TAU INF202 Software Engineering

Individuelles Projekt

Pflichtenheft

Projektdokumentation

Version: 2023.17.04

Status: in Review

Plant Management System

Verantwortliche/r:
Mahmutcan İlhandağ, e190503018@stud.tau.edu.tr
Oğuzhan Topal, e190503001@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
v1.0	Entwurf	07.04.2023	Oğuzhan Topal Mahmutcan İlhandağ	Dokument erstellt wurde.
v1.0	in Review	16.04.2023	Oğuzhan Topal Mahmutcan İlhandağ	Dokument v1.0 wurde für die Zustellung vorbereitet.
v1.0	in Review	17.04.2023	Oğuzhan Topal Mahmutcan İlhandağ	Link für Video hinzugefügt.

**) 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

Inhaltsverzeichnis

1. Einleitung

Dieses Dokument wird verwendet, um einen Prototyp eines Plant-Management-Systems zu erstellen,Anforderungen zu definieren und sie vollständig und konsistent zu beschreiben.

Anwendungsfälle und Anforderungen werden aus Sicht der Stakeholder definiert.

Die grafische Oberfläche zur Zustandsüberwachung von Anlagen ist aus der Sicht einer Person. vom Benutzer beschrieben.

Kapitel 2 „Ausgangszustand und Ziele“ beschreibt den Ausgangszustand und das Ziel, das mit dem PMS erreicht werden soll.

Kapitel 3 „Allgemeine Architektur“ beschreibt die konzeptionelle Architektur des Systems. Systeme und Komponenten.Hier sind auch zusätzliche Anforderungen an die Architektur oder Komponenten zu finden.

Kapitel 4 „Funktionale Anforderungen“ enthält eine Beschreibung der funktionalen Anforderungen durch die Prozessbeschreibungen (User Stories), Use Cases (Usage Fälle) und fachliche und fachliche Voraussetzungen (Anforderungen). Daten werden durch Datenmodelle definiert.

In Kapitel 5 „Nichtfunktionale Anforderungen“ werden die funktionalen Anforderungen vervollständigt erweitert Anforderungen, die keine funktionalen Anforderungen sind.

Die „Abnahmekriterien“ definieren in Kapitel 6 Lieferwerke, die ohne Genehmigung der Behörde abgenommen werden. Stakeholder können nicht manipuliert werden.

Kapitel 7 „Projektmeilensteine“ listet die wichtigsten Termine auf, die den Fortschritt bestimmen.

2. Ausgangssituation und Ziele

Einleitung
Das gewählte Projektthema ist die Pflege von Topfpflanzen. Ziel des Projekts ist die Lösung der „Nachverfolgungs- und Planungsprobleme im Pflegeprozess“ von Interessenten für den Anbau von Topfpflanzen.

In diesem Abschnitt werden der Ausgangszustand und die Ziele des „Plant Management System” zusammengefasst.

Problemstellung (Funktionalität)
Heutzutage gibt es viele Menschen, die sich mit der Kultivierung von Topfpflanzen beschäftigen. Aber die Menschen können ihren Pflanzen nicht immer die nötige Zeit zuteilen. Wenn nicht genügend Zeit zugeteilt werden kann, kann was an der Anlage getan werden muss, nicht bestimmt werden. In diesem Projekt soll eine Software entwickelt werden, die den Benutzer über die Bedürfnisse der Anlage informiert, indem die gesammelten simulierten Daten mit den vorgegebenen optimalen Werten verglichen werden.

Stakeholder (Anwender):
Topfpflanzen stehen im Mittelpunkt der Nutzung dieses Projektes. Jeder, der Topfpflanzen anbaut, kann diese Software verwenden. Darüber hinaus umfasst die primäre Zielgruppe in Anbetracht der vom Projekt bereitgestellten Dienste Benutzer, die arbeiten und nicht genügend Zeit haben.

Systemumfeld (Einsatzumgebung)
Da es sich bei dem Projekt um einen Prototyp handelt und die Daten der Anlage in der Computerumgebung simuliert werden, wird das gesamte System in der Computerumgebung funktionieren. Es wird kein Produkt in der physischen Umgebung geben.

Rahmenbedingung (Einschränkungen)
Die wichtigsten Einschränkungen, die vorgenommen werden müssen, beziehen sich auf die Entwicklungsumgebung und die zu verwendenden Tools:

- Als Software-Entwicklungstool soll entweder Eclipse verwendet werden,
- die Backend Applikationen sollen mit Java,
- die Frontend Applikationen mit JavaFX realisiert werden, und
- die Daten sollen in einem SQL-Datenbank abgespeichert werden.

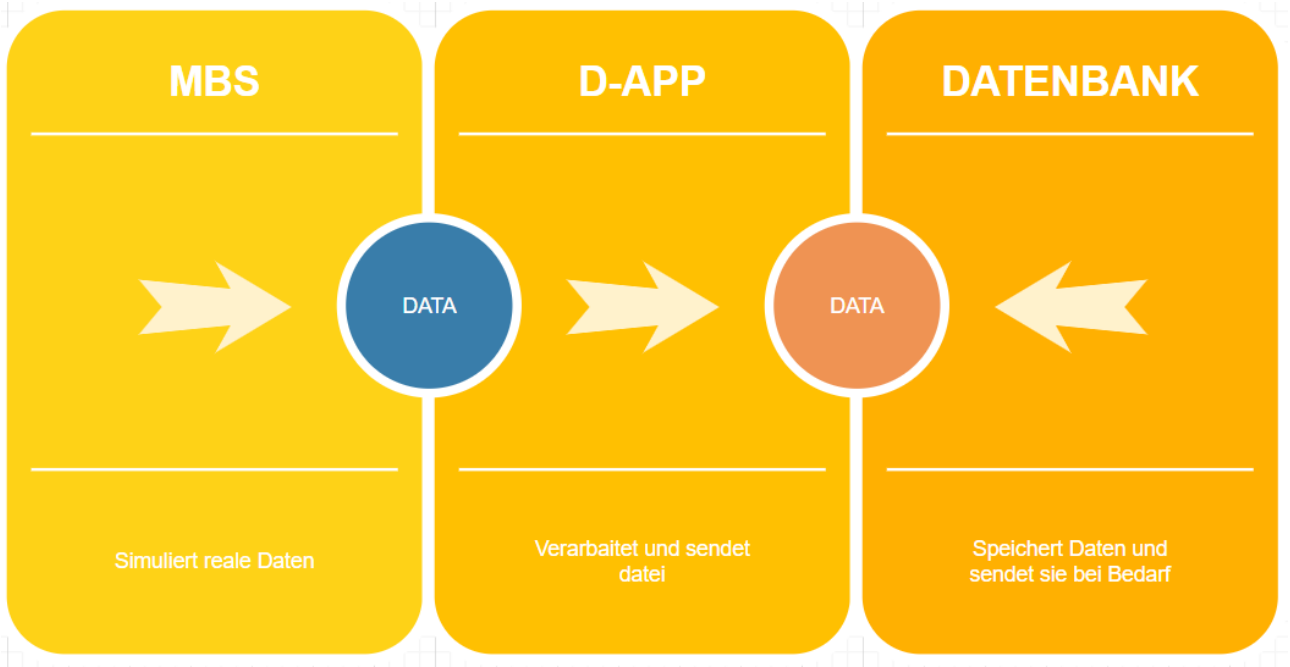
Ziele (Lösung)
Für den Prototypen soll eine Simulationsumgebung geschaffen werden. Die Daten werden in der Datenbank so gespeichert, als ob sie von den Sensoren am Blumentopf stammen, aber sie werden manuell eingegeben. Die Software verarbeitet diese Daten und informiert den Benutzer über verschiedene Situationen. Es analysiert die Daten und druckt den Bedarf der Pflanze als Nachricht auf dem Bildschirm.

3. Gesamtarchitektur

Einleitung
In diesem Abschnitt werden Systemkomponenten und ihre Funktionen definiert Erforderliche Systemkomponenten und Datenstrukturen
definiert und Arbeitsweisen modelliert werden
Dieser Abschnitt bietet ein besseres Verständnis und eine einfachere Verwendung des angeforderten Systems.

Gesamtarchitektur

Die Gesamtarchitektur sind in den folgenden Aspekten betrachtet:
- Plant Management System Infrastructure als Kontext,
- Systemarchitektur zur Identifikation der autonomen Komponenten.



Das System läuft auf einem einzelnen Computer und simuliert ein reales Plant-Management-System. Die zuvor vorbereiteten txt-Daten werden von MBS an die App gesendet, wo sie interpretiert und dem Benutzer mitgeteilt und zur Speicherung und späteren Anzeige an die Datenbank gesendet werden.

Komponente DA(Desktop-app)
Die DA-Komponente liest die Daten aus dem MBS und sendet sie zur Speicherung an die Datenbank. Anhand dieser Daten warnt es den Benutzer vor der Anlage.
Es hat auch eine UI, die es dem Benutzer ermöglicht, diese Daten zu sehen und zu verwenden.

Komponente MBS(Modul auf Blumentopf Simulation)

Die MBS-Komponente ist der Algorithmus, der die erforderlichen Daten in der datei.txt liest und in bestimmten Abständen an die DA sendet.
Diese Daten sind Licht, Feuchtigkeit und pH-Wert, MBS sendet sie innerhalb der angegebenen Zeit an die DA.

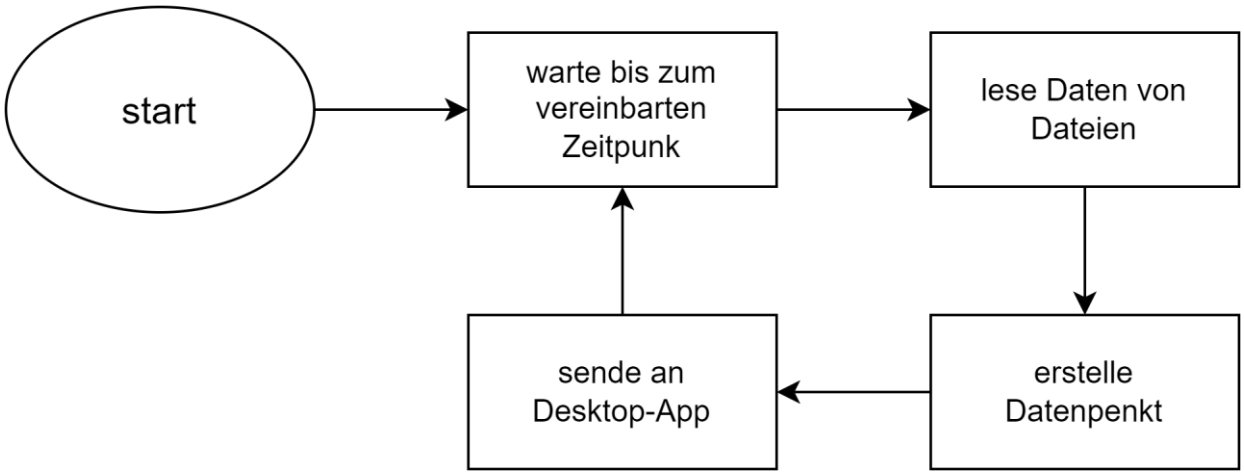
Komponente Datenbank
Die Datenbank speichert die Daten von der DA und sendet sie an die DA, wenn sie verwendet werden.
Mit SQL erstellt.

Externe Schnittstellen
User Interface (UI); Die Struktur, in der Benutzer Informationen und Warnungen sehen
datei.txt; Datei mit allen Informationen.

4. Funktionale Anforderungen

Einleitung
In diesem Kapitel sind Anforderungen (inklusive User Stories und Use Cases) an Gesamtsystem aber auch an die einzelnen Systemkomponenten definiert.

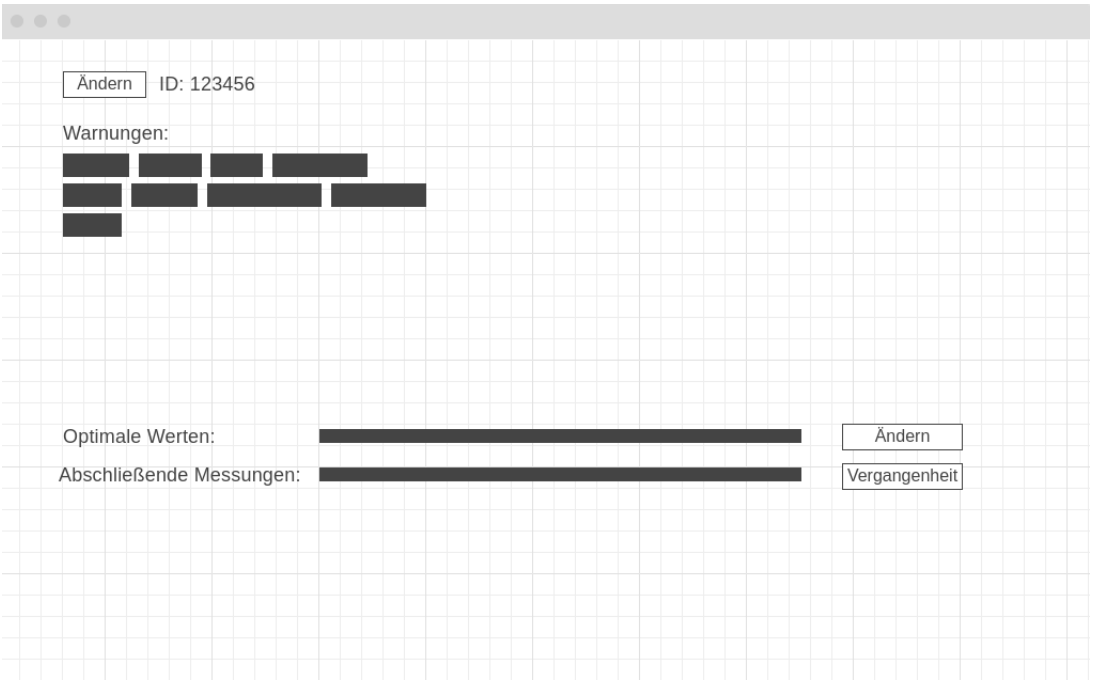
Modul auf Blumentopf Simulation Use Cases
- **/MBS-1/** Das Modul sollte in der Lage sein, Daten aus 3 verschiedenen Dateien zu lesen (3 Textdateien, die Licht-, pH- und Feuchtigkeitssensoren darstellen).
- **/MBS-2/** Das Modul muss terminiert werden. Er soll in bestimmten Zeitabständen eine Verbindung zum Client aufbauen und Daten senden. (Im realen Szenario wird dieses Zeitintervall mit 6 Stunden angenommen, in der Simulation wird es jedoch auf eine repräsentativ kurze Zeit eingestellt.)
- **/MBS-3/** Das Modul sollte die ihm zugewiesene ID aus einer ID-Datei in dem Folder lesen, in dem es arbeitet, und diese ID zu den gesendeten Daten hinzufügen.
- **/MBS-4/** Das Modul liest die Einstellungen aus einer Konfigurationsdatei, die die erforderlichen Einstellungen enthält. Kommunikation usw. lernt die notwendigen
- Informationen aus dieser Datei.



Desktop-App Use Cases
- **/DA-1/** Die Desktop App muss sich in regelmäßigen Abständen mit MBS verbinden und Daten auslesen können.
- **/DA-2/** Die Desktop App sollte die gelesenen Daten in der Datenbank speichern.
- **/DA-3/** Optimale Werte für die Topfpflanze sollten mit dem Desktop-APP-Programm in der Datenbank erfasst und bearbeitet werden.
- **/DA-4/** Die Desktop App soll die folgenden Situationen prüfen, indem sie die aktuell in die Datenbank hochgeladenen Daten mit den optimalen Daten vergleicht und die für die relevante Situation ermittelte Meldung auf dem Bildschirm ausgibt.
  - Ist das Umgebungslicht im Bereich, den es haben sollte?
  - Ist die Erde der Pflanze feucht genug?
  - Ist der pH-Wert des Bodens angemessen?
- **/DA-5/** Die Desktop App sollte in der Lage sein, die historischen Informationen aus der Datenbank zu lesen und sie dem Benutzer auf Anfrage anzuzeigen.
- **/DA-6/** Eine Desktop-Anwendung sollte in der Lage sein, mit mehr als einem MBS zu kommunizieren.

UI Use Cases

- **/UI-1/** MBS wird keine grafische Oberfläche haben, und die Anpassungen werden durch Bearbeiten der Konfigurationsdatei vorgenommen.
- **/UI-2/** Die Desktop-App wird eine grafische Oberfläche haben.
- **/UI-3/** Die Desktop-App druckt Warnmeldungen in der grafischen Benutzeroberfläche.
- **/UI-4/** Die Benutzeroberfläche der Desktop-App bietet Platz für eine optimale Datenbearbeitung.
- **/UI-5/** Es sollte möglich sein, zwischen den Modul-IDs auf der Oberfläche der Desktop-App umzuschalten.



Datenmodell

- **/DB-1/** Die Daten werden in einer externen SQL-Datenbank gespeichert.
- **/DB-2/** Werte von MBS und gewünschte optimale Werte werden in zwei getrennten Tabellen gehalten.
- **/DB-3/** Daten sollten zwischen dem Modul und der Anwendung in Paketen übertragen werden, die ID, Licht, Feuchtigkeit und pH enthalten.

Table: umweltWerte
ID (int)
zeit (time)
ph (double)
feuchtigkeit (int)
licht (int)

Table: optimalWerte
ID (int)
minPh (double)
maxPh (double)
minFeuchtigkeit (int)
maxFeuctigkeit (int)
minLicht (int)
maxLicht (int)

5. Nichtfunktionale Anforderungen

Einleitung
In diesem Kapitel sind die nicht-funktionalen Anforderungen ans Gesamtsystem aber auch an die einzelnen Systemkomponenten definiert.

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-Teil wird mit Java entwickelt.
- **/TOL-2/** Frontend-Teil wird mit Javafx entwickelt.
- **/TOL-3/** Die Daten sollten in einer SQL-Datenbank gespeichert werden.

Nicht-funktionale Anforderungen an die Teststrategie (Qualitätssicherung)

- **/TEST-1/** Alle Use Cases, User Stories und Anforderungen sollen getestet und berichtet werden.
- **/TEST-2/** Ob die MBS ihre Funktion erfüllt, wird selbst getestet.
- **/TEST-3/** Es wird getestet, ob die Desktop-App die Warnmeldungen in den richtigen Situationen auf dem Bildschirm ausgibt.

6. Abnahmekriterien

Das Projekt wird mit den folgenden Artefakten abgegeben:

- Dokumentation:
  - Pflichtenheft: *INF202-Plant Management System Pflichtenheft-2023.v.1.0.docx*
- Software
  - Link zu GitHub Projekt: *https://github.com/Gruppe1-Fulya/plantManagementSystem*
- Evidenz:
  - System/Software-Demo via Videoclip: *https://youtu.be/aZ3Xh72JcAY*

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

7. Projekt Meilensteine

Meilenstein M#1:

- Das Lastenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.

Meilenstein e#2 (Zwischenabgabe):

- Das Pflichtenheft-Entwurf ist fertiggestellt um mit dem Stakeholder abzustimmen.

Meilenstein M#2:

- Das Pflichtenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.

Meilenstein M#3:

- In 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 M#4:

- In 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 M#5:

- Das Projekt ist per Vereinbarung abgegeben.