Detailbeschreibung

# 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 sie 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 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).

# Aufgabenstellung

Für die Umsetzung des Parameterprozesses muss eine neue Angular Maske "Parameter" angelegt werden. Auf dieser Seite wird eine Tabelle mit Hilfe von AG-Grid erstellt. Für die Tabelle wird ein Endpunkt erstellt, welcher alle Parameter sucht und zurückliefert.

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. Dieser Endpunkt holt alle Einträge für den geöffneten 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.

Alle Spalten in der Haupt- und Untertabelle sollen sortierbar sein.

Alle Spaltenüberschriften, sowie alle Texte der Autovervollständigung müssen 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.

Zusätzlich wird ausserhalb der Tabelle eine Suche erstellt. Dadurch kann ein Benutzer einen Parameter nach seinem Namen und technischen Namen (Name des Parameters im System) suchen. Um die Suche zu vereinfachen, wird die Suche als Autovervollständigung erstellt, wodurch während der Eingabe Vorschläge der verfügbaren Parameter angezeigt werden. Damit ein Parameter in der Liste der Autovervollständigung erscheint, reicht es, wenn dessen Name oder technischer Name den eingegebenen Wert beinhaltet (keine Platzhalter nötig). Sobald ein Parameter in der Autovervollständigung ausgewählt wurde, wird der Name des Parameters in 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.  
Die Benutzereingaben müssen folgenderweise validiert werden:

* Alle Felder sind optional
* Die Suche 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 ausserhalb des Feldes klickt.
* Die Eingaben der Datumsfilter müssen überprüft werden, dass ein Datum (inkl. Zeit) eingegeben wurde. Ansonsten soll ein Fehlertext angezeigt werden.

Beim Testen wird der Store, welcher für die Seite benutzt wird, mittels Unit Tests getestet, welche automatisiert beim Code Check-In vom Jenkins ausgeführt werden. Zusätzlich werden alle Schnittstellen manuell mit Postman getestet. Für diese Tests muss ein Testprotokoll erstellt werden und die Ergebnisse aus allen Unit- und Postman-Tests müssen dokumentiert werden.

Für die Authentifizierung gibt es bereits den Identity Provider (IDP). Dieser ist bereits implementiert und dadurch kann man bei den Aufrufen der Endpunkte ermitteln, welcher Benutzer diese aufgerufen hat. Jedoch müssen die Endpunkte so geschützt werden, dass diese nur aufgerufen werden können, wenn der Benutzer das entsprechende Recht hat. Die Logik, durch welche die Rechte eines Benutzers ausgelesen werden können, ist bereits implementiert. Die Route kann mit dem bereits existierenden AppGuard geschützt werden, wodurch sichergestellt ist, dass nur berechtigte Benutzer auf die Seite zugreifen können.

# Mittel und Methoden

Programme:

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

Backend:

* Java
* Gradle für Build und Dependencies
* Spring

Frontend:

* Angular
* TypeScript
* NgRx
* OpenAPI

# Vorkenntnisse

Seit August 2021 arbeitet der Lernende im Team J2EE am Ergo Projekt, wodurch er die Projektstruktur kenn. Er konnte seitdem viele Kenntnisse zu Angular und Java erlernen und hat schon einiges für das Projekt umgesetzt.

# Vorarbeiten

* Umsetzungskonzept schreiben. Das Umsetzungskonzept ist zum jetzigen Zeitpunkt noch nicht erstellt, jedoch bis mindestens eine Woche vor Start der Arbeit abgeschlossen und freigegeben.
* Für die Arbeit braucht es keine weiteren Vorarbeiten für z.B. eine Datenbank, da diese und die benötigten Tabellen bereits existieren.
* Da bereits eine Codebasis und ein Webserver existiert muss hier nichts installiert/konfiguriert werden.

# Abgrenzungen

Jeder Parameter kann unterschiedliche Werte annehmen. So kann ein Parameter zum Beispiel einen Boolescher Wert, einen Text oder eine Selektion von Daten annehmen. Das Erstellen oder Bearbeiten von Parameterwerten würde somit zu viel Zeit in Anspruch nehmen und wird nach der IPA umgesetzt.

# 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

Als Quellen stehen sowohl die offizielle AG-Grid- als auch die offizielle Angular Material-Seite zur zur Verfügung.

# Arbeiten letzte 6 Monate

Bei der Arbeit im J2EE Team hat der Lernende die Navigation, einen Tip of the Day (Tipps zur Verwendung von Ergo) oder die Portierung des Artikelcockpits (Anzeige von Artikelinformationen) in Angular umgesetzt. Dafür hat er jeweils auch Umsetzungskonzepte geschrieben.