

# Sommerprüfung 2023

## **Ausbildungsberuf**

Fachinformatiker/Fachinformatikerin (VO 2020) Fachrichtung:  
Anwendungsentwicklung

## **Prüfungsbezirk**

Potsdam FI 1196-1 (AP T2V1)

Aland Mariwan

Identnummer: 1494891

E-Mail: kontakt@aland-mariwan.de, Telefon: +49 172 7134842

Ausbildungsbetrieb: GSSD mbH

Projektbetreuer: Peter Altmann

E-Mail: peteraltmann@gssd.de, Telefon: +49 3328 3534500

## **Thema der Projektarbeit**

Entwicklung einer Überwachung für Anwendungen.

# 1 Thema der Projektarbeit

Entwicklung einer Überwachung für Anwendungen.

## 2 Geplanter Bearbeitungszeitraum

Beginn: 18.02.2023

Ende: 21.04.2023

## 3 Ausgangssituation

Ein Kunde von der GSSD verwendet auf seinen Servern verschiedene Anwendungen, die als Dienste oder im Hintergrund laufen. Es muss sichergestellt werden, dass diese Anwendungen ständig funktionieren. Gegenwärtig gibt es ein Überwachungssystem, das nur unvollständige Zustände anzeigt.

## 4 Projektziel

Ziel ist die Erstellung einer Überwachung für ein System von Anwendungen. Auf einem Dashboard sollen die Status der Anwendungen dargestellt werden.

Die Umsetzung wird als Überwachungssystem realisiert, die es dem Administrator ermöglicht, alle verfügbaren Anwendungen auch in ihrer Zusammenarbeit zu überwachen. Die einzelnen Anwendungen müssen regelmäßig Informationen über ihren Zustand und ihre aktuelle Tätigkeit liefern. Diese Informationen schreiben die Anwendungen in Datenbanktabellen. Ein Backend liest die Daten aus der Datenbank, bewertet sie und zeigt die Ergebnisse im Frontend. Über ein Websocket werden die benötigten Informationen zwischen Frontend und Backend ausgetauscht. Auf dem Frontend kann der aktuelle Status von Anwendungen von jedem Nutzer überprüft werden, solange dieser eine Verbindung zum Frontend aufbauen kann. Über eine Filterfunktion kann die Ausgabe vom Backend eingeschränkt werden.

Das Überwachungssystem wird mit den Programmiersprachen HTML, VueJs, CSS, Bash, SQL, NodeJs realisiert. Die Anwendung wird vom Administrator überwacht und bei bestimmten Fehlermeldungen (Bsp.: Datenbank-Überlastung oder Anwendungsausfall) wird er auf geeignete Weise aufmerksam gemacht. Somit kann der Administrator reagieren.

## 5 Zeitplanung

Anlage 1

## 6 Anlagen

siehe Anlage 1

## 7 Präsentationsmittel

- PowerPoint
- Beamer
- Notebook

## 8 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

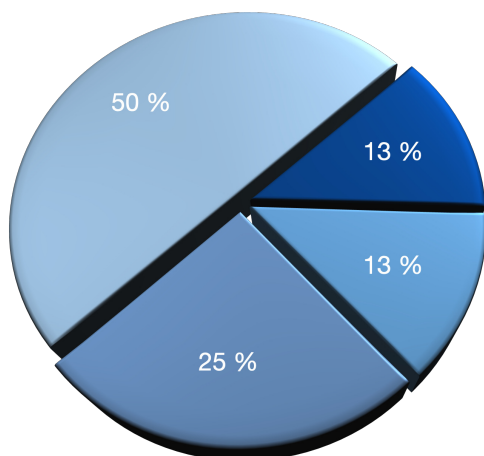
Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.

Anlage 1

# Projektstrukturplan erstellen

## Grafische und tabellarische Darstellung

- Planung und Konzeption
- Erstellung der Dokumentation
- Programmierung
- Abnahme



Die gesamte geschätzte Zeit beläuft sich auf 80h (h = Stunden):

Phase	Zeit in Stunden
Planung und Konzeption	20h

Phase	Zeit in Stunden
Programmierung	40h
Erstellung der Dokumentation	10h
Abnahme	10h
Total	80h

### Projektphasen mit Zeitplanung in Stunden

Planung und Konzeption	ca. 20h
Selbstständige Fortbildung	ca. 10h
Abprache mit der internen Entwicklungsabteilung für Schnittstellen und Frontend	ca. 5h
Projektstruktur bilden (Struktogramme und Programmablaufpläne)	ca. 5h
Programmierung	ca. 40h
Entwicklung des Backend	ca. 10h
Entwicklung der Websocket	ca. 10h
Entwicklung der Frontend	ca. 15h
Gesamte Testphase	ca. 5h
Erstellung der Dokumentation	ca. 10h
Anwenderdokumentation	ca. 2h
Technische Dokumentation	ca. 8h
Abnahme	ca. 10h
Qualitätssicherung	ca. 5h
interne Anwender	ca. 5h



ABSCHLUSSPRÜFUNG SOMMER 2023

FACHINFORMATIKERIN FÜR ANWENDUNGSENTWICKLUNG

DOKUMENTATION ZUR PROJEKTARBEIT

# Entwicklung einer Überwachung für Anwendungen.

Abgabetermin: Potsdam, den 26.04.2023

**Prüfungsbewerberin:**

Aland Mariwan  
Lortzingstraße 20  
14772 Brandenburg an der Havel

# GSSD

**Ausbildungsbetrieb:**

GSSD mbH  
Zehlendorfer Straße 5  
14513 Teltow

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>10</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Projektbeschreibung . . . . .	1
1.2. Projektziel . . . . .	1
1.3. Projektbegründung . . . . .	2
1.4. Projektschnittstellen . . . . .	2
1.5. Projektbegrenzung . . . . .	3
1.6. User-Cases . . . . .	3
<b>2. Projektplanung</b>	<b>4</b>
2.1. Projektorganisation . . . . .	4
2.2. Projektphasen . . . . .	5
2.3. Ressourcenplanung . . . . .	5
2.4. Entwicklungsprozess . . . . .	6
<b>3. Analysephase</b>	<b>6</b>
3.1. Ist-Analyse . . . . .	6
3.2. Workflow . . . . .	7
3.3. Soll-Analyse . . . . .	8
3.4. Wirtschaftlichkeitsanalyse . . . . .	9
3.4.1. "Make or Buy" -Entscheidung . . . . .	9
3.4.2. Projektkosten . . . . .	9
3.4.3. Amortisationsdauer . . . . .	9
3.5. Lastenheft . . . . .	9
<b>4. Entwurfsphase</b>	<b>10</b>
4.1. Zielplattform . . . . .	10
4.2. Architekturdesign . . . . .	10
4.3. Datenbank . . . . .	10
4.4. REST-API . . . . .	10
4.5. Entwurf der Benutzeroberfläche . . . . .	11
4.6. Maßnahmen zur Qualitätssicherung . . . . .	11
4.7. Deployment . . . . .	11
4.8. Pflichtenheft . . . . .	11
<b>5. Implementierungsphase</b>	<b>11</b>
5.1. Vorbereitung des Projektes . . . . .	11
5.2. Implementierung des Backendes . . . . .	12
5.3. Implementierung des Frontends . . . . .	12
5.4. Testen der Anwendung . . . . .	12



<b>6. Abnahmephase</b>	<b>12</b>
6.1. Code-Review . . . . .	12
6.2. Abnahme . . . . .	12
<b>7. Dokumentation</b>	<b>12</b>
<b>8. Fazit</b>	<b>13</b>
8.1. Soll-Ist-Vergleich . . . . .	13
<b>A. Anhang</b>	<b>14</b>
A.1. Pflichtenheft . . . . .	14

# Abkürzungsverzeichnis

**GSSD** Gesellschaft für Systemtechnik, Softwareentwicklung und Datenverarbeitungsservice

**SQL** Structured Query Language

**VM** Virtual Machine

**API** Application Programming Interface

**IDE** Integrated Development Environment

**UML** Unified Modeling Language

**ISO** International Electrotechnical Commission

**SQL** Structured Query Language

**VM** Virtual Machine

**API** Application Programming Interface

**IDE** Integrated Development Environment

**UML** Unified Modeling Language

**ISO** International Electrotechnical Commission

**SQL** Structured Query Language

**VM** Virtual Machine

**API** Application Programming Interface

**IDE** Integrated Development Environment

**UML** Unified Modeling Language

**ISO** International Electrotechnical Commission

**HTTP** Hypertext Transfer Protocol

**HTTPS** Hypertext Transfer Protocol Secure

**REST** Representational State Transfer

**API** Application Programming Interface

**JSON** JavaScript Object Notation

## **1. Einleitung**

In der folgenden Projektdokumentation stellt der Autor den Projektablauf dar, der im Rahmen seiner Ausbildung zum Fachinformatiker für Anwendungsentwicklung durchgeführt wurde. Die Durchführung der Projektarbeit erfolgt bei der Gesellschaft für Systemtechnik, Softwareentwicklung und Datenverarbeitungsservice mbH GSSD in Teltow. Zur Zeit beschäftigt die GSSD 4 Mitarbeiter und betreut verschiedene kleine und mittlere Unternehmen als IT-Dienstleister. Des Weiteren ist die GSSD Partner verschiedener Warenwirtschaftssoftware und bietet in diesem Umfeld Anpassung und Individualisierung an. Zweites Standbein ist der Vertrieb von Hardware für Unternehmen und Privatkunden sowie deren Installation und Wartung.

### **1.1. Projektbeschreibung**

Ein Kunde von der GSSD verwendet auf seinen Servern verschiedene Anwendungen, die als Dienste oder im Hintergrund laufen. Es muss sichergestellt werden, dass diese Anwendungen ständig funktionieren. Gegenwärtig gibt es ein Überwachungssystem, das nur unvollständige Zustände anzeigt. deswegen werde ich ein Überwachungssystem für verschiedene Anwendungen, die auf Servern laufen, entwickeln. Ziel des Projekts ist es, ein Dashboard zu erstellen, auf dem der Status der Anwendungen angezeigt wird. Das Überwachungssystem wird als Backend- und Frontend-Anwendung umgesetzt und ermöglicht es dem Administrator, alle verfügbaren Anwendungen sowie deren Zusammenarbeit zu überwachen. Die Anwendungen selbst liefern regelmäßig Informationen über ihren Zustand und ihre Tätigkeit, die in Datenbanktabellen geschrieben werden. Das Backend liest diese Informationen aus und bewertet sie, bevor sie im Frontend angezeigt werden. Der Austausch von Informationen zwischen Frontend und Backend erfolgt über ein Websocket. Jeder Nutzer kann den Status der Anwendungen auf dem Frontend überprüfen, solange er eine Verbindung hat. Eine Filterfunktion ermöglicht es dem Backend, die Ausgabe einzuschränken.

### **1.2. Projektziel**

Ziel ist die Erstellung einer Überwachung für ein System von Anwendungen. Auf einem Dashboard sollen die Status der Anwendungen echtzeitig dargestellt werden. Die Umsetzung wird als Überwachungssystem realisiert, die es dem Administrator ermöglicht, alle verfügbaren Anwendungen auch in ihrer Zusammenarbeit zu überwachen. Die einzelnen Anwendungen müssen regelmäßig Informationen über ihren Zustand und ihre aktuelle Tätigkeit liefern. Diese Informationen schreiben die Anwendungen in Datenbanktabellen. Ein Backend liest die Daten aus der Datenbank, bewertet sie und zeigt die Ergebnisse im Frontend. Über ein Websocket werden die benötigten Informationen zwischen Frontend und Backend ausgetauscht. Auf dem Frontend kann der aktuelle Status von Anwendungen von jedem Nutzer überprüft

werden, solange dieser eine Verbindung zum Frontend aufbauen kann. Über eine Filterfunktion kann die Ausgabe vom Backend eingeschränkt werden.

Der User soll in der Lage sein neue Sensoren zu registrieren.

Das Überwachungssystem wird mit den Programmiersprachen HTML, VueJs, CSS, Bash, SQL, NodeJs realisiert. Die Anwendung wird vom Administrator überwacht und bei bestimmten Fehlermeldungen (Bsp.: Datenbank-Überlastung oder Anwendungsausfall) wird er auf geeignete Weise aufmerksam gemacht. Somit kann der Administrator reagieren.

### 1.3. Projektbegründung

Die Überwachung von Anwendungen und Systemen ist ein entscheidender Aspekt im Bereich der Informationstechnologie, da sie eine frühzeitige Erkennung von Fehlern und Problemen ermöglicht, bevor diese sich auf den Betrieb und die Verfügbarkeit der Anwendungen auswirken können. Ein effektives Überwachungssystem ermöglicht es dem Administrator, die Gesamtleistung des Systems im Blick zu behalten, den Zustand der einzelnen Komponenten zu überwachen und schnell auf auftretende Probleme zu reagieren. Dadurch verbessert sich die Qualität und Verfügbarkeit von Anwendungen und die Kundenzufriedenheit steigt. Durch die Implementierung eines solchen Überwachungssystems können wir sicherstellen, dass Anwendungen und Systeme stets zuverlässig und fehlerfrei funktionieren. Wir können schnell auf mögliche Probleme reagieren, bevor sie sich auf den Betrieb auswirken und somit zur Gewährleistung der Verfügbarkeit des IT-Systems und somit haben wir einen reibungslosen IT-Infrastrukturbetrieb was die Kundenzufriedenheit steigert.

### 1.4. Projektschnittstellen

Das Projekt umfasst die folgenden Schnittstellen:

- Geschäftsführung: Der Geschäftsführer der Abteilung, für die das Überwachungssystem implementiert wird, muss über den Fortschritt des Projekts informiert werden und Entscheidungen treffen können, die das Projekt beeinflussen.
- IT-Abteilung: Die IT-Abteilung ist für die technische Umsetzung des Projekts verantwortlich und muss eng mit dem Projektteam zusammenarbeiten.
- Externe Dienstleister: Falls externe Dienstleister in die IT-Infrastruktur des Unternehmens eingebunden sind, müssen Schnittstellen definiert werden, um sicherzustellen, dass das Überwachungssystem auch auf diesen Systemen funktioniert.
- Benutzer: Die Benutzer des Systems müssen über Änderungen informiert und in das Projekt eingebunden werden, um sicherzustellen, dass das Überwachungssystem den Anforderungen der Benutzer entspricht.
- Datenbanken und Anwendungen: Das Überwachungssystem muss in der Lage sein, mit anderen Datenbanken und Anwendungen zu kommunizieren, um die erforderlichen Informationen zu sammeln und zu verarbeiten.

- Netzwerk- und Sicherheitsinfrastruktur: Die Implementierung des Überwachungssystems erfordert möglicherweise Änderungen an der Netzwerk- und Sicherheitsinfrastruktur des Unternehmens, die von anderen Abteilungen verwaltet werden. Daher müssen Schnittstellen definiert werden, um sicherzustellen, dass alle beteiligten Parteien informiert und beteiligt sind.

Die Umsetzung der Schnittstellen erfolgt in enger Zusammenarbeit zwischen dem Projektteam und den beteiligten Parteien. Es müssen klare Verantwortlichkeiten und Kommunikationswege definiert werden, um sicherzustellen, dass alle beteiligten Parteien über den Fortschritt des Projekts informiert sind und dass Probleme schnell gelöst werden können.

## 1.5. Projektbegrenzung

- Das Bauen der Sensor ist kein Bestandteil des Projekts.
- Das Projekt konzentriert sich ausschließlich auf die Überwachung von Anwendungen und Systemen innerhalb der internen IT-Infrastruktur des Unternehmens und auch externen Systeme oder Dienstleistungen.
- Das Überwachungssystem wird Alle mögliche Geräten oder Servern innerhalb der internen IT-Infrastruktur des Unternehmens und auch externen Systeme oder Dienstleistungen implementiert, die von hoher Priorität sind.
- Das Projekt beinhaltet keine Anpassungen an bestehenden Anwendungen oder Systemen, sondern konzentriert sich ausschließlich auf die Implementierung eines Überwachungssystems.
- Das Budget für das Projekt ist begrenzt und schließt keine umfangreichen Änderungen an der IT-Infrastruktur oder den Anwendungen ein.
- Die Implementierung des Überwachungssystems wird innerhalb eines bestimmten Zeitrahmens abgeschlossen, um sicherzustellen, dass es so schnell wie möglich in Betrieb genommen werden kann.
- Das Projekt umfasst keine Schulungen für Benutzer oder Mitarbeiter, sondern konzentriert sich auf die technischen Aspekte der Implementierung des Überwachungssystems.
- Das Überwachungssystem wird nur von einem bestimmten Team oder einer bestimmten Abteilung innerhalb des Unternehmens genutzt, und nicht von der gesamten Belegschaft.

## 1.6. User-Cases

- Das Überwachungssystem soll dem Administrator die Möglichkeit geben, den aktuellen Zustand des Systems mit einem Blick zu beurteilen und frühzeitig auf potenzielle Probleme hinzuweisen.

Akteur

- Administrator: Verantwortlich für die Überwachung des Systems und das Beheben von Problemen.

Beschreibung:

1. Der Administrator öffnet das Dashboard des Überwachungssystems.
2. Das Dashboard zeigt eine Übersicht aller verfügbaren Anwendungen und deren aktuellen Zustand an.
3. Der Administrator kann die Liste der Anwendungen nach verschiedenen Kriterien filtern, um nur relevante Informationen anzuzeigen.
4. Für jede Anwendung wird eine Bewertung angezeigt, die auf dem aktuellen Zustand und anderen Faktoren basiert.
5. Wenn eine Anwendung einen kritischen Zustand erreicht oder ein Problem auftritt, wird der Administrator benachrichtigt.
6. Der Administrator kann auf die Benachrichtigung reagieren, indem er das Problem löst oder einen anderen Schritt unternimmt, um das System wieder in den Normalzustand zu versetzen.
7. Der Administrator kann das Überwachungssystem anpassen, indem er die Bewertungsfaktoren für die Anwendungen ändert oder neue Anwendungen hinzufügt.

Alternativszenarien:

- Wenn das Dashboard nicht korrekt geladen werden kann, wird der Administrator über das Problem informiert und kann versuchen, das Problem zu lösen oder den technischen Support kontaktieren.
  - Wenn das Überwachungssystem nicht alle erforderlichen Informationen von den Anwendungen empfangen kann, wird der Administrator benachrichtigt und kann das Problem beheben oder die Anwendungen anpassen.
  - Wenn das Überwachungssystem nicht auf potenzielle Probleme hinweist, die der Administrator erwartet hat, kann der Administrator die Bewertungsfaktoren überprüfen und anpassen, um sicherzustellen, dass alle relevanten Faktoren berücksichtigt werden.
- Die Entwickler können neue Sensoren registrieren.

## 2. Projektplanung

### 2.1. Projektorganisation

Geleitet wird das Projekt durch den Autor. Ebenso erfolgt durch ihn die Ausarbeitung des Pflichtenheftes auf Grundlage des Lastenheftes Anhang A.1: Lastenheft auf Seite II, die daraus folgende Erarbeitung der einzelnen Komponenten und Prozessketten, sowie Aktivitäten im Gesamtprozess.

## 2.2. Projektphasen

Zur Durchführung des Projektes standen 80 Stunden zur Verfügung. Für die Ausarbeitung des Konzeptes wird die meiste Zeit eingeplant. Eine ausführliche Projektplanung ist im voraus nötig, allerdings kann auf Grund fehlender Erfahrung keine genaue Zeit geschätzt werden, es sind 15 Stunden angesetzt. Da der Autor zum Zeitpunkt der Durchführung dieser Projektarbeit mit den anzubindenden Komponenten und Problemen bei der täglichen Arbeit in der GSSD vertraut ist, wird ein Zeitaufwand von 6 Stunden für die Analysephase als realistisch eingeschätzt. Die weiteren Phasen werden als unkritisch eingestuft und sollten in der Ausarbeitung um maximal eine Stunde je Phase abweichen

Planung und Konzeption	ca. 20h
Selbstständige Fortbildung	ca. 10h
Absprache mit der internen Entwicklungsabteilung für Schnittstellen und Frontend	ca. 5h
Projektstruktur bilden (Struktogramme und Programmablaufpläne)	ca. 5h
Programmierung	ca. 40h
Entwicklung des Backend	ca. 10h
Entwicklung der Websocket	ca. 10h
Entwicklung der Frontend	ca. 15h
Gesamte Testphase	ca. 5h
Erstellung der Dokumentation	ca. 10h
Anwenderdokumentation	ca. 2h
Technische Dokumentation	ca. 8h
Abnahme	ca. 10h
Qualitätssicherung	ca. 5h
interne Anwender	ca. 5h

Abbildung 1: Projektphasen mit Zeitplanung in Stunden

Tabelle 1 zeigt die grobe Zeitplanung aus dem Projektantrag.

Dem Anhang A.2: Detaillierte Zeitplanung auf Seite III kann die detaillierte Zeiteinteilung entnommen werden.

## 2.3. Ressourcenplanung

Für das Projekt werden alle essentiellen Ressourcen durch die Standardausstattung eines IT-Unternehmens abgedeckt. Die Planungsschritte wurden im Büro der GSSD

durchgeführt und für die Erstellung des Projektplans standen dem Autor ein PC mit  $\text{\LaTeX}$  und WhiteStarUML in der freien Version GNU General Public License version 2.0 (GPLv2) zur Verfügung. Für die Gestaltung der Benutzeroberfläche wurde das Tool Figma gewählt. Mit seinen umfangreichen Funktionen konnte schnell und effektiv ein ansprechendes Design erstellt werden. Besonders produktiv war die Zusammenarbeit mit dem Auftraggeber durch die Möglichkeit des gemeinsamen Arbeitens in Figma. Für die Umsetzung des Projekts wurden folgende Ressourcen eingeplant:

- Personal:
  - Der Autor mit 80 Arbeitsstunden
  - Ein Projektbetreuer (Mitarbeiter der GSSD) mit 8 Arbeitsstunden
- Hardware:
  - Ein Notebook mit Zugang zum Intranet und Internet
- Software:
  - Visual Studio Code
  - NodeJs
  - VueJs
  - MariaDB Server/Client
  - WhiteStarUML
  - Figma
  - $\text{\LaTeX}$ .

Um Kosten zu sparen, wurde darauf geachtet, keine kostenpflichtige Software einzuplanen.

## 2.4. Entwicklungsprozess

Für die Gestaltung der Benutzeroberfläche wurde ein agiler Prozess gewählt, um kurzfristig auf die Wünsche des Auftraggebers eingehen zu können. Hierfür fanden regelmäßige Rücksprachen statt. Für die Entwicklung der Server-Komponenten wurde das Wasserfall-Modell gewählt, da der Autor keine späteren Korrekturen in der Architektur erwartet.

## 3. Analysephase

### 3.1. Ist-Analyse

Der Kunde der GSSD nutzt auf seinen Servern verschiedene Anwendungen als Dienste oder im Hintergrund. Um sicherzustellen, dass diese Anwendungen ständig funktionieren, wurde ein Überwachungssystem implementiert. Allerdings gibt es derzeit Probleme mit der Funktionalität des Systems, da es nur unvollständige Zustände anzeigt. Die Ist-Analyse ergab Schwachstellen im aktuellen Überwachungssystem, die verbessert werden müssen, um die Zuverlässigkeit und Funktionalität der überwachten Anwendungen sicherzustellen. Die Analyseergebnisse bilden die Grundlage für



die Planung und Umsetzung von Maßnahmen zur Verbesserung des Überwachungssystems.

### 3.2. Workflow

```
1  const EnergieTänzer = {
2    "sensor_id": "2",
3    "sensor_name": "Energie-Tänzer",
4    "sensor_type": "physical",
5    "data_type": "string",
6    "unit": "Tanzbewegungen",
7    "unit_display": "Tanzbewegungen",
8    "sensor_max": null,
9    "sensor_min": null,
10   "show": true,
11   "saveData": true,
12   "data": [],
13   "sensor_status": 'green',
14   "LastUpdated": new Date(),
15   "timestamp": Date.now(),
16   "expectedLast": 'Date',
17   "commendsql": "SELECT COUNT(*) FROM dance_moves WHERE energy > 10;"
18 }
```

Abbildung 2: Projektphasen mit Zeitplanung in Stunden

Der Sensor erfasst die Tanzbewegungen, die durch die Energie des Raums erzeugt werden, und speichert sie in der Datenbank. Außerdem hat er die Fähigkeit, eine SQL-Abfrage an die Datenbank zu senden, um die Anzahl der Tanzbewegungen zu zählen, bei denen eine bestimmte Menge an Energie erzeugt wurde.

Der Workflow für den Energie-Tänzer sieht folgendermaßen aus:

1. Der Sensor erfasst die Energie des Raumes und erzeugt Tanzbewegungen.
2. Die Tanzbewegungen werden in der Datenbank gespeichert und der Sensor aktualisiert seine Daten.
3. Die Daten werden dem Backend übermittelt und der Sensor kann entscheiden, ob er die Daten zum Frontend weiterleitet oder nicht.
4. Wenn eine SQL-Abfrage gesendet wird, sendet der Sensor die Abfrage an das Backend, das die Ergebnisse aus der Datenbank abrufen und an den Sensor zurücksendet.

5. Der Sensor aktualisiert seine Daten und ist bereit, weitere Tanzbewegungen zu erzeugen.

### 3.3. Soll-Analyse

Die Soll-Analyse beschreibt die Anforderungen an das Überwachungssystem, die der Kunde an das System stellt. Die Anforderungen an das Überwachungssystem sind in der Tabelle 2 aufgeführt.

Anforderung	Beschreibung
<b>Funktional</b>	
Diensteüberwachung	Überwachung der Dienste, die auf den Servern laufen
Alarmierung	Benachrichtigung im Falle von Störungen
Protokollierung	Aufzeichnung von Zustandsänderungen und Störungen
Registrieren	neue Sensor registrieren
<b>Nicht-Funktional</b>	
Benutzerfreundlichkeit	Einfache Bedienbarkeit des Systems
Zuverlässigkeit	Hohe Verfügbarkeit und Genauigkeit des Überwachungssystems
<b>Qualität</b>	
Skalierbarkeit	Möglichkeit zur Erweiterung des Systems
Flexibilität	Anpassung an unterschiedliche Anforderungen des Kunden
<b>Leistung</b>	
Echtzeitüberwachung	Echtzeitüberwachung der Zustände
Ressourcenschonung	Geringer Ressourcenverbrauch auf den überwachten Servern

Tabelle 1: Anforderungen an das Überwachungssystem

Die Überwachung aller relevanten Anwendungen auf den Servern des Kunden Eine schnelle und zuverlässige Erkennung von Fehlern oder Ausfällen Die Möglichkeit, Benachrichtigungen und Alarmer bei Fehlern oder Ausfällen zu generieren Eine übersichtliche Darstellung der überwachten Anwendungen und deren Zustände Eine einfache Konfiguration und Verwaltung des Überwachungssystems Zusätzlich sollten auch bestimmte Leistungsmerkmale definiert werden, um sicherzustellen, dass das Überwachungssystem den Anforderungen des Kunden entspricht. Hierzu können beispielsweise die maximale Anzahl der zu überwachenden Anwendungen, die Re-

aktionszeit bei Fehlermeldungen und die Verfügbarkeit des Überwachungssystems zählen.

### 3.4. Wirtschaftlichkeitsanalyse

#### 3.4.1. "Make or Buy" -Entscheidung

**TODO**

#### 3.4.2. Projektkosten

**TODO**

Es fallen neben Personalkosten auch die Kosten der verwendeten Hardware-Ressourcen an ,das heißt in diesem Fall die Abschreibung der Arbeitslaptops des Autors. Für das Projekt wird 80 Stunden Arbeitszeit veranschlagt. Für die Entwicklung wird der Lohn der Auszubildenden gerechnet mit 10,00 € pro Stunde angesetzt. Bei einigen Arbeiten war ein Mitarbeiter der Ferrari electronic AG beteiligt, für den 8 Stunden mit 25,00 € pro Stunde eingesetzt wird.

Für den Verschleiß des Laptops wird ein pauschalen Betrag von 40,00 € gerechnet.

Die Kosten lassen sich in der folgenden Tabelle zusammenfassen:

Vorgang	Zeit	Kosten pro Stunde	Kosten
Auszubildender	80 h	10,00 €	800,00 €
Mitarbeiter	8 h	25,00 €	200,00 €
Verschleiß			40,00 €
			<b>1040,00 €</b>

Tabelle 2: Kostenaufstellung

#### 3.4.3. Amortisationsdauer

**TODO**

### 3.5. Lastenheft

**TODO**

:

- **TODO**
- **TODO**
- **TODO**

## 4. Entwurfsphase

### 4.1. Zielplattform

#### **TODO**

Dieses Tool soll primär per Webbrowser bedienbar sein. Zusätzlich soll es erweiterbar als native Windows-, iOS- und Android-Applikation sein. Deshalb wurde für das Frontend das Framework Flutter die Programmiersprache gewählt. Sowohl Frontend als auch Backend wurden mit Hilfe von Visual Studio Code als Entwicklungsumgebung erstellt.

### 4.2. Architekturdesign

#### **TODO**

### 4.3. Datenbank

#### **Entscheidung Prozess :**

Entscheidung Prozess :

- **mysql -> Microsoft-server -> administrationaufwand.**
- **sql-lite -> mehr aufwand in der programmierung.**
- **nonsql datenbank -> einfach programmieren, kein Administrationsaufwand**
- **Mengenbetrachtung -> 100 Aufträge pro jahr sind nicht viel, dann ist das nicht notwendig einen Datenbank mit server zu bauen.**

**Der Go-Code für den Datenbankmodel ist im Anhang zu sehen.**

### 4.4. REST-API

#### **TODO**

### 4.5. Entwurf der Benutzeroberfläche

Während der Gestaltung der Benutzeroberfläche wurden zunächst einige Mockups erstellt (siehe Anhang für das Web-Frontend-Mockup). Dabei wurde darauf geachtet, dass die Bedienoberfläche bedienbar ist und ansprechend aussieht.

### 4.6. Maßnahmen zur Qualitätssicherung

#### TODO

Es wurde Unit- und Widget-Tests zu schreiben, mithilfe derer die Korrektheit der Implementierung überprüft werden kann.

Außerdem wurden regelmäßige Rücksprachen mit dem Leiter der Entwicklungsabteilung geplant, um frühzeitig ungewollte Abweichungen vom Ziel zu vermeiden.

Sämtlicher Quellcode der Software wurde unter die GIT-Versionsverwaltung gestellt, um zu jedem späteren Zeitpunkt die Historie der Softwareentwicklung nachvollziehen zu können. Es wurde eine Continuous Integration (CI) mithilfe von Jenkins realisiert, die nach jedem Push in das GIT-Repository automatisch die Software baut und die Unit- und Widget-Tests ausführt.

### 4.7. Deployment

#### TODO

### 4.8. Pflichtenheft

#### TODO

Auf der Basis von Lastenheft und im Hinblick auf die vorhandenen Ressourcen wurde ein Pflichtenheft erstellt. Dieses enthielt die präzisierten Anforderungen des Lastenheftes und deren geplanten Umsetzung, siehe im Anhang Abbildung .

## 5. Implementierungsphase

### 5.1. Vorbereitung des Projektes

#### TODO

### 5.2. Implementierung des Backendes

TODO

### 5.3. Implementierung des Frontends

TODO

### 5.4. Testen der Anwendung

TODO

## 6. Abnahmephase

### 6.1. Code-Review

Ein Code-Review wurde zuletzt durchgeführt, hier geht es darum, ob die Namen der Funktionen und Variablen richtig und sinnvoll geschrieben sind, ob alle Tests geschrieben, die notwendig sind, ob die Oberfläche schön aussieht und einfach zu bedienen ist und schließlich ob die Funktionalität wie erwünscht funktioniert.

### 6.2. Abnahme

Die Abnahme war kurz, weil eine Rücksprache mit dem Entwicklungsleiter (der Auftragsgeber) immer stattfand .

## 7. Dokumentation

latex parallel

## 8. Fazit

### 8.1. Soll-Ist-Vergleich

Durchs Vergleich zwischen dem Produkt was am Ende kommt und dem Lasten Heft, kann man sagen, dass der Produkt alle erwünschten Kriterien erfüllt.

## **A. Anhang**

### **A.1. Pflichtenheft**

Auf Basis von Lastenheft und im Hinblick auf die vorhandenen Ressourcen entstanden die folgenden Punkten im Pflichten Heft.

1. Plattform
  - a) Zur Entwicklung wird Visual Studio Code als Integrierte Entwicklungsumgebung genommen.
  - b) Damit das Projekt automatisch gebaut und getestet wird, wird Jenkins gewählt.
  - c) Für die Dokumentation wird Latex benutzt.
  - d) Nginx reverse proxy wird für die Verbindung zwischen den Client und den Server benutzt.
  - e) Github und Gitlab als Git Plattform
2. Datenbank
  - a) Ein SQL-lite Datenbank wird erstellt.
3. Oberfläche
  - a) Wird mit VueJs Programmiert
  - b) Soll Web, HandyApp und PC-App sein.
4. Backend
  - a) NodeJs wird als Programmiersprache benutzt