

TAU INF202 Software Engineering
Individuelles Projekt
Pflichtenheft

Projektdokumentation

Version: 2023.v1.0

Status: Entwurf

THE MENU

Verantwortliche/r:

Batuhan Kılaç, 190503022

Alp Eren Ünal, 190503023

Stakeholder: DI. Ömer Karacan, omer.karacan@tau.edu.tr

Dokumentenverwaltung

Dokument-Historie

| Version | Status *) | Datum | Verantwortlicher | Änderungsgrund |
|---------|-----------|------------|------------------|---|
| v0.2 | Entwurf | 02.04.2023 | Ö. Karacan | Vorlage wurde für die Studentenprojekte freigegeben |
| v1.0 | Entwurf | 16.04.2023 | Ö. Karacan | Vorlage wurde für die Studentenprojekte freigegeben |
| | | | | |

**) Sofern im Projekt nicht anders vereinbart, sind folgende Statusbezeichnungen zu verwenden (in obiger Tabelle und am Deckblatt):*

Dokument-Status: Entwurf / in Review / freigegeben (abgegeben)

Dokument wurde mit folgenden Tools erstellt:

Microsoft Office Word

Google Documents

Inhaltsverzeichnis

| | |
|--|-----------|
| 1. Einleitung | 4 |
| 2. Ausgangssituation und Ziele | 5 |
| 3. Gesamtarchitektur | 6 |
| 4. Funktionale Anforderungen | 7 |
| 5. Nichtfunktionale Anforderungen | 10 |
| 6. Abnahmekriterien | 11 |
| 7. Projekt Meilensteine | 12 |

1. Einleitung

Dieses Dokument dient dazu, die verbindlichen Anforderungen an die Digitalisierung einer Restaurantkarte zu definieren und diese vollständig und einheitlich zu erläutern.

Die Use Cases und Anforderungen sind aus der Sicht des Stakeholders beschrieben.

Die graphische Oberfläche zur Überwachung des Menüs ist aus der Sicht einer Kunde von Restaurant beschrieben.

Kapitel 2 „Ausgangslage und Ziele“ zeigt anschaulich die Ausgangslage und den Grund für die Wahl des digitalen Menüs.

Kapitel 3 „Gesamtarchitektur“ enthält Informationen über die verschiedenen Komponenten des Systems und wie sie miteinander interagieren.

Kapitel 4 „Funktionale Anforderungen“ beinhaltet die Definition funktionaler Anforderungen durch Prozessbeschreibungen (User Stories), Use Cases (Use Cases) sowie technische und fachliche Anforderungen (Requirements). Alle betriebsrelevanten Daten werden durch Datenmodelle definiert.

Kapitel 5 „Nichtfunktionale Anforderungen“ beinhaltet die funktionalen Anforderungen durch diejenigen Anforderungen erweitert, die keine funktionalen Anforderungen sind.

Kapitel 6 „Abnahmekriterien“ definiert Abgabearten, die nicht ohne Zustimmung der Stakeholder manipuliert werden können.

Kapitel 7 „Projekt Meilensteine“ enthält die bisher erreichten Meilensteine.

2. Ausgangssituation und Ziele

Einleitung

Das gewählte Thema bezieht sich auf den Dienstleistungssektor. In der heutigen Welt hat alles begonnen, digital zu werden. Die Menschen fordern schnellere und einfachere Systeme und die Digitalisierung erfüllt diese Nachfrage.

Dafür wird die Digitalisierung auch in Restaurants benutzt. Menüs von Restaurants werden digitalisiert.

Problemstellung (Funktionalität)

Die Zeit ist heutzutage alles. Die Kunden wollen keine Zeit verbringen.

Eine Lösung der Zeit zu verbringen soll eine schnelle und einfache und praktische Menü-Idee stellen.

Die Nachhaltigkeit ist ein beliebter Begriff für alle Menschen. Bei jeder Änderung des Menüs ein neues Menü von Grund auf neu zu erstellen, ist weder in Bezug auf die Nachhaltigkeit noch auf die Kosten ein praktikabler Weg.

Ein digitales Menü kann alle diese Problemen lösen.

Stakeholder (Anwender):

Größten Vorteile des digitalen Menüs besteht darin, dass ein Restaurantbesitzer, der so viel Kunden hat, Gewinn am meisten erzielen kann. Ein anderer Vorteil ist Dank der Nachhaltigkeit für die Umwelt.

Systemumfeld (Einsatzumgebung)

Für die praktische Nutzung könnte man auf dem Handy oder Laptop/PC dem Menü erreichen.

Rahmenbedingung (Einschränkungen)

Die wichtigsten Einschränkungen befinden sich in der Wahl der Software Tools.

- die Backend Applikationen sollen mit Java Framework und Spring Framework,
- die Frontend Applikationen mit Thymeleaf realisiert werden, und
- die persistenten Daten sollen in einem PostgreSQL abgespeichert werden.

Ziele (Lösung)

Ein Prototyp wird entwickelt, das einem Restaurants-Menü digitalisiert. Darüber soll die Daten ständig aktuell sein.

Die Persistenz soll mit einem Datenbanksystem erfolgen.

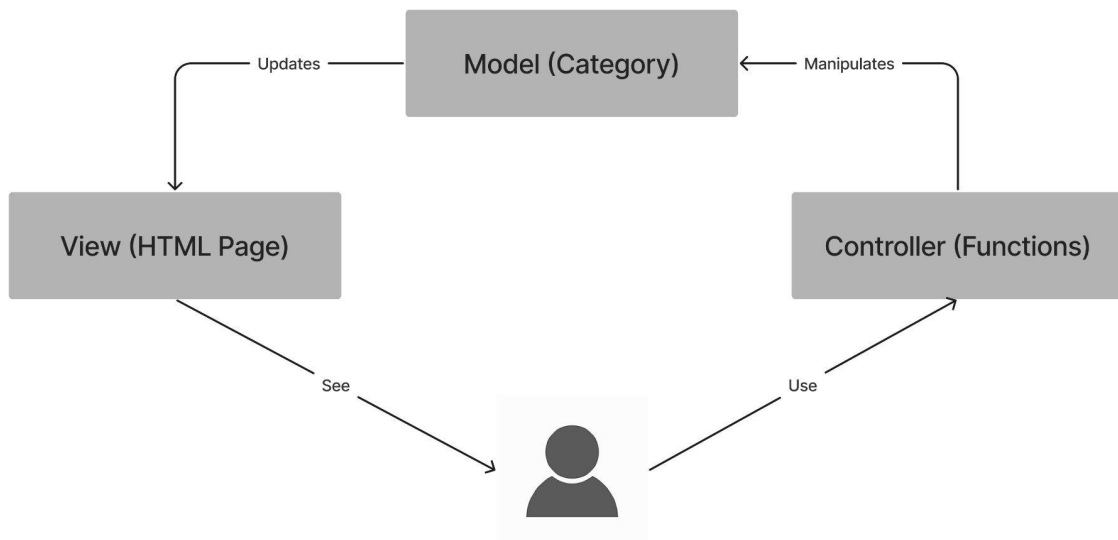
3. Gesamtarchitektur

Einleitung

In diesem Kapitel definieren wir die Gesamtarchitektur unseres Systems. Dies dient als kurze Einführung, um einen Eindruck von der Struktur des Systems zu bekommen. Es enthält Informationen über die Komponenten, die zum Aufbau dieses Systems verwendet werden. Es wird erklärt, wie Front-End und Back-End strukturiert sind und wie die Schnittstelle zwischen diesen Komponenten sie verbindet.

Gesamtarchitektur

Ein digitales Menü erfordert keine ausgeklügelte Architektur. Die Anforderungen an das System können in wenigen Teilen angegeben werden. Die Kategorien des Menüs und die Einträge sind die wichtigsten Datenmodelle. Diese sollten in einem persistenten Speicher (in unserem Fall eine PostgreSQL-Datenbank) gespeichert werden. Die Benutzer sollten in der Lage sein, einfach und schnell auf das Menü zuzugreifen. Der beste Weg, dies zu erreichen, ist die Verwendung von Web-Technologien, da Browser die am leichtesten zugängliche Plattform sind und sie keine Installation durch den Benutzer erfordern. Schließlich benötigen wir eine API, um die Daten abzurufen, zu bearbeiten und darzustellen. Wenn wir alle unsere Bedürfnisse berücksichtigen, ist ein einfaches MVC-Muster gut geeignet, um dies zu erreichen.



Picture 1: System_MVC_Design

4. Funktionale Anforderungen

Einleitung

In diesem Kapitel sind Anforderungen (inklusive User Stories und Use Cases) an Gesamtsystem aber auch an die einzelnen Systemkomponenten definiert. Außerdem gibt es ein Prototype.

UI Use Cases

- **/UI-1/** Die Menü und die spezifische Informationen (z.B. die allen Suppen unter Suppe-Menü) sollen ansehen werden.

(Ein Beispiel kann man mit dem Prototype "spesifik_Çorba_Menüsü" sehen.)



Prototype 1: Giriş_Sayfası auf Iphone 14 Pro



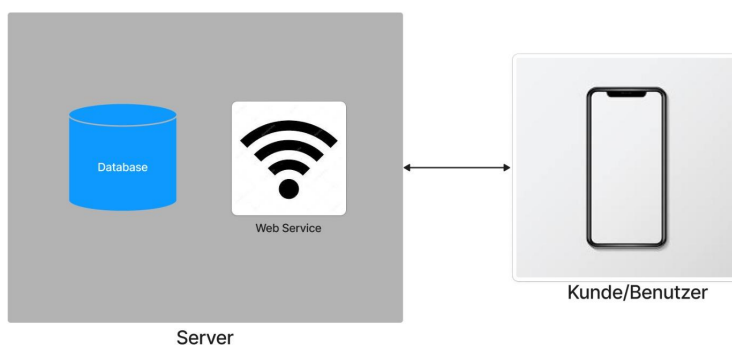
Prototype 2: spezifisch_Corba_Menüs auf iPhone 14 Pro

API Use Cases

- **/API-1/** Admin kann neue Optionen(Gericht oder Getränke) hinzufügen oder entfernen.

Technischen und fachliche Anforderungen

- **/SYS-1/** Die Komponenten sollen für die aktuellen Daten die Database (DBSM) nutzen.

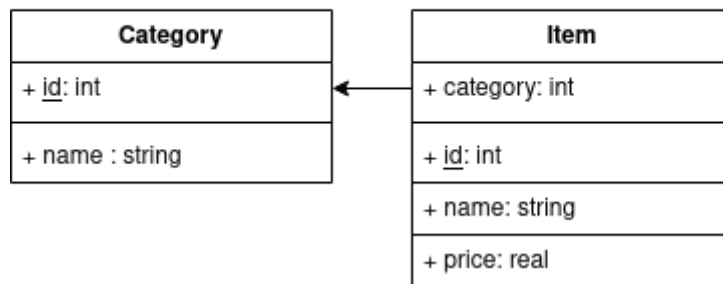


Picture 2: System Architektur.

Datenmodell

- **/DAT-1/** Die relevanten Systemparameter sind in einer externen Datenbank abzuspeichern.
- **/DAT-2/** Das Datenmodell soll die Parameter der API-Schnittstellen berücksichtigen.
- **/DAT-3/** Eine SQL-Datenbank ist für die Persistenz der Daten notwendig. Es kann ein Entity/Relationship Diagramm erstellt werden.

Beispieldatenmodelle



5. Nichtfunktionale Anforderungen

Einleitung

In diesem Kapitel sind die nicht-funktionalen Anforderungen ans Gesamtsystem aber auch an die einzelnen Systemkomponenten definiert. Es wird besonders auf die Software Qualität Wert gelegt

Nicht-funktionale Anforderungen an die Systemarchitektur (Architekturmuster, Deployment)

- **/SYS-1/** Deployment-Architektur: Das System ist klein genug, um es als Monolith zu halten, Datenbank und API können auf einem kleinen Server bereitgestellt werden.
- **/SYS-2/** Das Abrufen und Verarbeiten von Daten wird über eine REST-API gesteuert.

Nicht-funktionale Anforderungen an die Entwicklungsumgebung

- **/DEV-1/** Die Entwicklungsumgebung ist frei wählbar.

Nicht-funktionale Anforderungen an die Entwicklungswerkzeuge (Sprache, IDE, Frameworks)

- **/TOL-1/** Die Backend Applikation sollen mit Spring MVC Framework implementiert werden.
- **/TOL-2/** Die Frontend Applikation sollen mit Web-Technologien (HTML/JS) implementiert werden.
- **/TOL-3/** Die persistenten Daten sollen in einer SQL-Datenbank abgespeichert werden.

Nicht-funktionale Anforderungen an die Teststrategie (Qualitätssicherung)

- **/TEST-1/** Alle Use Cases, User Stories und Anforderungen sollen getestet und berichtet werden.
- **/TEST-2/** Unit-Tests sollten geschrieben werden, um die Datenbanktransaktionen zu überprüfen und die Admin-Rechte zu kontrollieren.

6. Abnahmekriterien

Die Abnahmekriterien sind durch den Stakeholder definiert und sie dürfen nur mit Zustimmung des Stakeholders neu definiert, geändert oder erweitert werden.

Das Projekt wird mit den folgenden Artefakten abgegeben:

- Dokumentation:
 - Pflichtenheft: template-inf202-projektname-pflichtenheft-2023.v1.0.doc
- Software
 - Link zu GitHub Projekt: [git-hub-link](#) (Dieser Link wird das Repository enthalten, wenn wir mit der Implementierung des Systems beginnen.)

Anm.: Die Abgabetermine der Projektartefakte werden durch den Stakeholder festgelegt!

7. Projekt Meilensteine

Folgende Meilensteine sind verbindlich definiert:

- Meilenstein #1: Das Lastenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.
- Meilenstein #2: Der Entwurf des Pflichtenhefts ist gegeben.