002 | Als Kunde möchte ich DVDs ausleihen können. [Z1] | 1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### Offene Fragen

[F1] Was genau bla….?

#### Zusatzinformationen

[Z1] Blub.

### NF.REQ: Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| NF.REQ.001 | Jeder User-Input muss auf Schädlichkeit geprüft werden (sodass ein Einschleusen von JavaScript-Code, SQL-Injection, u.ä. nicht möglich ist) | 1 |
| NF.REQ.002 | Der Quellcode muss gut dokumentiert sein. |  |
| NF.REQ.003 | Bla… |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Anhang

## Referenzen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** | **Titel/Beschreibung** | **Autor** | **Dokument oder URL** |
| 1 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |