

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Aplikace pro modelování a vizualizaci podnikových procesů

Plzeň, 2019

Petr Šrámek

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 27. 6. 2019

Petr Šrámek

Abstract

Application for business processes modeling and visualization

This thesis aims at the design and realization of an app that simulates and visualises company processes. This app will be used as part of the company's analysis, before the implementation of the enterprise resource planning system FLORES. The purpose of the final app is to make the company processes clearer and simpler, to secure the control of the consistency and wholeness of data inputs, to enable the visualisation of defined data and further work with it, including utilization of this data for the automatic configuration of parts of the FLORES system. The simplification of company processes modeling is achieved (besides a clear user interface and the automatic depiction of visualisation) also by enabling the work with data that are already saved in the FLORES system.

Abstrakt

Aplikace pro modelování a vizualizaci podnikových procesů

Cílem této práce je návrh a vytvoření aplikace, sloužící pro modelování a vizualizaci podnikových procesů, která bude používána v rámci analýzy podniku před implementací informačního systému FLORES. Smyslem výsledné aplikace je zpřehlednění a zjednodušení popisu podnikových procesů, zajištění kontroly konzistence a úplnosti zadávaných dat a umožnění s definovanými daty dále pracovat, vizualizovat je a využít daná data pro automatickou konfiguraci částí systému FLORES. Zjednodušení modelování podnikových procesů je kromě přehledného uživatelského rozhraní a automatického vykreslování vizualizace docíleno umožněním práce s již existujícími daty, uloženými v systému FLORES.

Obsah

1. Úvod.....	1
2. Podnikové procesy.....	2
2.1. Modelování a vizualizace podnikových procesů.....	4
2.1.1. BPMN	5
2.1.2. UML - Diagram aktivit	11
2.1.3. EPC	15
2.1.4. Shrnutí.....	18
2.1.5. Návrh notace pro vizualizaci podnikových procesů	22
2.2. Existující nástroje pro vizualizaci podnikových procesů	23
3. Stávající způsob popisu podnikových procesů	25
4. Specifikace požadavků uživatelů aplikace	29
4.1. Přístup k aplikaci	29
4.2. Modelování podnikových procesů a definování souvisejících dat.....	29
4.3. Vizualizace	30
4.4. Kontrola konzistence.....	30
4.5. Konfigurace systému FLORES.....	31
4.6. Technologie	31
5. Definování vhodné funkcionality aplikace	32
6. Implementace aplikace.....	33
6.1. Použité technologie.....	33
6.1.1. Klientská část	33
6.1.2. Serverová část a Web API	35
6.2. Návrh architektury	36
6.2.1. PHP frameworky	38
6.3. Struktura aplikace	39
6.4. Struktura pro ukládání dat	40
6.5. Výběr knihovny pro vizualizaci podnikových procesů	41
6.5.1. mxGraph	41
6.5.2. GoJS.....	41
6.5.3. yWorks	42
6.5.4. Diagramo.....	43

6.5.5.	Rappid	43
6.5.6.	Shrnutí.....	44
6.6.	Realizace.....	45
6.6.1.	Uživatelské rozhraní aplikace	45
6.6.2.	Komunikace aplikace s ERP systémem FLORES	48
6.6.3.	Přihlášení uživatele	49
6.6.4.	Převzetí dat z aplikace Interview a správa tagů.....	49
6.6.5.	Správa podnikových procesů a jejich kroků.....	51
6.6.6.	Vizualizace podnikových procesů	55
7.	Testování vytvořené aplikace	61
7.1.	Testovacích scénáře	61
7.2.	Automatické testy	62
8.	Závěr	63
	Seznam zkratk	64
	Použitá literatura	66
	Přílohy	68
A	Uživatelská příručka	68
B	Obsah přiloženého média	70

1. Úvod

Před implementací ERP (*Enterprise Resource Planning*) systému FLORES v konkrétním podniku, je nejdříve nutná detailní analýza jednotlivých podnikových procesů, zpracovávaná konzultanty během analytických schůzek se zainteresovanými uživateli z daného podniku. Cílem analýzy je komplexní pochopení podnikových procesů a sepsání podkladů pro následnou konfiguraci a nastavení ERP systému FLORES pro daný podnik. Jednotlivé podnikové procesy je podstatné definovat v kontextu vlastností a možností systému FLORES a je tedy chtěné popis jednotlivých procesů přímo navázat na funkcionalitu tohoto systému.

Pro správnou implementaci a nasazení ERP systému v podniku je nezbytné, aby definice podnikových procesů byla konzistentní, úplná a především přehledná a srozumitelná nejen pro konzultanty, kteří popis podnikových procesů vytvářejí, ale zároveň pro všechny zainteresované uživatele z daného podniku, kteří by u jednotlivých podnikových procesů měli potvrdit, že byly konzultantem pochopeny a zapsány korektně a odpovídají skutečnosti a chtěnému výsledku.

Cílem této práce je návrh a vytvoření aplikace, sloužící pro modelování a vizualizaci podnikových procesů, která bude používána v rámci analýzy podniku, před implementací ERP systému FLORES.

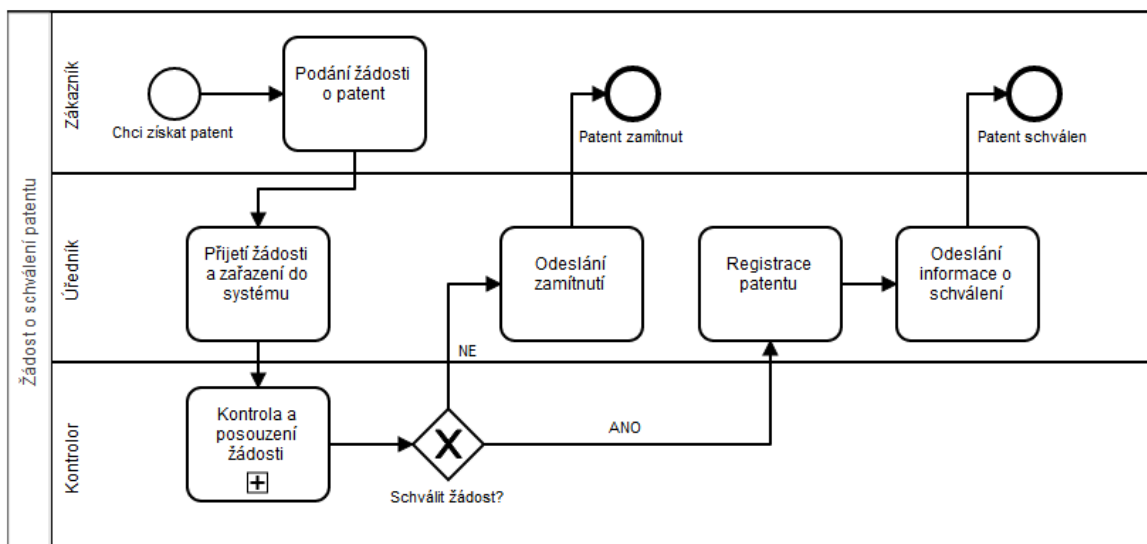
Výsledná aplikace nahradí současný nevyhovující způsob popisu podnikových procesů v obyčejném textovém editoru, který je velmi nepřehledný a neumožňuje automatické zpracování definovaných dat, jejich vizualizaci a kontrolu jejich úplnosti a konzistence. Aplikace zároveň usnadní vytváření popisu podnikových procesů díky propojení s již existujícími daty, uloženými v systému FLORES. Kromě samotné definice požadavků, bude část definovaných dat zároveň automaticky použita pro přímou konfiguraci systému FLORES.

Úvodní část práce se zabývá analýzou možností pro vizualizaci podnikových procesů, existujících nástrojů v této oblasti a stávajícího způsobu popisu podnikových procesů konzultanty společnosti FLORES Software. Znalosti, získané v úvodní části práce, jsou následně aplikovány při návrhu a vývoji výsledné aplikace.

2. Podnikové procesy

V této kapitole jsou definovány základní pojmy, týkající se podnikových procesů, popisu modelování podnikových procesů a jejich vizualizace. Po přečtení by měl čtenář vědět, co je to podnikový proces, jak jej lze modelovat a jaký druh grafické notace pro vizualizaci podnikových procesů je vhodné použít pro který případ užití.

Podnikový proces (business process) lze definovat jako souhrn samostatných aktivit/kroků, vykonávaných v jasně definovaném pořadí a vedoucích k dosažení konkrétního podnikového cíle. Jednoduchým příkladem podnikového procesu může být například proces „Posouzení nové žádosti o schválení patentu“, vizualizovaný na obrázku 1. Tento příklad slouží pouze pro základní představu, jak může model podnikového procesu vypadat. Uvedený proces se sestává z celkem šesti aktivit a tří zainteresovaných uživatelů. Jednotlivé prvky notace BPMN (*Business Process Modeling Notation*), která byla pro vizualizaci níže uvedeného modelu podnikového procesu použita, jsou detailně popsány v kapitole 2.1.1.



Obrázek 1 - Příklad podnikového procesu "Žádost o schválení patentu" modelovaného pomocí notace BPMN

Proces „Posouzení nové žádosti o schválení patentu“ začíná v levém horním rohu podáním žádosti o schválení patentu. Vstupem procesu je tedy dokument s žádostí o schválení patentu, podaný uživatelem „Zákazník“. Žádost se následně přesune k uživateli „Úředník“, který je zodpovědný za správu žádostí a komunikaci se zákazníky. Přijatá žádost je dále předána uživateli „Kontrolor“, který má na starost kontrolu a posouzení jednotlivých žádostí. Aktivitu kontroly a posouzení žádosti by bylo možné rozložit detailněji na více kroků, ze kterých se skládá. Pro základní vizualizaci procesu ale není nutné tento krok rozepisovat detailněji, díky čemuž je docíleno lepší přehlednosti modelu, jelikož je vizualizovaných méně aktivit. Aktivita kontroly je místo rozepsání jednotlivých kroků pouze označena symbolem +, který značí, že se jedná o

podproces, skládající se z více samostatných kroků. Na základě kontroly a posouzení žádosti je následně daná žádost buďto schválena a patent registrován nebo zamítnuta. Celý proces končí informováním zákazníka o výsledku schvalovacího procesu.

Základním prvkem podnikového procesu je aktivita. Jednotlivé aktivity na sebe vzájemně navazují, reagují na vnější i vnitřní podněty (události) a přeměňují prostřednictvím konkrétních uživatelů a nástrojů vstupy na výstupy. Kromě jednotlivých aktivit obsahují podnikové procesy také informace o vzájemné návaznosti (toku) jednotlivých aktivit, o zainteresovaných uživateli, informace o předávaných datech a další. Každý podnikový proces vždy musí mít na počátku nějaký vstup, který je v rámci daného procesu zpracován na výsledný výstup, odlišný od vstupu.

Zainteresovaní uživatelé (stakeholders) jsou pro podnikové procesy velmi důležitou součástí, jelikož právě oni mají v daném podniku zodpovědnost za vykonávání jednotlivých aktivit v rámci podnikových procesů a zajišťují jejich funkčnost. Uživatele lze dělit do dvou základních skupin [1]:

- **Přímo zainteresovaní uživatelé** - zajišťují přímé vykonávání jednotlivých aktivit, jako například účetní zajišťuje vydávání faktur nebo kuchař zajišťuje přípravu jídla.
- **Nepřímo zainteresovaní uživatelé** - ovlivňují, nebo jsou ovlivněni podnikovými procesy, ale nejsou do nich přímo zapojeni. Jedná se například o management nebo o dodavatele materiálů.

Podnikové procesy lze rozdělit z pohledu společnosti a zákazníka do tří kategorií [2, 3]:

- **Hlavní procesy** - Z pohledu společnosti se jedná o klíčové procesy, které přinášejí společnosti zisk. Poskytují zákazníkovi přidanou hodnotu a jedná se o jediné procesy, se kterými přichází zákazník do přímého kontaktu. Například se jedná o poskytnutí konkrétní služby zákazníkovi, či předání výsledného produktu.
- **Podpůrné procesy** - Jedná se o procesy, které zajišťují a udržují prostředí pro hladký průběh hlavních procesů. Na rozdíl od hlavních procesů nepřinášejí zákazníkovi přímo přidanou hodnotu a negenerují společnosti přímý zisk, ale jsou velmi důležité, jelikož ovlivňují výslednou kvalitu a efektivitu hlavních procesů a hlavní procesy by bez nich nemohly fungovat. Příkladem mohou být například logistika, správa zásobování, správa lidských zdrojů a další.
- **Řídící procesy** - Mají na starost koordinaci, monitorování a řízení hlavních a podpůrných procesů. Zajišťují tedy, aby jednotlivé podnikové procesy splňovaly své cíle a aby fungování celé organizace bylo efektivní a produktivní.

Ideální podnikový proces by měl splňovat následující 4 pravidla [1]:

- **Konečnost** - Má jasně definovaný počáteční a koncový bod, přičemž vždy existuje konečná posloupnost aktivit, jak se dostat z počátečního bodu do koncového.
- **Opakovatelnost** - Má neomezený počet iterací, kolikrát může proběhnout. Nemělo by se tedy stát, že by bylo možné proces v rámci životního cyklu podniku vykonat pouze nějaký omezený počet iterací a pak by jej již nešlo vykonávat.
- **Vytváří výsledný výstup** - Každá z aktivit v rámci podnikového procesu by měla mít jako výsledek své činnosti výstup, odlišný od svého vstupu. Pokud některá z aktivit tomuto požadavku neodpovídá a nemá na svém výstupu žádnou přidanou hodnotu oproti vstupu, je v rámci procesu zbytečná, protože v rámci procesu nevedla k žádné nové přidané hodnotě.
- **Flexibilita** - Měl by umožňovat snadnou implementaci případných změn a vylepšení, které ovlivní jen daný proces.

2.1. Modelování a vizualizace podnikových procesů

Modelování podnikových procesů je analytická činnost, sloužící pro vytvoření formálního popisu aktuálních a případně i nově plánovaných pracovních postupů (workflow), díky čemuž jim lze lépe porozumět, analyzovat je a optimalizovat, což ve výsledku zajistí efektivnější fungování podniku [4, 5]. Velkou výhodou je, že díky souhrnnému modelování veškerých podnikových procesů je jejich popis pro celou společnost standardizovaný a zamezí se například situaci, kdy má každé oddělení pro své potřeby své vlastní popisy procesů, které se napříč různými odděleními vzájemně liší.

Výsledkem modelování procesů je většinou grafický výstup - vizualizaci podnikových procesů, která poskytuje mnohem lepší srozumitelnost a přehlednost, než kdyby se jednalo pouze o text. Vizualizace obsahuje grafické znázornění jednotlivých procesů, jejich aktivit, zdrojů, zainteresovaných uživatelů a dalších informací, které se řídí podle pravidel zvolené notace. Přehlednost a srozumitelnost vizualizace tedy velmi záleží na volbě použité grafické notace. Existuje celá řada modelovacích notací (jazyků), které lze pro vizualizaci podnikových procesů využít, počínaje obecnými, jako například UML (*Unified Modeling Language*), až po specializované, vytvořené konkrétně pro tento účel, jako například BPMN a EPC (*Event-driven Process Chain*).

Výsledný model by měl být vytvářený z několika pohledů zároveň:

- **funkční** - obsahuje informace, jaké aktivity jsou v rámci procesu vykonávány a jaké zdroje jsou pro ně relevantní

- **dynamický** - jak na sebe jednotlivé aktivity (potažmo procesy) navazují a jak mezi sebou spolupracují
- **organizační** - kdo a případně kde danou aktivitu vykonává
- **informační** - jaké informace (data) jsou vytvářené či zpracovávány v jaké části procesu

2.1.1. BPMN

BPMN je modelovací notace, která je velmi rozšířeně využívána pro vizualizaci podnikových procesů [6, 7]. Tato notace byla vyvinutá organizací Business Process Management Initiative a jejím hlavním cílem bylo vytvoření notace, která bude dostatečně formální a jasná, ale zároveň lehce čitelná a srozumitelná pro všechny uživatele, počínaje podnikovými analytiky, kteří vytvářejí návrhy jednotlivých podnikových procesů, přes vývojáře implementující technologická řešení pro podporu fungování navržených procesů až po management, který má na starost monitorování a řízení těchto procesů. Od roku 2005 má údržbu BPMN na starost konsorcium Object Management Group (OMG), které roku 2011 vydalo novou, v současné době (2017) nejnovější verzi BPMN 2.0.

BPMN vychází z technik používaných u vývojových diagramů, přizpůsobených pro vytváření grafických modelů podnikových procesů a umožňující vizualizaci návaznosti a pořadí vykonávání jednotlivých aktivit. Sestává se ze sady grafických prvků, které umožňují snadný vývoj modelů podnikových procesů. Jednotlivé prvky byly zvolené tak, aby byly jednoznačně na první pohled vzájemně rozlišitelné a zároveň aby byly použité takové tvary prvků, které jsou běžně používané a zažité pro většinu modelovacích jazyků. Jednotlivé aktivity tedy například značí obdélníky se zaoblenými rohy, rozhodovací podmínky kosočtverce atd. Při výběru jednotlivých prvků notace byl hlavní důraz kladen na jednoduchost, ale zároveň muselo být možné jejich kombinací obsáhnout všechny myslitelné případy, které mohou v rámci modelování podnikových procesů nastat. Splnění obou zdánlivě rozporujících se požadavků zároveň bylo docíleno rozdělením grafických prvků notace BPMN 2.0 do pěti základních kategorií. To umožnilo výsledné použití malé skupiny kategorií, jejíž prvky budou mít standardní tvar a bude jim každý rozumět. Zároveň ale jednotlivé kategorie bude možné rozšířit o nové varianty standardních prvků, což umožní pokrytí všech požadavků, které mohou při modelování podnikových procesů nastat. Vzhledem k tomu, že nové prvky budou vycházet ze základních, nedojde k dramatické změně výsledné podoby celého modelu a vyzná se v něm tedy i člověk, který nezná přesný význam nového prvku.

Základními kategoriemi jednotlivých prvků jsou:

- Plovoucí prvky (Flow elements)
- Spojovací objekty (Connecting objects)
- Bazénové dráhy (Swimlanes)
- Artefakty (Artifacts)
- Data

Plovoucí prvky

Vzájemným propojením prvků z této kategorie vzniká vizualizace podnikových pracovních postupů. Představují tedy stěžejní stavební kameny pro vizualizaci modelovaného procesu. Tato kategorie obsahuje tři základní prvky:

- Aktivity (Activities)
- Události (Events)
- Rozhodovací brány (Gateways)

Aktivity reprezentují jednotlivé kroky/úkoly, které jsou v rámci daného procesu vykonávány. Jsou vykreslovány jako obdélník se zaoblenými rohy, který obsahuje uvnitř text popisující činnost dané aktivity, viz obrázek 2. Aktivity lze rozdělit na úkoly (task) a podprocesy (sub-process). Úkoly se využívají v případech, kdy se jedná o atomickou aktivitu - tedy o aktivitu, kterou již nelze dále rozdělit na posloupnost aktivit. Podprocesy se používají v komplexnějších procesech, kdy daný podproces sám o sobě vykonává určitý menší úkol, který lze detailněji rozložit na samostatný proces, ale jeho detailní rozložení není v konkrétním modelu potřeba znázorňovat. Daný podproces by ale měl být samozřejmě poté někde v rámci modelu podnikových procesů detailněji rozložený jako samostatný proces. Podproces se od úkolu odlišuje čtvercem s ikonou + ve spodní části prvku, viz obrázek 2 vpravo.

Při rozhodování, jestli bude při modelování procesu použitý prvek podproces, nebo přímo jeho jednotlivé kroky je nutné brát v potaz, komu má být výsledný model určený a jak detailně jej potřebuje daný uživatel mít vizualizovaný. Pokud je určený pro zákazníka, bude nejspíše stačit znázornit například jen obecný podproces „zpracování platby“ a nepotřebuje mít detailně rozepsané, jak zpracování platby konkrétně probíhá. Na druhou stranu pro daného vývojáře implementujícího daný systém je průběh zpracování velmi důležitý, takže si jej může zobrazit případně rozepsaný detailněji. Díky tomu je výsledný model mnohem přehlednější a neobsahuje přemrštěné množství informací.



Obrázek 2 - BPMN prvky Aktivita

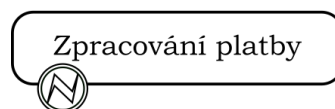
Události jsou reprezentované prvkem ve tvaru kruhu a jedná se události, které nastanou někdy v průběhu životního cyklu procesu a mají na něj určitý dopad. Události mohou být buďto externí nebo interní a mají obvykle nějaký spouštěč a důsledek. Kruhové prvky jsou v základní podobě prázdné, ale mohou uvnitř obsahovat ikonu, charakterizující konkrétněji typ dané události.

Dělí se na tři základní typy podle toho, v jaké části procesu nastanou: počáteční (na začátku procesu), prostřední (v průběhu procesu) a koncová (při dokončení procesu), viz obrázek 3.



Obrázek 3 - BPMN prvky Událost

Každý proces má zpravidla počáteční a koncovou událost. Navíc se může objevit prostředí událost, která slouží pro řízení procesu založeného na spuštění určité události v jeho průběhu. Prostřední událost je buďto znázorněná jako součást konkrétní aktivity, pokud může nastat v průběhu vykonávání dané aktivity, nebo samostatně v průběhu pracovního postupu. Příklad připnutí prostřední události k aktivitě je znázorněný na obrázku 4. Událost by u aktivity „Zpracování platby“ kontrolovala, jestli zpracování bylo úspěšné, případně by obsloužila výjimku, pokud by při zpracování platby nastal problém, například nedostatek finančních prostředků na účtu.



Obrázek 4 - Prostřední událost ověření zůstatku připnutá k aktivitě

Rozhodovací brány slouží pro řízení a směrování pracovního postupu. V notaci jsou značené kosočtvercem, který může být doplněn textem s rozhodující otázkou, viz obrázek 5. Stejně jako u událostí může být uvnitř kosočtverce navíc ikona, definující o jaký typ větvení se jedná. Rozvětvení pracovního postupu může být například exklusivní, kdy postup pokračuje jen jednou větví nebo paralelní, kdy postup pokračuje všemi větvemi zároveň.



Rozhodovací otázka

Obrázek 5 - BPMN prvek Rozhodovací brána

Bazénové dráhy

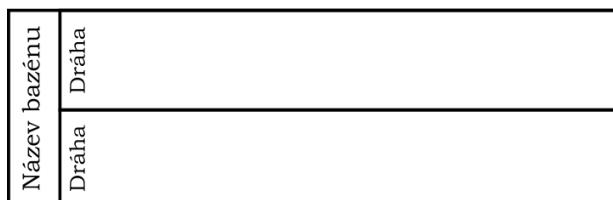
Prvky ze skupiny bazénových drah jsou využívány jako kontejnery pro vizuální oddělení jednotlivých aktivit procesu do různých kategorií. Do této kategorie se řadí prvky bazén a bazénové dráhy.

Bazén reprezentuje konkrétního účastníka v procesu (například odpovědného uživatele nebo podnikové oddělení), případně slouží pro ohraničení všech prvků patřících k jednomu procesu. Grafická reprezentace bazénu je znázorněná na obrázku 6.



Obrázek 6 - BPMN prvek bazén

Bazénové dráhy umožňují rozdělení bazénu na části, díky čemuž lze bazén horizontálně či vertikálně rozčlenit, viz obrázek 7. Slouží k organizaci prvků v bazénu do určitých kategorií. Pokud by byl bazén například rozdělený na podniková oddělení, pak by jednotlivé dráhy reprezentovaly konkrétní oddělení, například Účetní oddělení, Logistika a sklad, Marketingové oddělení atd.

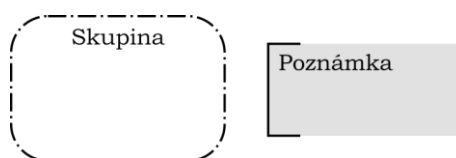


Obrázek 7 - BPMN prvek Bazén s dráhami

Artefakty poskytují doplňkové informace o modelovaných podnikových procesech. Nejčastěji se setkáváme s dvěma typy - textovými anotacemi a skupinami.

Textové anotace (poznámky) slouží k doplnění komentářů či detailnějších informací o plovoucích objektech. Nemají žádný vliv na průběh procesu, pouze slouží pro upřesnění informací. Jsou reprezentované obdélníkovým prvkem s částečným ohraničením, viz obrázek 8.

Skupina je obdélníkový objekt ohraničený čerchovanou čarou, který poskytuje možnost vizuálního seskupení prvků z různých kategorií do stejné skupiny. Prvek skupina je znázorněný na obrázku 8.



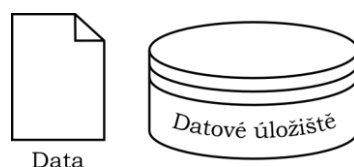
Obrázek 8 -BPMN prvek skupina a textová anotace

Data

Datové prvky umožňují v modelu vizualizovat, jestli jsou pro běh procesu potřebná určitá data, případně že v rámci životního cyklu procesu nějaká data vznikají. Existují dva základní prvky:

Datový objekt, značený prvkem ve tvaru symbolu pro dokument a s popisem, o jaká data se jedná, viz obrázek 9 vlevo. Prvek je prázdný, nebo může mít uvnitř ikonu šipky pro upřesnění, jestli se jedná o vstupní nebo výstupní data.

Datové úložiště umožňuje symbolizovat v modelu jakékoliv úložiště dat a je reprezentované prvkem zobrazeném na obrázku 9 vpravo.



Obrázek 9 - BPMN prvek pro označení dat a datového úložiště

Spojovací objekty

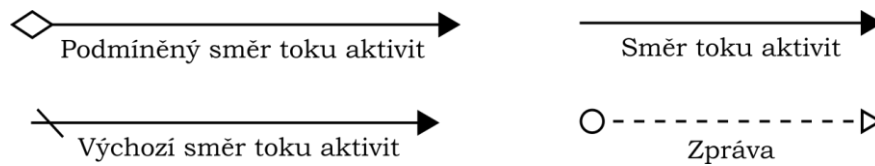
Plovoucí objekty jsou vzájemně propojené, čímž vytváří výslednou jednoznačně definovanou posloupnost aktivit v rámci procesu. K jejich propojení se používají spojovací objekty v podobě čáry, zakončené šipkou ve směru posloupnosti. Notace definuje tři druhy spojovacích objektů: směr toku aktivit, tok zpráv a přiřazení.

Směr toku aktivit je reprezentován plnou čarou a slouží k vzájemnému propojení plovoucích objektů, viz obrázek 10. Používá se pro spojení prvků v rámci stejného bazénu.

Podmíněný směr toku aktivit se od klasického toku aktivit liší symbolem kosočtverce na začátku toku, viz obrázek 10. Tímto prvkem lze nahradit použití klasické rozhodovací brány, přičemž jednotlivé podmínky přechodu jsou součástí popisku daného toku.

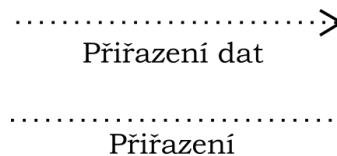
Výchozí směr toku značí směr toku v případě, že žádná z podmínek u podmíněného směru toku není splněna. Je reprezentován přeškrtnutím na začátku toku, viz obrázek 10.

Tok zpráv se používá k propojení plovoucích objektů mezi různými bazény a umožňuje tedy komunikaci mezi různými bazény pomocí zasílání zpráv [24]. Je reprezentován čárkovanou čarou s kolečkem na začátku, viz obrázek 10.



Obrázek 10 - BPMN spojovací prvky Podmíněný a Výchozí směr toku aktivit, Směr toku aktivit a Tok zpráv

Přiřazení je posledním typem spojovacích objektů. Je reprezentované tečkovanou čarou a slouží pro propojení textových anotací (poznámek) a dat spolu s plovoucími objekty. Čára může být zakončená šipkou, v závislosti na tom, jestli se jedná pouze o spojení dvou prvků nebo je potřeba vyjádřit vztah mezi cílovým prvkem a plovoucím objektem, viz obrázek 11.



Obrázek 11 - BPMN spojovací prvky pro přiřazení artefaktů

Výhody BPMN:

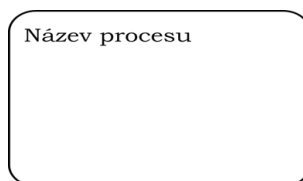
- Jedná se o poměrně novou notaci, která vychází z celé řady používaných modelovacích notací, díky čemuž obsahuje nejlepší myšlenky z nich a zároveň se tvůrci poučili z jejich nedostatků.
- Díky malému množství základních prvků je výsledný model dobře srozumitelný pro všechny zainteresované uživatele i bez znalostí z oblasti modelování podnikových procesů.
- Umožňuje vytvořit model libovolně složitého podnikového procesu.
- Verze BPMN 1.0 má přímou podporu převedení vizuálního modelu do jazyka BPEL (*Business Process Execution Language*), verze BPMN 2.0 má vlastní formát jazyka pro specifikaci procesu ve spustitelném formátu.

2.1.2. UML - Diagram aktivit

UML je modelovací jazyk vytvořený a spravovaný konsorciem OMG [8]. První verze UML byla zveřejněna roku 1995 a v současné době (leden 2018) je nejnovější verzí UML 2.5, zveřejněná roku 2015. Hlavním cílem této notace bylo v počátku poskytnout jednotný modelovací jazyk pro analýzu, návrh a dokumentaci softwarových systémů. V současné době se jedná o univerzální jazyk používaný například i pro modelování podnikových a dalších procesů. UML se na rozdíl od BPMN skládá z celé řady různých druhů diagramů, určených pro různé druhy vizualizací. Pro modelování podnikových procesů je používán Diagram aktivit, patřící do skupiny diagramů chování.

Diagram aktivit slouží k vizualizaci návaznosti vykonávání jednotlivých akcí nebo toku dat v rámci modelovaného systému. V následující části textu jsou představené nejdůležitější prvky Diagramu aktivit, využívané při modelování podnikových procesů [12, 13]:

Aktivita je v případě modelování podnikových procesů prvek, který reprezentuje celý podnikový proces. Slouží tedy k vizuálnímu seskupení všech prvků konkrétního podnikového procesu uvnitř obdélníku s lehce zaoblenými rohy, který obsahuje název procesu, viz obrázek 12.



Obrázek 12 - UML Diagram aktivit, prvek Aktivita

Počáteční a koncový bod jsou prvky, reprezentující počátek/konec procesu. Počáteční bod je symbolizován vyplněným kruhem a koncový bod menším vyplněným kruhem s dvojitým okrajem, viz obrázek 13. Prvky mohou navíc obsahovat text s popiskem doplňujícím informace o daném prvku.

Kromě těchto dvou bodů existuje také **konečný bod**, reprezentovaný kruhem s překříženými čarami uvnitř, viz obrázek 13. Konečný bod značí ukončení procesu v průběhu jeho vykonávání. Koncový bod na rozdíl od něj označuje úspěšné dokončení celého procesu.



Obrázek 13 - UML Diagram aktivit, prvky počáteční, koncový a konečný bod

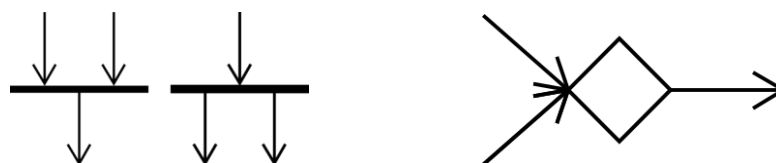
Akce je stejně jako v případě Aktivity u notace BPMN základním stavebním prvkem celého modelu procesu. Má stejný tvar, jako v případě BPMN, tedy obdélník se zaoblenými rohy a s názvem akce, kterou reprezentuje uvnitř, viz obrázek 14. Symbolizuje jednotlivé kroky/úkoly, ze kterých se modelovaný proces skládá. Prvek je možné dělit na akci a pod-aktivitu. Akce se využívají v případech, kdy danou akci už nelze rozdělit na posloupnost akcí. Pod-aktivity se používají v komplexnějších procesech, kdy daná pod-aktivita sama o sobě vykonává několik akcí, ale její detailní rozložení není v konkrétním modelu potřeba. Prvek pod-aktivity je od atomické akce odlišený ikonou v pravé spodní části, viz obrázek 14.



Obrázek 14 - UML Diagram aktivit, prvek reprezentující akci a pod-aktivitu

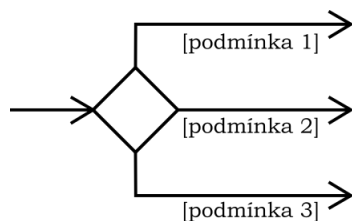
Sloučení a rozdělení jsou prvky používané pro sloučení více různých paralelních toků do jednoho společného, nebo naopak rozdělení jednoho toku do více různých paralelních toků. Tyto prvky jsou značeny vyplněným úzkým obdélníkem, který má vstupní a výstupní toky, viz obrázek 15. Při sloučení může proces pokračovat až v případě, kdy do tohoto prvku dorazí všechny toky a všechny se tedy synchronizují. Sloučení a rozdělení jsou v notaci BPMN zastoupeny jedním prvkem zároveň a to paralelní bránou.

Existuje také prvek uzlu sloučení, který slučuje více toků do jediného, ale jednotlivé toky na sebe nemusejí v uzlu vzájemně čekat a pokračují ihned dál. Uzel sloučení je znázorněn kosočtvercem, do kterého vede libovolný počet vstupních toků, ale má pouze jeden výstupní.



Obrázek 15 - UML Diagram aktivit, prvek sloučení (vlevo), rozdělení (uprostřed) a uzel sloučení (vpravo)

Rozhodovací prvek má podobně jako v jiných notacích tvar kosočtverce a jednotlivé toky ven jsou doplněné o text s rozhodovací podmínkou, viz obrázek 16. Počet výstupních toků je roven počtu rozhodovacích možností, ale vždy se na základě rozhodnutí vybere právě jeden.



Obrázek 16 - UML Diagram aktivit, rozhodovací prvek

Směr toku je využíván pro vzájemné propojení prvků modelu a je reprezentován plnou čarou, zakončenou šipkou bez výplně, viz obrázek 17.

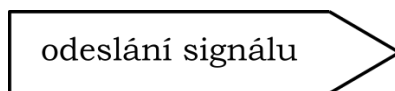


Obrázek 17 - UML Diagram aktivit, prvek směr toku

Události a signály

Signály se dělí na odeslání signálu (zahájení akce) a akceptování signálu (dokončení akce).

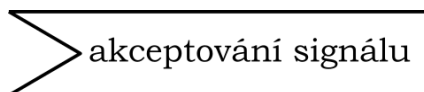
Odeslání signálu generuje událost definovanému příjemci, který může být i v jiném procesu. Po odeslání signálu proces pokračuje ihned dál a nečeká na jakékoliv potvrzení od příjemce. Prvek je znázorněn na obrázku 18.



Obrázek 18 - UML Diagram aktivit, prvek odeslání signálu

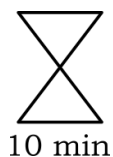
Akceptování signálu

Prvkem akceptování signálu může například začínat celý proces. Pokud prvek akceptování signálu nemá žádný předchozí prvek, spustí se definovaná událost ihned při přijetí signálu. Prvek je možné aktivovat opakovaně, po reakci na příchozí signál opět čeká na nový signál. V případě, že akceptování signálu není prvním prvkem procesu, je možné spustit událost až poté, co proces k tomuto prvku dojde. V modelu je reprezentován obdélníkem s chybějící částí, viz obrázek 19. Chybějící část značí čekání prvku na prvek odeslání signálu.



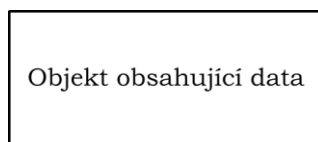
Obrázek 19 - UML Diagram aktivit, prvek akceptování signálu

Časová událost označuje buďto místo v průběhu vykonávání procesu, kde je nutné počkat určitou dobu, než bude proces pokračovat, nebo událost, která spouští akci v určitý okamžik, například na začátku každého dne, na konci měsíce atd. Značí se symbolem přesýpacích hodin, viz obrázek 20.



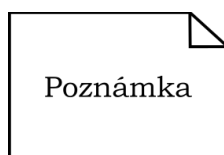
Obrázek 20 - UML Diagram aktivit, prvek časová událost

Objekt je znázorněn obdélníkem, viz obrázek 21 a je využíván pro vizualizaci dat, která se vyskytují v rámci modelovaného procesu.



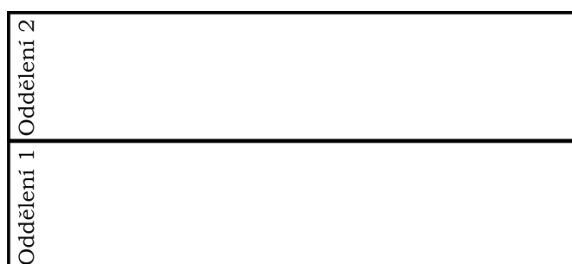
Obrázek 21 - UML Diagram aktivit, prvek objekt

Poznámka umožňuje doplnění modelu o komentáře a detailnější informace. Nemají žádný vliv na průběh procesu, slouží jen k doplnění informací. Prvek reprezentující poznámku je znázorněn na obrázku 22.



Obrázek 22 - UML Diagram aktivit, prvek poznámka

Oddělení je obdobou bazénových drah u notace BPMN a umožňuje vizuální rozdělení diagramu do různých oblastí, viz obrázek 23.



Obrázek 23 - UML Diagram aktivit, prvek oddělení

Základní prvky UML diagramu aktivit lze rozšířit o další prvky z jiných UML diagramů, díky čemuž lze modelovaný podnikový proces popsat detailněji [3]. Při rozšíření o další prvky může být ale výsledný model méně přehledný a složitější na pochopení pro běžné uživatele, jelikož by byla nutná znalost více různých UML diagramů a jejich notací.

Výhody UML diagramu aktivit:

- Notace známá všem softwarovým vývojářům.
- Díky malému množství základních prvků je výsledný model dobře srozumitelný pro všechny zainteresované uživatele i bez znalostí z oblasti modelování podnikových procesů.
- Umožňuje vytvořit model libovolně složitého podnikového procesu.

2.1.3. EPC

Event-driven Process Chain (EPC), neboli model procesu řízeného událostmi byl vyvinutý skupinou autorů Scheer, Keller a Nüttgens v rámci frameworku ARIS (*Architecture of Integrated Information System*) [10, 11]. Jedná se o modelovací jazyk pro popis pracovních postupů a k vizualizaci návaznosti (řetězení) jednotlivých událostí a aktivit v rámci podnikových procesů. EPC byl inspirován z velké části vývojovým diagramem - jedná se tedy v podstatě o vylepšený vývojový diagram. Při návrhu EPC byl kladen důraz na snadnou srozumitelnost výsledného modelu.

Událost je pasivním prvkem, který popisuje, za jakých okolností dochází ke spuštění aktivity/procesu, nebo naopak obsahuje informace o výsledku aktivity či procesu. Události odkazují na určitý okamžik v životním cyklu procesu a každý proces tímto prvkem vždy začíná a končí. V modelu je událost reprezentována prvkem ve tvaru šestiúhelníku, viz obrázek 24.



Obrázek 24 - EPC prvek značící událost

Funkce je naopak aktivním prvkem a popisuje transformaci svého vstupu na určitý výstup, tedy co se má v rámci dané části procesu vykonat. Z výše uvedeného popisu prvku událost je patrné, že každý prvek funkce má v rámci modelu jednoznačně definovaný počátek a konec pomocí událostí. V modelu je funkce značená obdélníkem se zaoblenými rohy, viz obrázek 25.

Funkce

Obrázek 25 - EPC prvek používaný pro vizualizaci funkce

Logické spojky

Logické spojky jsou používány pro spojení či rozdělení událostí a aktivit. Platí pravidlo, že na jednom konci logické spojky jsou buďto události nebo aktivity a na druhém konci vždy opačné prvky. Jsou rozlišovány tři základní typy logických spojek: AND, OR a XOR, pro které jsou jasně definovaná pravidla, jaké prvky se musí či nesmějí objevit na vstupu nebo výstupu dané logické spojky.

Spojka AND se aktivuje v momentě, kdy jsou na vstupu této logické spojky připravené všechny příchozí toky. Slouží tedy k synchronizaci všech vstupních či výstupních toků. Značí se kolečkem se symbolem logického operátoru AND uprostřed, viz obrázek 26. Pro spojku AND mohou nastat celkem čtyři možné případy, dva pro spojení a dva pro rozdělení:

- 1) více aktivit se spojí do jedné události
- 2) více událostí se spojí do jedné aktivity
- 3) jedna aktivita se rozdělí na více událostí
- 4) jedna událost se rozdělí na více aktivit

Spojka OR je aktivována na vstupu jednou nebo více cestami a na výstupu je jedna nebo více cest, kterými tok pokračuje dál. Spojka OR se značí kolečkem se symbolem logického operátoru OR uprostřed, viz obrázek 26 a nelze použít pro rozdělení jedné události na více aktivit. Může tedy dojít ke třem scénářům:

- 1) jedna nebo více aktivit se spojí do jedné události
- 2) jedna nebo více událostí se spojí do jedné aktivity
- 3) jedna aktivita se rozdělí na jednu nebo více událostí

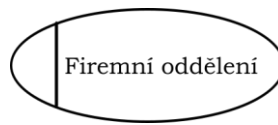
Spojka XOR slouží pro vzájemné vyloučení jednotlivých cest. Na vstupu se aktivuje na základě právě jedné cesty a na výstupu pokračuje také právě jednou cestou. Stejně jako u OR zde platí podmínka, že nelze použít pro rozdělení jedné události na více aktivit. Značí se kolečkem s nápisem XOR uprostřed, viz obrázek 26. Pro operátor XOR mohou nastat následující stavy:

- 1) jediná z aktivit spojí vše do jedné události
- 2) jediná z událostí spojí vše do jedné aktivity
- 3) jedna aktivita se rozdělí na právě jedinou z více událostí



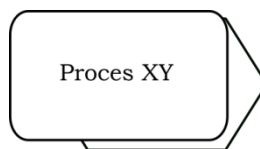
Obrázek 26 - EPC prvky pro logické spojky: OR, AND a XOR

Organizační jednotka slouží k vizualizaci, jaká konkrétní uživatelská role nebo podnikové oddělení je zodpovědné za konkrétní funkci. K dané funkci je organizační jednotka připojena plnou čarou. Prvek je znázorněn na obrázku 27.



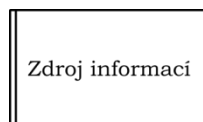
Obrázek 27 - EPC prvek znázorňující organizační jednotku

Procesní cesta umožňuje znázornění souvislostí a spojení aktuálně modelovaného procesu s jinými procesy. Lze tedy jednoduše modelovat například situaci, kdy aktuální proces pokračuje po dokončení předchozího procesu, přičemž celý předchozí proces je v modelu zastoupený právě tímto jediným prvkem. Prvek procesní cesta je vizualizovaný kombinací prvků událost a funkce, přičemž funkce je v popředí, viz obrázek 28.



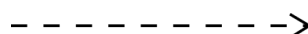
Obrázek 28 - EPC prvek Procesní cesta

Zdroj informací označuje zdroj informací, jako například vstupní a výstupní data. Prvek je symbolizován obdélníkem s dvojítm okrajem na levé straně, viz obrázek 29.



Obrázek 29 - EPC prvek symbolizující zdroj informací

Spojovací objekty umožňují vzájemné propojení jednotlivých prvků v rámci modelu. Pro připojení organizačních jednotek slouží plná čára bez šipky, pro připojení dat se používá plná čára zakončená šipkou a pro znázornění směru toku čerchovaná čára zakončená šipkou ve směru toku, viz obrázek 30.



Obrázek 30 - EPC spojovací prvek pro znázornění směru toku

Výhody EPC:

- Snadná vizualizace i složitých podnikových procesů jednoduchým spojením událostí, aktivit a logických spojek.
- Díky malému množství prvků (základní notace jich má pouze pět) lze výsledný model snadno pochopit i bez znalostí z oblasti modelování podnikových procesů.

2.1.4. Shrnutí

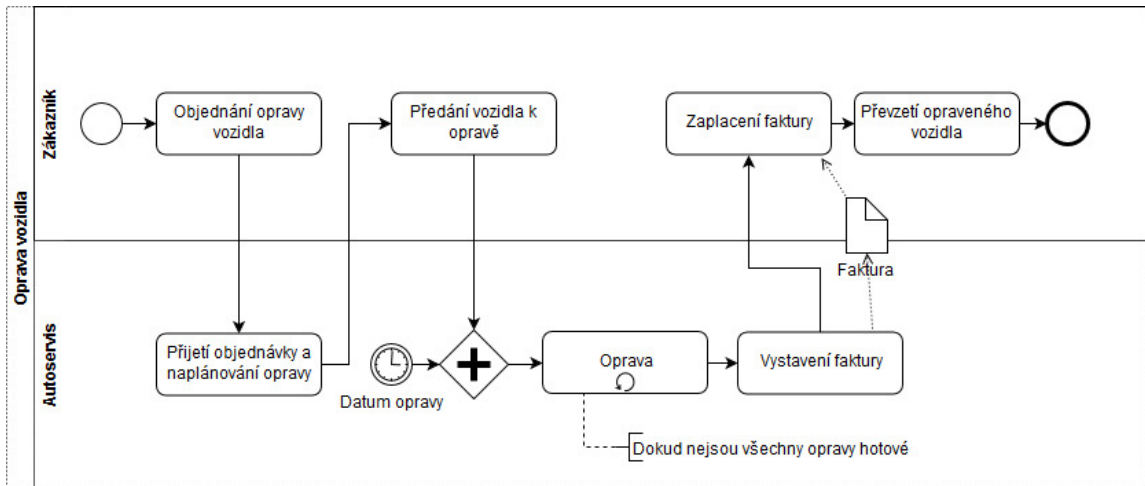
Před učiněním rozhodnutí, jakou notaci využít pro modelování podnikových procesů, je hlavním úkolem položit si otázku, jaký má být význam výsledného modelu a za jakým účelem je vytvářený [14]. Pro zodpovězení této otázky pomůže definování a pochopení podnikových cílů, které při výběru určí správný směr.

Základními otázkami při výběru vhodné notace mohou být:

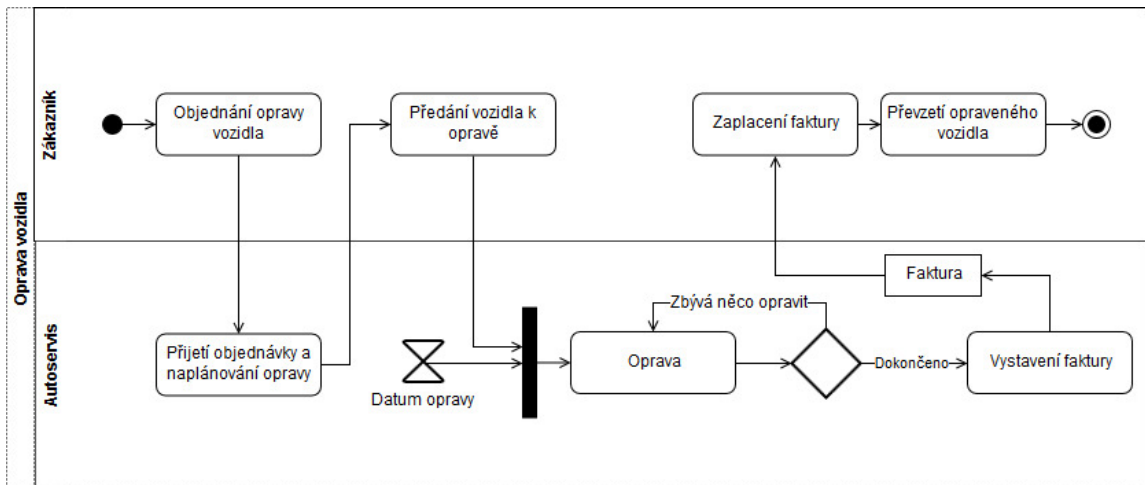
- Za jakým účelem se model podnikových procesů vytváří?
- Pro jaké uživatele bude daný model především určený?
- S kým se budou jednotlivé namodelované podnikové procesy konzultovat, s analytikem, managementem, softwarovým vývojářem?

Pokud výsledný model nemusí mít podrobně do detailu popsané jednotlivé podnikové procesy a jejich aktivity, je dobrou volbou notace EPC, která poskytuje vyváženost mezi striktním dodržováním pravidel a jednoduchým způsobem modelování procesů. EPC je také vhodnější, pokud je model určený managementu, který požaduje měření různých metrik a vizualizaci podnikové strategie, jako například Balanced Scorecard model (metoda pro strategické hodnocení výkonnosti firmy) nebo Objective Tree (metoda pro převedení nejasné koncepce návrhu do konkrétnějších požadavků zákazníků). Naopak pokud je model určený například softwarovým vývojářům, kteří potřebují mít k dispozici detailní informace o jednotlivých procesech, pak je vhodnější možností zvolit notaci BPMN nebo UML diagram aktivit. Model vytvořený s použitím BPMN či UML je navíc ve výsledku graficky více efektivní a stejný diagram bude mít oproti EPC menší velikost.

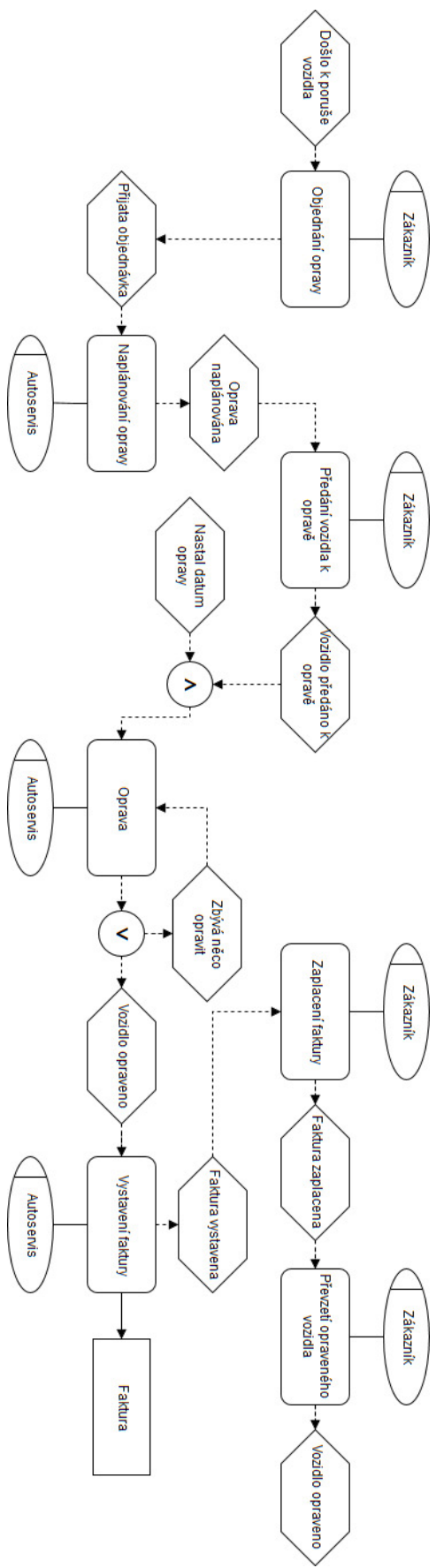
Názorný příklad obsahující vizualizaci modelu podnikového procesu „Oprava vozidla“ pomocí notací BPMN, UML Diagramu aktivit a EPC si lze prohlédnout na obrázcích 31, 32 a 33.



Obrázek 31 - vizualizace podnikového procesu „Oprava vozidla“, pomocí BPMN



Obrázek 32- vizualizace podnikového procesu „Oprava vozidla“, pomocí UML diagramu aktivit



Obrázek 33 - vizualizace podnikového procesu „Oprava vozidla“, pomocí EPC

Z příkladu je patrný znatelný rozdíl, mezi kompaktností modelů vytvořených za použití UML Diagramu aktivit a BPMN oproti modelu vytvořenému s využitím notace EPC [15]. Větší velikost celého EPC modelu podnikového procesu může vést k nepřehlednosti a horší srozumitelnosti pro uživatele. V příkladu výše lze také pozorovat, že srozumitelnost a čitelnost vizualizace podnikových procesů modelovaných pomocí UML Diagramu aktivit a BPMN je pro všechny zainteresované osoby, počínaje analytiky, přes softwarové vývojáře až po manažery v podstatě stejná.

V některých případech může být při výběru notace pro modelování podnikových procesů také důležité, jak snadno lze výsledný grafický model převést na spustitelný kód, například na jazyk WS-BPEL (*Web Services Business Process Execution Language*), často označovaný pouze zkráceně BPEL (*Business Process Execution Language*), který slouží pro jednoznačný popis podnikových procesů a umožňuje jejich automatizaci [9]. Zatímco BPMN verze 2.0 přímo obsahuje podporu převodu podмноžiny svých prvků (prvků BPMN z verze 1) do jazyka BPEL, u UML Diagramu aktivit není mapování prvků do jazyka BPEL nijak definováno. Tento rozdíl se snaží smazat řada externích řešení pro podporu převedení diagramu aktivit do jazyka BPEL, ale nenabízejí zcela automatizované mapování.

Při porovnání UML Diagramu aktivit a BPMN z hlediska komplexnosti jednotlivých grafických prvků, dostupných pro modelování reálných podnikových procesů platí, že je možné prostřednictvím dostupných grafických prvků obou notací popsat stejné, i velmi složité, podnikové procesy. Šířka pokrytí procesů, které lze danou notací modelovat, je tedy v obou případech stejná. Jediným rozdílem je, že v rámci UML Diagramu aktivit je v některých případech nutné použít více grafických prvků, zatímco v BPMN lze daný případ zobrazit s využitím jediného prvku, takže je výsledný model vytvořený pomocí BPMN menší a díky tomu i přehlednější.

2.1.5. Návrh notace pro vizualizaci podnikových procesů

V rámci tématu této diplomové práce bude vizualizace podnikových procesů sloužit především pro komunikaci konzultanta se zainteresovanými uživateli, což mohou být uživatelé napříč celým podnikem, počínaje ředitelem a vedoucími pracovníky až po účetní a skladníky a je tedy nutné předpokládat, že tito uživatelé nemusí mít žádné nebo jen malé vědomosti z oblasti vizualizace podnikových procesů.

Na základě výše uvedené analýzy jednotlivých notací pro grafickou vizualizaci podnikových procesů jsem došel k rozhodnutí vycházet z BPMN 2.0. Pro lepší přehlednost výsledného modelu bude omezený počet prvků na následující prvky:

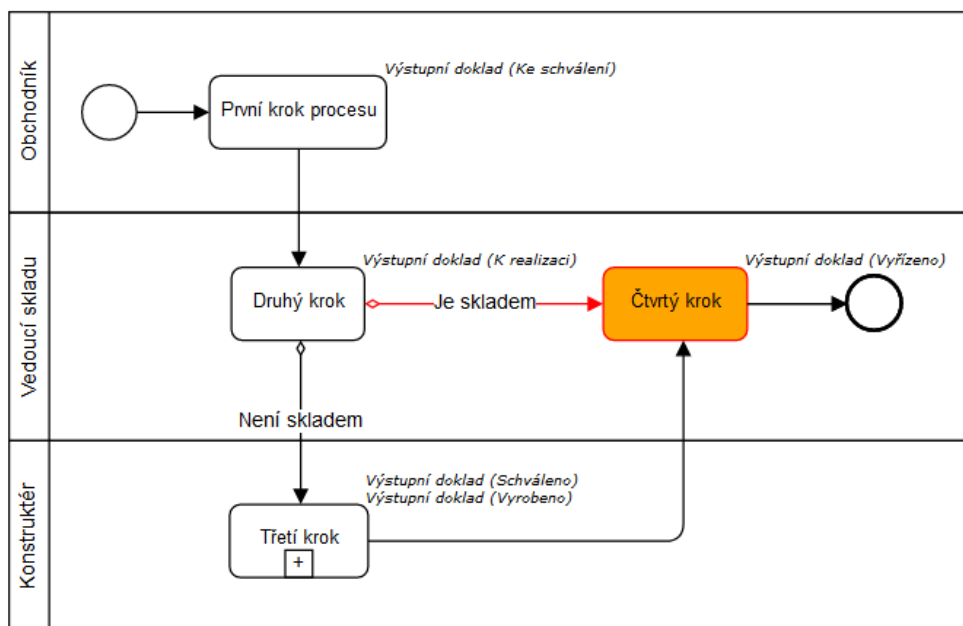
- Aktivita (úkol, podproces)
- Události (počáteční a koncová)
- Spojovací objekty (směr toku aktivit a podmíněný směr toku aktivit)
- Bazén
- Bazénové dráhy

V rámci snížení počtu použitých prvků pro zvýšení přehlednosti výsledného modelu bude místo rozhodovacích bran zvolená vizualizace větvení pouze prostřednictvím podmíněných toků, přičemž každý podmíněný tok obsahuje popis podmínky, za jakých podmínek proces daným tokem může pokračovat. Z popisu podmínek bude jasné patrné, jestli je jedná o exklusivní větvení, nebo paralelní.

Jednotlivé prvky vizualizace procesu budou umístěné uvnitř bazénu, který bude reprezentovat daný proces. Jednotlivé dráhy uvnitř bazénu budou reprezentovat zainteresované uživatele (uživatelské role).

Oproti standardu notace BPMN se bude ve výsledné vizualizaci lišit pouze vizualizace vstupních a výstupních dokladů. Na základě požadavku zadavatele na omezení vizualizovaných dat budou ve vizualizaci zobrazovány pouze výstupní doklady ve formě prostého textu. Pokud jsou tedy na výstupu jednotlivých kroků procesu generována nějaká data, jednotlivé výstupní doklady budou vypsány vpravo nahoře u daného procesu.

Zároveň bude kontrolováno, jestli výstupní data předchozího kroku obsahují data, která jsou vyžadována jako vstupní data následujícího kroku. Pokud tomu tak není, bude tato chyba v konzistenci dat vizualizována změnou barvy daného toku a cílového kroku procesu. Návrh, jak by mohla vypadat výsledná vizualizace podnikového procesu dle výše popsané notace, je znázorněn na obrázku 34.



Obrázek 34 - Ukázka vizualizace podnikového procesu dle navržené notace

2.2. Existující nástroje pro vizualizaci podnikových procesů

Tato kapitola se věnuje popisu již existujících nástrojů pro modelování a vizualizaci podnikových procesů, kterých v současné době existuje celá škála a nabízejí řadu možností pro modelování podnikových procesů a práci s nimi.

Existující nástroje lze rozdělit do dvou hlavních kategorií:

- Komplexní nástroje pro modelování, vizualizaci, analýzu, optimalizaci a testování podnikových procesů.
- Základní nástroje, umožňující modelování a vizualizaci podnikových procesů, bez možností hlubší analýzy a funkcí pro optimalizace podnikových procesů.

Jako zástupce komplexních nástrojů lze uvést například nástroje Enterprise architect¹ nebo Visual Paradigm². Komplexní nástroje umožňují modelování a vizualizaci podnikových procesů s využitím různých notací, kontrolu konzistence dat napříč jednotlivými procesy, dbají na dodržování nejlepších praktik pro modelování podnikových procesů a jejich vzájemného propojení. Zároveň umožňují zobrazování přehledů a analýz nad definovanými daty nebo například definování scénářů, umožňujících animaci a simulaci vykonávání procesů pro kontrolu průchodnosti jednotlivých procesů, vzájemných návazností a optimalizaci jednotlivých procesů a využití zdrojů.

¹ <https://sparxsystems.com/products/ea/index.html>

² <https://www.visual-paradigm.com>

Funkcionalitu těchto nástrojů lze jakkoliv rozšířit prostřednictvím vlastních rozšíření a je tedy možné si v případě potřeby napsat vlastní rozšíření pro dosažení požadované funkcionality.

Ze základních nástrojů lze jako zástupce zmínit například nástroj Lucidchart³ nebo yEd Graph Editor⁴. Ve většině případů se jedná o nástroje zaměřené na modelování a vizualizaci grafů obecně a kromě základního modelování podnikových procesů a jejich vizualizace s využitím různých notací nabízejí žádné komplexnější funkce, se kterými se lze setkat u výše zmíněných nástrojů.

Cílem výsledné aplikace není komplexní práce s podnikovými procesy a jejich exaktní nadefinování do detailu, tudíž většina hlavních vlastností a předností komplexních nástrojů by zůstala nevyužitá. Jedná se ale o velmi dobrou inspiraci, jaký má potenciál modelování podnikových procesů, práce s nimi a jejich vizualizace a jakou funkcionalitu lze pro zkvalitnění a zefektivnění práce s podnikovými procesy použít.

V kapitole 6.5 se nachází analýza existujících knihoven pro vizualizaci podnikových procesů, patřících do kategorie základních nástrojů, ze kterých byla následně nejvíce vyhovující knihovna vybrána pro implementaci vizualizace podnikových procesů ve výsledné aplikaci.

³ <https://www.lucidchart.com/pages>

⁴ <https://www.yworks.com/products/yed>

3. Stávající způsob popisu podnikových procesů

Cílem této kapitoly je seznámení čtenáře se stávajícím způsobem popisu podnikových procesů, ke kterému dochází v rámci analýzy před nasazením ERP systému FLORES v konkrétním podniku.

Před samotným nasazením ERP systému FLORES v nějakém podniku je nejprve nutná důkladná analýza veškerých podnikových procesů a informací s nimi souvisejícími. Analýzu zpracovávají konzultanti v průběhu analytických schůzek se zainteresovanými uživateli z daného podniku a cílem této analýzy je komplexní pochopení a sepsání podkladů popisujících dané podnikové procesy. Sepsané podklady následně slouží pro konfiguraci a nastavení ERP systému FLORES pro daný podnik.

Na základě získaných informací je nutné vyhodnotit nejen úzká místa, která lze nasazením systému odstranit či zefektivnit, ale zároveň je také důležité porovnat jednotlivé podnikové postupy a procesy v kontextu vlastností a možností systému FLORES. Každý podnik je jedinečný a může mít své specifické procesy, které ERP systém v základní konfiguraci vůbec nepodporuje, případně jsou implementovány trochu rozdílným způsobem. Cílem tedy není jen slepé zmapování a popsání jednotlivých podnikových procesů, ale je zároveň chtěné dané procesy přímo navázat na funkcionalitu systému a přizpůsobit je do finální podoby, ve které budou v ERP systému FLORES implementovány. Během analýzy je tedy nutné najít kompromis mezi přizpůsobením funkčnosti zákazníkovi na míru a přizpůsobení procesů do podoby podporované systémem FLORES.

Zápisy z jednotlivých analytických schůzek byly dříve zapisovány v libovolném textovém editoru a každý konzultant vytvářel zápis svým vlastním způsobem. V závěru analýzy následně konzultant na základě zápisů z jednotlivých schůzek ručně v libovolném textovém editoru vyplňoval informace o každém podnikovém procesu, jeho jednotlivých krocích a další důležité informace do tabulek v předem definované struktuře. Výsledný dokument, který vyplněním všech těchto tabulek vzniknul, se nazývá Projekt implementace a obsahuje popis jednotlivých podnikových procesů a jejich kroků.

Zápis informací do daných tabulek je ale pro běžné uživatele velmi nepřehledný a je velmi těžké se v nich zorientovat, pochopit informace v nich uvedené a především vidět souvislosti a návaznosti jednotlivých kroků. To představuje značný problém, jelikož je pro správnou implementaci a nasazení ERP systému v podniku nezbytné, aby zainteresovaní uživatelé z daného podniku u jednotlivých podnikových procesů potvrdili, že byly konzultantem pochopeny a zapsány korektně a odpovídají skutečnosti a chtěnému výsledku. Zároveň výsledný zápis Projektu implementace neumožňuje snadný způsob pro automatizované získání informací, jako například získání všech

uživatelských rolí, typů a stavů dokladů a dalších dat, případně vykonávání dalších akcí nad danými daty.

Zároveň bylo některými zákazníky vyžadováno vytvoření vizualizace podnikových procesů pro lepší přehlednost informací. V takových případech bylo nutné danou vizualizaci vytvářet ručně v nějakém externím nástroji, do kterého musel konzultant ručně znovu přepisovat a definovat jednotlivé informace zvlášť.

Ukázku, jak vypadá definice procesu a jednoho z jeho kroků pomocí výše popsaného způsobu zadávání informací do tabulek v textovém editoru, si lze prohlédnout v tabulkách 2 a 3, přičemž v tabulce 2 je definice procesu „Vyskladnění“ a následující tabulka obsahuje definici jednoho z kroků daného procesu.

Proces	SK001 Vyskladnění			
Kód	01			
Aktivace	Naplánovaná úloha uvnitř IS FLORES			NE
	Naplánovaná úloha mimo IS FLORES			NE
	Jiné procesy, zásobník	VYD	V	ANO
	Mimo systém (např. telefonát)			NE
Kapitola	04 – Logistika a sklad			
Cíl	V pravidelných intervalech ověřit expedovatelnost objednávek přijatých...			
Aktivace	PR002K01			
Ukončení	PR002K01/--			
Popis procesu	V pravidelných intervalech dochází ke kontrole, zda...			

Tabulka 1 - Definice podnikového procesu pomocí tabulky v dokumentu Projekt implementace

Krok	SK001K01 Převzetí k vyskladnění			
Vstupuje	Typ dokladu	Řada	Stav	Odpovědná role
	Výdejka	xx	V přípravě	Skladník
Vystupuje	Výdejka	xx	Převzato	Skladník
Vykonává	Skladník			
Popis	Za zásobníku AM VYD V přípravě odebírají skladníci jednotlivé výdejky ke zpracování. Zahájení vyskladňování nad výdějkou provede skladník v IS tlačítkem Změna stavu a určením stavu Převzato.			
Podproces	-			
Následující krok	SK00102			

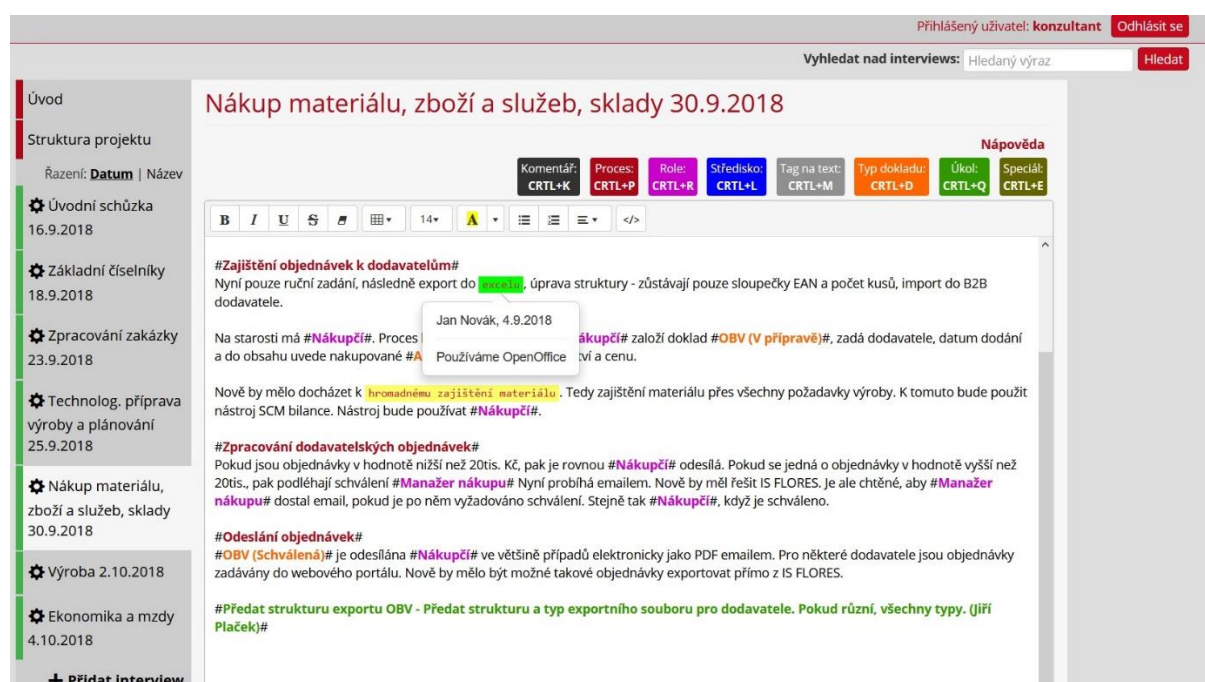
Tabulka 2 - Definice jednoho z kroků výše uvedeného podnikového procesu

Dokument Projekt implementace a zápisy z jednotlivých schůzek měl konzultant uložené pouze offline ve svém zařízení a zákazníkovi museli být jednotlivé dokumenty zasílány. To vedlo k problémům s předáváním a údržbou aktuálních verzí jednotlivých dokumentů a případnému doplňování připomínek zákazníkem.

V roce 2018 byla za účelem optimalizace a sjednocení vytváření zápisů z jednotlivých schůzek a odstranění některých z výše zmíněných problémů implementována a nasazena nová webová aplikace s názvem Interview, sloužící k vytváření strukturovaných zápisů z analytických schůzek v uceleném formátu. Tato aplikace je v současné době používána všemi konzultanty společnost FLORES Software pro vytváření zápisů analytických schůzek a ukládá jednotlivá data přímo do databáze systému FLORES.

Přístup do aplikace Interview mají předem definovaní uživatelé, kteří se dělí na konzultanty s právy měnit obsah jednotlivých zápisů a na návštěvníky. Návštěvníci mají omezená práva pouze na zobrazování jednotlivých zápisů a vkládání komentářů k jednotlivým textům.

Stěžejní vlastností aplikace je možnost používání chytrých tagů v jednotlivých zápisech. Tyto tagy označují důležité údaje, jako například procesy, uživatelské role, střediska, typy a stavy dokladů a další. Obrázek 35 obsahuje ukázkou konkrétního zápisu v aplikaci Interview, přičemž lze vidět strukturu textu, použití komentářů a chytrých tagů.



The screenshot displays the 'Interview' application interface. At the top, it shows the user is logged in as 'konzultant' and provides a search bar for interviews. The main content area is titled 'Nákup materiálu, zboží a služeb, sklady 30.9.2018'. On the left, there is a sidebar with a project structure and a list of interviews, including 'Úvod', 'Struktura projektu', 'Úvodní schůzka', 'Základní číselníky', 'Zpracování zakázky', 'Technolog, příprava výroby a plánování', 'Nákup materiálu, zboží a služeb, sklady', 'Výroba', and 'Ekonomika a mzdy'. The main text area contains several paragraphs of text with various smart tags (e.g., #Zajištění objednávky k dodavatelům#, #Nákupčí#, #Manažer nákupu#, #OBV (Schválená)#, #Odeslání objednávky#, #Předat strukturu exportu OBV#) and a comment from 'Jan Novák, 4.9.2018'. A toolbar at the top of the text area allows for formatting and tagging. The interface is clean and professional, designed for collaborative work and documentation.

Obrázek 35 - Aplikace Interview, sloužící k zápisu analytických schůzek

Díky použití chytrých tagů je následně možné celý zápis strojově zpracovat a získat z něj informace o jednotlivých zadaných datech a vzájemných souvislostech mezi nimi. Na obrázku 36 se nachází příklad zobrazení informací ke konkrétnímu procesu,

konkrétně jaké tagy se v daném procesu vykytují a v jakých konkrétních zápisech je daný proces zmíněný.

The screenshot displays a web interface for process management. On the left is a vertical navigation menu with the following items: 'Cíle zavedení a klíčové procesy', 'Harmonogram a osnova', 'Organizační struktura', 'Procesy' (highlighted with a red bar), 'Role', 'Typy dokladů', 'Speciály', 'Komentáře', 'Úkoly', and 'Uživatelé'. The main content area is titled 'Procesy - Zobrazení vzájemných souvislostí'. Below the title, there is a search bar containing the text 'Zajištění objednávek k dodavatelům' and a red button labeled 'Zobrazit výskyty'. The selected process is 'Zajištění objednávek k dodavatelům (1)'. Under the heading 'Role v procesu:', there is a list item: '· **Nákupčí** (Nákup materiálu, zboží a služeb, sklady 30.9.2018)'. Under 'Typy dokladů v procesu:', there are two items: '· **Artikl (Schválený)** (Nákup materiálu, zboží a služeb, sklady 30.9.2018)' and '· **OBV (V přípravě)** (Nákup materiálu, zboží a služeb, sklady 30.9.2018)'. Under 'Střediska v procesu:', the text reads: 'V procesu se v současné době nenachází žádné středisko'. Under 'Texty k procesu:', there is a link: '**Nákup materiálu, zboží a služeb, sklady** (Přejít do Interview)'. The interface uses a clean, modern design with a light grey background and red accents for buttons and highlights.

Obrázek 36 - Zobrazení informací o vybraném procesu uvedeném v zápise

4. Specifikace požadavků uživatelů aplikace

V této kapitole jsou definovány jednotlivé požadavky zadavatele práce, které by měla výsledná aplikace pro modelování a vizualizaci podnikových procesů splňovat.

Hlavními cíli aplikace jsou:

- Usnadnění vytváření Projektu implementace, díky využití dat, definovaných v rámci aplikace Interview.
- Jednoduché modelování a správa podnikových procesů a jejich kroků, dle požadované struktury dat a s využitím objektů systému FLORES.
- Kontrola konzistence zadávaných dat.
- Výpis varování v případě vyhodnocení neúplných či nevyužívaných dat.
- Přehledná vizualizace podnikových procesů a jejich kroků.
- Jednoduché a přehledné uživatelské rozhraní s responzivním vzhledem.
- Přístup do aplikace rozdělený na konzultanty a obyčejné uživatele.
- Automatická konfigurace částí systému FLORES a propojení zadávaných dat s konkrétními agendami tohoto systému.

4.1. Přístup k aplikaci

Přístup do aplikace bude povolený pouze přihlášeným uživatelům. Ověření přihlášení bude probíhat ověřením prostřednictvím uživatele systému FLORES. Jsou definovány dvě uživatelské role, konkrétně:

- Konzultant
 - Kompletní přístup a správa všech záznamů
- Obyčejný uživatel
 - Pouze prohlížení jednotlivých záznamů

V počáteční fázi definování jednotlivých podnikových procesů bude mít k aplikaci přístup pouze konzultant, který má plná práva na vytváření nových záznamů a jejich správu. Po naplnění aplikace daty bude udělen přístup „obyčejným uživatelům“, typicky zainteresovaným osobám ve firmě. Tito uživatelé budou mít právo pouze prohlížet jednotlivé podnikové procesy, jejich kroky a informace k nim uvedené.

4.2. Modelování podnikových procesů a definování souvisejících dat

V první řadě je důležité umožnit konzultantům převod dat, definovaných v rámci aplikace Interview, do aplikace Projekt implementace. S využitím těchto dat bude následně aplikace umožňovat konzultantům definovat jednotlivé podnikové procesy,

jejich kroky a další související informace, které by standardně byly uvedené v dokumentu Projekt implementace, viz kapitola 3.

Je chtěné, aby při modelování podnikových procesů bylo zadáváno jen nezbytně nutné množství dat a aby zadávané informace byly provázány s objekty, definovanými v systému FLORES.

4.3. Vizualizace

Pro přehledné zobrazení definovaných procesů a usnadnění konzultace se zainteresovanými osobami bude aplikace umožňovat vizualizaci podnikových procesů, která přinese oproti současnému systému zápisů jednotlivých procesů a jejich kroků do tabulek zvýšení celkové přehlednosti a srozumitelnosti informací o jednotlivých podnikových procesech. Vizualizace tedy povede ke značnému zjednodušení při procházení a ujasňování informací ohledně fungování jednotlivých podnikových procesů se zainteresovanými uživateli a přispěje ke snížení vzniku nedorozumění a špatně nadefinovaných podnikových procesů z důvodu nesrozumitelnosti informací pro zákazníka.

Zainteresovaní uživatelé napříč celým podnikem mají různý obor a stupeň vzdělání, což je velmi důležité brát v potaz při výběru notace pro modelování jednotlivých podnikových procesů. Z důvodu přehlednosti a snadné srozumitelnosti výsledného modelu podnikových procesů i pro uživatele, kteří uvidí vizualizaci podnikových procesů poprvé v životě, je kladen důraz na jednoduchost vizualizace.

4.4. Kontrola konzistence

Při definování jednotlivých podnikových procesů bude aplikace zároveň zajišťovat automatickou kontrolu konzistence zadávaných dat dle předem definovaných pravidel. Pro jednotlivé podnikové procesy je definovaná sada pravidel, jejichž dodržování je nutné ohlídat, aby byla zajištěna konzistence dat. Je tedy chtěné, aby aplikace umožňovala automatickou kontrolu konzistence zadávaných dat a případně konzultanta upozornila, pokud by došlo k jejich nedodržení. Validace dat zajišťuje, že aplikace neumožní uživateli pokračovat v požadované akci, pokud není konkrétní pravidlo dodrženo. Díky tomu dojde k zamezení vzniku nekonzistentních dat, ke kterým u dřívějšího způsobu vytváření Projektu implementace docházelo, jelikož nebyla možná jejich automatizovaná kontrola a vynucování.

Kromě validace bude aplikace umožňovat zobrazení upozornění v případě, že vyhodnotí nějaká data jako nevyužitá či neúplná. Například v situaci, kdy je v aplikaci nadefinována určitá uživatelská role, která se ale nevyskytuje v žádném z podnikových procesů. Taková situace může vést k tomu, že by v systému byla daná role definována

zbytečně a je možné ji odstranit, nebo že je nutné dohledat procesy, ve kterých by se daná role měla vyskytovat a zjistit, z jakého důvodu se v nich nevyskytuje (proces byl přiřazen jiné uživatelské roli, proces není definován atd.) Takové vyhodnocení je pro aplikaci jednoduché, pokud má k dispozici všechna potřebná data, ale v předchozím způsobu zápisu Projektu implementace mohl konzultant takové situace snadno přehlédnout.

Příklad některých pravidel:

- Procesy a kroky mají vyplněné všechny povinné informace.
- Krok má na výstupu jiná data, než na vstupu (transformuje je). Typicky dojde ke změně stavu dokladu, pokud by k takové transformaci nedošlo, krok nedává smysl.
- Není možné propojení kroku procesu a krokem jiného procesu.
- Kontrola, jestli požadovaný vstupní doklad kroku je výstupním dokladem předchozího kroku.

4.5. Konfigurace systému FLORES

Při modelování procesů bude zároveň docházet přímo ke konfiguraci některých částí ERP systému FLORES, jako například naplnění systému konkrétními uživatelskými rolemi a stavy dokladů.

Zároveň aplikace bude umožňovat propojení jednotlivých kroků podnikových procesů přímo s agendou systému FLORES, aby bylo následně možné použít popis kroku přímo v konkrétních agendách systému FLORES jako uživatelský manuál.

4.6. Technologie

Vzhledem k chtěnému budoucímu propojení s aplikací Interview a snadné dostupnosti aplikace z různých zařízení byla pro realizaci aplikace zadavatelem zvolená webová aplikace, přičemž byl stanovený požadavek na webový server Apache a programovací jazyk PHP. Tyto technologie byly vybrány s ohledem na integraci s aplikací Interview a vzhledem ke zkušenostem zaměstnanců společnosti FLORES s danými technologiemi. Pro umožnění přístupu z různých zařízení je důležitý responzivní vzhled aplikace.

Požadavky zadavatele, ovlivňující použité technologie byly následující:

- Využití pouze nástrojů bez licenčních poplatků.
- Kompletní kontrola nad danou aplikací a jejím vývojem. Možnost přizpůsobení aplikace dle vlastních požadavků.
- Propojení aplikace se systémem FLORES prostřednictvím Web API.

5. Definování vhodné funkcionality aplikace

Na základě předchozích kapitol je v této kapitole definována vhodná funkcionality výsledné aplikace.

Výsledná webová aplikace bude přímo propojena s ERP systémem FLORES, ze kterého budou prostřednictvím Web API získávána a zároveň zpět ukládána všechna data, důležitá pro běh aplikace. Aplikace bude také využívat data, uložená prostřednictvím aplikace Interview, která se také nacházejí v systému FLORES.

Při modelování podnikových procesů a kroků v aplikaci bude dodržována definovaná struktura dat a zadávané informace budou propojené s objekty systému FLORES. U ukládaných dat bude kontrolována jejich úplnost a bude hlídána konzistenci dat. Zároveň budou uživatelům zobrazována upozornění, pokud budou nějaká z dat vyhodnocena jako neúplná nebo například pokud nejsou některá data zatím nikde používána.

Aplikace bude umožňovat vizualizaci podnikových procesů, s využitím notace popsané v kapitole 2.1.5. Jednoduchosti a snadné srozumitelnosti vizualizace bude docíleno použitím malého množství prvků a zobrazení jen nezbytně nutných informací, důležitých pro účely konzultace jednotlivých podnikových procesů se zainteresovanými uživateli. Díky vizualizaci podnikových procesů budou na první pohled jasně viditelné jednotlivé vazby a návaznosti mezi kroky a odpadne nutnost stránkování v dokumentu Projekt implementace a hledání navazujících kroků a procesů.

Ovládání aplikace bude snadné a intuitivní aby cílovému uživateli bylo na první pohled jasné, jak s aplikací pracovat. Vzhled aplikace bude responzivní, aby bylo možné k aplikaci přistupovat z jakýchkoliv zařízení s připojením k internetu.

Přístup do aplikace bude umožněn pouze na základě autentizace uživatele. Uživatelé budou ověřováni přes uživatele systému FLORES a budou rozděleni do dvou uživatelských skupin, konkrétně Konzultant a Obyčejný uživatel.

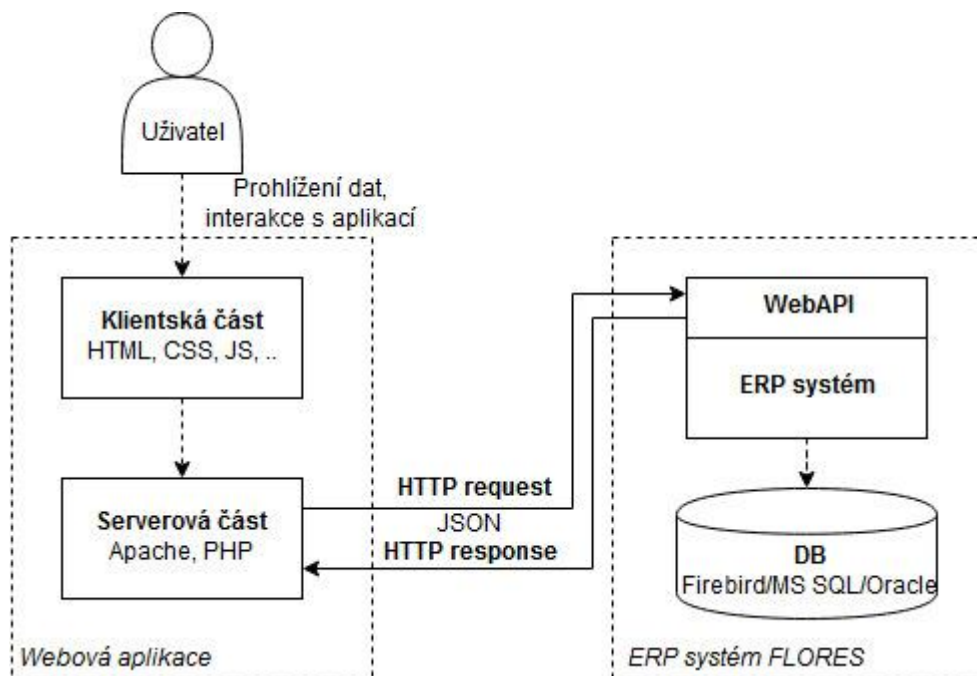
Při zadávání dat prostřednictvím aplikace bude zároveň přímo docházet ke konfiguraci ERP systému FLORES pro daného zákazníka. V systému FLORES budou tedy přímo v průběhu vytváření podnikových procesů a jejich kroků definovány jednotlivé role, stavy existujících dokladů systému FLORES a další informace. Aplikace bude zároveň umožňovat propojení jednotlivých kroků podnikových procesů přímo s agendou systému FLORES, jejichž popis bude sloužit jako návod pro firemní zaměstnance, jak vykonávat jednotlivé podnikové procesy prostřednictvím ERP systému FLORES.

6. Implementace aplikace

Na základě analýzy potřeb cílových uživatelů a definované funkcionality aplikace jsou jasně definovány požadavky, které má výsledná aplikace splňovat. Tato kapitola se věnuje návrhu architektury aplikace, definování použitých technologií a popisu implementace a funkcionality výsledné aplikace.

6.1. Použité technologie

Aplikaci lze rozdělit na webovou aplikaci, která se skládá z klientské a serverové části a na externí ERP systém FLORES, se kterým aplikace komunikuje prostřednictvím Web API, zasíláním HTTP požadavků. Obrázek 37 znázorňuje rozdělení aplikace na tyto části a jejich vzájemné propojení.



Obrázek 37 - Rozdělení aplikace na klientskou část, serverovou část a externí ERP systém FLORES

6.1.1. Klientská část

Klientská část slouží pro zobrazení výstupu uživatelům a umožňuje interakci uživatelů s aplikací. Mezi hlavní požadavky na výsledný vzhled a uživatelské rozhraní aplikace patřila především možnost používání aplikace na různých zařízeních, jednoduchý, přehledný a moderní vzhled. Výsledné barvy šablony a logo vycházejí z grafického manuálu společnosti FLORES.

V klientské části jsou použité:

Knihovny:

- Bootstrap 4⁵
- jQuery⁶
- jQuery UI⁷
- Datatables⁸
- Summernote⁹
- mxGraph¹⁰

Technologie:

- HTML5 (*Hypertext Markup Language*)
- CSS3 (*Cascading Style Sheets*)
- JavaScript (JS)
- AJAX (*Asynchronous JavaScript and XML*)
- SVG (*Scalable Vector Graphics*)

Bootstrap

Vzhled aplikace byl implementován s využitím volně dostupné open source knihovny Bootstrap. Verze verze 4.3.1, která je publikována pod MIT licencí. Tato knihovna umožňuje rychlý vývoj responzivního vzhledu webových aplikací, a to nejen díky předpřipraveným CSS třídám a definicím, ale zároveň také obsahuje JavaScriptové funkce, sloužící pro interaktivní zadávání dat, validaci vstupních polí formulářů, zobrazování dialogů a další.

Datatables

Pro zobrazení HTML tabulek a práci s daty v nich umístěných je v aplikaci použitý jQuery plug-in Datatables, verze 1.10.18, který je dostupný pod MIT licencí. Tento plug-in umožňuje přehlednou práci s tabulkami, především díky rychlému vyhledávání nad všemi daty, možností řazení jednotlivých sloupců a široké škále nastavení. Nastavení například umožňuje nastavení vyhledávání a řazení jen pro určité sloupce, přiřazení různých stylů pro jednotlivé sloupce a další.

⁵ <https://getbootstrap.com>

⁶ <https://jquery.com>

⁷ <https://jqueryui.com>

⁸ <https://datatables.net>

⁹ <https://summernote.org>

¹⁰ <https://github.com/jgraph/mxgraph>

Summernote

Pro snadné a pohodlné psaní textů slouží WYSIWYG (*What you see is what you get*) editor, který je v aplikaci implementován prostřednictvím knihovny Summernote, verze 0.8.5. Knihovna Summernote je dostupná pod MIT licenci a umožňuje snadnou implementaci a přizpůsobení na míru přesně dle potřeb konkrétní aplikace, ať už pomocí nastavení možností formátování textu, tak vytvářením vlastních ovládacích prvků a funkcí.

jQuery

Aplikační logika aplikace v klientské části je implementována v programovacím jazyku JavaScript. Pro pohodlnější a přehlednější psaní kódu je v aplikaci využita knihovna jQuery, verze 3.1.1, dostupná pod MIT licenci. Knihovna zjednodušuje práci s DOM (*Document Object Model*)¹¹ objekty a nabízí řadu funkcí, například pro obsluhu AJAX asynchronních HTTP (*Hypertext Transfer Protocol*) požadavků. Zároveň je dostupné velké množství plug-inů a nadstaveb pro tuto knihovnu, například sada funkcí jQuery UI, verze 1.12.1, obsahující funkce pro uživatelskou interakci, animace, efekty prvků a další.

mxGraph

Vizualizaci podnikových procesů umožňuje knihovna mxGraph, verze 3.9.0, umožňující vytváření a vizualizaci diagramů, včetně podpory BPMN. Knihovna je publikovaná pod licenci Apache 2.0 a pro vykreslování dat využívá standard SVG. Detailnější popis knihovny se nachází v kapitole 1.2.1.

6.1.2. Serverová část a Web API

Serverová část obsahuje aplikační logiku, řídicí funkčnost celé aplikace a obsluhu HTTP požadavků.

V serverové části jsou použité technologie:

- Apache¹²
- PHP¹³ (*PHP: Hypertext Preprocessor*)
- JSON¹⁴ (*JavaScript Object Notation*)
- REST (*Representational state transfer*)

Obsluhu jednotlivých klientských HTTP požadavků zajišťuje open source webový HTTP server Apache, uvolněný pod licenci Apache License 2.0.

¹¹ Objektová reprezentace HTML elementů, pro práci s HTML elementy prostřednictvím JavaScriptu

¹² <https://httpd.apache.org>

¹³ <https://php.net>

¹⁴ <https://www.json.org>

PHP

Serverová část aplikace je implementována s použitím skriptovacího programovacího jazyka PHP, verze 7.3. Jazyk PHP byl vybrán vzhledem k tomu, že je v současné době používán pro vývoj všech webových aplikací ve společnosti FLORES a usnadní to tedy budoucí údržbu a rozvoj této aplikace a její integraci s aplikací Interview. Jazyk PHP umožňuje vytváření dynamických webových, přičemž zajišťující zpracování klientských požadavků a vrací zpět odpověď. Při vývoji PHP aplikace je možné využít velké množství knihoven/modulů, zajišťujících například podporu různých protokolů, služících pro práci s datovými úložišti a další.

Web API

Pro komunikaci webové aplikace s ERP systémem FLORES slouží Web API, přičemž webová aplikace je klientem, který zasílá HTTP požadavky a aplikační server systému FLORES na tyto požadavky odesílá odpovědi. Rozhraní je založené na architektuře REST, která definuje čtyři metody přístupu ke zdrojům, označované zkratkou CRUD, tedy:

- **Create** (Post)
- **Retrieve** (Get)
- **Update** (Put)
- **Delete** (Delete)

Data, předávaná prostřednictvím Web API, jsou ve formátu JSON. Web API je pro aplikaci stěžejní, jelikož aplikace nemá přímý přístup k datům, uloženým v databázi. Data mohou být uložena v různých databázích, konkrétně Firebird, Microsoft SQL a Oracle, ale přístup k jednotlivým datům a práci s nimi zajišťuje výhradně ERP systém FLORES na základě zaslaných HTTP požadavků prostřednictvím rozhraní a není možné z aplikace přistupovat přímo rovnou do databáze.

6.2.Návrh architektury

Architektura aplikace vychází z návrhového vzoru MVC (*Model-view-controller*), který rozděluje aplikaci do tří logických celků [25]:

- **Model**
- **View** (pohled)
- **Controller** (kontroler)

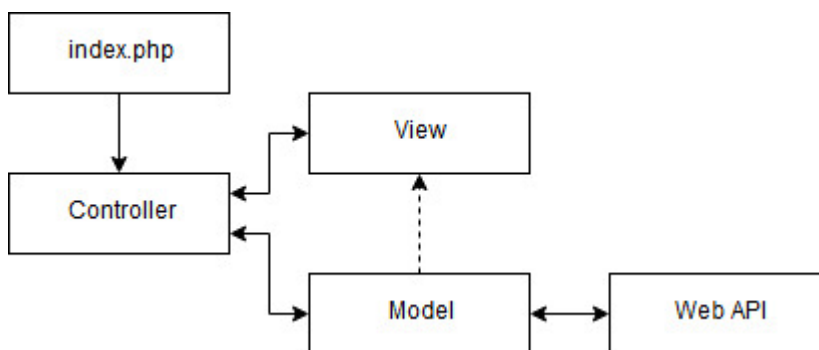
Model je část aplikace, ve které je definována veškerá business logika. Stará se tedy o přístup k datům, jejich zpracování a předávání aplikaci. Také obsluhuje komunikaci s externími aplikacemi, výpočet hodnot a vracení výsledků na základě klientských požadavků. Jedná se tedy o část kódu, která je umístěná v serverové části webové aplikace.

View reprezentuje klientskou část aplikace, tedy vše, co uživatelé vidí a s čím případně mohou interagovat. Jedná se typicky o soubory, definující jednotlivé HTML elementy, soubory obsahující kaskádové styly a také o JavaScriptový kód. Výsledná webová stránka, která se uživateli zobrazí, může být kombinací více různých souborů, definujících výsledný pohled a obsluhujících interakci uživatele s danou stránkou.

Controller slouží k propojení částí model a view a zároveň odděluje business logiku od částí, patřících do view. Jedná se o vstupní bod celé aplikace, jelikož zajišťuje obsluhu veškerých požadavků, které přijdou na server. Na základě příchozího požadavku zajistí kontroler zavolání konkrétních metod a pohledů a následně vrátí aplikaci odpověď na daný požadavek.

Mezi hlavní výhody použití návrhového vzoru patří jasně definovaná struktura, která nutí programátora rozdělit jednotlivé části kódu na vzájemně nezávislé celky, díky čemuž nedochází ke vzniku „spaghetti code“, tedy velmi obtížně udržitelného nestrukturovaného kódu. Strukturovanost kódu vede k jednodušší údržbě a přehlednosti výsledného kódu. Zároveň model MVC umožňuje rozdělení práce na jednotlivé části, které lze vyvíjet a testovat nezávisle na ostatních částech. Oddělení jednotlivých částí aplikace také umožňuje snadnější změny. Pokud by se například měl změnit nějaký externí zdroj dat za jiný, stačí malá změna pouze konkrétní třídy/funkcí, zajišťující danou funkcionalitu v části model a není potřeba hledat v celém kódu, kde všude je nutné změny provést.

Výsledný návrh architektury aplikace je zobrazený na obrázku 38. Celý proces začíná odesláním požadavku od klienta na webový server. Tento požadavek jako první zpracuje soubor *index.php*, který zajistí počáteční inicializaci aplikace a na základě vyhodnocení daného požadavku jej předá do příslušného kontroleru. Kontroler zajistí vykonání potřebné business logiky zavoláním funkcí modelu a následně inicializuje potřebné pohledy, které vrátí zpět uživateli a které mohou obsahovat data z modelu. Veškerou komunikaci s externím systémem FLORES zajišťuje model, který odesílá požadavky prostřednictvím Web API a zpracovává odpovědi.



Obrázek 38 - Architektura aplikace

6.2.1. PHP frameworky

V současné době existuje celá řada PHP frameworků, založených na architektuře MVC, které umožňují rychlý vývoj aplikací, díky využití připravených funkcí, komponent a nástrojům pro debugování a testování aplikací [26]. Mezi známé PHP frameworky patří například Yii 2¹⁵ nebo Nette¹⁶.

Hlavní výhody a přínosy použití kvalitního PHP frameworku jsou následující:

- Předpřipravené obecné funkce a komponenty, umožňující rychlý vývoj standardních, stále se opakujících aktivit, jako je například definování formulářů a zpracování jejich dat, obsluha spojení a komunikace s databázemi a další.
- Zvýšení bezpečnosti aplikace. Jednotlivé funkce jsou vyladěné, zabezpečené na základě nejlepších praktik a pravidelně aktualizované.
- Kvalitně a kompletně zpracovaná dokumentace.
- Aktivní komunita, která případně může pomoci s řešením problémů.

Nevýhody:

- Nutnost počátečního pochopení, jak daný framework funguje a načerpání zkušeností, aby byly jednotlivé komponenty a funkce používány efektivně a správně.
- Funkce jsou psané obecně, pro pokrytí všech možných scénářů a pro řešení specifických problémů na míru nejsou příliš vhodné.
- Závislost aplikace na externím kódu, nad kterým nemá uživatel žádnou kontrolu.

Používání PHP frameworků přináší pro vývoj aplikací řadu výhod a pokud by uživatel začínal s vývojem aplikace od píky, jednalo se o aplikaci větších rozměrů a nevedla by mu závislost aplikace na externím kódu, bylo by použití některého z PHP frameworků jednoznačnou volbou.

Netriviální úlohou je volba kvalitního frameworku, odpovídajícího potřebám vyvíjených aplikací. Volba nekvalitního frameworku, případně frameworku, který nemá dostatečnou komunitu vývojářů a přestane se postupem času vyvíjet a aktualizovat by vedla k problémům s údržbou již vytvořených aplikací. Po zvolení vhodného frameworku následuje nutnost počátečního pochopení, jak daný framework funguje a načerpání zkušeností, aby byly při vývoji aplikace jednotlivé komponenty a funkce používány efektivně a přesně tak, jak jsou zamýšlené. Z těchto důvodů je pro zvolení konkrétního frameworku a rozhodnutí, že bude používán pro vývoj všech firemních

¹⁵ <https://www.yiiframework.com>

¹⁶ <https://nette.org>

aplikací nutná nejen důkladná analýza dostupných možností, ale také kapacita jednotlivých zaměstnanců pro zkoumání a učení se používat daný framework. Zároveň představuje problém, pokud se má výsledná aplikace propojit s nějakou již existující aplikací, která v daném frameworku není napsaná.

Používání některého z PHP frameworků je dobrou cestou pro vývoj a udržitelnost aplikací, ale vyžaduje detailní analýzu a volnou kapacitu jednotlivých vývojářů.

Pro vývoj této aplikace nebyl žádný z PHP frameworků využitý a to především z důvodu chtěné následné integrace s aplikací Interview do jednotné aplikace, která bude využívat sadu sdílených funkcí, stejný šablonovací systém a další komponenty. Setrvání u současné architektury aplikací, používané pro webové aplikace vyvíjené společností FLORES, zároveň umožní snadnou správu a udržitelnost aplikace i jinými vývojáři ve firmě, bez nutnosti učení se pracovat s neznámým frameworkem.

6.3.Struktura aplikace

V této kapitole je popsána struktura adresářů s kódem výsledné aplikace.

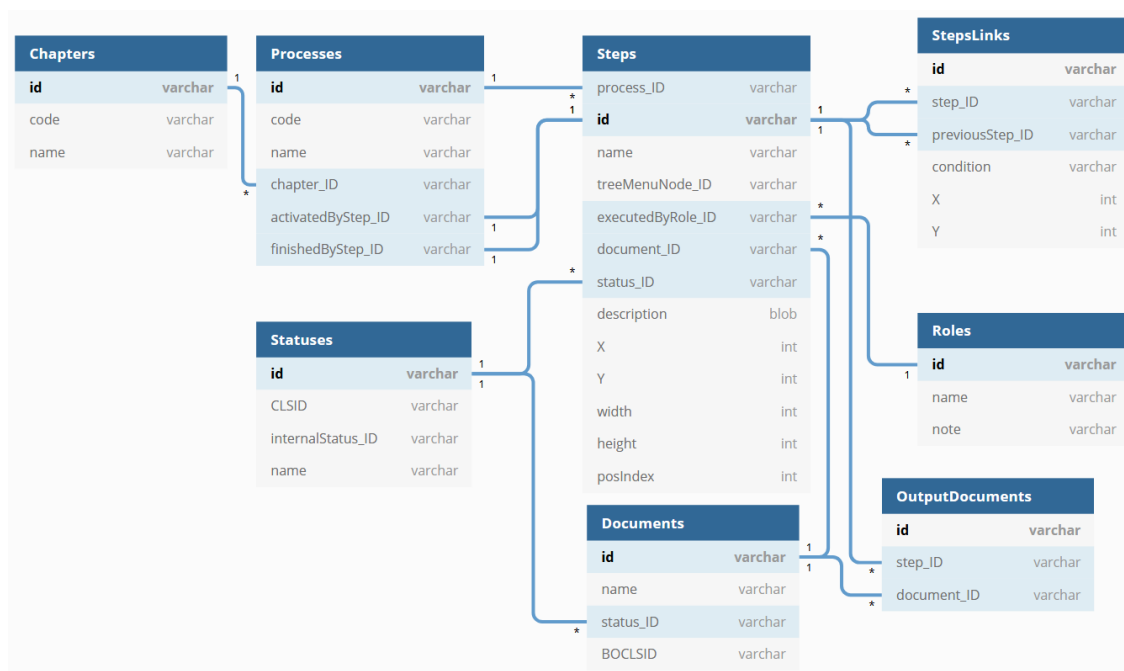
Soubory aplikace jsou rozdělené do následujících modulů:

- **Root**
 - index.php
 - .htaccess
 - favicon.ico
- **Configs**
 - Konfigurační soubory
 - Soubory s textovými řetězci pro podporované jazykové mutace
- **Controllers**
 - Kontrolery pro jednotlivé akce obsluhované aplikací
- **Core**
 - Abstraktní rodičovské třídy pro modely a kontrolery
 - Třídy PHP objektů
- **Models**
 - Modely pro vykonávání jednotlivých scénářů aplikace
 - Třída pro obsluhu komunikace s Web API
- **Views**

- Šablona aplikace
- Pohledy pro jednotlivé scénáře
 - **js**
 - JavaScriptové soubory
 - **mxGraph**
 - Soubory knihovny mxGraph
 - **css**
 - CSS soubory
 - **img**
 - obrázky používané v aplikaci
 - **font**
 - fonty používané knihovnami Bootstrap a Summernote

6.4. Struktura pro ukládání dat

Relační model databáze, vycházející ze struktury ukládání dat v systému FLORES je zobrazený na obrázku 39. Jedná se o zjednodušený model, obsahující pouze tabulky a pole, které jsou významné pro implementaci aplikace.



Obrázek 39 - Relační model databáze pro ukládání dat modelovaných procesů

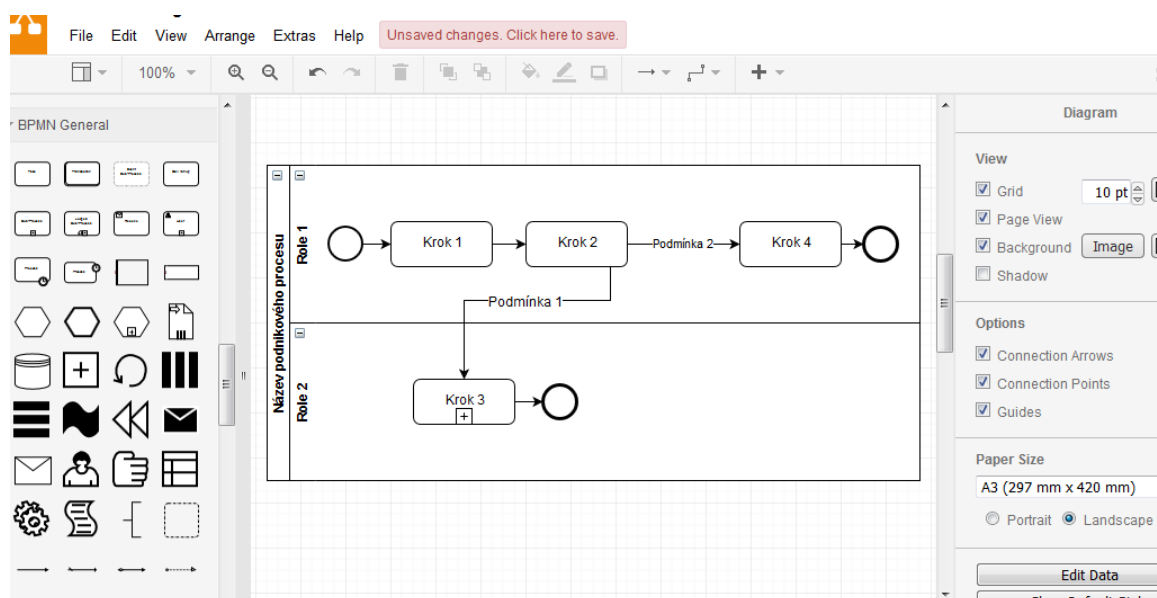
6.5. Výběr knihovny pro vizualizaci podnikových procesů

Jelikož se výsledná aplikace, která vznikne v rámci této diplomové práce, bude integrovat do již hotového interního systému postaveného na technologiích PHP a JavaScript, je tato kapitola zaměřena na porovnání existujících JS knihoven, umožňujících vytváření a vizualizaci diagramů, případně přímo podnikových procesů.

6.5.1. mxGraph

Knihovna mxGraph¹⁷ umožňuje vytváření a vizualizaci diagramů, včetně podpory BPMN [16]. Je volně dostupná pod licencí Apache 2.0 a pro vykreslování dat využívá standard SVG. Na této knihovně je postavený například online diagramový nástroj Draw.io¹⁸. Knihovna mxGraph je vyvíjená od roku 2005 a stojí za ní komunita vývojářů, kteří stále aktivně pracují na jejím vývoji.

Ukázku vizualizace podnikového procesu, vytvořenou v editoru graphEditor, jehož jádrem je knihovna mxGraph a který je společně s touto knihovnou volně dostupný, si lze prohlédnout na obrázku 40.



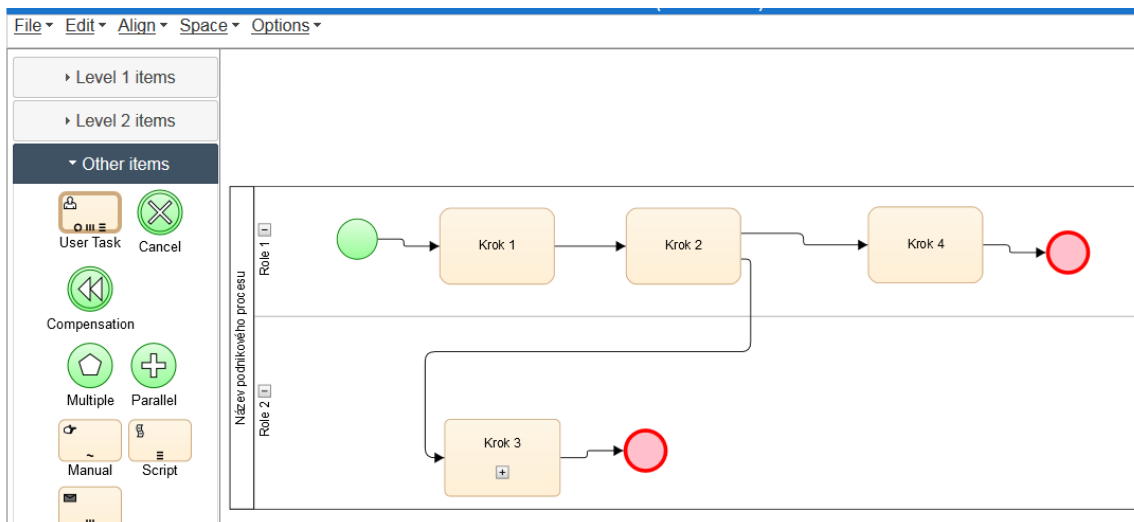
Obrázek 40 - Ukázka použití knihovny mxGraph (graphEditor)

6.5.2. GoJS

Knihovna GoJS využívá pro vizualizaci diagramů technologii HTML5 Canvas. Pro komerční použití této knihovny je nutné zakoupení licence, jejíž součástí je i podpora [17]. Knihovna podporuje BPMN. Obrázek 41 obsahuje náhled editoru a vizualizace podnikového procesu pomocí knihovny GoJS [18].

¹⁷ <https://github.com/jgraph/mxgraph>

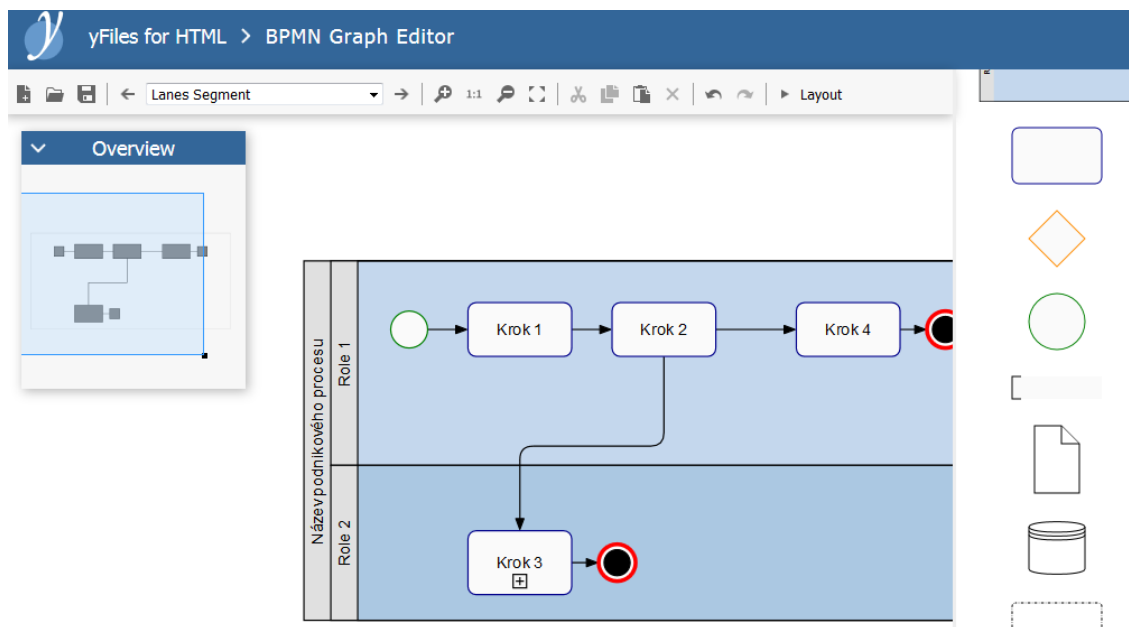
¹⁸ <https://www.draw.io>



Obrázek 41 - Ukázka použití knihovny GoJS

6.5.3. yWorks

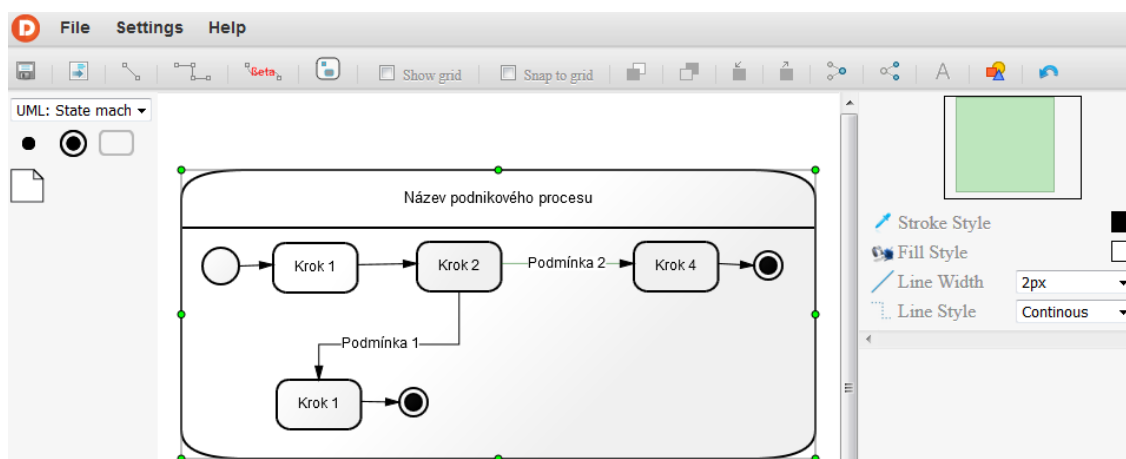
Knihovna yWorks, dostupná pod komerční licencí, nabízí propracované možnosti pro práci s celou řadou diagramů a pro jejich vizualizaci, včetně vizualizace podnikových procesů pomocí BPMN [19]. Knihovna využívá standard SVG a kromě JavaScriptové verze yFiles for HTML je dostupná také ve verzích pro programovací jazyky Java/JavaFX a .NET. Ukázka editoru a vizualizace podnikových procesů pomocí této knihovny se nachází na obrázku 42.



Obrázek 42 - Ukázka použití knihovny yWorks

6.5.4. Diagramo

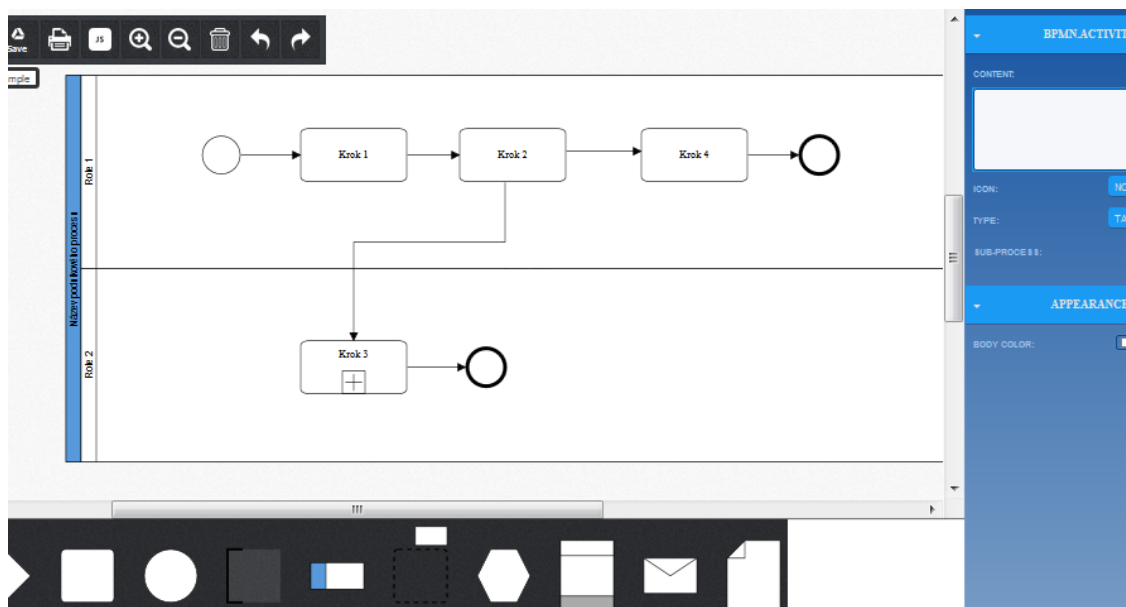
Knihovna Diagramo¹⁹ nabízí základní možnosti pro vytváření jednoduchých diagramů. Ukázku editoru Diagramo si lze prohlédnout na obrázku 43.



Obrázek 43 - Ukázka použití knihovny Diagramo

6.5.5. Rappid

Jádro knihovny Rappid²⁰ je postavené na knihovně JointJS, která je volně dostupná pod licencí Mozilla Public License Version 2.0. Editor Obrázek 44 obsahuje ukázku této knihovny.



Obrázek 44 - Ukázka použití knihovny Rappid

¹⁹ <http://diagramo.com>

²⁰ <https://www.jointjs.com>

6.5.6. Shrnutí

Tabulka 1 obsahuje souhrn základních vlastností výše uvedených knihoven, které jsou relevantní pro integraci dané knihovny do aplikace, která bude výsledkem této diplomové práce. Informace, uvedené v této tabulce jsou platné ke dni 26. 5. 2019.

	mxGraph	GoJS	yWorks	Diagramo	Rappid
Licence	Apache 2.0	Komerční ²¹	Komerční ²²	Apache 2.0	Komerční ²³
Technologie	SVG	HTML5 Canvas	SVG	HTML5 Canvas	SVG
Aktuálnost (poslední commit)	11. 3. 2019 ²⁴	23. 5. 2019 ²⁵	-	27. 5. 2014 ²⁶	21. 5. 2019 ²⁷
Četnost commitů	pravidelná	pravidelná	-	pravidelná	pravidelná
BPMN prvky	✓	✓	✓	-	✓
Tok s popiskem	✓	✓	✓	✓	-
Guidelines	✓	-	✓	-	-
Ovladatelnost	1	2	2	1	2

Tabulka 3 - Porovnání jednotlivých knihoven pro vizualizaci podnikových procesů

Knihovny jsou porovnávány na základě několika klíčových parametrů, kde stěžejním parametrem je licence, pod kterou je daná knihovna dostupná, používaná technologie (HTML5 Canvas x SVG) a samozřejmě také jednoduchost využití knihovny pro vytvoření vizualizace business procesů dle požadavků definovaných v kapitole 1.1.5.

Pro vytváření a vizualizaci diagramů jsou u webových aplikací v dnešní době používány především technologie HTML5 Canvas a SVG [22, 23].

U technologie **SVG** je každý objekt reprezentován jako samostatný DOM, díky čemuž se s danými objekty pracuje velmi snadno prostřednictvím JavaScriptu. Technologie SVG je velmi rychlá při vykreslování menšího množství objektů libovolné velikosti a naopak není vhodná pro vykreslování velkého množství objektů, při kterém dosahuje horších výsledků, než technologie HTML5 Canvas. Objekty definované pomocí SVG jsou nezávislé na rozlišení a je možná libovolná škálovatelnost jejich rozměrů.

Technologie **HTML5 Canvas** vykresluje celý výsledný obraz jako jeden objekt, díky čemuž je vhodný pro vykreslování velkého množství objektů. Při vykreslování může ale

²¹ <https://www.nwoods.com/sales/index.html>

²² <https://www.yworks.com/products/yfiles-for-html/sla>

²³ <https://www.jointjs.com/license>

²⁴ <https://github.com/jgraph/mxgraph>

²⁵ <https://github.com/NorthwoodsSoftware/GoJS>

²⁶ <https://github.com/alexgheorghiu/diagramo>

²⁷ <https://github.com/clientIO/joint>

docházet k horší kvalitě vykreslovaného textu a jednotlivé objekty při zvětšení velikosti ztrácejí kvalitu.

V případě vizualizace podnikových procesů je tedy vhodnější použití technologie SVG. Není potřeba vykreslovat tisíce objektů najednou, ale je důležitá snadná práce s objekty prostřednictvím JavaScriptu a libovolná škálovatelnost jednotlivých objektů může být při práci s modelem podnikového procesu také výhodou.

Pro výběr knihovny a budoucí rozvoj a údržbu aplikace je zároveň podstatná aktivita autorů, tedy jestli je daná knihovna stále podporována a rozvíjena o nové funkce a jsou udržovány funkce stávající. To lze vyzkoušet z četnosti commitů a data posledního commitu, pokud jsou zdrojové kódy knihovny a jednotlivé commity zveřejněné například na platformě <https://github.com>. V tabulce je zároveň řádek označující četnost, což znamená, že poslední commit není výjimečný, ale jsou v pravidelných, typicky přibližně měsíčních intervalech či častější.

Ovladatelnost a intuitivní práce s daným editorem je také velmi důležitou součástí úspěchu výsledné aplikace. Jednotlivé knihovny byly oznámkovány v rozmezí hodnot 1 - 5, přičemž 1 značí intuitivní a snadné ovládání a 5 nepoužitelnost.

Známku 1 získaly knihovny mxGraph a Diagramo, které nabízejí jednoduché a intuitivní ovládání a editor se chová přesně dle očekávání. U knihovny GoJS je hůře zpracované přesouvání uchycení jednotlivých toků. Knihovně Rappid je snížení známky zapříčiněné nutností měnit názvy jednotlivých elementů prostřednictvím postranního panelu, což není intuitivní. Důvodem pro snížení známky knihovně yWorks bylo zejména neintuitivní změna názvů prvků, kterou lze vyvolat pouze stisknutím klávesy F2 a je nutné změnu potvrdit stisknutím klávesy Enter, jinak dojde k její ztrátě.

6.6. Realizace

Tato kapitola popisuje realizaci výsledné aplikace, fungování stěžejních částí aplikace, ukázky algoritmů a částí kódu.

6.6.1. Uživatelské rozhraní aplikace

Šablona

Základním prvkem uživatelského rozhraní aplikace je soubor *template.php*, ve kterém se nachází šablona aplikace, sloužící jako kostra pro vykreslování jednotlivých stránek aplikace uživateli. Díky tomu nedochází k opakování kódu pro jednotlivé stránky aplikace a kód je snadno udržitelný.

Výpis tabulek

Pro výpis tabulek byla v rámci této práce zvolená knihovna Datatables, které uživatelům poskytuje intuitivní a snadnou možnost pro práci s tabulkami. Ukázka zobrazení tabulky s využitím této knihovny se nachází na obrázku 45.



Zobrazit 10 záznamů najednou Vyhledat:

Název	Poznámka		
Obchodník		⚙	🗑
Obsluhující		⚙	🗑
Pokladní		⚙	🗑
Pracovník	Jakýkoliv interní pracovník	⚙	🗑
Prodejce		⚙	🗑
Produktový manažer	Servisní a výrobní dokumentace, vazba na úřady ČSÚ apod.	⚙	🗑
Programátor		⚙	🗑
Reklamační technik	Posuzuje "hlášení skladu" o chybě	⚙	🗑
Název role z Interview	Popis role	+	↔
Serv. elektro		⚙	🗑

Zobrazeno 31. - 40. z celkových 59 záznamů

Předchozí 1 2 3 4 5 6 Další

Obrázek 45 - Zobrazení tabulky s využitím knihovny Datatables

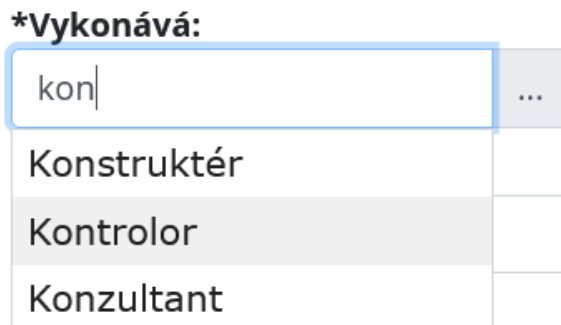
Nad daty v tabulkách je možné provádět vyhledávání, stránkování položek, řazení vybraných sloupečků a spoustu dalších akcí. Jednotlivé nastavení lze definovat nad konkrétními sloupci tabulky. Nastavení tabulky zajišťuje například následující JavaScriptový kód:

```
2 ▼ $('#chaptersTable').DataTable({
3     order: [ 1, "asc" ], //defaultní řazení dat podle 2. sloupce
4     "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, 'Vše'] ], //možnosti stránkování
5     pageSize: 10, //zobrazených záznamů na stránce
6 ▼     columnDefs: [
7         { orderable: false, targets: [2,3] }, //zakázání řazení nad danými sloupci
8         { bSearchable: false, targets: [2,3] }, //zakázání vyhledávání nad danými sloupci
9
10        //přiřazení CSS tříd konkrétním sloupcům
11        { className: "text-left-td nowrap ", "targets": [0,1] },
12        { className: " text-left-center", "targets": [2] },
13
14        { targets: [0,1], type: 'locale-compare' } //nastavení řazení textů podle české abecedy
15    ]
16 });
```

Automatické doplňování polí

Usnadnění zadávání dat, při vyplňování jednotlivých polí, umožňuje využití widgetu Autocomplete, který je součástí knihovny jQuery UI. Tento widget umožňuje automatické napovídání z definované množiny položek. Během vyplňování pole jsou tedy uživateli nabízené takové položky, které odpovídají již zadanému textu a není

nutné znát přesný název chtěné položky nebo psát pokaždé celý název ručně, viz obrázek 46.



Obrázek 46 - ukázka automatického napovídání možností

Responzivní vzhled

Responzivní vzhled aplikace je postavený na knihovně Bootstrap 4. Pro zajištění responzivního rozložení jednotlivých částí webu, využívá Bootstrap systém rozdělení obrazovky na 12 stejně velkých částí a je tedy možné definovat, jak velkou část obrazovky mají jednotlivé elementy zabírat. Zároveň je možné definovat, jakým způsobem se má rozložení a styly jednotlivých prvků šablony přizpůsobit, na základě různých velikostí obrazovky cílového zařízení. Díky tomu lze vzhled aplikace velmi dobře optimalizovat, aby výsledné zobrazení bylo přehledné na libovolných zařízeních s různou velikostí obrazovek.

CSS soubor knihovny Bootstrap je doplněný souborem *main.css*, který obsahuje vlastní styly, případně upravuje předdefinované styly do požadované podoby.

Alternativní zápis PHP kódu

Pro použití PHP kódu uvnitř HTML šablon jednotlivých stránek aplikace je využíván alternativní zápis PHP kódu²⁸, který zajišťuje vyšší přehlednost výsledného kódu a snadné pochopení i pro vývojáře, vytvářející pouze HTML šablony aplikace. Například pro vypsání dat z PHP je použitý zkrácený zápis:

```
<?=Model::$lang['visualization']?>
```

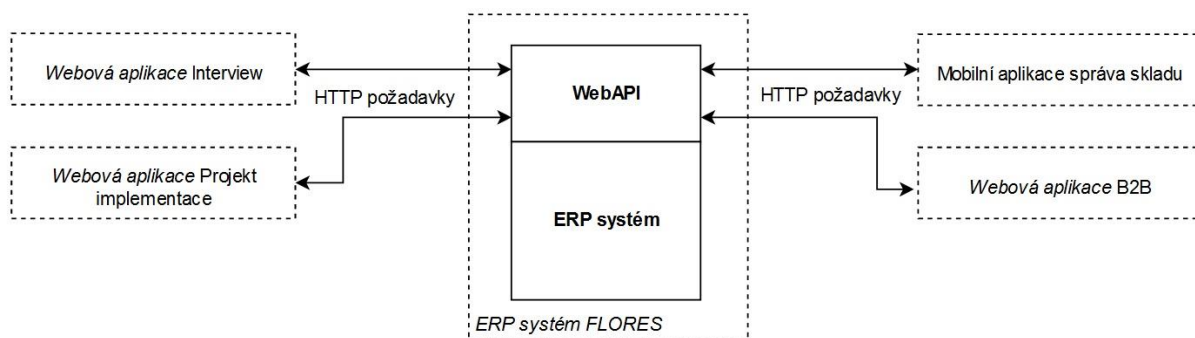
Podmínku if lze alternativním zápisem zapsat následovně:

```
<?php if ($a == 5): ?>
    Libovolný HTML kód...
<?php endif; ?>
```

²⁸ <https://www.php.net/manual/en/control-structures.alternative-syntax.php>

6.6.2. Komunikace aplikace s ERP systémem FLORES

Jak bylo již zmíněno v kapitole 6.1.2, komunikace aplikací s ERP systémem FLORES probíhá prostřednictvím Web API, založeného na architektuře REST, viz obrázek 47.



Obrázek 47 - Model komunikace aplikací s ERP systémem FLORES

Obsluhu této komunikace zajišťuje třída *API_calls*, umístěná v souboru */Models/API_calls.php*. Data předávaná prostřednictvím Web API jsou ve formátu JSON. Obsluhu jednotlivých HTTP požadavků zajišťuje PHP knihovna cURL, což značně usnadňuje práci s HTTP požadavky. Pro získání všech ID a názvů kroků konkrétního procesu ve formátu JSON stačí například pouze zavolat funkci:

```
curl_get('Steps?select=ID,Name&where=Process_ID+eq+'A011');
```

Obsluha dané funkce je následující:

```
2 private function curl_get($restAPIURL) {
3     $ch = curl_init('http://192.168.1.15:8095/DB/'.$restAPIURL);
4     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
5     curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
6     curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json',
7         'Authorization: Basic jmeno:heslo'));
8
9     $response = json_decode(curl_exec($ch));
10    $checkcURL = $this->check_cURL($ch);
11
12    if($checkcURL !== true) {
13        return $response->description; //popis chyby
14    }
15    else {
16        return $response; //data vrácená z ERP systému
17    }
18 }
```

U zasílaných HTTP požadavků je důležité zkontrolovat kód odpovědi, přičemž kódy značící úspěšné vykonání požadavku jsou uvedené níže (kód - typ HTTP požadavku):

- 200 - PUT, GET
- 201 - POST
- 204 - DELETE

Pokud je vykonání požadavku úspěšné a jsou očekávána nějaká data, jsou tato data zpracována funkcí `processResponse($data)`, která ověří, jestli byla vrácena

požadovaná data, případně jestli byla vrácená chybová hláška a data se nepodařilo získat.

6.6.3. Přihlášení uživatele

Přístup do aplikace je povolený pouze přihlášeným uživatelům. Pokud se o přístup pokusí neautentizovaný uživatel, je přesměrován na přihlašovací obrazovku. Ověření přihlášení probíhá prostřednictvím systému FLORES a Web API. Pokud volání proběhne úspěšně a vrátí zpět informace o daném uživateli, dojde k uložení informací o daném uživateli a přihlášení uživatele do aplikace. Jednotlivé informace o přihlášeném uživateli jsou uloženy v `$_SESSION`, která obsahuje instanci třídy *User*, definované v souboru */Core/User.php*.

Důležitou informací je, jestli má přihlášený uživatel přidělenou roli konzultant, či nikoliv. Na základě této role je následně určeno, jaká má daný uživatel práva a jaké akce může tedy v aplikaci vykonávat.

6.6.4. Převzetí dat z aplikace Interview a správa tagů

V kapitole 3 je popsána aplikace Interview, pomocí které jsou během analytických schůzek ukládány do systému FLORES informace o podnikových procesech, uživatelských rolích a další informace. Pro převzetí části těchto dat, se kterými je chtěné dále pracovat a použít je v aplikaci Projekt implementace, slouží obrazovky aplikace, popsané níže.

Převzetí procesů

První z těchto obrazovek je stránka s názvem „Interview -> Projekt“. Hlavním prvkem této stránky je tabulka, obsahující abecedně seřazené procesy, zadané v aplikaci Interview, viz obrázek 48. Uživatel může zaškrtnutím vybrat libovolné procesy, které chce převzít do aplikace Projekt implementace. Každý z procesů je možné přiřadit do určité kapitoly a kliknutím na tlačítko „Převzít procesy“ dojde k uložení daných procesů mezi procesy, se kterými lze pracovat v rámci aplikace Projekt implementace. Již převzaté procesy jsou označeny ikonou a nelze je vybrat znovu.

Jednotlivé kapitoly představují tematické bloky, zastřešující určitou nadřazenou agendu pro skupinu podnikových procesů. Jedná se tedy například o kapitoly Výroba, Účetnictví, Docházka atd. Pro vytváření a správu kapitol slouží obrazovka s názvem „Správa kapitol“.

Přihlášený uživatel: **Jiří Hájek** Odhlásit se

Správa kapitol Interview -> Projekt Procesy Správa tagů Vizualizace

Převzetí procesů z Interview do PI

V tabulce níže označte procesy Interview, které se mají převzít do Projektu implementace


Označit vše

Zobrazit záznamů najednou Vyhledat:

	Název procesu	↑↓	Přiřazená kapitola
✓	Proces 1		Výroba <input type="text"/>
<input type="checkbox"/>	Proces 2		Docházka <input type="text"/>
<input type="checkbox"/>	Proces 3		Účetnictví <input type="text"/>
<input type="checkbox"/>	Proces 4		Docházka <input type="text"/>

Zobrazeno 1. - 4. z celkových 4 záznamů Předchozí **1** Další

Převzít procesy



Obrázek 48 - Obrazovka pro převzetí procesů z Interview

Převzetí a správa tagů

Druhou obrazovkou, umožňující převzetí dat definovaných v rámci aplikace Interview, je stránka „Správa tagů“. Stránka byla navržena s ohledem na předpokládané větší množství různých typů tagů, které budou postupně přidávány, a umožňuje tedy práci s různými typy tagů. Pro výběr typu tagu slouží nabídka, umístěná v pravé horní části obrazovky, viz obrázek 49. Na základě zvoleného typu se přizpůsobí zbytek stránky.

Aby bylo zřejmé, jaký je stav zobrazeného tagu a odkud pochází, je na začátku každého řádku tabulky označení barvou, přičemž každá barva označuje určitý stav. Například při zvolení typu tagu Role se zobrazí tabulka s výpisem jednotlivých uživatelských rolí, definovaných jak v aplikaci Interview, tak i v aplikaci Projekt implementace. Jednotlivé role jsou odlišeny nejen barevnou značkou na začátku řádku, ale zároveň také sloupečky s povolenými akcemi, které lze nad danou rolí vykonávat. Role založené v aplikaci Interview je možné buďto propojit s existující rolí z aplikace Projekt implementace, nebo lze vytvořit kopii dané role. Aby nedocházelo k zobrazení stejné role dvakrát (pokud je daná role definována v obou aplikacích), jsou skryté všechny

role z aplikace Interview, které již byly zpracované. V případě potřeby si je ale lze zobrazit stisknutím tlačítka „Zobrazit vše“.

Role aplikace Projekt implementace jsou přímo konkrétní objekty systému FLORES a lze je editovat či odstranit. Odstranění je možné pouze v případě, pokud není daná role přiřazená k nějakému kroku procesu či aktivitě v rámci systému FLORES.

Správa tagů Vyberte typ tagu: Role

+ Vytvořit novou roli Zobrazit vše

Z Interview - přiřazen Z Interview - NEpřiřazen FLORES role

Zobrazit 10 záznamů najednou Vyhledat:

Název	Poznámka		
Obchodník		⚙️	🗑️
Obsluhující		⚙️	🗑️
Pokladní		⚙️	🗑️
Pracovník	Jakýkoliv interní pracovník	⚙️	🗑️
Prodejce		⚙️	🗑️
Produktový manažer	Servisní a výrobní dokumentace, vazba na úřady ČSÚ apod.	⚙️	🗑️
Programátor		⚙️	🗑️
Reklamační technik	Posuzuje "hlášení skladu" o chybě	⚙️	🗑️
Název role z Interview	Popis role	+	↔️
Serv. elektro		⚙️	🗑️

Zobrazeno 31. - 40. z celkových 59 záznamů

Předchozí 1 2 3 4 5 6 Další

Obrázek 49 - Obrazovka pro správu rolí

Nad jednotlivými rolemi je definována kontrola, jestli jsou všechny role z Interview zpracované a jestli jsou jednotlivé role aplikace Projekt implementace přiřazené krokům procesů. V případě, že některé z rolí dané kontrole neodpovídají, je o tom uživatel informován hláškou.

6.6.5. Správa podnikových procesů a jejich kroků

Pro správu podnikových procesů slouží obrazovka „Procesy“, obsahující výpis jednotlivých podnikových procesů a jejich kroků, viz obrázek 50.

Procesy lze filtrovat dle přiřazené kapitoly, výběrem požadované kapitoly v pravé horní části obrazovky. Zvolená kapitola se zároveň přenesou do URL, aby bylo případně možné odkazovat se rovnou pouze na procesy určité kategorie.

V tabulce s výpisem procesů jsou defaultně zobrazeny pouze řádky jednotlivých procesů. Kroky jednotlivých procesů si lze zobrazit kliknutím na řádek daného procesu,

případně pro hromadné zobrazení kroků všech procesů lze využít tlačítko „Zobrazit všechny kroky“.

Obrázek 50 - Obrazovka pro správu podnikových procesů a jejich kroků

Podnikové procesy

Pro vytvoření nového procesu slouží modální dialog, ve kterém uživatel vyplní název nového procesu, přiřadí jej do kapitoly a přidělí mu kód. Pole kód je možné nechat aplikací automaticky doplnit následujícím číslem v pořadí procesů zvolené kapitoly. Zadaný název procesu a kód musejí být unikátní, jinak nelze proces vytvořit. Řádek procesu obsahuje v pravé části funkční tlačítka, umožňující vytvoření nového kroku daného procesu, zobrazení vizualizace aktuálního procesu, jeho editaci a případně odstranění.

Kroky procesů

Po vytvoření procesu následuje definování jednotlivých kroků daného procesu. Kroky jsou definovány zadáním následujících informací:

- aktivuje/ukončuje proces
- unikátní název
- odpovědná uživatelská role
- uzel aktivního menu systému FLORES
- vstupní dokument
- výstupní dokument(y)
- předchozí krok(y)
- podproces
- popis (uživatelský návod)

Odpovědnou uživatelskou roli lze doplnit pouze z existujících rolí, definovaných v systému FLORES, případně lze kliknutím na tlačítko u daného pole vytvořit novou FLORES roli.

Uzel aktivního menu reprezentuje konkrétní záložku systému FLORES, viz obrázek 51. V levé části obrazovky jsou vidět například uzly aktivního menu pro objednávky vydané a přijaté. Přiřazení určitého uzlu aktivního menu v definici kroku procesu bude sloužit pro zobrazování nadefinovaných uživatelských návodů v popisu kroku, přímo na dané záložce v systému FLORES.

PŘEHLED OBCHODNÍCH PŘÍPADŮ

PŘED NABÍDKOU

Odkaz	Jméno firmy	Datum Ukončení	Datum Příštího kroku	RT Zboží	RT Služby	RT Ostatní	C
Otevřít	Malvia.cz s.r.o.	10.8.2016	11.7.2016	55 000	0	0	Eva Kusá

Před nabídkou celkem

Odpovědná osoba	RT Zboží	RT Služby	RT Ostatní	RT Zboží (pravděpodobnost)
Eva Kusá	55 000	0	0	22 000

NABÍDKA

Odkaz	Jméno firmy	Datum Ukončení	Datum Příštího kroku	RT Zboží	RT Služby	RT Ostatní	C
Otevřít	Alfa s.r.o.	31.3.2017	19.7.2016	350 000	10 000	567 890	Eva Kusá
Otevřít	Teleshop GmbH	28.2.2017	17.8.2016	350 000	10 000	400 000	Jaroslav Novák
Otevřít	Komodit a.s.	30.1.2017	17.8.2016	350 000	10 000	0	Jaroslav Novák
Otevřít	ABC s.r.o.	4.5.2017	5.7.2016	300 000	100 000	250 000	Karel Vysoký
Otevřít	GETA Ltd.	19.8.2016	2.8.2016	150 000	60 000	30 000	Karel Vysoký

Nabídky celkem

Odpovědná osoba	RT Zboží	RT Služby	RT Ostatní	RT Zboží (pravděpodobnost)
Eva Kusá	350 000	10 000	567 890	210 000
Jaroslav Novák	700 000	20 000	400 000	420 000
Karel Vysoký	450 000	160 000	280 000	240 000

SMLUVNÍ JEDNÁNÍ

Odkaz	Jméno firmy	Datum Ukončení	Datum Příštího kroku	RT Zboží	RT Služby	RT Ostatní	C
Otevřít	Stopro s.r.o.	25.2.2017	7.7.2016	350 000	0	567 890	Eva Kusá
Otevřít	Zetastav s.r.o.	27.4.2017	26.7.2016	350 000	10 000	567 890	Jaroslav Novák

Smluvní jednání celkem

Odpovědná osoba	RT Zboží	RT Služby	RT Ostatní	RT Zboží (pravděpodobnost)
Eva Kusá	350 000	0	567 890	280 000
Jaroslav Novák	350 000	10 000	567 890	315 000

Obrázek 51 - Ukázka výpisu uzlů aktivního menu systému FLORES

Jednou z nejdůležitějších informací, při definování kroků podnikových procesů, jsou vstupní a výstupní doklady a jejich stavy. Prostřednictvím dokladů je krok procesu propojený s objekty jádra systému FLORES, se kterými se ve výsledku v samotném systému FLORES pracuje. Stavy těchto dokladů reprezentují jednotlivé uživatelské akce v systému FLORES, přičemž každá akce vykonaná uživatelem systému FLORES znamená změnu stavu aktuálně zpracovávaného dokladu.

Doklady, přiřazené k jednotlivým krokům, lze vybírat pouze z existujících objektů jádra systému FLORES, ale je možné přiřazení vlastní názvu, pod který se s doklady

následně v aplikaci a v systému FLORES pracuje. Každý doklad musí mít přiřazený stav, který lze buďto vybrat kliknutím na některý z existujících stavů, již předdefinovaných v systému FLORES, nebo lze vytvářet stavy nové, viz obrázek 52. V dialogu pro vytváření nových stavů dokladů je zároveň zobrazená jako nápověda tabulka s výpisem stavů, které byly pro daný doklad definovány v rámci aplikace Interview. Nově vytvářené stavy dokladů se ukládají přímo do systému FLORES, kde jsou přiřazené zvolenému objektu jádra systému FLORES.

x

Vytvořit nový dokument

***Typ dokladu:**

***Vlastní název dokladu:**

Vytvořit nový stav dokladu

Vytvořit

Existující stavy:

Plánováno	9
Storno	6
V přípravě	4
Vyřízeno	8

Stavy z Interview:

Výdejka - Ke schválení
Výdejka - Schválená

Obrázek 52 - Dialog pro výběr stavu dokladu, případně pro vytvoření nového dokladu a jeho stavu

Výpis uživatelských stavů systému FLORES si lze prohlédnout na obrázku 53.

Třída	Systémový stav	Uživatelské pojmenování stavu
Výdejka	Plánováno	Plánováno
Výdejka	Storno	Storno
Výdejka	Vyřízeno	Vyřízeno
Výdejka	V přípravě	V přípravě
Vrácení pokladního výdeje	Vyřízeno	Vyřízeno
Druh nepřítomnosti	V přípravě	Nový 1 příprava
Ostatní příjem	Vyřízeno	Vyřízeno
Převodka výdej	Plánováno	Plánováno
Převodka výdej	Vyřízeno	Vyřízeno
Převodka výdej	V přípravě	V přípravě
Převodka výdej	Storno	Storno

Obrázek 53 - Ukázka výpisu uživatelských stavů v systému FLORES

Definování vzájemných vazeb mezi jednotlivými kroky umožňuje pole pro zadání předchozích kroků a případně pole pro doplnění podmínky přechodu. Ve výběru předchozích kroků lze zvolit pouze kroky daného procesu. Pro identifikaci podprocesů slouží pole podproces, do kterého lze vyplnit jakýkoliv z existujících procesů.

Popis kroku, který následně může sloužit jako uživatelský návod pro cílového uživatele systému FLORES, je možné napsat prostřednictvím WYSIWYG editoru Summernote.

Pro lepší orientaci mezi zobrazenými kroky procesu je možné přizpůsobit si pořadí vypsání jednotlivých kroků. Ke změně pořadí slouží šipky, umístěné vedle názvu kroku.

Při vytváření jednotlivých podnikových procesů, jejich kroků a následné editaci je kontrolováno a případně vynucováno dodržování definovaných pravidel, týkajících se unikátnosti některých polí, ověřování, že jsou zadané informace zadávány opravdu jen z povolených množin existujících objektů a že jsou vyplňovaná data kompletní a obsahují všechny požadované informace. Pro zadávání a editaci informací, definujících jednotlivé kroky podnikových procesů, slouží modální dialog, viz obrázek 54.

Detail kroku Odeslání objednávky

Ukončuje proces Aktivuje proces

*Název kroku: Odeslání objednávky

*Přifazený k procesu: Zpracování objednávky

*Vykonává: Logistik skladu

*Vstupní doklad: OBP - K odeslání

Výstupní doklad(y):

Navazuje na kroky:

Podmínka přechodu:

Podproces:

OBP - Vyřizeno

Faktura - Vydaná

Ověření dostupnosti ()

Nápověda

Popis kroku procesu Odeslání objednávky...

Uložit Zavřít

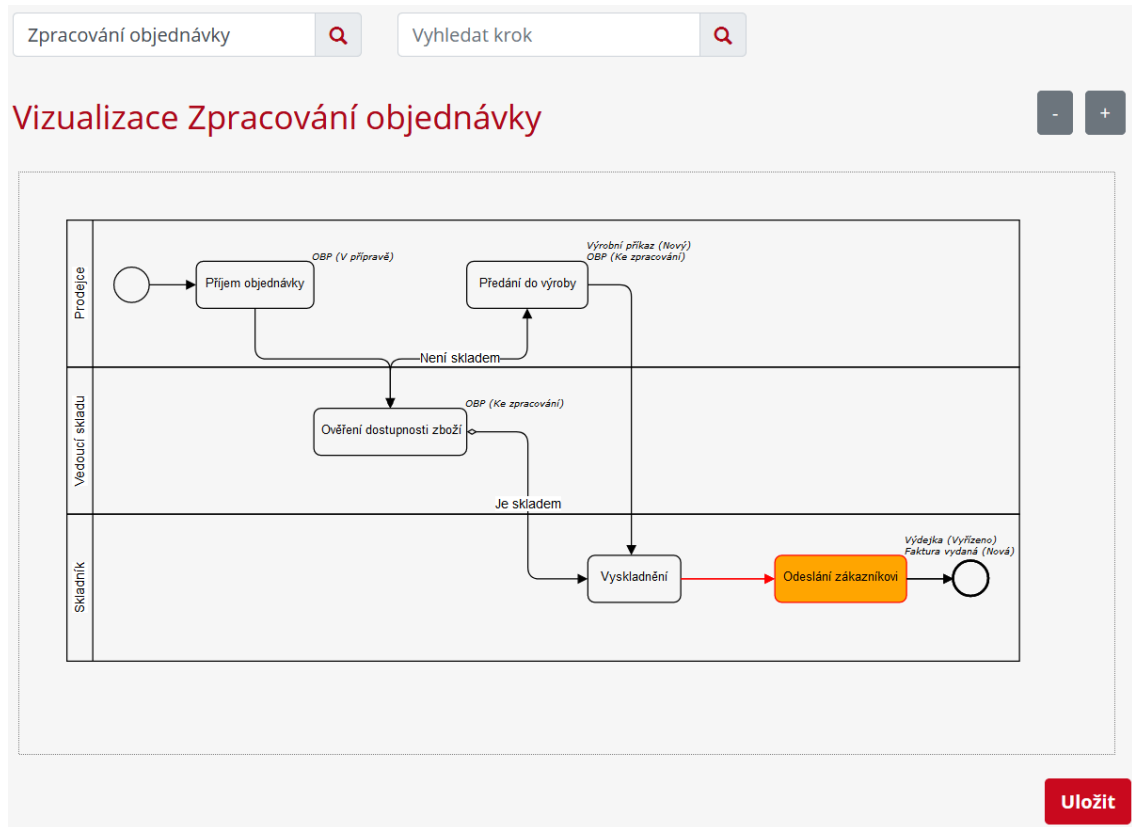
Obrázek 54 - Dialog pro vytváření a editaci kroků procesů

6.6.6. Vizualizace podnikových procesů

Pro vizualizaci podnikových procesů byla na základě analýzy, viz kapitola 6.5, vybrána knihovna mxGraph, nabízející celou škálu možností pro vizualizaci dat a práci s nimi. Popis notace, použité pro vizualizaci podnikových procesů v aplikaci, se nachází v kapitole 2.1.5. Pro potřeby aplikace bylo použité pouze základní jádro této knihovny, tedy vykreslování SVG elementů, přiřazení elementů rodičovským elementům, změna pozice a možnost práce s vazbami mezi jednotlivými elementy. Nad danou základní

funkčností jsou poté vytvářeny vlastní funkce, umožňující vykreslování vlastních SVG elementů a jejich stylů, nastavení obsluhy událostí vyvolaných uživatelem a další.

Na obrazovce s vizualizací podnikových procesů, viz obrázek 55, je možné vyhledávání podle názvů procesů a kroků.



Obrázek 55 - Ukázka obrazovky vizualizace podnikových procesů

Konfigurace a nastavení vlastností vizualizace

Konfigurace vizualizace, definující požadované vlastnosti grafu a nastavení vlastností, stylů a obsluhy událostí vykreslovaných prvků, se nachází v souboru *graphConfig.js*. Příklad definování stylu vykreslení pro prvek kroku procesu si lze prohlédnout v ukázce kódu níže:

```

60 // Step element
61 style = mxUtils.clone(style); //reset style to basic style
62 style[mxConstants.STYLE_SHAPE] = mxConstants.SHAPE_RECTANGLE;
63 style[mxConstants.STYLE_FONTSIZE] = 10;
64 style[mxConstants.STYLE_ROUNDED] = true;
65 style[mxConstants.STYLE_HORIZONTAL] = true;
66 style[mxConstants.STYLE_VERTICAL_ALIGN] = 'middle';
67 delete style[mxConstants.STYLE_STARTSIZE];
68 style[mxConstants.STYLE_LABEL_BACKGROUNDCOLOR] = 'none';
69 graph.getStylesheet().putCellStyle('process', style);

```

Vykreslení vlastního tlačítka, definovaného pro konkrétní skupinu prvků vizualizace a nastavení jeho funkčnosti zajišťuje následující část kódu:

```
235 ▾ /*****
236 * Delete button
237 *****/
238 var deleteIcon = createImage('img/delete.png');
239 deleteIcon.setAttribute('title', 'Odstranit');
240 deleteIcon.style.cursor = 'pointer';
241 deleteIcon.style.width = '16px';
242 deleteIcon.style.height = '16px';
243
244 ▾ mxEvent.addGestureListeners(deleteIcon, mxUtils.bind(this, function(evt) {
245     mxEvent.consume(evt); // Disables dragging the image
246 }));
247
248 ▾ mxEvent.addListener(deleteIcon, 'click', mxUtils.bind(this, function(evt) {
249     /* Confirmation dialog */
250     var confirmation = confirm("Opravdu chcete odstranit krok " + steps[this.state.cell.id].getName() + "?");
251
252     if(confirmation) {
253         /* On confirm try to remove selected step from DB */
254         removeStep(this.state.cell.id, steps[this.state.cell.id].getName(), this);
255     }
256     mxEvent.consume(evt);
257 }));
258
259 this.domNode.appendChild(deleteIcon);
260 this.graph.container.appendChild(this.domNode);
261 this.redrawTools();
262
```

Inicializaci výsledného grafu vykonává funkce `main(DOM_container)`, umístěná v souboru `prepare_graph.php`. Tato funkce nejdříve zavolá načtení všech požadovaných vlastností grafu ze souboru `graphConfig.js`, následované načtením jednotlivých prvků pro aktuálně vybraný podnikový proces ze systému FLORES. Načtená data jsou zkontrolována, jestli obsahují všechny důležité informace pro zobrazení vizualizace a pokud ano, spustí se algoritmus pro vykreslení jednotlivých prvků vizualizace procesu.

Prvky typu krok procesu jsou reprezentovány JS třídou `Step (js/Step.js)`, jenž obsahuje veškeré informace, potřebné pro vizualizaci kroku a s ním souvisejících prvků. Informace o prvcích typu bazénová dráha jsou reprezentovány třídou `Pool (js/Pool.js)`.

Vykreslení vizualizace podnikového procesu

Pokud je vizualizace podnikového procesu zobrazována poprvé a ještě nejsou v databázi uloženy žádné informace o rozměrech či souřadnicích umístění jednotlivých prvků, je nutné zajistit automatické vykreslení a rozmístění daných prvků. Automatické vykreslení vizualizace vede k usnadnění práce koncového uživatele, jelikož pro vykreslení vizualizace, na základě nedefinovaných procesů a jejich kroků, není nutná žádná uživatelská akce. Uživatel samozřejmě může kdykoliv vykreslení prvků upravit dle svých požadavků, případně doladit nedokonalosti vzniklé automatickým rozmístěním prvků.

Vizualizace vynucuje definování počátečního kroku procesu. Pokud není žádný krok nastavený jako počáteční, při pokusu o zobrazení vizualizace se uživateli zobrazí dialog, informující o dané skutečnosti a obsahující možnost výběru počátečního kroku z existujících kroků, případně vytvoření nového počátečního kroku. V opačném případě dojde ke spuštění algoritmu, zajišťujícího postupné vykreslení všech prvků vizualizace. Zjednodušený zápis tohoto algoritmu si lze prohlédnout v ukázce níže:

```

1  model.beginUpdate(); //start graph update
2
3  try {
4      // Draw processe's first step
5      drawStep(parentElement, null, firstStepRole, firstStepObject, '');
6
7      // If current step was not already visited, insert into @queue
8      if(!visited.includes(firstStepObject)) {
9          queue.push(firstStepObject);
10     }
11
12     // Draw all next steps connected with current step
13     while(queue.length) {
14         currentStep = steps[queue.shift()]; // Pop step from queue
15
16         currentRole = currentStep.getExecutedByRole(); // Current step role
17         nextSteps = currentStep.getNextSteps(); // All next steps starting from current step
18
19         visited.push(currentStep); //add current step into array of visited steps
20
21         // Draw all next steps for current step
22         for(var i = 0; i < nextSteps.length; i++) {
23
24             //If next step is in same pool as current step, add condition length to X coordinate
25             if(currentRole == nextSteps[i].getRole()) {
26                 cX += nextSteps[i].condition.length;
27             }
28
29             //Draw step into graph
30             drawStep(parentElement, currentStep, currentRole, nextSteps[i], nextSteps[i].condition);
31         }
32     }
33 }
34 finally {
35     model.endUpdate(); //finish graph update
36 }

```

Algoritmus začíná vykreslením prvního kroku procesu a jeho vložením do fronty kroků ke zpracování. Následuje while cyklus, který postupně vybírá jednotlivé kroky z fronty, přičemž vykreslí všechny následující kroky pro aktuálně vybraný krok. Pokud následující krok ještě nebyl algoritmem zpracován, je po vykreslení přidán do fronty ke zpracování.

V ideálním případě, tedy pokud jsou nadefinovány všechny vazby mezi jednotlivými kroky a jednotlivé kroky na sebe vzájemně navazují, by byl tento algoritmus dostačující pro kompletní vykreslení celého grafu. Samozřejmě ale může nastat situace, že je některá z vazeb přerušena, případně daný krok zatím nemá vůbec žádnou vazbu definovanou. Pro takové případy slouží následně algoritmus, který projde všechny zatím nezpracované kroky a zajistí jejich vykreslení, přičemž přednost mají kroky, které jsou považovány za nové počátky toku, tedy které nemají přiřazené žádné předchozí kroky, ale pouze kroky následující, viz ukázka na následující stránce.

```

1  var notConnected = []; // Steps without any connection
2
3  //Draw all steps which are not visited yet
4  Object.keys(steps).forEach(function(step, index) {
5
6      if(visited.includes(this[step])) {
7          return; //skip all already visited steps
8      }
9
10     // At first draw all steps which have any next step and none previous
11     if(this[step].getPrevious().length == 0 && this[step].getNext().length > 0) {
12         drawStep(parentElement, null, this[step].getRole(), this[step], '');
13
14         // If current step was not already visited, insert into @queue
15         if(!visited.includes(this[step])) {
16             queue.push(this[step]);
17         }
18
19         // While cycle for draw all next steps connected to current step
20         drawNextSteps(parentElement);
21     }
22     // Add step into array of steps without connection
23     else {
24         notConnected.push(this[step]);
25     }
26 }, steps);
27
28 // Finally draw all steps without any connection
29 for(var i = 0; i < notConnected.length; i++) {
30     drawStep(parentElement, null, notConnected[i].getRole(), notConnected[i], '');
31 }

```

Jednotlivé části algoritmu pro vykreslení vizualizace jsou dekomponovány do samostatných funkcí, zajišťujících provedení dílčích částí vykreslení. Každý prvek má definovaný rodičovský prvek, kterým je v případě většiny prvků bazénová dráha, pouze v případě samotných bazénových drah je to kořenový prvek grafu, kterým je prvek typu bazén. Cyklus while pro obsluhu všech kroků vložených do fronty, se nachází ve funkci:

`drawNextSteps (parent).`

Vykreslování jednotlivých kroků a souvisejících elementů obstarává funkce:

`drawStep(parent, current_step, role_next, next_step, condition).`

Funkce pro vykreslení kroku následně případně volá další potřebné funkce, jako například:

`drawConnection(pool, condition, current_step, next_step, color),`

zajišťující vykreslení vzájemných vazeb mezi kroky. Barva vazby je určena na základě kontroly návaznosti výstupního dokladu aktuálního kroku a vstupního dokladu kroku následujícího. Funkce zároveň ověřuje, jestli se má vykreslit podmíněný směr toku (v případě, že je definována podmínka pro přechod mezi kroky), nebo obyčejný směr toku. Funkce `drawSubProcessIcon (ID)` umožňuje vykreslování podprocesů.

Při vykreslování aktuálně zpracovávaného kroku dochází zároveň k následujícím akcím:

- Vykreslení počátečního/koncového prvku, pokud je proces zahájen či ukončen aktuálně zpracovávaným krokem.

- Kontrola existence rodičovského elementu (bazénové dráhy) pro daný krok, případně vykreslení potřebné bazénové dráhy, definované dle přiřazené zodpovědné role na aktuálním kroku.
- Kontrola existence vazby mezi všemi kroky
- Kontrola návaznosti výstupního dokumentu z aktuálního kroku a vstupního dokumentu v kroku navazujícím.
- Ověření, jestli má krok již definované souřadnice určující jeho polohu. Pokud nemá, určení polohy kroku probíhá posunutím X-ové souřadnice předchozího kroku o šířku daného kroku a přičtení konstanty délky směru toku. V případě, že přechod mezi danými kroky obsahuje podmínku a dané kroky jsou v totožné bazénové dráze, je k výsledné X-ové souřadnici zároveň přičtena délka textu dané podmínky.

Práce s vizualizací podnikových procesů

Ve výsledné vizualizaci procesů má uživatel možnost měnit uspořádání jednotlivých kroků, jejich rozměrů, vzájemné vazby a přiřazenou bazénovou dráhu. Pro zkvalitnění práce s vizualizací jsou při přesouvání jednotlivých prvků zobrazovány vodící čáry, zobrazující aktuální polohu vzhledem k ostatním prvkům vizualizace. Prostřednictvím tlačítek, umístěných na jednotlivých krocích, je možné zobrazení dialogu s detailními informacemi o daném kroku, případně smazání daného kroku a s ním souvisejících dat. V pravé horní části obrazovky se nachází tlačítka, nabízející možnost přiblížení a oddálení celé vizualizace podnikového procesu.

7. Testování vytvořené aplikace

Tato kapitola popisuje způsob testování výsledné aplikace, pro ověření její funkčnosti. Aplikace byla testována prostřednictvím testovacích scénářů, manuálně prováděných uživateli a pomocí automatických testů, vytvořených prostřednictvím nástroje Selenium²⁹.

7.1. Testovacích scénáře

První částí testování jsou o manuální testy, sloužící pro kontrolu zobrazení a optimalizaci uživatelského rozhraní a ovládání aplikace. Testující uživatelé vykonávají v aplikaci sled akcí, dle předem definovaného scénáře, popisujícího jednotlivé úkony. Popis testovacího scénáře, na základě kterého probíhalo testování aplikace je uvedený níže:

- 1) Přihlášení se do aplikace zadáním přiděleného uživatelského jména a hesla
- 2) Vytvoření nové kapitoly, její editace a odstranění
- 3) Převzetí procesů „Plánování výroby“ a „Zpracování objednávky“, definovaných v rámci aplikace Interview do aplikace Projekt implementace a jejich přiřazení do nadřazených kapitol
- 4) Převzetí role „Obchodní zástupce“, definované v aplikaci Interview do aplikace Projekt implementace
- 5) Vytvoření nového podnikového procesu
- 6) Pokus o přidání kroku procesu s nepovolenou rolí, následně s neúplnými daty
- 7) Vytvoření správně definovaných kroků procesu a jejich vzájemných vazeb
- 8) V daném procesu zároveň vytvořit krok bez jakékoliv vazby na ostatní kroky a krok porušující podmínku pro změnu stavu dokladu
- 9) Zobrazení vizualizace nadefinovaného kroku
- 10) Oprava nahlášených varování, aby byla výsledná vizualizace podnikového procesu zobrazená korektně bez varování
- 11) Změna rozložení jednotlivých kroků v aplikaci a ověření uložení daných změn

²⁹ <https://www.seleniumhq.org>

7.2. Automatické testy

Nástroj Selenium umožňuje snadné vytváření automatických testů webových aplikací v různých webových prohlížečích. Díky tomu není nutné při každé úpravě aplikace manuálně testovat jednotlivé testovací scénáře a umožňuje rychlou kontrolu stěžejních funkcí aplikace. Pro vykonání testů je nutné propojení nástroje s ovladačem vykreslovacího jádra konkrétního webového prohlížeče. Jednotlivé testy jsou psány v programovacím jazyku Java.

Automatický test obsahuje následující testovací scénář:

- 1) Přihlášení uživatele do aplikace
- 2) Vytvoření nového procesu
- 3) Přidání kroku k danému procesu
- 4) Odstranění kroku
- 5) Odstranění procesu
- 6) Základní test vizualizace
 - a. Zobrazení procesu ve vizualizaci
 - b. Zobrazení dvou kroků, jejichž vazba nesplňuje podmínku a shodu výstupního dokladu kroku se vstupním dokladem kroku následujícího
 - c. Oprava vstupního dokladu kroku
 - d. Změna přiřazení role přiřazené ke kroku, přesunutím kroku do jiné bazénové dráhy
 - e. Uložení a kontrola, jestli došlo ke změně role daného kroku

V rámci výše zmíněného testování byla odhalena řada chyb a nedostatků aplikace, které byly následně postupně opravovány. Zároveň na základě uživatelského testování vyplynulo množství užitečných námětů, pro zlepšení přehlednosti a ergonomie ovládání aplikace.

Kromě výše uvedených testů je také důležité ověření validity jednotlivých HTML stránek, generovaných aplikací, prostřednictvím nástroje dostupného na adrese <https://validator.w3.org>. Nevalidní HTML vede k horší přístupnosti dané aplikace a zároveň může způsobovat chyby ve funkčnosti aplikace.

8. Závěr

Cílem této práce byl návrh a vytvoření aplikace, umožňující modelování a vizualizaci podnikových procesů, která bude používána v rámci analýzy podniku před implementací ERP systému FLORES. Před začátkem vytváření samotné aplikace bylo nutné provést analýzu možností modelování a vizualizace podnikových procesů a existujících nástrojů v této oblasti. Zároveň bylo stěžejní porozumět stávajícímu způsobu popisu podnikových procesů konzultanty společnosti FLORES Software a chtěným zlepšením stávajícího způsobu. Znalosti, získané v úvodní části práce, byly aplikovány při návrhu a vývoji výsledné aplikace. Z existujících notací pro vizualizaci podnikových procesů, byla vybrána notace BPMN a byla definována funkčnost aplikace, odpovídající požadavkům zadavatele.

Výsledná aplikace, vytvořená v rámci této práce, je připravená pro nahrazení současného nevyhovujícího způsobu popisu podnikových procesů v obyčejném textovém editoru.

Oproti současnému způsobu popisu dat přináší aplikace přehlednější a uživatelsky přívětivější způsob modelování a vizualizace podnikových procesů, srozumitelné i obyčejným uživatelům. Aplikace umožňuje automatické zpracování definovaných dat, kontrolu jejich úplnosti a konzistence. Zároveň nasazení této aplikace značně usnadní modelování jednotlivých podnikových procesů, díky propojení s již existujícími daty, uloženými v systému FLORES.

Kromě samotné definice požadavků, na konfiguraci systému FLORES, je část dat definovaných v aplikaci zároveň automaticky použita pro přímou konfiguraci systému FLORES. Jedná se především o uživatelské role, nastavení změn stavů jednotlivých dokladů (objektů) systému FLORES a možnost propojení popisu jednotlivých kroků s konkrétními agendami tohoto systému.

Tato práce splňuje zadání v celém rozsahu a výsledná aplikace splňuje požadavky a očekávání zadavatele práce.

Funkčnost výsledné aplikace bude dále rozvíjena, především co se týče rozšíření možností práce s vizualizací podnikových procesů a analýzy definovaných dat. Analýza a vizualizace dat má mnohem větší potenciál a tato funkcionalita bude dále rozvíjena. Je plánováno propojení výsledné aplikace s již existující aplikací Interview do jedné aplikace. Zároveň je do budoucna plánována možnost importu předpřipravených vzorových procesů a jejich kroků do aplikace, díky čemuž bude možné importovat celé celky popisující obecné podnikové procesy, čímž se ušetří práce konzultanta při zadávání standardních podnikových procesů.

Seznam zkratek

AJAX, *Asynchronous JavaScript and XML* - obecné označení sady technik, umožňujících pomocí asynchronních operací měnit obsah stránky bez nutnosti znovunačtení celé dané stránky.

ARIS, *Architecture of Integrated Information System* - nástroj využívající Event-driven Process Chain pro modelování podnikových procesů.

BPMN, *Business Process Modeling Notation* - modelovací notace, která je velmi rozšířeně využívána pro vizualizaci podnikových procesů.

CRUD, *Create, Read, Update, Delete* - označení čtyř základních operací pro práci s daty, konkrétně vytváření, čtení, aktualizace a mazání

CSS, *Cascading Style Sheets* - jazyk pro popis stylu zobrazení HTML elementů

DOM, *Document Object Model* - stromový model pro reprezentaci jednotlivých objektů HTML stránky, vytvářený webovým prohlížečem.

EPC, *Event-driven Process Chain* - modelovací jazyk pro popis návaznosti (řetězení) jednotlivých událostí a aktivit v rámci podnikového procesu.

ERP, *Enterprise Resource Planning* - komplexní software pro řízení a integraci všech firemních procesů, jako například účetnictví, evidenci docházky, zpracování mezd, personalistiku, plánování pracovních cest a další.

HTML5, *Hypertext Markup Language (verze 5)* - standardní značkovací jazyk pro webové stránky.

HTTP, *Hypertext Transfer Protocol* - protokol pro komunikaci mezi klientem a serverem

JS, *JavaScript* - skriptovací jazyk

JSON, *JavaScript Object Notation* - formát pro ukládání a výměnu dat.

MVC, *Model-view-controller* - návrhový vzor, definující rozdělení aplikace do tří nezávislých celků, konkrétně na model, pohled a kontroler

REST, *Representational state transfer* - architektura definující sadu omezení, pro vytváření webových služeb.

OMG, *Object Management Group* - mezinárodní konsorcium, zabývající se vývojem technologických standardů

PHP, *PHP: Hypertext Preprocessor* - skriptovací jazyk

SVG, *Scalable Vector Graphics* - formát pro definování vektorové grafiky

UML, *Unified Modeling Language* - jednotný modelovací jazyk pro analýzu, návrh a dokumentaci softwarových systémů

WS-BPEL, *Web Services Business Process Execution Language* - jazyk pro jednoznačný popis a automatizaci podnikových procesů, běžně se lze setkat se zkráceným zápisem BPEL

Použitá literatura

[1] sai.padmanabhan, *What Actually is a Business Process?* [online], [cit. 20. 10. 2017]. Dostupné z: <<https://kissflow.com/bpm/business-process>>.

[2] *Types of processes in an organization* [online], [cit. 20. 10. 2017]. Dostupné z: <<https://www.synergycom.in/Types-of-processes.html>>.

[3] Cyril Klimeš, *MODELOVÁNÍ PODNIKOVÝCH PROCESŮ* [online], [cit. 31. 10. 2017]. Dostupné z: <<http://www1.osu.cz/~zacek/mopop/mopop.pdf>>.

[4] Nishadha, *Business Process Modeling Tutorial (BPM Guide Explaining Features)* [online], [cit. 1. 11. 2017]. Dostupné z: <<http://creately.com/blog/diagrams/business-process-modeling-tutorial>>.

[5] *Beginners Guide to Business Process Modeling* [online], [cit. 1. 11. 2017]. Dostupné z: <<https://www.smartsheet.com/beginners-guide-business-process-modeling>>.

[6] Stephen A. White, *Introduction to BPMN* [online], [cit. 21. 10. 2017]. Dostupné z: <http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf>.

[7] *Introduction to BPMN Part I* [online], [cit. 21. 10. 2017]. Dostupné z: <<https://www.visual-paradigm.com/tutorials/bpmn1.jsp>>.

[8] *About the Unified Modeling Language Specification Version 2.5* [online], [cit. 27. 10. 2017]. Dostupné z: <<http://www.omg.org/spec/UML>>.

[9] GEAMBAŞU, Cristina Venera. *BPMN vs UML activity diagram for business process modeling*, 2012, 11.4: 637-651.

[10] Prof. Ing. Ivo Vondrák, CSc., *METODY BYZNYS MODELOVÁNÍ* [online], [cit. 30. 10. 2017]. Dostupné z: <http://vondrak.cs.vsb.cz/download/Metody_byznys_modelovani.pdf>.

[11] Schmidt, J. W., and Axel Wienberg, *A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes* [online], [cit. 29. 9. 2018]. Dostupné z: <<https://pdfs.semanticscholar.org/884e/374af5c4e8cc1a68aa42a5c228191af48385.pdf>>.

[12] *Activity Diagram* [online], [cit. 15. 9. 2018]. Dostupné z: <<https://www.uml-diagrams.org/activity-diagrams-reference.html>>.

[13] *Activity Diagrams* [online], [cit. 30. 10. 2017]. Dostupné z: <<https://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams>>.

- [14] Marco Fagnoli, *EPC vs BPMN: Reviewing Modelling Notations* [online], [cit. 31. 10. 2017]. Dostupné z: <<http://blog.leonardo.com.au/epc-vs-bpmn-reviewing-modelling-notations>>.
- [15] TSCHESCHNER, Willi. *Transformation from EPC to BPMN. Business Process Technology*, 2006, 1.3: 7-21.
- [16] *mxGraph* [online], [cit. 14. 10. 2018]. Dostupné z: <<https://github.com/jgraph/mxgraph>>
- [17] *NorthWood Software Pricing* [online], [cit. 14. 10. 2018]. Dostupné z: <<https://www.nwoods.com/sales/index.html>>
- [18] *GoJS demo* [online], [cit. 14. 10. 2018]. Dostupné z: <<https://gojs.net/latest/extensions/BPMN.html>>
- [19] *yFiles for HTM: LBPMN Graph Editor* [online], [cit. 15. 10. 2018]. Dostupné z: <<https://live.yworks.com/demos/complete/bpmn/index.html>>
- [20] *Diagramo demo* [online], [cit. 16. 10. 2018]. Dostupné z: <<http://diagramo.com/editor/editor.php>>
- [21] *Rappid demo* [online], [cit. 16. 10. 2018]. Dostupné z: <<http://resources.jointjs.com/demos/bpmn>>
- [22] *HTML5 Canvas vs. SVG vs. div* [online], [cit. 9. 7. 2018]. Dostupné z: <<https://stackoverflow.com/questions/5882716/html5-canvas-vs-svg-vs-div>>
- [23] Maria Antonietta Perna, *Canvas vs. SVG: Choosing the Right Tool for the Job* [online], [cit. 10. 7. 2018]. Dostupné z: <<https://www.sitepoint.com/canvas-vs-svg-choosing-the-right-tool-for-the-job>>
- [24] Gregor Polančič, *Understanding BPMN Connections* [online], [cit. 3. 2. 2019]. Dostupné z: <http://www3.cis.gsu.edu/dtruex/courses/CIS4120/Sessions/Session%203%20-%20Level%201%20BPMN/Introduction_to_BPMN/Understanding-the-bpmn-connections-Orbus.pdf>
- [25] PITT, Chris, *Pro PHP MVC*, 2012, ISBN 978-1-4302-4165-2
- [26] Jarkko Laine, *Should You Use a PHP Framework? Five Pros and Cons* [online], [cit. 9. 6. 2018]. Dostupné z: <<https://code.tutsplus.com/tutorials/should-you-use-a-php-framework-five-pros-and-cons--cms-28905>>

Přílohy

A Uživatelská příručka

Spuštění aplikace

Pro spuštění aplikace je nutný HTTP server Apache, verze 2.4, s povoleným modulem cURL, povoleným používáním .htaccess souboru a povoleným přepisováním URL pomocí mod_rewrite. Na daném serveru musí být zároveň nainstalované PHP, verze 7.1 a vyšší.

Zároveň je nutné mít zprovozněný a nakonfigurovaný systém FLORES a spuštěný Web API server, pro obsluhu komunikace se systémem FLORES. Samotný Web API server ani systém FLORES nejsou součástí odevzdávané práce a pro potřeby spuštění aplikace je tedy nutné domluvit se na zprovoznění testovacího prostředí s autorem práce, případně na osobním předvedení aplikace a zpřístupnění zařízení s nainstalovaným systémem FLORES a Web API serverem.

Obsluha aplikace

Přihlášení uživatele do aplikace probíhá zadáním uživatelského jména a hesla, kterými se uživatel přihlašuje do systému FLORES.

Po úspěšném přihlášení dojde k přesměrování na záložku s přehledem podnikových procesů a jejich kroků, viz obrázek 56. Pro navigaci mezi jednotlivými obrazovkami aplikace slouží navigační tlačítka, umístěná v panelu v horní části obrazovky.

Typický postup práce s aplikací je následující:

- 1) Definování kapitol jednotlivých podnikových procesů, prostřednictvím záložky „Správa kapitol“.
- 2) Převzetí dat, definovaných v rámci aplikace Interview, prostřednictvím záložek „Interview -> Projekt“ a „Správa tagů“.
- 3) Definování jednotlivých podnikových procesů, jejich kroků a potřebných dat prostřednictvím záložky „Procesy“.
- 4) Vizualizace podnikových procesů prostřednictvím záložky „Vizualizace“.
- 5) Správa definovaných dat, prostřednictvím záložek správa tagů, „Procesy“ a „Vizualizace“.

Název	Vykonává	Typ (stav) vstupního dokladu	Typ (stav) výstupního dokladu
- Přijem objednávky ▼	Prodejce	OBP (Nová)	OBP (V přípravě)
- Odeslání zákazníkovi ▲ ▼	Skladník	Výdejka (Plánováno)	Výdejka (Vyřizeno) Faktura vydaná (Nová)
- Ověření dostupnosti zboží ▲ ▼	Vedoucí skladu	OBP (V přípravě)	OBP (Ke zpracování)
- Předání do výroby ▲ ▼	Prodejce	OBP (Ke zpracování)	Výrobní příkaz (Nový)
- Vyskladnění ▲	Skladník	OBP (Ke zpracování)	Výdejka (Plánováno)

Obrázek 56 - Obrazovka s přehledem podnikových procesů a jejich kroků

Detail kroku procesu si lze zobrazit kliknutím na název daného kroku. Šipky u názvu kroku slouží pro změnu aktuální pozice zobrazení kroku v rámci procesu. Nové kroky lze přidávat prostřednictvím tlačítka s ikonou +, nacházejících se v každém řádku jednotlivých procesů.

Prostřednictvím pole „Vyhledat“ je možné prohledávat všechny existující podnikové procesy a jejich kroky a obsah tabulky se automaticky omezí pouze na záznamy, obsahující hledaný výraz.

Záložka vizualizace nabízí možnost zobrazení vizualizace definovaných podnikových procesů. Na této záložce je možné vyhledávání zadáním názvu procesu či kroku. Ve výsledné vizualizaci je možné měnit uspořádání jednotlivých kroků, jejich rozměrů, vzájemné vazby a přiřazenou bazénovou dráhu.

Pomocí tlačítek, umístěných na jednotlivých krocích, je možné zobrazení dialogu s detailními informacemi o daném kroku, případně smazání daného kroku a s ním souvisejících dat. V pravé horní části obrazovky se nachází tlačítka, nabízející možnost přiblížení a oddálení celé vizualizace podnikového procesu.

B Obsah přiloženého média

Součástí této práce je přiložené CD, obsahující následující soubory a adresáře:

- *Application/*
- adresář obsahující veškeré zdrojové soubory aplikace, použitých knihoven a související data
- *Testing/*
- adresář, obsahující zdrojové soubory automatických testů
- *Poster/*
- adresář obsahující poster, popisující tuto práci
- *Sramek_A16N0062P_DP_2019.pdf* - text této práce