



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Analýza možnosti elektronizace podnikových proces
<b>Student:</b>	Alexey Kim
<b>Vedoucí:</b>	Ing. Radek Hronza
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Informa ní systémy a management
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

- 1) Seznamte se s oblastí procesního ízení (Business Process Management).
- 2) Analyzujte existující zp soby elektronizace podnikových proces . Zam te se p edevším na platformu pro elektronizaci proces Camunda.
- 3) Ve spolupráci s vedoucím záv re né práce definujte dva TO-BE procesy z agendy zpracovávání žádostí studijního odd lení pro doktorské studium na VUT FEL a kompletn jej v platform Camunda elektronizujte.
- 4) Pr b h elektronizace vybraného procesu popište. A to v etn ešených problém a vydefinování nezbytných podmínek pro umožn ní elektronizace vybraných proces ve zmín né platform .
- 5) Zhodno te a vy íslete veškeré náklady na po ízení nezbytného HW a SW vybavení, kompletní elektronizaci zvolených proces a zajišt ní (alespo ) jednoletého provozu (v etn nezbytné správy a uživatelské podpory).
- 6) Analyzujte a uve te p ínosy elektronizace zmín ného procesu pro chod studijního odd lení pro doktorské studium na VUT FEL.

### Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
řídící kan

V Praze dne 31. íjna 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **Analýza možnosti elektronizace podnikových procesů**

*Alexey Kim*

Vedoucí práce: Ing. Radek Hronza

9. ledna 2017



---

## Poděkování

Chtěl bych poděkovat Ing. Radku Hronzovi za vedení mé bakalářské práce, cenné rady a odborný dohled. Děkuji také své rodině za podporu během studia. V neposlední řadě děkuji všem, kdo mě podporovali a jakkoli mně pomáhali.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. ledna 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Alexey Kim. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Kim, Alexey. *Analýza možnosti elektronizace podnikových procesů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Tato práce řeší problematiku elektronizace podnikového procesu v BPM nástroji Camunda a následné vyčíslení nákladů na údržbu a uživatelskou podporu. Nejprve čtenář bude seznámen s oblastí procesního řízení, způsoby modelování procesů a nástroji pro modelování procesů. Následně v praktické části bude řešena elektronizace procesu v Camundě a analýza přínosů elektronizace procesu pro vybranou organizaci včetně vyhodnocení nákladů spojených s nasazením a údržbou Camundy.

**Klíčová slova** Podnikový proces, procesní řízení, metody elektronizace procesů, webová aplikace

---

# Abstract

This bachelor's thesis aims to solve the problem of process electrionization in Camunda BPM platfrom and further analysis of costs for technical and client support. Firstly reader will be familiarized with process management sphere, processes modeling types and tools. Proces electrionization in Camunda and analysis of benefits for chosen organization including evaluation of costs for implementation and support will be solved in implementation part.

**Keywords** Business process, process management, process electronization types, web application

---

# Obsah

Úvod	1
<b>1 Teoretická část</b>	<b>3</b>
1.1 Procesní řízení . . . . .	3
1.2 Proces a podnikový proces . . . . .	4
1.3 Elektronizace podnikových procesů . . . . .	12
<b>2 Upřesnění cílů a popis řešeného problému</b>	<b>21</b>
2.1 Popis problematiky vybrané organizace . . . . .	21
2.2 Definice a výběr dvou procesů pro elektronizaci . . . . .	22
2.3 Upřesnění hlavního cíle závěrečné práce a jeho rozpad na další podcíle . . . . .	25
<b>3 Praktická část</b>	<b>27</b>
3.1 Definice průběhu procesů F15 a F12 . . . . .	27
3.2 Zprovoznění prostředí Camunda . . . . .	31
3.3 Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT . . . . .	36
<b>4 Vyhodnocení praktické části</b>	<b>57</b>
4.1 Zhodnocení průběhu elektronizace . . . . .	57
4.2 Vyčíslení nákladů . . . . .	59
4.3 Analýza přínosů nasazení Camundy . . . . .	64
<b>Závěr</b>	<b>67</b>
<b>Literatura</b>	<b>69</b>
<b>A Rozhraní studentForm.html</b>	<b>71</b>
<b>B Rozhraní TaskAssignmentListener.java</b>	<b>73</b>

<b>C</b>	<b>Rozhraní CreatePdf.java</b>	<b>75</b>
<b>D</b>	<b>Grafické rozhraní formulářů v procesu F15</b>	<b>79</b>
<b>E</b>	<b>Grafické rozhraní formulářů v procesu F12</b>	<b>83</b>
<b>F</b>	<b>Seznam použitých zkratk</b>	<b>89</b>
<b>G</b>	<b>Obsah přiloženého CD</b>	<b>91</b>

---

## Seznam obrázků

1.1	Grafická prezentace tokových objektů . . . . .	10
1.2	Grafická prezentace spojovacích objektů . . . . .	11
1.3	Podnikový proces s použitím plaveckých drah[1] . . . . .	12
1.4	2016 Gartner Magic Quadrant for Intelligent Business Process Management Suites[2] . . . . .	14
1.5	Architektura procesního Enginu Camundy BPM[3] . . . . .	17
1.6	Camunda Tasklist . . . . .	18
1.7	Camunda Cockpit . . . . .	19
1.8	Camunda Admin . . . . .	20
2.1	Stavový diagram formuláře [4] . . . . .	23
3.1	Procesní diagram procesu F15 . . . . .	28
3.2	Procesní diagram procesu F12 . . . . .	30
3.3	Nastavení Maven projektu . . . . .	34
3.4	Rozhodovací tabulka pro výběr školitele . . . . .	39
3.5	Rozhodovací tabulka pro výběr vedoucího katedry . . . . .	41
3.6	Rozhodovací tabulka pro výběr předsedy ORO . . . . .	43
3.7	Rozhodovací tabulka pro výběr prvního oponenta . . . . .	52
3.8	Rozhodovací tabulka pro výběr druhého oponenta . . . . .	52
D.1	Výběr procesu . . . . .	79
D.2	Vyplnění formuláře . . . . .	79
D.3	Posouzení školitele . . . . .	80
D.4	Posouzení vedoucího katedry . . . . .	80
D.5	Posouzení předsedy oborové rady . . . . .	81
D.6	Vložení čísla jednacího . . . . .	81
D.7	Tisk dokumentu, zajištění podpisu a odeslání . . . . .	82
D.8	Úprava formuláře . . . . .	82
E.1	Vyplnění formuláře . . . . .	83

E.2	Posouzení školitele . . . . .	84
E.3	Posouzení předsedy oborové rady . . . . .	84
E.4	Posouzení předsedy oborové rady . . . . .	85
E.5	Doplnění dokumentů studentem . . . . .	85
E.6	Formální kontrola žádosti . . . . .	86
E.7	Posouzení oponenta A . . . . .	86
E.8	Posouzení oponenta B . . . . .	87
E.9	Seznámení s posudky . . . . .	87
E.10	Vložení čísla jednacího . . . . .	88
E.11	Tisk dokumentu, zajištění podpisu a odeslání . . . . .	88

---

# Seznam tabulek

1.1	Typy, způsob řízení a všeobecná charakteristika podnikových procesů [5]	6
3.1	Uživatelské účty	36
3.2	Testovací e-maily skupin uživatelů	37
3.3	Nastavení procesu F15	37
3.4	Atributy úlohy „Vyplnění formuláře“	38
3.5	Hodnoty atributu „studentSubject“	38
3.6	Atributy úlohy „Posouzení školitele“	40
3.7	Hodnoty atributu „approveMentor“	40
3.8	Nastavení elementu „Exclusive Gateway“	41
3.9	Atributy úlohy „Posouzení vedoucího katedry“	42
3.10	Atributy úlohy „Posouzení předsedy oborové rady“	44
3.11	Nastavení procesu F12	47
3.12	Atributy úlohy „Vyplnění formuláře“	48
3.13	Atributy úlohy „Posouzení školitele“	49
3.14	Atributy úlohy „Posouzení předsedy oborové rady“	50
3.15	Atributy úlohy „Formální kontrola žádosti“	51
3.16	Atributy úlohy „Posouzení oponenta A“	52
3.17	Atributy úlohy „Posouzení oponenta B“	53
4.1	Základní konfigurace serveru	59
4.2	Možnosti zakoupení serveru	60
4.3	Náklady na elektronizaci procesů F12 a F15	62
4.4	Skupina podpory a údržby - první 3 měsíce od spuštění procesů	62
4.5	Skupina podpory a údržby - alokace po 3. měsících od spuštění procesů	63
4.6	Roční náklady na provoz Camundy	64





---

# Úvod

## Předmluva

Procesní řízení je přístup, který řeší problematiku řízení společnosti za předpokladu, že je těsně provázán s vnitropodnikovou strategií a má aktivní podporu vrcholového managementu společnosti. Avšak procesní řízení není možné nebo je velmi neproduktivní bez podpory speciálních nástrojů, v dnešní době softwarových nástrojů. V této práci se budu věnovat problematice elektronizace podnikových procesů.

## Motivace a cíl

Hlavní důvod zkoumání této oblasti je obrovský potenciál rozvoje organizace pomocí optimalizace práce díky správně navrženým a elektronizovaným procesům. Elektronizace procesů umožňuje šetřit jeden z hlavních pracovních prostředků - čas. Cílem této práce je ukázka možnosti elektronizace procesů v nástroji Camunda.

## Struktura práce

V teoretické části své práce jsem se zaměřil na hlavní aspekty procesního řízení, způsoby modelování procesů, nástroje pro modelování procesů a především podrobněji se podívám na nástroj pro elektronizaci procesů Camunda.

V následné praktické části jsem se zaměřil na elektronizaci mnou a mým vedoucím vybraného podnikového procesu a vyčíslení nákladů na pořízení nutného HW a SW pro zajištění alespoň jednoletého provozu. Na konci provedu analýzu přínosů elektronizace vybraného procesu pro chod organizace.



---

# Teoretická část

## 1.1 Procesní řízení

V dnešní době jsou prostor a podmínky pro vedení byznysu velmi příhodné. Velká konkurence na trhu nutí firmy stále zlepšovat kvalitu zboží a služeb, čas a způsoby dodání, efektivnost práce. Pro tyto účely existuje řada metodik, nástrojů a dalších způsobů optimalizace práce, ale základní a nezbytnou metodikou je sledování výkonnosti a plánování jednotlivých úkolů neboli „Procesní řízení“. V této kapitole se podíváme na to, jakým způsobem se řídí procesy a jaké k tomu existují přístupy. V odborné literatuře je pod pojmem procesní řízení myšleno:

*„Procesní řízení (angl. Business Process Management) - představuje systémy, postupy, metody a nástroje trvalého zajištění, maximální výkonnosti a neustálého zlepšování podnikových a mezipodnikových procesů, které vycházejí z jasně definované strategie organizace a jejichž cílem je naplnit stanovené strategické cíle.“ [5]*

### 1.1.1 Způsoby řízení procesů

Pro řízení procesů existuje řada metodik, ale každý podnik se snaží přizpůsobit procesy podle vlastností projektu, podniku, týmu lidí atd. Nejvýhodnější variantou je situace, kdy každý člen týmu přesně ví, co má dělat a za co odpovídá, nebo když jsou procesy navrženy tak, že se navzájem zlepšují a doplňují. Těžko říct, jak by mohly fungovat moderní velké korporace jako Apple, Microsoft apod. bez podpory moderních technologií v oblasti řízení procesů. To není skoro vůbec možné, ale přesto nejsou technologické nástroje hlavními katalyzátory úspěšného podniku. Základ úspěchu tvoří dobře nastavený, poskládaný a stejně smýšlející tým nadšených lidí, který maximálně využívá technologické zdroje pro svoji práci. Než tým začne pracovat na procesu, manažer (zadavatel, koordinátor) procesu nejdříve musí jasně definovat cíl procesu a hlavní prostředky a správně určit toho, kdo tento proces bude vykonávat. Správně

naplánovaný proces může přinést extra výhody, například šetření lidských a finančních zdrojů z důvodu včasného dokončení procesu nebo zlepšení procesu pro použití v budoucnu. [6]

Úrovně řízení lze nejlépe popsat pomocí stupňů řízení dle modelu CMM (viz dole) (Capability Maturity Model) [7]:

1. Neexistující řízení: Procesy a jejich řízení jsou zcela chaotické.
2. Počáteční (Initial): Procesy jsou realizovány ad hoc.
3. Opakované (Repeatable): Dodržuje se určitá kázeň nezbytná pro provádění základních opakovaných procesů.
4. Definovaná (Defined): Procesy organizace jsou zdokumentovány.
5. Řízená (Managed): Procesy jsou řízeny a provádí se měření jejich výkonnosti pomocí KPI.
6. Optimalizovaná (optimized): Procesy jsou trvale zlepšovány, existuje inovační cyklus při procesech a řízení.

### 1.1.2 Přístupy k řízení podniku

Organizace může přistupovat k řízení své činnosti pomocí tří základních přístupů [8]:

1. Projektový přístup - je především zaměřen na unikátní procesy, které jsou jedinečné nebo se opakují velmi málo (např. jednou za 2 roky). Postup řešení těchto procesů se obvykle nalézá v průběhu realizace.
2. Procesní přístup - na rozdíl od projektového přístupu je zaměřen na procesy, které se opakují docela často. Hlavní myšlenka procesního přístupu je řízení všech činností a zdrojů jako jednoho procesu. Jsou to vstupy a výstupy pro proces, monitorování a zlepšování.
3. Funkční přístup - práce je rozdělena podle specializace a úkoly se dělí na malé lehčí úkoly. Takové úkoly pak může vyřešit i méně kvalifikovaný tým.

## 1.2 Proces a podnikový proces

Z předchozí kapitoly vyplývá, že procesní řízení řeší problematiku zlepšení a maximální výkonnosti podnikových procesů. Ale co to je podnikový proces?

V dnešní době se každý člověk setkává s procesy. Například proces koupení mléka v obchodu nebo proces uzavření smlouvy v megakorporaci. Správně ošetřený proces je zárukou úspěchu celého projektu. Slovo proces (z latinského processus, což je tvar slovesa procedere – „postupovat, vyvíjet se“) můžeme

potkat i v oblastech fyziky a chemie, ale v rámci této bakalářské práce nás bude zajímat oblast řízení informačních systémů. V této oblasti už definujeme pojem „podnikový proces“. V odborné literatuře můžeme potkat spousta definic:

1. *„Práce lidí v organizacích se vždy odehrává prostřednictvím jejich činností, které se řetězí do procesů. Proces je sled činností, které na sebe vzájemně navazují, vytvářejí tok práce postupující od jednoho člověka k druhému a tvoří hodnotu. Každý proces má nějaké vstupy, nějaké výstupy a spotřebovává nějaké zdroje. Každý proces je spuštěn nějakou událostí. Procesy tedy rozhodně musí být nějak nastavené a musí být nějak řízené, aby nezavládl úplný chaos.“* [7]
2. *„Proces je organizovaná skupina vzájemně souvisejících činností a/nebo procesů, které procházejí jedním nebo více organizačními útvary či jednou (podnikový proces) nebo více spolupracujícími organizacemi (mezi-podnikový proces), které spotřebovávají materiální, lidské, finanční a informační vstupy a jejichž výstupem je produkt, který má hodnotu pro externího nebo interního zákazníka.“* [5]
3. *„Podnikovým procesem zpravidla rozumíme objektivně přirozenou posloupnost činností, konaných s úmyslem dosažení daného cíle v objektivně daných podmínkách.“* [6]
4. *„Podnikový proces je „souhrn činností, který vyžaduje jeden nebo více druhů vstupů a vytváří výstup, který má pro zákazníka hodnotu.“* [9]

Pokud mluvíme o podnikových procesech, tak musíme vždy pamatovat na to, že jednu z hlavních rolí hraje čas. Definovali jsme proces jako posloupnost činností neboli časovou posloupnost. Každá činnost je provedena v jistém čase, což nám vždy umožňuje porovnávat jednotlivé činnosti a procesy.

Neoddělitelné části podnikového procesu jsou

1. cíl
2. úmysl
3. objektivní přirozenost postupu
4. objektivně dané podmínky

Jasně definovaný cíl je hlavním charakteristickým rysem podnikového procesu, na rozdíl od popisů životních cyklů objektů, kde požadavek na cíl nemá smysl. Samotný proces je vždy závislý na okolních faktorech, které působí na činnosti tohoto procesu přímo nebo nepřímo. [6]

### 1.2.1 Typy podnikových procesů

V praktickém světě jsou procesy vždy rozděleny na jednotlivé kategorie, přičemž jeden proces může být začleněn do více kategorií. Uspořádání samotných procesů je jedním z prvních kroků ke zlepšení toku práce. Kategorie, do které bude spadat nějaký proces, je zvolená podle potřeb, závažnosti, náročnosti a dalších atributů. Základní rozdělení procesů se rozděluje do dvou odvětví: první jsou vnitropodnikové procesy a druhé jsou procesy externí. Do těchto dvou kategorií můžeme začlenit úplně každý proces firmy. Další zásadní odvětví jsou: procesy zaměřené na prodej produktu (proces nákupu) nebo jeho výrobu, procesy krátkodobé (výroba a prodej) a dlouhodobé prosperity (výroba a výzkum), technologické nebo informační atd. Obecné rozdělení procesů podle obtížnosti (složitě, středně složitě, jednoduché apod.) z hlediska procesního managementu nemá smysl, protože běh procesu je vždy závislý na okolních podmínkách; náročnost se může měnit několikrát.

Jedním z nejčastějších způsobů rozdělení je rozdělení na hlavní, řídicí a podpůrné procesy. Výhodou tohoto rozdělení je přehlednost, srozumitelnost a poskytování nejdůležitějších informací o procesech. Díky tomu můžeme snadněji stanovit prioritu vykonání procesů. Základní charakteristiky hlavních, řídicích a podpůrných procesů jsou uvedeny v následující tabulce. [5]

Typ procesu	Způsob, jakým má být řízen	Přidává hodnotu?	Probíhá napříč organizací?	Má externí zákazníky?	Generuje tržby (zisk)?
Hlavní	výkonově	ANO	ANO	ANO	ANO
Řídicí	nákladově	NE	ANO	NE	NE
Podpůrný	výkonově, možnost outsourcingu	ANO	NE	NE	NE

Tabulka 1.1: Typy, způsob řízení a všeobecná charakteristika podnikových procesů [5]

### 1.2.2 Vykonání podnikových procesů

Organizace by měla v praxi vykonávat procesy dle definovaného průběhu. Pouze tím bude zajištěno, že vznikají výstupní produkty v požadované kvalitě.[10]

Vykonání procesů lze obvykle zajistit dvěma způsoby.

1. **Papírová podoba** - dle takového typu vykonání procesů podniky pomocí speciálního systému jenom zajišťují určitý průběh podnikových

procesů. Těmito systémy můžou být například směrnice, metodické pokyny apod.

2. **Elektronická podoba** - vykonávání procesů je zajištěno pomocí výpočetní techniky. Tím pádem dojde k elektronizaci podnikových procesů. Elektronizované procesy podstatně ulehčují práci pracovníkům, urychlují oběh informací.

Výhoda elektronizace procesů je obrovská a především se spočívá v:

1. **Zrychlení práce** - jednotlivé úkoly se budou vykonávat mnohem rychleji díky přehledným formulářům, diagramům a obecné rychlosti přenosu informace pomocí počítačů.
2. **Bezpečnost** - míra bezpečnosti je mnohem vyšší než u papírové podoby, protože každý uživatel v systému BPM má určitá práva a hodně se snižuje pravděpodobnost ztráty informace.
3. **Evidence** - průběhy procesů se ukládají do databáze, což umožňuje v budoucnu zpětně dohledávat tyto procesy, vyrábět statistiky použití, definovat problémová místa pro budoucí úpravu procesů.

Předtím než podnikový proces začne svůj životní cyklus v rámci nějaké BPM platformy, je potřeba ho na začátku „zakreslit“. V nakresleném procesu musí být zachycená logika průběhu celého procesu. V následující podkapitole ukážeme, jaké způsoby existují pro kreslení neboli modelování podnikových procesů.

### 1.2.3 Způsoby modelování podnikových procesů

Hledání nových inovačních cest ke zlepšení podnikových procesů se považuje za velkou konkurenční výhodu pro společnost. Historicky můžeme všechny snahy o vylepšení procesů rozdělit na tři časové okamžiky, tak zvané „vlny“. První vlna vnikla v první polovině 20. století a procesy byly implicitní a neautomatizované. Druhá vlna vznikla na konci 20. století, kdy procesy prošly reengineeringem a byli integrované do ERP systémů. Tyto systémy byly velmi omezené z pohledu schopnosti poskytnout kontrolu nad procesy. Třetí vlna BPM nám přinesla rozšíření pracovních toků a seskupila různá procesní řešení do jednoho celku. Hlavní význam 3. vlny je možnost a schopnost vytvořit jediný koncept pro vyjádření procesů a poskytnout různé úhly pohledu na ten samý proces. To znamená, že různé skupiny uživatelů mohou vidět proces různě a pracovat s ním, jak jim to vyhovuje [5]. Změny se dotkly i způsobů procesního modelování. Dnes jsme schopni vytvořit interaktivní grafický procesní model, který je mnohem přehlednější než jiné dodnes známé prezentační modely. Existuje řada způsobů, jak můžeme namodelovat podnikový proces.

Nejpoužívanější metodiky zakreslení podnikových procesů jsou: UML, procesní mapa, Petriho síť a diagram datových toků. Podle specifických potřeb každého podniku nebo v jedné konkrétní situaci můžeme vybírat mezi těmito typy modelování, protože každá metodika má svoje určité výhody, o kterých se budeme bavit v dalších podkapitolách.

Za posledních 10 let byla vytvořena řada zajímavých a mocných nástrojů pro modelování procesů. Základní logikou těchto nástrojů je popis procesů pomocí grafických objektů, logicky propojených mezi sebou. Všechny tyto nástroje můžeme začlenit do tří kategorií:

1. Nástroje pro simulaci - důkladná analýza dat. Hlavní výhodou je sledování objektu přes celý proces při různých parametrech.
2. CASE (Computer Aided Software Engineering) nástroje - poskytují platformu pro znázornění a popis vzájemných vztahů procesů. Generování zdrojového kódu z namodelovaného procesu také většinou umožňuje generování dokumentace z modelu.
3. Nástroje pro přehled procesních toků - jednoduché nástroje, které slouží jen k transformování slovní definice procesu do grafické podoby.

Pro grafické znázornění procesních modelů v dnešní době existuje řada vhodných modelovacích jazyků. Příklady modelovacích jazyků:

1. Unified Modeling Language (UML)
2. Business Process Model and Notation (BPMN)
3. Event-driven process chain (EPC)
4. Petriho síť
5. Finite State Machine (FSM)
6. Subject Oriented Business Process Management (S-BPM)
7. Yet Another Workflow Language (YAWL)

I když se během práce na první pohled zdá, že tyto jazyky jsou velmi podobné, každý jazyk se do značné míry odlišuje a výstupní modely představují různé charakteristiky. V rámci této bakalářské práce nás bude zajímat jenom notace BPMN z důvodu, že je to nejvíc používaná světová notace pro zakreslení procesu, a vzhledem k tomu, že zakreslení procesů pro platformu Camunda probíhá právě v notaci BPMN, je potřeba porozumět základům potřebným pro následující praktickou část závěrečné práce.



### 1.2.3.1 Business Process Management Notation 2.0

1. „*BPMN - Business Process Management Notation 2.0 (BPMN 2.0) - je soubor principů a pravidel, který slouží pro grafické znázorňování podnikových procesů pomocí procesních diagramů. Jinými slovy lze říci, že jde o standard pro modelování podnikových procesů.*“ [11]
2. „*BPMN 2.0 is a formal industry standard that enables business process management systems to design and execute well-defined processes. BPMN 2.0 is applicable across a range of design and development tasks spanning from pure business descriptions to executable artifacts and the highest level of detail. As such, BPMN 2.0 is also a good format for the interchange of process resources between tools and repositories.*“ [12]

Účelem vytvoření notace je, aby každý firemní uživatel (případně zákazník) pochopil vlastnosti a podstatu nějakého procesu. Hlavní operační prvky BPMN jsou grafické objekty. Díky nim tato notace poskytuje srozumitelný přehled a vazby úkolů procesu. Rozdíl mezi BPMN 2.0 a UML je velice malý. BPMN 2.0 se věnuje přímo modelování podnikových procesů. UML má 14 typů diagramů, každý má speciální diagram aktivit, který slouží pro popis podnikových procesů. V této bakalářské práci nás bude více zajímat BPMN 2.0. Čtyři základní kategorie BPMN procesů jsou:

1. Tokové objekty (Flow objects)
2. Spojovací objekty (Connecting Objects)
3. Plavecké dráhy (Swimlanes)
4. Artefakty (Artifacts)

#### Tokové objekty

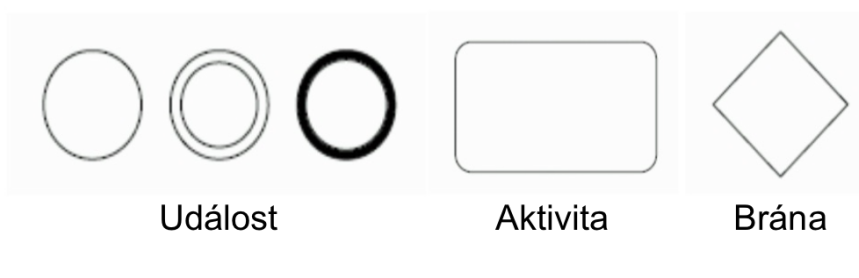
BPMN definuje BPD (Business Process Diagram), což je grafická implementace BPMN. Řídící množinu klíčových objektů procesního diagramu tvoří Tokové objekty. Jsou to Událost (Event), Aktivita (Activity) a Brána (Gateway).

**Události** mají přímý vliv na tok procesu, zpravidla jsou vyvolány nějakou akcí (Trigger). Události mohou být počáteční, střední a koncové. Grafickou prezentací události je obyčejný kruh. Podle vztahu k procesu definujeme tři typy událostí - počáteční (Start), střední (Intermediate) a koncovou (End). Každý proces musí začínat počáteční událostí. [13][14]

**Aktivita** je základním kamenem BPMN 2.0 notace. Představuje obecný grafický prvek (většinou ve tvaru obdélníku) a představuje nějakou práci. Rozlišujeme dva typy aktivit: Úloha (Task) a Subproces (Sub-Process). Pokud

chceme zobrazit nějakou aktivitu a víme, že se skládá jenom z jedné akce, neboli je atomická, tak použijeme Task. V opačném případě, když akce je složitá a můžeme ji rozdělit na několik částí, použijeme Sub-Process. [13][14]

Poslední objekt je **brána**, používaná pro rozdělení a sloučení procesního toku. Grafickým znázorněním je otočený čtverec. Během procesu může jedna aktivita způsobit různé výsledky. Brána je přesně ten rozhodovací bod, kde zvažujeme podmínky a děláme rozhodnutí.



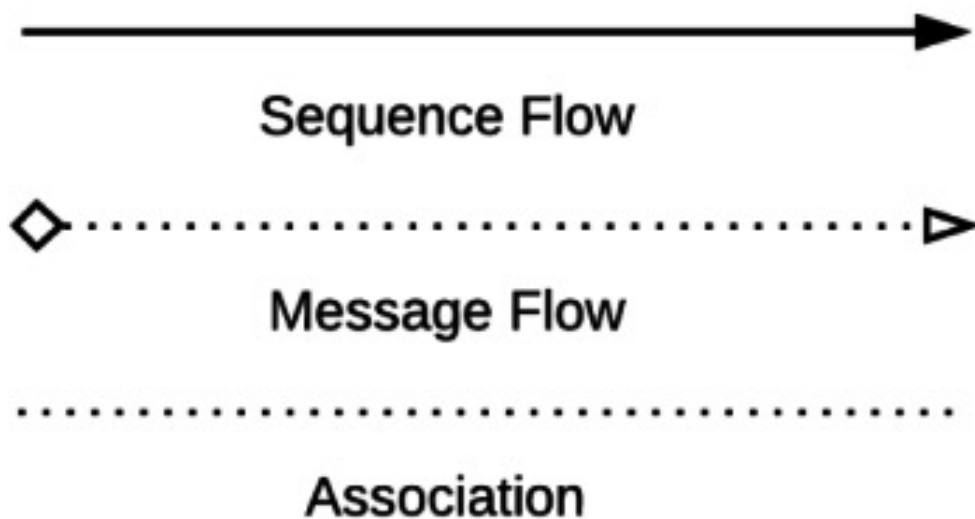
Obrázek 1.1: Grafická prezentace tokových objektů

### Spojovací objekty

Aby nějaká sada objektů mohla reprezentovat nějaký business proces, musí být všechny tyto objekty mezi sebou logicky propojeny. Notace BPMN má tři typy spojovacích objektů.

1. Sekvenční tok (Sequence Flow) zaplavované
2. Tok zpráv (Message Flow)
3. Asociace (Association)

Sekvenční tok je hlavní spojovací objekt a je prezentovaný plnou čarou s vyplněnou šipkou. Úkolem tohoto objektu je určovat, v jakém pořadí budou aktivity procesu vykonávány. Sekvenční tok lze používat jenom v rámci jedné plavečkové dráhy (viz dále). Tok zpráv je prezentován pomocí přerušované čáry s prázdnou šipkou na konci. Určuje tok zpráv mezi různými účastníky procesu. Posledním objektem je Asociace. Používá se pro zobrazení výstupů a vstupů různých aktivit. [13][14]



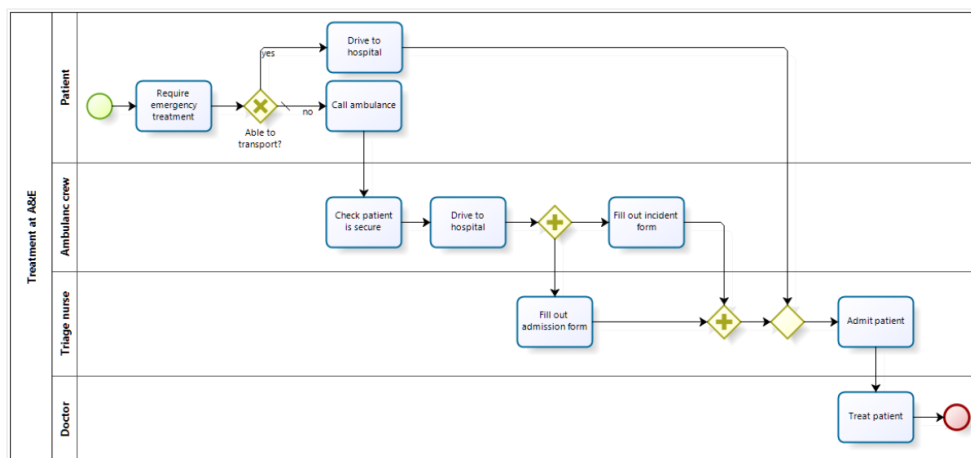
Obrázek 1.2: Grafická prezentace spojovacích objektů

### Plavecké dráhy

Na první pohled se může zdát, že pojem Plavecká dráha nemůže mít nic společného s modelováním podnikových procesů. Ale když pochopíme účel a logiku plavecké dráhy v obyčejném bazénu, uvidíme, že tomu tak není. V bazénu je účelem plavecké dráhy to, aby každý plavec měl svůj pevně daný prostor pro plavání a v žádném případě nemohl změnit dráhu. Přesně takový koncept existuje v BPM. Dále už budeme mluvit o plavecké dráze jako o objektu BPMN. Plavecká dráha v notaci BPMN 2.0 je ve formě obdélníku, který reprezentuje nějakého účastníka podnikového procesu. Účastníkem může být nějaká osoba (např. zákazník, dodavatel), ale také skupina nebo objekt (např. tým vývojářů, analytické oddělení). Dráha obsahuje tokové objekty příslušného účastníka. Obvykle je plavecká dráha zobrazená horizontálně, ale v některých případech to může být vertikálně. Sémanticky neexistuje žádný rozdíl, jsou to jenom různé způsoby prezentace.

Může nastat situace, kdy potřebujeme rozdělit nějakou dráhu na více částí. Tyto jednotlivé části se jmenují Cesta (angl. Lane). Například máme dráhu Oddělení vývojářů, kde můžou pracovat Android a iOS vývojáři. [13][14]

## 1. TEORETICKÁ ČÁST



Obrázek 1.3: Podnikový proces s použitím plaveckých drah[1]

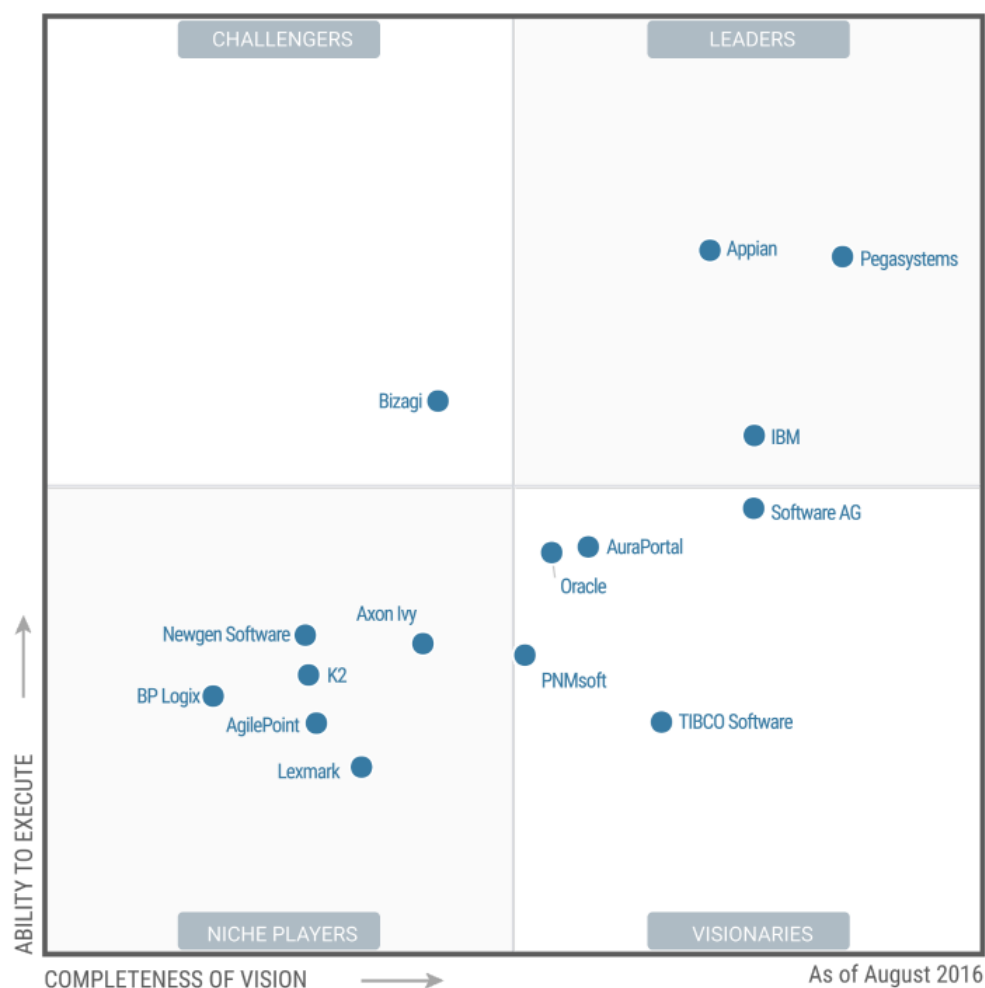
### Artefakty

Někdy potřebujeme podrobněji popsat určitou část procesu, zdůraznit zdroje, definovat skupiny apod. Přesně pro tyto účely v notaci BPMN máme speciální objekty - Artefakty. Artefakty umožňují reprezentovat objekt vně aktuálně běžícího procesu. Artefakty slouží jenom pro informatické účely a dokumentaci objektu. Data, která popisují proces, poznámky, objekty, anotace a skupiny - jsou Artefakty. [14]

### 1.3 Elektronizace podnikových procesů

Všechny způsoby modelování podnikových procesů, které byly zmíněny výše, se objevily na začátku 20. století. Prvními z nich byly Ganttovy diagramy, diagramy datových toků a funkční diagramy. První nástroje pro modelování procesů měly za úkol primárně popsat cenu a čas práce, avšak moderní nástroje se zaměřují na multifunkční aktivity. Významný přínos do vývoje modelovacích nástrojů vnesly BPR (Business Process Reengineering) a BPM (Business Process Management), většinou podle těchto forem probíhá modelování procesů v moderních nástrojích. Dnes na trhu existuje velká konkurence mezi firmami, které vyrábí software. Desítky firem bojují o zákazníky v oblasti BPM a samozřejmě někdo má větší úspěch než ostatní. V celkovém přehledu společností, které vyrábějí různý software, a konkrétně soft pro řízení podnikových procesů, nám pomůže Gartnerův magický kvadrant (Gartner Magic Quadrant). Gartnerův kvadrant je průzkumná metodika a zároveň grafická prezentace vyhodnocení pozice a progresu podniku v nějaké technologické sféře vůči ostatním. Magický kvadrant je dvoudimenzionální matice, která zobrazuje silné stránky a rozdíly mezi podniky. Matice je rozdělena na čtyři sekce [15].

1. **Specializovaní hráči (Niche players)** - mají velmi úzké zaměření a jsou úspěšní v nějakém konkrétním segmentu trhu. Mají omezené schopnosti inovace a zvýšení výkonu. Zpravidla jsou to nové podniky nebo to také mohou být podniky, které se zaměřují jenom na specifickou část zákazníků.
2. **Lídři (Leadres)** - vrcholová skupina v poskytování služeb a zboží nejlepší kvality. Důležitá věc je, že lídři jsou schopni udržet vedoucí pozice v průběhu vývoje trhu. Jejich projekty nejvíce ovlivňují celkové směřování trhu.
3. **Vizionáři (Visionaries)** - mají představu, jak se bude chovat trh v budoucnu, ale nejsou si jistí, jestli budou schopni dodávat svoje řešení na trh v plném rozsahu (nedostatek financování apod.)
4. **Vyzývatelé (Challengers)** - mají dobré postavení na trhu díky kvalitní realizaci svých projektů. Ale nemají dlouhodobou vizi pro udržení svých pozic. Většinou mají dobré finanční prostředky a lidské zdroje, ale nemají představu, jak dál vylepšovat produkt a kam mají směřovat. Vyzývatelé mají největší šanci se přesunout do sekce Lídři, pokud správně definují svou vizi.



Obrázek 1.4: 2016 Gartner Magic Quadrant for Intelligent Business Process Management Suites[2]

V následujících podkapitolách se podrobněji podíváme na nástroje, které umožňují elektronizaci a údržbu podnikových procesů a především se zaměříme na nástroj Camunda BPM.

### 1.3.1 Pega 7

Americká softwarová společnost Pegasystems se specializuje na produkty v oblasti CRM (Customer Relationship Management) a BPM. Společnost se hodně věnuje Case Managementu, který se aktivně integruje do BPM produktů Pegasystems. V roce 2013 společnost prezentovala produkt pro modelování procesů Pega 7 BPM. Dnes je to hlavní produkt společnosti zaměřený čistě na

modelování a simulaci podnikových procesů. Web-aplikace umožňuje jak business analytikům, tak i technicky vzdělaným uživatelům bez potíží pracovat v aplikaci díky velmi dobře rozpracovanému User Experience designu. Webové prostředí pro práci se nazývá Design Studio a má všechny nutné elementy pro práci. Logika práce s procesy v Design Studiu je založena na notaci BPMN 2.0 a poskytuje uživateli klasické plavecké dráhy, objekty a navíc podporuje funkci drag-and-drop pro rychlejší modelování procesních toků. Veškerá práce v Pega 7 se točí kolem rozhodovacích řešení, která ovlivňují proces pomocí správy událostí (Event manager). Velkou výhodou oproti jiným platformám je integrace takzvané „Firemní sociální sítě“. Tato síť je představena jako okamžitá výměna zpráv (klasický chat), živé sledování modelování a okamžité komentování modelu, RSS zprávy pro účastníky procesů a řada dalších.

#### 1.3.2 IBM Business Process Manager

Společnost IBM několik let vyvíjela svůj nástroj pro modelování podnikových procesů. Jako monopol ve sféře poskytování softwarových produktů, IBM provedla akvizice firem, které také působí v tomto segmentu trhu. Finální produkt představuje řešení několika různých firem a jádro systému tvoří technologie od společnosti Lombardi. Hlavní úložiště procesních aktivit se nazývá procesní centrum (Process Center). Na rozdíl od Pega 7, IBM BPM platforma pracuje se dvěma prostředími pro návrh procesů, Process Designer a Integration Designer. Process Designer je prostředí pro modelování složitých dlouhotrvajících procesů v notaci BPMN, pokud jde o prostředí Integration Designer, mluvíme o procesech v notaci BPEL (Business Process Execution Language, jazyk ve formátu XML pro popis procesů) na aplikační úrovni většinou pro back-end vývojáře.

#### 1.3.3 Camunda BPM

Společnost Camunda byla založena Jakubem Freundem a Bernardem Ruckem v roce 2008 a dnes v ní pracuje kolem 40 lidí. Společnost není velká a v současné době nekonkuruje lídrům IBM, Pegasystems a Appian, ale můžeme ji zařadit k vrcholovým vizionářům v této oblasti. Od začátku se společnost věnovala problematice řízení, efektivnímu návrhu a realizaci podnikových procesů. Společnost Camunda se také zúčastnila definování standardu BPMN 2.0. Mezi významné zákazníky Camundy patří společnosti Allianz, T-Mobile, AT&T, Sony DADC a řada dalších. Název produktu pro modelování procesů je Camunda BPM (Business Process Management). Je to open-source platforma, která umožňuje modelování nejen v BPMN 2.0, ale také CMMN 1.1 a DMN 1.1. Jádro platformy tvoří Model Execution Engine. Práce platformy záleží na zvolené notaci - BPMN (práce s procesy), CMMN (práce s událostmi), DMN (práce s rozhodnutími). V následujících podkapitolách je krátce popsán

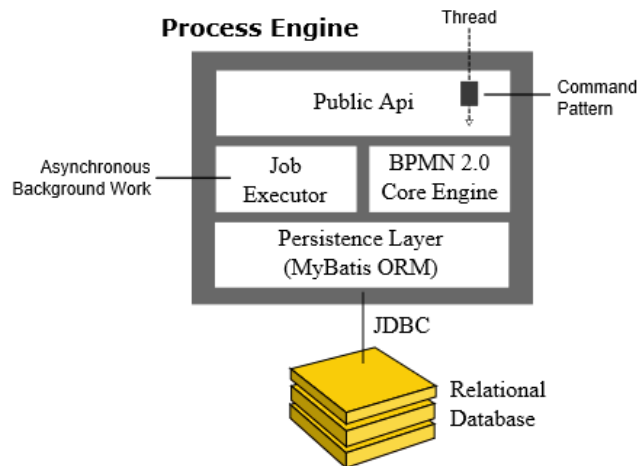
princip práce Camundy, více informací a technických specifikací lze nalézt na webu společnosti <https://docs.camunda.org/manual/7.4/>

### 1.3.3.1 Architektura

Camunda BPM je platforma založená na Javě. Všechny hlavní komponenty jsou napsány v Javě a společnost se každým rokem snaží představit nové vylepšení a rozšíření pro platformu JVM (Java Virtual Machine). Ale také existuje podpora pro ostatní vývojáře, kteří nepracují s Javou. Speciálně pro ně Camunda BPM představuje REST API, které umožňuje modelování procesů pomocí připojení na vzdálený procesní engine. Camunda BPM může být použita jako samostatný procesní server a zároveň jako integrovaná část nějaké Java aplikace. [3]

1. **Public Api** - umožňuje komunikaci Java aplikací a procesního engine. Různé odpovědnosti procesního engine jako Process Repository, Runtime Process Interaction, Task management jsou rozděleny na samostatné servery. Thready, které vstupují do procesu, ihned dostávají svůj typ (například thread transakce).
2. **BPMN 2.0 core engine** - je jádro platformy Camunda BPM, zodpovídá za samotný proces modelování a převod BPMN 2.0 XML modelu do Java objektů.
3. **Job executor** - hlavní funkcí je správa asynchronních procesů na pozadí (např. časovač) nebo jakékoliv asynchronní pohyby v modelovaných procesech.
4. **The persistence layer** - perzistentní úroveň je zodpovědná za trvalé procesy v relační databázi.





Obrázek 1.5: Architektura procesního Enginu Camundy BPM[3]

### 1.3.3.2 Scenérie (Případy) spuštění Camundy

Camunda je velmi flexibilní platforma, která podporuje různé scénáře spuštění. V této kapitole jsou popsány způsoby spuštění, které se nejvíce vyskytují v praxi. [3]

1. **Embedded process engine (Integrovaný procesní engine)** - v tomto případě procesní engine funguje jako knihovna v Java aplikaci. Zastavení a spuštění procesního engine je řízené přímo z aplikace. Můžeme spustit několik procesních engineů v případě práce se sdílenou databází.
2. **Shared, Container-Managed Process Engine** - procesní engine je spustitelný pouze přes speciální aplikační server. Aplikační server může být sdílený, a tím pádem spouštět několik engineů na různých aplikacích.
3. **Standalone (Remote) Process Engine Server** - procesní engine je prezentován jako online služba. Různé aplikace mohou interagovat s engineem přes vzdálený komunikační kanál. Nejjednodušší způsob práce s procesním engineem je komunikace pomocí REST API.

### 1.3.3.3 Camunda Modeler

Camunda modeler je desktopová aplikace pro modelování a úpravu procesních diagramů v notaci BPMN a rozhodovacích tabulek DMN (Decision Model and Notation). Má velmi nativní uživatelské rozhraní, což umožňuje vývojářům a business analytikům pracovat ve stejném nástroji na stejném modelu. Kromě

grafického modelování Camunda Modeler umožňuje editaci nastavení potřebných pro technické spuštění procesů. Díky tomu, že Camunda Modeler přímo pracuje s DMN a BPMN xml soubory, developéři mohou kombinovat práci v modeleru spolu s preferovaným IDE (například Eclipse, Netbeans, IntelliJ). [3]

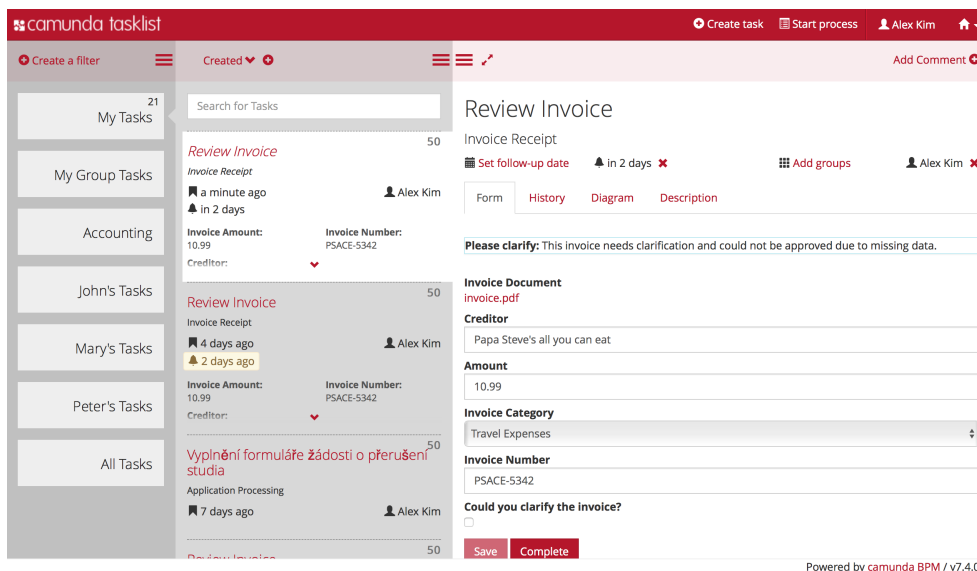
### 1.3.3.4 Uživatelské rozhraní Camunda BPM

Po spuštění procesu, veškerá práce s aktivitami, nastavení práv, správa uživatelů apod. se řeší pomocí lokálního webového rozhraní. Jsou to tři základní sekce:

1. Tasklist
2. Cockpit
3. Admin

#### Tasklist

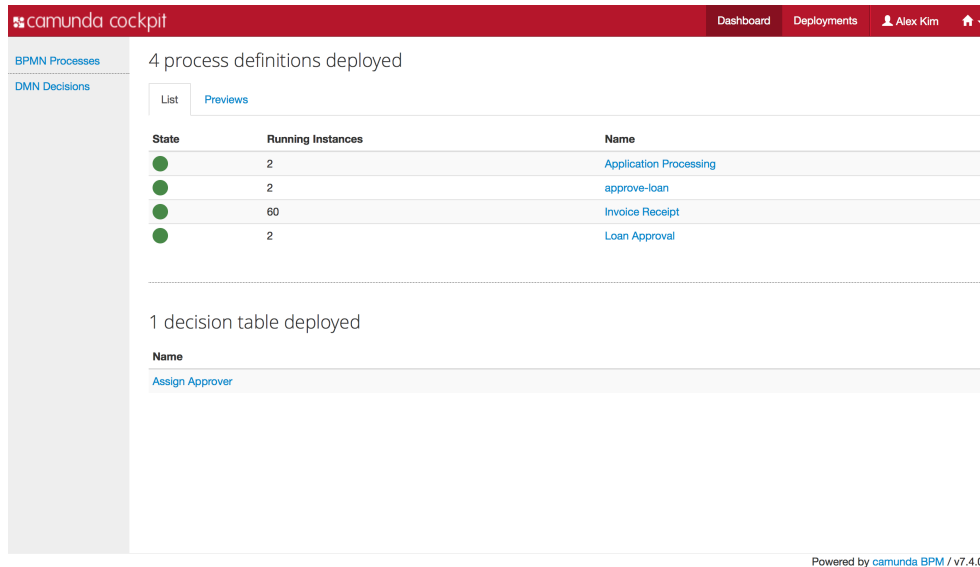
Koneční uživatelé využívají Camunda Tasklist pro organizaci a práci na úkolech, které mají přiřazené v rámci určitého procesu. V průběhu BPMN procesu bude procesní engine generovat úkoly na základě předem definovaného BPMN modelu. Po kliknutí na úkol ze seznamu úkolů se uživateli zobrazí definovaný formulář tohoto úkolu. Po ukončení formuláře uživatel musí potvrdit vyplnění úkolu (tlačítko Complete), pak procesní engine bude pokračovat v běhu procesu. [3]



Obrázek 1.6: Camunda Tasklist

## Cockpit

Sekce Cockpit je především určená pro administrátory, kteří sledují procesy. Administrátor nemusí být jenom technicky vzdělaná osoba, ale může to být i běžný pracovník business departmentu. Camunda Cockpit má velmi flexibilní rozhraní a je možné přizpůsobit dle potřeb určitých skupin uživatelů. Cockpit také poskytuje analýzu využití procesů a jejich instancí. [3]



The screenshot displays the Camunda Cockpit interface. At the top, there is a navigation bar with 'camunda cockpit', 'Dashboard', 'Deployments', and a user profile 'Alex Kim'. The main content area is divided into two sections: 'BPMN Processes' and 'DMN Decisions'. Under 'BPMN Processes', it shows '4 process definitions deployed'. Below this, there are tabs for 'List' and 'Previews'. A table lists the process definitions:

State	Running Instances	Name
●	2	Application Processing
●	2	approve-loan
●	60	Invoice Receipt
●	2	Loan Approval

Below the table, it shows '1 decision table deployed' with a table listing the decision table:

Name
Assign Approver

At the bottom right, it says 'Powered by camunda BPM / v7.4.0'.

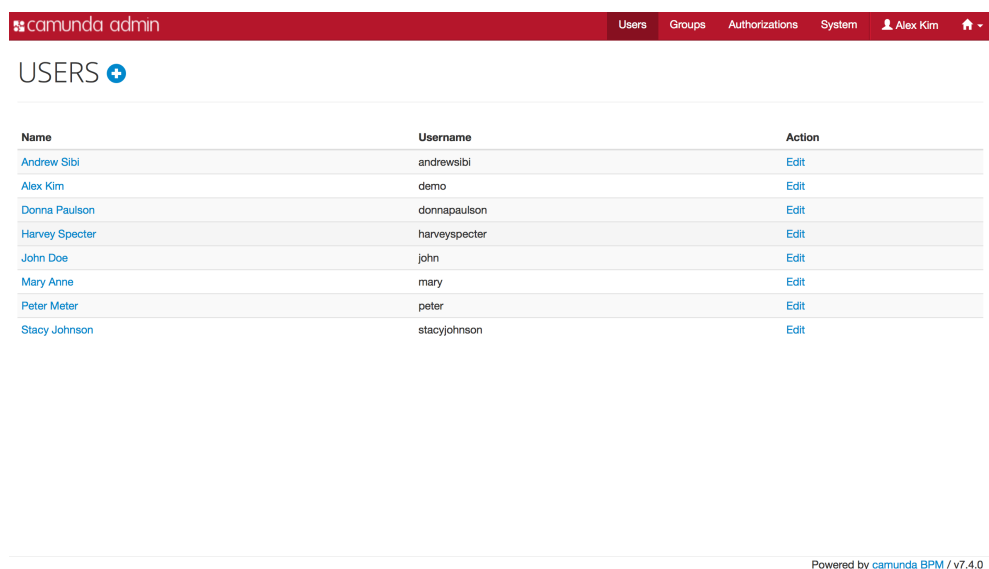
Obrázek 1.7: Camunda Cockpit

## 1. TEORETICKÁ ČÁST

---

### Admin

Stejně jako Cockpit je primárně určená pro administrátory systému. Administrátoři přidělují přístupová práva uživatelům. Všechna práva, která budou nastavená v sekci Camunda Admin, budou aplikovaná na všechny integrační úrovně (REST API, Camunda web aplikace). [3]



The screenshot shows the Camunda Admin interface. At the top, there is a red navigation bar with the Camunda logo and the text 'camunda admin'. To the right of the logo are navigation links: 'Users', 'Groups', 'Authorizations', 'System', and a user profile for 'Alex Kim'. Below the navigation bar, the main content area is titled 'USERS' with a plus icon. Underneath, there is a table with three columns: 'Name', 'Username', and 'Action'. The table lists several users, each with a corresponding 'Edit' link in the 'Action' column.

Name	Username	Action
<a href="#">Andrew Sibi</a>	andrewsibi	<a href="#">Edit</a>
<a href="#">Alex Kim</a>	demo	<a href="#">Edit</a>
<a href="#">Donna Paulson</a>	donnapaulson	<a href="#">Edit</a>
<a href="#">Harvey Specter</a>	harveyspecter	<a href="#">Edit</a>
<a href="#">John Doe</a>	john	<a href="#">Edit</a>
<a href="#">Mary Anne</a>	mary	<a href="#">Edit</a>
<a href="#">Peter Meter</a>	peter	<a href="#">Edit</a>
<a href="#">Stacy Johnson</a>	stacyjohnson	<a href="#">Edit</a>

Powered by [camunda BPM](#) / v7.4.0

Obrázek 1.8: Camunda Admin

# Upřesnění cílů a popis řešeného problému

## 2.1 Popis problematiky vybrané organizace

Po konzultaci s vedoucím závěrečné práce jsem rozhodl vybrat Fakultu elektrotechnickou Českého vysokého učení technického v Praze (ČVUT) jako organizaci, pro kterou budu provádět elektronizaci vybraných procesů, konkrétně studijní oddělení doktorského studia. ČVUT je jedna z nejstarších vysokých škol v Evropě. Dnes má vysoká škola osm fakult (stavební, strojní, elektrotechnická, jaderná a fyzikálně inženýrská, biomedicínského inženýrství, informačních technologií, dopravní a architektury) a studuje na ní více než 21.000 studentů. ČVUT je rozsáhlá organizace, která se vždy snaží vylepšit procesy pro zvýšení efektivity práce všech fakult. Důkazem toho je velké množství různých informačních systémů jako KOS, moodle, systém závěrečných prací a mnoho dalších. Navíc všechny fakulty samostatně vyvíjí vnitřní informační systémy a optimalizují procesy jak pro interní, tak i pro externí účely. Nás bude konkrétně zajímat optimalizace a rozvoj podnikových procesů na fakultě elektrotechnické (FEL).

Fakulta elektrotechnická vznikla v roce 1950. Dnes na fakultě studuje více než 6.000 studentů a pracuje 730 zaměstnanců. Roční rozpočet přesahuje 500 milionů Kč. Fakulta v současnosti uskutečňuje bakalářské, magisterské a doktorské programy. FEL funguje jako samostatná organizace v rámci ČVUT a pro řízení procesů a práce využívá metodiky procesního řízení. Od roku 2014 interní organizace CZM (Centrum znalostního managementu), která působí v rámci fakulty, provozuje speciální procesní portál, kde řeší vylepšení a návrh podnikových procesů. Po analýze současného stavu elektronizovaných procesů na fakultě bylo zjištěno, že současná papírová podoba procesů doktorského studia ne zcela vyhovuje jeho účastníkům. Na elektronizaci procesů bakalářských a magisterských programů se pracuje už poměrně dlouho, ale pro doktorské studium zatím moc uděláno nebylo. Na základě uživatelských požadavků cen-

trem znalostního managementu byl vytvořen dokument obecných požadavků na úpravu stávajícího informačního systému pro vybrané agendy doktorského studia. Následující seznam požadavků považuji za klíčový, kompletní seznam lze najít v dokumentu „Obecné požadavky na funkcionality KOSu“ [16]. Klíčové požadavky na IS jsou:

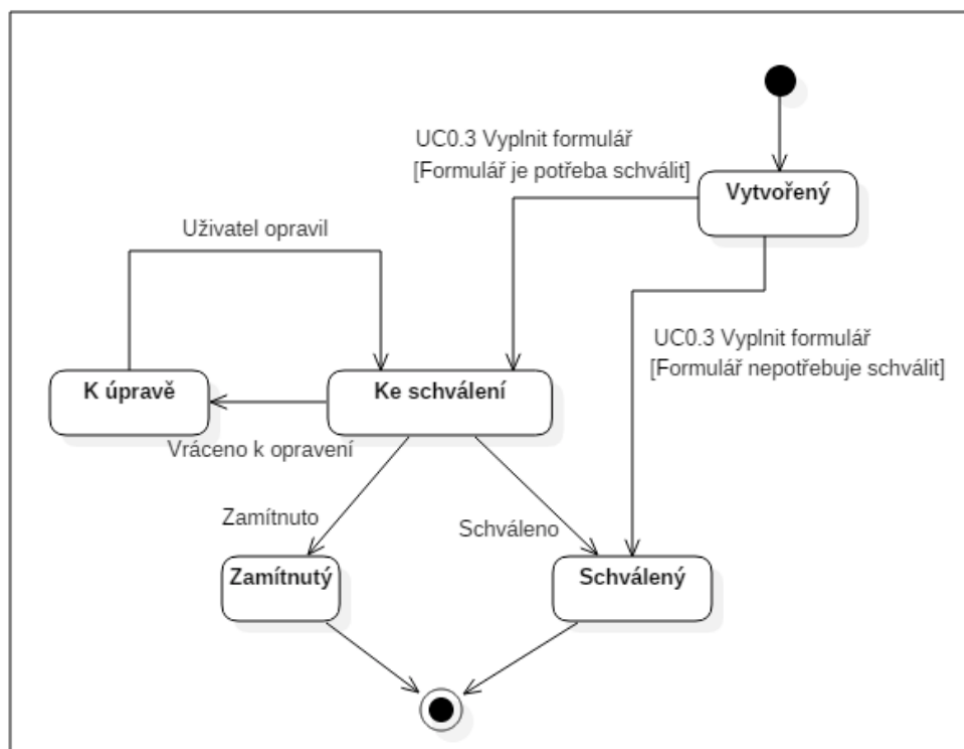
1. Snaha co nejvíce agendy odbavit on-line.
2. Uchování veškeré historie o průběhu studia doktoranda v elektronické podobě.
3. Schvalovací procedury vybraných agend v IS jsou řešeny pouze elektronicky.

Vyřešení těchto požadavků má dopad na více než 400 studentů doktorského studia a dalších zaměstnanců FELu. Navržené procesy lze upravit podle součinnosti všech programů doktorského studia ČVUT. Z toho důvodu teoretický dopad v budoucnu může být na všechny studenty doktorského studia ČVUT (1952 studentů podle matriky studentů ke dni 20.1.2016 [17]). V následující podkapitole bude popsáno, jaké procesy a proč budeme elektronizovat v rámci praktické části.

### 2.2 Definice a výběr dvou procesů pro elektronizaci

V roce 2015 Centrum znalostního managementu FEL ČVUT v rámci projektu „Elektronizace agendy doktorského studia“ vytvořilo analytickou zadávací dokumentaci pro úpravy KOSu. Jedním z definovaných cílů bylo zavést do KOSu elektronické formuláře, které nahradí většinu papírových formulářů. Každý formulář je určen pro konkrétní uživatele, kteří budou tento formulář zpracovávat bez nutnosti tisku jejich papírové podoby. Všechny formuláře určené k elektronizaci jsou podrobně rozepsány v analytické dokumentaci „Zadávací dokumentace pro úpravy KOS“ část „1.1 - Formuláře“ na webu [wiki.cvut.cz](http://wiki.cvut.cz). Dokumentace je veřejně dostupná. [18]

Životní cyklus všech formulářů popsaných v dokumentaci odpovídá následujícímu stavovému diagramu (2.1). Stav „Ke schválení“ u některých formulářů se může rozpadnout na další stavy v závislosti na počtu schvalovacích stran. Existují také formuláře, u kterých není schvalovací proces a nemůžou být zamítnuté.



Obrázek 2.1: Stavový diagram formuláře [4]

### Popis stavů [4]

1. Vytvořený - formulář je vytvořen a odeslán uživatelem. Formulář lze editovat v tomto stavu, do doby odeslání do schvalovacího procesu.
2. Ke schválení - formulář je ve schvalovacím procesu. Uživatel ve schvalovacím procesu může formulář schválit, zamítnout nebo vrátit k úpravě. Žádná další možnost neexistuje.
3. Schválený - formulář je schválen a uložen.
4. K úpravě - schvalovatel vrátil formulář zadavateli, aby ho upravil, případně doplnil a pustil do schvalovacího procesu od začátku.
5. Zamítnutý - formulář prošel schvalovacím procesem ale byl zamítnut. Pokud zadavatel bude chtít odeslat formulář do schvalovacího procesu, musí založit novou žádost.

Dokumentace popisuje celkem 17 různých typů žádostí pro doktorské studium, které CZM plánuje elektronizovat. Některé z následujících formulářů jsou již implementované nebo částečně implementované. Tyto formuláře jsou [18]:

## 2. UPŘESNĚNÍ CÍLŮ A POPIS ŘEŠENÉHO PROBLÉMU

---

1. F1 - Přihláška k doktorskému studijnímu programu
2. F2 - Rámcové téma disertační práce
3. F3 - Změna osobních údajů
4. F4 - Individuální studijní plán
5. F5 - Předmět vyučovaný na jiné univerzitě
6. F6 - Průběžné hodnocení doktoranda
7. F7 - Žádost o potvrzení o studiu
8. F8 - Žádost o uznání předmětu
9. F9 - Žádost o změnu formy studia
10. F10 - Přihláška k odborné rozpravě
11. F11 - Přihláška ke státní doktorské zkoušce
12. F12 - Žádost o povolení obhajoby disertační práce
13. F13 - Posudek školitele
14. F14 - Posudek oponenta
15. F15 - Žádost o přerušování doktorského studia
16. F16 - Žádost o ukončení doktorského studia
17. F17 - Detaily přijímacího řízení

Po konzultaci s vedoucím závěrečné práce jsem provedl výběr klíčových procesů jak pro studijní oddělení, tak i vhodných pro elektronizaci. Z našeho pohledu procesy F9, F10, F11, F12 a F15 mají největší význam jelikož jejich elektronizace je poměrně složitější než elektronizace ostatních procesů, což nám umožní ukázat větší možnosti Camundy. Po detailnějším průzkumu jsme vybrali dva finální procesy pro následnou elektronizaci v další části této práce. Usoudili jsme, že procesy **F12 - Žádost o povolení obhajoby disertační práce** a **F15 - Žádost o přerušování doktorského studia** jsou nejlepší kandidáti pro kompletní elektronizaci. Hlavní důvody výběru těchto procesů jsou:

1. Procesy zahrnují schvalovací proces s více než jediným schvalovatelem.
2. Procesy dodnes ještě nebyly implementované ani se to neplánuje v průběhu následujícího roku.



### 2.3. Upřesnění hlavního cíle závěrečné práce a jeho rozpad na další podcíle

---

3. F15 - tento proces má větší dopad na ostatní procesy (posuny termínů). Navazuje na další proces opětovného zápisu do studia.
4. F12 - proces obsahuje několik různých skupin uživatelů. Existuje velká pravděpodobnost toho, že nebude možné zastihnout všechny schvalovatele v jeden okamžik. Tudíž vyřízení žádosti v papírové podobě je časově náročné.

## 2.3 Upřesnění hlavního cíle závěrečné práce a jeho rozpad na další podcíle

V tento okamžik máme veškeré potřebné informace, abychom mohli přistoupit k vypracování praktické části. Pochopili jsme, že díky metodám procesního řízení lze výrazně zvýšit efektivitu řízení podniku. Víme, jaké výhody přináší metoda elektronizace podnikových procesů a jaké k tomu dnes existují nejpoužívanější softwarové nástroje. Za použití všech těchto znalostí v následující praktické části této závěrečné práce se pokusíme vyřešit následující úkoly:

1. **Analýza současného průběhu procesů F15 a F12** - rozebereme současný stav procesů na fakultě.
2. **Definice TO-BE stavu procesů F15 a F12** - na základě analýzy současného stavu vybraných procesů navrhujeme potřebné změny pro elektronizaci a procesní diagram.
3. **Zprovoznění prostředí Camunda** - popíšeme nutné kroky pro instalaci a spuštění Camundy včetně podpůrného softwaru. Provedeme nastavení všech nástrojů pro úspěšnou elektronizaci procesů.
4. **Elektronizace podnikového procesu** - je primární cíl práce. Spolu s vedoucím práce jsme rozhodli, že v rámci praktické části práce budeme elektronizovat dva procesy studijního oddělení Fakulty elektrotechnické Českého vysokého učení technického v Praze. V následující kapitole bude popsáno, s jakými procesy a jakým způsobem budeme pracovat. Popíšeme logiku proměnných a jejich typy.
5. **Zhodnocení průběhu elektronizace procesů** - provedeme analýzu výstupů z praktické části. Rozebereme problémy, které nastaly při elektronizaci, a způsoby jejich eliminace v budoucnu.
6. **Vyčíslení nákladů na pořízení nutného HW a SW** - v této podkapitole se zaměříme hlavně na časový odhad realizace, hodnocení nákladů včetně pořízení lidských a časových zdrojů, abychom mohli zajistit minimálně jednoletý provoz Camundy.

## 2. UPŘESNĚNÍ CÍLŮ A POPIS ŘEŠENÉHO PROBLÉMU

---

7. **Analýza přínosů nasazení Camundy** - definujeme určité přínosy pro vybranou organizaci a možnosti budoucího rozvoje.

---

## Praktická část

V praktické části práce se pokusíme naplnit cíle stanovené v podkapitole 2.3. Tyto cíle vyřešíme pomocí metod a nástrojů, které jsem popsal v teoretické části. Cíle budeme naplňovat postupně. Nejdřív definujeme aktuální stav procesů F12 a F15 a následně definujeme TO-BE stavy těchto procesů. Po finální definici provedeme kompletní elektronizaci procesů včetně podrobného popisu jednotlivých částí a nutných podmínek elektronizaci. Na konci kapitoly spočítáme náklady na elektronizaci a definujeme přínosy pro studijní oddělení FEL ČVUT.

### 3.1 Definice průběhu procesů F15 a F12

#### 3.1.1 Analýza současného průběhu procesů F15 a F12

Současné procesy F15 a F12 nejsou ani částečně elektronizované. Tyto procesy normálně na fakultě existují a jsou zajištěné v papírové podobě. Existuje jenom stručná zadávací dokumentace obecných požadavků na rozšíření funkcionality informačního systému KOS [16] a dokument se stručným popisem všech formulářů určených k elektronizaci [18]. Avšak tato dokumentace je velmi nedostatečná pro kompletní elektronizaci zmíněných procesů. Podle informace od vedoucího práce procesy F12 a F15 se nebudou implementovat minimálně během následujícího roku.

Papírová podoba těchto procesů je velmi pracná a časově náročná. Po schůzce s referentkou pro doktorské studium bylo zjištěno, že průměrný čas zpracování těchto žádostí je 14 dnů. Jelikož se může stát že nějaký účastník procesů bude mít dovolenou nebo služební cestu a nebude možné jeho vyjádření zajistit, čas zpracování se protahuje o další dny.

Jak bylo psáno výše, na FEL studuje více než 400 studentů doktorského studia. Každý student má svoji fyzickou papírovou složku na studijním oddělení, kam se ukládají všechny tištěné dokumenty studenta včetně všech podaných žádostí. Archivace těchto dokumentů je složitý proces jak z hlediska

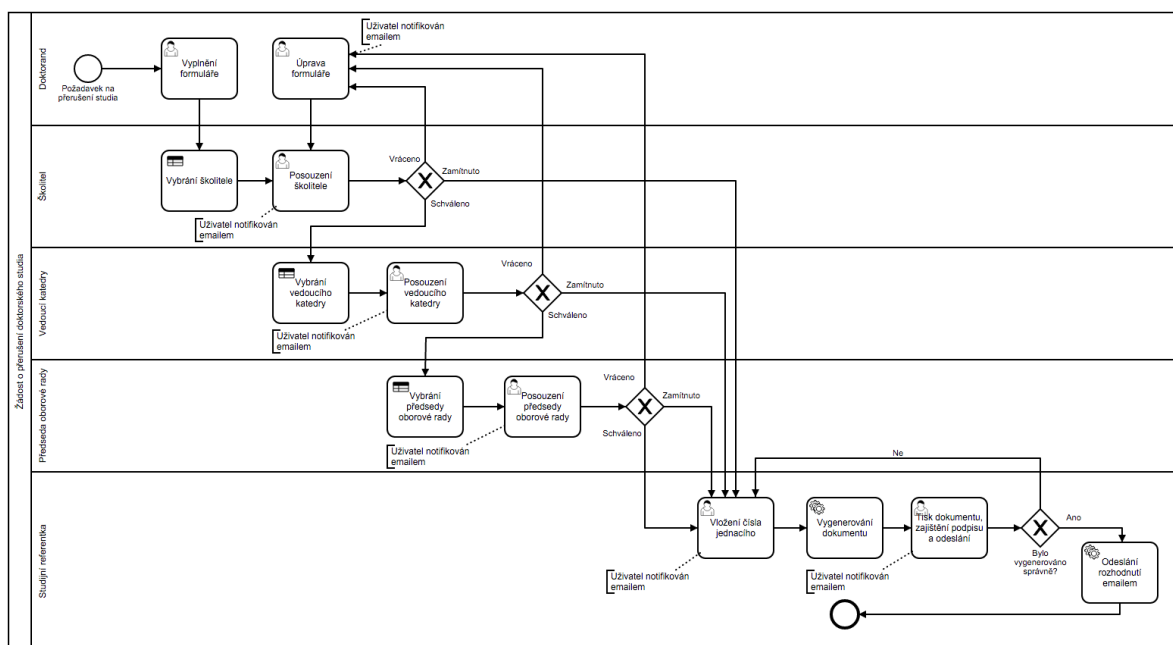
### 3. PRAKTICKÁ ČÁST

časového, tak i z hlediska bezpečnosti. Za tyto dokumenty aktuálně odpovídají jenom dvě referentky studijního oddělení pro doktorské studium.

Elektronizace procesů F12 a F15 by především zkrátila dobu zpracování žádosti, jelikož nebude potřeba se vyjadřovat k žádostem fyzicky a stačí jenom prostřednictvím procesního portálu. Na druhou stranu by pomohla k správě dokumentů vzniklých během těchto procesů (archivace, dohledávání), protože tyto dokumenty budou uloženy elektronicky na serveru.

#### 3.1.2 Definice TO-BE stavu procesů F15 a F12

##### 3.1.2.1 Průběh procesu F15 ve formě procesního diagramu



Obrázek 3.1: Procesní diagram procesu F15

#### Průběh procesu ve formě textového popisu

Proces „Žádost o přerušení doktorského studia“ lze využít kdykoliv v průběhu akademického roku. Proces je určen pro doktorandy, kteří se rozhodli z nějakých důvodů přerušit studium na FEL. Doktorand má následující možnosti v procesu:

1. Vytvoření žádosti o přerušení studia.
2. Úprava žádosti v případě vracení k přepracování.

#### 3. Seznámení s rozhodnutím prostřednictvím elektronické pošty.

Další uživatelé procesu, jako školitel, vedoucí katedry a předseda oborové rady, mají stejné možnosti v procesu:

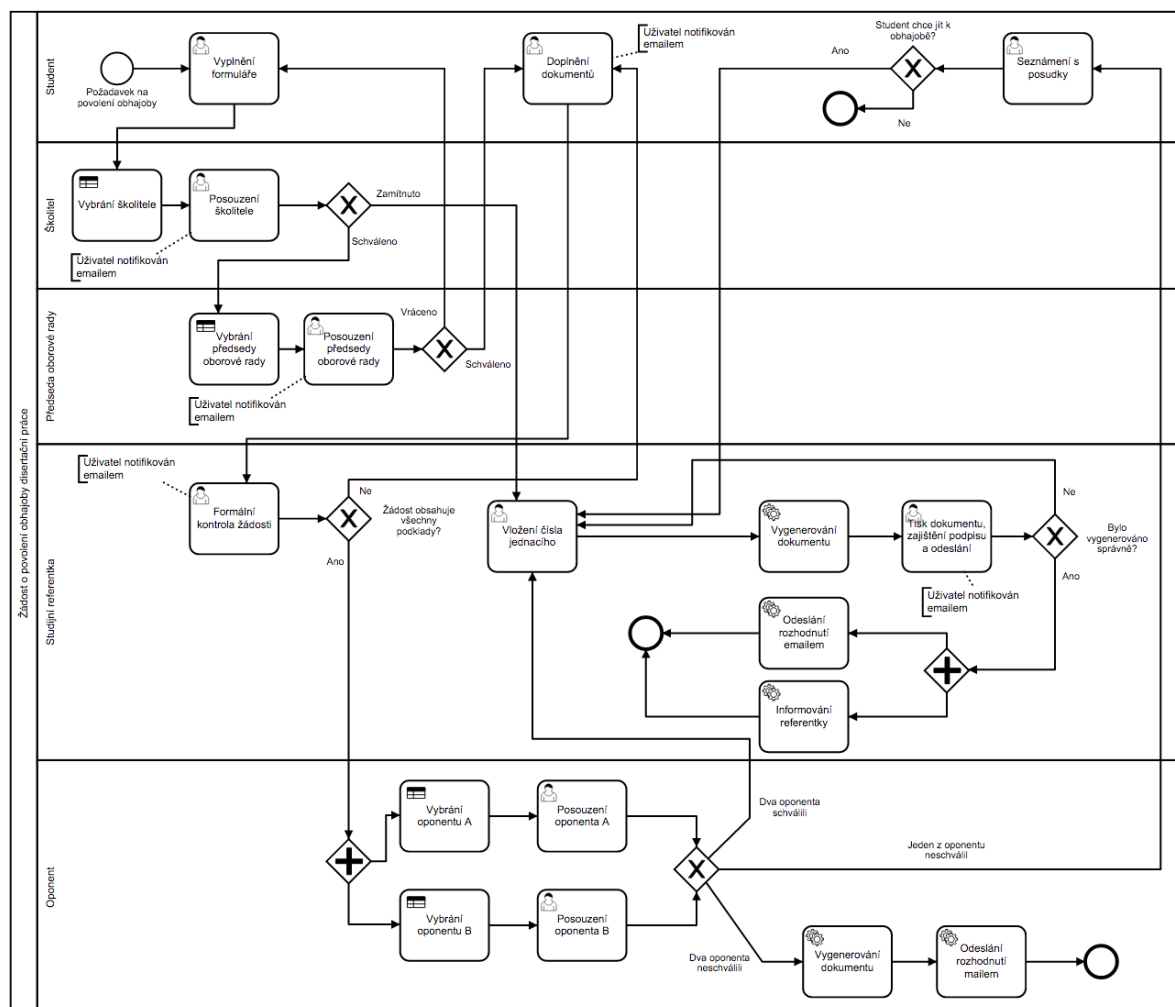
1. Schválit, vrátit k přepracování doktorandovi a kompletně zamítnout žádost.
2. Vlastní vyjádření v podobě komentáře k žádosti.

Poslední uživatel procesu je studijní referentka, která zajišťuje tisk, podpis a odeslání rozhodnutí k podané žádosti. Její funkce v procesu jsou:

1. Vložení čísla jednacího k žádosti.
2. Zajištění podpisu a odeslání žádosti doktorandovi poštou.

### 3. PRAKTICKÁ ČÁST

#### 3.1.2.2 Průběh procesu F12 ve formě procesního diagramu



Obrázek 3.2: Procesní diagram procesu F12

#### Průběh procesu ve formě textového popisu

Proces je určen pro studenty, kteří složili státní doktorskou zkoušku a chtějí se přihlásit k obhajobě své disertační práce. Na rozdíl od procesu F15 student musí spolu s žádostí přiložit odpovídající dokumenty. Do procesu jsou zapojeni student doktorského studia, školitel, předseda oborové rady, oponent a studijní referentka.

Student má při procesu následující možnosti:

1. Vytvoření žádosti o povolení obhajoby disertační práce.

2. Doplnění potřebných dokumentů.
3. Seznámení s rozhodnutím prostřednictvím elektronické pošty.

Školitel má při procesu následující možnosti:

1. Vložit posudek k práci.
2. Vlastní vyjádření v podobě komentáře k žádosti.
3. Schválit nebo kompletně zamítnout žádost.

Předseda oborové rady má následující možnosti:

1. Vlastní vyjádření v podobě komentáře k žádosti.
2. Schválit nebo vrátit k přepracování studentovi.

Studijní referentka má následující možnosti:

1. Vložení čísla jednacího k žádosti.
2. Zajištění podpisu a odeslání žádosti doktorandovi poštou.
3. Vlastní vyjádření v podobě komentáře k žádosti.
4. Schválit nebo vrátit k přepracování studentovi.

Oponent má následující možnosti:

1. Vložení posudku.
2. Vlastní vyjádření v podobě komentáře k žádosti.
3. Schválit nebo zamítnout žádost.

## 3.2 Zprovoznění prostředí Camunda

V této kapitole bude popsáno, jakým způsobem se elektronizuje proces v nástroji Camunda a nastavení všech k tomu potřebných úprav. Obsah této části závěrečné práce slouží jako referenční manuál pro uživatele, aby díky této implementaci se seznámil s prostředím Camundy a byl schopen stejný nebo podobný proces elektronizovat samostatně.

Aby uživatel úspěšně elektronizoval proces, je potřeba splnit následující implementační kroky.

1. **Instalace a příprava Camundy a podpůrného softwaru**
2. **Příprava Java projektu pro BPMN 2.0**
3. **Modelování procesu v Camunda Modeler**

Každý krok podrobněji rozebereme v samostatných podkapitolách. Kroky jsou řazené chronologicky, tudíž pokračovat v dalším kroku se doporučuje jenom v případě splnění kroků předcházejících. Samozřejmě některé mezikroky lze vynechat, avšak abychom co nejvíce eliminovali možnost chyby, budeme ty mezikroky popsané v podkapitolách považovat jako nutné k provedení.

#### 3.2.1 Instalace a příprava Camundy a podpůrného softwaru

Camunda umožňuje svým uživatelům několik různých způsobů práce s Camunda BPM. Mimo klasického BPMN 2.0 nabízí také CMMN 1.1, DMN 1.1, Spring Framework, Java EE 7 a Apache Maven. Vzhledem ke specifice práce pro naše účely budeme pracovat s BPMN 2.0.

Abychom mohli nainstalovat Camundu BPM a Camundu Modeler, nejprve musíme ověřit, aby naše vývojové prostředí obsahovalo:

1. Java JDK 1.7+
2. Apache Maven (nepovinné, pokud není instalováno, můžeme využít už integrovaný Maven v Eclipsu)
3. Moderní internetový prohlížeč (Firefox, Chrome nebo Internet Explorer 9+)

##### 3.2.1.1 Instalace Camundy BPM

Je potřeba stáhnout distribuci Camundy BPM. Pro různé aplikační servery lze stáhnout různé distribuce. My budeme pracovat s distribucí pro Apache Tomcat. Na webových stránkách Camundy (<https://camunda.org/download/>) lze stáhnout aktuální verzi Camundy BPM. Nejaktuálnější stabilní verze k okamžiku vypracování této práce je 7.6.0 (od 30.11.2016). Pro instalaci stačí stáhnout cílový archiv a rozbalit na lokálním disku. Této složce budeme říkat „HOME\_CAMUNDA“. Během elektronizace procesu jsem narazil na problém lokalizace českého jazyka. Po komunikaci s oddělením podpory Camundy bylo zjištěno, že se jedná o chybu na jejich straně a jako „workaround“ zatím můžeme ručně změnit kódování. Je potřeba přidat řádek „set JAVA\_OPTS=-Dfile.encoding=UTF-8“ do souboru „HOME\_CAMUNDA/camunda-bpm-tomcat-7.6.0/server/apache-tomcat/bin/setenv.sh“ nebo „setenv.bat“ v případě Windows distribuce. Pro spuštění Camundy BPM na OS Windows je třeba nastavit systémovou proměnou „JAVA\_HOME“ (pokud není nastavena) na odpovídající „JDK“ knihovnu. Na Internetu existuje hodně návodů, jak to udělat, například [https://www.mkyong.com/java/how-to-set-java\\_home-on-windows-10/](https://www.mkyong.com/java/how-to-set-java_home-on-windows-10/).



### 3.2.1.2 Instalace Camundy Modeler

Camunda Modeler je desktopová aplikace pro práci s procesními diagramy v notaci BPMN, „case management“ diagramy a rozhodovacími tabulkami. Přehlednost a jasnost uživatelského rozhraní umožňuje pracovat v tomto nástroji jak vývojářům, tak i buisness analytikům. Camunda Modeler je podporován operačními systémy Windows, Linux a MacOS. My budeme vytvářet diagramy v modeleru pro MacOS 10.12.1. Ale je to úplně jedno, na jakém OS budeme pracovat, jelikož výstupní soubor z modeleru je stejný pro všechny platformy. Instalační soubor je dostupný na adrese <https://camunda.org/download/modeler/>. Po stažení stačí archiv rozbalit a spustit „Camunda Modeler.app“.

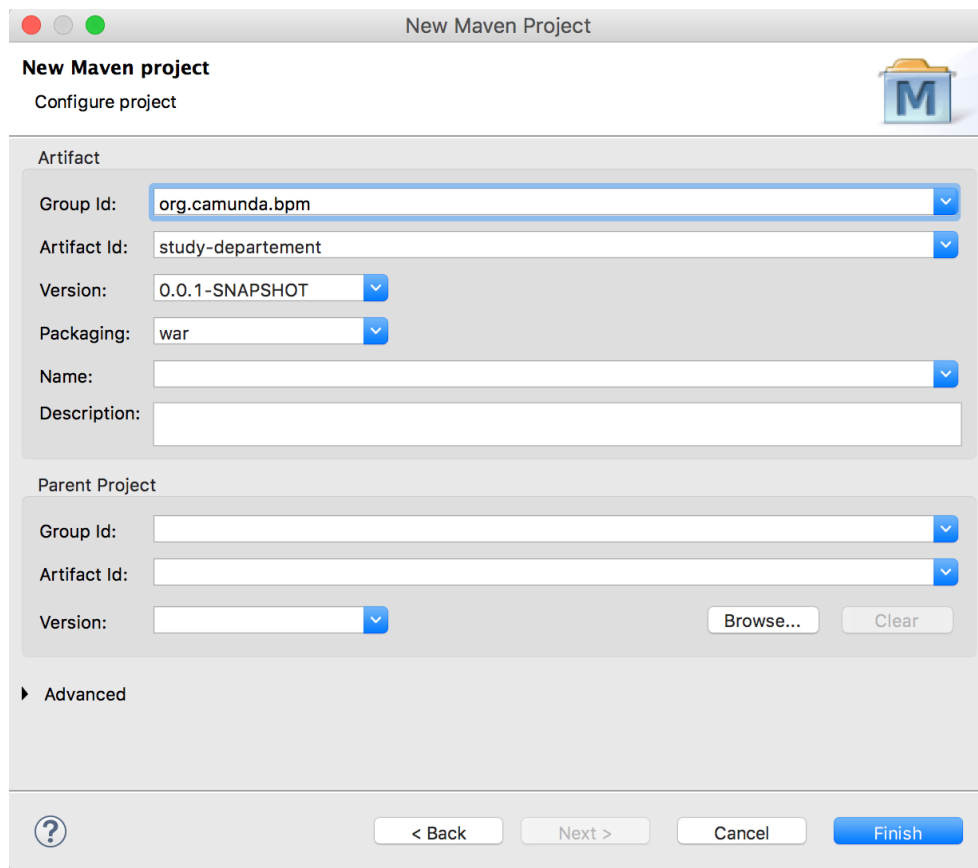
### 3.2.1.3 Instalace Eclipse

Eclipse je platforma pro vývoj softwaru. Samostatná platforma představuje relativně omezený počet funkcionalit. Hlavní myšlenka práce s tímto vývojovým prostředím je rychlý vývoj díky rozšíření funkcionality o vlastní pluginy [19]. V roce 2015 Camunda podporovala plugin pro modelování přímo v Eclipsu, ale tuto funkčnost již přestala podporovat v roce 2016, a proto je nutné mít nainstalovaný modeler jako desktopovou aplikaci. Poslední verze Eclipse Neon je dostupná na adrese <https://www.eclipse.org/downloads/>. Po spuštění instalačního balíčku (Eclipse Installer) bude uživateli nabízena instalace různých vývojových prostředí, ale nás konkrétně zajímá „Eclipse IDE for Java Developers“. Podrobný návod instalace včetně konfigurací je popsán během instalace. Projektovou složku nastavíme na „HOME\_CAMUNDA/workspace“.

## 3.2.2 Příprava Java projektu pro BPMN 2.0

Po instalaci a nastavení domovského adresáře je potřeba vytvořit nový Maven projekt. Maven projekt je speciální nástroj od společnosti Apache Software Foundation, který určen pro správu a automatizaci aplikací založených přednostně na Javě [20]. Nastavení projektu je popsáno na následující obrázku. Je důležité zvolit formát projektu „war“ pro pochopení Camundou.

### 3. PRAKTICKÁ ČÁST



Obrázek 3.3: Nastavení Maven projektu

V následujícím kroku je potřeba nastavit náš Maven projekt jako procesní aplikaci. Proto musíme doplnit již existující soubor „pom.xml“ v kořenovém adresáři našeho projektu „study-department“. Po doplnění závislostí z následujícího kódu, projekt lze zkompileovat jako procesní aplikaci (vybrat „pom.xml“, pravý klik a vybrat „Run as / Maven install“).

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.camunda.bpm</groupId>
      <artifactId>camunda-bom</artifactId>
      <version>7.6.0</version>
      <scope>import</scope>
      <type>pom</type>
    </dependency>
  </dependencies>
</dependencyManagement>

<dependencies>
  <dependency>
    <groupId>org.camunda.bpm</groupId>
    <artifactId>camunda-engine</artifactId>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

```

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.0.1</version>
  <scope>provided</scope>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.3</version>
      <configuration>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
  </plugins>
</build>

```

Abychom mohli využít možnosti odeslání emailu a generování PDF dokumentu během našich procesů, je třeba přidat odpovídající „maven dependencies“. Pro využití doplňujících knihoven přidáme další závislosti do souboru „pom.xml“, jsou to:

```

<dependencies>

  <dependency>
    <groupId>org.camunda.bpm.extension</groupId>
    <artifactId>camunda-bpm-mail-core</artifactId>
    <version>1.0.0</version>
  </dependency>

  <dependency>
    <groupId>org.apache.pdfbox</groupId>
    <artifactId>pdfbox</artifactId>
    <version>1.8.13</version>
  </dependency>

  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-email</artifactId>
    <version>1.2</version>
  </dependency>
</dependencies>

```

Dále je třeba vytvořit nový Java Class v „src/main/java“, v našem případě s názvem balíčku „org.camunda.bpm.proapp“. Třídou pojmenujeme „StudentProcessApplication“. Tato třída tvoří rozhraní mezi naší aplikací a procesním enginem. Nastavení procesní třídy:

```

package org.camunda.bpm.proapp;

import org.camunda.bpm.application.ProcessApplication;
import org.camunda.bpm.application.impl.ServletProcessApplication;

@ProcessApplication
public class StudentProcessApplication extends ServletProcessApplication{

}

```

Poslední krok v přípravě projektu je vytvoření „META-INF/processes.xml“ v adresáři „src/main/resources“. Ten soubor umožňuje konfiguraci spuštění procesu do procesního enginu [3]. Pro defaultní nastavení je možné nechat soubor prázdným.

### 3.3 Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

#### 3.3.1 Vytvoření a nastavení uživatelských účtů

Pro ukázkou elektronizovaných procesů založíme několik skupin uživatelů, nutných pro běh procesů. Po spuštění Camunda BPM v sekci Admin musíme vytvořit tyto uživatele rozdělené podle kategorie:

Username	Heslo	Role v procesu	Skupina uživatele
mentor1	mentor1	školitel A	mentors
mentor2	mentor2	školitel B	mentors
mentor3	mentor3	školitel C	mentors
head1	head1	vedoucí katedry A	headDeps
head2	head2	vedoucí katedry B	headDeps
head3	head3	vedoucí katedry C	headDeps
chairman1	chairman1	předseda oborové rady A	chairmen
chairman2	chairman2	předseda oborové rady B	chairmen
chairman3	chairman3	předseda oborové rady C	chairmen
studref1	studref1	studijní referentka A	studref
student1	student1	student A	students
student2	student2	student B	students
student3	student3	student C	students

Tabulka 3.1: Uživatelské účty

Po vytvoření uživatelů je nutné nastavit odpovídající práva v Camundě pro všechny skupiny uživatelů. Pro naše účely bude stačit přidat všem skupinám právo na spuštění „Task Listu“ a právo přihlášení do Camundy. Studentovi navíc přidat právo spuštění procesu.

Pro odeslání notifikací o nových úlohách v portálu nastavíme všem osobám soukromé e-maily podle následující tabulky. Z testovacích důvodů jsem vytvořil každé skupině uživatelů jenom jeden e-mail. V budoucnu lze tyto emaily zaměnit za reálné uživatelské e-maily.

### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

Skupina uživatelů	E-mail	Heslo k e-mailu
mentors	mentor.camunda@gmail.com	camunda123
headDeps	headdep.camunda@gmail.com	camunda123
chairmen	chairman.camunda@gmail.com	camunda123
studref	studref.camunda@gmail.com	camunda123
students	student.camunda@gmail.com	camunda123

Tabulka 3.2: Testovací e-maily skupin uživatelů

### 3.3.2 Modelování procesů v Camunda Modeler

#### 3.3.2.1 Modelování procesu F15 - Žádost o přerušení studia

Naše zvolené procesy F15 a F12 budeme modelovat podle principu znázorněného na diagramu 2.1. Výsledné procesy se budou trochu lišit od původního diagramu, ale celkový koncept bude zachován. Nejprve je potřeba připravit rozhraní stran účastníků procesu. V tom nám pomůžou plavecké dráhy. Hlavní plaveckou dráhu pojmenujeme „Žádost o přerušení doktorského studia“ a definujeme počáteční nastavení dráhy. V panelu nastavení (Properties Panel) upravíme následující hodnoty.

Hodnota	Název
Name	Žádost o přerušení studia
Process Id	F15process
Process Name	Žádost o přerušení studia
Executable	ANO (checkbox)
ID	Je generováno automaticky

Tabulka 3.3: Nastavení procesu F15

Další vnořené dráhy pojmenujeme podle účastníků procesu. Účastníky procesu přerušení doktorského studia jsou: student, školitel, vedoucí katedry, předseda oborové rady a studijní referentka. Každý proces se začíná počáteční událostí. V našem případě počáteční událost vzniká na straně studenta s názvem „Požadavek na přerušení studia“. Abychom nebyli omezeni nějakým určitým studentem, do hodnoty „Initiator“ zadáme proměnnou „requestor“, do které se pak automaticky přiřadí hodnota toho, kdo bude spouštět proces. Počáteční událost bude mít jenom jeden atribut, a to je upozornění pro uživatele o zahájení procesu (proměna „alert“ typu „string“ s hláškou „Klepnutím na tlačítko „Start“ zahájíte proces podání žádosti.“).

#### User Task - Vyplnění žádosti

### 3. PRAKTICKÁ ČÁST

---

Po zahájení procesu student musí vyplnit žádost. Proto dalším krokem v modelování bude element „User Task“. Pojmenujeme ho „Vytvoření formuláře“. Do pole „Assignee“ je třeba vepsat hodnotu „\${requestor}“, aby úloha skutečně vznikla na straně zadavatele procesu. Formulář děláme podle existujícího formuláře a lze se na něj podívat na <https://www.fel.cvut.cz/cz/education/forms/>. Náš formulář bude obsahovat úplně stejné položky. V tomto user tasku je třeba definovat, jaké pole bude uživatel upravovat. Existuje několik možností, jak definovat atributy pro vyplnění uživatelem. Nejjednodušší způsob je vytvoření a nastavení těchto proměnných přímo v modeleru. Avšak během vypracování práce bylo zjištěno, že u některých proměnných vzniká problém s nastavením přístupových práv. Z tohoto důvodu formuláře pro všechny účastníky procesu budeme vytvářet v HTML formátu. Camunda podporuje HTML formuláře a poskytuje více možností než standardní modeler. V Eclipse ve složce projektu „HOME\_CAMUNDA/workspace/student-departement/src/main/webapp/“ vytvoříme dokument „studentForm.html“, kde definujeme následující atributy:

ID	Type	Label	Validace
firstName	string	Jméno	required
lastName	string	Příjmení	required
studentSubject	enum	Studijní obor	required
mentor	string	Školitel	required
stopFrom	date	Přerušení studia OD	required
stopTill	date	Přerušení studia DO	required
reason	string	Zdůvodnění žádosti	required

Tabulka 3.4: Atributy úlohy „Vyplnění formuláře“

Atribut „studentSubject“ typu „enum“ (list) má následující hodnoty.

Value ID	Name
o1	Obor A
o2	Obor B
o3	Obor C
o4	Obor D

Tabulka 3.5: Hodnoty atributu „studentSubject“

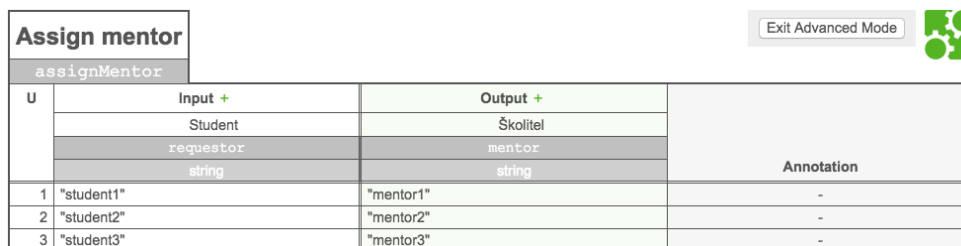
Kompletní návod, jak definovat proměnné v HTML formátu, aby je rozpoznala Camunda, lze najít na adrese <https://docs.camunda.org/get-started/bpmn20/forms/>. Nebo se podívat přímo na už vypracovaný soubor v přílohách k této práci „Příloha A“. Poslední krok v tomto „User Tasku“ je přidání cesty

### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

k vytvořenému formuláři. V modeleru na našem tasku do záložky „Forms“ přidáme cestu k souboru „embedded:app:studentForm.html“.

#### Business Rule Task - Vybrání školitele

Každý student doktorského studia má od začátku studia přiřazeného školitele. Proto vybrání školitele probíhá na základě identity studenta. Výběr školitele nám umožní rozhodovací tabulka, kterou lze vytvořit v Camunda Modeler (záložka „Create new DMN table“). Přístupový klíč k tabulce nastavíme na proměnnou „assignMentor“. Vstupní atribut v tomto případě bude proměnná „requestor“, obsahující účet uživatele, který spouští proces. Výstupní atribut „mentor“ bude obsahovat účet vedoucího, kterému bude přiřazená úloha. Rozhodovací DMN tabulka vypadá následně:



Assign mentor		Output +		Annotation
U	Input +	Školitel		
	Student	mentor		
	requestor	string		
1	"student1"	"mentor1"	-	
2	"student2"	"mentor2"	-	
3	"student3"	"mentor3"	-	

Obrázek 3.4: Rozhodovací tabulka pro výběr školitele

Tabulku uložíme do projektové složky „HOME\_CAMUNDA/workspace/student-departement/src/main/resources/“. Poslední krok ve výběru vedoucího je nastavení modelovacího elementu „Vybrání školitele“ podle následujících parametrů:

1. **ID** - chooseMentor
2. **Name** - Vybrání školitele
3. **Implementation** - DMN
4. **Decision Ref** - assignMentor
5. **Binding** - latest
6. **Result Variable** - mentor
7. **Map Decision Result** - singleEntry

### User Task - Posouzení školitele

Po vyplnění a odesílání žádosti musí vzniknout nová úloha na straně školitele práce. Úloha bude mít název „Posouzení školitele“ a atribut „Assignee“ nastaven na „{Mentor}“, abychom přiřadili úlohu osobě zvolené studentem v původní žádosti. Atributy úlohy jsou totožné s atributy v tabulce 3.12. Jenom validace všech atributů je nastavena na „readonly“, aby osoba nemohla měnit obsah původně zadaných položek. Úloha „Posouzení školitele“ navíc obsahuje hodnoty, pomocí kterých se školitel bude vyjadřovat k žádosti. Tyto atributy jsou:

Value ID	Name	Type	Validation
mentorComment	Poznámka školitele	string	required
approveMentor	Vyjádření školitele	enum	required

Tabulka 3.6: Atributy úlohy „Posouzení školitele“

Každý schvalovatel v procesu má tři možnosti rozhodnutí. Aby uživatel pokračoval v procesu, musí vybrat jednu z nich.

Hodnoty atributu „approveMentor“ obsahují tyto možnosti:

Value ID	Name
v1	Vráceno
v2	Zamítnuto
v3	Schváleno

Tabulka 3.7: Hodnoty atributu „approveMentor“

Podobně jako u úlohy „Vyplnění žádosti“ vytvoříme HTML formulář s odpovídajícími hodnotami a definujeme cestu k souboru. Vypracovaný formulář lze najít na příloženém k této bakalářské práci CD disku s názvem „mentorForm.html“. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

Proces pokračuje dál na základě hodnoty atributu „approveMentor“. V případě, že školitel vrátil formulář, na straně studenta musí vzniknout nová úloha „Úprava formuláře“ (bude popsáno v následujících paragrafech). Po úpravě student posílá žádost dál k úloze „Posouzení školitele“. Pokud školitel zvolil zamítnutí žádosti, pak na straně studijní referentky vzniká úloha „Vložení čísla jednacího“ (bude popsáno v následujících paragrafech).

V případě schválení žádosti školitelem proces pokračuje dál na stranu vedoucího katedry. Tyto tři toky jsou řízené na základě atributu „approveMentor“ v události „Posouzení školitele“ a odchozím modelovacím elementu „Exclusive



### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

Gateway“. Je potřeba nastavit logiku odchozích šipek podle následujících parametrů:

Name	Condition type	Expression
Vráceno	Expression	„\${approveMentor=='v1'}“
Zamítnuto	Expression	„\${approveMentor=='v2'}“
Schváleno	Expression	„\${approveMentor=='v3'}“

Tabulka 3.8: Nastavení elementu „Exclusive Gateway“

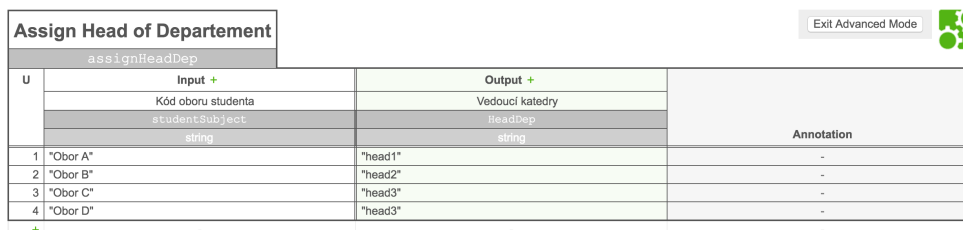
Poslední úprava na úloze bude implementace notifikace uživatele o přiřazené úloze prostřednictvím e-mailu. Tuto funkčnost zajistíme pomocí vlastní Java třídy, která bude posílat notifikaci na e-mail uživatele. V Eclipsu „HOME\_CAMUNDA/workspace/student-departement/src/main/Java/“ vytvoříme třídu „TaskAssignmentListener“. Hlavní rozhraní třídy tvoří jedna metoda „notify“, která především zjistí aktuálního uživatele, dostane e-mail příslušného uživatele a odešle zprávu uživateli. Kompletní kód s komentáři lze najít v přílohách k práci [B]. Je třeba nastavit parametry úlohy v záložce „Listeners - Task Listener“ podle následujících parametrů.

1. **Event Type** - create (zajistí spuštění třídy po vytvoření úlohy)
2. **Listener Type** - Java Class
3. **Java Class** - org.camunda.bpm.proapp.TaskAssignmentListener

### Business Rule Task - Vybrání vedoucího katedry

Pokud školitel schválil žádost, na straně vedoucího katedry vzniká Business Rule Task „Vybrání vedoucího katedry“. Pomocí rozhodovací DMN tabulky přiřadíme úlohu odpovídajícímu vedoucímu.

Rozhodovací tabulka „assignHeadDep.dmn“ vypadá následně:



Assign Head of Departement		assignHeadDep		Exit Advanced Mode
U	Input +	Output +	Annotation	
	Kód oboru studenta	Vedoucí katedry		
	studentSubject	headDep		
	string	string		
1	"Obor A"	"head1"	-	
2	"Obor B"	"head2"	-	
3	"Obor C"	"head3"	-	
4	"Obor D"	"head3"	-	
+	-	-	-	

Obrázek 3.5: Rozhodovací tabulka pro výběr vedoucího katedry

### 3. PRAKTICKÁ ČÁST

---

Rozhodovací tabulku lze vytvořit v Camunda Modeler (záložka Create new DMN table). Přístupový klíč k tabulce nastavíme na proměnou „assignHeadDep“. Rozhodování, jakému vedoucímu přiřadit úlohu, provedeme na základě zvoleného studentem oboru, jelikož víme, jaké obory má každý vedoucí katedry na starosti. Proto vstupní rozhodovací parametr nastavíme na „studentSubject“ a na základě jeho hodnoty určíme vedoucího katedry. Výstupní atribut „HeadDep“ bude obsahovat vybraného vedoucího. Tabulku uložíme do projektové složky „HOME\_CAMUNDA/workspace/student-department/src/main/resources/“. Poslední krok ve výběru vedoucího je nastavení modelovacího elementu „Vybrání vedoucího katedry“ podle následujících parametrů:

1. **ID** - chooseHeadDep
2. **Name** - Vybrání vedoucího katedry
3. **Implementation** - DMN
4. **Decision Ref** - assignHeadDep
5. **Binding** - latest
6. **Result Variable** - HeadDep
7. **Map Decision Result** - singleEntry

#### User Task - Posouzení vedoucího katedry

Po vybrání školitele pomocí DMN tabulky, následný krok v procesu je úloha „Posouzení vedoucího katedry“ na straně vedoucího katedry. Parametr úlohy „Assignee“ nastavíme na proměnnou „#{HeadDep}“. Úloha má totožné atributy jako úloha „Posouzení školitele“ bez atributu „approveMentor“, validaci atributu „mentorCommet“ je nutné nastavit na „readonly“. Specifické atributy pro posouzení vedoucího katedry jsou:

Value ID	Name	Type	Validation
HeadDepComment	Poznámka vedoucího katedry	string	required
approveHeadDep	Vyjádření vedoucího katedry	enum	required

Tabulka 3.9: Atributy úlohy „Posouzení vedoucího katedry“

Podobně jako u úlohy „Vyplnění žádosti“ vytvoříme HTML formulář s odpovídajícími hodnotami a ukážeme cestu k souboru. Vypracovaný formulář lze najít na příloženém k této bakalářské práci CD disku s názvem

### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT


„HeadDepForm.html“. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

Atribut „approveHeadDep“ má totožné hodnoty popsané v tabulce 3.7. V tomto okamžiku vzniká shodná situace jako u posouzení školitele. Liší se v tom, že element „Exclusive Gateway“ je řízen podle parametru „approveHeadDep“ a má totožné hodnoty popsané v tabulce 3.8. V případě, že vedoucí katedry žádost vrátil, na straně studenta musí vzniknout úloha „Úprava žádosti“ (bude popsáno v následujících paragrafech). Po úpravě na straně studenta musí proces zase projít schválením školitele. Pokud vedoucí katedry zamítl žádost, na straně studijní referentky vzniká úloha „Vložení čísla jednacího“ (bude popsáno v následujících paragrafech).

Notifikaci uživatele o nové úloze zajistíme úplně stejným způsobem jako u školitele, jenže teď není potřeba vytvářet nový Java Class. Stačí jenom nastavit parametry úlohy „Listeners - Task Listener“ podle návodu popsaného na konci části „User Task - Posouzení školitele“.

#### Business Rule Task - Vybrání předsedy oborové rady

Tento Business Rule Task funguje na stejném principu jako úloha „Vybrání vedoucího katedry“. Za předpokladu, že vedoucí katedry žádost schválil, pomocí DMN tabulky vybereme odpovídající osobu ze skupiny předsedů. Přístupový klíč nastavíme na „assignChairman“. Na základě oboru jsme schopni určit příslušného předsedu. Proto vstupním parametrem bude „studentSubject“. Výstupním atributem bude „chairman“.



The screenshot shows a configuration window for a Business Rule Task named "Assign chairman". The window has a title bar with "Assign chairman" and "Exit Advanced Mode" buttons. Below the title bar is a table with the following structure:

Input +		Output +		Annotation
U				
	Kód oboru studenta		Předseda oborové rady	
	studentSubject		chairman	
	string		string	
1	*Obor A*	*chairman1*		-
2	*Obor B*	*chairman2*		-
3	*Obor C*	*chairman3*		-
4	*Obor D*	*chairman3*		-

Obrázek 3.6: Rozhodovací tabulka pro výběr předsedy ORO

Nastavení modelovacího elementu „Vybrání předsedy oborové rady“ upravíme podle následujících parametrů:

1. **ID** - chooseChairman
2. **Name** - Vybrání předsedy oborové rady
3. **Implementation** - DMN
4. **Decision Ref** - assignChairman

### 3. PRAKTICKÁ ČÁST

---

5. **Binding** - latest
6. **Result Variable** - chairman
7. **Map Decision Result** - singleEntry

#### User Task - Posouzení předsedy oborové rady

Finální schvalovatel procesu je předseda oborové rady. Za předpokladu, že vedoucí katedry žádost schválil, vzniká na straně předsedy úloha „Posouzení předsedy oborové rady“. Jako u předchozích schvalovatelů má úloha stejné atributy s úlohou „Posouzení školitele“ bez atributu „approveMentor“, validaci atributů „mentorComment“ a „HeadDepComment“ je nutné nastavit na „readonly“). Jako ostatní, předseda musí mít svoje atributy pro vyjádření. Jsou to:

Value ID	Name	Type	Validation
chairmanComment	Poznámka předsedy oborové rady	string	required
approveChairman	Poznámka předsedy oborové rady	enum	required

Tabulka 3.10: Atributy úlohy „Posouzení předsedy oborové rady“

Hodnoty atributu „approveChairman“ je potřeba nastavit podle tabulky 3.7. Modelovací element „Exclusive Gateway“ je tentokrát řízen podle hodnot „approveChairman“. Jelikož žádný další schvalovatel v procesu není, existují dvě možnosti pokračování v procesu. Pokud předseda vrátil formulář, tak musí být vytvořena úloha na straně studenta „Úprava žádosti“. Po úpravě studentem proces pokračuje na straně školitele v úloze „Posouzení školitele“. Po rozhodnutí předsedy (žádost schválena nebo zamítnuta) proces pokračuje na straně studijní referentky v úloze „Vložení čísla jednacího“ (bude popsáno v následujících částech).

Podobně jako u úlohy „Vyplnění žádosti“ vytvoříme HTML formulář s odpovídajícími hodnotami a definujeme cestu k souboru. Vypracovaný formulář lze najít na příloženém k této bakalářské práci CD disku s názvem „chairmanForm.html“. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

Notifikaci uživatele o nové úloze zajistíme úplně stejným způsobem jako u školitele, jenže teď není potřeba vytvářet nový Java Class. Stačí jenom nastavit parametry úlohy „Listeners - Task Listener“ podle návodu popsaného na konci části „User Task - Posouzení školitele“.

### User Task - Úprava formuláře

Tato úloha vzniká na straně studenta v momentě, kdy někdo ze schvalovatelů procesu vrátil formulář k přepracování studentovi. Studentovi se zobrazuje stejný formulář jako v úloze „Vyplnění formuláře“ s přidáním atributů „mentorComment“, „HeadDepComment“, „chairmanComment“ s validací „readonly“, aby uživatel viděl důvod vrácení žádosti k přepracování. Po opravě formuláře proces pokračuje na straně školitele. V modeleru u úlohy nastavíme parametr „Assignee“ na „\${requestor}“.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „repairForm.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

### User Task - Vložení čísla jednacího

Studijní referentka nemá rozhodování v této žádosti. V této úloze musí jenom vložit číslo jednací pro následné generování a poslání dopisu studentovi. V našem procesu budeme mít jenom jednu konkrétní referentku, avšak v budoucnu lze přidat rozhodování výběru referentky na základě popisu v úloze „Business Rule Task - Vybrání vedoucího katedry“.

Formulář pro studijní referentku je stejný jako formulář v úloze „Posouzení předsedy oborové rady“. Jenom atribut „HeadDepComment“ nastaven na „readonly“. Vlastní atribut pro vložení čísla jednacího je „mailNumber“ typu „String“ a validaci „required“.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „studrefForm.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

### Service Task - Vygenerování dokumentu

Po vložení čísla jednacího musí proběhnout generování dokumentu. Generování probíhá na pozadí a zajištěno speciální Java třídou. V Eclipsu „HOME\_CAMUNDA/workspace/student-departement/src/main/Java/“ vytvoříme třídu „CreatePdf“ [C]. Pomocí rozhraní „JavaDelegate“ můžeme přistupovat k procesním proměnným včetně editace a čtení. Proto je třeba zajistit správnou definici nutné metody „execute“ a importování potřebných knihoven. Více o rozhraní „JavaDelegate“ lze najít na <https://docs.camunda.org/manual/7.5/user-guide/process-engine/delegation-code/>.

Jelikož Camunda neposkytuje možnost generování dokumentu ve formátu PDF, musíme využít externí knihovny. Po doporučení služby technické podpory Camundy jsem vybral knihovnu „Apache PDFBox“. Hlavní důvod je

### 3. PRAKTICKÁ ČÁST

---

dostatek potřebných pro naše účely možností a je veřejně dostupná na adrese <https://pdfbox.apache.org/>.

Žádost se generuje na základě schvalovacích atributů „approveMentor“, „approveHeadDep“ a „approveChairman“. Výstupem mohou být 4 typy rozhodnutí. Jsou to:

1. Žádost zamítnuta školitelem
2. Žádost zamítnuta vedoucím katedry
3. Žádost zamítnuta předsedou oborové rady
4. Žádost schválena

Výsledná žádost se ukládá do projektové složky „HOME\_CAMUNDA/Decisions/“ Kompletně vypracovanou třídu lze najít v přílohách k práci.

#### **User Task - Tisk dokumentu, zajištění podpisu a odeslání**

Po vygenerování dokumentu musí studijní referentka zkontrolovat dokument, fyzicky zajistit podpisy a odeslat rozhodnutí studentovi na poštu. Vygenerovaný dokument se zasílá referentce na e-mail pro následnou kontrolu. To zajišťuje speciální Java třída „DocumentSendListener“. Její rozhraní je skoro totožné s třídou „TaskAssignmentListener“ [B], pouze tentokrát e-mail v třídě má typ „MultiPartEmail“, což umožňuje posílat e-mail s přílohou. Kompletně rozpracovanou třídu lze najít na příloženém k práci CD disku. Způsob napojení třídy je popsán v předchozích částech.

Formulář pro studijní referentku je stejný jako formulář v úloze „Vložení čísla jednacího“, bez atributu vložení čísla jednacího a navíc je přidán atribut „generated“ typu checkbox potvrzující správnost vygenerovaného dokumentu. V případě, že je checkbox odškrtnout, proces pokračuje úlohou „Vložení čísla jednacího“. Pokud referentka zaškrtnla potvrzující checkbox, proces pokračuje v Service Tasku „Odeslání rozhodnutí mailem“.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „checkForm.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### **Service Task - Odeslání rozhodnutí emailem**

Finální Service Task, kde se odesílá rozhodnutí žádosti studentovi. Jelikož je to Service Task, kterému nelze nastavit řešitele, nemůžeme využít třídu „TaskAssignmentListener“ [B], kde určujeme e-mail na základě aktuálního řešitele úlohy. Musíme vytvořit Java třídu „FinalNotification“. Implementace třídy je skoro totožná s třídou „DocumentSendListener“. Rozdíl je v nasazení rozhraní „Java Delegate“, protože potřebujeme zjistit uživatele, který začal

celý proces (hodnota proměnné „requestor“). Proto není třeba zjišťovat aktuálního řešitele, ale rovnou nastavit na zadavatele procesu. Implementaci třídy lze najít na přiloženém CD k práci.

### 3.3.2.2 Modelování procesu F12 - Žádost o povolení obhajoby disertační práce

Popis procesu v této sekci bude stručnější než popis procesu F15, protože z hlediska implementace se tento proces moc neliší od procesu F15, ale byznys logika tohoto procesu je dost odlišná. Z toho důvodu v popisu procesu budu podrobněji rozepisovat odlišné části od procesu F15 a odkazovat na proces F15 v případě totožných nebo velmi podobných částí implementace.

Největší rozdíl v implementaci spočívá v přidání nové funkcionality oproti procesu F15. Tato funkcionality je možnost nahrání a stažení souborů přímo ve formuláři přidělené úlohy. Hlavní myšlenkou procesu F12 je předání dokumentů studenta schvalovatelům, kteří posoudí, jestli student může jít k obhajobě, nebo ne. Proto tato funkčnost je klíčová v tomto procesu.

Nejdřív nastavíme výchozí parametry procesu v panelu „Properties Panel“ podle následujících parametrů:

Hodnota	Název
Name	Žádost o povolení obhajoby disertační práce
Process Id	F12process
Process Name	Žádost o povolení obhajoby disertační práce
Executable	ANO (checkbox)
ID	Je generováno automaticky

Tabulka 3.11: Nastavení procesu F12

Definice plaveckých drah je totožná s definicí v procesu F15. Účastníci procesu povolení obhajoby disertační práce jsou: student, školitel, předseda oborové rady a studijní referentka. Počáteční událost vzniká na straně studenta s názvem „Požadavek na obhajobu disertační práce“. Abychom nebyli omezeni nějakým určitým studentem, do hodnoty „Initiator“ zadáme proměnnou „requestor“, do které se pak automaticky přiřadí hodnota toho, kdo bude spouštět proces. Počáteční událost bude mít jenom jeden atribut, a to je upozornění pro uživatele o zahájení procesu (proměna „alert“ typu „string“ s hláškou „Klepnutím na tlačítko „Start“ zahájíte proces podání žádosti.“).

#### User Task - Vyplnění formuláře

### 3. PRAKTICKÁ ČÁST

---

Žádost o povolení obhajoby disertační práce má více atributů než žádost o přerušení studia, ale implementace rozhraní formuláře je stejná jako v procesu F15. Je třeba vytvořit formulář pro uživatele s následujícími parametry:

ID	Type	Label	Validace
firstName	string	Jméno	required
lastName	string	Příjmení	required
degree	string	Titul	required
birthDate	Date	Datum narození	required
birthPlace	string	Místo narození	required
birthNumber	string	Rodné číslo	required
contactAddress	string	Adresa pro doručování	required
studentSubject	enum	Studijní obor	required
department	enum	Katedra	required
studyType	enum	Forma studia	required
studyStartDate	Date	Datum zahájení studia	required
masterExamDate	Date	Datum vykonání státní doktorské zkoušky	required
thesisCZ	string	Název disertační práce - česky	required
thesisEN	string	Název disertační práce - anglicky	required
chairmanComment	string	Komentář předsedy oborové rady	readonly

Tabulka 3.12: Atributy úlohy „Vyplnění formuláře“

Specifický element žádosti je „`THESIS_MASTER_DOCUMENT`“. Je to proměnná typu „File“, která bude obsahovat práci studenta ve formátu PDF. Camunda umožňuje implementaci nahrání a stažení dokumentu pomocí HTML formuláře. Podrobný popis práce s nahráním a stažením souborů pomocí HTML formulářů lze najít na adrese <https://docs.camunda.org/manual/7.5/reference/embedded-forms/controls/files/>. Definice proměnné pro nahrání dokumentu vypadá následujícím způsobem:

```
<div class="form-group">
<label for="THESIS_MASTER_DOCUMENT">Master Thesis in PDF format</label>
<input type="file"
  cam-variable-name="THESIS_MASTER_DOCUMENT"
  cam-variable-type="File"
  cam-max-file-size="1000000" />
</div>
```

Název proměnné musí být unikátní v rámci jednoho procesu. Je třeba dát na to pozor, aby nedošlo k přepsání obsahu proměnné. Vypracovaný formulář studenta lze najít na příloženém k práci CD. Formulář má stejný název jako formulář pro vyplnění studentem v procesu F15 „studentForm.html“, ale je



### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

uložen v doplňující složce „F12“ pro eliminaci kolize s procesem F15. Pro detailnější popis nastavení této úlohy viz podkapitulu 3.3.2.1, úloha „User Task - Vyplnění žádosti“.

#### **Business Rule Task - Vybrání školitele**

Vybrání školitele na základě rozhodovací DMN tabulky je realizováno úplně stejným způsobem jako v procesu F15, viz 3.3.2.1 úloha „Business Rule Task - Vybrání školitele“.

#### **User Task - Posouzení školitele**

Školitel po obdržení úlohy musí vložit svůj posudek k práci studenta, kterou může stáhnout přímo z formuláře úlohy. Po vložení posudku rozhodnout, zda tuto žádost schvaluje, nebo zamítá, a napsat k tomu důvod. To všechno zajišťují následující atributy formuláře školitele. Jsou to:

Value ID	Name	Type	Validation
mentorComment	Poznámka školitele	string	required
approveMentor	Rozhodnutí školitele	enum	required

Tabulka 3.13: Atributy úlohy „Posouzení školitele“

Speciální atribut pro vložení posudku formou PDF přílohy je „OPINION\_MASTER\_DOCUMENT“ typu „File“. Vypracovaný formulář lze najít na příloženém k práci CD s názvem „mentorForm.html“.

Implementace notifikace uživatele je řešena stejným způsobem jako v procesu F15, viz 3.3.2.1 úloha „User Task - Posouzení školitele“.

#### **Business Rule Task - Vybrání předsedy oborové rady**

Vybrání předsedy oborové rady na základě rozhodovací DMN tabulky je realizováno úplně stejným způsobem jako v procesu F15, viz 3.3.2.1 úloha „Business Rule Task - Vybrání předsedy oborové rady“.

#### **User Task - Posouzení předsedy oborové rady**

Jak bylo zmíněno na začátku této kapitoly, předseda může žádost schválit nebo vrátit studentovi na přepracování. V případě vrácení studentovi musí proces zase projít školitelem, než se dostane k předsedovi. Unikátní atributy předsedy pro posouzení:

### 3. PRAKTICKÁ ČÁST

---

Value ID	Name	Type	Validation
chairmanComment	Poznámka předsedy oborové rady	string	required
approveChairman	Poznámka předsedy oborové rady	enum	required

Tabulka 3.14: Atributy úlohy „Posouzení předsedy oborové rady“

Nastavení notifikace uživatele o vzniklé úloze a nastavení „assignment“ atributu je popsáno v kapitole 3.3.2.1 úloha „User Task - Posouzení předsedy oborové rady“. Vypracovaný formulář lze najít na příloženém k práci CD s názvem „chairmanForm.html“.

#### User Task - Doplnění dokumentů

Tato úloha vzniká na straně studenta v případě, že předseda oborové rady schválil žádost. Student musí doložit zbývající potřebné dokumenty. Ne všechny dokumenty lze doložit elektronicky z důvodu, že podle směrnice ČVUT, jsou vyžadované originály některých dokumentů. Po konzultaci s vedoucím jsem definoval dokumenty, které student nebo školitel může přiložit pomocí elektronického formuláře. Jsou to:

1. Žádost o povolení obhajoby - samotný HTML formulář, vyplněný studentem.
2. Doktorská práce v elektronické podobě - práce ve formátu PDF.
3. Životopis - dokument ve formátu PDF.
4. Seznam vlastních publikací - dokument ve formátu DOCX.
5. Seznam předmětů absolvovaných mimo FEL - dokument ve formátu DOCX.
6. Posudek školitele - vložen školitelem v úloze „User Task - Posouzení školitele“.
7. Teze - dokument ve formátu PDF.

Konkrétně v této úloze student musí nahrát životopis, seznam vlastních publikací, seznam předmětů absolvovaných mimo FEL a tezi. Za to odpovídají 4 samostatné atributy typu „File“. Pokud soubor nebude nahrán, Camunda automaticky neumožní pokračovat v procesu.

Notifikace uživatele, formulář a nastavení úlohy jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“ procesu F15. Formulář má název „studentFormRepair.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### User Task - Formální kontrola žádosti

Po vložení potřebných dokumentů studentem proces pokračuje na straně studijní referentky. Na rozdíl od procesu F15, studijní referentka může žádost vrátit studentovi na doplnění s vlastní poznámkou. Referentka zkontroluje žádost hlavně na správnost vložených podkladů. Speciální atributy pro posouzení jsou:

Value ID	Name	Type	Validation
referentComment	Poznámka studijní referentky	string	required
approved	Žádost obsahuje všechny potřebné podklady?	checkbox	required

Tabulka 3.15: Atributy úlohy „Formální kontrola žádosti“

Notifikace uživatele, formulář a nastavení úlohy jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“ procesu F15. Formulář má název „referentForm.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### User Task - Vložení čísla jednacího

Pokud oba oponenty schválili žádost nebo školitel, na straně studijní referentky vznikne nová úloha vložení čísla jednacího, aby systém vygeneroval odpovídající rozhodnutí a referentka mohla zajistit odeslání tohoto rozhodnutí studentovi poštou. V našem procesu budeme mít jenom jednu konkrétní referentku, avšak v budoucnu lze přidat rozhodování výběru referentky na základě popisu v úloze „Business Rule Task - Vybrání vedoucího katedry“.

Formulář pro studijní referentku je stejný jako formulář v úloze „Formální kontrola žádosti“. V tomto formuláři pouze nebudeme zobrazovat příložené dokumenty, jelikož referentka se k nim vyjádřila v předchozí úloze. Vlastní atribut pro vložení čísla jednacího je „mailNumber“ typu „String“ a validaci „required“.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „studrefForm.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### Business Rule Task - Vybrání oponentu A

Vybrání prvního oponentu na základě rozhodovací DMN tabulky je realizováno podobným způsobem jako v procesu F15, viz 3.3.2.1 úloha „Business Rule Task - Vybrání předsedy oborové rady“. Výstupní atribut je „oponentA“.

### 3. PRAKTICKÁ ČÁST

Assign oponentA			
chooseOponentA			
U	Input +	Output +	Annotation
	Student	Oponent	
	requestor	oponentA	
	string	string	
1	"student1"	"mentor1"	-
2	"student2"	"mentor2"	-
3	"student3"	"mentor3"	-
+	-	-	-

Obrázek 3.7: Rozhodovací tabulka pro výběr prvního oponenta

#### Business Rule Task - Vybrání oponentu B

Vybrání druhého oponentu na základě rozhodovací DMN tabulky je realizováno podobným způsobem jako v procesu F15, viz 3.3.2.1 úloha „Business Rule Task - Vybrání předsedy oborové rady“. Výstupní atribut je „oponentB“.

Assign oponentB			
chooseOponentB			
U	Input +	Output +	Annotation
	Student	Oponent	
	requestor	oponentB	
	string	string	
1	"student1"	"chairman1"	-
2	"student2"	"chairman2"	-
3	"student3"	"chairman3"	-
+	-	-	-

Obrázek 3.8: Rozhodovací tabulka pro výběr druhého oponenta

#### User Task - Posouzení oponenta A

Oponent se musí vyjádřit k práci studenta vložím posudku a nastavením hodnoty atributu „approveOponentA“ na odpovídající hodnoty (zamítnuto nebo schváleno). Speciální atributy pro posouzení jsou:

Value ID	Name	Type	Validation
oponentCommentA	Poznámka oponenta A	string	required
approveOponentA	Rozhodnutí oponenta A	enum	required

Tabulka 3.16: Atributy úlohy „Posouzení oponenta A“

### 3.3. Elektronizace procesů studijního oddělení doktorského studia FEL ČVUT

Jelikož úloha běží v paralelním vlákně s úlohou „User Task - Posouzení oponenta A“, musíme nastavit parametr úlohy „Asynchronous After“ a „Exclusive“ na hodnotu „true“, aby úloha počkala na dokončení paralelního vlákna. Je to nutné aby při rozhodování podle proměnných v úlohách „User Task - Posouzení oponenta A“ a „User Task - Posouzení oponenta B“ nevznikl problém nenačtení proměnných do paměti.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „oponentFormA.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### User Task - Posouzení oponenta B

Oponent se musí vyjádřit k práci studenta vložením posudku a nastavením hodnoty atributu „approveOponentB“ na odpovídající hodnoty (zamítnuto nebo schváleno). Speciální atributy pro posouzení jsou:

Value ID	Name	Type	Validation
oponentCommentB	Poznámka oponenta A	string	required
approveOponentB	Rozhodnutí oponenta A	enum	required

Tabulka 3.17: Atributy úlohy „Posouzení oponenta B“

Je třeba nastavit parametry „Asynchronous After“ a „Exclusive“ na hodnotu „true“ (viz předchozí paragraf).

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „oponentFormB.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### User Task - Seznámení s posudky

Pokud alespoň jeden z oponentu zamítl žádost studenta, vzniká úloha ve které student se musí seznámit s posudky a následně rozhodnout zda chce jít k obhajobě nebo zamítnout žádost. Speciální atribut pro rozhodnutí je „approveStudent“. V případě, že student chce jít k obhajobě, proces pokračuje dál na straně studijní referentky.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „studentFormInfo.html“ a lze se na něj podívat na příloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### Service Task - Vygenerování dokumentu

Vygenerování PDF dokumentu žádosti je skoro stejným způsobem jako v pro-

### 3. PRAKTICKÁ ČÁST

---

cesu F15, viz 3.3.2.1 úloha „Service Task - Vygenerování dokumentu“. Protože v této žádosti používáme jiné proměnné, musíme vytvořit další Java třídu „CreatePdfF12.java“. Rozhraní této třídy se liší od třídy „CreatePdf.java“ [C] tím, že v nové třídě přestupujeme do proměnných definovaných ve formulářích procesu F12. Jinak logicky implementace je skoro totožná.

#### **User Task - Tisk dokumentu, zajištění podpisu a odeslání**

Po vygenerování dokumentu studijní referentka musí zkontrolovat dokument, fyzicky zajistit podpisy a odeslat rozhodnutí studentovi na poštu. Vygenerovaný dokument se zasílá referentce na e-mail pro následnou kontrolu. Implementace úlohy je totožná s implementací úlohy v procesu F15, viz 3.3.2.1 úloha „User Task - Tisk dokumentu, zajištění podpisu a odeslání“.

Formulář pro studijní referentku je stejný jako formulář v úloze „Vložení čísla jednacího“, bez atributu vložení čísla jednacího a navíc je přidán atribut „generated“ typu checkbox potvrzující správnost vygenerovaného dokumentu. V případě, že je checkbox odškrtnout, proces pokračuje v úloze „Vložení čísla jednacího“. Pokud referentka zaškrtnla potvrzující checkbox, proces pokračuje v Service Tasku „Odeslání rozhodnutí mailem“ a „Informování referentky“.

Notifikace uživatele a formulář jsou řešeny stejným způsobem jako v úloze „Posouzení školitele“. Formulář má název „checkForm.html“ a lze se na něj podívat na přiloženém CD k této práci. Tento formulář je vytvořen na stejném principu jako formulář „studentForm.html“ [A].

#### **Service Task - Odeslání rozhodnutí emailem**

Odeslání rozhodnutí e-mailem je realizováno úplně stejným způsobem jako v procesu F15, viz 3.3.2.1 úloha „Service Task - Odeslání rozhodnutí emailem“.

#### **Service Task - Informování referentky**

Informování referentky je nutné, aby referentka byla upozorněna, že může spouštět další proces stanovení komise. Pro notifikaci vytvoříme novou třídu „ReferentNotification.java“ na základě už existující třídy pro notifikaci „TaskAssignmentListener.java“ [B]. Ale tentokrát nepotřebujeme zjišťovat řešitele úlohy, abychom zjistili jeho email, jelikož víme, že máme v systému jenom jednu studijní referentku. Proto třídní proměnnou „assignee“ nastavíme rovnou na účet referentky „studref1“. Přílohu vyřešíme stejně jako v třídě „FinalNotification.java“. Vypracovanou třídu lze najít na přiloženém CD k této práci.

### 3.3.3 Spuštění procesů

Poslední krok elektronizace procesů je spuštění procesů. Za předpokladu, že jsou splněny kroky v podkapitolách 3.3.1 a 3.3.2, můžeme spustit námi připravené procesy. Pro úspěšné spuštění je potřeba udělat následující kroky:

1. Spustit server Apache Tomcat - pomocí terminálu ve složce Camundy BPM (složka „HOME\_CAMUNDA/camunda-bpm-tomcat-7.6.0“) spustit script „start-camunda.sh“ nebo „start-camunda.bat“ v případě OS Windows. V standardním webovém prohlížeči se otevře úvodní stránka Camundy s rozcestníkem.
2. Namodelované procesy a DMN tabulky zkopírovat do domovské složky Eclipse přímo do našeho projektu „study-departement/src/main/resources/“. HTML formuláře do složky „study-departement/src/main/webapp/“.
3. Zkompilovat proces - vybrat v projektové složce Eclipse soubor „study-departement/pom.xml“, pravý klik a zvolit „Run as / Maven install“. Poté ve složce „study-departement/target“ se objeví ke spuštění připravený soubor „study-departement-0.0.1-SNAPSHOT.war“.
4. Soubor „study-departement-0.0.1-SNAPSHOT.war“ zkopírovat do serverové složky „HOME\_CAMUNDA/camunda-bpm-tomcat-7.6.0/server/apache-tomcat/webapps“. Server automaticky rozbalí balíček přibližně během 10 vteřin.

Teď jsou procesy úspěšně elektronizovány a lze je spustit. Do Camundy BPM se lze přihlásit pomocí výchozích účtů v tabulce 3.1. Výchozí nastavení Camundy je pro nás postačující, abychom provedli ukázkou elektronizovaného procesu. Jenom uživatelé ze skupiny „students“ mají oprávnění spouštět proces, proto je nutné primárně se přihlásit pod jedním z těchto uživatelů. V Camunda Tasklist po zmáčknutí tlačítka „Start Process“ uživatel uvidí nabídku ke spuštění připravených procesů. Výběrem procesu (např. Žádost o přerušování studia) uživatel přejde podle modelu do startovací události. Chování procesu je popsáno v předchozí kapitole. Ukázkou průchodem elektronizovanými procesy z uživatelského pohledu je k nalezení v příloze D a E.





---

## Vyhodnocení praktické části

### 4.1 Zhodnocení průběhu elektronizace

V této kapitole podrobně rozebereme všechny výstupy z předchozí praktické části. Jedná se o manažerské shrnutí výstupů, doporučení pro budoucí vývoj. Na konci také spočítám veškeré náklady na zajištění provozu Camundy a rozeberu konkrétní přínosy pro chod studijního oddělení pro doktorské studium FEL ČVUT.

Výstupy z předchozí praktické části jsou:

1. Analýza současného průběhu procesů F12 a F15 a návrh jejich TO-BE stavů.
2. Návod k zprovoznění prostředí Camundy a podpůrného softwaru.
3. Připravené ke spuštění elektronizované procesy.

#### 4.1.1 Zhodnocení analýzy a návrhu procesů F12 a F15

Během analýzy stávajícího stavu procesů bylo zjištěno, že připravená dokumentace představuje jenom seznamovací charakter. Výsledný návrh procesů je založen na základě zkušeností mého vedoucího v oblasti návrhu podnikových procesů, mých osobních schůzek s referentkou studijního oddělení pro doktorské studium FEL, vnitřních přepisů ČVUT a zkoumáním „best practice“ v oblasti modelování procesů. Výstupní procesní diagramy odpovídají reálnému průběhu procesů na fakultě, tudíž v případě, že nedojde k reálnému nasazení Camundy na FEL, tyto diagramy lze stejně využít pro vysvětlení chodu procesů ostatním byznys uživatelům.

### 4.1.2 Zhodnocení návodu zprovoznění Camundy a podpůrného softwaru

Návod k instalaci je přizpůsoben pro konkrétní elektronizaci vybraných procesů. Lze ale podle tohoto návodu připravit prostředí pro většinu procesů zmíněných v kapitole 2.2. Svoji implementaci jsem začínal na verzi Camundy 7.5.0, avšak během práce společnost vydala další stabilní verzi 7.6.0, kde bylo opraveno 148 bugů vzhledem k poslední stabilní verzi. Celková úprava prostředí pro novou verzi byla z mé strany minimální, ale pomohla se vyhnout potenciálním problémům, které mohly vzniknout během implementace. Pokud čtenář bude chtít implementovat nestandardní rozšíření (jako v mém případě generování PDF dokumentů), doporučuji projít fórum Camundy, kde se řeší různé problémy a návrhy na implementaci. Fórum je veřejně dostupné na adrese <https://forum.camunda.org/>.

### 4.1.3 Zhodnocení elektronizace procesů

Výstupem z předchozí kapitoly jsou dva ke spuštění připravené elektronizované procesy. Celková elektronizace dvou procesů mně zabrala 260 hodin čistého času.

Během implementace se vyskytly následující závažné problémy:

1. Lokalizace českého jazyka - během implementace jsem se narazil na problém zobrazení specifických písmen české abecedy. Řešení problému je popsáno v sekci 3.2.1.1.
2. Načtení formulářů vytvořených přímo v Camunda Modeler - problém vznikl v momentě implementace úlohy „User Task - Vyplnění formuláře“. Ve formuláři se vyskytují proměnné s validací „readonly“, aby je uživatel nemohl měnit. Avšak v následujících úlohách při nastavení validace těchto proměnných na „required“, Camunda hlásila chybu. Z tohoto důvodu jsem se rozhodl implementovat HTML formuláře.

Nejpracnější částí implementace bylo zprovoznění odesílání reálných notifikací e-mailem a generování PDF rozhodnutí o žádosti. Na těchto dvou částech jsem strávil cca 40 procent z celkového počtu odpracovaných hodin. Ve vygenerovaných PDF jsem neřešil grafické formátování, jelikož to není cílem práce.

Abychom výsledné procesy F15 a F12 reálně spustili do provozu FEL, je třeba provést několik implementačních úprav. Tyto úpravy jsou:

1. Provést integraci serverové části Camundy s existujícím IS KOS.
2. Do serverové části Camundy napojit reálnou databázi studentů a pracovníků ČVUT.
3. Upravit grafickou podobu výsledných žádosti ve formátu PDF.

Během implementace se ukázalo, že proces F12 lze vytvořit pomocí úpravy procesu F15. Tato úprava hlavně spočívala v přidání možnosti nahrání a stažení souborů pomocí HTML formulářů (viz 3.3.2.2). Podařilo se výrazně zkrátit čas na implementaci procesu F12 díky správně nazřeným Java třídám v procesu F15, které jsem použil s drobnými úpravami, nebo dokonce i bez úprav (třída „TaskAssignmentListener.java“) [B].

## 4.2 Vyčíslení nákladů

Tato část práce se věnuje nákladům na zprovoznění Camundy, pořízení nutného softwaru a hardwaru, zajištění podpory. Výpočet nákladů je orientován na zajištění alespoň dvouletého provozu na fakultě elektrotechnické ČVUT. Náklady rozdělíme na 4 skupiny:

1. Zakoupení a instalace pevných součástí.
2. Zakoupení a instalace programového zabezpečení.
3. Elektronizace procesů F12 a F15.
4. Podpora provozu Camundy.

### 4.2.1 Zakoupení a instalace pevných součástí

Především je třeba zajistit stabilní server, který je schopen zpracovávat procesy a sloužit jako úložiště dat. Vzhledem k tomu, že na tento server v budoucnu lze navázat složitější procesy než F12 a F15 a rozšířit možnosti stávajících procesů, doporučuji udělat výkonovou rezervu pro budoucí rozvoj. Jak bylo zmíněno v kapitole 2.1, na doktorských programech studuje více než 400 lidí. Navíc musíme počítat lidi spojené s doktorským studiem, tj. pedagogy, pracovníky studijní části. Server by měl být postaven podle následujících parametrů:

<b>Procesor</b>	Intel Xeon E3-1225 v3 (3.2 GHz, 8MB Cache)	8.900 Kč
<b>Operační paměť</b>	16GB (1600MHz, DDR3)	2.900 Kč
<b>Pevný disk</b>	1TB	1.600
<b>Operační systém</b>	Linux distribuce	0 Kč
<b>Zdroj</b>	600W	1.300 Kč
<b>Celková cena</b>		<b>14.700 Kč</b>

Tabulka 4.1: Základní konfigurace serveru

Není nutné, aby server byl konfigurován přesně podle popsaných parametrů. Parametry popsané v tabulce 4.1 je třeba považovat za minimální.

#### 4. VYHODNOCENÍ PRAKTICKÉ ČÁSTI

---

Serverový počítač lze sestavit manuálně nebo zakoupit již hotové řešení odpovídající požadovaným parametrům (např. Dell PowerEdge T20, HP Proliant, Fujitsu Primergy apod.). V takovém případě není nutné řešit doplňující náklady v podobě kabelů, šroubů, skříní a dalších součástí potřebných k fyzické instalaci. Celková minimální cena serveru bez instalace je **23.990 Kč** (cena Dell PowerEdge T20) nebo **14.700 Kč** bez ceny skříně, drátů apod. Velký rozdíl mezi těmito cenami spočívá ve výběru operačního systému. V případě hotové sestavy Dell PowerEdge T20 má uživatel v ceně OS Microsoft Server 2012. Separátně tato distribuce stojí **22.600 Kč** OEM licence. Všechny ceny jsou orientační ke dni 25.12.2016 v České republice podle jednoho z největších poskytovatelé výpočetní techniky „Alza.cz a.s“ (IČO 27082440).

Typ	Popis	Celková cena
Typ A	Zakoupení součástí zvlášť a jejich samostatná instalace, OS Linux	14700 Kč (cena zahrnuje jenom základní součástky)
Typ B	Zakoupení hotového serveru Dell PowerEdge T20 (nebo podobný), předplacený OS Microsoft Server 2012	23990 Kč
Typ C	Zakoupení součástí zvlášť a jejich samostatná instalace, OS Microsoft Server 2012	37300 Kč (cena zahrnuje jenom základní součástky)

Tabulka 4.2: Možnosti zakoupení serveru

Optimální řešení pro zakoupení budou volby A a B. Typ C je náročnější z hlediska instalace a dražší kvůli samostatné licenci OS Microsoft Server 2012. Výhoda volby B především spočívá v šetření času na zprovoznění serveru. Díky nainstalovanému OS Microsoft Server 2012 uživatel má od začátku možnost práce s nebo bez GUI, jasné ovládání díky Server Manageru, DAC (Dynamic Access Control) a řadu dalších podpůrných rozšíření (více na <https://www.microsoft.com/cs-cz/cloud-platform/windows-server>). Volba serveru typu B je dobrá v případě potřeby rychlého zprovoznění serveru a spravování alespoň 75 procentní zátěže, což v našem případě není. V našem případě se elektronizace procesů předpokládá až za rok, navíc se předpokládá provoz jenom procesů F12 a F15, tudíž většina rozšíření by byla zbytečná.

Výhoda volby A spočívá v menší ceně a možnosti většího přizpůsobení serveru potřebám FEL díky otevřenému OS Linux. Konfigurace a instalace s OS Linux zabere více času. Obecně OS Linux má více výhod oproti OS Windows z důvodu bezpečností (většina uživatelů nemá „root“ oprávnění, skupina uživatelů Linux je výrazně menší než Windows), inteligentnější podpora aktualizací, možnost konfigurovat správné programové vybavení dle vlastních

potřeb. Dle mého názoru volba serveru typu A bude pro FEL nejlepší z hlediska dlouhodobého využití.

#### 4.2.2 Zakoupení a instalace programového zabezpečení

Jednou z hlavních výhod Camundy je veřejná dostupnost. Avšak společnost poskytuje možnost zakoupení „Enterprise“ verze Camundy. Pro počáteční provoz v prvním roce není třeba zakupovat rozšířenou verzi. Funkčnost veřejné „Community“ verze bohatě postačí pro provoz procesů na fakultě. Placenou distribuci má smysl zakoupit až ve druhém roce provozu nebo v okamžiku, kdy se fakulta rozhodne implementovat rozsáhlé doplnění vlastních funkcí a zprovoznění Camundy v ostatních oblastech FEL. „Enterprise“ verze především umožňuje prohledávat historii spuštění procesů, instancí procesů, editaci DMN tabulek přímo v průběhu procesu apod. Na základě osobního telefonního hovoru s představitelem Camundy jsem zjistil, že základní minimální licence Camundy bude stát 44.900 EUR (cca 1.212.300 Kč) ročně. Ten poplatek lze vždy zrušit v jakýkoliv okamžik a všechny „Enterprise“ rozšíření zůstanou. Avšak v momentě zrušení poplatku organizace ztrácí nárok na online podporu a nasazení nových „Patch“ releasů pro verzi Camundy. Přesná cena se počítá na základě počtu procesních instancí, nikoliv na základě počtu uživatelů. Základní „Enterprise“ licence zahrnuje 60.000.000 procesních uzlů. Více o „Camunda Enterprise“ lze najít na adrese „<https://camunda.com/bpm/enterprise/>“.

Nástroje Camunda Modeler a Eclipse Neon jsou veřejně dostupné. Místo Eclipse Neon lze využít jiné vývojové prostředí, avšak mnou nebyly nalezeny žádné nedostatky při práci s tímto programem.

#### 4.2.3 Elektronizace procesů F12 a F15

Jak jsem psal výše, na elektronizaci procesu jsem strávil 260 hodin práce. Elektronizace zahrnuje analýzu a návrh procesů, modelování, implementaci funkcionality uživatelských úloh. Bohužel jsem neměl moc zkušeností s elektronizací procesů, modelováním a konfigurací serveru. V reálných podmínkách podniku by tato práce byla vypracována minimálně čtyřmi lidmi: byznys analytik (analýza, návrh a modelování procesů), vývojář (implementační část), backend vývojář (zprovoznění serveru) a tester (testování procesů, serveru). Analýza a návrh procesů F12 a F15 by zkušenému byznys analytiku zabrala 40 hodin, včetně externích schůzek a vlastní rešerše. Implementace a spuštění procesů pro Java programátora s alespoň dvouletou praxí by zabrala 80 hodin práce. Na následující tabulce lze se podívat na výslednou cenu (včetně DPH, sociální a životní pojištění) na elektronizaci procesů F12 a F15 vzhledem k průměrným platům odborníků ve sféře ICT v České republice, podle Českého statistického úřadu [21].

#### 4. VYHODNOCENÍ PRAKTICKÉ ČÁSTI

Pozice	Proces	Rozsah práce	Plat
Byznys analytik	F15 a F12	40 člh.	9600 Kč (240 Kč/hod)
Java vývojář	F15 a F12	80 člh.	20800 Kč (260 Kč/hod)
Správce serveru	F15 a F12	20 člh.	5200 Kč (260 Kč/hod)
QA tester	F15 a F12	50 člh.	8000 Kč (160 Kč/hod)
<b>Celková cena</b>	F15 a F12	190 člh.	<b>43.600 Kč</b>

Tabulka 4.3: Náklady na elektronizaci procesů F12 a F15

Hodinová alokace je spočtená na základě vlastní provedené práce na procesech F12 a F15.

#### 4.2.4 Podpora provozu Camundy

Po nasazení Camundy nezbytnou částí je podpora a údržba. Organizace musí zajistit trvalou podporu v podobě zaškolení zaměstnanců a studentů v práci s Camundou, údržbu serveru (aktualizace databáze, dostatečná bezpečnost apod.), údržbu procesů (byznys logika, statistiky použití). Pro tyto účely je nutné zajistit speciální skupinu pracovníků, která bude řešit podporu a rozvoj dalších procesů na fakultě. Pro první dva měsíce od spuštění procesů, je třeba mít trvalou alokaci týmu podpory. Hlavní úkol skupiny podpory v první měsíce provozu je oprava hlášených chyb. Nejvyšší alokaci potřebují vývojáři a QA tester. V následující tabulce bude zobrazen rozpočet a alokace jenom pro podporu Camundy a procesů F12 a F15 pro první a druhý měsíc od nasazení.

Pozice	Počet lidí	Alokace	Plat
Byznys analytik	1	20.5 hod/m	4.920 Kč
Java vývojář	1	30 hod/m	7.800 Kč
Správce serveru	1	15 hod/m	3.900 Kč
QA tester	1	40 hod/m	6.400 Kč
<b>Celkem</b>	4	105.5 hod/m	<b>23.020 Kč/m</b>

Tabulka 4.4: Skupina podpory a údržby - první 3 měsíce od spuštění procesů

Od třetího měsíce by mělo dojít ke snížení všech alokací do minima, za předpokladu že převážná většina chyb bude opravena. V této fázi by tým měl dál řešit hlášené chyby a případné změny na procesech.

Pozice	Počet lidí	Alokace	Plat
Byznys Analytik	1	10 hod/m	2.400 Kč
Java vývojář	1	15 hod/m	3.990 Kč
Správce serveru	1	10 hod/m	2.600 Kč
QA tester	1	20 hod/m	3.200 Kč
<b>Celkem</b>	4	55 hod/m	<b>12.190 Kč/m</b>

Tabulka 4.5: Skupina podpory a údržby - alokace po 3. měsících od spuštění procesů

Alokace v tabulkách 4.4, 4.5 jsou sestaveny na základě konzultaci s projektovým manažerem společnosti eMan s.r.o (IČO 27203824), která vyvíjí mobilní a webové aplikace na zakázku a působí na trhu od roku 2004. Mezi její zákazníky patří velké společnosti jako Škoda, EON, UNIQA, Česká spořitelna a řada dalších.

V případě, že organizace rozhodne implementovat další procesy a rozvoj Camundy na FEL, počet pracovníků ve skupině údržby a podpory by se měl rozšiřovat dle aktuálních potřeb. Avšak minimální skupina pro podporu Camundy a procesů F12 a F15 se musí skládat ze čtyř lidí (viz. předchozí tabulka).

Společnost Camunda podporuje své zákazníky se zakoupenou „Enterprise“ verze nejen rozšířením funkcionality standardního Cockpitu, ale poskytuje na základě SLA (Service Level Agreement) možnost kontaktování podpory každý pracovní den (8:00 - 17:00). Camunda zaručuje maximální dobu odezvy do 8 hodin v případě kritických nebo blokuujících chyb. Jak jsem psal v podkapitole 4.2.2, zakoupení „Enterprise“ verze Camundy má smysl ve chvíli pokročilého rozvoje Camundy na FEL. Pro zajištění základních procesů doktorského studia bude stačit obecný helpdesk na webu Camundy.

Společnost Camunda také poskytuje možnost zaškolení zaměstnanců přímo speciálním odborníkem. Zaškolení probíhá formou workshopu v délce 4-5 dnů. Cena jednoho dne workshopu včetně všech nákladů je 2.200 EUR (cca 59.400 Kč).

#### 4.2.5 Celkové náklady na provoz Camundy

Z předchozích podkapitol lze sestavit tabulku celkových nákladů na zajištění provozu Camundy na fakultě elektrotechnické včetně podpory a údržby procesů F12 a F15. Pomocí dat v tabulkách 4.2, 4.3, 4.4, 4.5 lze spočítat cenu nasazení a údržby Camundy během 2 let od nasazení.

#### 4. VYHODNOCENÍ PRAKTICKÉ ČÁSTI

---

Rok provozu	1	2
<b>Server typu A</b>	14.700 Kč	-
<b>Tým podpory</b>	167.940 Kč	146.280 Kč
<b>Elektronizace procesů F12 a F15</b>	43.600 Kč	-
<b>Celková cena</b>	226.240 Kč	146.280 Kč

Tabulka 4.6: Roční náklady na provoz Camundy

V tabulce 4.6 není započten pronájem místnosti pro server, jelikož FEL má vlastní prostory.

Pokud FEL rozhodne o rozšíření procesů, je třeba zajistit dostatečný počet členů týmu vývoje a podpory. To má za následek vznik mzdových nákladů pro nové pracovníky. Pro elektronizaci nových procesů je třeba zajistit další finanční rezervu. Při velkém nárůstu elektronizovaných procesů musíme uvažovat o „Enterprise“ verzi Camundy a případné workshopy (viz podkapitoly 4.2.2, 4.2.4)

### 4.3 Analýza přínosů nasazení Camundy

V této kapitole se podíváme na určité přínosy, které přináší integrace Camundy do fakulty elektrotechnické. Tyto přínosy rozdělíme na dvě základní kategorie:

#### Přímé výhody

1. Šetření času.
2. Bezpečnost informace.

#### Nepřímé výhody

1. Spokojenost zaměstnanců.
2. Větší přizpůsobivost procesů.

#### 4.3.1 Přímé výhody

##### 4.3.1.1 Šetření času

Na základě analýzy současného stavu procesů popsané v kapitole 3.1.1, můžeme definovat přínosy elektronizace procesů na fakultě. Hlavní přínos elektronizace je šetření času na vyplnění a zpracování žádostí. Především to výrazně ušetří čas doktorandovi na zajištění všech podpisů a studijní referentce na zajištění formálních záležitostí. Aktuálně student doktorského studia musí sám



shánět všechny potřebné podpisy. Zpracování žádostí v procesech F12 a F15 v papírové podobě trvá 14 dnů. Díky elektronizaci tento čas lze výrazně zmenšit, jelikož většina úloh na procesech nevyžadují fyzickou přítomnost všech uživatelů v jednom místě. V nejlepším případě tyto procesy lze vyřešit během 1-3 dnů. V případě problémů v systému lze snadno dohledat v jaké fázi nastal problém a rychle ho opravit.

Díky elektronizaci procesů lze vyřešit problém archivace dokumentů, který nastává u procesů v papírové podobě. Zároveň lze výrazně zmenšit čas na vyhledávání dokumentů a zvýšit efektivitu tvorby statistik.

### 4.3.2 Bezpečnost informace

Sama o sobě míra bezpečnosti elektronizovaných procesů je mnohem vyšší než u papírové podoby. Vzhledem k tomu, že při práci s dokumenty v papírové podobě existuje hodně rizik jako například poškození dokumentů, ztráta, vyplnění chybných údajů, je doporučeno minimalizovat uchovávání dokumentů a dat v papírové podobě. Všechny tyto problémy lze téměř kompletně eliminovat díky elektronizovaným procesům. Dokumenty v elektronické podobě nelze fyzicky poškodit. Může dojít k poškození nebo ztrátě souboru dokumentu, ale v tomto případě lze vrátit dokument do původního stavu pomocí záloh na serveru. Pomocí implementace validací v elektronických formulářích lze minimalizovat počet chyb během vyplnění údajů ve formuláři, které mohou nastat během ručního vyplňování z nepozornosti.

### 4.3.3 Nepřímé výhody

#### 4.3.3.1 Spokojenost zaměstnanců

Jeden z nejdůležitějších výkonných faktorů organizace je spokojenost jejich zaměstnanců, v našem případě navíc i studentů. Příjemné pracovní prostředí motivuje uživatele a pomáhá se soustředit na opravdu důležité věci. Strukturované úlohy v elektronické podobě přináší jasnost v definici zadání, formu a podmínky splnění zadání, časové ohraničení. Paralelně manažeři mají přehled o tom, co se děje, v reálném čase.

#### 4.3.3.2 Větší přizpůsobivost procesů

Díky Camundě administrátor systému nebo jiná oprávněná osoba může měnit chod procesu v době jeho průběhu v reálném čase. Lze měnit implementaci, byznys logiku bez velkých zásahů do systému. Existuje možnost zastavení, stornování procesu. Na základě statistik (Camunda Heat Map v Enterprise verzi) definovat nejvíc problémové části procesu, úlohy nebo konkrétního uživatele. Takové možnosti pomohou FELu se přizpůsobit vnějším měnícím okolnostem a posunou řízení procesů na zcela novou úroveň.



---

# Závěr

Po seznámení čtenáře se základy procesního řízení a způsobem modelování procesů jsem ukázal využití těchto znalostí při modelování a následující elektronizaci reálně existujících procesů studijního oddělení FEL ČVUT. V praktické části byly vyřešeny následující úkoly:

1. **Analýza současného průběhu procesů F15 a F12** - současný stav procesů F12 a F15 na fakultě, podkapitola 3.1.1.
2. **Definice TO-BE stavu procesů F15 a F12** - na základě analýzy současného stavu vybraných procesů jsem navrhl potřebné změny pro elektronizaci a procesní diagram, podkapitola 3.1.2.
3. **Zprovoznění prostředí Camunda** - popsal jsem nutné kroky pro instalaci a spuštění Camundy včetně podpůrného softwaru, podkapitola 3.2.
4. **Elektronizace podnikového procesu** - provedl jsem kompletní elektronizaci procesů F12 a F15 pro studijní oddělení FEL ČVUT, včetně podrobného popisu procesů. Návod k elektronizaci procesů lze najít v podkapitole 3.3.
5. **Zhodnocení průběhu elektronizace procesů** - provedl jsem analýzu výstupů z praktické části včetně popisů problémů, které nastaly v průběhu elektronizace. Popsáno v podkapitole 4.1.
6. **Vyčíslení nákladů na pořízení nutného HW a SW** - spočítal jsem náklady potřebné k zprovoznění Camundy na studijním oddělení FEL a možnosti budoucího rozvoje. Popsáno v podkapitole 4.2.
7. **Analýza přínosů nasazení Camundy** - definoval jsem určité přínosy pro vybranou organizaci. Popsáno v podkapitole 4.3.

## ZÁVĚR

---

Cílem mé práce byla elektronizace procesů pro studijní oddělení FEL ČVUT a seznámení s platformou Camunda. Z mé strany vidím velký potenciál v použití platformy Camunda BPM. Na základě provedené práce bych doporučil zprovoznění Camundy nejen na fakultě elektrotechnické, ale i na celém ČVUT.

---

## Literatura

- [1] blog, B.: Process mapping for the uninitiated - part 3. 2013. Dostupné z: <http://bruce-levitan.blogspot.cz/2013/04/process-mapping-for-uninitiated-part-3.html>
- [2] Pegasystems: 2016 Gartner Magic Quadrant for Intelligent Business Process Management Suites. 2016. Dostupné z: <https://www.pega.com/gartner-ibpms-2016>
- [3] services, C.: Get started with Camunda. 2015. Dostupné z: <https://docs.camunda.org/get-started/>
- [4] centrum znalostního managementu: Zadávací dokumentace pro úpravy KOSu, 0 - Obecné informace k analytické zadávací dokumentaci. 2016. Dostupné z: <https://drive.google.com/file/d/0Bxs3penhtDH8a0FoRDcydWlvMmM/view>
- [5] Šmída, F.: *Zavádění a rozvoj procesního řízení ve firmě*. Grada Publishing, a.s., 2007, první vydání, ISBN 978-80-247-6380-4.
- [6] Řepa, V.: *Procesně řízená organizace*. Grada Publishing, a.s., 2012, první vydání, ISBN 978-80-247-4128-4.
- [7] Managementmania: Řízení procesů (Process Management). Leden 2016. Dostupné z: <https://managementmania.com/cs/rizeni-procesu>
- [8] Chang, J. F.: *Process Management Systems*. Taylor and Francis Group, 2006, první vydání, ISBN 0-8493-2310-X.
- [9] Michael Hammer, J. C.: *Reengineering - radikální proměna firmy, Manifest revoluce v podnikání*. Management Press, 1995, první vydání, ISBN 80-85603-73-X.
- [10] Hronza, R.: Procesní portál – od motivace až po současnost. Prosinec 2015. Dostupné z: <http://blog.czm-cvut.cz/procesni-portal>

- [11] Object Management Group, I.: Business Process Model and Notation (BPMN). 2011. Dostupné z: <http://www.omg.org/spec/BPMN/2.0/>
- [12] Journal, I. B. P. M.: BPM Voices: What is BPMN 2.0 and why does it matter? 2011. Dostupné z: [http://www.ibm.com/developerworks/websphere/bpmjournal/1112\\_col\\_jensen/1112\\_jensen.html](http://www.ibm.com/developerworks/websphere/bpmjournal/1112_col_jensen/1112_jensen.html)
- [13] Řepa, V.: *Podnikové procesy, procesní řízení a modelování*. Grada Publishing, a.s., 2007, druhé vydání, ISBN 978-80-247-2252-8.
- [14] Weske, M.: *Business Process Management*. Springer-Verlag Berlin Heidelberg, 2007, první vydání, ISBN 978-3-540-73521-2.
- [15] Gartner, I.: Gartner Magic Quadrant. 2016. Dostupné z: [http://www.gartner.com/technology/research/methodologies/research\\_mq.jsp](http://www.gartner.com/technology/research/methodologies/research_mq.jsp)
- [16] centrum znalostního managementu: Obecné požadavky na funkcionality KOSu. 2015. Dostupné z: <https://drive.google.com/file/d/0Bxs3penhtDH8TGd0bnY4SzYwUWM/view>
- [17] Ministerstvo Školství, M. a. T.: Výkonové ukazatele VŠ v ČR. 2016. Dostupné z: [http://dsia.uiv.cz/vystupy/vu\\_vs\\_f2.html](http://dsia.uiv.cz/vystupy/vu_vs_f2.html)
- [18] centrum znalostního managementu: Zadávací dokumentace pro úpravy KOSu, 1.1 - Formuláře. 2015. Dostupné z: <https://drive.google.com/file/d/0Bxs3penhtDH8TVNvQzFQQ3hSRnc/view>
- [19] Eclipse: Eclipse documentation - Current Release, Eclipse Neon. 2016. Dostupné z: <http://help.eclipse.org/neon/index.jsp>
- [20] Foundation, A. S.: Maven – Introduction. 2016. Dostupné z: <https://maven.apache.org/what-is-maven.html>
- [21] Český Statistický Úřad: ICT odborníci - Mzdy ICT odborníků v datech. 2016. Dostupné z: <https://www.czso.cz/csu/czso/ict-odbornici>

---

## Rozhraní studentForm.html

```
<form role="form" name="reapirForm">
<!-- Date popup -->
<script cam-script type="text/form-script">
  $scope.open = function($event) {
    $event.preventDefault();
    $event.stopPropagation();

    $scope.dateFieldOpened = true;
  };
</script>
<!-- Date popup -->
<script cam-script type="text/form-script">
  $scope.open2 = function($event) {
    $event.preventDefault();
    $event.stopPropagation();

    $scope.dateFieldOpened2 = true;
  };
</script>

<div class="form-group">
  <label for="firstName">Jmeno</label>
  <input cam-variable-name="firstName"
    cam-variable-type="String"
    id="firstName"
    class="form-control"
    type="text"
    required />

</div>

<div class="form-group">
  <label for="lastName">Prijmeni</label>
  <input cam-variable-name="lastName"
    cam-variable-type="String"
    id="lastName"
    class="form-control"
    type="text"
    required />

</div>

<div class="form-group">
  <label for="studentSubject">Obor</label>
  <select cam-variable-name="studentSubject"
    cam-variable-type="String"
    class="form-control">
    <option>Obor A</option>
    <option>Obor B</option>
    <option>Obor C</option>
    <option>Obor D</option>
  </select>
</div>
```

## A. ROZHRANÍ STUDENTFORM.HTML

---

```
<div class="form-group">
<label for="stopFrom">Preruseni studia OD</label>
<p class="input-group">
  <input type="text"
    cam-variable--name="stopFrom"
    cam-variable--type="Date"
    class="form-control"
    datepicker-popup="yyyy-MM-dd'T'HH:mm:ss"
    is-open="dateFieldOpened" />

  <span class="input-group-btn">
    <button type="button"
      class="btn btn-default"
      ng-click="open($event)">
      <i class="glyphicon glyphicon-calendar"></i>
    </button>
  </span>
</p>
</div>

<div class="form-group">
<label for="stopTill">Preruseni studia DO</label>
<p class="input-group">
  <input type="text"
    cam-variable--name="stopTill"
    cam-variable--type="Date"
    class="form-control"
    datepicker-popup="yyyy-MM-dd'T'HH:mm:ss"
    is-open="dateFieldOpened2" />

  <span class="input-group-btn">
    <button type="button"
      class="btn btn-default"
      ng-click="open2($event)">
      <i class="glyphicon glyphicon-calendar"></i>
    </button>
  </span>
</p>
</div>

<div class="form-group">
<label for="reason">Zduvodneni zadosti</label>
<div class="controls">
<textarea cam-variable--name="reason"
  cam-variable--type="String"
  rows="4"
  cols="80"
  required></textarea>
</div>
</div>

</form>
```



---

# Rozhraní TaskAssignmentListener.java

```
package org.camunda.bpm.proapp;

import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.SimpleEmail;
import org.camunda.bpm.engine.IdentityService;
import org.camunda.bpm.engine.delegate.DelegateTask;
import org.camunda.bpm.engine.delegate.TaskListener;
import org.camunda.bpm.engine.identity.User;
import org.camunda.bpm.engine.impl.context.Context;

public class TaskAssignmentListener implements TaskListener {
    //Set Mail Server Properties
    private static final String HOST = "smtp.gmail.com";
    private static final String USER = "notify.camunda@gmail.com";
    private static final String PWD = "camunda123";
    private final static Logger LOGGER = Logger.getLogger(TaskAssignmentListener.class.getName());

    public void notify(DelegateTask delegateTask) {
        String assignee = delegateTask.getAssignee();
        String taskId = delegateTask.getId();
        if (assignee != null) {
            // Get User Profile from User Management
            IdentityService identityService = Context.getProcessEngineConfiguration().getIdentityService();
            User user = identityService.createUserQuery().userId(assignee).singleResult();

            if (user != null) {
                // Get Email Address from User Profile
                String recipient = user.getEmail();

                if (recipient != null && !recipient.isEmpty()) {
                    Email email = new SimpleEmail();
                    email.setCharset("utf-8");
                    email.setHostName(HOST);
                    email.setAuthentication(USER, PWD);
                    //AK Gmail security fix
                    email.setTLS(true);
                    try {
                        email.setFrom("noreply@camunda.org");
                        email.setSubject("Task assigned: " + delegateTask.getName());
                        email.setMsg("Please complete: http://localhost:8080/camunda/app/tasklist/default/#/task/"
                            + taskId);
                        email.addTo(recipient);
                        email.send();
                        LOGGER.info("Task Assignment Email successfully sent to user'");
                    } catch (Exception e) {
                        LOGGER.error("Error sending email: " + e.getMessage());
                    }
                }
            }
        }
    }
}
```

## B. ROZHRANÍ TASKASSIGNMENTLISTENER.JAVA

---

```
+ assignee + " 'with address'" + recipient + " '.");
    } catch (Exception e) {
        LOGGER.log(Level.WARNING, "Could not send email to assignee", e);
    }

    } else {
        LOGGER.warning("Not sending email to user "
            + assignee + " ', user has no email address.");
    }

    } else {
        LOGGER.warning("Not sending email to user "
            + assignee + " ', user is not enrolled with identity service.");
    }
}
}
}
```

# Rozhraní CreatePdf.java

```
package org.camunda.bpm.proapp;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.edit.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import org.camunda.bpm.engine.delegate.DelegateExecution;
import org.camunda.bpm.engine.delegate.JavaDelegate;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class CreatePdf implements JavaDelegate{

    public void execute(DelegateExecution execution) throws Exception {
        //definition of process variables
        String firstName = (String) execution.getVariable("firstName");
        String lastName = (String) execution.getVariable("lastName");
        String studentSubject = (String) execution.getVariable("studentSubject");
        Date stopFrom = (Date) execution.getVariable("stopFrom");
        Date stopTill = (Date) execution.getVariable("stopTill");
        String reason = (String) execution.getVariable("reason");
        String mentorComment = (String) execution.getVariable("mentorComment");
        String HeadDepComment = (String) execution.getVariable("HeadDepComment");
        String chairmanComment = (String) execution.getVariable("chairmanComment");
        String mailNumber = (String) execution.getVariable("mailNumber");
        //approving variables
        String approveMentor = (String) execution.getVariable("approveMentor");
        String approveHeadDep = (String) execution.getVariable("approveHeadDep");
        String approveChairman = (String) execution.getVariable("approveChairman");
        //set variables
        execution.setVariable("approveMentor", approveMentor);
        execution.setVariable("approveHeadDep", approveHeadDep);
        execution.setVariable("approveChairman", approveChairman);
        execution.setVariable("firstName", firstName);
        execution.setVariable("lastName", lastName);
        execution.setVariable("studentSubject", studentSubject);
        execution.setVariable("stopFrom", stopFrom);
        execution.setVariable("stopTill", stopTill);
        execution.setVariable("reason", reason);
        execution.setVariable("mentorComment", mentorComment);
        execution.setVariable("HeadDepComment", HeadDepComment);
        execution.setVariable("chairmanComment", chairmanComment);
        execution.setVariable("mailNumber", mailNumber);
        //Change format for pdfapachebox to string
        DateFormat df = new SimpleDateFormat("MM/dd/yyyy_HH:mm:ss");
        String from = df.format(stopFrom);
        String to = df.format(stopTill);
        //Debug in console
    }
}
```

## C. ROZHRAŇÍ CREATEPDF.JAVA

---

```
System.out.println("approveMentor□is:□" + approveMentor);
System.out.println("approveHeadDep□is:□" + approveHeadDep);
System.out.println("approveChairman□is:□" + approveChairman);

System.out.println("Create□Simple□PDF□file□with□Text");
String fileName = System.getProperty("user.home")
    + "/Desktop/HOME_CAMUNDA/Decisions/rozhodnuti.pdf";
PDDocument doc = new PDDocument();
PDPage page = new PDPage();

doc.addPage(page);

PDPageContentStream content = new PDPageContentStream(doc, page);
//Header of form
if (approveMentor.equals("v2")) {
    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 26);
    content.moveTextPositionByAmount(220, 750);
    content.drawString("Denied□by□mentor");
    content.endText();
} else if (approveHeadDep.equals("v2")) {
    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 26);
    content.moveTextPositionByAmount(220, 750);
    content.drawString("Denied□by□head□of□department");
    content.endText();
} else if (approveChairman.equals("v2")) {
    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 26);
    content.moveTextPositionByAmount(220, 750);
    content.drawString("Denied□by□chairman");
    content.endText();
} else {
    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 26);
    content.moveTextPositionByAmount(220, 750);
    content.drawString("Application□is□approved");
    content.endText();
}

//Form body (constant)
    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 16);
    content.moveTextPositionByAmount(80, 700);
    content.drawString("Name:□" + firstName);
    content.endText();

    content.beginText();
    content.setFont(PDType1Font.HELVETICA, 16);
    content.moveTextPositionByAmount(80, 650);
    content.drawString("Surname:□" + lastName);
    content.endText();

    content.beginText();
    content.moveTextPositionByAmount(80, 600);
    content.drawString("Student□subject:□" + studentSubject);
    content.endText();

    content.beginText();
    content.moveTextPositionByAmount(80, 550);
    content.drawString("Study□interruption□from:□" + from);
    content.endText();

    content.beginText();
    content.moveTextPositionByAmount(80, 500);
    content.drawString("Study□interruption□to:□" + to);
    content.endText();

    content.beginText();
```

---

```

        content.moveTextPositionByAmount(80,450);
        content.drawString("Reason of interruption: " + reason);
        content.endText();

// Comments
if (approveMentor.equals("v2")) {
    content.beginText();
    content.moveTextPositionByAmount(80,400);
    content.drawString("Mentor comment: " + mentorComment);
    content.endText();
} else if (approveHeadDep.equals("v2")) {
    content.beginText();
    content.moveTextPositionByAmount(80,400);
    content.drawString("Head of Department comment: " + HeadDepComment);
    content.endText();
} else if (approveChairman.equals("v2")) {
    content.beginText();
    content.moveTextPositionByAmount(80,400);
    content.drawString("Chairman comment: " + chairmanComment);
    content.endText();
} else {
    content.beginText();
    content.moveTextPositionByAmount(80,400);
    content.drawString("Mentor comment: " + mentorComment);
    content.endText();

    content.beginText();
    content.moveTextPositionByAmount(80,350);
    content.drawString("Head of Department comment: " + HeadDepComment);
    content.endText();

    content.beginText();
    content.moveTextPositionByAmount(80,300);
    content.drawString("Chairman comment: " + chairmanComment);
    content.endText();
}

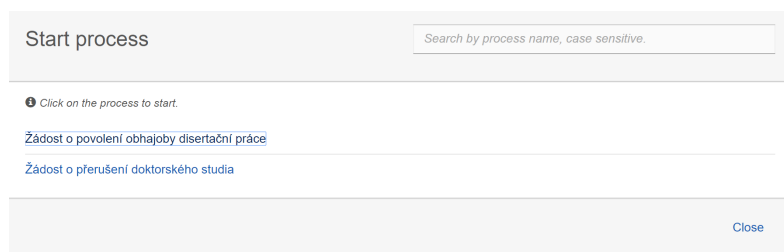
//Close file
content.close();
doc.save(fileName);
doc.close(); content.close();
doc.save(fileName);
doc.close();

System.out.println("your file created in: "
    + System.getProperty("user.dir"));
}
}

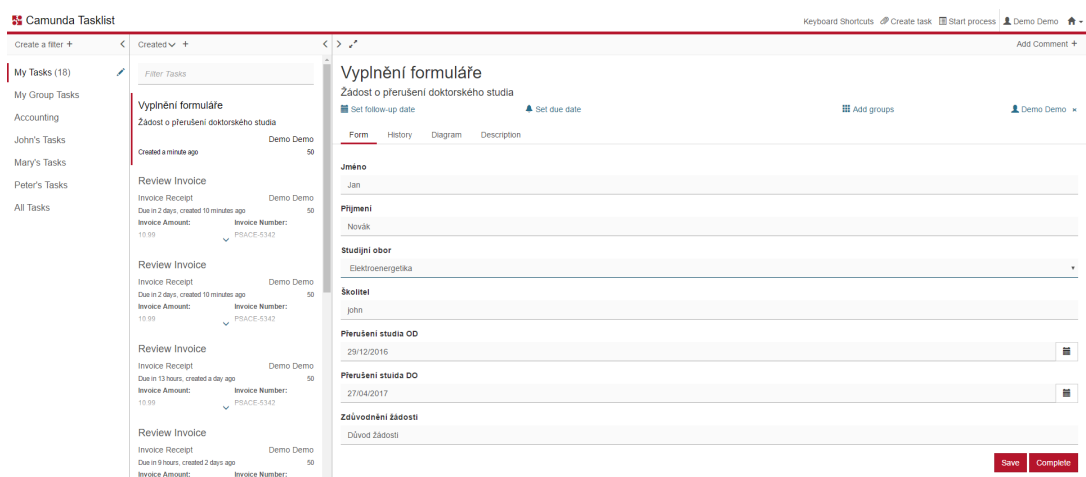
```



# Grafické rozhraní formulářů v procesu F15



Obrázek D.1: Výběr procesu



Obrázek D.2: Vyplnění formuláře

## D. GRAFICKÉ ROZHRANÍ FORMULÁŘŮ V PROCESU F15

Created + < > ↗ Add Comment +

My Tasks (1) Filter Tasks

My Group Tasks

Posouzení školitele John Doe 50  
Created a minute ago

Posouzení školitele

Set follow-up date Set due date Add groups John Doe +

Form History Diagram Description

Jméno studenta  
Jan

Příjmení studenta  
Novák

Studijní obor  
Elektroenergetika

Školitel  
John

Přerušení studia OD  
29/12/2016

Přerušení studia DO  
27/04/2017

Zdůvodnění žádosti  
Důvod žádosti

Poznámka školitele  
Testováta poznámka

Vyjádření školitele  
Schváleno

Save Complete

Obrázek D.3: Posouzení školitele

Created + < > ↗ Add Comment +

Filter Tasks

Posouzení vedoucího katedry Vedoucí katedry A 50  
Created a few seconds ago

Posouzení vedoucího katedry

Přerušení studia DO  
2017-01-26T00:00:00

Zdůvodnění žádosti  
Test Duvod

Komentář školitele  
Mentor comment

Komentář vedoucího katedry  
headDep comment

Rozhodnutí vedoucího katedry  
Vráčeno

Save Complete

Obrázek D.4: Posouzení vedoucího katedry



Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Posouzení předsedy oborové rady**  
 Předseda oborové rady A  
 Created a few seconds ago 50

**Komentář školitele**  
 Mentor comment

**Komentář vedoucího katedry**  
 headDep comment

**Komentář předsedy oborové rady**  
 Chairman comment

**Rozhodnutí vedoucího katedry**  
 Schváleno

Save Complete

Obrázek D.5: Posouzení předsedy oborové rady

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Vložení čísla jednacího**  
 Studijní Referentka A  
 Created a few seconds ago 50

**Komentář školitele**  
 Mentor comment

**Komentář vedoucího katedry**  
 headDep comment

**Komentář předsedy oborové rady**  
 Chairman comment

**Vložte číslo jednací**  
 AVA2000

Save Complete

Obrázek D.6: Vložení čísla jednacího

## D. GRAFICKÉ ROZHRANÍ FORMULÁŘŮ V PROCESU F15

The screenshot shows a Camunda BPM form interface. The title is "Tisk dokumentu, zajištění podpisu a odeslání". The form is currently in the "Form" view, with tabs for "Form", "History", "Diagram", and "Description". The form content includes a checkbox "Žádost vygenerována správně?" which is checked. Below this are several input fields: "Číslo jednací" (AVA2000), "Jméno" (Bob), "Příjmení" (Silent), "Obor" (Obor A), "Přerušení studia OD" (2017-01-03T00:00:00), and "Přerušení studia DO". The left sidebar shows a task list with the same title, user "Studijní Referentka A", and a status of "Created a few seconds ago". The bottom right corner indicates "Powered by camunda BPM / v7.6.0".

Obrázek D.7: Tisk dokumentu, zajištění podpisu a odeslání

The screenshot shows a Camunda BPM form interface. The title is "Úprava formuláře". The form is currently in the "Form" view, with tabs for "Form", "History", "Diagram", and "Description". The form content includes a checkbox "Žádost o přerušení doktorského studia" which is checked. Below this are several input fields: "Komentář školitele" (Mentor comment), "Komentář vedoucího katedry", "Komentář předsedy oborové rady", "Jméno" (Bob), "Příjmení" (Silent), and "Obor". The left sidebar shows a task list with the same title, user "Student A", and a status of "Created a minute ago".

Obrázek D.8: Úprava formuláře

# Grafické rozhraní formulářů v procesu F12

The screenshot shows a web interface for filling out a form. On the left, there is a sidebar with a 'Filter Tasks' button and a task titled 'Vyplnění formuláře' (Form filling) with a sub-title 'Žádost o povolení obhajoby disertační práce' (Request for permission to defend a thesis). The task is assigned to 'Student A' and was created 'a minute ago' with a progress indicator of '50'. The main form area contains several fields: 'Datum vykonání státní doktorské zkoušky' (Date of state doctoral exam), 'Název disertační práce - česky' (Thesis title - Czech), 'Název disertační práce - anglicky' (Thesis title - English), 'Disertační práce ve formátu PDF' (Thesis in PDF format) with a file selection button 'Выберите файл' (Select file) and the text 'Файл не выбран' (File not selected), and 'Komentář předsedy oborové rady' (Comment of the department council chair). At the bottom right, there are 'Save' and 'Complete' buttons.

Obrázek E.1: Vyplnění formuláře

## E. GRAFICKÉ ROZHHRANÍ FORMULÁŘŮ V PROCESU F12

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Posouzení školitele**

Created a few seconds ago Školitel A 50

Název disertační práce - anglicky  
thesis

Disertační práce ve formátu PDF [loanApprovalProcess.bpmn](#)

Posudek školitele ve formátu PDF  
 Файл не выбран

Komentář školitele

Rozhodnutí školitele  
Zamítnuto

Obrázek E.2: Posouzení školitele

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Posouzení předsedy oborové rady**

Created a few seconds ago Předseda oborové rady A 50

Název disertační práce - anglicky  
thesis

Disertační práce ve formátu PDF [loanApprovalProcess.bpmn](#)

Posudek školitele ve formátu PDF [loanApprovalProcess.bpmn](#)

Komentář předsedy oborové rady  
comment předsedy ORO

Rozhodnutí předsedy oborové rady  
Schváleno

Obrázek E.3: Posouzení předsedy oborové rady

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Posouzení předsedy oborové rady**  
 Předseda oborové rady A  
 Created a few seconds ago 50

Název disertační práce - anglicky  
 thesis

Disertační práce ve formátu PDF [loanApprovalProcess.bpmn](#)

Posudek školitele ve formátu PDF [loanApprovalProcess.bpmn](#)

Komentář předsedy oborové rady  
 comment předsedy ORO

Rozhodnutí předsedy oborové rady  
 Schváleno

Save Complete

Obrázek E.4: Posouzení předsedy oborové rady

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Doplnění dokumentů**  
 Žádost o povolení obhajoby disertační práce  
 Student A  
 Created a minute ago 50

Posudek školitele ve formátu PDF [loanApprovalProcess.bpmn](#)

Komentář předsedy oborové rady  
 comment předsedy ORO

Komentář studijní referentky

Vlastní životopis  
 Выберите файл prace.pdf

Seznam vlastních publikací  
 Выберите файл prace.pdf

Seznam předmětů absolvovaných mimo FEL  
 Выберите файл prace.pdf

Teze  
 Выберите файл prace.pdf

Save Complete

Obrázek E.5: Doplnění dokumentů studentem

## E. GRAFICKÉ ROZHRAŇÍ FORMULÁŘŮ V PROCESU F12

The screenshot shows a web interface for 'Formální kontrola žádosti' (Formal request control). The left sidebar contains a 'Filter Tasks' dropdown and the task title 'Formální kontrola žádosti' with the role 'Studijní Referentka A' and a 'Created a few seconds ago' timestamp. The main content area includes several sections: 'Posudek školitele ve formátu PDF' (loanApprovalProcess.bpmn), 'Komentář předsedy oborové rady' (comment předsedy ORO), 'Vlastní životopis' (prace.pdf), 'Seznam vlastních publikací' (prace.pdf), 'Seznam předmětů absolvovaných mimo FEL' (prace.pdf), and 'Teze' (prace.pdf). Below these is a 'Komentář studijní referentky' (comment stud ref) text area and a checked checkbox 'Žádost obsahuje všechny potřebné podklady?'. At the bottom right are 'Save' and 'Complete' buttons.

Obrázek E.6: Formální kontrola žádosti

The screenshot shows a web interface for 'Posouzení oponenta A' (Evaluation of reviewer A). The left sidebar contains a 'Filter Tasks' dropdown and the task title 'Posouzení oponenta A' with the role 'Školitel A' and a 'Created a few seconds ago' timestamp. The main content area includes: 'Název disertační práce - anglicky' (thesis) text area, 'Disertační práce ve formátu PDF' (loanApprovalProcess.bpmn), 'Posudek oponenta ve formátu PDF' (Выберите файл prace.pdf) file upload, 'Komentář oponenta' (comment oponenta A) text area, and 'Rozhodnutí oponenta' (Schváleno) dropdown menu. At the bottom right are 'Save' and 'Complete' buttons.

Obrázek E.7: Posouzení oponenta A

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Posouzení oponenta B**  
 Předseda oborové rady A  
 Created a minute ago 50

Název disertační práce - anglicky  
 thesis

Disertační práce ve formátu PDF [loanApprovalProcess.bpmn](#)

Posudek oponenta ve formátu PDF  
 Выберите файл | prace.pdf

Komentář oponenta  
 komment oponenta B

Rozhodnutí oponenta  
 Schváleno

Save Complete

Obrázek E.8: Posouzení oponenta B

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Seznámení s posudky**  
 Žádost o povolení obhajoby disertační práce  
 Student A  
 Created a few seconds ago 50

Disertační práce ve formátu PDF [loanApprovalProcess.bpmn](#)

Komentář oponenta A  
 comment oponenta A

Posudek oponenta A [prace.pdf](#)

Komentář oponenta B  
 komment oponenta B

Posudek oponenta A [prace.pdf](#)

Rozhodnutí studenta  
 Zrušit žádost

Save Complete

Obrázek E.9: Seznámení s posudky

## E. GRAFICKÉ ROZHRAŇÍ FORMULÁŘŮ V PROCESU F12

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Vložení čísla jednacího**  
Studijní Referentka A  
Created a few seconds ago 50

**Komentář školitele**  
Mentor comment

**Komentář vedoucího katedry**  
headDep comment

**Komentář předsedy oborové rady**  
Chairman comment

**Vložte číslo jednací**  
AVA2000

Save Complete

Obrázek E.10: Vložení čísla jednacího

Created ▾ + < > ↗ Add Comment +

Filter Tasks

**Tisk dokumentu, zajištění podpisu a odeslání**  
Studijní Referentka A  
Created a few seconds ago 50

**Tisk dokumentu, zajištění podpisu a odeslání**  
Set follow-up date Set due date Add groups Studijní Referent... ×

Form History Diagram Description

Žádost vygenerována správně?

**Číslo jednací**  
AVA2000

**Jméno**  
Bob

**Příjmení**  
Silent

**Obor**  
Obor A

**Přerušení studia OD**  
2017-01-03T00:00:00

**Přerušení studia DO**

Powered by camunda BPM / v7.6.0

Obrázek E.11: Tisk dokumentu, zajištění podpisu a odeslání



## Seznam použitých zkratek

**HW** Hardware

**SW** Software

**CMM** Capability Maturity Model

**KPI** Key Performance Indicators

**BPM** Business Process Management

**ERP** Enterprise resource planning

**UML** Unified Modeling Language

**BPMN** Business Process Management Notation

**API** Application Programming Interface

**IDE** Integrated Development Environment

**PDF** Portable Document Format

**HTML** HyperText Markup Language

**DPH** Daň z přidané hodnoty

**ICT** Information and Communication Technologies

**SLA** Service Level Agreement

**XML** Extensible markup language



---

## Obsah přiloženého CD

	readme.txt	.....	stručný popis obsahu CD a návod na spuštění
	HOME_CAMUNDA.zip	.....	soubor se spustitelnou formou implementace
	src		
		impl	..... zdrojové kódy implementace
		thesis	..... zdrojová forma práce ve formátu $\text{\LaTeX}$
	text	.....	text práce
		thesis.pdf	..... text práce ve formátu PDF