

TAU INF202 Software Engineering

Individuelles Projekt

Pflichtenheft

Projektdokumentation

Version: 2023.v1.0

Status: In Review

Projektname: AutoXpert Auto-Service

Verantwortliche/r:

Mücahid Eren KOÇ

e190503044@stud.tau.edu.tr

İbrahim Alper ÖLÜÇ

e2005030016@stud.tau.edu.tr

Stakeholder:

DI. Ömer KARACAN, omer.karacan@tau.edu.tr

Dokumentenverwaltung

Dokument-Historie

Version	Status *)	Datum	Verantwortlicher	Änderungsgrund
v1.0	In Review	17.04.2023	M.Eren KOÇ İ.Alper ÖLÜÇ	Die erste Version des Pflichtenheftes wurde erzeugt.

**) 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
- Figma
- Lucidchart

Inhaltsverzeichnis

1. Einleitung	4
2. Ausgangssituation und Ziele	5
3. Gesamtarchitektur	7
4. Funktionale Anforderungen	10
5. Nichtfunktionale Anforderungen	13
6. Abnahmekriterien	15
7. Projekt Meilensteine	16
8. Referenzen	17

1. Einleitung

Dieses Dokument dient dazu die fürs Prototyping eines Informationsverwaltungssystem für eine Auto-Service Firma verbindlich gestellten Anforderungen zu definieren, und sie vollständig und konsistent zu beschreiben.

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

Im Kapitel 2 „Ausgangssituation und Ziele“ sind die Ausgangssituation und der Grund zur Wahl von Auto Service Informationsverwaltungssystem anschaulich dargestellt.

Im Kapitel 3 „Gesamtarchitektur“ sind die physikalische und die konzeptionelle Architektur des Systems und die wichtigsten Komponenten, die Anwender und die notwendigen Kommunikationsschnittstellen dargestellt. Hier sind auch zusätzliche Anforderungen an die Architektur oder Komponenten zu finden.

Im Kapitel 4 „Funktionale Anforderungen“ beinhaltet die Beschreibung der funktionalen Anforderungen durch die Ablaufbeschreibungen (User Stories), die Anwendungsfällen (Use Cases), und technischen und fachlichen Anforderungen (Requirements). Alle betriebsrelevanten Daten werden durch die Datenmodellen definiert.

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

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

Im Kapitel 7 „Projekt Meilensteine“ sind die wichtigsten Termine aufgelistet, die den Fortschritt die Teilergebnisse des Projektes definieren.

Im Kapitel 8 „Referenzen“ sind die wichtigsten Referenzen aufgelistet.

2.Ausgangssituation und Ziele

Einleitung

Das ausgewählte Thema ist aus den Dienstleistungen der Automobileindustrie, wo die Fahrzeuge nach dem entsprechenden Verfahren an die zuständigen Abteilungen übergeben werden, wo sie überprüft und repariert werden.

Das Pflichtenheft erfasst, dass einige Informationen, die für den Unternehmensservice erforderlich sind, in einer digitalen Umgebung aufbewahren, verarbeiten und diese Informationen für andere Teile des Unternehmens verfügbar machen. 'Informationen' wurde in Datenmodel aus 4. Kapitel detailliert erklärt.

Darüber hinaus wird es diese Informationen aufbewahren, um die Zuverlässigkeit des Unternehmens sicherzustellen, seine Rechtmäßigkeit zu dokumentieren.

Problemstellung (Funktionalität)

Viele Bereiche enthalten Informationen, die erfasst und gespeichert werden müssen. Diese Informationen wurden zu verschiedenen Zeiten unterschiedlich gehalten. Bei der Registrierung und Änderung von Informationen können viele Fehler auftreten.

Als Lösung dazu ist es, Daten digital zu speichern und zu modifizieren, dadurch wird die Zuverlässigkeit und Dauerhaftigkeit der Informationen erhöht.

Ein weiteres Problem ist der Informationsaustausch zwischen Service-Abteilungen. Ein Beispiel dafür ist, wenn das Auto Reparaturabteilung ist, will es einer Route gemäß den vorher festgelegten Schäden folgen. Das wird über eine Datenbank an die erforderlichen Informationen gelangen.

Stakeholder (Anwender):

In diesem Projekt wird der Schade Informationen und Abrechnungsprozess von Autos in einem Autoservice digitalisiert. Einem Mitarbeiter des Service werden Informationen präsentiert und er erhält die Möglichkeit, diese zu aktualisieren.

Systemumfeld (Einsatzumgebung)

Da die Informationen in einem Auto Service digitalisiert werden, läuft es in einem Computer von Auto Service

Rahmenbedingung (Einschränkungen)

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

- Als Software-Entwicklungstool soll entweder Eclipse oder IntelliJ verwendet werden,
- die Backend Applikationen sollen mit Java Framework und Spring Framework,
- die Frontend Applikationen mit JavaFX Rich Client Technologie realisiert werden, und
- die persistenten Daten sollen in einer SQL-Datenbank abgespeichert werden.

Ziele (Lösung)

Es soll eine Software entwickelt werden, mit der wir den Servicestatus der Autos aktualisieren und sichtbar machen können.

Zugriff und Austausch von Informationen von Autostatus zwischen Abteilungen des Unternehmens durch eine Datenbank erfolgen.

Die Persistenz soll mit einem Datenbanksystem erfolgen.

3. Gesamtarchitektur

Einleitung

In diesem Kapitel sind die Systemgrenzen definiert, die mit adäquaten Abbildungen präzisiert. Unter Berücksichtigung der Systemgrenzen sind per Anwender die externen Schnittstellen definiert (graphische Schnittstellen und API). 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 konzeptionelle Architektur des Systems besteht aus folgenden drei Hauptkomponenten:

- Benutzeroberfläche
- Backend
- Datenbank

Die Benutzeroberfläche (UI) umfasst eine Login- und Hauptseite mit allgemeinen Funktionen. Das Server-Backend ist zuständig für die Verarbeitung von Anfragen der Benutzer und die Kommunikation mit der Datenbank.

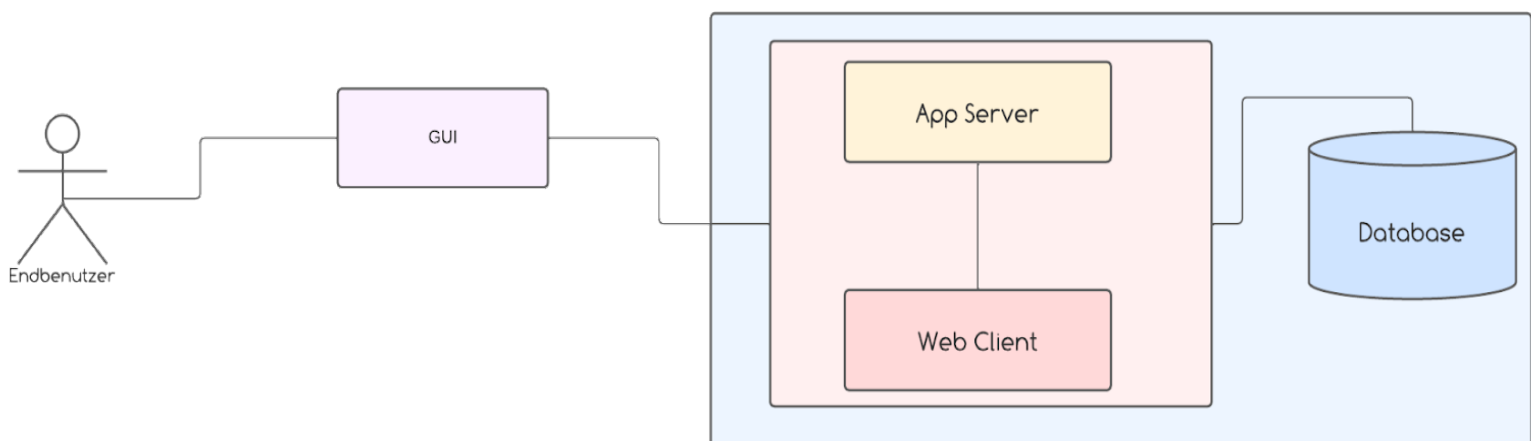


Abbildung 1 für die Konzeptionelle Architektur

Komponente

Benutzeroberfläche:

- Login-Seite:
 - o Die Schnittstelle, die autorisierte Benutzer verwenden, um sich beim System anzumelden.
- Hauptseite:
 - o Schnittstelle, die allgemeine Funktionen umfasst und es ihr ermöglicht, je nach ausgewählter Funktion verschiedene Operationen auszuführen.

Backend:

- Informationsverwaltung:
 - o Überwacht werden von Fahrzeuginformationen
 - o Aktualisierung der Informationen nach dem Bearbeitung am Fahrzeug

Datenbank:

- Die Datenbank sollte Kunden, ihre Fahrzeuge und ihre Schäden gemäß dem gegebenen Diagramm speichern.

Notiz: Mehrere Attribute, die innerhalb einer Ellipse geschrieben sind, sind separate Attribute (werden als Ellipsen betrachtet). Es ist so gemacht, um Verwirrung zu vermeiden.

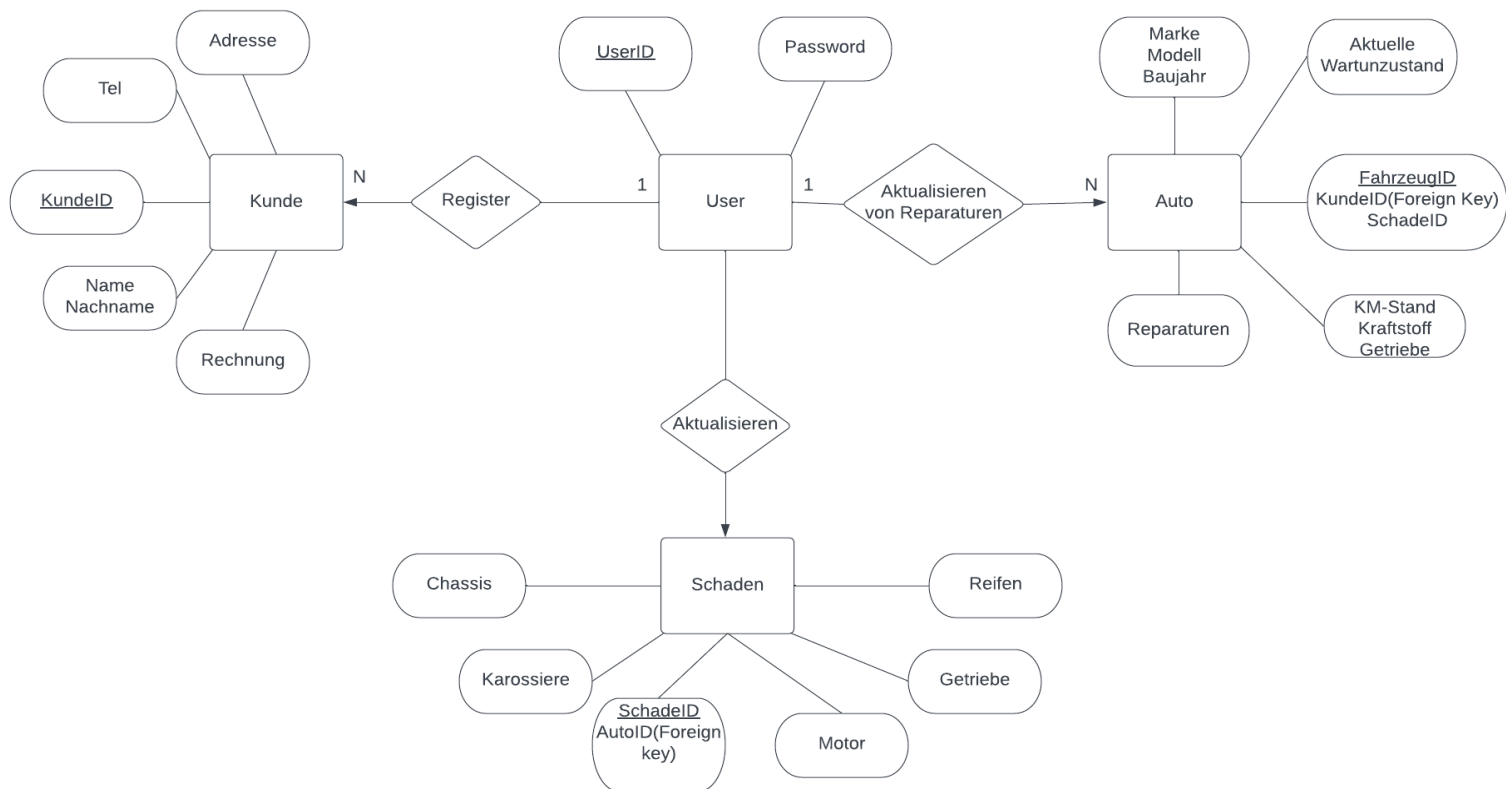


Abbildung 2 für das E/R Diagramm der Datenbank

4.Funktionale Anforderungen

Einleitung

Dieses Kapitel beinhaltet über Gesamtsysteme aber auch Systemkomponenten. Es beschreibt, was das System leistet und was es anbieten soll.

Es beschreibt, wie sich das System verhält und Vorgang der funktionalen Anforderungen ablaufen, über Use-Cases.

UI Use Cases

/UI-0/ Die Anmeldung am Programm erfolgt mit ID und Passwort. Die ID und das Passwort der Mitarbeitern, das Programm verwenden, werden von einem Administrator festgelegt.

- Wenn die ID oder das Passwort falsch eingegeben werden, wird eine Fehlermeldung an den Benutzer ausgegeben.

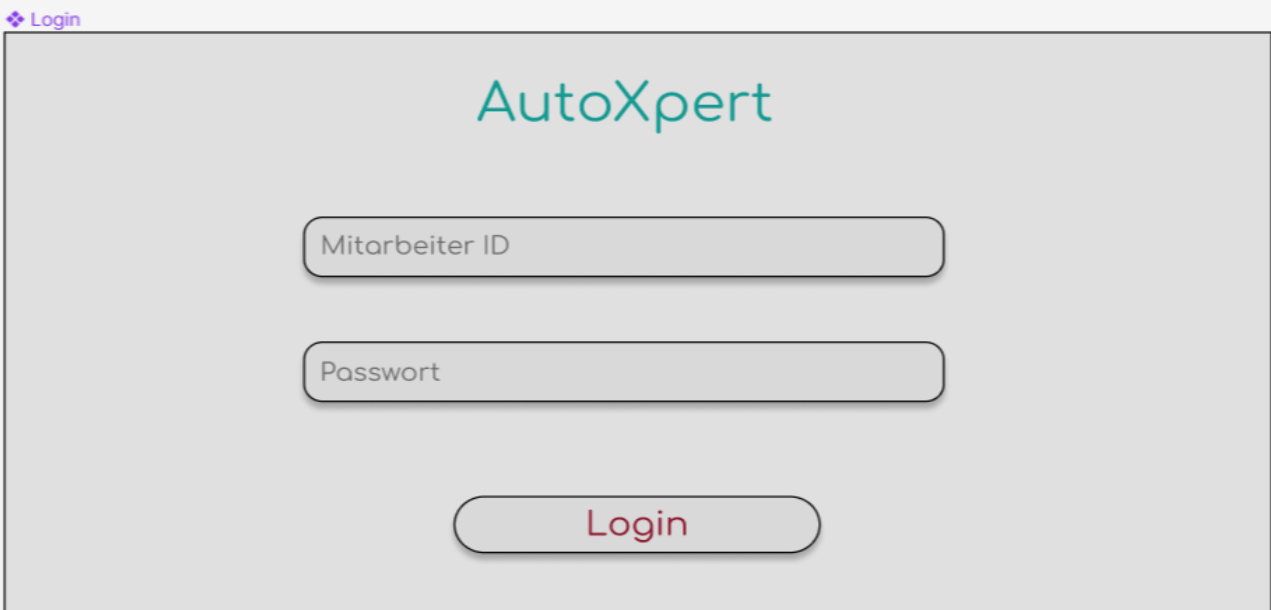


Abbildung 3 für die Use Case Login

/UI-1/ Neue Kunden und deren Fahrzeuge werden gemäß den im Datenmodell festgelegten Parametern in das System aufgenommen. Das Fahrzeug wird mit dem Status Sachverstand gespeichert.

/UI-2/ Fahrzeugschadeninformationen werden vom Benutzer aktualisiert.

/UI-3/ Der Status des Fahrzeugs wird von den Status Sachverstand auf den Status zur Reparatur aktualisiert.

/UI-4/ Nachdem die Reparatur abgeschlossen ist, werden die durchgeführten Reparaturen in das System eingegeben. Der Status des Fahrzeugs wird von den Status Reparatur zur Fertig aktualisiert.

API Use Cases

/API-1/ Mit dem Dienstsysteem RESTful GET Service können Sie auf die Auto Informationen auf der Datenbank zugreifen, falls Abteilungen vom Service wünschen oder um sie dem Benutzer zu präsentieren.

/API-2/ Um Aktualisierung, erst wird mit GET Web Service Informationen aus der Datenbank zugegriffen. Dann werden mit Set Web Service entsprechende Änderungen vorgenommen.

Technischen und fachliche Anforderungen

1. Das System muss auf einer PC laufen.
2. Die Anwendung muss als Server-Client-Architekturmodell implementiert werden.
3. Die Anwendung soll mit Java, Spring Boot und JavaFX entwickelt werden.
4. Als Datenbanktechnologie soll MySQL eingesetzt werden.

Datenmodell

- **/DAT-1/** Die relevanten Systemparameter sind in einer externen Datenbank abzuspeichern.
 - o Die Liste der Parameter wird später durch den Stakeholder zur Verfügung gestellt.
- **/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 Parameter der API-Schnittstellen berücksichtigen.
- **/DAT-5/** Das Datenmodell soll im UML-Klassendiagramm modelliert werden.

Datenmodell

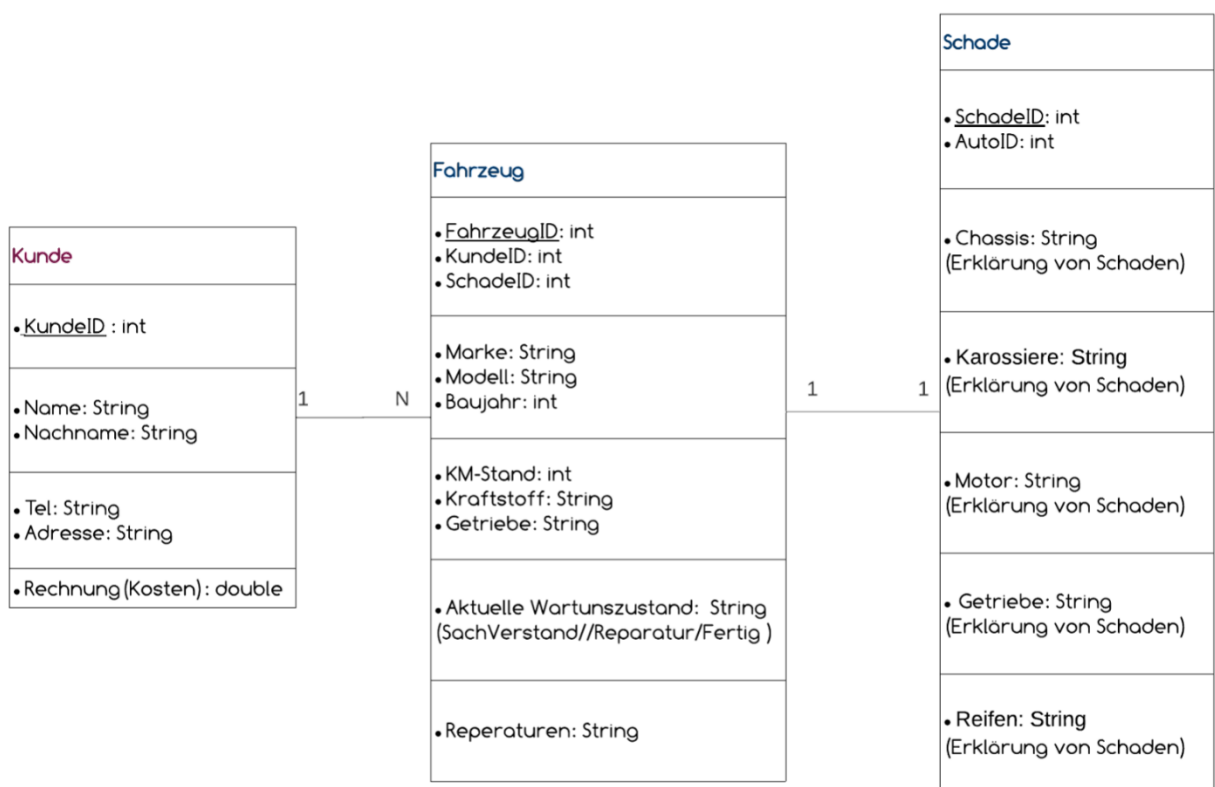


Abbildung 4 für die Use Case Login

5.Nichtfunktionale Anforderungen

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

/SYS-1/ Das System muss eine Architektur, die basierend auf Client-Server-Architekturmodell ist, genutzt werden, um eine einfache Bereitstellung und Wartung zu ermöglichen.

/SYS-2/Die Kommunikation der Systemkomponenten untereinander unterliegt der Beschränkungen der REST-Architektur und die Interaktionen den RESTful Webservices.

/SYS-3/ Das System muss auf Betriebssystem Windows laufen.

Nicht-funktionale Anforderungen an die Benutzererfahrung

/UX-1/ Die Nutzung des Systems sollte auf Basis von PC-Grundkenntnissen möglich sein, da die IT-Kenntnisse der potenziellen Anwender der Software variieren.

/UX-2/ Die Oberfläche sollte eindeutig und widerspruchsfrei gestaltet werden.

Nicht-funktionale Anforderungen an die Entwicklungsumgebung

/DEV-1/ Als Entwicklungsumgebung wird IntelliJ Idea verwendet.

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

/TOL-1/ Die Backend-Applikationen sollen mit Java und Spring Boot Framework implementiert werden.

/TOL-2/ Die Frontend Applikation soll mit JavaFX Rich Client Technologie realisiert werden.

/TOL-3/ Die persistenten Daten sollten in einer SQL-Datenbank gespeichert werden.

/TOL-4/ Um die Datenbank zu realisieren, wird MySQL verwendet.

Nicht-funktionale Anforderungen an die Teststrategie (Qualitätssicherung)

/TEST-1/ Jeder System-Service soll Black-Box getestet werden, siehe ISTQB-2018.

/TEST-2/ Alle Use Cases, User Stories und Anforderungen sollen getestet und berichtet werden.

6. Abnahmekriterien

Das Projekt wird mit den folgenden Artefakten abgegeben:

- Dokumentation:
 - o Pflichtenheft:
AutoXpert-INF202-Pflichtenheft-2023.v1.0.docx
- Software
 - o Link zu GitHub Projekt:
- Evidenz:
 - o System/Software-Demo via Videoclip: **<td>**

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: 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

SmartHome-PKW Ladestation-Beispielprojekt-Pflichtenheft:

[INF202-SmartHome-PKW Ladestation-Pflichtenheft-2023.v0.1.doc](#)

Java - <https://www.java.com>

Spring Boot - <https://spring.io/projects/spring-boot>

JavaFX - <https://openjfx.io>

MySQL - <https://www.mysql.com>