

*TAU INF202 Software Engineering*  
*Individuelles Projekt*  
**System Testspezifikation**

Version: 24.05.2023

Plant Management System

Verantwortliche/r:

Mahmutcan İlhandag, 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	Entwurf	24.05.2023	Mahmutcan İlhandağ Oğuzhan Topal	Dokument erstellt wurde.

*\*) 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**

<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1. Einleitung</b>	<b>4</b>
<b>2. System Überblick</b>	<b>5</b>
<b>3. System Testfälle und Rückverfolgbarkeit</b>	<b>6</b>

## 1. Einleitung

In diesem Dokument soll der Entwurf und die Planung der Testanwendungen erfolgen, um nachvollziehen zu können, ob die von uns entwickelte Software „Plant Management System“ die im Pflichtenheft festgelegten Anforderungen korrekt erfüllt.

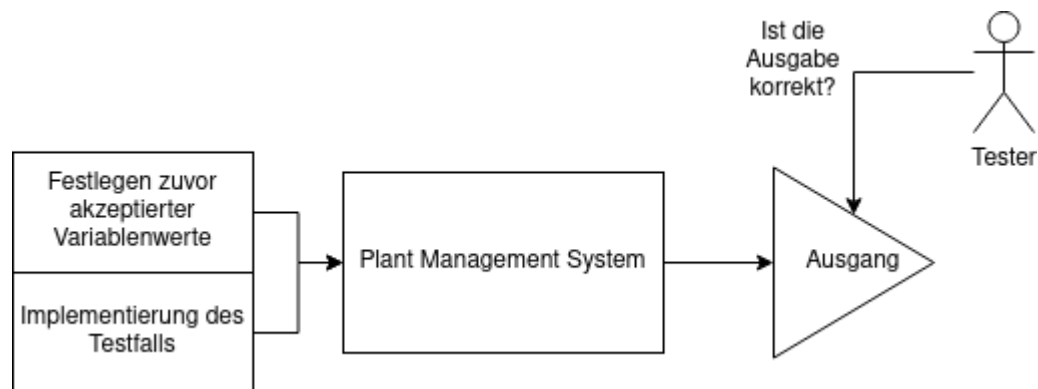
In *System Überblick*, werden wir die allgemeine Struktur des Systems und die ausgeführten Testprozesse darlegen.

In *System Testfälle und Rückverfolgbarkeit*, definieren wir die Tests, die wir durchführen werden, und zeigen, welchen Anwendungsfällen diese Tests entsprechen. Wir verraten, welche Architekturteile mit den durchgeführten Tests in Zusammenhang stehen.

## 2. System Überblick

Welche Kriterien die von uns entwickelte Software „Plant Management System“ erfüllen muss, haben wir zuvor anhand der Anwendungsfälle im Pflichtenheft ermittelt. Die Tests, die wir durchführen werden, wurden ebenfalls basierend auf diesen Anwendungsfällen als Ausgangspunkt erstellt.

Im nächsten Kapitel „System Testfälle und Rückverfolgbarkeit“ wird jeder Testschritt in einer separaten Tabelle detailliert beschrieben. Während der Testdurchführung wird jede Tabelle separat angewendet und ausgewertet. Sollte der Test fehlschlagen, werden die zugehörigen Architektur Einheiten überprüft und die Probleme behoben.



### 3. System Testfälle und Rückverfolgbarkeit

Die Tabellen zeigen, welchen Anwendungsfällen die Tests entsprechen. Aus Gründen der besseren Lesbarkeit wird es nicht in einem separaten Abschnitt angezeigt.

Test:	TEST001
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).
Architektur Komponenten:	ReadDaten
Testbeschreibung:	Testen, ob die mbs die Daten aus den Dateien, in denen sie aufgezeichnet sind, korrekt lesen kann.
Vorbereitungen:	Vorbereitung von pH- Licht- und Feuchtigkeits-TXT-Dateien.
Erwartete Ergebnisse:	Die Daten in der vorbereiteten TXT-Datei und die von der Funktion zurückgegebenen Daten sind konsistent.

Test:	TEST002
Use Cases:	/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.
	/DA-1/ Die Desktop App muss sich in regelmäßigen Abständen mit MBS verbinden und Daten auslesen können.
	/DB-3/ Daten sollten zwischen dem Modul und der Anwendung in Paketen übertragen werden, die ID, Licht, Feuchtigkeit und pH enthalten.

	/DA-2/ Die Desktop App sollte die gelesenen Daten in der Datenbank speichern.
Architektur Komponents:	Wait, Config, SendPackage, Recieve, WriteData
Testbeschreibung:	Testen, dass MBS die notwendigen Einstellungen aus der Einstellungsdatei liest und ein für diese Einstellungen geeignetes Paket vorbereitet und versendet.
Vorbereitungen:	Vorbereitung von Config File . 0 <= double ph <= 14 --- 0 <= double licht <= 100 --- 0 <= double feuchtigkeit <= 100
Erwartete Ergebnisse:	1- Empfangen eines Datenpakets vom Kommunikationsport des DA in bestimmten Zeitintervallen.
	2- In diesem Datenpaket stimmen die ID-Informationen mit den ID-Informationen in der Konfigurationsdatei überein.
	3- Die Variablenwerte im Datenpaket stimmen mit den ermittelten Werten überein.
	4- Die Lieferzeit des Pakets stimmt mit der Konfigurationsdatei überein.
	5- Die Daten in der Datenbank stimmen mit den Variablen überein.

Test:	TEST003
Use Cases:	/DA-3/ Optimale Werte für die Topfpflanze sollten mit dem Desktop-APP-Programm in der Datenbank erfasst und bearbeitet werden.
Architektur Komponents:	ReadOptimalData, WriteOptimalData
Testbeschreibung:	Testen, dass MBS die notwendigen Einstellungen aus der Einstellungsdatei liest und ein für diese Einstellungen geeignetes Paket vorbereitet und versendet.
Vorbereitungen:	0 <= double optph <= 14 0 <= double optlicht <= 100 0 <= double optfeuchtigkeit <= 100

<b>Erwartete Ergebnisse:</b>	Bearbeiten der Tabelle mit den optimalen Daten in der Datenbank und korrektes Lesen der aufgezeichneten Daten.
------------------------------	--

<b>Test:</b>	TEST004
<b>Use Cases:</b>	/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?
<b>Architektur Komponenten:</b>	CheckPH, CheckLight, CheckWater, SendMessageToUser
<b>Testbeschreibung:</b>	Test der Datenanalyse und Nachrichtenerstellung.
<b>Vorbereitungen:</b>	0 <= double ph <= 14 0 <= double licht <= 100 0 <= double feuchtigkeit <= 100 Optimale Werte in die Datenbank eingegeben.
<b>Erwartete Ergebnisse:</b>	1- Die durch den Vergleich der Daten mit den optimalen Daten erzielten Ergebnisse sind aussagekräftig.
	2- Die erstellten Nachrichtentexte sind aussagekräftig und in Bezug auf Sprache/Rechtschreibung korrekt.

<b>Test:</b>	TEST005
<b>Use Cases:</b>	/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.
<b>Architektur Komponenten:</b>	ReadAllData



<b>Testbeschreibung:</b>	Testen der Lesbarkeit historischer Daten aus der Datenbank.
<b>Vorbereitungen:</b>	Vorgefertigte Beispieldatenbank.
<b>Erwartete Ergebnisse:</b>	Die gelesenen Daten stimmen mit der Beispieldatenbank überein.

<b>Test:</b>	TEST006
<b>Use Cases:</b>	/DA-6/ Eine Desktop-Anwendung sollte in der Lage sein, mit mehr als einem MBS zu kommunizieren.
<b>Architektur Komponenten:</b>	SwitchPlant
<b>Testbeschreibung:</b>	Test der Kreuzung zwischen verschiedenen Pflanzen.
<b>Vorbereitungen:</b>	Vorgefertigte Beispieldatenbank.
<b>Erwartete Ergebnisse:</b>	1- Die gelesenen Daten gehören zur ausgewählten Anlage.
	2- Alle Anlagen im System werden aufgelistet.
	3- Beim Aktualisieren der Werkauswahl wird der Programmkontext entsprechend aktualisiert.