

TAU INF202 Software Engineering
Individuelles Projekt
Pflichtenheft

Projektdokumentation

Version: 2023.v1.0

Status: Abgabe

Take A Side

Verantwortliche/r:

Merve Damla İnce, e200503039@stud.tau.edu.tr

Damla Ersoy, e200503005@stud.tau.edu.tr

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

Dokumentenverwaltung

Dokument-Historie

| Version | Status *) | Datum | Verantwortlicher | Änderungsgrund |
|---------|-------------|------------|------------------------------|---|
| v1.0 | freigegeben | 25.02.2023 | Ö. Karacan | Vorlage wurde für die Studentenprojekte freigegeben |
| v1.1 | Entwurf | 4.04.2023 | M.Damla İnce Damla Ersoy | Pflichtenheft Entwurfsphase |
| v1.2 | Abgabe | 17.04.2023 | M. Damla İnce Damla Ersoy | Pflichtenheft Abgabe |

**) 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 Docs

Canva

Lucid

Inhaltsverzeichnis

| | |
|--|-----------|
| 1. Einleitung | 4 |
| 2. Ausgangssituation und Ziele | 5 |
| 3. Gesamtarchitektur | 6 |
| 4. Funktionale Anforderungen | 7 |
| 5. Nichtfunktionale Anforderungen | 8 |
| 6. Abnahmekriterien | 9 |
| 7. Projekt Meilensteine | 10 |
| 8. Referenzen | 11 |

1. Einleitung

Dieses Dokument dient dazu für die Realisierung einer Anwendungs verbindlich gestellten Anforderungen zu definieren, vollständig und konsistent zu beschreiben.

Die Use Cases und Anforderungen wird aus der Sicht des Stakeholders erklärt.

Im Kapitel 2 "Ausgangssituation und Ziele" wird die Ausgangssituation und der Grund zur Realisierung dieser Anwendung deutlich dargestellt.

Im Kapitel 3 "Gesamtarchitektur " wird die physikalische und die konzeptionelle Architektur des Systems, und die wichtigsten Subsysteme (Komponenten), die Anwender und die notwendigen Kommunikationsschnittstellen dargestellt.

Im Kapitel 4 "Funktionale Anforderungen" beinhaltet die Beschreibung der funktionalen Anforderungen, die Anwendungsfällen (Use Cases), Datenmodellen und technischen und fachlichen Anforderungen (Requirements).

Im Kapitel 5 "Nichtfunktionale Anforderungen" wird die funktionalen Anforderungen durch diejenigen Anforderungen erweitert, die keine funktionalen Anforderungen sind.

Im Kapitel 6 "Abnahmekriterien" wird die Abgabeartefakten festgelegt, die ohne Abstimmung des Stakeholders nicht zu manipulieren sind.

Im Kapitel 7 "Projekt Meilensteine" wird die wichtigsten Termine aufgelistet, die den Fortschritt die Teilergebnisse des Projektes definieren.

Im Kapitel 8 "Referenzen" wird sind die wichtigsten Referenzen aufgelistet.

2. Ausgangssituation und Ziele

Einleitung

Das ausgewählte Thema ist aus dem Unterhaltungsbereich und den sozialen Medien, wo letztlich immer mehr bekannt werden.

Das Pflichtenheft erfasst eine Anwendung, indem man bestimmte Objekte aus bestimmten Kategorien (Bücher, Filme, Spielen) votiert. Dadurch hat jede Objekte einen Vote-rate, was von Benutzern zu sehen ist

Dieses Kapitel beschreibt den Zweck der Erstellung dieser Anwendung, die Probleme, die sie bei ihrer Erstellung lösen wird, die Benutzergruppe, von der sie profitieren wird, unter welchen Bedingungen und mit was für einem System sie funktionieren wird.

Problemstellung (Funktionalität)

Lösen des Entscheidungsproblems durch Auflisten der Attribute von verwandten Objekten, die für andere Benutzer interessant sind, aufgrund der begrenzten Zeit, die Menschen der Unterhaltungswelt widmen können.

Um diesen Entscheidungsprozess beizubringen, würden Objekten mit bestimmten Attributen durch den Benutzern gesehen werden.

In diesem Projekt können Benutzer Entscheidungen einfacher treffen.

Stakeholder (Anwender):

Dieses Projekt könnte von der Verwendung des Systems für ein breites Publikum profitieren, darunter alle, die sich für Filme, Bücher und Videospiele interessieren. Sie können einfach die Anwendung benutzen, wo alle Objekten und Dienste zur Verfügung stehen.

Systemumfeld (Einsatzumgebung)

Das ganze System wird mit Software realisiert, da es keine physikalischen Komponenten außer einen Computer benötigt. Anwendung würde auf einem Computer laufen.

Rahmenbedingung (Einschränkungen)

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

- Als Software-Entwicklungstool könnte Eclipse verwendet werden,
- die Backend Applikationen sollen mit Java Framework und Spring Framework verwendet werden,
- die Frontend Applikationen mit JavaFX realisiert werden,
- die Daten sollen in einer Windows SQL-Datenbank abgespeichert werden.

Ziele (Lösung)

Es soll eine Anwendung entwickelt werden, die gewünschte Anfragen erfüllt und visuell erstellt.

Um das zu ermöglichen, könnten Komponenten kommunizierbar sein. Externe Kommunikationen wird durch Web Service realisiert. Die Persistenz soll mit einem Datenbanksystem erfolgen.

3. Gesamtarchitektur

Einleitung

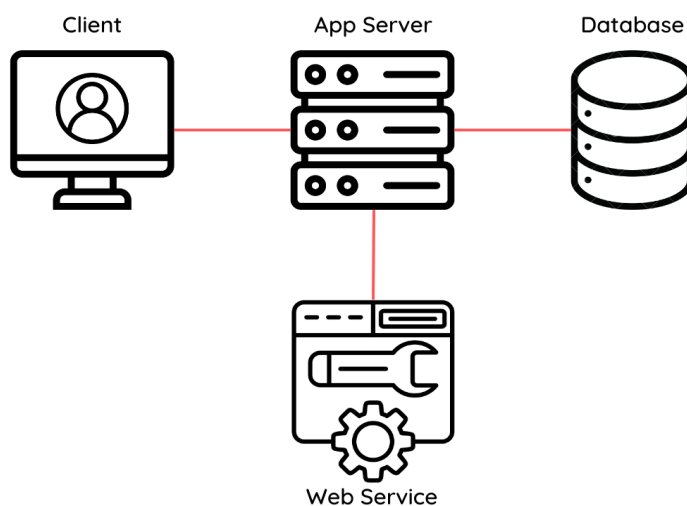
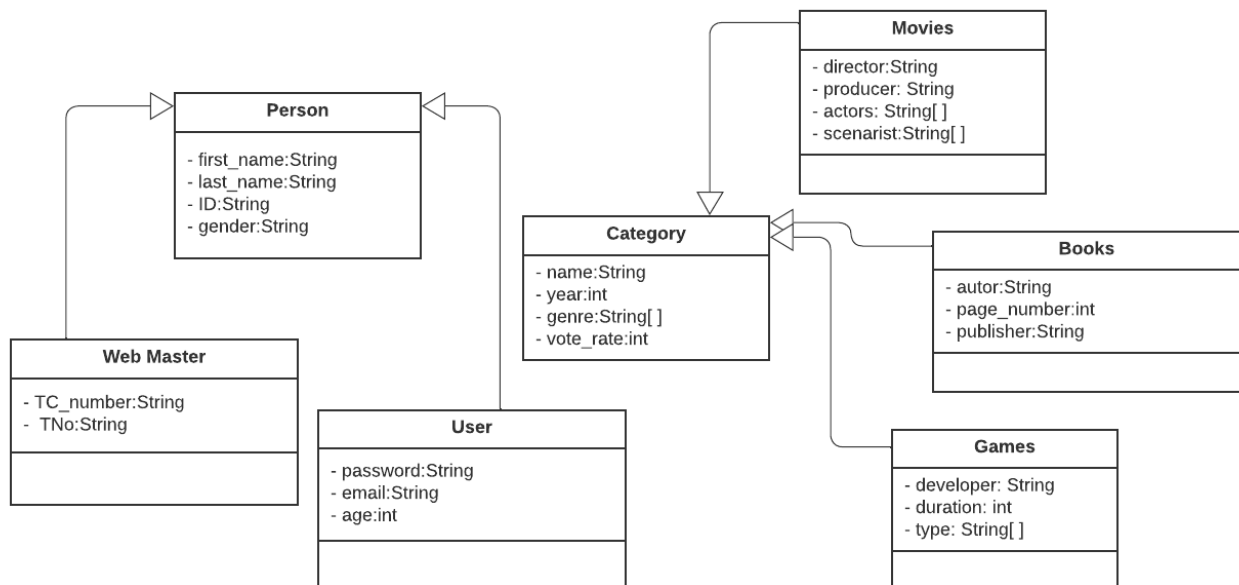
Dieses Kapitel gibt Auskünfte über Systemgrenzen mit Bildern. Externe Schnittstellen wie grafische Schnittstellen und API werden definiert. Die notwendigen Systemkomponenten und Datenstrukturen sind definiert und modelliert.

Dieses Kapitel führt zu dem besseren Verständnis des angeforderten Systems und dadurch genaue Definition der funktionalen und nicht-funktionalen Anforderungen.

Gesamtarchitektur

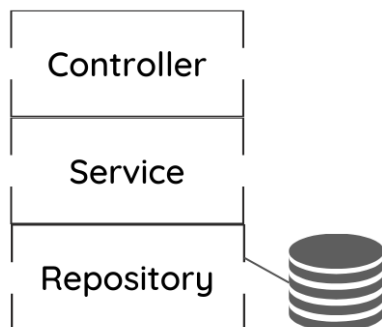
Die Gesamtarchitektur sind in den folgenden Aspekten betrachtet:

- Daten Strukturen
- Systemarchitektur zur Identifikation der autonomen Komponenten
- Komponentenarchitektur (Architekturmuster).



Konzept Architektur

Die Komponentenarchitektur würde MVC (Model-View-Controller) Architekturmuster entsprechen. Man kann Spring-Framework für detaillierte Informationen recherchieren.



Architekturmuster MVC für Systemkomponenten

Komponente Client-Application

- Es läuft auf dem Gerät von Benutzern. Anwendung wird die Interaktion zwischen Benutzer und App Server ermöglichen.
- Benutzer können Vote, Suche und Ansehen Aktionen mithilfe davon realisieren.

Komponente App Server

- App Server verarbeitet Benutzeranfragen.
- Es führt Geschäftslogik aus, ermöglicht Funktionen der Anwendung.
- Es kommuniziert mit der Datenbank und sendet Antworten zurück an die clientseitige Anwendung.

Komponente Database

- Die Daten der Anwendung wird in Database gespeichert und verwaltet.

Komponente Web Service

- Dieses Komponent ermöglicht den einfachen Zugriff auf den Webdienst durch verschiedene clientseitige Anwendungen.
- Es ermöglicht die Kommunikation.

Externe Schnittstellen

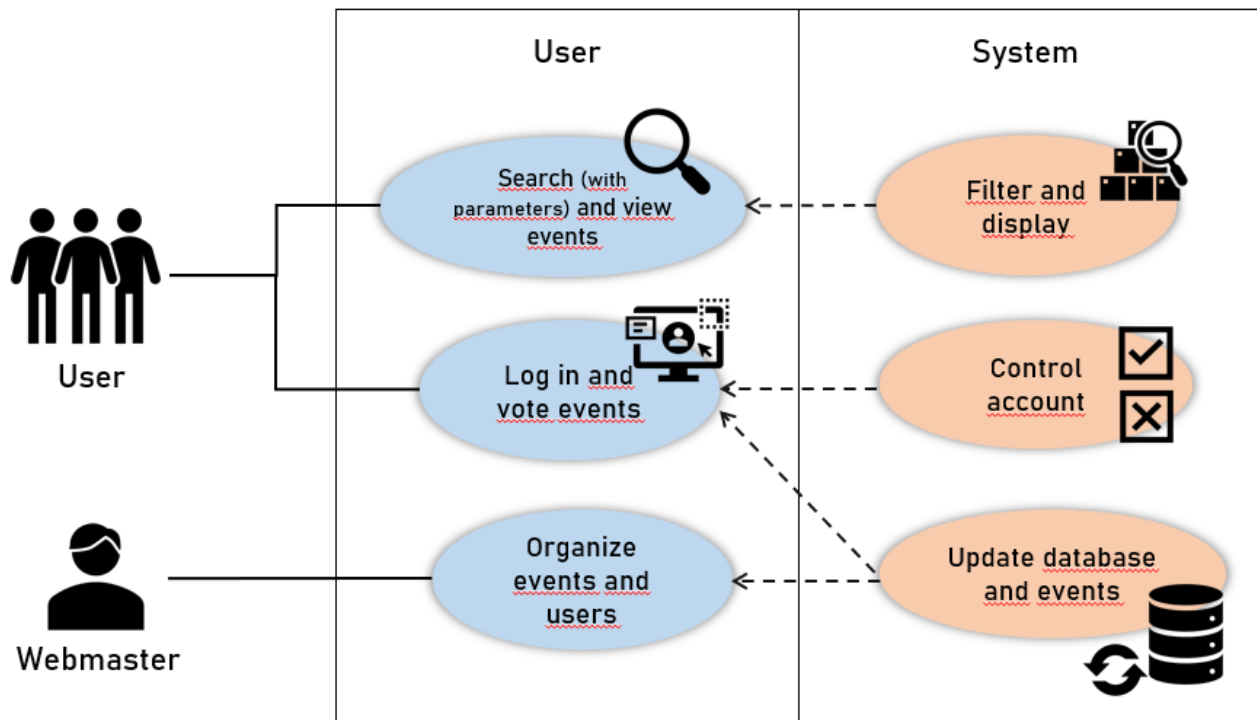
Für die Schnittstellen kann man das Kapitel "Funktionale Anforderungen" und Konzept Architektur ansehen.

4. Funktionale Anforderungen

Einleitung

In diesem Kapitel sind Anforderungen (inklusive User Stories und Use Cases) an Gesamtsystem aber auch an die einzelnen Systemkomponenten definiert. Darüber hinaus wird die GUI Prototype Schritt für Schritt eingeführt.

Use Cases and User Stories



USE CASE 1 <Log in and vote events>

Initial Assumption/Pre-Condition

- Der Benutzer muss im System registriert sein.

What Can Go Wrong

- Anmeldung (Log in) des Benutzers kann fehlschlagen (falsche E-Mail- oder Passworteingabe).

Normal Operationen

- **Step 1:** Benutzer melden sich mit ihren eigenen Passwörtern und E-Mail-Adressen bei ihren Konten an.
- **Step 2:** Nach dem Einloggen in ihr Konto sehen die Benutzer die Objekte, für die sie abstimmen möchten. Sie treffen eine binäre Wahl zwischen zwei Optionen, Like und Dislike.
- **Step 3:** Die Vote-Rates werden entsprechend den neuen Daten, die der Benutzer eingegeben hat (like oder dislike), aktualisiert.

System State on Completion/Post-Condition

- Benutzer können Objekte nach Parametern suchen und anzeigen.

USE CASE 2 <Search(with parameters) and view events>

Initial Assumption/Pre-Condition

- Der Pfad zur Eingabe der Parameter sollte korrekt sein, Attribute sollten dort geschrieben werden, wo sie hingehören.

What Can Go Wrong

- Ein Attribut, das nicht zum Objekt gehört, kann vom Benutzer eingegeben werden.

Normal Operationen

- **Step 1:** Der Benutzer zeigt die Top-Ten-Listen basierend auf den Vote-Rate aller drei Kategorien auf der Startseite an. (ohne Suche, schon verfügbar)
- **Step 2:** Der Benutzer wählt das Objekt aus, das er durchsuchen möchte, und zeigt die Attribute des zugehörigen Objekts an, die ausgewählt werden können.
- **Step 3:** Nach Auswahl der gewünschten Attribute wird gefiltert und das gefilterte Suchergebnis angezeigt.

USE CASE 3 <Organize events and users>

Initial Assumption/Pre-Condition

- Der Webmaster muss im System registriert sein.

What Can Go Wrong

- Anmeldung (Log in) des Webmasters kann fehlschlagen (falsche E-Mail- oder Passworteingabe).

Normal Operationen

- **Step 1:** Neue Objekte werden dem System vom Webmaster hinzugefügt.
- **Step 2:** Der Webmaster kann einige Benutzer aus dem System löschen.

System State on Completion/Post-Condition

- Aktualisieren der Datenbank gemäß neu eingegebenen oder gelöschten Daten.

Technischen und fachliche Anforderungen

UI Use Cases

- **/UI-1/** Im LOGIN PAGE werden Anmeldeinformationen der im System registrierten Konten angefordert. Um sich als Webmaster bei dem Konto anzumelden, muss der User-Anmeldeseite eine Anmeldeschaltfläche für den Webmaster hinzugefügt werden, siehe Abbildung 1- LOGIN PAGE
- **/UI-2/** Nach der Anmeldung stoßen Benutzer auf die HOMEPAGE. Auf dieser Seite werden die Top-Ten-Listen von Objekten in allen drei Kategorien nach ihren aktuellen Vote-Rate angezeigt, siehe Abbildung 2- HOMEPAGE
- **/UI-3/** Außerdem sollte jede Seite außer der LOGINPAGE eine Schaltfläche haben, um zur HOMEPAGE zurückzukehren.
- **/UI-4/** Auf der SEGMENT PAGE sollen Listen und einzelne Objekte nach ihren eingegebenen Parametern gefiltert werden, siehe Abbildung 3- SUCHE NACH LISTE IN EINEM SEGMENT PAGE und Abbildung 4- SUCHE NACH OBJEKT IN EINEM SEGMENT PAGE
- **/UI-5/** Und auf der SEGMENT PAGE sollten Benutzer in der Lage sein, für das ausgewählte Suchobjekt in binärer Form zu stimmen, siehe Abbildung 4- SUCHE NACH OBJEKT IN EINEM SEGMENT PAGE

- /UI-6/ Auf der WEBMASTER HOMEPAGE können Benutzer, die sich als Webmaster anmelden, gemäß den angegebenen Parametern löschen und hinzufügen, siehe Abbildung 5- WEBMASTER HOMEPAGE



Abbildung 1- LOGIN PAGE



Abbildung 2- HOMEPAGE

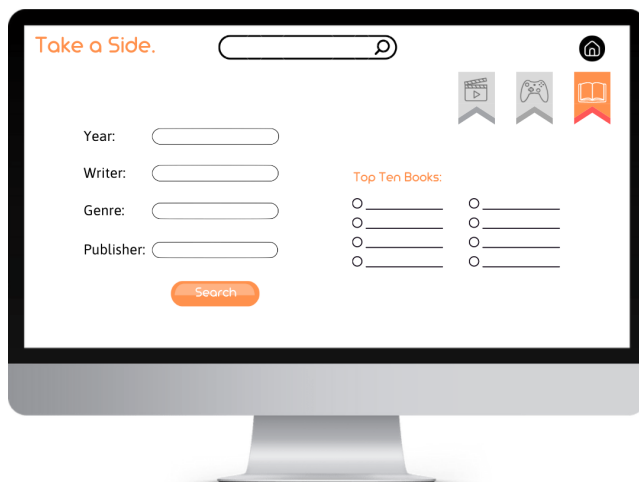


Abbildung 3- SUCHE NACH LISTE
IN EINEM SEGMENT PAGE



Abbildung 4- SUCHE NACH OBJEKT
IN EINEM SEGMENT PAGE



Abbildung 5- WEBMASTER HOMEPAGE

Datenmodell

- **/DAT-1/** Die relevanten Systemparameter sind in einer externen Datenbank abzuspeichern.
- **/DAT-2/** Ein externes DBMS (eine SQL-Datenbank) ist für die Persistenz der Daten notwendig. Es kann ein Entity/Relationship Diagramm erstellt werden.
- **/DAT-3/** Das DBMS soll zum Speichern der Daten per Komponente gewählt und eingerichtet sein.
- **/DAT-4/** Das Datenmodell soll die beschreibende Parameter der Use Cases berücksichtigen.
- **/DAT-5/** Das Datenmodell soll im UML Klassendiagramm modelliert werden.

5. Nichtfunktionale Anforderungen

Einleitung

Dieses Kapitel enthält die Informationen über die nichtfunktionale Anforderungen wie Kriterien des Systemverlaufs und einzelne Systemkomponente. Es ist erklärt, wie dieses System implementiert wird und was während der Realisierung des Systems benutzt wird.

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

- **/SYS-1/** Es ist gefördert, dass jede Systemkomponente ohne Abhängigkeit miteinander in ihren eigenen Betriebssystemen Prozesse laufen kann. Deshalb weist das System eine verteilte Architektur auf.
- **/SYS-2/** Die Systemkomponenten werden anhand von WebServices kommunizieren.

Nicht-funktionale Anforderungen an die Entwicklungsumgebung

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

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

- **/TOL-1/** Backend wird mit Java und Java Spring Framework implementiert.
- **/TOL-2/** Frontend wird mit JavaFX implementiert.
- **/TOL-3/** Die Daten werden im Windows SQL gespeichert.

Nicht-funktionale Anforderungen an die Teststrategie (Qualitätssicherung)

- **/TEST-1/** Alle Use Cases, User Stories und Anforderungen werden mit Hilfe von JUnit getestet und berichtet.

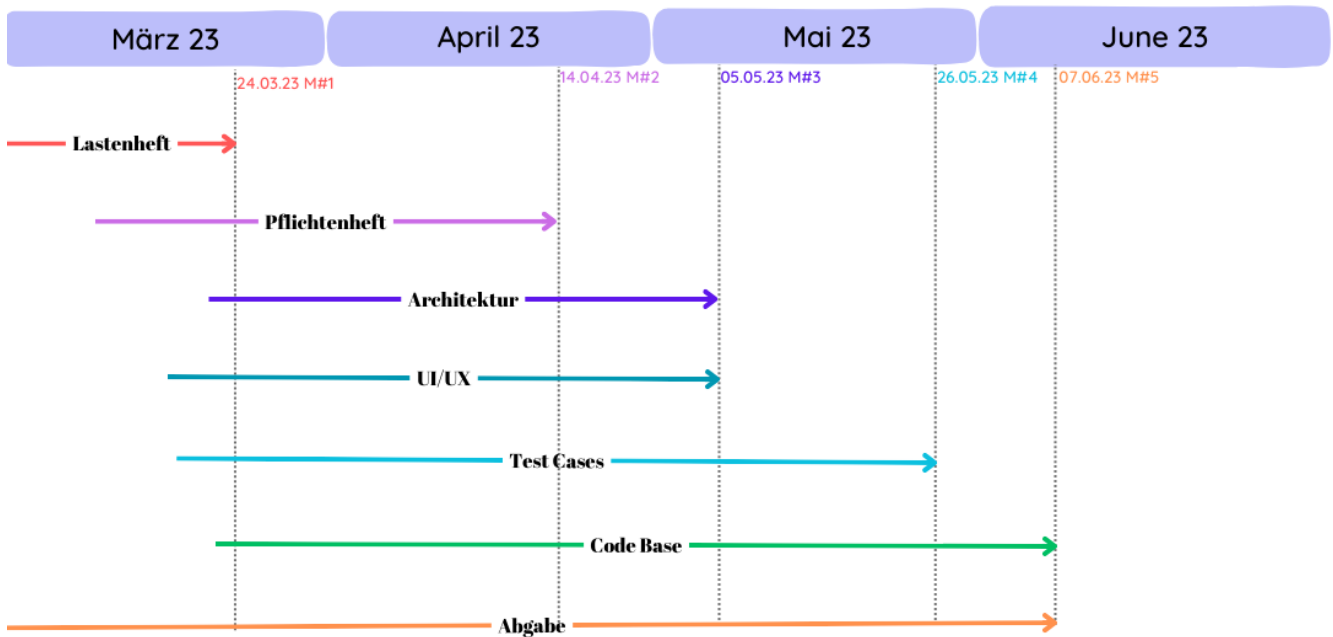
6. Abnahmekriterien

Das Projekt wird mit den folgenden Artefakten abgegeben:

- Dokumentation:
 - Pflichtenheft: *INF202-Take A Side-Pflichtenheft-2023.v1.0.pdf*
- Software
 - Link zu GitHub Projekt: *<https://github.com/orgs/Gruppe1-Fulya/teams/doubled>*

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

7. Projekt Meilenstein



Hier unten liegen die durch Stakeholder festgelegten Meilensteine.

Abgabedaten für M#1 → 24.03.2023

M#2 → 17.04.2023

M#3 → 05.05.2023

M#4 → 26.05.2023

M#5 → 07.06.2023

Meilenstein #1: Das Lastenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.

Meilenstein #2: Das Pflichtenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.

Meilenstein #3: (An diesem Meilenstein ist die Architektur im Vordergrund.)

Komponenten-basierte Architektur ist entworfen und komponentenweise implementiert.

Die externen Schnittstellen (Webservices) wurden entworfen/implementiert.

Die GUI Komponente ist ansatzweise fertig.

Meilenstein #4: (An diesem Meilenstein ist das Testen im Vordergrund.)

Die Test-Cases wurden aus den Use Cases abgeleitet und ansatzweise beschrieben.

Die Testumgebung ist vorbereitet und ein Smoke Test ist durchgeführt.

Meilenstein #5

Das Projekt ist per Vereinbarung abgegeben.

8. Referenzen

- INF202-SmartHome-PKW Ladestation-Pflichtenheft-2023.v0.1.doc
- INF202-VO#5#6-VO-System Requirements WS-Pflichtenheft.v1.pdf
- <https://deutsche-projekt-akademie.de/2022/04/funktionale-und-nicht-funktionale-anforderungen/>
- <https://www.acquisa.de/magazin/lastenheft-pflichtenheft>