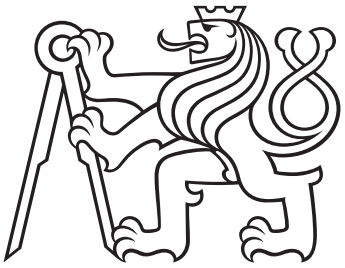
České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra počítačů
Studijní program: Softwarové inženýrství a technologie



Možnosti v rozšíření funkcionalit platformy Camunda

Possibilities in extension of functions in Camunda platform

BAKALÁŘSKÁ PRÁCE

Vypracoval: Filip Nácovský
Vedoucí práce: Ing. Radek Hronza, Ph.D.
Rok: 2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nácovský** Jméno: **Filip** Osobní číslo: **483452**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Možnosti v rozšíření funkcionalit platformy Camunda

Název bakalářské práce anglicky:

Possibilities in extension of functions in Camunda platform

Pokyny pro vypracování:

1) Seznamte se s oblastí procesního řízení, známého také jako Business Process Management. Zaměřte se zejména na možnosti elektronizace procesů prostřednictvím BPMS nástrojů
2) Analyzujte podrobněji BPMS nástroj Camunda.
3) Navrhněte možné způsoby rozšíření tohoto nástroje o nové funkcionality/moduly.
4) Vybrané návrhy implementujte, řádně otestujte a vyhodnoťte jejich reálné přínosy

Seznam doporučené literatury:

[1] Camunda docs [online]. [cit. 2020-10-07]. Dostupné z: https://docs.camunda.org/manual/7.13/
[2] ŘEPA, Václav. Procesně řízená organizace. Praha: Grada, 2012. ISBN 978-80-247-4128-4.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Radek Hronza, Ph.D., Procesní konzultant/analytik, IBPM Solutions s.r.o.

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **22.02.2021** Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Radek Hronza, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. | Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne ....

Filip Nácovský

Poděkování

Děkuji Ing. Radkovi Hronzovi, Ph.D. za vedení mé bakalářské práce a za podnětné návrhy, které ji obohatily.

Filip Nácovský

Název práce:
Možnosti v rozšíření funkcionalit platformy Camunda

Autor: Filip Nácovský

Studijní program: Softwarové inženýrství a technologie

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Radek Hronza, Ph.D.
IBPM Solutions s.r.o.

Abstrakt: Tato práce se zabývá představením základů procesního řízení a elektronizace procesů. Nejprve je rozebrána úvodní teorie v rámci procesního řízení a procesů, následně je uvedena konkrétní platforma na elektronizaci procesů - Camunda. V rámci praktické části je nejprve uvedena analýza možných rozšíření Camundy. Následně jsou daná rozšíření implementována a otestována. Závěr praktické části obsahuje zhodnocení přínosů.

Klíčová slova: procesní řízení, automatizace procesů, camunda, externí worker

Title:
Possibilities in extension of functions in Camunda platform

Author: Filip Nácovský

Abstract: This thesis deals with the introduction into business process management and process automation. Firstly, it offers the introductory theory into process management and processes, then a specific platform for process automation - Camunda - is presented. The practical part begins with an analysis of possible extensions to Camunda which are then tested and implemented. The practical part concludes with an evaluation of the benefits.

Key words: BPM, process automation, camunda, external worker

Obsah

Seznam použitých zkratek

BPM - Business Process Management

BPMS - Business Process Management Software / Business Process Management System

BPMN - Business Process Model and Notation

CMMN - Case Management Model and Notation

DMN - Decision Model and Notation

IT - Informační technologie

IoT - Internet of Things (internet věcí)

SW - Software

API - Application programming interface (aplikační rozhraní)

JVM - Java Virtual Machine

JSON - JavaScript object notation

HTML - Hyper Text Markup Language

Seznam obrázků

Přílohy

0.1 Archiv obsahující sestavené procesy a implementované workery

- Podsložka procesy obsahuje hlavní proces a proces přihlášek
- Podsložka workery obsahuje implementované workery

Úvod

Za posledních 50 let došlo k obrovskému rozvoji počítačů. Počítače jsou dnes ve všech odvětvích, ale stále se můžeme ve spoustě případů setkat se skutečností, že činnosti, které by mohl počítač vykonávat, vykonávají lidé. Z toho důvodu je potřeba elektronizovat procesy, aby se lidé mohli více věnovat činnostem, které počítače dělat nemohou.

Cílem teoretické části této práce je přiblížení problematiky elektronizace procesů a dále konkrétní platformy Camunda, která je velice účinný nástroj pro elektronizaci procesů a zároveň je dostupná zdarma. Cílem praktické části je nejprve analyzovat vhodné rozšíření platformy Camunda a následně toto rozšíření implementovat. Dále je vytvořen reálný proces, který dané rozšíření demonstruje a na závěr je celý funkční celek otestován.

Motivace k tomuto tématu práce spočívá v nedávném setkání s platformou camunda, která obsahuje velice zajímavé možnosti v rámci elektronizace procesů. Z toho důvodu vzniká tato práce, která blíže seznamuje s Camundou a dále Camundu rozšiřuje v rámci praktické části.

Práce je strukturována do dvou základních celků - teoretická a praktická část. V rámci teoretické části je nejprve obsaženo seznámení s obecnou problematikou procesního řízení - co je to proces, jaký má proces životní cyklus v rámci procesního řízení a s elektronizací procesů. Následně se práce věnuje obecným vlastnostem softwarů pro podporu procesního řízení a notacím užívaným v rámci elektronizace procesů. Dále je čtenář seznámen s konkrétním softwarem pro elektronizaci procesů - Camundou. V rámci této části je popsána její infrastruktura a architektura.

Úvod do praktické části obsahuje upřesnění cílů této práce. Následně je uvedena analýza možných rozšíření platformy Camunda. Na základě výstupu analýzy je dále uvedena tvorba procesů, na kterých je vybrané rozšíření implementováno. Dále je uveden postup implementace vybraných rozšíření. Po implementační kapitole je uveden postup propojení implementovaných částí s procesem a spuštění celého celku. Na závěr jsou všechny implementované části otestovány a vyhodnoceny přínosy.

Kapitola 1

Teoretická část

1.1 Procesní řízení

Tato kapitola popisuje základy procesního řízení (anglicky Business process management - zkratka BPM). Nejprve seznamuje s pojmem proces a blíže přibližuje vnímání procesů v rámci této práce. Dále obsahuje stručné kořeny a základy procesního řízení a na závěr je popsán životní cyklus procesů v rámci procesního řízení.

1.1.1 Základy procesního řízení

"Procesním řízením se rozumí řízení firmy takovým způsobem, v němž business (podnikové) procesy hrají klíčovou roli".[1] Základem pro efektivní procesní řízení je pochopení základní logiky byznysu dané organizace. Je tedy potřeba pochopit zákadní činnosti a vazby mezi nimi v návaznosti na cíle. Díky procesům a procesnímu řízení lze dosáhnout dynamiky ve fungování dané organizace. To je potřebné pro efektivní přizbůsobení novým možnostem i rychlou reakci na případné hrozby. Procesy jsou základ pro fungování celé organizace.[2]

1.1.2 Proces

O procesech slýcháme v životě často a s oblibou toto slovo využívá mnoho lidí a nejvíc je daný pojem oblíbený v rámci IT firem. Proces může být chápán mnoha různými způsoby a tím jde do pozadí skutečná podstata procesů. V této práci je proces chápán jako **„objektivně přirozená posloupnost činností, konaných s úmyslem dosažení daného cíle v objektivně daných podmínkách"**.[1]

V rámci uvedené defince lze pozorovat posloupnosti činností, které jsou seřazeny v čase. Jde tedy o posloupnost činností, které lze znázornit také na časové ose, kde je každá činnost konána v určitém časovém rámci. Ke každému procesu také patří cíl procesu a jeho úmysl. Jde o dosažení daného cíle v určitých podmínkách organizace.[1]

Procesy se dělí do tří základních skupin. Jedná se o procesy **hlavní, řídící a podpůrné**. Hlavní procesy jsou určené k vytváření hodnot (generování zisku dané organizace). Řídící procesy, jak již z názvu vypovídá, řídí chod dané organizace. Podpůrné procesy mají za úkol podporovat hlavní a řídící procesy.[2]

1.1.3 Životní cyklus procesů v rámci procesního řízení

Dle [2] lze životní cyklus procesů rozdělit do 6 fází, které se dají dále rozdělit do 3 milníků - definice a dokumentace průběhu procesů, vykonávání procesů a monitoring a vyhodnocení klíčových ukazatelů.

Definice a dokumentace průběhu procesů

- **1. Fáze - analýza.** Analýza vyžaduje pochopení fungování dané organizace. Je nutné zmapovat strategické cíle a její strukturu. Dále je do analýzy potřeba zahrnout výstupy z fáze 6, pokud již nějaké existují. Cílem je identifikace požadavků na proces pomocí analýzy známých dat.

- **2. Fáze - návrh.** Návrh využívá informace získané během anlýzy pro návrh změn v organizační struktuře organizace a v rámci procesů. Cílem této fáze je aktualizace organizační struktury a tvorba procesního modelu.

Vykonávání procesů

- **3. Fáze - implementace.** Základem této fáze je nová organizační struktura a procesní model z fáze návrhu. V implementační fázi dochází k realizaci navrhovaných změn do chodu organizace.

- **4. Fáze - vykonávání procesů.** Po implementační fázi je žádoucí začít vykonávat dané procesy novým optimalizovaným způsobem. Pokud jsou procesy v rámci organizace pouze v „papírové"podobě, je nutné zajistit u zmaměstnanců používání nových procesů například pomocí směrnic. Naopak pokud organizace využívá elektronizované procesy (např. pomocí BPMS viz. kapitola 1.2), je přechod k novým procesům výrazně jednodušší - software pomáhá zaměstnancům při práci s novými procesy.

Monitoring a vyhodnocení klíčových ukazatelů

- **5. Fáze - monitoring.** Monitoring probíhá souběžně s fází vykonávání procesů. Cílem této fáze je zjištění slabých míst a nedostatků nově navržených procesů.

- **6. Fáze - vyhodnocení.** V rámci vyhodnocení jsou přebírána data z monitorovací části. Tato data je nutné podrobit analýze a následně vyhodnotit nedostatky návrhů, které se předají opět do fáze 1.

1.2 Elektronizace (Automatizace) procesů

Elektronizace procesů je přístup, ve kterém organizace používá technologii k organizování jejich zaměstnanců a systému v rámci pracovního postupu. Existují předpoklady, že zavedení elektronizace procesů společně s procesním řízením může organizaci učetřit až 90 procent provozních výdajů [3].

Elektronizace procesů umožňuje mít kontrolu nad různými otázkami jako vývoj, plánování, prodej, interakce se zákazníky apod. Dále pomáhá šetřit operační výdaje a uvolňovat lidské zdroje pro vykonávání složitějších úkolů, které nemohou být elektronizovány a zároveň minimalizuje možnost lidské chyby v elektronizovaných procesech. Zdroj celé kapitoly 1.2 [3].

1.2.1 Typy elektronizace

- **Základní** elektronizace je zaměřena na jednoduché práce v rámci organizace. Například použití centrálního nástroje pro správu komunikace zaměstnanců.
- **Procesní** elektronizace řídí podnikové procesy. Má větší možnosti než základní elektronizace a může být kontrolována dalším softwarem nebo aplikacemi.
- **Integrační** elektronizace je více komplexní než procesní. Umožňuje sledovat, jak lidé vykonávají úkoly a tyto akce reprodukovat. Lidé musí ale určit pravidla, podle kterých se daná integrace chová. Například lze integrovat BPMS a podpůrný zákaznický software. Tato integrace může analyzovat výsledky ze zákaznické podpory a pokud je to nutné, přiřadí řešení zaměstnancům firmy.
- **S umělou inteligencí**. Přidání umělé inteligence k integračnímu softwaru umožňuje dělat rozhodnutí podobné člověku. Systém dělá rozhodnutí na základě naučených postupů z minulosti a jejich analýzy.

1.2.2 Proč elektronizovat procesy

Zaměstnanci pořádají schůzky, požadují schvalování, revidují dokumenty a pracovní postupy, trasují informace atd. Ve spoustě organizací jsou tyto akce stále prováděny manuálně. Následné hledání například správné verze dokumentu vyžaduje mnoho času a není zaručeno, že zaměstnanec neudělá chybu. A pokud neudělá chybu daný zaměstnanec, mohl udělat chybu někdo jiný v rámci řízení společnosti. Všechny tyto chyby stojí peníze.

Při elektronizaci procesů dojde k zajištění několika důležitých faktorů [3]:

- **Redukce chybovosti**. Při správné implementaci elektronizace lze snížit chybovost na minimum. Počítač se narozdíl od člověka nepřehlédne a dělá práci vždy stejně precizně.
- **Redukce časové náročnosti**. Tyto repetetivní typy procesů jsou prováděny mnohem rychleji počítačem než člověkem. Počítač také dokáže pracovat bez přestávek.
- **Redukce výdajů**. Díky elektronizaci daných procesů dochází k velké finanční úspoře.

Z výše uvedených informací vychází, že po nasazení elektronizace procesů lze vykonávat procesy rychleji, levněji a lépe.

1.3 BPMS

Dnes známá koncepce BPMS se vyvinula v prvních deseti letech nového tisíciletí. Došlo k částečné standardizaci přístupu k automatizaci procesů, ale přesné vymezení BPMS je stále předmětem diskuzí. Tato kapitola uvádí **funkce a přínosy BPMS** podle [4].

1.3.1 Funkce BPMS

BPMS je balíček nástrojů pokrývající jednotlivé fáze životního cyklu procesu. Standardně společnost využívá nástroje pro všechny fáze od jednoho výrobce z důvodu jednodušší implementace. Tato podkapitola propojuje jednotlivé fáze životního cyklu procesu z minulé kapitoly s BPMS.

- **Analýza a popis procesů.** Popis procesů v rámci BPMS probíhá způsobem, který je srozumitelný i manažerům a vlastníkům procesů, diky tomu dochází k lepší spolupráci byznysu a vývoje. V rámci popisu dochází k vytvoření přehledu vstupů a výstupů, cíle, časového toku a jednotlivých aktivit procesu.

- **Modelování procesů.** Modelování procesů zahrnuje kompletní popis procesu a jeho aktivit způsobem, který je standardizovaný pro BPMS. Obecně výrobci uznávaná notace v rámci BPMS je notace BPMN, aktuálně dostupná ve verzi BPMN 2.0 (bližší seznámení viz. kapitola 1.3 - Notace).

  Nicméně i přes používání standardizované notace výrobci do svých systémů nezahrnují celou šíři notace BPMN, ale jen vybrané části. Navíc někteří výrobci přidávají i vlastní funkcionality. Z těchto důvodů jsou stále obtížně přenositelné modely napříč výrobci.

  Pokud je již hotový model, BPMS umožňuje simulaci průběhu namodelovaného procesu s předem nastavenými testovacími daty. Simulace umožňuje ověření kapacit zdrojů, nákladů na proces apod.

- **Implementace procesní aplikace.** Funkce BPMS zaměřené na vlastní vývoj procesní aplikace obvykle vyžadují vyšší znalosti vývoje SW aplikací. Nový model procesu je přebírán vývojářským týmem, který má za úkol propojení procesu s podpůrnými informačními systémy. Integrovatelnost s podpůrnými informačními systémy je klíčová pro efektivní vývoj.

  Procesní aplikace není primárně určena pro poskytování aplikační logiky, tj. provádění výpočtů, složitější práce s databázemi apod. Pokud je v rámci průběhu procesní aplikace nutné využít další aplikační logiku, je tato funkce zavolána ve formě IT služby, která je implementována v rámci BPMS. Někteří výrobci nabízejí knihovny s předpřipravenými IT službami.

  Druhá oblast, kterou je nutné zajistit, je tvorba uživatelského rozhraní (většinou formulářů). Některé BPMS obsahují nástroje pro návrh formulářů pomocí systému drag and drop (táhni a pusť). Systém drag and drop formulářů představuje výhodu pro byznys analytiky, kteří mohou tvořit jednoduché formuláře bez znalosti programování. Pro složitější formuláře a integraci podpůrných systémů do formulářů je znalost programování nutná.

- **Spuštění a provoz procesů.** Další součástí BPMS je procesní server, který zajišťuje spouštění instancí procesů, správné fungování podpůrných systémů a

monitoring probíhajících procesů. Dále poskytuje možnost administrace procesních aplikací.

Většina BPMS systémů obsahuje portálové řešení, které uživatelům umožňuje v internetovém prohlížeči spravovat procesní aplikaci. Uživatelé, kteří mají přiřazenou aktivitu v rámci průběhu procesu, zde mohou danou aktivitu vyřídit. Vyřízení probíhá pomocí formuláře vygenerovaného v rámci BPMS nebo propojením s jiným informačním systémem. Pokud organzaci tato webová aplikace nevyhovuje, je možné integrovat vlastní aplikaci. Práva uživatelů v rámci webové aplikace nastavuje administrátor nebo je možné propojení se systémy pro správu identit.

V rámci podpory lidských aktivit a spolupráce BPMS obsahuje dvě důležité funkce - notifikace a zastupování. **Notifikace** může být navázána na většinu elementů v rámci procesu. Do notifikace lze přidat informace o probíhajícím procesu, procesní proměnné, vykonavetele aktuálního úkolu apod. Notifikace se po vygenerování odešle danému uživateli/uživatelům. Pokud uživatel, který má přiřazený úkol, onemocní, odjede na pracovní cestu nebo z jiného důvodu nemůže vyřídit úkol, je možné využít funkce **Zastupování**. BPMS obsahuje nástroje, díky kterým je v takovém případě úkol předán zastupujícímu uživateli, který ho může vyřídit.

- **Monitoring procesů.** Procesní aplikace obsahuje on-line informace o všech probíhajících instancích procesů v reálném čase. Lze sledovat aktuálně aktivní úkoly, procesní proměnné a historii vykonaných úkolů (kdo je vykonal, kdy byly vykonány, ...). Díky této možnosti lze ihned reagovat na případné problémy v průběhu procesů.

- **Zlepšování procesů.** Díky zavedení BPMS v rámci organizace lze rychle reagovat na požadavky na změnu procesu. Změny, na které byly dříve potřeba dny, týdny nebo měsíce času, lze díky BPMS vykonat za několikanásobně kratší dobu.

1.4 Notace

Tato kapitola popisuje notace užívané v rámci procesního řízení. Hlavní je notace BPMN, aktuálně využívaná ve verzi 2.0, která slouží pro modelování samotných procesů. Dále jsou zmíněny notace DMN a CMMN, které jsou pro tuto práci spíše okrajové.

1.4.1 BPMN

BPMN (Business Process Model and Notation) je notace sloužící k modelování procesů. Hlavní předností této notace je vysoká míra srozumitelnosti i pro lidi, kteří nejsou zaměřeni na IT. V rámci této notace lze zachytit kompletní průběh procesu, který lze následně nasadit do procesního enginu a spustit.

BPMN obsahuje čtyři základní prvky, ze kterých se skládají procesy a dále artefakty, které nemají vliv na průběh procesu. Mezi prvky s vlivem na průběh procesu patří úkoly (činnost, task), události (event), brány (gateway) a spojovací objekty (connector). Zdroj této kapitoly: [5]

- **Úkoly** jsou jednotlivé kroky procesu, které je potřeba v rámci průběhu procesu vykonat. Existuje několik typů úkolů, mezi základní typy patří uživatelský úkol (human task), procesní úkol (process task). Uživatelský úkol obvykle obsahuje formulář a vyžaduje zásah uživatele. Procesní úkol je vykonán automaticky informačním systémem. Mezi další typy úkolů patří například úkol s možností odeslání zprávy nebo skriptovací úkol pro vykonání scriptu.

- **Události** přímo ovlivňují průběh procesu. Mohou proces začít, skončit nebo jinak ovlivnit (zpoždění, ukončení, ...). Existuje několik typů událostí, například časová událost (timer event) pro zpoždění procesu o danou dobu nebo emailová událost (mail event), která může spouštět proces na základě emailu.

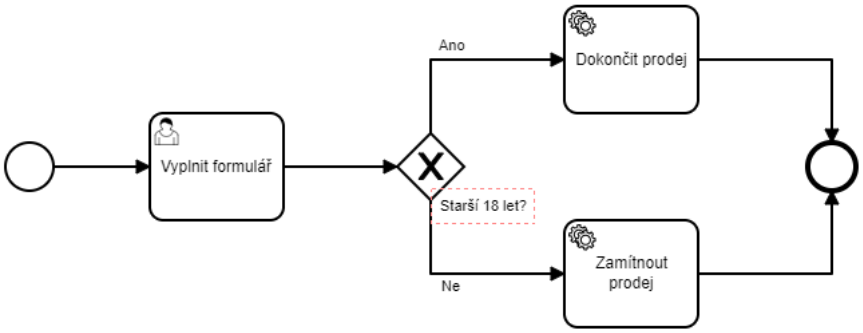
- **Brány** se v rámci BPMN značí čtvercem či kosočtvercem a mají dvě základní funkce - rozbíhání a spojování toku procesu. Rozbíhání obvykle rozvětvuje proces do více paralelních větví a na jednotlivé větve lze aplikovat podmínky určující pokračování procesu danou větví. Spojovací brány spojují paralelní větve zpět do hlavního toku procesu.

- **Spojovací objekty** slouží pro spojování předešlých objektů v rámci procesu a mají tři typy - sekvenční tok, tok zpráv a asociaci. Sekvenční tok určuje pořadí aktivit v rámci procesu a značí se nepřerušovanou čárou. Tok zpráv se značí přerušovanou čárou s prázdnou šipkou a určuje tok zpráv mezi dvěma účastníky procesu. Asociace spojuje objekty s dodatečnými informacemi a značí se přerušovanou čárou.

- **Artefakty** označují doplňující informace, ale nemají vliv na průběh procesu. Jsou to například poznámky, které lze přidat do procesu z důvodu uvedení dodatečných informací nebo datový objekt, který poskytuje informaci o tom, že nějaká aktivita pracuje s daty.

Následující obrázek zobrazuje příklad jednoduchého diagramu vytvořeného touto notací. Vlevo proces začíná startovací událostí. Následuje uživatelský úkol „Vyplnit formulář". V závislosti na vyplnění formuláře následná brána rozhodne, zda proces

pokračuje vrchní nebo spodní cestou. Obě tyto cesty obsahují systémové úkoly, které se automaticky provedou a proces je zakončen konečnou událostí nejvíce vpravo.



Obrázek 1.1: Příklad jednoduchého procesního diagramu vytvořeného notací BPMN 2.0. Zdroj: autor.

1.4.2 Další související notace s BPMN

- **DMN** je notace poskytující možnost modelovat organizační rozhodnutí, která mohou být díky DMN znázorněna v diagramech a případně automatizována. V rámci specifikace DMN je možnost provázání s BPMN modely. Rozhodnutí v rámci procesu jsou znázorněna decision requirements diagramem, který využává DMN notaci.[6]
- **CMMN** je grafická notace používaná pro zachycení pracovních metod, které jsou založeny na udržení aktivit, které mohou být vykonávány v nepředvídatelném pořadí v závislosti na vývoji situace. CMMN rozšiřuje možnosti BPMN. [7]

1.5 Camunda

Camunda je framework založený na jazyku java. Podporuje BPMN pro tvorbu procesních diagramů, CMMN pro řízení případů a DMN pro řízení business rozhodnutí. Tato kapitola poskytuje přehled o licencích Camundy, dále je popsána infrastruktura jednotlivých částí a závěr této kapitoly je věnován architektuře. Zdroj celé kapitoly:[8]

1.5.1 Licence

Camunda nabízí dvě edice svého BPMS - comunitní a enterprise. **Komunitní edice** je kompletně open-source a obsahuje základní BPMS. V rámci komunitní edice jsou obsaženy BPMN engine, DMN engine, Modeler, Tasklist a kokpit se zákadní funkcionalitou (více o těchto částech v kapitole 1.4.2 - Infrastruktura). **Enterprise edice** obsahuje všechny části z komunitní edice a k tomu navíc rozšířenou funkcionalitu kokpitu, optimalizační nástroje a kompletní podporu včetně školení.

1.5.2 Infrastruktura

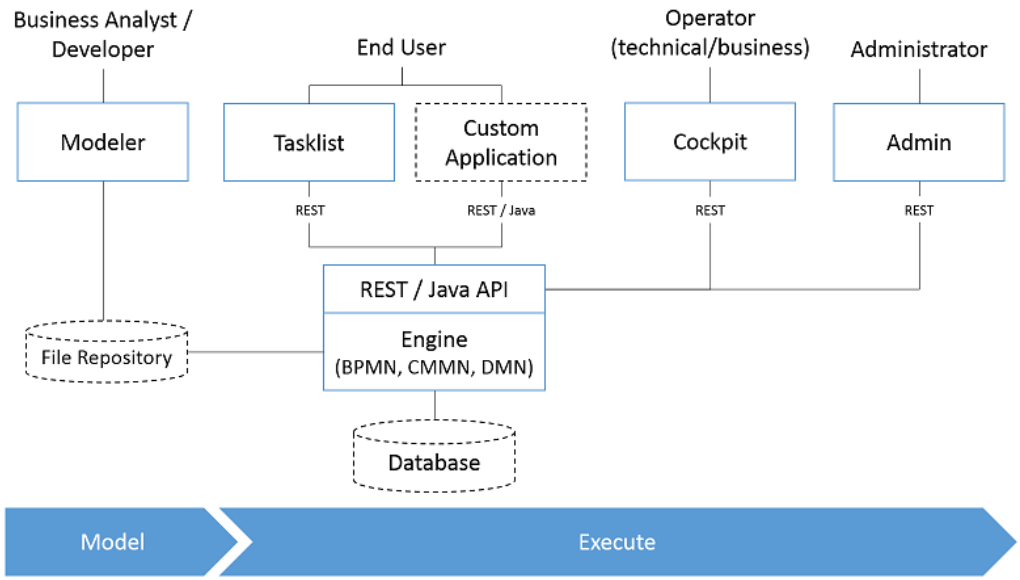
Infrastrukturu camundy tvoří několik částí. Pro integraci s java aplikacemi slouží Procesní engine, modelování diagramů zajišťuje Camunda modeler a bpmn.io. Pro správu procesů camunda nabízí webovou aplikaci.

- **Procesní engine** je knihovna určená pro java aplikace a nabízí sadu nástrojů pro integraci s Camundou. Mezi hlavní funkce této knihovny patří možnost spouštět procesy, vytvářet procesní proměnné, potvrzovat dokončení aktuálních procesních úkolů a sbírat data o aktuálně běžících procesech. Dále tato knihovna zahrnuje možnosti sestavovat dotazy do databáze v rámci procesních instancí a historie.
- **Camunda modeler** je aplikace umožňující modelování diagramů v rámci BPMN 2.0, CMMN 1.0 a DMN 1.3. Navíc umožňuje také přímé nasazování modelů do Camundy a jejich následné spouštění.
- **Bpmn.io** se skládá z několika javascriptových knihoven. Slouží pro prezentaci BPMN 2.0, CMMN 1.0 a DMN 1.3 v rámci webových stránek. Dále umožňuje také diagramy a tabulky na webu přímo modelovat.
- **Webová aplikace** slouží pro kompletní přehled o procesech. Skládá se ze tří hlavních částí - Tasklist, Cockpit a Admin. S webovou aplikací lze kromě Procesního enginu komunikovat také přes REST API.
  - **Camunda Cockpit** umožňuje přehledně sledovat všechny aktuálně běžící procesy. U každé procesní instance je umožněn náhled na procesní diagram se zvýrazněním aktuálního úkolu. Dále obsahuje přehled proměnných uložených v rámci procesní instance a případné incidenty, pokud došlo při průběhu procesu k chybě.
  - **Camunda Tasklist** obsahuje přehledný seznam aktuálně aktivních uživatelských úkolů. Úkoly lze jednoduše řadit nebo vyhledávat. Po zobrazení detailu úkolu se zobrazí formulář přiřazený danému úkolu, který lze vyplnit a potvrzením posunout proces k dalšímu kroku.

  - **Camunda Admin** je administrátorská sekce camundy umožňuje správu přístupových práv uživatelských účtů a uživatelských skupin v rámci Camundy.
  - **REST API** umožňuje komunikaci s webovou aplikací. Jeho cílem je poskytnout přístup do všech rozhraní enginu. Pro komunikaci přes REST API musí být uživatel autorizován. Rozhraní umožňuje kompletní správu webové aplikace - administrace i procesů. Lze přes něj získávat data o procesech a procesních proměnných, spouštět a ukončovat procesy i zjišťovat historii.

- **Ukládání dat** probíhá v rámci databáze. Camunda neposkytuje vlastní databázové řešení pro ukládání dat. Obsahuje však možnost integrace všech předních databázových systémů (Microsoft SQL, MySQL, PostgreSQL, Oracle, ...). Po napojení Camundy k dané databázi dojde k automatickému vytvoření databázového schéma, kam jsou následně ukládána procesní data a data potřebná pro chod Camundy.

Následující obrázek graficky znázorňuje výše zmíněné části v rámci infrastruktury camundy.



Obrázek 1.2: Infrastruktura Camundy. Zdroj: [8]

1.5.3 Architektura

Hlavní komponenty Camundy jsou napsány v programovacím jazyku Java. Camunda se zaměřuje především na Java vývojáře, kterým se snaží dodat všechny nástroje potřebné pro design, implementaci a běh procesů v rámci JVM. Ale i přes hlavní zaměření na Java vývojáře se Camunda snaží být dostupná pro všechny. Z toho důvodu obsahuje také REST API pro řízení procesů. Camunda může být použita jako samostatný procesní engine na serveru i jako součást Java aplikace.

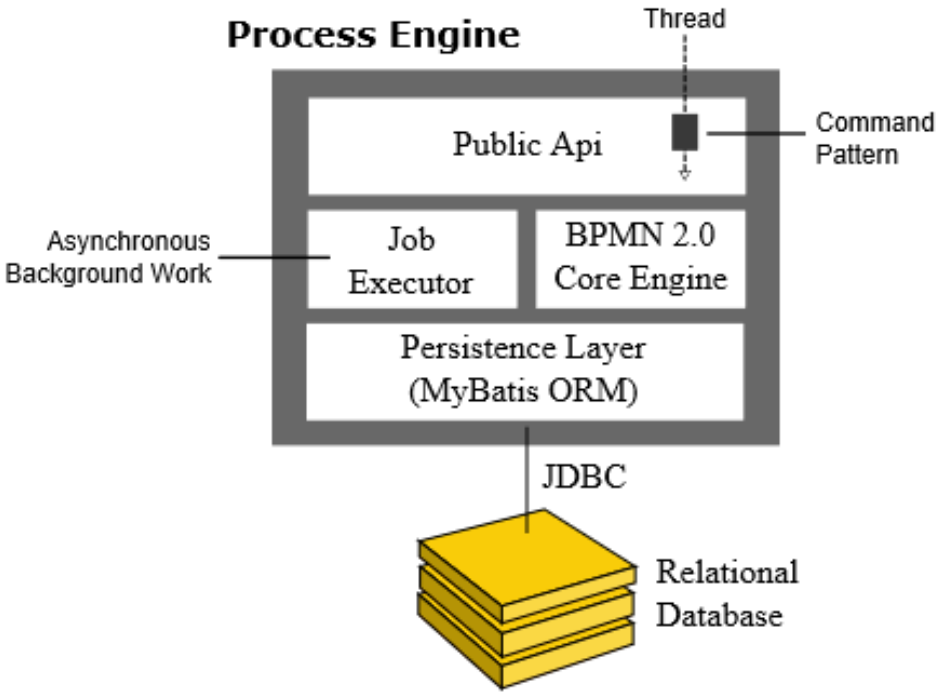
- **Architektura procesního enginu**

– **Veřejné aplikační rozhraní (Public API)** umožňuje Java aplikacím interagovat s procesním enginem. Procesní engine má několik různých odpovědností (procesní repozitář, řízení úkolů v procesu, řízení běhu procesů, ...) a všechny tyto odpovědnosti jsou rozděleny do vlastních služeb. Jednotlivé interakce s procesním enginem jsou moderovány do kontextu vláken ve stylu transakcí.

– **BPMN 2.0 engine** je jádro procesního enginu. Umožňuje spouštění enginu pro grafy a BPMN 2.0 parsovacího nástroje, který transformuje BPMN 2.0 XML soubory na java objekty.

– **Job executor (vykonavatel práce)** je odpovědný za zpracování asynchronní práce na pozadí - např. časovače.

– **Persistentní vrstva** je odpovědná za ukládání stavu procesních instancí do relační databáze. Pro objektové mapování je použit MyBatis mapping engine.



Obrázek 1.3: Grafické znázornění architektury procesního enginu. Zdroj: [8]

- **Architektura platformy Camunda.** Platforma BPM Camunda je framework, který je flexibilní a lze ho využívat různými způsoby. Zde jsou uvedeny nejčastěji užívané způsoby použití.

  – **Vložený procesní engine** - procesní engine je použit v rámci aplikace jako knihovna. Tento způsob umožňuje jednoduchá spouštění a zastavení procesů v rámci běhu aplikace. Je možné spouštět více vložených procesních enginů v rámci sdílené databáze.

- **Sdílený kontejnerově řízený procesní engine** - procesní engine je spuštěn v rámci kontejneru (application server, ...) a je poskytován jako služba - lze být sdílen všemi aplikacemi nasazenými v daném kontejneru.
- **Samostatný server s procesním enginem** - procesní engine je poskytován jako síťová služba. Aplikace s přístupem na internet mohou využívat tento procesní engine. Nejjednodušší cesta využití daného enginu je přes REST API.
- **Clustering model (shlukovací model)** - vhodný při potřebě zajistit škálovatelnost nebo zvýšenou odolnost vůči poruchám. procesní engine je poskytován ostatním uzlům v clusteru (shluku) a každá instance procesního enginu je napojena na společnou sdílenou databázi. Díky společné databázi mohou být jednotlivé requesty zasílané do enginu vyhodnocovány různými procesními instancemi - možnost distribuce requestů na méně zatížené uzly a pokud nastane chyba v některém uzlu, může být nahrazen ostatními uzly.

Kapitola 2

Praktická část

V rámci této kapitoly je uveden postup praktické části práce. Nejdříve jsou upřesněny cíle praktické části. Dále je uvedena analýza, kde je čtenář seznámen s implementačními možnostmi, které jsou uvažovány jako možné téma. Tyto možnosti jsou následně upřesněny do konkrétního tématu práce.

Dále se práce věnuje tvorbě procesu, na kterém je implementované rozšíření demonstrováno. Po ukázce procesu následuje část popisující průběh implementace vybraných rozšíření a propojení s procesem.

Na závěr praktické části je implementované rozšíření důkladně otestováno a jsou zhodnoceny přínosy praktické části práce.

2.1 Upřesnění cílů praktické části

Aby mohlo dojít k naplnění cílů této závěrečné práce, je žádoucí v tuto chvíli upřesnit vše, na co by se měla zaměřit praktická část. Z toho důvodu v následujícím textu dojde k:

1. **Analýza a identifikace možných rozšíření do platformy Camunda (kapitola 2.2)**.
2. **Implementace vybraných rozšíření (kapitoly 2.5, 2.6)**.
3. **Otestování implementovaných rozšíření (kapitola 2.9)**.
4. **Vyhodnocení přínosů implementovaných rozšíření (kapitola 2.10)**.

2.2 Analýza

Tato kapitola popisuje postup při analýze implementační části práci. Nejprve jsou zkoumány možnosti, které by bylo možné implementovat v rámci této práce. Dále jsou tyto možnosti upřesněny v konkrétní výběr tématu.

2.2.1 Analýza

První část zahrnuje upřesnění možností, které by bylo vhodné implementovat. Jedná se o širší specifikaci prvotních myšlenek, která je upřesněna do konkrétní podoby v následujících kapitolách.

Analýza se nejprve zaměřuje na Camunda Modeler a Camunda Webapps, které tvoří funkční celek ekosystému Camunda pro tvorbu a sledování průběhu procesů. Dále je zaměřena na blog Camundy [9], kde lze najít mnoho inspirativních článků.

Rozšíření Camunda webapps

Jako první jsou zkoumána možná rozšíření v rámci aplikace Camunda webapps. Aby bylo možné plně analyzovat rozšíření Camunda webapps, je nutné analyzovat průběh spuštěného elektronizovaného procesu a v rámci porůběhu posoudit možná rozšíření. - Dále možnost 1.

Možnosti rozšíření:

- Vylepšení prezentace JSON [1] proměnných - Camunda webapps umožňuje použít proměnné typu JSON, ale neumožňuje strukturované zobrazení[2] pro lepší čitelnost.

Vyhodnocení přínosů:

- Strukturované zobrazení proměnných typu JSON pro lepší čitelnost lidmi.

Odhad pracnosti implementace:

- Seznámení se zdrojovým kódem Camunda webapps - 15 h.
- Implementace - 10 h.
- Testování - 5 h.
- Celkový odhad - 30 h.

Omezení:

- Nedostatečná reálná zkušenost s Camundou pro určení nových přínosných funkcionalit v rámci Camunda webapps.

[1] JSON - JavaScript Object Notation je hojně využívaný formát pro výměnu dat. Jeho výhody jsou jednoduchá strojová generovatelnost a zároveň možnost číst člověkem [10]

[2] V rámci průběhu procesu lze zobrazit data proměnných - pokud je potřeba zobrazit data JSON proměnné v Camunda webapps, je strukturovaná forma přehlednější a z toho důvodu čitelnější pro lidi

Rozšíření Camunda modeleru

Dále je zkoumána aplikace Camunda modeler. Analýza probíhá v rámci tvorby BPMN Diagramu. V průběhu tvorby se analyzují možnosti rozšíření této aplikace. - Dále možnost 2.

Možnosti rozšíření:

- Pole typu soubor v Camunda forms - Camunda forms nabízejí možnost „naklikat" formulář přímo z užiatelského rozhraní Camunda modeleru, ale v rámci seznamu možných typů pole není typ soubor.

Vyhodnocení přínosů:

- Přínos tohoto rozšíření je umožnění tvorby „klikatelných formulářů" s poli pro nahrávání souborů.

Odhad pracnosti implementace:

- Seznámení se zdrojovým kódem Camunda modeler - 15 h.
- Implementace funkcionality v Camunda modeler - 20 h.
- Seznámení se zdrojovým kódem Camunda webapps - 15 h.
- Implementace použití funkcionality v Camunda webapps - 100 h.
- Testování - 10 h.
- Celkový odhad - 160 h.

Omezení:

- Při rozšíření možností formulářových polí v rámci modeleru je nutné upravit i další části ekosystému Camunda - zobrazení a použití daných polí v rámci Camunda webapps.

Možnosti z blogu Camunda

Další možnosti rozšíření přináší blog společnosti Camunda, kde lze najít mnoho inspirativních článků. [9] Při procházení těchto článků je objeven článek o externích (remote) workerech[3], které nabízí rozšiřující logiku (popř. napojení na API[4] dalších služeb) v rámci elektronizovaného procesu. - Dále možnost 3.

Možnosti rozšíření:

- Tvorba externích (remote) workerů pro komunikaci Camundy s API externích služeb. [11]

[3]Externí (remote) worker je aplikace, která vykonává pro proces v Camundě dodatečnou logiku, případně komunikuje s API dalších služeb. Camunda v průběhu procesu zavolá daný worker, který vykoná vlastní funkcionalitu a vrátí zpracovaná data do procesu. [11]

[4]Application programming interface se používá při vývoji softwaru. Jedná se o rozhraní, které obsahuje funkce, pomocí kterých lze komunikovat s danou aplikací - obvykle vytvářet a získávat data. [12]

Vyhodnocení přínosů:

- Externí workery, které jsou využitelné v široké škále procesů a jednoduše přenositelné mezi procesy.
- Zasazení workerů do reálného procesu[5] - demonstrace možností Camundy.

Odhad pracnosti implementace jednoho workeru (práce obsahuje více než jeden):

- Seznámení s dokumentací a způsobem implementace - 10 h.
- Implementace potřebných funkcionalit - 30 h.
- Propojení s procesem - 5 h.
- Testování - 5 h.
- Celková odhadovaná pracnost vývoje jednoho workeru - 50 h.

Omezení:

- Nutné najít vhodné API, se kterými proces komunikuje pomocí daných externích workerů.

2.2.2 Výběr vhodných rozšíření

Tato kapitola obsahuje výběr vhodných rozšíření, která mají největší význam a jsou realizovatelná v rozsahu bakalářské práce. Zvolená implementace je následně realizována v rámci této práce.

Sumarizace a zhodnocení přínosů:

- Přínos možnosti 1 lze nahradit JSON strukturovacími nástroji dostupnými online.
- Možnost 2 přináší do Camundy kompletně novou funkci.
- Možnost 3 přináší tvorbu nových funkčních celků s dodatečnou logikou využitelnou v průběhu procesu.
- **Vyhodnocení** - S ohledem na míru přínosu výše identifikovaných možných rozšíření lze uvést následující sestupné seřazení jednotlivých variant - 1, 3, 2.

Rozsáhlost implementace:

- Možnost 1 obsahuje nejlehčí implementaci z uvedených možností - úprava prezentace JSON formátu - odhad 30 h.
- Možnost 2 obsahuje implementaci formulářového pole a dále úpravu Camunda webapps - náročná implementace - odhad 160 h.
- Možnost 3 je v rámci rozsáhlosti implementace úměrná množství implementovaných externích workerů - odhad 50 h / worker.

[5]Je zapotřebí vytvořit reálný proces pro demonstraci funkcionalit implementovaných workerů

- **Vyhodnocení** - Možnost 3 je vhodná vzhledem k možnosti úpravy náročnosti podle počtu externích workerů.

Následuje tabulka vyhodnocení. Jednotlivé řádky obsahují číslo varianty, dále vyhodnocení přínosů na stupnici 1 - 10 (10 nejlepší) a časovou náročnost.

Varianta	**Přínosnost**	**Časová náročnost**
1	5	30h
2	8	160h
3	9	50h/worker

Tabulka 2.1: Tabulka shrnutí možností implementace

Na základě vyhodnocení přínosů a rozsáhlosti implementace je vybrána **tvorba externích workerů**. Tato volba je z důvodu přínosnosti, která zahrnuje použitelnost v široké škále procesů a zároveň jednoduchou přepoužitelnost mezi procesy. Navíc je tato možnost realizovatelná v rozsahu bakalářské práce. Externí workery jsou zasazeny do procesu, který demonstruje jejich funkcionalitu.

2.3 Tvorba procesu vybraného rozšíření

Tato kapitola obsahuje nejprve výběr vhodných API, dále sestavení tématu procesu na základě vybraných API, use-case diagram (diagram případů užití) a tvorbu BPMN diagramu. Tvorba BPMN diagramu procesu probíhá v aplikaci Camunda modeler.

2.3.1 Výběr vhodných API

Aby bylo možné implementovat možnost 3, je nutné specifikovat API, které jsou použity v rámci externích workerů. API jsou hledány v rámci webu AnyAPI [13]

Vlastnosti, které musí dané API splňovat:

1. Použití zdarma - v rámci práce nejsou k dispozici finanční zdroje.
2. Být veřejně dostupné.

Byly vybrány tyto API:

- OMDB API - OMDB je databáze filmů a seriálů. Nabízí API pro vyhledávání podle názvů, roku vydání, žánrů apod. Pro využití maximálně 1 000 requestů (požadavků) denně je zdarma. [14]
- Google API - Google je používán na celém světě a obsahuje mnoho služeb. V rámci většiny služeb googlu existuje také API pro správu. V rámci této práce je použito API pro Google sheets (tabulky google) [15], Google Drive API (tvorba tabulek na google disku) [16] a Gmail API (odesílání emailů). [17]

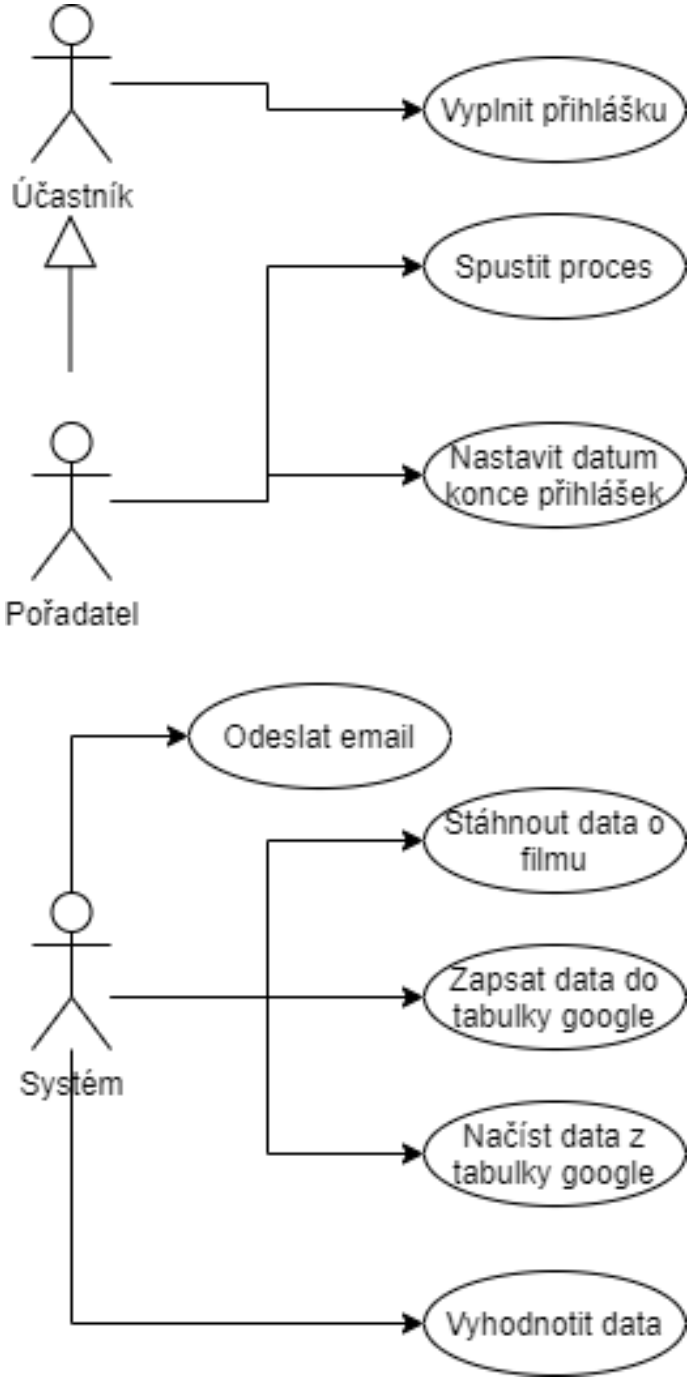
2.3.2 Sestavení tématu procesu

Na základě vybraných API v rámci analýzy se nabízí tvorba procesu **Pořádání filmového večera**. Předpokládaný průběh procesu:

1. Pořadatel spustí proces a nastaví datum konce přihlašování.
2. Zájemce o filmový večer vyplní formulář (email, návrh na film) v rámci druhého procesu, jehož odeslání spustí událost (event) v procesu.
3. Systém stáhne data o filmu z OMDB API.
4. Systém zanese daného uživatele do tabulky členů (Google sheets API) a zadaný film do tabulky filmů (Google sheets API) (se všemi informacemi - popis, hodnocení apod.).
5. Po skončení přihlášek na filmový večer systém stáhne tabulku filmů (Google sheets API) a zvolí film s nejvyšším hodnocením.
6. Systém rozešle všem účastníkům email (Gamil) s informací o zvoleném filmu.
7. Konec procesu.

2.3.3 Use-case diagram (diagram případů užití)

Tato kapitola obsahuje use-case diagram sestavený na základě procesu z předchozí kapitoly.



Obrázek 2.1: Use-case diagram vytvořený v nástroji draw.io. Zdroj: autor.

2.3.4 Tvorba BPMN diagramů

V rámci této kapitoly je uveden postup tvorby BPMN diagramů hlavního procesu a procesu přihlášek. Nejprve jsou určeny jednotlivé části, které jsou následně sestaveny do výsledných diagramů.

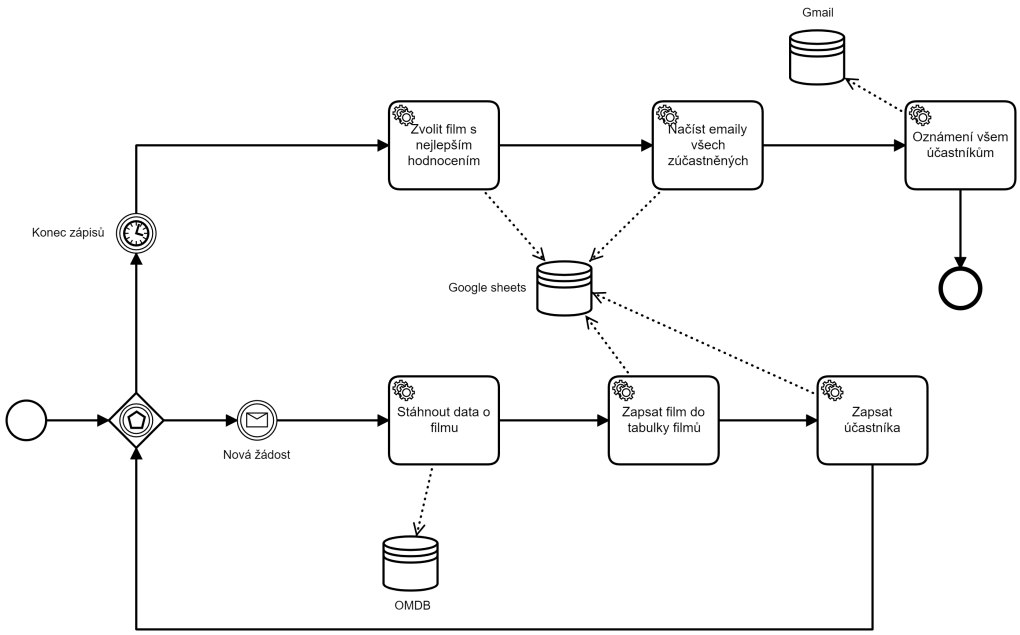
Postup

1. Identifikace úkolů uživatele.
2. Identifikace úkolů služby a emailu (service task, send task).
3. Identifikace bran (gateway).
4. Identifikace událostí (event).
5. Sestavení procesu (BPMN diagram).

- **Úkoly uživatele** - vyplnění formuláře v procesu přihlášek
- **Úkoly služby a emailu** (service task, send task):
  - Stáhnout data o filmu.
  - Zapsat film do tabulky google.
  - Zapsat účastníka do tabulky google.
  - Vyhodnotit film s nejlepším hodnocením.
  - Načíst emaily účastníků z google tabulky.
  - Odeslat emaily účastníkům s oznámením o vybraném filmu.
- **Události** (event):
  - Událost zprávy (message event) - spustí část procesu pro zaznamenání jedné přihlášky. Je spuštěna vyplněním formuláře z procesu přihlášek.
  - Časová událost (timer event) - spustí část procesu pro vyhodnocení filmů a odeslání emailu účastníkům. Je spuštěna po vypršení časového limitu pro přihlášky.
- **Brány** (gateway):
  - Brána událostí (event gateway) - mezi událostmi z předchozího bodu. Před vypršením časového limitu na přihlášky je aktivní část pro přijímání přihlášek. Po vypršení limitu je spuštěna část pro vyhodnocení.

2.3.5 Stavba BPMN diagramu hlavního procesu

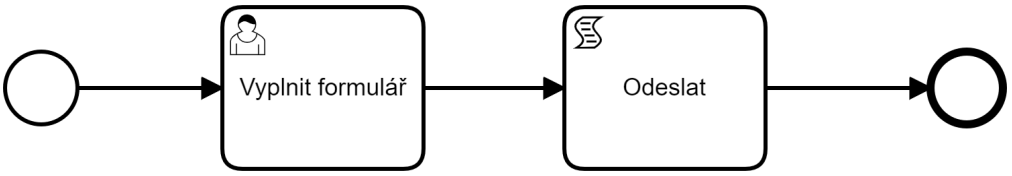
Části z předchozí kapitoly jsou sestaveny do výsledného BPMN diagramu. Také jsou v diagramu uvedeny služby, se kterými jednotlivé úkoly (task) komunikují pomocí externích workerů vytvořených v rámci této práce.



Obrázek 2.2: BPMN diagram hlavního procesu. Zdroj: autor.

2.3.6 Stavba BPMN diagramu přihlášek

V rámci úkolu uživatele je vyplnění formuláře, který spustí podproces zpracování přihlášek v hlavním procesu.



Obrázek 2.3: BPMN diagram procesu přihlášek. Zdroj: autor.

2.4 Externí workery

Tato kapitola obsahuje implementaci společnou pro všechny implementované workery. Implementace specifické pro jednotlivé workery jsou uvedeny v následujících kapitolách. Společné vlastnosti:

- Worker musí obsahovat název, pomocí kterého je následně volán z procesu - názvy jednotlivých workerů jsou uvedeny ve zdrojových kódech pod názvem TOPIC_ID.
- Při spuštění workeru musí být spuštěna init metoda, která umožní Camundě daný worker využívat
- Při spuštění workeru musí být zadána v argumentech URL adresa REST rozhraní spuštěné Camundy a číslo, které určí počet spuštěných workerů.
- V rámci init metody je vytvořen „naslouchač událostí", který je aktivní po celou dobu provozu workeru a čeká na zavolání z procesu. Pokud je worker zavolán, naslouchač spustí specifikovanou metodu.
- V rámci dané metody lze přistupovat k informacím o procesu vč. procesních proměnných.
- Po skončení práce workeru je zapotřebí potvrzení, že je úkol hotový, aby mohl proces pokračovat na další úkol. V tomto kroku lze také v procesu vytvářet nové proměnné.

2.5 OMDB Worker

Jako první je implementován worker komunikující s OMDB API. Tato kapitola obsahuje nutné kroky pro aktivaci OMDB API a následně průběh implementace daného workeru.

2.5.1 Aktivace OMDB API

Pro funkčnost tohoto workeru je zapotřebí aktivace na webových stránkách OMDB API[6]. Postup:

1. Přejít na kartu API Key.
2. V panelu Generate API Key zaškrtnout FREE account type.
3. Vyplnit email -> Sumbit.
4. Na email z bodu 3 příjde vygenerovaný API Key.

2.5.2 Implementace

V rámci implementace OMDB API je zapotřebí seznámení s obsluhou daného API. Tato kapitola obsahuje postup při vytváření daného workeru.

Zabezpečení OMDB API

OMDB API autentizuje requesty (požadavky) pomocí vygenerovaného API Key z předchozí kapitoly, který je vložen jako parametr do URL. V rámci free aktivace je možné odeslat až 1000 requestů (požadavků) denně. Vygenerovaný apikey je nutné vložit do proměnné apikey v OMDBApi.java v projektu.

OMDB

- OMDB API umožňuje stahování informací o filmech podle názvu, který účastník zadal v rámci přihlášky.
- Podle názvu filmu zadaného účastníkem je sestaven request (požadavek) a odeslán na OMDB API.
- OMDB API odpovídá ve formátu JSON a obsahuje data o filmu nebo chybovou hlášku, pokud byl například název nevalidní.
- JSON vrácený z OMDB API je uložen v rámci procesu, aby mohl být následně zpracován Google Services workerem.

[6]Webová adresa stránek http://www.omdbapi.com/

2.6 Google Services Worker

Dále jsou implementovány externí workery pro google services. Tato kapitola obsahuje nejprve nutné kroky pro aktivaci přistupování k API v rámci Google účtu. Dále je uveden průběh implementace daných workerů.

2.6.1 Google účet

Externí worker komunikující s Google službami se neobejde bez Google účtu. Google účet lze založit přímo na webových stránkách Googlu.

Pro použití Google API musí být v rámci Google účtu tato funkce aktivována. Tato kapitola obsahuje postup aktivace služeb, které jsou nutné pro funkčnost implementovaného workeru.

Aktivace Google API probíhá v rámci Google Cloud Platform Console[7], kde je zapotřebí přihlášení Google účtem. Pro tento projekt je zapotřebí aktivovat API pro **Google Sheets, Google Drive a Gmail** s právy pro čtení i změnu dat.

Stručný postup:

1. V rámci console pro APIs and Services založit nový projekt.
2. Přejít na záložku Credentials -> CONFIGURE CONSENT SCREEN -> vyplnit informace o aplikaci a kontaktní email -> SAVE AND CONTINUE.
3. ADD OR REMOVE SCOPES -> Vyhledat Google Drive API a zaškrtnou scopes .../auth/docs a .../auth/drive -> SAVE AND CONTINUE.
4. ADD USERS -> vložit emaily uživatelů, kteří mohou používat dané api dokud je status Testing [8] -> SAVE AND CONTINUE.
5. V projektu z bodu 1 zvolit ENABLE APIS AND SERVICES a vybrat Google Drive API -> ENABLE.
6. V dalším oknu dát Create credentials - nutné pro možnost komunikace s API (vyplnit using Google Drive API a accessing User data) -> NEXT.
7. Opět se objeví výběr scopes, mělo by tam být předvyplněno Google Drive API (pokud není -> vybrat viz. bod 3) -> SAVE AND CONTINUE.
8. Vybrat application type Desktop app -> CREATE.
9. Dojde k vygenerování credentials - > stáhnout tlačítkem DOWNLOAD.
10. Enable Google Sheets API a Gmail API stejným způsobem jako Google Drive API (credentials již není nutné vytvářet).

Vygenerované credentials slouží jako autentizace, z toho důvodu je zapotřebí jejich vložení do resources[9] v rámci projektu.

[7] Dostupné na adrese https://console.developers.google.com/?hl=cs

[8] Nově založený projekt je automaticky ve fázi testování a mohou ho používat pouze uživatelé zařazení mezi Test Users

[9] Konkrétní složka /resources/cz/cvut/fel/nacovfil/googleservicesworker/api

2.6.2 Implementace

V rámci implementace Google services workeru je zapotřebí seznámení s obsluhou tří různých API - Google Drive API, Google Sheets API a Gmail API. Tato kapitola obsahuje postup při vytváření daného workeru.

Zabezpečení Google API

Přes API lze zasahovat do dat užívatele, z toho důvodu je nutná autentizace google účtu v rámci komunikace s API. Autentizace probíhá pomocí credentials vygenerovaných v rámci předchozí kapitoly.

Další informace ohledně zabezpečení:

- Pokud je aplikace v rámci Google Cloud Console vedena ve fázi testování, mohou ji využívat pouze uživatelé uvedení mezi Test users.
- Pokud se aplikace poprvé (nebo po delší době) připojuje na daný google účet, automaticky dojde k otevření formuláře, v rámci kterého je nutné přihlášení do daného Google účtu a potvrzení důvěryhodnosti aplikace.
- V rámci jednotlivých API lze definovat scopes (rozsahy). Pomocí scopes lze určovat práva aplikace. V rámci tohoto workeru jsou umožněna všechna práva, aby bylo možné vytvářet soubory, psát do tabulek a odesílat emaily, ale je možné například umožnit aplikaci data (soubory, emaily) pouze číst.

Google Drive

- Google Drive API je zapotřebí pro vytváření nových tabulek Google, do kterých se následně automaticky zapisují získaná data.
- Názvy tabulek jsou generovány s aktuálním datem a náhodným řetězcem znaků délky 32, aby nedocházelo k duplicitním názvům.
- Po vytvoření tabulky je vygenerován unikátní identifikátor, pomocí kterého lze následně k tabulce přistupovat. Tento identifikátor je uložen do procesu a slouží k následnému zapisování dat do tabulky.

Google Sheets

- Google sheets API je zapotřebí pro zapisování a čtení dat z Google tabulek.
- Pokud ještě nebyla vytvořena tabulka pro zápis filmů nebo účastníků, je zavolána funkce pro tvorbu tabulek a tabulka se vytvoří.
- K jednotlivým tabulkám je přistupováno pomocí identifikátoru, který je po vytvoření tabulky uložen do procesu.
- Při zapisování je nutné určit, zda se budou data v tabulce novými daty přepisovat nebo budou data doplňována na nové řádky - v rámci tohoto projektu jsou data doplňována na nové řádky.
- Zápis dat:

  - Pro zápis dat do tabulky je zapotřebí z dat vytvořit seznam seznamů objektů, který si lze představit jako seznam řádků s daty pro jednotlivé buňky tabulky.
  - Z procesu je dostupný JSON daného filmu. JSON je nutné nejprve převést na Java objekt a následně z tohoto objektu vytvořit seznam objektů reprezentujících řádek tabulky, který je zasazen do dalšího seznamu (celkově reprezentující seznam seznamů objektů z předchozího bodu)
  - Pokud film neexistuje, není do tabulky zapsáno nic.
- Čtení:
  - Při čtení dat je vrácen seznam seznamů objektů reprezentující jednotlivé řádky tabulky.
  - Při vyhodnocení filmů je ze seznamu z předchozího bodu zapotřebí získat data reprezentující hodnocení jednotlivých filmů a následně seřadit filmy podle hodnocení -> vítezný film je uložen do procesu.
  - Při načtení účastníků jsou načteny jejich emaily z tabulky účastníků -> uložení seznamu emailů do procesu.

Gmail

- Gmail API je zapotřebí pro odeslání výsledků vyhodnocení filmů jednotlivým účastníkům. Ve zdrojovém kódu GmailApi.java je nutné vyplnit email google účtu, pro který je aktivováno API.
- Z procesu jsou načteny informace o vítězném filmu a seznam účastníků.
- Z dat z předchozího bodu je sestaven email:
  - Obsah emailu je vytvořen jako HTML[10] dokument, aby bylo možné podobu emailu stylovat (nastavení barvy, velikosti, tučnosti písma atd.).
  - Email je odeslán jako hromadný všem účastníkům najednou.
  - Pokud je tabulka filmů prázdná (například pokud účastníci zadali názvy neexistujících filmů), je odesláno upozornění o prázdné tabulce filmů.
  - Pokud je prázdný seznam účastníků, odeslání emailu neproběhne.

Z výše uvedených Google API jsou sestaveny dva workery. První worker implementuje Google Drive API a Google Sheets API. Druhý worker implementuje Gmail API.

[10]Hyper Text Markup Language - standardní značkovací jazyk pro vytváření webových stránek. Obsahuje elementy, pomocí kterých je tvořena struktura webu. [18]

2.7 Propojení jednotlivých částí

Tato kapitola obsahuje průběh propojení jednotlivých částí do funkčního celku. Nejprve je uvedeno propojení hlavního procesu s jednotlivými externími workery. Dále je uveden postup propojení procesu přihlášek s hlavním procesem.

2.7.1 Propojení externích workerů s hlavním procesem

Tato kapitola obsahuje informace týkající se propojení procesu s externím workerem.

- Volání externích workerů probíhá v rámci service tasks (úkolů služby). Aby bylo možné volat externí worker, je zapotřebí znát jeho jméno, které je zadáno ve zdrojovém kódu při spuštění workeru.
- Pro volání externího workeru je zapotřebí nastavit External (externí) implementaci daného service task a do kolonky Topic vložit název workeru.
- Aby mohl být service task volající worker úspěšně dokončen, musí být worker spuštěný na správné URL adrese[11]. Pokud by spuštěný nebyl, proces se na daném tasku zastaví a čeká na spuštění workeru.

2.7.2 Propojení procesu přihlášek s hlavním procesem

Hlavní proces obsahuje část pro přijímání přihlášek na filmový večer. Tuto část je zapotřebí spouštět z procesu přijímání přihlášek a zároveň předat data z formuláře mezi těmito procesy. Nastavení spojení mezi procesy:

- **Hlavní proces** - V hlavním procesu toto spojení zprostředkovává Message event (událost zprávy). Pokud je tento event aktivní, čeká na případné zavolání. Jakmile je zavolán, je spuštěna část procesu navázána na tento event. Event je volán pomocí jména, které je u něj nastaveno.
- **Proces přijímání přihlášek** - v procesu pro přijímání přihlášek je po vyplnění formuláře zapotřebí zavolat Message event z hlavního procesu. Toto zavolání je prováděno pomocí scriptu. V tomto scriptu se vytváří korelace s Message event (pomocí jména) a následně se předávají data získaná z formuláře. Po předání dat je spuštěna část procesu navázána na Message event v hlavním procesu s daty z formuláře.

Po úspěšném propojení procesů a workerů je záměr této práce již funkční. V následující kapitole je uveden postup pro spuštění Camunda webapps, aby bylo možné celý proces nasadit, spustit a otestovat.

[11] Pokud je worker spuštěný na jiné URL adrese, Camunda nemá možnost ho volat.

2.8 Spuštění Camunda webapps

Tato kapitola obsahuje nejprve postup pro spuštění Camunda webapps. Následně je uveden postup nasazení procesního diagramu do Camundy, aby bylo možné proces spustit.

2.8.1 Camunda webapps

Pro potřeby této práce je spuštění Camunda webapps v rámci Dockeru[12]. Je využita verze Camunda webapps upravena[13] pro Docker, která je zdarma. Postup:

1. Stáhnout a nainstalovat Docker desktop[14].
2. Udělat pull Camunda webapps v rámci dockeru, příkaz: docker pull camunda/camunda-bpm-platform:latest.
3. Spustit Camunda webapps na portu 8080 (Pokud je tento port obsazen jinou aplikací, je zapotřebí změnit číslo portu), příkaz: docker run -d –name camunda -p 8080:8080 camunda/camunda-bpm-platform:latest.
4. Jakmile dojde ke spuštění, je možné v Dockeru vidět stav RUNNING u aplikace Camunda.
5. Pokud je aplikace ve stavu RUNNING, lze k ni přistupovat ve webovém prohlížeči[15].

2.8.2 Nasazení a spuštění procesů

Aby bylo možné spustit proces v rámci Camunda webapps, je zapotřebí tento proces nejdříve nasadit (nahrát do Camunda webapps). Postup pro nasazení procesu (tento postup je nutné provést pro hlavní proces i proces přihlášek):

1. Otevřít proces v aplikaci Camunda modeler.
2. Zvolit Deploy current diagram (ikonka nacházející se na horní liště druhá zprava).
3. Ve formuláři vyplnit deployment name a do kolonky vložit REST endpoint[16] Camunda webapps (aplikace musí být spuštěna).
4. Potvrdit tlačítkem Deploy (mělo by se zobrazit potvrzení Deployment succeeded).

[12]Docker je aplikace umožňující spouštění aplikací v kontejnerech, pokud aplikace vyžaduje konkrétní operační systém, server apod. V dockeru lze nastavit podmínky, které aplikace potřebuje, aniž by byl ovlivněn hlavní systém počítače. [19]

[13]Github projektu je na adrese https://github.com/camunda/docker-camunda-bpm-platform

[14]Na adrese https://www.docker.com/products/docker-desktop

[15]Na adrese http://localhost:8080/camunda/ (Pokud byl zvolen jiný port, je zapotřebí ho napsat místo 8080), přihlašovací jméno: demo, heslo: demo

[16]Pokud je Camunda spuštěna podle návodu z předchozí kapitoly na portu 8080, jedná se o adresu http://localhost:8080/engine-rest

Pokud je proces nasazen, je možné ho spustit. Postup pro spuštění hlavního procesu:

1. Otevřít záložku Tasklist ve webové aplikaci Camunda webapps.
2. Na horní liště zvolit Start process a ze seznamu vybrat hlavní proces.
3. Zvolit datum konce přihlášek (po vypršení data dojde k vyhodnocení) -> Start -> Zobrazení potvrzení o startu procesu -> Proces je možné sledovat v sekci Cockpit.

Pokud je spuštěn hlavní proces, je možné zasílat přihlášky. **Aby bylo možné přihlášky a následně získaná data vyhodnocovat, musejí být spuštěny také všechny externí workery vytvořené v rámci této práce**. Postup pro vyplnění přihlášky:

1. Spustit proces přihlášek dle návodu na spuštění procesu.
2. V sekci Tasklist v záložce All Tasks je formulář, který je možné nárokovat tlačítkem Claim v pravém horním rohu a následně vyplnit Jméno, email a název filmu.
3. Po odeslání formuláře jsou data předána hlavnímu procesu a vyhodnocena, zároveň jsou ihned zapsána do tabulek filmů a účastníků na Google drive.

Data z poslední obdržené přihlášky lze vidět také v procesních proměnných v Cockpitu.

2.9 Testování

Camunda webapps je spuštěna a procesy fungují. Poslední úkol v rámci této práce je Testování. Tato kapitola obsahuje sérii testovacích scénářu pro důkladné otestování vytvořených procesů a implementovaných externích workerů. Každý scénář obsahuje postup a vyhodnocení. Pro účely testování je nastaven Timer event na 3 minuty.

2.9.1 Testovací scénáře

Scénář 1 - žádná přihláška není podána

První testovací scénář zahrnuje vyhodnocení filmového večera bez podaných přihlášek.

Postup:

1. Spustit hlavní proces a počkat na vyhodnocení bez podaných přihlášek.

- **Očekávaný výsledek**: Proces skončí bez odeslání emailů (nejsou účastníci).

Scénář 2 - podání přihlášek s neexistujícími názvy filmů

Druhý testovací scénář zahrnuje vyhodnocení filmového večera s podanými přihláškami obsahujícími názvy filmů, které neexistují.

Postup:

1. Spustit hlavní proces.
2. Podat dvě přihlášky s neexistujícími názvy filmů.
3. Nechat proces vyhodnotit podané přihlášky.

- **Očekávaný výsledek**: Účastníkům je odeslána informace o prázdné tabulce filmů.

Scénář 3 - podání validních přihlášek

Třetí testovací scénář zahrnuje vyhodnocení filmového večera s podanými přihláškami obsahujícími validní data.

Postup:

1. Spustit hlavní proces.
2. Podat dvě přihlášky s existujícími názvy filmů.
3. Nechat proces vyhodnotit podané přihlášky.

- **Očekávaný výsledek**: Účastníkům je odeslána informace o vítězném filmu (filmu s nejvyšším hodnocením) se základními informacemi o filmu.

Scénář 4 - podání validních a nevalidních přihlášek zároveň

Čtvrtý testovací scénář zahrnuje vyhodnocení filmového večera s podanými přihláškami obsahujícími validní i nevalidní názvy filmů.

Postup:

1. Spustit hlavní proces.
2. Podat dvě přihlášky s existujícími názvy filmů.
3. Podat dvě přihlášky s neexistujícími názvy filmů.
4. Nechat proces vyhodnotit podané přihlášky.

- **Očekávaný výsledek**: Účastníkům je odeslána informace o vítězném filmu (filmu s nejvyšším hodnocením) se základními informacemi o filmu, nevalidní data jsou ignorována.

Závěr

V rámci testování jsou provedeny zmíněné testovací scénáře a nalezené chyby opraveny.

2.10 Vyhodnocení přínosů praktické části

- Jako nejlepší možnost rozšíření v rámci rozsahu této práce vyšly v rámci analýzy (kapitola 2.2) externí workery.
- V rámci rozšiřování možností Camundy jsou externí workery velmi účinné, protože hotový worker lze již jednoduše volat z jednotlivých procesů a pokud je vyžadována změna, je nutné ji provést pouze na jednom místě (ve zdrojovém kódu workeru).
- Další výhodou je jednoduchá přenositelnost workerů mezi zařízeními.
- V rámci Google services workeru (kapitola 2.6) je implementována komunikace s google službami, které jsou masově využívány na celém světě. Po drobné změně lze využít worker v jakémkoli procesu, který vyžaduje zapisovat a číst data z Google tabulek nebo odesílat emaily přes Gmail.
- V rámci OMDB workeru (kapitola 2.5) je implementována komunikace s OMDB API. OMDB worker je také využitelný v jakémkoli procesu vyžadujícím získávat informace o filmech nebo seriálech.
- Oba workery jsou úspěšně implementovány a v rámci práce demonstrovány na reálném procesu (kapitola 2.3.2).
- Kapitola 2.7 obsahuje postup propojení implementovaných workerů s procesem.
- Práce obsahuje také postup pro efektivní způsob provozu Camundy v rámci dockeru, tímto způsobem lze spustit camundu na jakémkoli zařízení bez nutnosti větších zásahů do systému (kapitola 2.8).
- Celý funkční celek je na závěr důkladně otestován (kapitola 2.9).

2.11 Možnosti dalšího rozšíření této práce

Praktická část práce je věnována tvorbě externích workerů komunikujících s externími službami. Toto téma nabízí spoustu možností rozšíření o další workery, které mohou opět komunikovat s externími API nebo vykonávat dodatečnou logiku využitelnou v průběhu procesu. Případně se dají rozšířit stávající workery o další funkcionality.

Závěr

Práce reaguje na jednu z potřeb moderní doby - nezatěžovat lidi se stereotypními činnostmi, které může vykonávat počítač. Toho lze dosáhnout pomocí elektronizace procesů.

Nejprve jsou rozebrány základy v rámci procesního řízení, které jsou nezbytnou znalostí pro elektronizaci procesů. Dále je čtenář seznámen se základy elektronizace procesů, životním cyklem procesů a notacemi nutnými pro elektronizaci procesů. Následuje již konkrétní open source software pro elektronizaci procesů - Camunda a informace ohledně této platformy.

Hlavním cílem této práce je analyzovat možná rozšíření plaformy Camunda a následně toto rozšíření implementovat a otestovat.

V rámci analýzy (kapitola 2.2) jsou nejprve vybrány různé možnosti rozšíření. Tyto možnosti jsou porovnány z různých hledisek a je z nich vybrána možnost implementace externích workerů jako rozšíření platformy Camunda, která vyšla z analýzy jako nejlepší.

Implementační část (kapitoly 2.5 a 2.6) obsahuje postup při implementaci jednotlivých externích workerů s informacemi o zabezpečení jednotlivých API, nastavením účtů apod.

Z důvodu efektivní demonstrace funkčnosti implementovaných workerů je vytvořen reálný proces (kapitola 2.3.2) s názvem Organizace filmového večera, ve kterém jsou všechny implementované workery úspěšně použity a otestovány.

Hlavní cíl práce - analyzovat možná rozšíření, následně je implementovat a důkladně otestovat, byl splněn. Dále z této práce vyplývá, že externí workery jsou vhodná cesta k rozšiřování platformy Camunda.

Autorovi tato práce přináší spoustu nových užitečných poznatků o platformě Camunda a elektronizaci procesů. Je velice zajímavé tvořit externí workery, které komunikují s tak masivně využívanými službami jako jsou Google tabulky nebo Gmail. Jedná se o mnoho nových získaných znalostí, které mohou být užitečné kdykoli v budoucnu.

Bibliografie

1. ŘEPA, Václav. *Procesně řízená organizace*. Praha: Grada, 2012. ISBN 978-80-247-4128-4.
2. HRONZA, Radek. *Řízení výkonnosti v akademickém prostředí*. Praha. Disertační práce, České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra ekonomiky, manažerství a humanitních věd, 2016.
3. *Everything You Need to Know About Business Process Automation*. 2020. Dostupné také z: `https://www.smartsheet.com/understanding-evolution-and-importance-business-process-automation`. (cit: 10.12.2020).
4. ŠABATOVÁ, Ivana. *Podnikové procesy pod kontrolou: Principy a přínosy implementace systémů pro řízení procesů (BPMS)*. Praha. Galeos a.s., [n.d.].
5. VAŠÍČEK, Petr. *Úvod do BPMN*. Dostupné také z: `http://bpm-sme.blogspot.com/2008/03/3-uvod-do-bpmn.html`. (cit: 25.10.2020).
6. STASIOWSKI, Dominik. *Elektronizace procesů pomocí BPM nástroje Camunda*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.
7. *What is Case Management Model and Notation (CMMN)*. Dostupné také z: `https://www.visual-paradigm.com/guide/cmmn/what-is-cmmn/`. (cit: 20.11.2020).
8. *The Camunda BPM Manual*. 2020. Dostupné také z: `https://docs.camunda.org/manual/7.14/`. (cit: 01.11.2020).
9. *Blog - Camunda*. Dostupné také z: `https://camunda.com/blog/`. (cit: 5.4.2021).
10. *Úvod do JSON*. Dostupné také z: `https://www.json.org/json-cz.html`. (cit: 16.4.2021).
11. RÜCKER, Bernd. *Remote workers and idempotency*. Dostupné také z: `https://camunda.com/blog/2017/08/remote-workers-and-idempotency/`. (cit: 5.4.2021).
12. KAĎOUSKOVÁ, Barbora. *Co je to API a jaké jsou možnosti jeho využití*. Dostupné také z: `https://www.rascasone.com/cs/blog/co-je-api`. (cit: 16.4.2021).
13. *Documentation and Test Consoles for Over 1400 Public APIs*. Dostupné také z: `https://any-api.com/`. (cit: 6.4.2021).
14. *OMDb API*. Dostupné také z: `http://www.omdbapi.com/`. (cit: 6.4.2021).
15. *Google Sheets API*. Dostupné také z: `https://developers.google.com/sheets/api/quickstart/java`. (cit: 6.4.2021).

16. *Google Drive API*. Dostupné také z: `https://developers.google.com/drive/api/v3/about-sdk`. (cit: 6.4.2021).

17. *Gmail API*. Dostupné také z: `https://developers.google.com/gmail/api/quickstart/java`. (cit: 6.4.2021).

18. *What is HTML?* Dostupné také z: `https://www.w3schools.com/html/html_intro.asp`. (cit: 4.5.2021).

19. *Developers bring their ideas to life with Docker*. Dostupné také z: `https://www.docker.com/why-docker`. (cit: 5.5.2021).