Ein Bild, das Text, ClipArt enthält.

Automatisch generierte Beschreibung

**Portierung der Konfigurationsmaske von Ergo**

Elias Mattern  
10.02.2023  
Informatik Entwicklung J2EE  
Coop Genossenschaft

# Versionierung



Tabelle 1: Versionsverlauf

# Kurzfassung

Inhaltsverzeichnis

[Versionierung 1](#_Toc128473496)

[Kurzfassung 2](#_Toc128473497)

[Teil I: Obligatorische Kapitel 5](#_Toc128473498)

[1 Detaillierte Aufgabenstellung 5](#_Toc128473499)

[1.1 Ausgangslage 5](#_Toc128473500)

[1.2 Beschreibung der Aufgabe 5](#_Toc128473501)

[1.3 Abgrenzungen 5](#_Toc128473502)

[1.4 Anforderungen 5](#_Toc128473503)

[1.5 Erwartungen 5](#_Toc128473504)

[1.6 Tests 5](#_Toc128473505)

[1.7 Nutzen für Coop 5](#_Toc128473506)

[1.8 Dokumentation 5](#_Toc128473507)

[1.9 Grober, Provisorischer Zeitplan 6](#_Toc128473508)

[1.10 Projektmethode 7](#_Toc128473509)

[1.11 Vorgesehene Meilensteine 7](#_Toc128473510)

[1.12 Mittel und Methoden 7](#_Toc128473511)

[1.13 Vorkenntnisse 7](#_Toc128473512)

[1.14 Vorarbeiten 7](#_Toc128473513)

[1.15 Neue Lerninhalte 7](#_Toc128473514)

[1.16 Arbeiten in den letzten 6 Monaten 7](#_Toc128473515)

[2 Projektaufbauorganisation 8](#_Toc128473516)

[3 Zeitplan 9](#_Toc128473517)

[3.1 Soll/Ist Vergleich 10](#_Toc128473518)

[4 Schlusswort 11](#_Toc128473519)

[Teil II: Projektdokumentation 12](#_Toc128473520)

[1 Projektmanagement 12](#_Toc128473521)

[2 Organisation der Arbeitsergebnisse 12](#_Toc128473522)

[3 Versionsverwaltung 12](#_Toc128473523)

[4 Softwarearchitektur 12](#_Toc128473524)

[5 Systemumgebung und Schnittstellen 12](#_Toc128473525)

[6 Lösungsevaluation 12](#_Toc128473526)

[7 Technologien 12](#_Toc128473527)

[8 Use Case Diagramm 12](#_Toc128473528)

[9 Entity Relationship Model 12](#_Toc128473529)

[10 Testkonzept 13](#_Toc128473530)

[11 Testen mit Postman 13](#_Toc128473531)

[12 Unit Tests 13](#_Toc128473532)

[13 Funktionale Tests 13](#_Toc128473533)

[14 Testprotokoll 13](#_Toc128473534)

[15 Technische Dokumentation 13](#_Toc128473535)

[Anhang 14](#_Toc128473536)

[1 Quellenverzeichnis 14](#_Toc128473537)

[2 Tabellenverzeichnis 15](#_Toc128473538)

[3 Abbildungsverzeichnis 16](#_Toc128473539)

[4 Glossar 17](#_Toc128473540)

Teil I: Obligatorische Kapitel

# Detaillierte Aufgabenstellung

## Ausgangslage

Zurzeit kann ein Admin Benutzer mithilfe einer Adminseite, Parameter setzen, bearbeiten oder löschen. Um die Seite benutzerfreundlicher zu machen, soll diese überarbeitet werden. Ziel dieser IPA ist es, die Seite in Angular zu portieren, um die Anwendung zu vereinfachen und zu verbessern.

Im Ergo (Mobiles Warenwirtschaftssystem der Coop Verkaufsstellen) soll der Parametrisierungsprozess portiert werden. Das Warenwirtschaftssystem ist in viele Prozesse (Arbeitsbereiche) aufgeteilt. Der Prozess der Parametrisierung dient dazu, Parameter zu setzen und zu bearbeiten. Durch die Parameter können verschiedene Aspekte des Systems konfiguriert werden. Da nur Ergo-Administratoren Parameter erfassen oder ändern dürfen, muss sichergestellt werden, dass nur berechtigte Benutzer auf die Webmaske zugreifen können. Da die alte Seite unübersichtlich geworden ist, soll diese von JavaServer Faces (JSF) in Angular portiert werden. Dabei sollen die Funktionen übernommen und verbessert werden.

Die IPA wird in einem separaten Featurebranch durchgeführt, so dass weder das Ergo Team Einfluss auf die IPA hat, noch die IPA Einfluss auf die Arbeit des Ergo Teams. Der Featurebranch ist mit der identischen Pipeline ausgerüstet wie der Master-Branch, exkl. Deployment auf Testsystem. Die IPA wird am Hauptsitz von Coop in Basel stattfinden.

Die Laufzeitumgebung für die IPA ist WSL2/Ubuntu. Application/Webserver ist embedded tomcat 9, d.h. dieser ist bereits Bestandteil des Projektes/Codes und brauchen keine separate Installation/Konfiguration.

Die Datenbank ist Oracle. Notwendige Schemas für die Umsetzung der IPA sind bereits vorhanden, bzw. können jederzeit erstellt werden (1 Schema als Backend für die Applikation, kann mit produktiven Daten gefüllt werden, 1 Schema für Integrationstests).

## Beschreibung der Aufgabe

Für die Umsetzung des Parameterprozesses muss eine neue Angular Maske "Parameter" angelegt werden. Auf dieser Seite wird eine Tabelle mithilfe von AG-Grid erstellt. Die Tabelle wird alle Parameter beinhalten. Dabei werden folgende Spalten erstellt:

* Name des Parameters.
* Technischer Name (Name des Parameters im System).
* Kategorie (Bereich der Applikation, für welchen der Parameter eingesetzt wird).
* Typ (Zeigt den Datentyp des Parameters an).
* Einsatz (Zeigt auf, für welches System der Parameter verwendet wird).
* Kontextattribute (Zusätzliche Werte, welche ein Parameter hat).
* Standardwerte des Parameters.

Die Angaben werden bei der Dokumentation der Implementation erläutert.

Für die Tabelle wird ein Endpunkt erstellt, welcher alle Parameter sucht und zurückliefert.

### Was ist ein Parameter

Ein Parameter wird verwendet, um das Ergo zu konfigurieren. Dabei können die Administratoren im Ergo diese Parameter setzen und bearbeiten. Die Parameter können unterschiedliche Aspekte der Applikation steuern. Ebenfalls können die Parameter unterschiedliche Einstellungen haben. Somit kann das Ergo für die verschiedenen Verkaufsstellen unterschiedlich konfiguriert sein. Um diese Einstellungen einsehen zu können, wird es eine Untertabelle geben.

### Untertabelle

Um die Einträge eines Parameters einsehen zu können, wird es für jeden Parameter in der Tabelle eine aufklappbare Untertabelle geben. Um die Daten der Untertabelle zu laden, wird ebenfalls ein Endpunkt erstellt. Die Untertabelle wird folgende Spalten beinhalten:

* Gültig von (Startdatum des Eintrages)
* Gültig bis (Enddatum des Eintrages)
* Modus (Zeigt, wer von der Einstellung betroffen ist)
* Modus-Wert (Zeigt im Detail an, wer von der Einstellung betroffen ist)
* Kontextattribute (Spezifische Einstellungen des Parameters)
* Wert des Parameters

Dieser Endpunkt holt alle Einträge für den ausgewählten Parameter. Bei dieser Untertabelle wird es möglich sein, die einzelnen Einträge zu löschen. Dabei kann jeweils nur eine Zeile gelöscht werden. Es können keine neuen Einträge erstellt oder aktualisiert werden.

### Filtern und sortieren

Alle Spalten, welche Texte beinhalten, sollen in der Haupt- und Untertabelle sortierbar sein. Dafür gibt es unter der Spalte ein Eingabefeld, mit welchem nach Einträgen gesucht werden kann.

### Übersetzungen

Alle Spaltenüberschriften, sowie alle Texte der Autovervollständigung müssen in drei Sprachen (Deutsch, Französisch und Italienisch) übersetzt sein. Die Daten in der Tabelle und die möglichen Werte in der Autovervollständigung werden vom Backend geladen, wodurch diese im Frontend nicht übersetzt werden können. Für alle Icons soll ein übersetzter Tooltip existieren, damit für den Benutzer klar ist, was beim Klick auf das Icon passiert.

### Parametersuche

Zusätzlich wird ausserhalb der Tabelle eine Suche erstellt. Dadurch kann ein Benutzer einen Parameter nach seinem Namen oder technischen Namen suchen. Um die Suche zu vereinfachen, wird die Suche als Autovervollständigung erstellt, wodurch während der Eingabe Vorschläge der verfügbaren Parameter angezeigt werden. Ein Parameter wird in der Suche vorgeschlagen, falls der Name oder der technische Name des Parameters den eingegebenen Wert beinhaltet. Sobald ein Parameter in der Autovervollständigung ausgewählt wurde, wird der Name des Parameters im Eingabefeld angezeigt und die Spalten "Name" und "techn. Name" in der Tabelle gemäss der Auswahl in der Autovervollständigung gefiltert.

Für diese Vorschläge wird ein Endpunkt benötigt, welcher die Vorschläge lädt.

### Validierung

Die Benutzereingaben müssen folgenderweise validiert werden:

* Alle Felder sind optional
* Die Parametersuche hat eine Maximallänge von 255 Zeichen
* Die Vorschläge in der Autovervollständigung sollen erst nach der Eingabe von drei Zeichen angezeigt werden
* Wenn ein Suchbegriff in der Autovervollständigung eingegeben, jedoch kein Eintrag ausgewählt wurde, soll ein Fehlertext angezeigt werden. Der Fehlertext wird erst angezeigt, wenn der Benutzer das Suchfeld verlässt.
* Die Eingaben der Datumsfilter müssen überprüft werden, dass ein Datum (inkl. Zeit) eingegeben wurde. Ansonsten soll ein Fehlertext angezeigt werden.

### Authentifizierung

Für die Authentifizierung gibt es bereits den Identity Provider (IDP). Mithilfe dieses IDPs kann der Benutzer beim Aufrufen einer Schnittstelle identifiziert werden. Somit kann seine Rolle und Berechtigungen überprüft werden. Die Schnittstellen müssen dadurch geschützt werden, damit nur berechtigte Benutzer diese verwenden können. Die Logik, durch welche die Rechte eines Benutzers ausgelesen werden können, ist bereits implementiert. Im Frontend existiert ein AppGuard. Mit diesem kann die Seite geschützt werden. Er entscheide, ob ein Benutzer die Berichtigung hat, die Seite aufzurufen.

## Abgrenzungen

Im Rahmen der IPA wird die Seite für die Parametrisierung in Angular erstellt. Dabei werden jedoch nicht alle Funktionen umgesetzt werden können. Somit ist das Erstellen so wie das Bearbeiten der Einträge von Parametern nicht Teil der IPA. Dies wird nach der IPA umgesetzt werden. Grund dafür ist, dass alle Parameter unterschiedlich aufgebaut sind. Somit kann die Erstellung der Einträge für jeden Parameter unterschiedlich sein.

## Anforderungen

## Erwartungen

## Testen

Während der IPA wird die Umsetzung getestet werden. Dafür wird ein Testkonzept geschrieben, welches aufzeigt wie beim Testen vorgegangen wurde.

Während der IPA wird der Store, welcher für die Seite benutzt wird, getestet werden. Für diesen werden entsprechende Unit Tests geschrieben. Diese Tests werden automatisch beim Check-In des Codes ausgeführt. Zusätzlich werden alle Schnittstellen manuell mit Postman getestet. Für alle ausgeführten Tests wird es ein Testprotokoll geben. Dieses wird alle Tests aufzeigen und die Resultat der Tests zeigen

## Nutzen für Coop

Durch die Portierung des Parametrisierung-Prozess wird die Arbeit mit den Parametern für die Admins von Ergo vereinfacht. Durch die Einsetzung von modernen Technologien

## Dokumentation

IPA Bericht:  
Der IPA Bericht wird in zwei Teilen aufgeteilt. Zu Beginn kommt die Projektdokumentation. Dort wird der Auftrag, die Planung sowie das Vorgehen beschrieben. Im zweiten Teil wird die Implementierung dokumentiert. Dort werden das Projektmanagement, die Technologien und das Vorgehen der IPA beschrieben.

Im Anhang wird sich das Umsetzungskonzept sowie die technische Dokumentation für die Teammitglieder befinden.

Zusätzliche Dokumente:  
Im IPA Bericht wird sich kein Code befindet. Dieser wird zusätzlich in einer ZIP Datei abgegeben. Ebenfalls gibt es ein zusätzliches Dokument für die Arbeitsjournale.

## Grober, Provisorischer Zeitplan



## Projektmethode

Die IPA wird mit der Wasserfallmethode umgesetzt. Dieses Modell wurde zusätzlich mit einem Rücksprung erweitert. Dies ermöglicht es, von einer Phase in die vorherige zu wechseln. Das ermöglicht bei der Umsetzung, Probleme schnell zu beheben.

## Vorgesehene Meilensteine

* **Seite erstellt, autorisiert und Parameter und deren Einträge werden angezeigt**

Beim Erreichen des ersten Meilensteines soll die Seite mit den Tabellen erstellt sein. Dabei werden alle Parameter geladen.

* **Parameter können gesucht und im Autocomplete ausgewählt werden**

Nach Abschluss des zweiten Meilensteines sollte der Benutzer mithilfe eines Eingabefeldes nach Parametern suchen können.

* **Einträge können aus der Untertabelle gelöscht werden**

Die Seite soll beim dritten Meilenstein bereits abgeschlossen sein. Somit wird noch die Funktion hinzugefügt, um Einträge aus der Untertabelle löschen zu können.

* **IPA ist fertig und bereit für Abgabe**

Der letzte Meilenstein ist die Fertigstellung der Dokumentation und die Abgabe.

Die Meilensteine spiegeln jeweils die Funktionen der Seite wider. Dabei wurde geschaut, dass die Funktionen vollendet sind und bereits ohne die folgenden Meilensteine funktionstüchtig sind.

## Mittel und Methoden

Für diese IPA wurden folgende Mittel und Methoden eingesetzt.

Hardware:

* Coop-Entwicklerclient mit Windows 10

Programme:

* IntelliJ als Entwicklungsumgebung
* WSL (Windows-Subsystem für Linux)
* Oracle SQL Developer
* Git für die Versionierung

Backend:

* Java
* Gradle für Build und Dependencies
* Spring für Rest-Controller

Frontend:

* Angular
* TypeScript
* NgRx für ein reaktives State-Management im Frontend
* OpenAPI für die Generierung der Schnittstellen zwischen Front- und Backend

Im Abschnitt [*Technologien*](#_Technologien) werden die verschiedenen Programme und Technologien im Detail beschrieben.

## Vorkenntnisse

Seit August 2021 arbeite ich im Team J2EE am Ergo Projekt, wodurch ich die Projektstruktur kenne. Seitdem konnte ich viele Kenntnisse zu Angular und Java sammeln und habe schon einiges für das Projekt umgesetzt.

## Arbeiten in den letzten 6 Monaten

Bei der Arbeit im J2EE Team habe ich die Navigation, einen Tip of the Day (Tipps zur Verwendung von Ergo) und die Portierung des Artikelcockpits (Anzeige von Artikelinformationen) in Angular umgesetzt. Für die Umsetzung dieser Aufgaben wurden jeweils Konzepte geschrieben.

## Vorarbeiten

Vor der IPA wurden bereits einige Vorarbeiten erledigt. Diese waren nötig, um mit der IPA starten zu können

Umsetzungskonzept:  
Für die Umsetzung wird ein Konzept benötigt. Dieses beschreibt die Aufgabe und das Ziel der Arbeit. Das Umsetzungskonzept musste freigegeben werden, damit die Arbeit durchgeführt werden durfte. Diese Freigabe erfolgte vor der IPA. Dabei wurden auch Mockups erstellt.

Datenbankschemas aktualisieren:  
Am Tag vor der IPA wurden aktuelle Daten in die Datenbankschemas geladen, damit bei der IPA die richtigen Daten vorhanden sind.

Ergo Libraries:  
Ergo Libraries ist eine eigene Library für das Projekt Ergo. Um Konflikte nach der IPA zu verhindern, wurde diese bereits vor der IPA erweitert. Die Library beinhaltet unter anderem eine Liste der portierten Prozesse. Anhand dieser Liste kann entschieden werden, ob die JSF oder die Angular Seite geladen wird. In der Library wurde der Prozess der Parametrisierung zu dieser Liste hinzugefügt.

Da bereits eine Codebasis und ein Webserver existiert musste nichts installiert/konfiguriert werden.

## Neue Lerninhalte

Die IPA beinhaltet einige neue Lerninhalte.

* Verwendung von AG-Grid zur Erstellung einer Tabelle.
* Untertabellen erstellen.
* Suche mit Autocomplete erstellen.
* Erstellung einer Suche, die über mehrere Spalten filtern kann.

Die Erfahrungen im J2EE werden mir helfen diese neuen Lerninhalte anzugehen.

# Projektaufbauorganisation

Der Auftraggeber sowie die verantwortliche Fachkraft der Arbeit ist Patrick Maurer. Zusammen mit ihm wurde die Umsetzung für die Portierung des Parametrisierung-Prozesses ausgearbeitet. Er unterstütze mich beim Erstellen des Umsetzungskonzepts und bereitete mich auf die Arbeit vor. Ebenfalls involviert ist der Hauptexperte Jäggi Dominique und der Nebenexperte Dominik Müller, welche vor der IPA und während der IPA für Fragen zur Verfügung stehen und die Arbeit gemeinsam mit Patrick Maurer beurteilen. Im Projekt war zusätzlich Anita Hartmeyer und Hans-Michael Ruckstuhl von der Abteilung Warenwirtschaft.



# Zeitplan

Für die IPA wurde ein detaillierter Zeitplan erstellt. Die Basis für diesen Zeitplan war der grobe Zeitplan der Detailbeschreibung. Für die Umsetzung des Zeitplanes mussten die Meilensteine jeweils in Arbeitsschritte aufgeteilt werden. Dafür wurde analysiert, welche Schritte für den Meilenstein erforderlich sind. Dabei war es wichtig zu wissen welche Aspekte der Applikationen erweitert werden müssen oder welche Funktionen neu sind. Für die jeweiligen Arbeitsschritte wurde der Aufwand geschätzt. Dadurch konnte ein Soll / Ist Zeitplan erstellt werden.

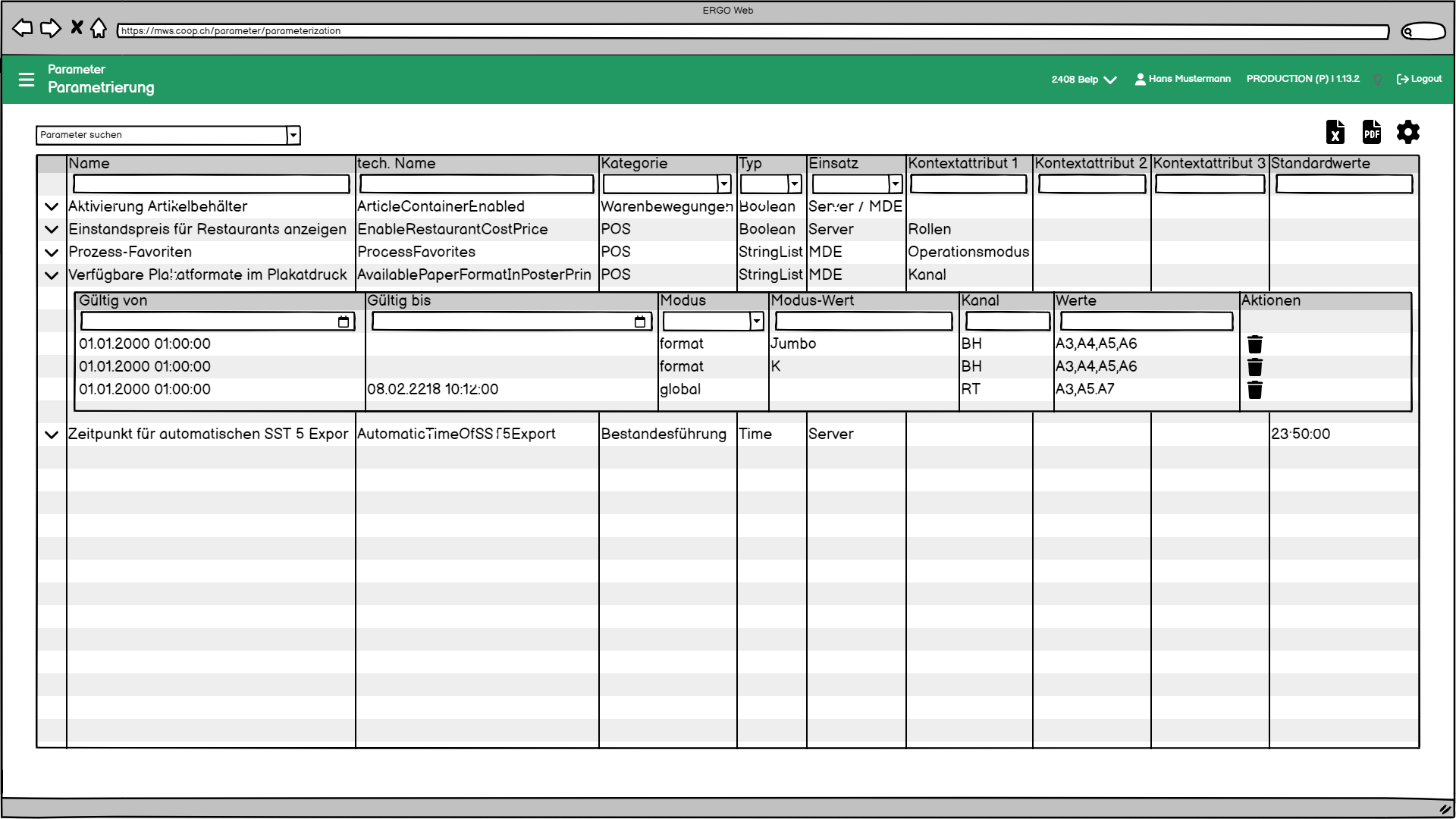
## Soll/Ist Vergleich



# Mockups

Für die Angular Seite wurden vor der IPA einige Mockups erstellt. Diese helfen, die Aufgabe zu verstehen und die Seite umzusetzen. Auf den Mockups sind die einzelnen Komponenten und Funktionen, welche implementiert werden, ersichtlich.

Dieses Mockup zeigt auf, wie die Seite aufgebaut wird. Dabei sieht man alle Spalten der Haupt- und Untertabelle. Oben Links ist zusätzlich noch die Suche ersichtlich. Gegenüber der Suche befinden sich drei Buttons. Zum einen kann man dort die Tabelle als PDF oder Excel herunterladen, andererseits befinden sich dort die Einstellungen der Tabelle. Diese werden gebraucht, um die Tabelle einzustellen oder als PDF zu speichern. In der Untertabelle finde man die Buttons, um Einträge aus der Tabelle zu löschen.



Weitere Mockups sind auf PkOrg zu finden. Diese zeigen die einzelnen Funktionen im Detail.

# Schlusswort

Teil II: Projektdokumentation

# Projektmanagement

Im Rahmen dieses Projektes wurde eine Projektmanagement-Methode angewandt. Diese Methode ist das Wasserfallmodell. Für die Umsetzung wurde dieses Modell erweitert, um einen besseren Arbeitsablauf zu ermöglichen.

## Wasserfallmodell

Das Wasserfallmodell ist ein lineares Modell. Dabei wird das Projekt in verschiedene Phasen unterteilt. Diese Phasen repräsentieren meist die grossen Meilensteine eines Projektes. Nachdem eine Phase abgeschlossen wurde, wird in die nächste Phase übergegangen. Das Wasserfallmodell gibt klare Vorgaben zum Ablauf des Projektes und ist einfach strukturiert. Durch die klare Struktur ist es möglich das Projekt gut zu kontrollieren. Das Wasserfallmodell ist ein lineares Modell, was bedeutet, dass es nicht möglich ist, von einer Phase zurückzugehen. (tutorialspoint, kein Datum). Dies ist ein Nachteil, da beim Programmieren oftmals erst bei der Umsetzungen Probleme klar werden und bei der Planung nicht alles berücksichtigt werden kann. Dadurch ist es nicht möglich, die Planung nochmals zu überarbeiten.

## Rücksprung

Da es beim Wasserfallmodell nicht möglich ist in eine vorherige Phase zurückzukehren, könnte dies zu Problemen bei der Umsetzung führen. Um dem Problem des Wasserfallmodells zu entgehen, wurde dieses um einen Rücksprung erweitert. Dieser Rücksprung ermöglicht es, in jeder Phase des Projektes wieder in die vorherige Phase zurückzukehren. (Wikipedia, 2023). Das folgende Modell zeigt das Wasserfallmodell mit Rücksprung auf, welches für dieses Projekt angewendet wurde:

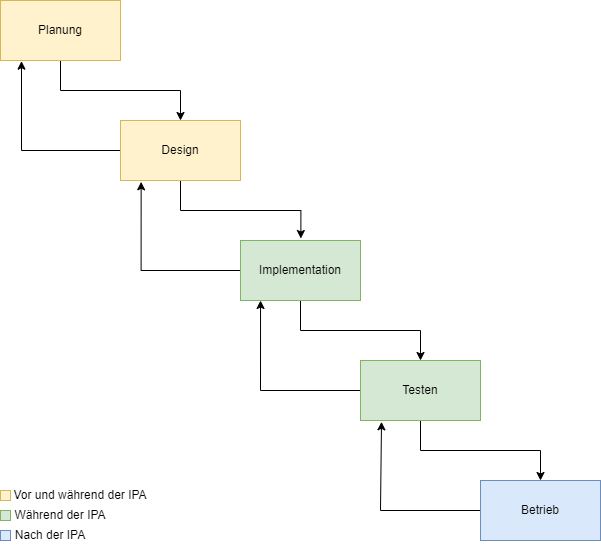


Abbildung 1: Wasserfallmodell mit Rücksprung

### Planung

Im ersten Schritt wird die Umsetzung für die IPA geplant. Dafür wurde ein Umsetzungskonzept geschrieben und genehmigt. Ebenfalls wurde ein grober Zeitplan erstellt, welcher den Ablauf in kürze aufzeigt. Während der IPA wird dieser Plan detaillierter aufgebaut und das Vorgehen geplant.

### Design

Zum Abschnitt Design gehören die Mockups. Diese wurden bereits vor der IPA umgesetzt. Ebenfalls wird es ein ERM geben und Modell für die Systemumgebung. Diese werden während der IPA erstellt.

### Implementation

In dieser Phase beginnt die Umsetzung. Anhand des Umsetzungskonzeptes soll die Implementation durchgeführt werden. Bei allfälligen Problemen kann jederzeit ein Rücksprung erfolgen, um die Planung anzupassen.

### Testen

Der Store und die Schnittstellen werden getestet. Dafür werden Testfälle beschrieben und ausgeführt. Sobald Fehler bei den Tests auftreten, wird ein Rücksprung auf die Implementation erfolgen. Dies ermöglicht, die Fehler zu korrigieren.

### Betrieb

Die letzte Phase tritt nach dem Projekt ein. In dieser Phase wird die Seite gewartet und verbessert. Ebenfalls wird die Seite noch erweitert werden. So wird noch das Erstellen und Bearbeiten von Parametereinträgen implementiert.

# Organisation der Arbeitsergebnisse

Das Ergo Projekt ist auf dem GIT von der Firma Ergon gesichert. Alle Änderungen werden auf diesem gespeichert. Alle Dokumente für die IPA liegen auf dem Coop One Drive. Die Ordnerstruktur ist wie folgt aufgebaut:

Im Hauptverzeichnis befindet sich ein GIT Repository. In diesem befindert sich der IPA Bericht und einige Unterordner:

**/Anhang**

Dieses Verzeichnis wird alle Dokumente für den Anhang beinhalten. Dabei gibt es eine zusätzliche Word Datei, welche den Anhang beinhaltet.

**/Modelle**

Hier werden alle Modelle für die IPA abgelegt.

**/pkorg**

Alle Dateien, welche sich auf dem PkOrg befinden, sind in diesem Verzeichnis abgelegt. Somit wird sichergestellt, dass jederzeit auf diese zugegriffen werden kann.

**/Tag X**

Für jeden Arbeitstag wird es einen Ordner geben. Dieser beinhaltet das Arbeitsjournal für den Tag

**/Vorarbeiten**

Die erledigten Vorarbeiten befinden sich in diesem Ordner.

**/Vorlagen**

In diesem Verzeichnis werden alle Vorlagen abgelegt, welche in der IPA benutzt werden.

# Versionsverwaltung

Während der IPA muss sichergestellt werden, dass die Daten nicht verloren gehen können. Dafür werden alle Daten versioniert. Die Versionierung erlaubt es, ältere Versionen wiederherzustellen. Somit kann bei Fehlern wieder auf eine funktionsfähige Version zugegriffen werden. Alle Daten für das Ergo Projekt werden bei der Ergon gesichert und versioniert. Dafür wird GIT benutzt. Die Dokumente der IPA werden auf dem Coop One Drive gespeichert und versioniert. Zusätzlich werden die Daten auf einem privatem GIT Repository und einem privaten Google Drive gespeichert. Somit sind die Daten nicht nur an einem Ort gespeichert. Das private GIT Repository wird benutzt, um auf ältere Versionen zuzugreifen. Dafür wird jeden Tag die neuste Version gepusht.

## Zugriff auf ältere Versionen

### IntelliJ

In IntelliJ gibt es eine Übersicht aller Versionen des Ergos. Um auf eine ältere Version zugreifen zu können, kann in dieser Übersicht die gewünschte Version ausgewählt werden. Mit der Option "Reset Current Branch to Here" kann zur ausgewählten Version gewechselt werden. Somit konnte während der IPA jederzeit wieder auf einen alten Stand zugegriffen werden.

### Dokumentation

# Technologien

Programme:

* IntelliJ als Entwicklungsumgebung
* WSL (Windows-Subsystem für Linux)
* Oracle SQL Developer
* Git für die Versionierung

Backend:

* Java
* Gradle für Build und Dependencies
* Spring für Rest-Controller

Frontend:

* Angular
* TypeScript
* NgRx für ein reaktives State-Management im Frontend
* OpenAPI für die Generierung der Schnittstellen zwischen Front- und Backend

# Softwarearchitektur

# Use Case Diagramm

# Systemumgebung und Schnittstellen

# Entity Relationship Model (ERM)

Die Daten der Parameter liegen in einer Oracle Datenbank. Dort befinden sich vier Tabellen für die Parameter. Um die Beziehungen der Tabellen zu verstehen, wurde ein ERM der Tabellen entworfen. Ein ERM zeigt die Tabellen und ihre Beziehungen auf. Dabei besteht ein ERM aus verschieden Aspekten:

* Entitäten
* Primärschlüssel und Fremdschlüssel
* Beziehungen und Kardinalitäten

## Entitäten

Im ERM werden alle Entitäten der Tabelle aufgezeigt. Eine Entität ist eine Spalte der Tabelle. Dabei wird aufgezeigt, welche Datentypen die einzelnen Spalten beinhalten.

## Primärschlüssel und Fremdschlüssel

In der Tabelle ist jeweils eine Entität, welche den Primärschlüssel repräsentiert. Ein Primärschlüssel ist ein eindeutig identifizierbarer Wert. Oftmals ist dies die "ID" einer Zeile. Mit dem Primärschlüssel kann eine Zeile identifiziert werden. Dies wird verwendet, um spezifische Daten von der Datenbank abzufragen.

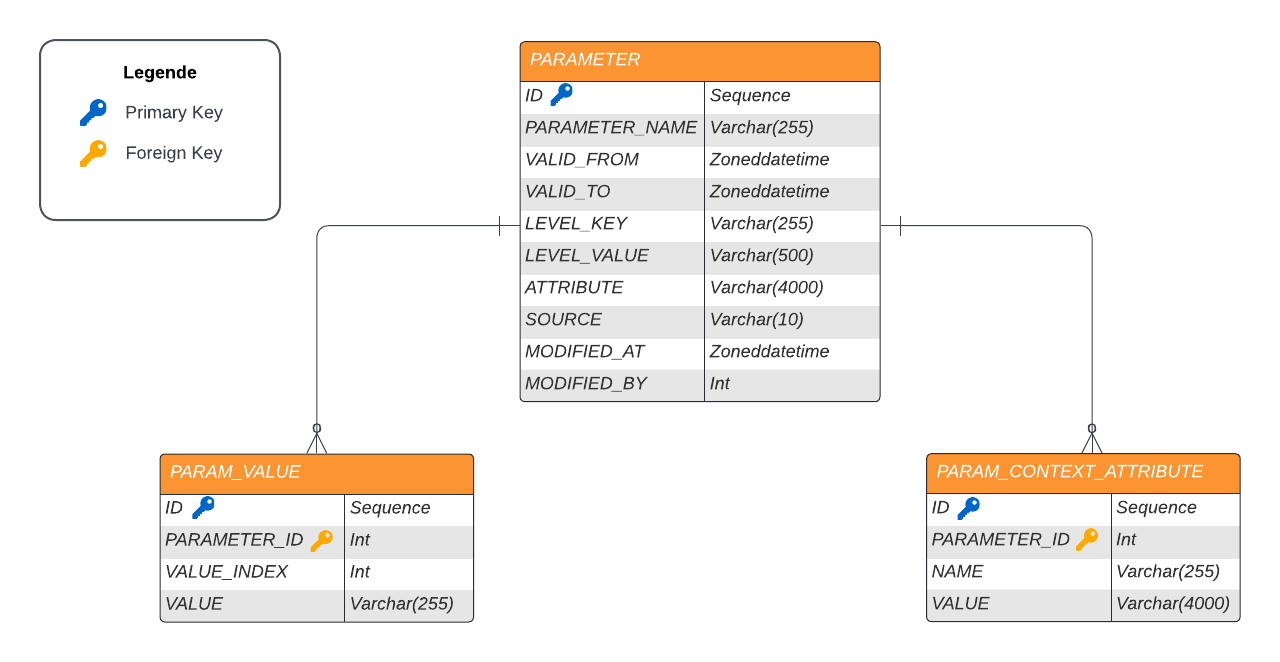
Ein Fremdschlüssel zeigt eine Beziehung zu einer anderen Tabelle innerhalb einer Datenbank an und wird als Referenz verwendet. Dabei hat der Fremdschlüssel den Wert des Primärschlüssel von der anderen Tabelle. Mithilfe des Fremdschlüssels kann die Zeile in der Tabelle gefunden werden, auf die sich der Fremdschlüssel bezieht. Somit können Tabellen miteinander verknüpft werden und Abfragen über mehrere Tabellen hinweg ausgeführt werden.

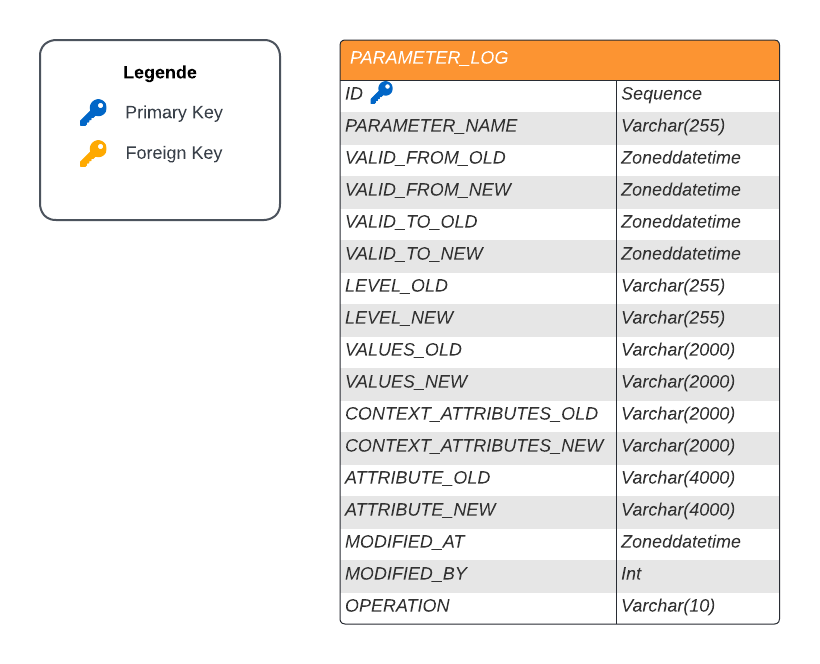
## Beziehungen und Kardinalitäten

In einem ERM werden die Beziehungen der Tabellen mit Hilfe von Kardinalitäten dargestellt. Diese zeigen auf, welche Tabellen in Verbindung miteinander sind. Dabei gibt es Verschiedene Kardinalitäten. Eine Kardinalität besagt, wie viele Entitäten von einer anderen Entität existieren können. (Indeed Editorial Team, 2023). Die folgenden Kardinalitäten existieren:

* 1:1 (Eine Entität hat eine Beziehung zu einer anderen Entität).
* 1: n (Eine Entität kann mehrere Beziehungen haben).
* n: m (Mehrere Entitäten können mehre Beziehungen haben).

## ERM der Parameter Tabellen





Wie im ERM zu sehen, gibt es die Tabellen Parameter, Param\_Value, Param\_Context\_Attribute und Parameter\_Log. Wie man direkt sieht, gibt es keine beziehungen zur Paramerter\_Log Tabelle. Das liegt daran, dass diese nur dafür benötigt wird, die Interaktionen mit dem Parameter zu loggen. Zum Beispiel wird ein Eintrag erstellt, sobald ein Eintrag aus der Tabelle Param\_Value gelöscht wird. Dafür werden jedoch die anderen Tabellen nicht benötigt.

# Technische Dokumentation

# Testkonzept

# Testen mit Postman

# Unit Tests

# Funktionale Tests

# Testprotokoll

Anhang

# Quellenverzeichnis

Wikipedia. (9. Februar 2023). *Wasserfallmodell*. Abgerufen am 09. März 2023 von Wikipedia: https://de.wikipedia.org/wiki/Wasserfallmodell#:~:text=Erweiterungen%20des%20einfachen,Erkenntnissen%20beim%20Testen.

# Tabellenverzeichnis

[Tabelle 1: Versionsverlauf 1](#_Toc127869543)

# Abbildungsverzeichnis

[Abbildung 1: Wasserfallmodell mit Rücksprung 18](#_Toc129255906)

# Glossar