Fachhochschule Brandenburg
University of Applied Sciences

Implementierung von Prozessen: Stundenplanung

Fachbereich: Master-Wirtschaftsinformatik

Autoren: Lionel Yimtchui

Ines Kouka

Dozent: Prof. Dr. Vera Meister

Abgabedatum: 21.07.2018

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Am Anfang des Semesters waren wir damit beauftragt, eine Stundenplanung zu erstellen. Bzw. ein BPMN-Prozess zu modellieren und zu implementieren genauso mit der DMN-Tabelle. Und am Ende in der Lage zu sein, diese im Camunda BPM ausführen zu können.

1. Beschreibung des Soll-Prozesses

Das Ziel von diesem Kapital ist unser BPMN-Prozess und DMN-Modell zu beschreiben.

1.1. BPMN-Stundenplanung

Am erstens soll der Dozent vor dem Semesteranfang ihre Angaben zu Lehrveranstaltungen eingeben. Danach werden die Angaben angepasst. Dann kommt unsere DMN-tabelle im Einsatz nämlich für die Auswahl des Raumes für alle Lehrveranstaltung und die ausgewählten Räume werden gebucht. Weiterhin wird der Stundenplan auf Vollständigkeit und Überschneidungsfreiheit geprüft danach auf Korrektheit. Und wenn der Stundenplan nicht korrekt ist, bekommt jeder Dozent eine E-Mail. Weiterhin Wenn der korrekt ist, wird er nach Fachbereichen erstellt. dann ist der Prozess abgeschlossen.

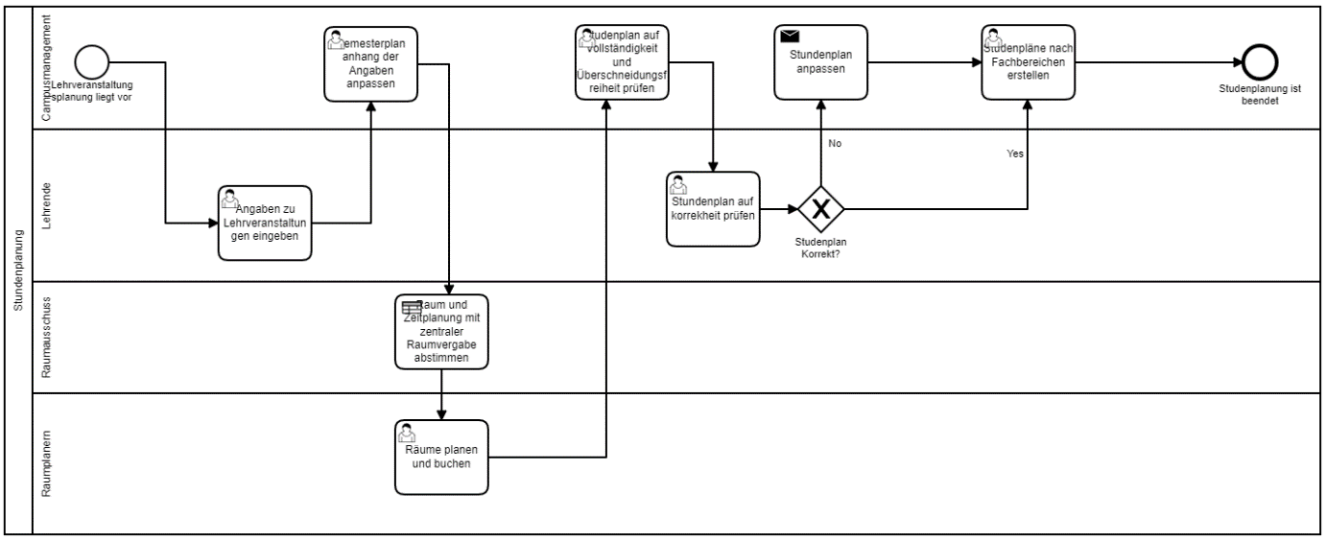


Abbildung 1:BPMN-Stundenplanung

1.2. DMN-Tabelle- Räume und Zeitplanung abstimmen

Für unsere Stundenplanung muss ein Raum für jede Vorlesung nach bestimmten Kriterien bzw. Vorlesungstag und Vorlesungsname ausgewählt werden.

Für das Fach Enterprise Knowledge Engineering, wenn der Wochentag Montag als Input gewählt ist, wird als Output(Raumvergabe) der Raum 319 zugewiesen. Und wenn die Wochentage Donnerstag und Freitag ausgewählt sind, wird die Meldung Vorlesungstag schon besetzt angezeigt.

Für das Fach Implementierung von Prozessen: wenn der Wochentag Donnerstag ausgewählt ist, wird der Raum 321 zugewiesen. Wenn Freitag und Montag ausgewählt sind, ist die Meldung Vorlesungstag schon besetzt zu sehen.

Für das Fach Grundlagen Masch. Lernens: wenn der Wochentag Freitag ausgewählt. Ist der Raum 209 für das Fach zugewiesen. Und für die anderen Wochentage: Donnerstag und Montag erscheint die Meldung die Meldung Vorlesungstag schon besetzt.

Räume und Zeitplanung abstimmen

decision

F	Input +		Output +
	Vorlesungsname Fach	Vorlesungstag Vorlesungstag	Raumvergabe
	string	string	string
1	"Enterprise Knowledge Engineering"	"Montag"	"319"
2	"Enterprise Knowledge Engineering"	"Donnerstag"	"Vorlesungstag schon besetzt"
3	"Enterprise Knowledge Engineering"	"Freitag"	"Vorlesungstag schon besetzt"
4	"Implementierung von Prozessen"	"Donnerstag"	"321"
5	"Implementierung von Prozessen"	"Freitag"	"Vorlesungstag schon besetzt"
6	"Implementierung von Prozessen"	"Montag"	"Vorlesungstag schon besetzt"
7	"Grundlagen masch. Lernens"	"Freitag"	"209"
8	"Grundlagen masch. Lernens"	"Montag"	"Vorlesungstag schon besetzt"
9	"Grundlagen masch. Lernens"	"Donnerstag"	"Vorlesungstag schon besetzt"
10	-	-	-
11	-	-	-
+	-	-	-

Abbildung 2:DMN-Räume und Zeitplanung Abstimmen

2. Implementierung des Soll-Prozesses

Ziel dieses Kapitels ist die Beschreibung der eingesetzten Programme und wie der Prozess implementiert wurde.

2.1. Vorbereitungen zur Implementierung

Eine Test- und Entwicklungsumgebung waren notwendig für die Implementierung des Soll-Prozesses. Weiterhin wird beschrieben, was für Programme und Tools dafür eingesetzt werden.

2.1.1. Testumgebung:

Die Testumgebung erfordert ein Windows 10 Computer, wo die Teste auf dem lokalen Server durchgeführt werden. Und was genau notwendig ist für die technische Implementierung, weil es damit die Möglichkeit gibt, eine grafische Benutzeroberfläche zur Verfügung zu haben.um die Fehlersuche beim jeder Test-version des Prozesses zu erleichtern.

- **Webserver – Camunda Tomcat**:

Ein Camunda Tomcat Server ist auch erforderlich. Zur Lokalen Nutzung muss es entpackt aber nicht installiert werden. Des weiterem brauchen wir auch ein Java Development Kit(JDK).

Lokaler Web-Zugriff: http://localhost:8080/camunda-welcome/index.html

Benutzer und Password: demo / demo

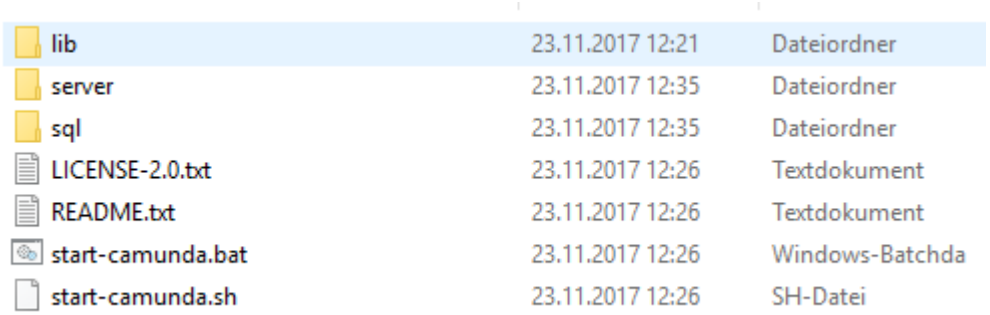


Abbildung 3:Camunda Tomcat

2.1.2. Entwicklungsumgebung:

Was mit der Entwicklungsumgebung angeht, wurde dafür mehrere Programme installiert

- **Entwicklungsumgebung – Eclipse:**

Die Process Engine von Camunda wurde auf der objektorientierten Programmiersprach Java entwickelt. Zur Entwickelt wurde aus diesem Grund das Java basierte Entwicklungsumgebung Eclipse mit Standardeinstellungen installiert.

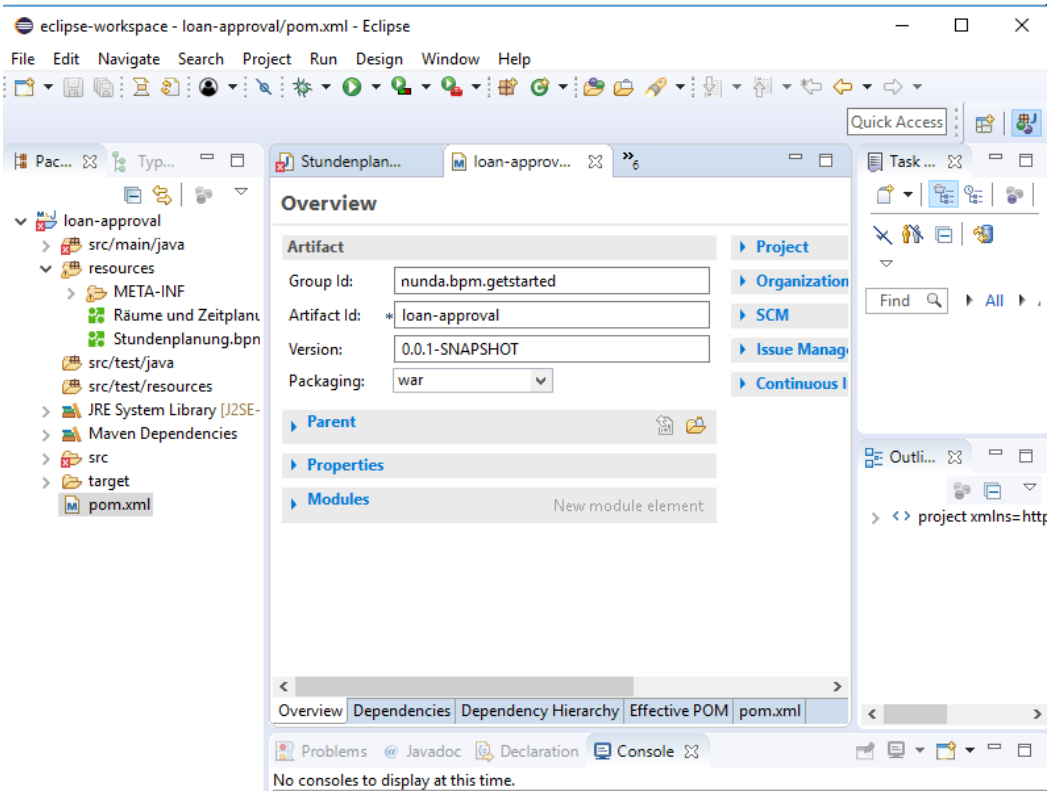


Abbildung 4:Eclipse

- **Entwicklungsumgebung – Camunda Modeler:**

Camunda Modeler ist benutzt um Prozesse zu Modellieren. Aber auch zur technischen Implementierung genutzt. Genauso wie beim Camunda Tomcat muss auch der Modeler nicht installiert, sondern nur entpackt werden.

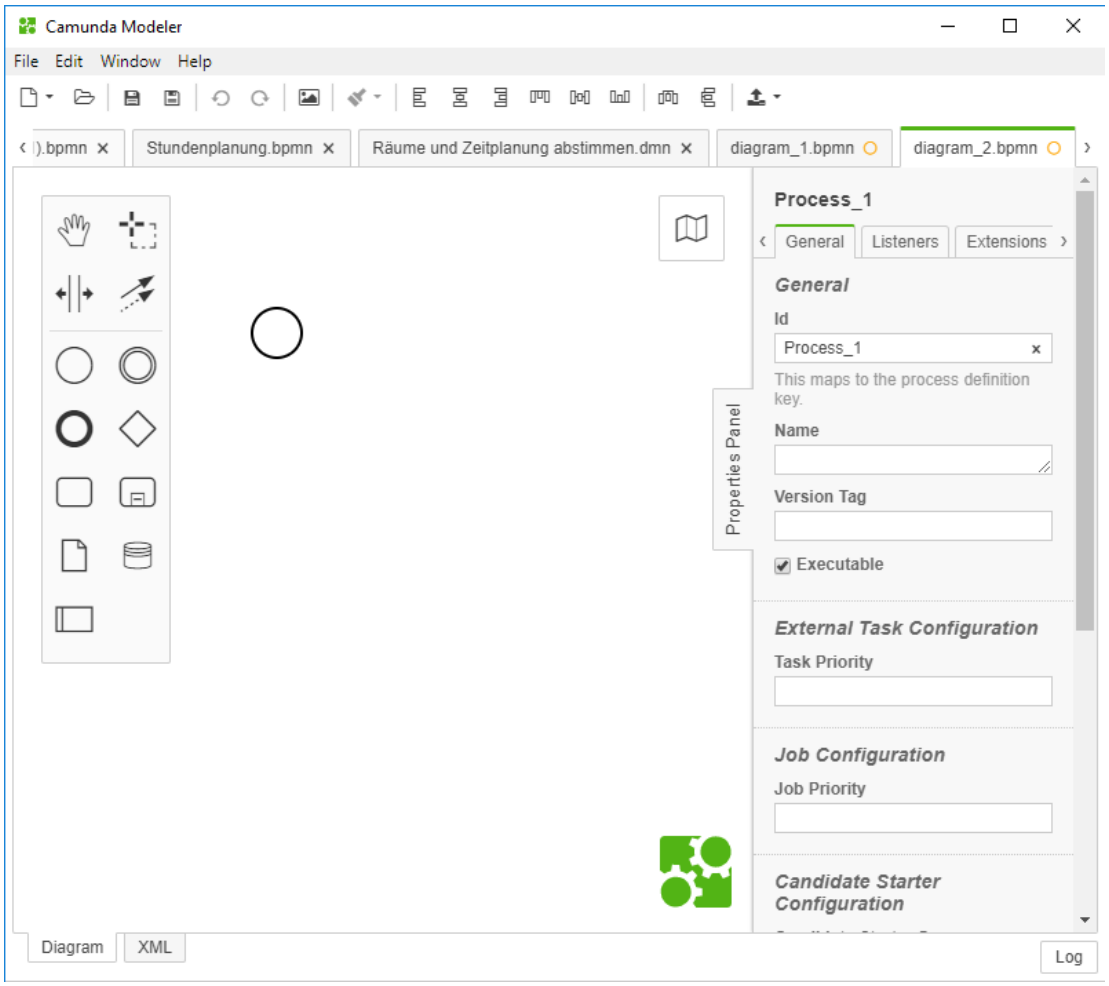


Abbildung 5:Camunda Modeler

2.1.3. Prozessanwendung:

„Bevor das Soll-Modell mit technischen Informationen vorgelegt und somit von der Process Engine ausgeführt werden kann, dann muss in Eclipse eine Prozessanwendung zusammengebunden werden. Diese bildet die Schnittstelle zwischen der Process Engine und der eignen Umgebung.

. Camunda bietet insgesamt vier verschiedene Schnittstellen an (vgl. [Camunda 2017g]):

- **Servlet Process Application**:

Ein Servlet ist eine Java-Klasse, die Anfragen von einem Webserver verarbeitet. Die Process Engine läuft auf einem Apache Tomcat Server.

- **Enterprise JavaBeans Process Application**:

Bei dieser Variante arbeitet die Process Engine auf einem speziell entwickelten Java Enterprise Edition Anwendungsserver

- **Embedded Process Application**:

Die Process Engine läuft eingebettet auf einem Java Plattform (Standard Edition). Dadurch ist es der Prozessanwendung nicht möglich, Prozesse automatisiert zu starten. Jeder Prozess muss manuell vom Anwender aufgerufen werden.

- **Spring Process Application**:

Hierbei läuft die Process Engine innerhalb eines Java-basierten Spring Frameworks. 42 Aufgrund des eingesetzten Camunda Tomcat Servers, wurde die Prozessanwendung als Servlet Process Application implementiert. Dazu wurden die folgenden drei Dateien erstellt:

- **Maven Projekt (POM.xml):**

„Maven ist ein Build-Management-Tool der Apache Software Foundation und basiert auf Java. Mit ihm kann man insbesondere Java-Programme standardisiert erstellen und verwalten“[1]. Über ein so genanntes Projektmodell (engl. Project Object Modell=POM) werden die Metadaten, die erstellt werden, beschreiben. Dank dieser Beschreibungen werden alle notwendigen Bibliotheken in die eigene Prozessanwendung importiert.

[1] https://de.wikipedia.org/wiki/Apache_Maven

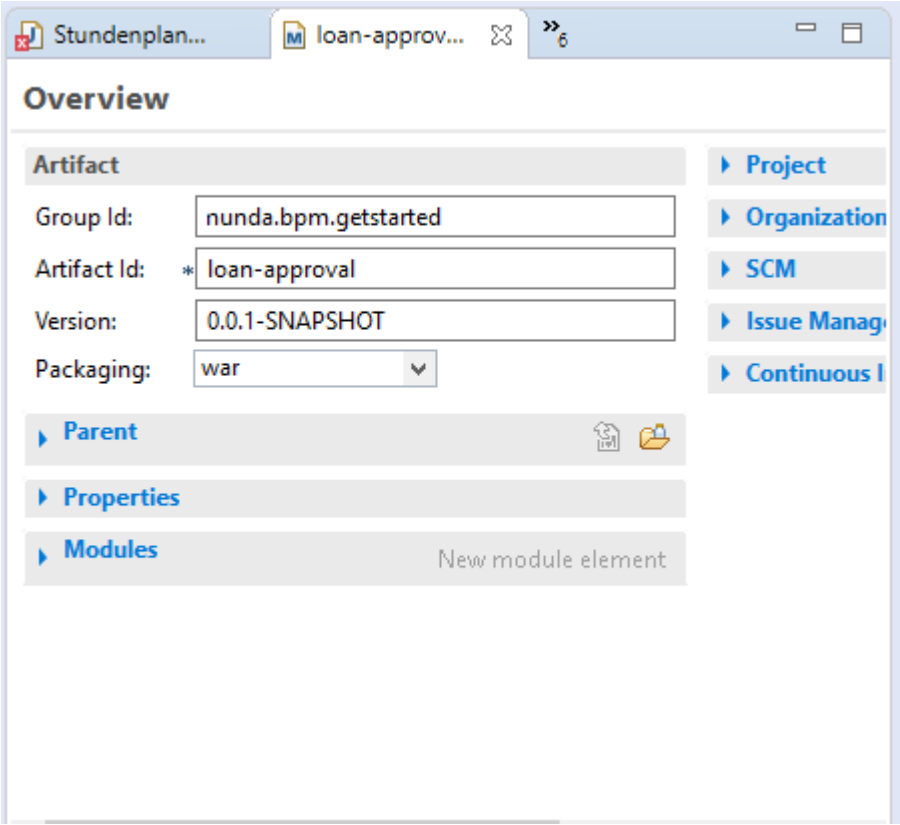


Abbildung 6:Maven Projekt

- **Deployment Descriptor (processes.xml)**

Der Deployment Descriptor ist eine XML basierte Konfigurationsdatei. Für die Vorbereitung müssten die folgenden drei Eigenschaften definiert werden (vgl. [Camunda 2017h]):“[2]

3. Technische Modellierung

Nachdem alle notwendigen technischen Informationen, Tools eingereicht und installiert würden. kann jetzt mit der technischen Implementierung angefangen werden. In diesem Abschnitt wird alle Schritte erläutert. Um den Soll-Prozess mit allen technischen Informationen oder Verknüpfungen anreichern zu können, wurde im Camunda Modeler das „Properties Panel“ genutzt. Jedes Notationselement verfügt über ein individuelles Panel. Der Modeler übersetzt alle Einträge zeitgleich in XML. Durch das Wechseln der Benutzersicht kann die XML-Syntax auch direkt eingegeben werden.

[2] MasterarbeitKaySpannberg

3.1. Prozessdiagramm

Damit der Prozess, der Modelliert wurde, ausführbar wird, wurde der Pool auf „Executable“ gesetzt.

(Der Wert wird dadurch auf „true“ gesetzt). Wenn nein, kann die Process Engine diesen nicht automatisch ausführen. Außerdem muss immer eine eindeutige „ID“ (Stundenplanung) und „Process ID“ (Stundenplanung_1) vergeben sein. Alle anderen Werte sind optional oder sind in diesem Fall nicht relevant.

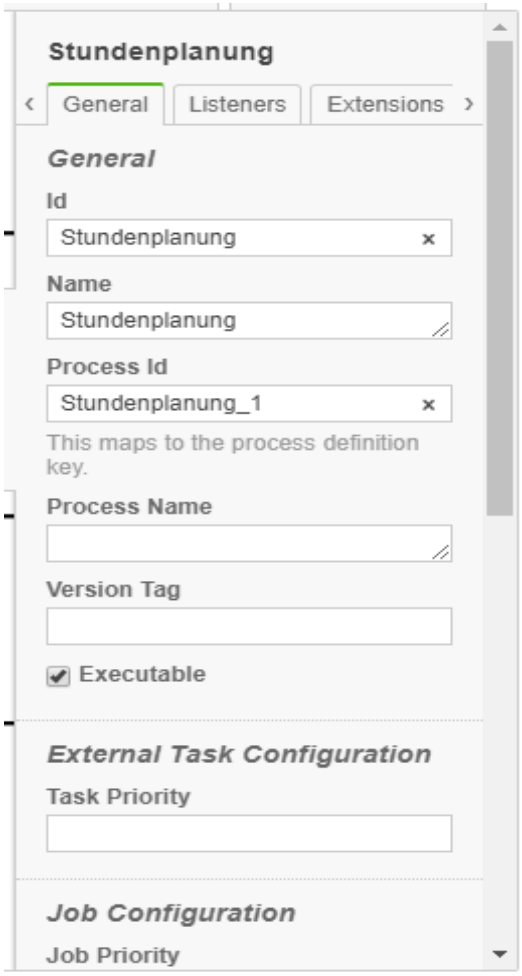


Abbildung 7:Ausführbares Prozessdiagramm

Bildquelle: Screenshot, Camunda Modeler

3.1. Benutzer-Aufgabe

Die Benutzer-Aufgaben werden über Formulare (engl. Forms) definiert. Und diese befindet sich bei der Properties Panel von Camunda Model. Dabei handelt es sich z.B. um HTML Formulare, mit denen eine Benutzeroberfläche erstellt wird. Diese wird dann in der Camunda Tasklist abgebildet.

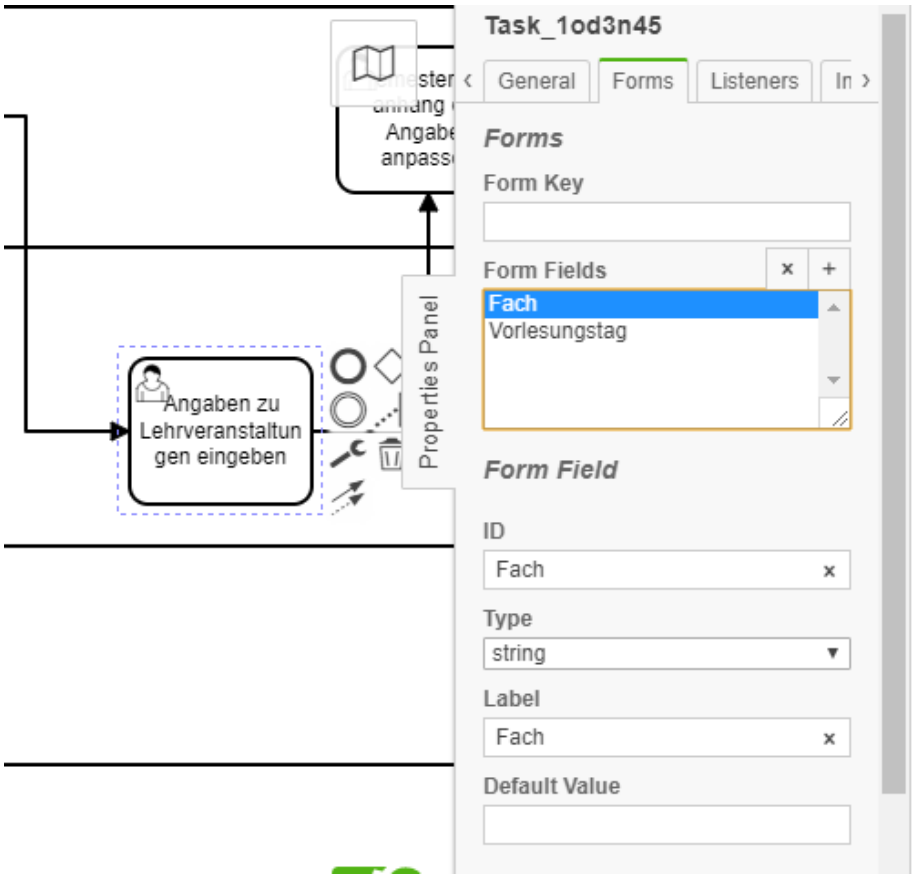


Abbildung 8:User Task

3.2. Nachrichten-Aufgabe

Eine Nachrichten-Aufgabe ist auf technischer Ebene von der Funktionalität mit der Service-Aufgabe gleichzusetzen. Diese ermöglichen das Aufrufen von externen Services (automatisiert), indem sie das Prozessmodell mit der API verbinden. In Camunda erfolgt dieser Vorgang grundsätzlich über Java-Code. Die Nachrichten-Aufgaben stellen im Prototypen eine Verbindung zu einem E-Mail-Server her und versenden darüber die automatisierten E-Mails an den Bewerber. Auf Eclipse wurde eine Java Class erstellt (siehe Abbildung 9). Auf dem Camunda-

panel werden einige Einstellungen vorgenommen. Zum Beispiel als **Implementierung= Java Class, Java Class= org.camunda.bpm.getstarted.loanapproval**. (siehe Abbildung10)

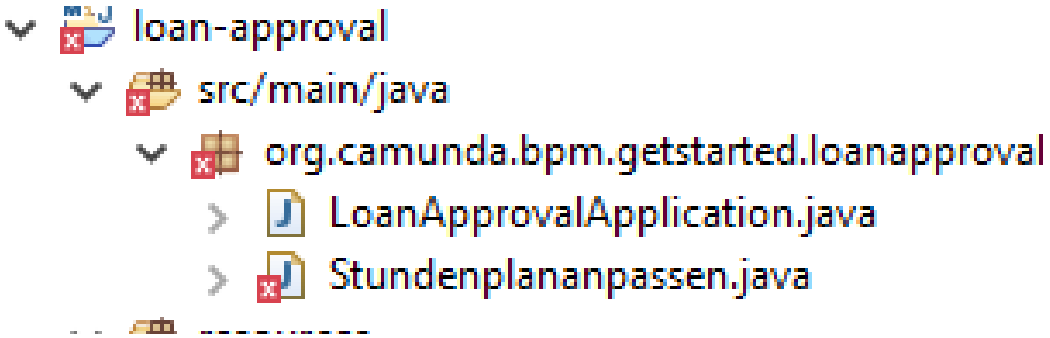


Abbildung 9: Eclipse: Stundenplananpassen.java

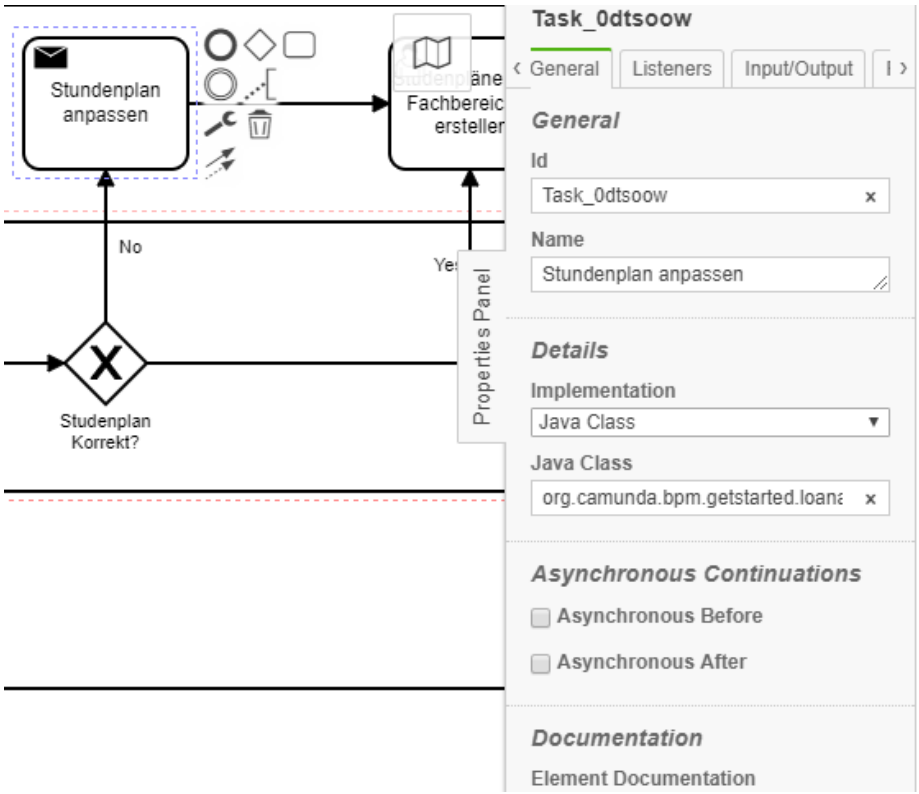


Abbildung 10: Send Task

3.3. DMN-Aufgabe und DMN-Tabelle

Die DMN Aufgaben wurden über die Decision Engine (DMN-Tabelle) umgesetzt. Um z.B. die Räume und Zeitplanung abstimmen zu können, wurde bei der „Decision Ref“ (decision) und bei der Decision-ID (DMN-Tabelle) der gleiche Wert („decision“) eingetragen

Räume und Zeitplanung abstimmen

`decision`

F	Input +		Output +
	Vorlesungsname Fach	Vorlesungstag Vorlesungstag	Raumvergabe
	string	string	string
1	"Enterprise Knowledge Engineering"	"Montag"	"319"
2	"Enterprise Knowledge Engineering"	"Donnerstag"	"Vorlesungstag schon besetzt"
3	"Enterprise Knowledge Engineering"	"Freitag"	"Vorlesungstag schon besetzt"
4	"Implementierung von Prozessen"	"Donnerstag"	"321"
5	"Implementierung von Prozessen"	"Freitag"	"Vorlesungstag schon besetzt"
6	"Implementierung von Prozessen"	"Montag"	"Vorlesungstag schon besetzt"
7	"Grundlagen masch. Lernens"	"Freitag"	"209"
8	"Grundlagen masch. Lernens"	"Montag"	"Vorlesungstag schon besetzt"
9	"Grundlagen masch. Lernens"	"Donnerstag"	"Vorlesungstag schon besetzt"
10	-	-	-
11	-	-	-
+	-	-	-

Abbildung 11:DMN-Tabelle

Neben der Verknüpfung mussten noch weitere Parameter definiert werden. Bei der DMN-Tabelle mussten die Auswertungsvorschrift (First), die Eingabe- sowie Ausgabefelder, die Datentypen (z.B. String) und die Entscheidungsregeln definiert werden. Dazu wurde:

- die Auswertungsvorschrift auf „First“ gesetzt, „möglicherweise stellen Sie fest, dass die Trefferrichtlinie in diesem Beispiel nicht "eindeutig" ist, sondern "zuerst" (als "F" markiert). Dies bedeutet, dass die Entscheidungs-Engine die Regeln auswertet und die Auswertung beendet, sobald eine gültige Regel gefunden wurde.“[3]

- der Vorlesungsname und Vorlesungstag als Text (String) und Eingabefelder (Input) definiert.
- die Raumvergabe als Test (String) und Ergebnisfeld bzw. Ausgabefeld (Output) definiert.

[3] https://camunda.com/dmn/

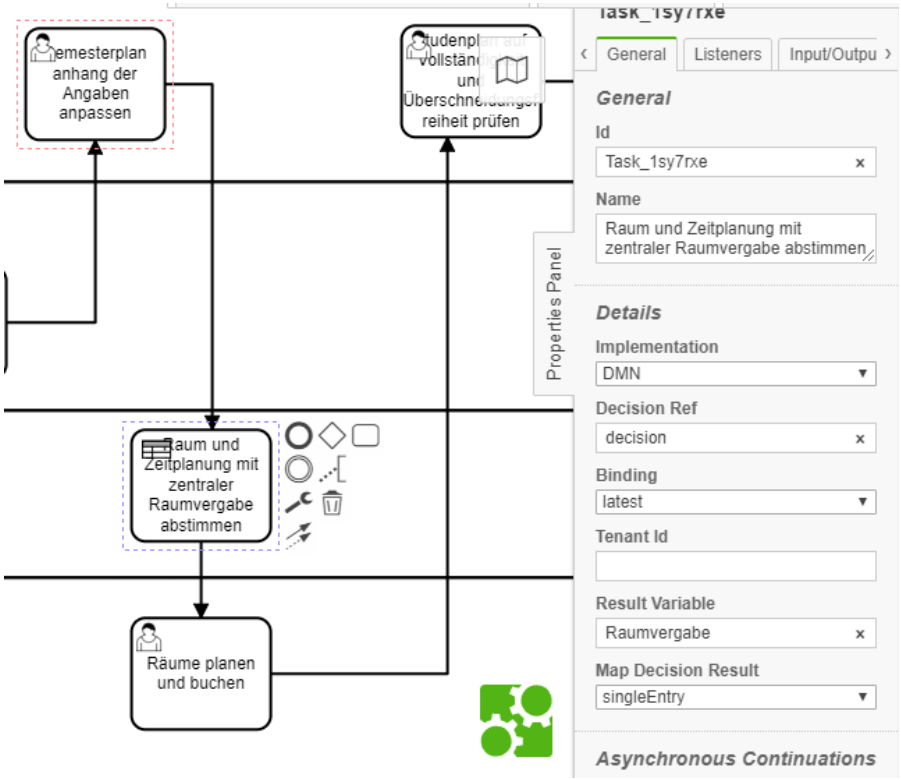


Abbildung 12:DMN-Task

Bildquelle: Screenshot, Camunda Modeler

Bei der DMN-Aufgabe mussten verschiedene Entscheidungsparameter für die Process Engine definiert werden. Dazu wurde:

- das Binding auf „latest“ gesetzt, da immer die neuste (letzte) Entscheidung genommen werden soll und keine Benutzerdefinierte.
- die Result Variable wurde mit dem Wort „Raumvergabe“ definiert. Diese Variable enthält den kompletten Output aus der DMN-Tabelle.
- die Map Decision Result wurde als „singleEntry“ definiert, weil nur ein Output und nur eine zutreffende Regel von der DMN-Tabelle erwartet wird.

Bei der anderen DMN-Aufgabe („Räume und Zeitplanung abstimmen“) wurde die Map Decision Result als „singleResult“ deklariert. Dabei wird die Result Variable als HashMap abgespeichert, wodurch die einzelnen Outputs der DMN-Tabelle nicht mehr einzeln von der Process Engine abgerufen werden können.

3.4. Gateway

Beim Daten-basierten exklusiv Gateway werden die nachfolgenden Entscheidungsflüsse ausgewertet. Bzw. ob der Stundenplan korrekt ist? Weiterhin bei dem Camunda-Panel haben wir als Name (Stundenplan Korrekt?) und als id (ExclusiveGateway_0uhxumh).

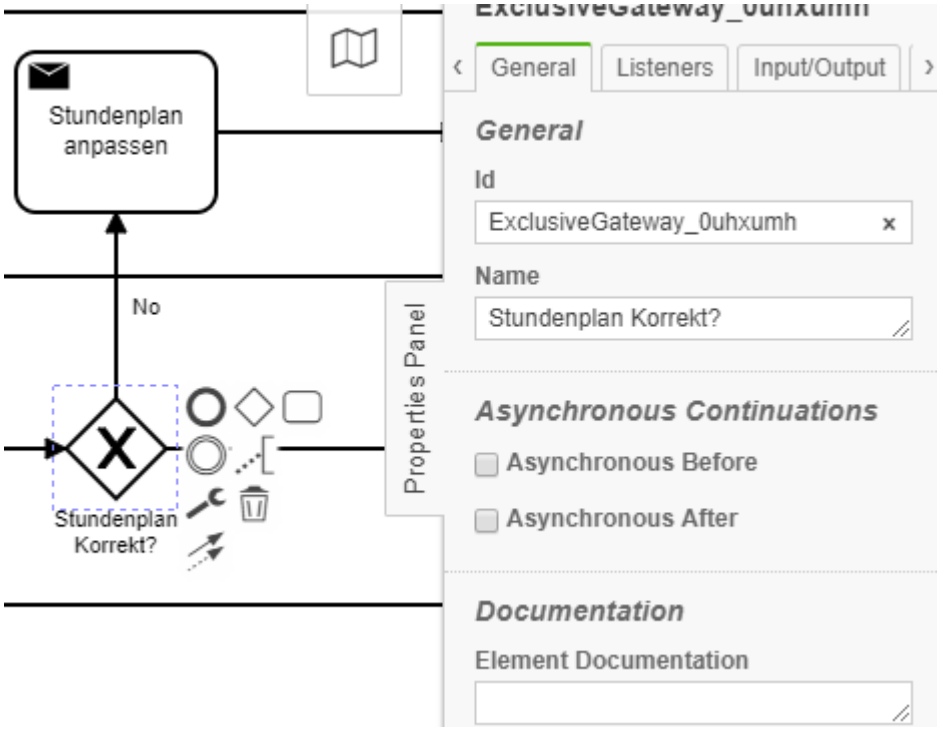


Abbildung 13:Gateway

Wenn der Stundenplan nicht korrekt ist, wird der Entscheidungsflüsse wie folgendes eingestellt. Und bei der Camunda Modeler nämlich bei Properties Panel sind folgende Einstellungen eingesetzt.

- Name = No,
- Expression = ${approved==false},
- Condition Type = Expression

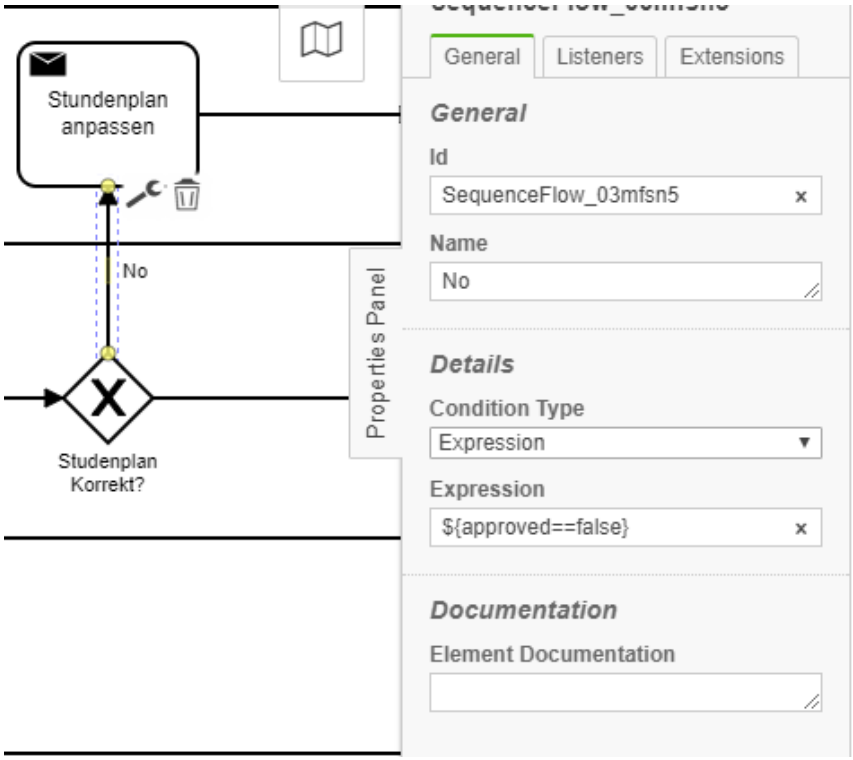


Abbildung 14:Sequenzflüsse-No

Wenn der Stundenplan korrekt ist, wird die Entscheidungsflüsse wie folgendes eingestellt. Beim Daten-basierten exklusiv Gateway werden die nachfolgenden Entscheidungsflüsse ausgewertet. Der Fluss, welcher „wahr“ (engl. true) ist, wird ausgewählt und der Prozess fortgesetzt.

- Name= Yes
- Condition Type = Expression
- Expression= ${approved==true}

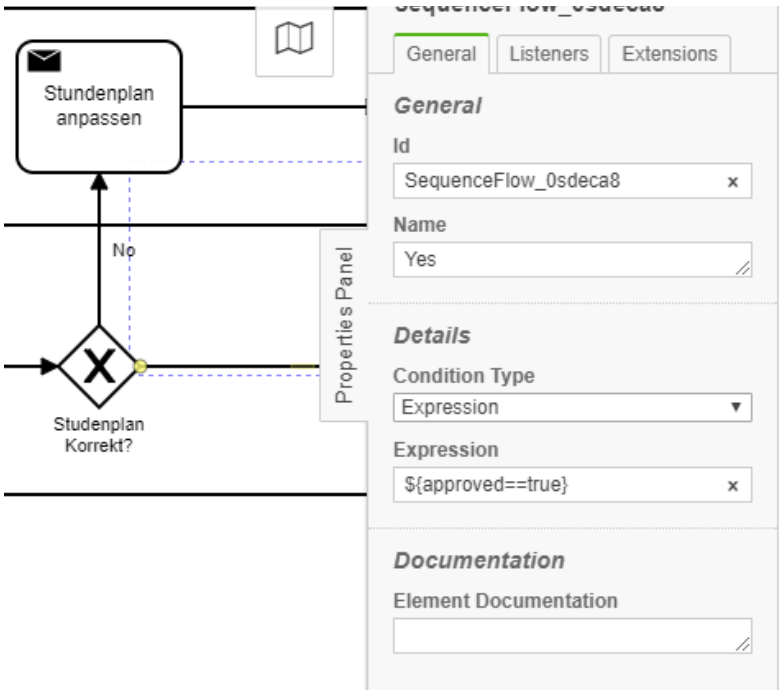


Abbildung 15:Sequenzflüße-yes

4. Technische Implementierung in Camunda BPM

Das Ziel dieses Kapitels ist zu zeigen, wie unsere modellierte Prozesse auf einer Benutzeroberfläche aussehen wird.

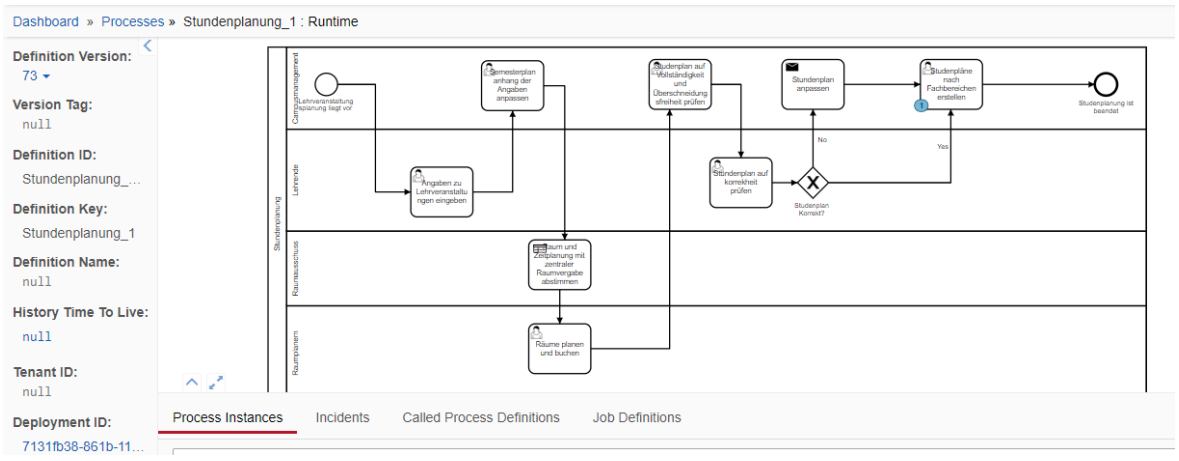
4.1. Modeler

Der camunda Modeler ist eine einfach zu bedienende App für die Bearbeitung von BPMN-Prozess Diagrammen und DMN-Entscheidungstabellen. Sie können die erstellten Dateien zu den camunda-Motoren einsetzen, um Sie auszuführen.[4]

4.2. Camunda Cockpit

Mit camunda Cockpit können Sie Arbeitsabläufe und Entscheidungen in der Produktion überwachen, um technische Probleme zu entdecken, zu analysieren und zu lösen. Cockpit ist das perfekte Werkzeug für den technischen Prozessbetrieb.[5]

In unserem Fall hat der Benutzer da noch die Möglichkeit, der BPMN-Prozess (siehe Abbildung 14.) und die DMN-Tabelle (siehe Abbildung 16.) zu sehen. Damit kann er auch wissen, in welchem Schritt der Prozess sich gerade befindet. Außerdem können die unterschiedlichen Angaben, die bei Task List eingegeben wurden, gesehen werden. noch dazu kann die getroffene Wahl der DMN-Tabelle angezeigt werden (siehe Abbildung 17,18.)



[4] https://camunda.com/products/
[5] https://camunda.com/products/

Abbildung 16:Cockpit-BPMN

Bildquelle: Camunda BPM Schreenshot

F	Input		Output
	Vorlesungsname	Vorlesungstag	Raumvergabe
	string	string	string
1	"Enterprise Knowledge Engineering"	"Montag"	"319"
2	"Enterprise Knowledge Engineering"	"Donnerstag"	"Vorlesungstag schon besetzt"
3	"Enterprise Knowledge Engineering"	"Freitag"	"Vorlesungstag schon besetzt"
4	"Implementierung von Prozessen"	"Donnerstag"	"321"
5	"Implementierung von Prozessen"	"Freitag"	"Vorlesungstag schon besetzt"
6	"Implementierung von Prozessen"	"Montag"	"Vorlesungstag schon besetzt"
7	"Grundlagen masch. Lernens"	"Freitag"	"209"
8	"Grundlagen masch. Lernens"	"Montag"	"Vorlesungstag schon besetzt"
9	"Grundlagen masch. Lernens"	"Donnerstag"	"Vorlesungstag schon besetzt"

Abbildung 17:Cockpit-DMN

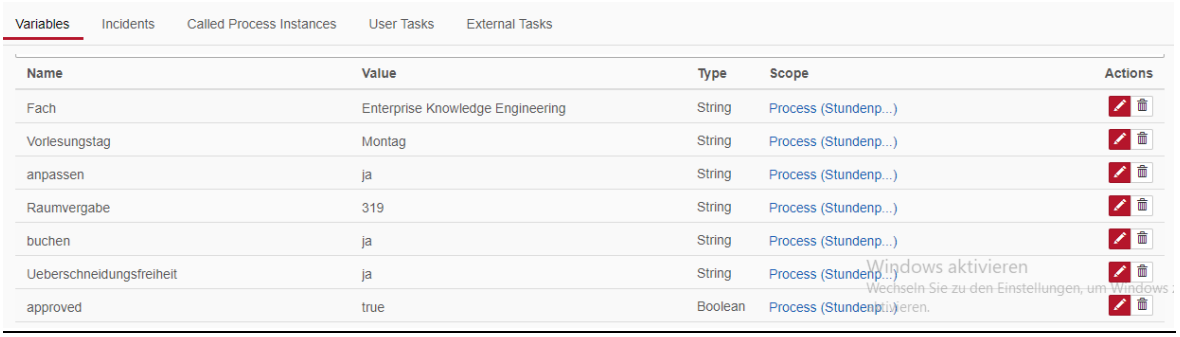


Abbildung 18:Camunda Cockpit Variables

4.3. Camunda Tasklist

Die camunda tasklist ist eine gebrauchsfertige Web-Anwendung, die es den Endverbrauchern ermöglicht, an den Ihnen zugewiesenen Aufgaben zu arbeiten. Es bietet zusätzliche Sichtbarkeit bei der Verwendung der camunda-Workflow-Engine für den User Task Management[6]

Der Benutzer hat hier die Möglichkeit, alle Aufgaben zu sehen und an wem diese Aufgaben zugewiesen sind (siehe Abbildung19).

[6] https://camunda.com/products/

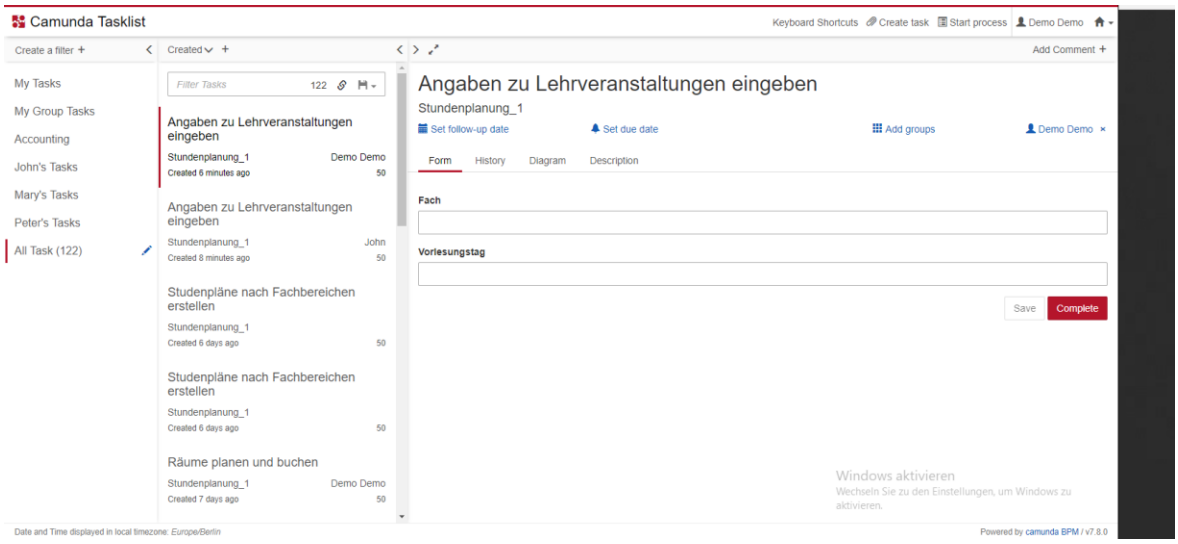


Abbildung 19:Camunda Tasklist

4.4. Admin

Verwalten Sie Ihre camunda Web-Anwendung oder Rest-API-Nutzer, organisieren Sie Sie in Gruppen und gewähren Sie Berechtigungen mit camunda admin. Sie können auch eine bestehende Benutzerverwaltung über LDAP integrieren.[7]

5. Github

Hier wurde eine Repository mit dem Namen: lionelyimtchui/Stundenplanung erstellt, wo unser BPMN-Prozess: Stundenplanung und DMN-Tabelle: Räume und Zeitplanung abstimmen gespeichert wurden.

Der Link: https://github.com/lionelyimtchui/Stundenplanung

[7] https://camunda.com/products/

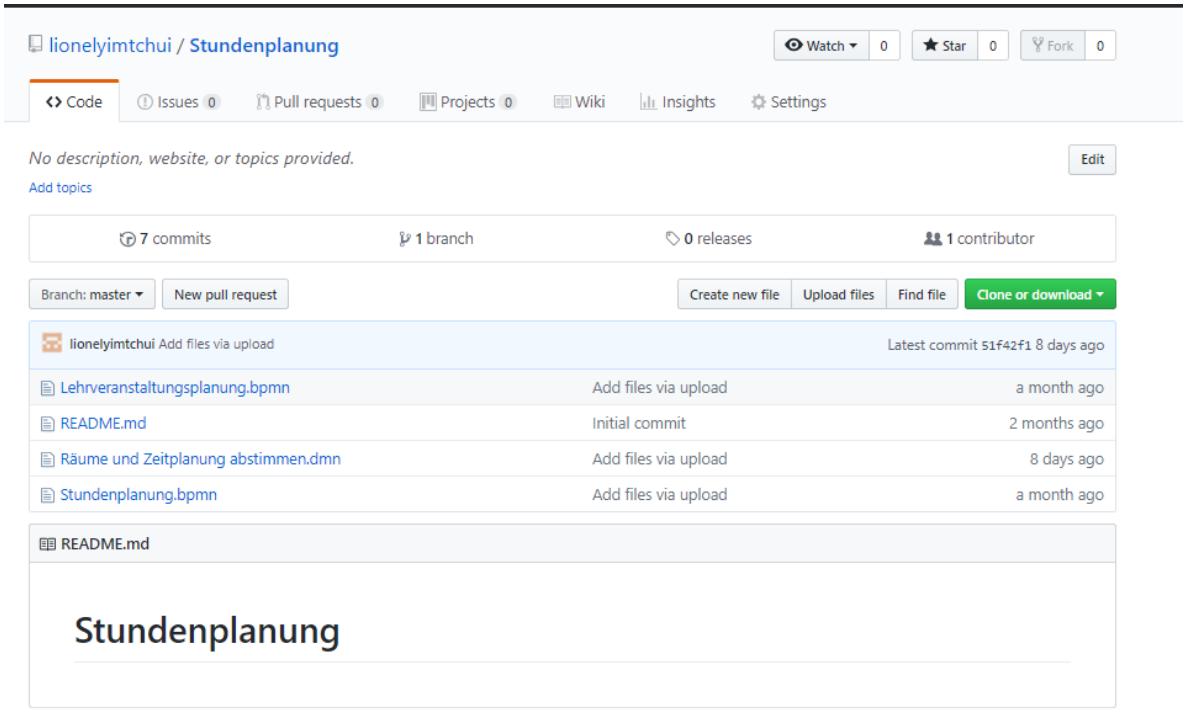


Abbildung 20: Github

Fazit

In Laufe der Aufgabe Implementierung der Stundenplanung haben wir einige Schwierigkeiten getroffen. Besonders bei dem **Send Task: Stundenplanung anpassen**. Wie Camunda keine passende Unterstützung dafür anbietet, haben wir erstmal eine Java Class erstellt, und gegoogelt, ob wir einen passenden Code finden. Zwar haben wir was gefunden aber bei der Ausführung des Prozesses hat genau diese Aktivität leider nicht geklappt. Ansonsten haben die andere Aktivität gut funktioniert.

Literaturverzeichnis

https://camunda.com/products/

http://maven.apache.org/what-is-maven.html

MasterarbeitKaySpannberg