

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Personální informační systémy
Bakalářská práce

Autor: Václav Lanžhotský
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Tomáš Nacházel, Ph.D.

Hradec Králové

Září 2023

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 10.8.2023

vlastnoruční podpis

Václav Lanžhotský

Poděkování:

Tímto bych rád poděkoval vedoucímu mé bakalářské práce Ing. Tomáš Nacházel, Ph.D. za všestrannou pomoc, množství cenných a inspirativních rad, podnětů, doporučení, připomínek a zároveň za velkou trpělivost s obdivuhodnou ochotou při konzultacích poskytnutých ke zpracování této práce.

Anotace

Bakalářská práce se věnuje problematice personálních informačních systémů. Čtenář po jejím prostudování získá vhled nejen do problematiky samotného návrhu aplikace, ale i do návazností personální problematiky a v neposlední řadě bude seznámen s metodikou návrhu aplikace. V práci není opomenut ani vývoj personálních informačních systémů a jejich důležitost pro 21. století.

Annotation

Title: Human resources information systems

The bachelor thesis is devoted to the problem of human resources information systems. After studying it, the reader will gain insight not only into the design of the application itself, but also into the continuity of personnel issues and, last but not least, will be familiar with the methodology of the application design. The development of human resources information systems and their relevance for the 21st century is not neglected in the work.

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Systémy	3
3.1	Informační systémy (IS)	3
3.2	Cyklus života IS.....	3
3.2.1	Plánování a příprava systému.....	4
3.2.2	Analýza a návrh IS	5
3.2.3	Implementace systému	7
3.2.4	Příprava na zavedení	8
3.2.5	Provoz a užití systému	9
3.2.6	Rozvoj a optimalizace aplikace.....	10
4	Personalistika	12
4.1	Obory personalistiky.....	12
4.2	Využití informačních technologií v oblasti personalistiky.....	12
4.3	GDPR	13
5	Personální informační systémy	14
5.1	Historie a vývoj personálních informačních systémů	14
5.2	Výhody personálních informačních systémů.....	14
5.3	Možné nevýhody personálních informačních systémů	15
5.4	Dostupné personální informační systémy.....	15
5.5	Srovnání používaných personálních informačních systémů.....	16
6	Teorie modelování aplikací	17
6.1	Procesní modelování v praxi	17
6.1.1	UML.....	17
6.1.2	BPMN 2.0	22

7	Návrh personálního informačního systému.....	28
7.1	Business process view.....	28
7.2	Collaboration view.....	31
7.3	Conversation view	32
7.4	Požadavky na systém.....	32
7.5	Use case model	34
7.6	Class model	35
8	Shrnutí výsledků.....	36
9	Závěry a doporučení	37
10	Seznam použité literatury	38
11	Přílohy.....	40
12	Datové přílohy	40

Seznam obrázků

Obr. 1	Životní cyklus informačního systému	4
Obr. 2	Model fáze plánování a přípravy IS.....	5
Obr. 3	Detailní zaměření fáze analýza a návrh.....	6
Obr. 4	Fáze implementace.....	8
Obr. 5	Diagram fáze zavedení.....	9
Obr. 6	Diagram fáze provozu	10
Obr. 7	Popis úloh ve fázi další rozvoj a optimalizace	11
Obr. 8	Zobrazení třídy Účet	18
Obr. 9	Popis třídy BankovníÚčet.....	19
Obr. 10	Typy viditelnosti UML atributů a operací.....	20
Obr. 11	Příklad asociace.....	21
Obr. 12	Příklad asociace.....	21
Obr. 13	Značky flow objektů	22
Obr. 14	Typy BPMN událostí.....	23
Obr. 15	Základní druhy BPMN aktivit.....	24

Obr. 16 Přehled typů bran.....	25
Obr. 17 Ukázka aplikace plaveckých drah v BPMN.....	26
Obr. 18 Výčet nejčastějších datových objektů.....	26
Obr. 19 Hlavní aktivity aplikace	28
Obr. 20 Nástup zaměstnance	29
Obr. 21 Revize dokumentů uchazeče	29
Obr. 22 Vedení evidence	30
Obr. 23 Komunikace s účetní firmou	30
Obr. 24 Ukončení pracovního poměru zaměstnance	31
Obr. 25 Conversation view.....	32
Obr. 26 Funkční požadavky	33
Obr. 27 Nefunkční požadavky	33
Obr. 28 Případy užití.....	34

Seznam tabulek

Tabulka 1 Srovnání dostupných personálních informačních systémů.....	16
--	----

1 Úvod

Během posledních desetiletí se díky technologickému pokroku velmi posunuly i informační technologie. Každý tedy začal přemýšlet nad tím, jak by se daly využít k zefektivnění činností. Jednou z těchto činností je i zpracování personální problematiky. Dříve se vše psalo v ruce, maximálně na psacím stroji, ale informační technologie nám dnes dávají možnost zrychlení a zpřehlednění, kterých dříve nebylo možné dosáhnout.

Veškeré činnosti v personalistice jsou tedy mnohem více zautomatizovány, ale základním nástrojem je databáze, ve které jsou data uložena. Do této databáze se ukládá velké množství údajů, které by byly jinak nepřehledné, ale v dnešním světě je možné vyfiltrovat zobrazení pouze několika prvků, se kterými se dále pracuje.

Tato práce se tedy zaměřuje na komplexní návrh systému, který by mohl sloužit k těmto účelům v nejmenovaném řetězci rychlého občerstvení. Aktuálně probíhá evidence v tabulkovém editoru, který je pro vedení společnosti nedostačující a je tedy nutné navrhnout lepší řešení.

2 Cíl práce

Hlavním cílem této práce je zmapování dostupných personálních informačních systémů pro konkrétní pobočku řetězce rychlého občerstvení. Dle výstupů zkoumání se následně autor pokusí navrhnout nové efektivnější řešení. Pro kompletní pochopení této práce je nutné přiblížit čtenáři problematiku personálních informačních systémů a jejich návrhu.

Prvním z celků jsou informační systémy, kde bude vysvětlena jejich důležitost a současně úspora práce, kterou informační systémy přinášejí. Důraz je kladen i na teoretické pochopení návrhu systému, tedy kroků, které musí nastat, aby mohl být implementován funkční systém.

Jedním z cílů je prozkoumání dostupných systémů na trhu a jejich srovnání. To je blíže popsáno v kapitole číslo 5.

V práci není opomenuta ani problematika personální, kterou je pro modelování celého systému třeba pochopit a zmínka je i o personálních informačních systémech a jejich historii.

V kapitole číslo 7 je představena teorie modelování aplikací, tedy jazyky UML a BPMN, které jsou hlavními pilíři jak této práce, tak světa objektového modelování.

Celá práce je uzavřena konkrétním návrhem aplikace, který vychází z předešlých kapitol a seznamuje čtenáře s BPMN a UML diagramy, které popisují možnou implementaci systému.

3 Systémy

3.1 Informační systémy (IS)

Prvně je potřeba definovat pojem systém. Dle Aristotela jsou mnohé složité věci jako celek složitější než jen souhrn částí, ze kterých se skládají. Na systém může mít každá osoba jiný pohled, primárně dle toho, jak moc je zainteresovaná do dané problematiky. Systémem se tedy rozumí jakýkoliv celek, jehož části mají něco společného a dokážou spolu interagovat.

V oboru informačních technologií si nejobecněji dokážeme představit jakýkoliv operační systém nebo vnitřní podnikový systém. Tyto systémy usnadňují život jejich uživatelům a čím více personifikovaný daný systém je, tím více zjednoduší práci jeho uživatelů. Při tvorbě těchto systémů je vždy ku prospěchu intenzivní komunikace s programátorem, aby systém opravdu splňoval veškerá očekávání. Informační systémy se obvykle týkají velkých korporátních firem nebo zaměstnanců stejného zaměření – účetní a personalisté. Účelem těchto systémů je zajištění správných informací na správném místě ve správný čas. (1) K naplnění účelu jsou potřebné informační a komunikační technologie (ICT), často je tedy používána zkratka IS/ICT.

Pro zamýšlení se nad IS je dobré nejprve porozumět danému tématu. Z toho důvodu jsou také doporučeny mechanismy, které pomáhají při tvorbě nových systémů jak koncovým uživatelům, tak programátorům aplikací.

„Nacházíme se na prahu síťové inteligence – věku, z něhož se rodí nová ekonomika, nová politika i nová společnost“, jak popisuje Don Tapscott. (2)

3.2 Cyklus života IS

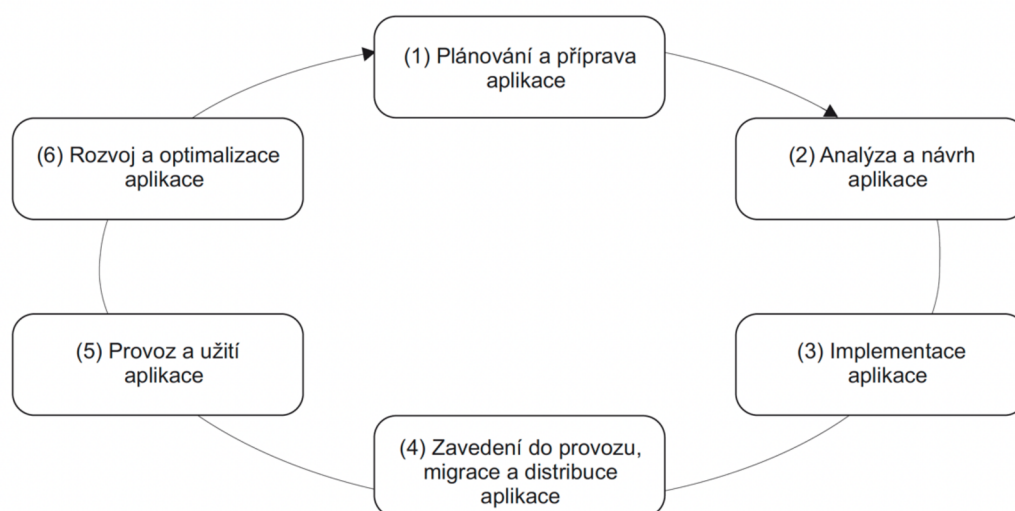
Pod pojmem cyklus života IS si můžeme představit mnoho věcí. Aktuálně je pro nás zásadní pohled při vytváření IS. Tento rozsáhlý celek lze logicky rozdělit na více částí, které na sebe navazují, dávají důležité informace programátorovi a současně nutí zákazníka k zamýšlení se nad novými problémy. Pojem životní cyklus IS je používán hlavně z důvodu, že při dokončení systému se dostáváme opět na začátek, kdy navrhujeme optimalizaci, nové možnosti IS, případně se zamýšlíme nad zadáním zcela nového IS.

Začínáme vždy tím, co je potřeba udělat, co vlastně očekáváme. Jaké vlastnosti má IS mít, k čemu ho chceme využívat, kdo bude koncový uživatel. Poté proběhne stanovení cílů, kde se domluvíme, kdo, do kdy, co udělá tak, aby se společně došlo až do cíle. Jednotlivých fází je více, proto se jim budeme věnovat ještě detailněji.

Rozdělení fází životního cyklu IS: (3)

- Plánování a příprava systému
- Analýza a návrh systému
- Implementace systému
- Zavedení do provozu
- Provoz a užití systému
- Rozvoj a optimalizace systému

Na obrázku číslo 1 je viditelná posloupnost jednotlivých fází společně s tím, jak poslední fáze opět přechází do té první.



Obr. 1 Životní cyklus informačního systému

Zdroj: (3)

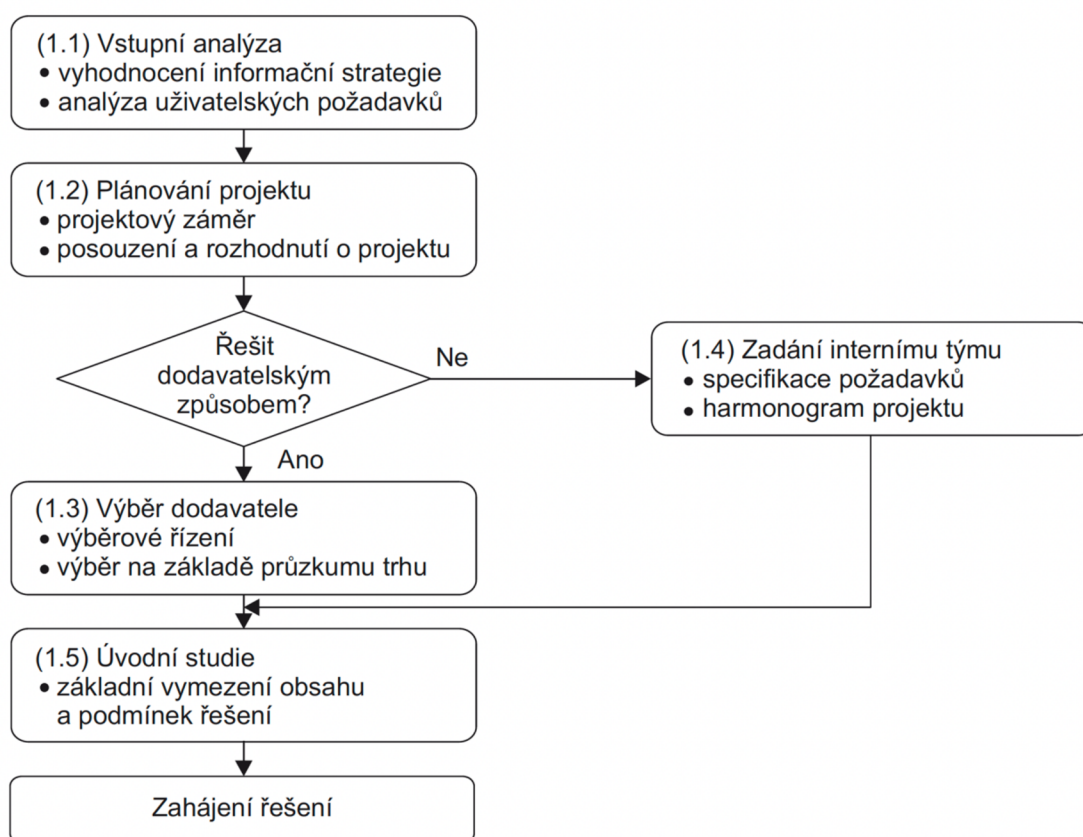
3.2.1 Plánování a příprava systému

Cílem plánování a přípravy je přemýšlet nad tím, co se od systému očekává, pro koho je určen a jaké mají být výstupy. Obrázek 2 zobrazuje jednotlivé úkony této

fáze. Během této fáze musí také padnout rozhodnutí, jestli budeme IS realizovat, neboť na začátku fáze máme pouze záměr realizace.

Pokud se rozhodneme pro realizaci IS, musíme najít vhodného dodavatele a zadat poptávku. Na základě došlých nabídek vybereme komplexním vyhodnocením nejvhodnější.

Je potřeba si uvědomit, že dle výsledku této fáze se bude realizovat celý projekt, proto je důležitá i dostatečná podpora ze strany vedoucích pracovníků směrem k zaměstnancům při návrhu nového systému.



Obr. 2 Model fáze plánování a přípravy IS

Zdroj: (3)

3.2.2 Analýza a návrh IS

Pro tuto fázi je stěžejní důkladná analýza podnikového fungování, neboli zdokumentování veškerých požadavků na daný IS. Je tedy důležité stanovit správně celý proces, případně navrhnout nový, který by již daný systém reflektoval.

Je příhodnější, když se této fázi bude věnovat více členů organizace, neboť každý zodpovídá za nějakou část. Pro tvorbu nového IS je nejlepší zapojit vlastní zaměstnance, tedy ty, kteří budou systém používat nejčastěji.

V této fázi již probíhá definice základních funkcí, datových typů. Analýza návrh je současně důležitou fází pro stanovení architektury budovaného systému a následné rozdělení do kroků, které se budou realizovat.

Tato fáze by se dala tedy rozdělit na analýzu a návrh. Při analýze probíhá analyzování stávajících IS, kontrola dat, databází a následně návrh na zjednodušení, zpřesnění, případně změnu podnikových procesů. (3)

Dimenze	Popis
Funkce/procesy	<ul style="list-style-type: none"> • Detailní specifikace požadavků – model případů užití. • Detailní návrh tříd (designový diagram tříd, sekvenční diagramy).
Data	<ul style="list-style-type: none"> • Návrh fyzické struktury dat. • Návrh persistence dat s ohledem na výkonnost a dobu odezvy. • Návrh testovacích dat. • Určení zdrojů dat pro výchozí naplnění datové základny a návrh transformačních procedur pro tato data.
Aplikační SW	<ul style="list-style-type: none"> • Detailní návrh softwarové architektury aplikace – softwarových modulů (vstupy, výstupy, algoritmus).
Technologická infrastruktura	<ul style="list-style-type: none"> • Upřesnění architektury a dimenzování technologické infrastruktury (typy a parametry HW, potřebné počty licencí infrastrukturního SW). • Detailní návrh konfigurace jednotlivých počítačů. • Návrh umístění jednotlivých zařízení v místnostech. • Návrh úprav budov a místností. • Detailní návrh přenosových cest (typ, použité protokoly, propočty propustnosti sítě). • Nákup HW a infrastrukturního SW pro vývoj a testování. • Harmonogram instalace HW.
Uživatelské rozhraní	<ul style="list-style-type: none"> • Návrh komunikace pro jednotlivé kategorie uživatelů. • Návrh obrazovek (prototypy) a odsouhlasení s uživateli. • Návrh systému nápovědy a chybových hlášení.
Personální, sociální a etická	<ul style="list-style-type: none"> • Revize plánu školení. • Návrh školicích materiálů.
Organizace a legislativa	<ul style="list-style-type: none"> • Promítnutí automatizovaných funkcí do organizačních a pracovních předpisů a do funkčních náplní. • Definitivní návrh přístupových práv ve vazbě na organizační strukturu.

Obr. 3 Detailní zaměření fáze analýza a návrh

Zdroj: (1)

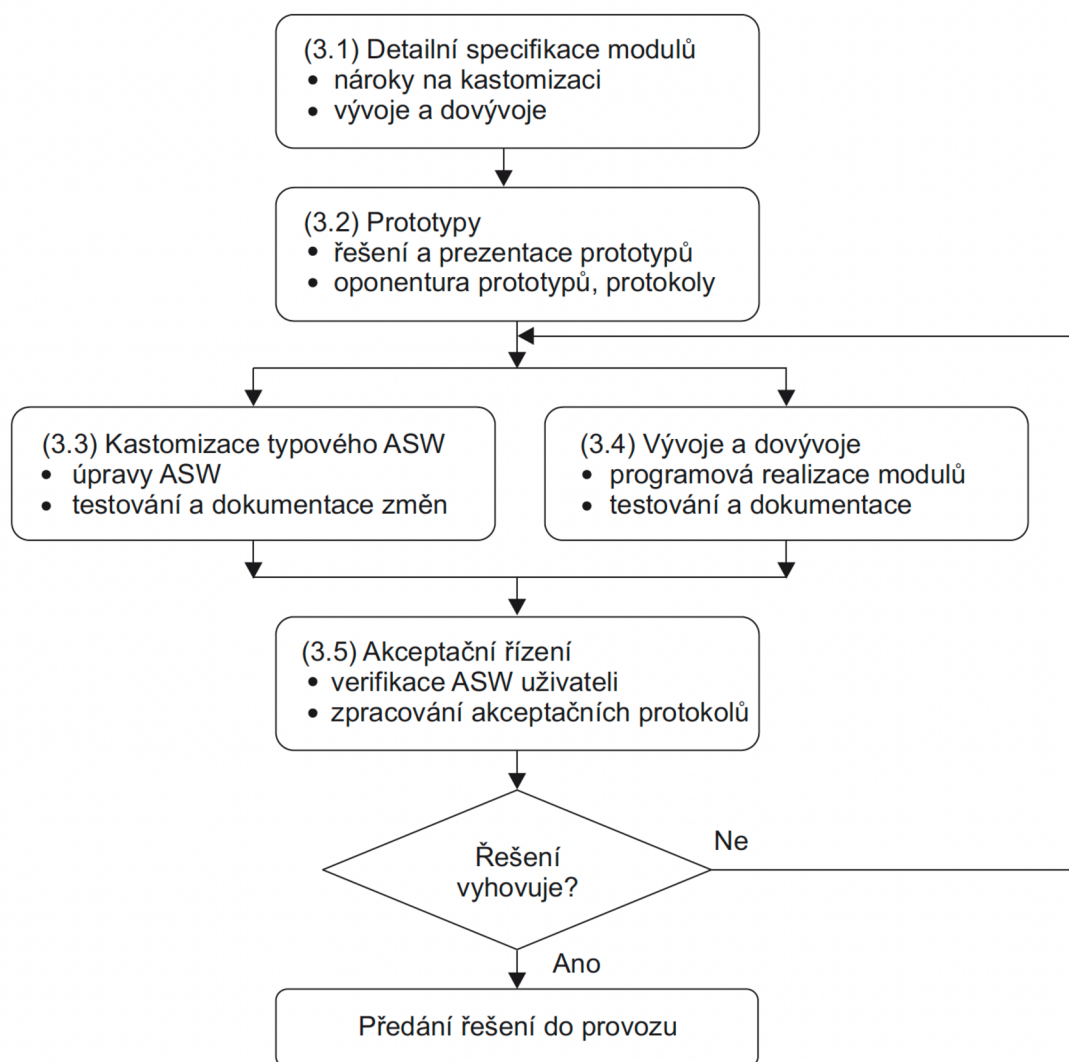
3.2.3 Implementace systému

Do této fáze přecházíme, pokud máme stanovenou architekturu systému, harmonogram a jazyk, ve kterém se bude systém vyvíjet. Současně musí být i jasné rozdělení na jednotlivé části.

Implementaci v pojetí jedné fáze přípravy systému lze chápat jako fázi technologické realizace systému. Ještě konkrétněji se jedná o přesnou specifikaci jednotlivých programových modulů. Následně se kustomizují funkce typového softwaru nebo se vyvíjí či dovyvíjí specializované, tedy nestandardní programové moduly. (3)

Na obr. 4 můžeme vidět jednotlivé kroky této fáze. Důležitou částí jsou prototypy, které zobrazují zkušební vzory, tedy operace, které budeme v systému provádět. V případě dobré realizace prototypů můžeme opět odbourat některé pozdější problémy.

Během této fáze je důležité systém i dostatečně otestovat a kompletovat dokumentaci pro pozdější zaučení a znalost systému. Podstatnou součástí je i odhalení všech omezení a výjimek, tedy stavů, za kterých aplikace může / nemůže fungovat.



Obr. 4 Fáze implementace

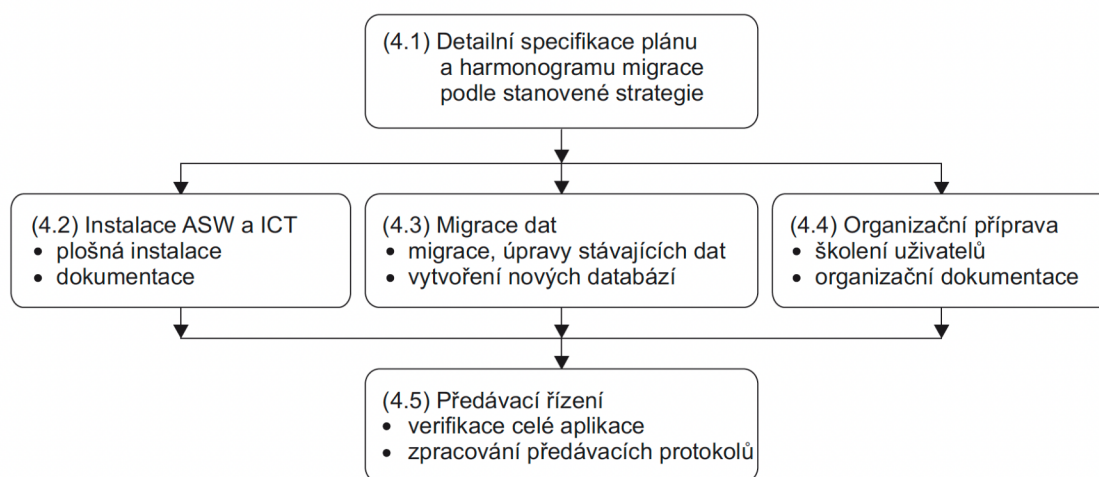
Zdroj: (3)

3.2.4 Příprava na zavedení

Zavedení je potřeba provádět dle předem stanoveného plánu a je třeba postupovat velmi opatrně. Mohlo by totiž dojít k narušení dat. Systém je sice otestován, ale některé chyby se ladí až při nasazení. Důležité je tedy i to, aby obě strany důkladně prověřily systém v testovacím prostředí, kde nemůže dojít k narušení dat a společně se rozhodly, zdali je systém připraven k nasazení do provozu. Testovacích dat sice nemusí být takové množství jako těch reálných, ale velkou část chyb by nám to mělo odhalit.

Před uvedením do provozu je ještě potřeba ověřit, zdali je odběratel připraven po hardwarové stránce na roll-out. Mohlo dojít k navýšení potřebného místa na disku nebo jiných parametrů, které by se musely řešit upgradem hardwaru.

Na obr. 5 si lze povšimnout jednotlivých činností v rámci fáze zavedení. Fáze zavedení končí předávacím řízením. Následuje předání mezi dodavatelem a odběratelem, kdy odběratel stvrzuje, že systém plní veškeré body ze zadání, současně se kontroluje odezva, validují se data.

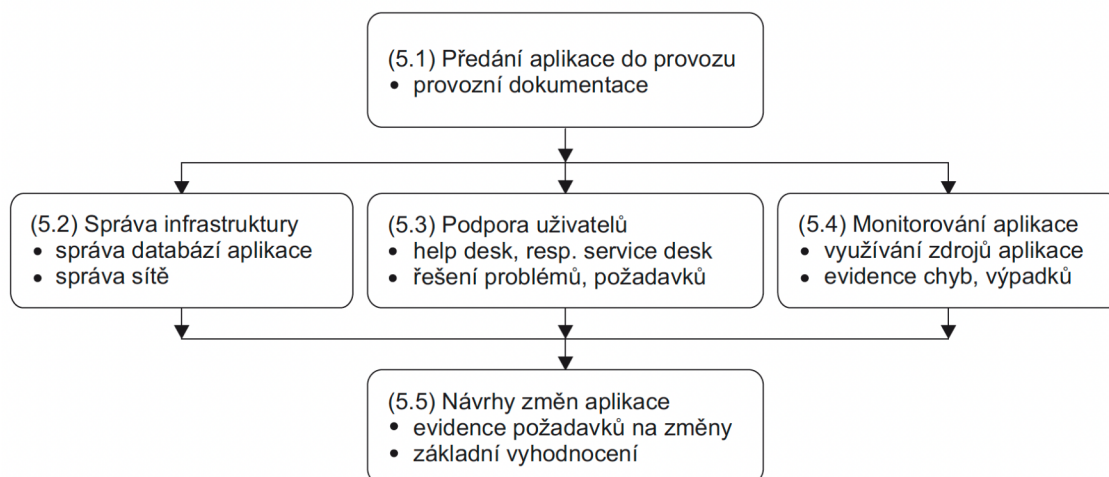


Obr. 5 Diagram fáze zavedení

Zdroj: (3)

3.2.5 Provoz a užití systému

Tato fáze se rozbíhá v momentu, kdy je dokončena migrace a velmi souvisí s fází poslední (Rozvoj a optimalizace aplikace). Jsou zde zahrnuty běžné údržbové operace, provozní servis a permanentní konzultační činnost (3). V této fázi se i formulují statistiky užívání, případně change requests, tedy požadavky na změnu aplikace. Po odbourání těchto prvotních problémů se přechází do fáze poslední.



Obr. 6 Diagram fáze provozu

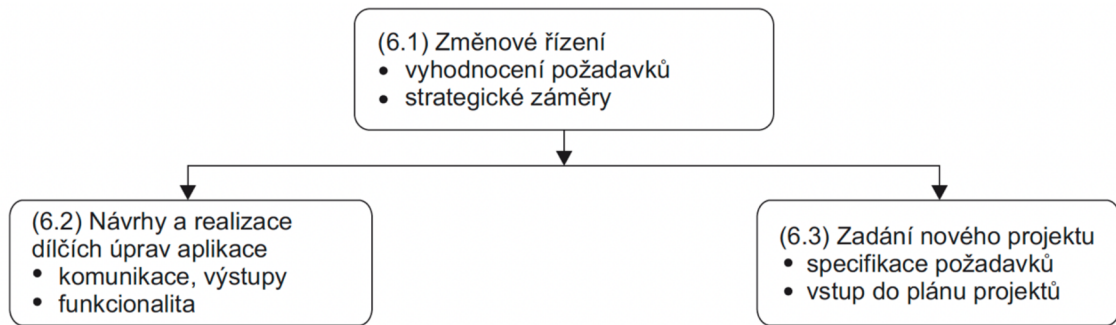
Zdroj: (3)

Z obr. 6 je patrné, že je potřeba předem sjednat servisní podporu ze strany dodavatele, případně získat přístup do kódu aplikace a pověřit někoho na straně odběratele tímto úkolem. Po zavedení nicméně nalezneme několik dalších věcí, jež je potřeba v aplikaci doladit. Současně mohly nastat nějaké změny, které v zadávací dokumentaci nebyly zohledněny.

V budoucnu bude také jistě potřeba pracovat na některých úpravách, ať už legislativních, nebo provozních. Pokud by se systém nevyvíjel, velmi rychle bychom se dostali do stavu, kdy budeme muset řešit systém nový.

3.2.6 Rozvoj a optimalizace aplikace

Tuto fázi můžeme považovat za tu zcela nejdelší, neboť v této fázi jsme po nasazení systému až do případného nasazení nového systému, tedy začátku nového cyklu. Během provozu systému můžeme dojít k tomu, že aplikace je v některých krocích neefektivní a tudíž bude potřeba IS i změnit. Může přijít úprava zákona, dle které bude nutné IS upravit. Případně můžeme systém rozšířit o další funkcionality.



Obr. 7 Popis úloh ve fázi další rozvoj a optimalizace

Zdroj: (3)

Z diagramu na obrázku číslo 7 lze vyčíst, že ne vždy se systém bude rozšiřovat a upgradovat, ale je také možné přistoupit k výměně celého systému. Tedy dostat se na začátek cyklu a začít s celým systémem úplně od začátku. K tomuto rozhodnutí se většinou uchýlíme v momentu, kdy je naše aplikace neefektivní, pomalá a je potřeba opravit velké množství funkcí případně jiných systémových vlastností.

4 Personalistika

„Personalistika je souhrnný název pro obor, který se zabývá řízením a rozvojem lidí v organizaci nebo firmě“. Někdy se také označuje jako HR – z anglického Human Resources, tedy lidské zdroje. (4)

Původ slova personalistika je z latinského personalis – osoba. Z toho již lze vydedukovat, že se personalistika bude zabývat osobami. Dle definice se věnuje lidským zdrojům firmy nebo organizace. Oblastí, které personalistika zahrnuje je velké množství a můžeme tedy vidět pod tímto pojmem mnoho. Pro účely našeho zkoumání nehledáme konkrétní činnost personalistiky, ale soustředíme se na činnosti všechny.

4.1 Obory personalistiky

Dle serveru *Vema* se personalistika dá rozdělit na následující části: (4)

- Nábor nových pracovníků
- Péče o stávající zaměstnance
- Administrativa a strategie firmy

Při mírném použití abstrakce si můžeme představit to, že tyto tři celky opravdu zahrnují veškerou náplň oboru personalistiky. Velkou část těchto úkonů lze provádět digitálně, ať už je to příprava smluv, docházky a další.

4.2 Využití informačních technologií v oblasti personalistiky

S příchodem 21. století a čím dál tím větší snaze po automatizaci všeho možného se odehrál i velký posun v oblasti informačních technologií a jejich využití v personalistice. Veškeré evidence zaměstnanců je totiž možné vést v databázích. Tisk smluv, případně jiných administrativních dokumentů, je otázka několika kliknutí, neboť stačí zvolit dotyčného zaměstnance a vytisknout jakýkoliv dokument. Pokročilé systémy již většinou mají vzory dokumentů importované a stačí tedy dodání dat o zaměstnanci. To systémy zvládnou během několika málo sekund a na pár kliknutí. Pro personalisty tedy v uplynulých letech nastal velký

posun, kdy místo poznámek na papírech psaných rukou případně psacím strojem mohli začít používat tabulkové editory, případně pokročilejší databázové systémy.

Změny nastávají nejen v oblasti evidence zaměstnanců, ale i v metodice jejich náboru, kde lze také reflektovat moderní dobu. Mnoho pracovních pohovorů je v dnešní době realizováno přes aplikace umožňující videohovory. To může velmi ulehčit práci personalistům, zároveň jim to nabízí i nové možnosti vedení pracovního pohovoru. Před několika lety o této verzi pracovního pohovoru málokdo věděl.

4.3 GDPR

Zkratka GDPR vychází z anglického názvu General Data Protection Regulation. Dne 25.5.2018 vyšlo v platnost toto nařízení Evropského parlamentu a Rady (EU) 2016/679 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a volném pohybu těchto údajů. (5) Toto nařízení upravuje mimo jiné zabezpečení dokumentů a dalších materiálů, kde jsou uvedeny osobní údaje fyzických osob.

Pro účely našeho zkoumání je dobré zmínit i tento fakt, neboť informační systémy jsou zabezpečené a reflektují tyto právní úpravy. V systémech se běžně nastavují role uživatelů, je tedy zaručeno, že se k osobním údajům nedostane nikdo nepovolaný. Současně jsou data šifrována a je tedy omezené i riziko jakéhokoli typu hackerského útoku.

5 Personální informační systémy

Personální informační systém je konkrétní systém využitelný v personalistice a spojuje tedy kapitoly 4. a 5. Jedná se o nástroj, který velmi ulehčuje práci personalistům, kteří, jak jsem již psal, mají opravdu velkou náplň práce.

5.1 Historie a vývoj personálních informačních systémů

Historie personálních informačních systémů se začala psát v šedesátých letech dvacátého století. Jednalo se, z dnešního pohledu, o jednoduché tabulkové editory, kde byly realizovány některé části výpočtů mezd a platů. Od Západu se systémy během sedmdesátých let postupně dostaly i do tehdejšího Československa.

Systémy se časem vyvíjely a začalo se v nich uchovávat čím dál více informací. Na začátku přibývaly informace o zaměstnancích – adresa, číslo dokladu, kvalifikace. Vzhledem k socialistickému režimu v Československu se evidovaly i pro tehdejší režim důležité informace, tedy členství v KSČ, funkce ve straně, členství v odborech. Bylo to na úkor kompetencí a dovedností jednotlivých osob, neboť se přihlíželo primárně ke členství ve straně než ke specializaci v oboru.

Od devadesátých let dvacátého století začalo docházet k ještě většímu rozvoji personálních informačních systémů. Hlavním důvodem bylo to, že čím dál tím více lidí začalo rozumět technice a chtěli na systémech pracovat. Proto v České republice bylo založeno velké množství firem se specializací na tvorbu, implementaci a servis personálních informačních systémů.

Aktuální situace na trhu je velice dobrá. Personálních systémů je nabízena celá řada a zákazník si tedy může dle parametrů vybrat ten, který mu vyhovuje nejvíce. Současně je dostupná IT podpora, která dokáže pro zákazníka doplnit chybějící moduly, importy, atd. Nejnovějším trendem jsou webové aplikace umožňující uživatelům vstupovat do systému z jakéhokoli místa a zařízení. Dávají tak uživateli ještě větší komfort při práci se systémem.

5.2 Výhody personálních informačních systémů

Jak již bylo řečeno, systémy a počítačové technika obecně velmi ulehčují mnoho procesů a v personalistice tomu není jinak. Velké usnadnění nacházejí primárně v evidenci, tedy databázích, kde je uloženo velké množství záznamů.

Největším přínosem je fakt, že je vše na jednom místě a personalisté nemusí využívat kdejaké kartotéky, sklady, nebo jiné druhy a místa evidence. Díky evidenci v databázi je také možno dělat srovnání jednotlivých hodnot. Pro personalisty je systém vhodný i díky snadné aktualizaci údajů, která je v tomto oboru velmi potřebná. U systémů jsou většinou zajištěny exporty dat, není tedy problém cokoliv tisknout, přenášet do jiných tabulek, či dokumentů.

Výsledkem všech těchto ulehčení je fakt, který oceňují zejména majitelé a vysoce postavení manažeři firem. Díky informačním systémům lze ušetřit peníze za mzdy. Pokud je vynaloženo dostatečné úsilí k implementaci systémů do provozu, například pomocí nástrojů jazyka UML, pak by aplikace měla zjednodušovat co nejvíce firemních procesů. Každý proces, který systém vykoná místo zaměstnance, šetří peníze. Je tedy potřeba počítat s počáteční investicí, která by se vždy podnikateli měla vrátit. Otázkou zůstává, za jak dlouho.

5.3 Možné nevýhody personálních informačních systémů

Ve 4. kapitole jsme se seznámili s možnými problémy při návrhu personálního informačního systému. To může být nedostatečný návrh objednavatele a to hlavně ze strany uživatelů, kteří budou daný systém používat. Je proto důležitá důkladná příprava, aby se nezapomnělo na žádnou část firemních procesů. Systém musí být navrhnut tak, aby myslel na všechny možnosti, které mohou nastat. Současně by měl pokrývat co nejvíce firemních procesů, aby došlo k co největšímu ulehčení práce zaměstnanců. Pokud je takto tvořen systém na míru, je důležité i zapojení vedení firmy. To posuzuje kvalitu výsledného systému a je tedy nutná shoda i s nimi při závěrečné validaci projektu.

5.4 Dostupné personální informační systémy

Nejčastěji firmy sahají po lokálních řešeních, které jsou nastaveny na lokální zákony a jsou tedy vhodné pro firmu sídlící na území daného státu. Jinak tomu není ani v případě firmy na pozadí této práce, která užívá software od české firmy Vema, která se věnuje zpracování mezd a současně i školení uživatelů jejich systému ať už v práci se systémem samotným, ale i v novinkách personální agendy.

Doplňkem k tomuto systému je systém od další české firmy Anet. Ten slouží k plánování směn a evidenci docházky. Vzhledem k tomu, že tyto systémy jsou schopny generovat mnoho exportů, je jejich provázání opravdu snadné. Současně systém Anet podporuje i externí systém „píchaček“ a je tedy možné nahrávat data o odpíchnutí z jiného zdroje.

5.5 Srovnání používaných personálních informačních systémů

Z dostupných pramenů lze zjistit některé parametry nejpoužívanějších personálních informačních systémů, jak je patrné z tabulky číslo 1 (6). Dle referencí firmy velmi často využívají řešení Vema, které je přívětivé nabízenými funkcemi. (6) Většina dostupných systémů nabízí převážně evidenci zaměstnanců a rozvrh pracovní doby. Některé ovšem umožňují i zpracování mezd a evidenci elektronických dokumentů. Některá prázdná pole tabulky nebylo možné dohledat vzhledem k tomu, že systémy jsou placené a není tedy možné zdarma získat dokumentaci daného systému.

Tabulka 1 Srovnání dostupných personálních informačních systémů.

	Vema (7)	Sloneek (8)	Okbase (9)	Anet (10)	Aptien (11)	Vario (6)
Evidence	Ano	Ano	Ano	Ano	Ano	Ano
Zpracování mezd	Ano	-	-	-	-	Ano
Elektronická komunikace	Ano	-	-	-	-	-
Rozvrh pracovní doby	Ano	Ano	Ano	Ano	-	-
Elektronické dokumenty	-	Ano	Ano	-	Ano	-
Převodní příkazy	Ano	-	-	-	-	-
Support	Ano	-	Ano	Ano	-	Ano
Free verze	-	Omezený čas	-	-	-	-

Zdroj: vlastní zpracování

6 Teorie modelování aplikací

Návrh aplikace realizujeme formou procesního modelování. Tato forma je vhodná pro to, aby si koncový uživatel promyslel veškeré požadavky na systém a programátor následně neřešil velké množství úprav aplikace.

Dle Dumase se podnikový proces skládá z následujících částí:

- Aktivita – Dílčí činnost, kterou koná jeden z objektů. Je vždy vyvolána jinou událostí či aktivitou.
- Událost – Skutečnost, která může nastat a vyvolat tak navazující aktivity. Aktivita též může vyvolat událost.
- Rozhodnutí – Místo, kde se dle vstupních dat rozhodujeme, kterým směrem se vydáme.
- Účastníci procesu – Lidé, uskupení lidí, systémy, které vstupují do procesu.
- Výstup – Výsledek ukončení procesu. Může nastat, ale nemusí a pokud nastane, tak může být pozitivní nebo negativní. (12)

6.1 Procesní modelování v praxi

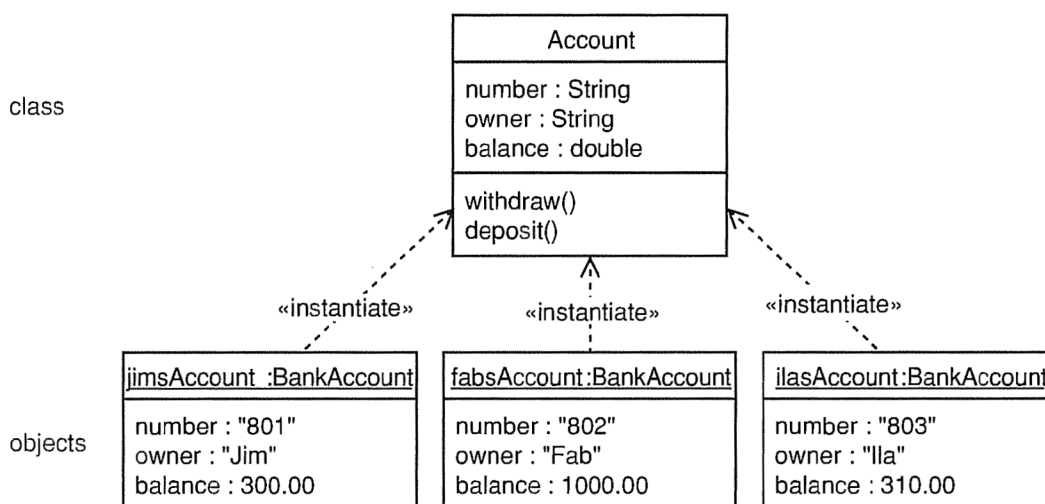
V dnešním světě existuje mnoho notací, které se využívají pro procesní modelování, my se zaměříme na notaci BPMN 2.0 a UML. Všechny notace jsou popsány v ISO standardech, aby byly unifikovány. Nejčastěji se tedy můžeme setkat se standardy ISO 9000 a ISO 9001, jež pojednávají o kritických firemních procesech. Normy UML a BPMN jsou standardizovány normami ISO 19505 a ISO 19510.

6.1.1 UML

Unified modeling language (UML) je standardizovaný modelovací jazyk. Je to jeden z jazyků, který je používán v komunikaci mezi analytikem a vývojářem aplikace. Slouží primárně k definování, designu a specifikaci nového SW. Během prvních let 21. století došlo k dokončení verze 2.0, která přinesla podstatná rozšíření.

6.1.1.1 UML třída

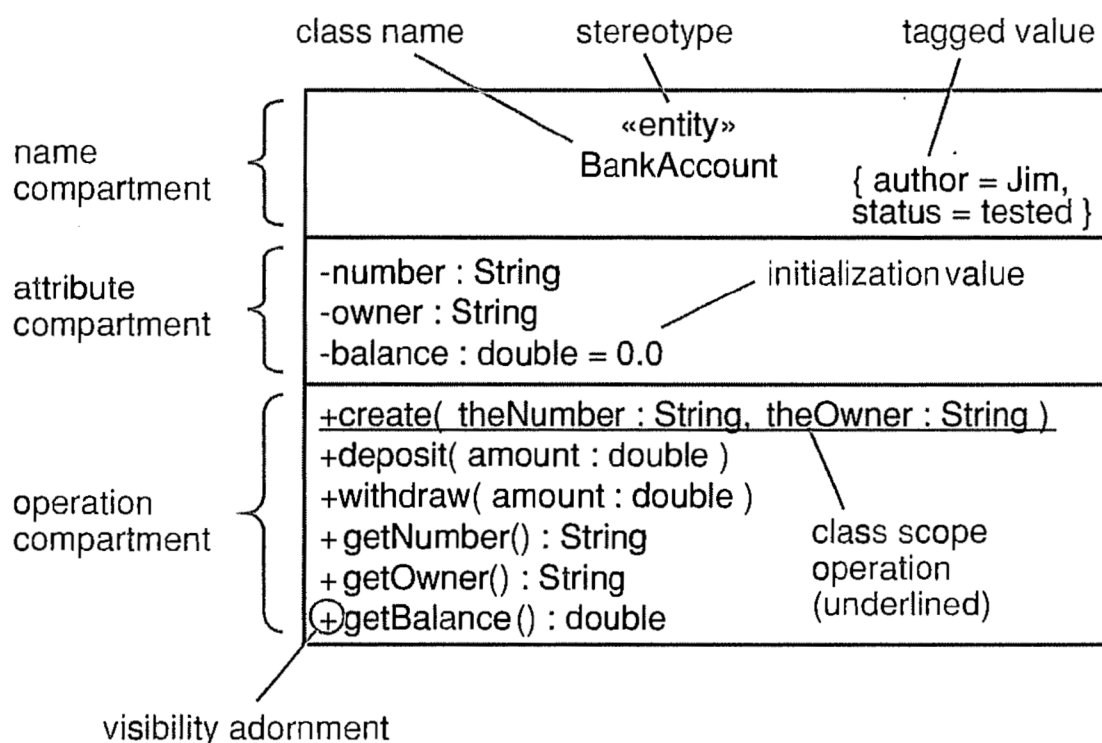
Mezi základní pojmy z UML se jednoznačně řadí třída. Arlow definuje třídu jako strukturu (množinu rysů) všech objektů daného typu (13). Všechny objekty této třídy musí obsahovat stejné atributy (vlastnosti) a relace (spojení mezi prvky modelu), atributy však mohou mít různé hodnoty. Na obrázku číslo 8 je namodelována třída Účet mající atributy *číslo*, *vlastník* a *zůstatek*, dále obsahuje metody *vybrat()* a *uložit()*.



Obr. 8 Zobrazení třídy Účet

Zdroj: (13)

Na obrázku číslo 9 je zobrazená třída BankovníÚčet, je zde poukázáno na několik dalších vlastností, jako je typ viditelnosti a typ hodnoty atributu, tyto vlastnosti budou zmíněny v následující kapitole.



Obr. 9 Popis třídy BankovníÚčet
Zdroj: (13)

6.1.1.2 Atributy a operace tříd

Na obrázcích číslo 8 a 9 jsme si již mohl všimnout tzv. velbloudí notace, tedy zápisu, kdy každé slovo má velké počáteční písmeno při zápisu více slov, to ovšem platí pouze pro názvy tříd. Pro názvy atributů a operací platí to, že začínají písmenem malým a až případné další slovo v řetězci začíná písmenem velkým.

Atributy mohou mít jednu ze základních 4 viditelností. Implementační jazyky mohou využívat i další, ale UML podporuje pouze ty, které máme zobrazeny na obrázku číslo 10. Jedná se o vlastnost, která určuje, kdo bude mít k danému atributu přístup a můžeme tak přístup omezit pouze na operace dané třídy, nebo pro jakoukoliv operaci, která má k třídě přístup.

Adornment	Visibility name	Semantics
+	Public visibility	Any element that can access the class can access any of its features with public visibility
-	Private visibility	Only operations within the class can access features with private visibility
#	Protected visibility	Only operations within the class, or within children of the class, can access features with protected visibility
~	Package visibility	Any element that is in the same package as the class, or in a nested subpackage, can access any of its features with package visibility

Obr. 10 Typy viditelnosti UML atributů a operací

Zdroj: (13)

Každý atribut musí obsahovat typ hodnoty daného atributu. Za základní typy UML považuje Integer (celé číslo), Boolean (hodnota true nebo false), String (posloupnost znaků) a UnlimitedNatural (celé číslo větší nebo rovné nule). Každý jazyk následně přidává své další datové typy. Existují i složené datové typy pro datum (LocalDate), seznam (ArrayList) a další.

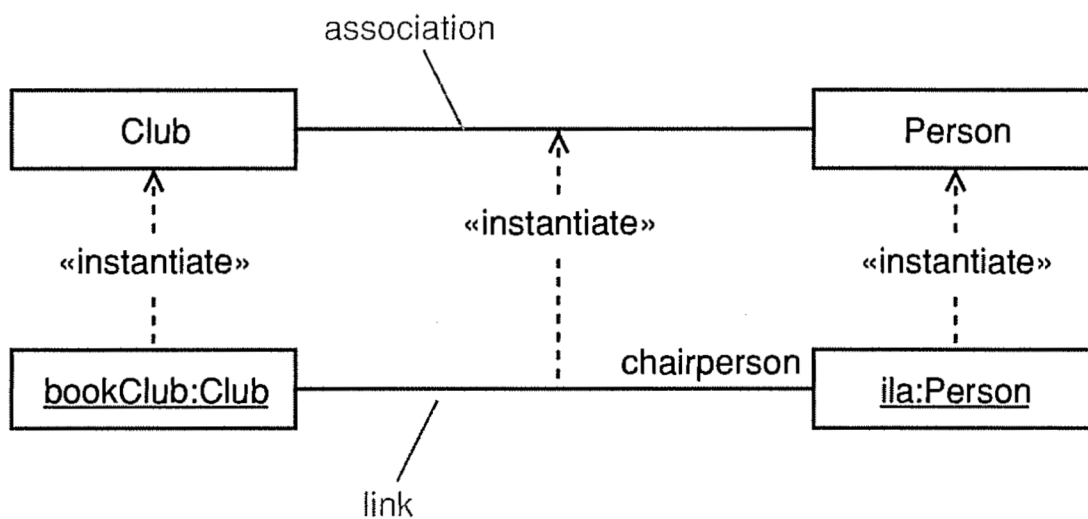
Operace jsou funkce, které se váží k určité třídě a definují její chování. Mají svůj název, seznam argumentů a typ návratové hodnoty. Stejně jako atributy mají i operace svou viditelnost.

6.1.1.3 Relace

„Relace jsou sémantickými vazbami mezi modelovanými prvky – jsou způsobem, jímž v jazyce UML spojujeme předměty a abstrakce dohromady.“ (13)

Pod pojmem relace si tedy můžeme představit vazbu mezi jednotlivými objekty. Pokud dva objekty mají v objektově orientovaném programování mezi sebou vazbu, mohou spolu komunikovat a předávat si zprávy.

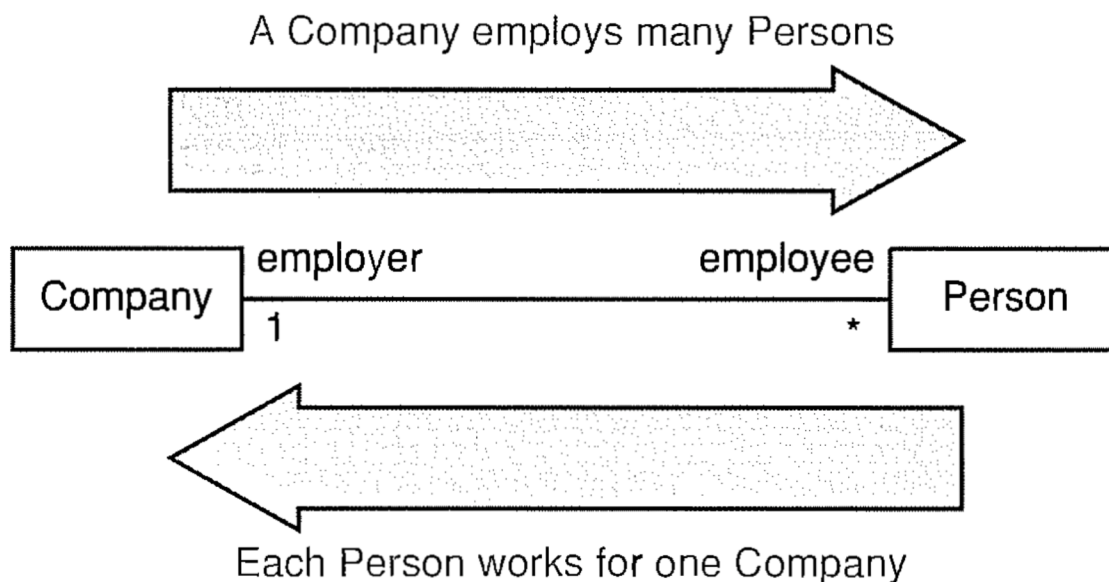
Základním typem vazby je asociace. Na obrázku číslo 11 je velmi jednoduše popsáno, jak lze asociaci využít. Jedná se o spojení mezi třídami *Klub* a *Osoba*. Díky tomuto spojení spolu mohou komunikovat i instance těchto tříd *knižníKlub* a *jim*.



Obr. 11 Příklad asociace

Zdroj: (13)

U většiny relací určujeme takzvanou multiplicitu (násobnost). Multiplicita je omezení pro počet objektů třídy, které se v daný okamžik účastní relace. Na obrázku číslo 12 je znázorněna multiplicita mezi třídou *Firma* a *Osoba*. Současně obrázek zobrazuje role, tedy popis toho, v jaké roli se daná třída v této relaci ocitá.



Obr. 12 Příklad asociace

Zdroj: (13; 1; 1)

Dalšími typy vazeb jsou agregace a kompozice. Tyto vazby jsou zpřesněné asociace v návrhovém modelu. Agregace je volná vazba mezi dvěma objekty,

zatímco kompozice je příkladem velmi pevné vazby mezi objekty. Nejsnazší je si tyto závislosti představit na příkladech z praxe. Jako agregaci bychom mohli chápat závislost počítač – periferní zařízení a jako kompozici strom – listy. (13)

Ve vazbě typu agregace funguje celek jako dominantní prvek a řídí chod relace. Kompozice funguje na podobném principu, ale na rozdíl od agregace je možné, aby součást vždy patřila pouze jedinému celku.

6.1.2 BPMN 2.0

Business process model and notation (BPMN) je grafické znázornění vykreslující procesy firmy pomocí procesních diagramů. Tato notace se v uplynulém desetiletí vyvinula do verze 2.0. Hlavní myšlenkou této notace je předání srozumitelného diagramu, který pochopí všichni aktéři, včetně zadavatele projektu.

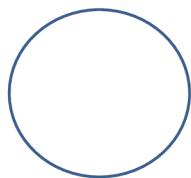
V následujících kapitolách se budeme podrobněji věnovat základním BPMN objektům, což nám pomůže lépe pochopit následný návrh SW v BPMN.

6.1.2.1 Flow objekty

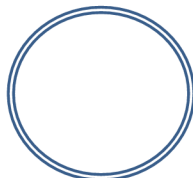
Flow (tokové) objekty jsou základním prvkem BPMN diagramů. Vyobrazují chování daných procesů. Základní stavební kameny tedy jsou události (eventy), aktivity, brány.

Události zahajují, přerušují, nebo ukončují trvání procesu. Mohou na ně mít vliv i jiné procesy. Události lze rozdělit na startovací (vstupní), středové (vnitřní), končící (koncové). Na obrázku číslo 13 jsou znázorněny jednotlivé značky flow objektů.

Startovací událost



Středová událost



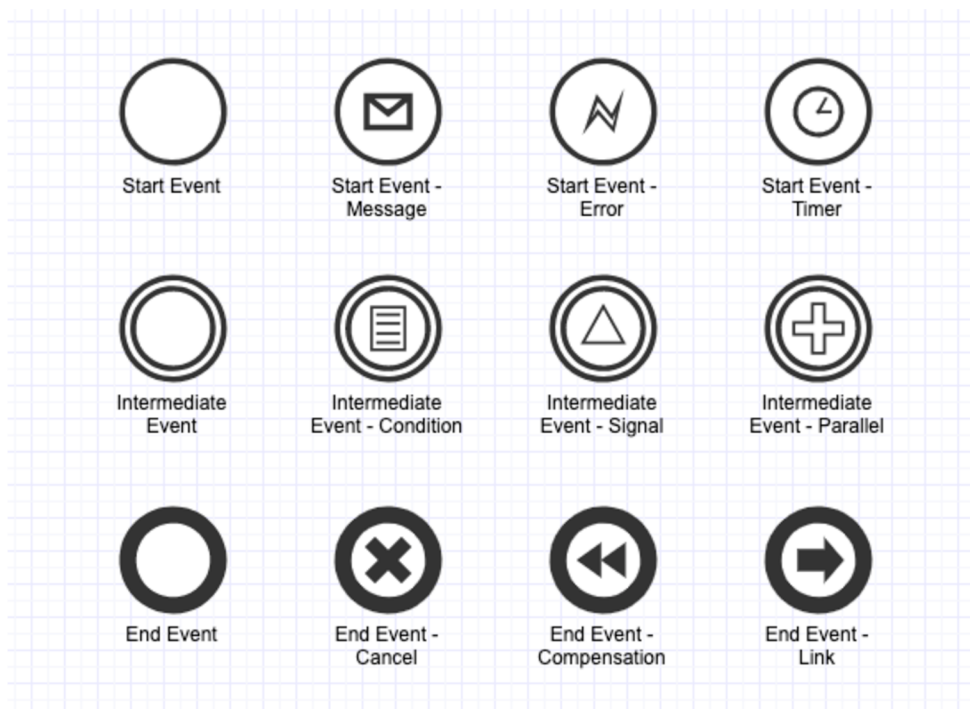
Končící událost



Obr. 13 Značky flow objektů

Zdroj: vlastní zpracování

Dle typu čáry lze soudit, jestli je daná událost startovací, středová, nebo končící. Současně lze definovat typ události dle příčiny jejího vzniku. Událost tedy může vyvolat obdržení nové zprávy, časovač, chybu, signál, a další. Na obrázku číslo 14 je přehled základních typů událostí. Tyto události mohou většinou nabývat všech tří typů a obdržení zprávy tedy může znamenat zahájení procesu. Odeslání zprávy může celý proces ukončit, případně vyvolat další proces.



Obr. 14 Typy BPMN událostí
Zdroj: (14; 15)



Obr. 15 Základní druhy BPMN aktivit
Zdroj: (16)

Aktivity spadají také mezi flow objekty a jejich hlavním cílem je zobrazit vykonání činností. Aktivita by měla být dále nedělitelná, aby byl proces dostatečně podrobný a mohl být zkoumán v diagramu. Systematické značky těchto objektů jsou podobné značkám událostí, ale jsou znázorněny v obdélníčkách místo koleček. Na obrázku číslo 15 jsou znázorněny základní druhy BPMN aktivit. Značky se většinou nacházejí v levém horním okraji aktivity. Nejčastěji se může jednat o klasickou aktivitu (task), případně se objevují i přijetí (receive) či odeslání zpráv (send), ale i uživatelské aktivity (user) a kódové aktivity (script), které vykonává script na pozadí.

Posledním objektem v této skupině je brána, neboli gate. Tyto prvky tvoří společně s aktivitami a událostmi základní pilíře BPMN diagramu. Brány se používají k rozhodování při toku procesu. Systematickou značkou brány je kosočtverec. Dle typu se uvnitř objevuje symbol určující typ brány. Jednotlivé typy bran jsou představeny na obrázku číslo 16.

Základním typem brány je exkluzivní brána (exclusive gateway). Tato brána posílá signál na jedno vlákno dle rozhodnutí ANO / NE. Pokud by tato brána čekala na signál, je jí jedno, kolikrát přijde, ona ho tolikrát pošle. Událostní brána (event-based gateway) vysílá signál při splnění událostní podmínky. Paralelní brána

(parallel gateway) se používá při rozdělení signálu na více signálů, případně při čekání na signál se čeká na doběhnutí všech signálů. Za zmínku stojí ještě inkluzivní brána (inclusive gateway), ta pošle vlákna na všechny aktivity vyhovující podmínce. Naproti tomu komplexní brána (complex gateway) je brána, kterou si může uživatel definovat tak, aby splnila jeho veškeré požadavky.



Obr. 16 Přehled typů bran

Zdroj: (17)

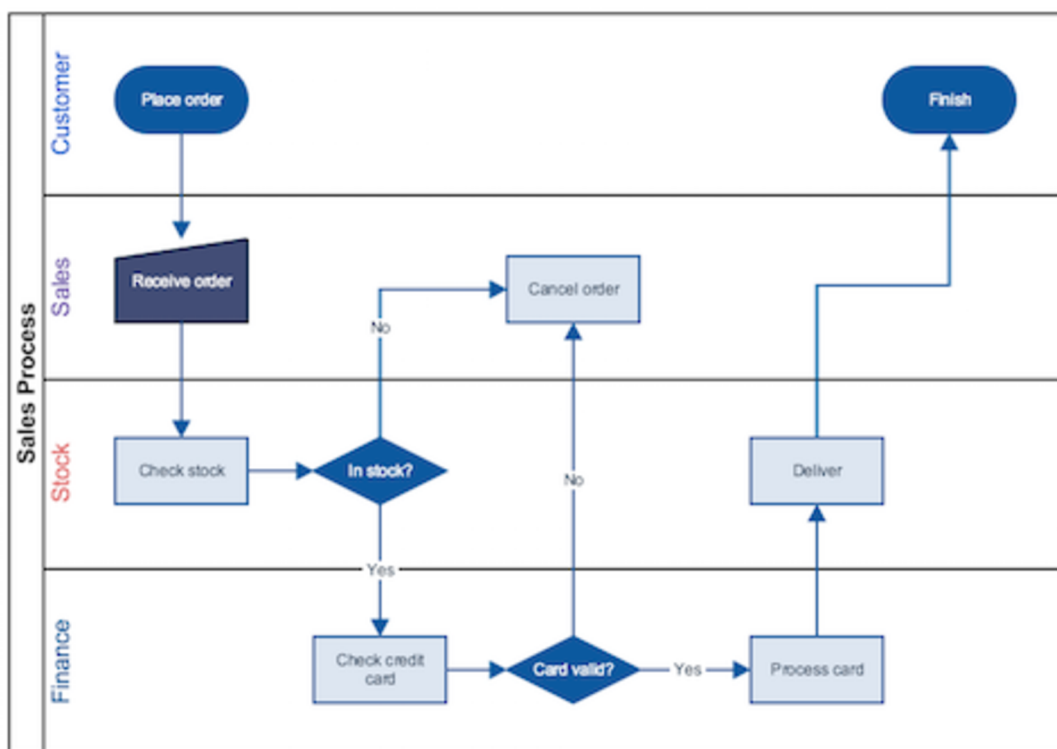
6.1.2.2 Spojovací objekty

Další skupinou objektů jsou spojovací objekty. Ty slouží k propojení flow objektů a vytváří tak tok mezi jednotlivými objekty. Nejběžněji se pro tato spojení používá sekvenční vazba (sequence flow), která definuje směr toku signálu. Současně lze použít i asociaci, nebo vazbu mezi zprávami (message flow) zobrazující směr toku zprávy.

6.1.2.3 Plavecké dráhy

Dalším objektem jsou plavecké dráhy (swimlanes) zobrazené v bazénu. Plavecký bazén slouží k zobrazení aktérů ve firemním procesu. Bazén lze totiž rozdělit na jednotlivé dráhy a každou dráhu vynahradiť pro jednoho aktéra. Proces je díky tomuto členění přehlednější, neboť je vidět, co který aktér (oddělení firmy)

provádí. Na obrázku 17 je naznačeno členění firmy na jednotlivé dráhy a ukázka objednávky v on-line obchodu.

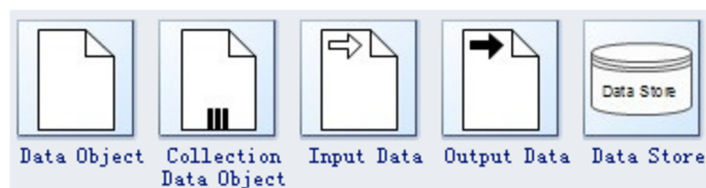


Obr. 17 Ukázka aplikace plaveckých drah v BPMN

Zdroj: (15)

6.1.2.4 Datové objekty

Další skupinou objektů jsou datové objekty, které jednotlivé aktivity využívají při svých činnostech. Mohou do nich tedy ukládat data, případně z nich číst a upravovat je. Obrázek číslo 18 zobrazuje nejčastější datové objekty. Největším objektem je datové úložiště (data store), dále se často objevují vstupní (input data) a výstupní data (output data) sloužící k odesílání a získávání dat. Základní data objekt (data object) se používá hlavně pro snadné dokumenty použité v daném procesu.



Obr. 18 Výčet nejčastějších datových objektů

Zdroj: (18)

6.1.2.5 Artefakty

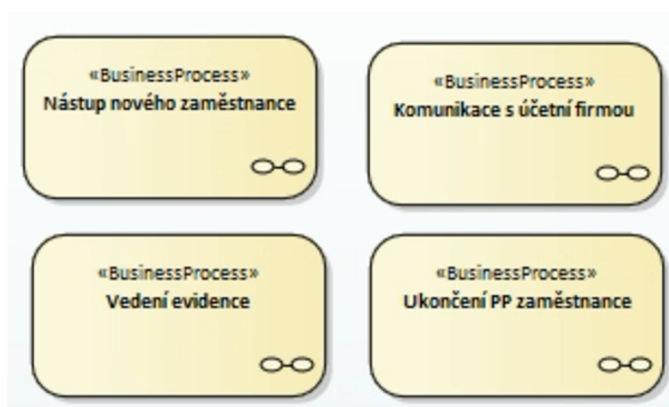
Poslední skupinou objektů BPMN jsou artefakty, které slouží pro doplnění informací k danému procesu. Nejčastěji se jedná o anotace, které se zobrazují jako text spojený tečkovanou přímkou s daným objektem. Dalším artefaktem je skupina, která slouží k logickému seskupení podobných či návazných aktivit v rámci jednoho procesu.

7 Návrh personálního informačního systému

Nyní se můžeme přesunout k návrhu našeho vlastního personálního informačního systému. V následujících kapitolách si detailně rozebereme požadavky na systém, podíváme se na procesní modely a ukážeme si návrh modelu tříd.

7.1 Business process view

Podstatou tohoto zobrazení je podrobně rozčlenit veškeré aktivity na nejmenší možné, aby bylo možné detailně zobrazit firemní procesy. Na obrázku číslo 19 jsou zobrazeny 4 hlavní procesy, které se rozpadají na ještě menší aktivity.

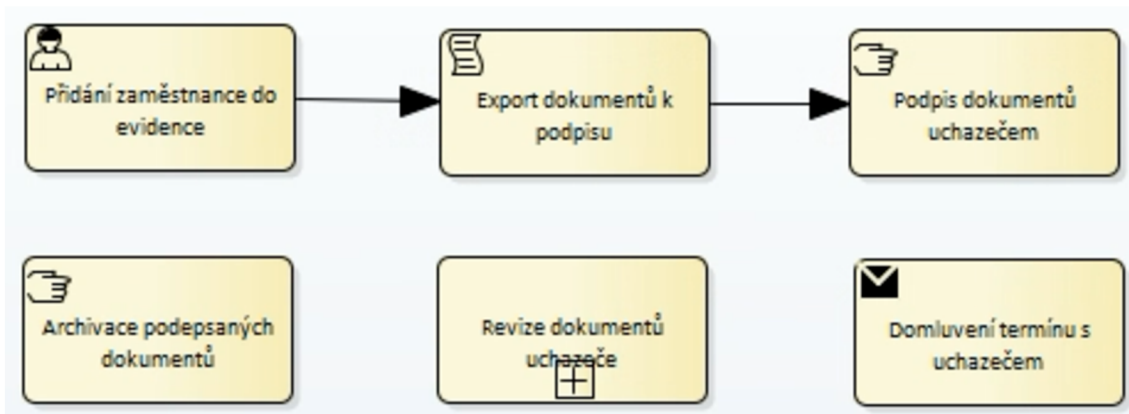


Obr. 19 Hlavní aktivity aplikace

Zdroj: vlastní zpracování

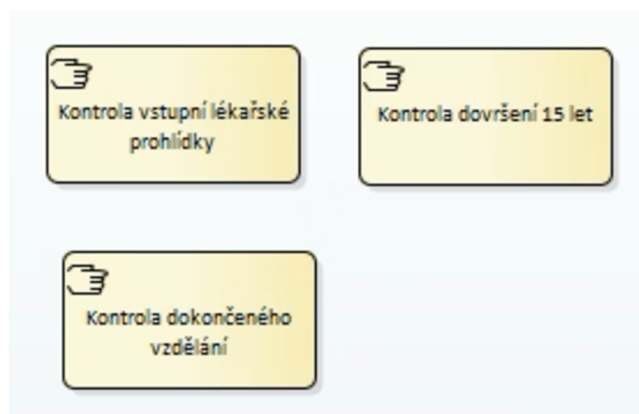
Ve firmách obecně probíhají procesy tak, že nastoupí nový zaměstnanec, poté zaměstnanec pracuje a následně někteří skončí. Proto i já firemní procesy proberu v tomto pořadí.

Nástup nového zaměstnance je velký firemní proces, neboť se musí mnoho informací zaevidovat a je nutné zkontrolovat větší množství údajů, které jsou neměnné. Na obrázku číslo 20 jsou vidět jednotlivé aktivity firemního procesu *Nástup zaměstnance*. Obrázek číslo 21 ještě zobrazuje rozpadlou aktivitu *Revize dokumentů uchazeče*.



Obr. 20 Nástup zaměstnance

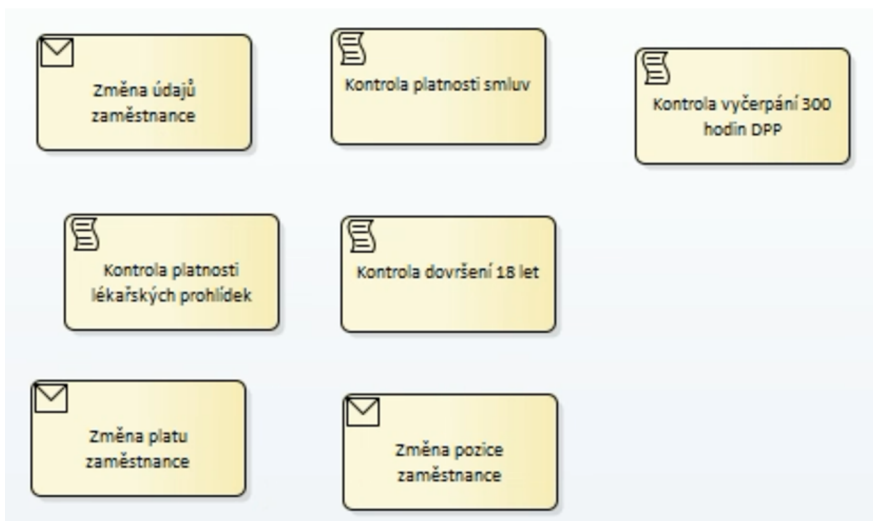
Zdroj: vlastní zpracování



Obr. 21 Revize dokumentů uchazeče

Zdroj: vlastní zpracování

Dalším firemním procesem je vedení evidence. Během toho, co zaměstnanec pracuje ve firmě, totiž může dojít ke změnám v jeho kontaktních údajích, může se změnit číslo účtu nebo jiné údaje. Současně v mnoha firmách dochází k úpravám platu během trvání pracovního poměru a je tedy také nutná změna v jeho kartě. Obrázek číslo 22 je detailním pohledem na firemní proces *Vedení evidence*.



Obr. 22 Vedení evidence

Zdroj: vlastní zpracování

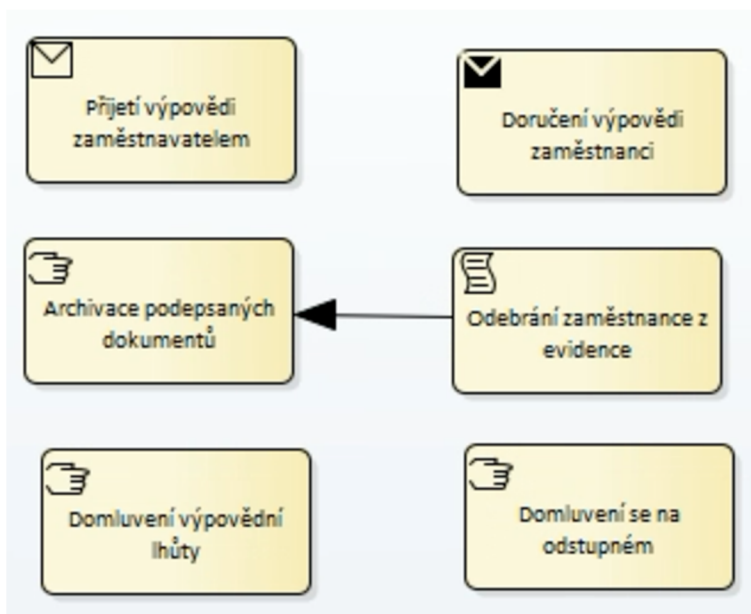
Při zaměstnávání zaměstnanců je důležitá komunikace s účetní. V našem případě není účetní a administrativní pracovník jedna osoba. Proto je nutný export zaměstnaneckých karet, aby nemuselo docházet k tomu, že se budou dané změny ručně psát e-mailem, či jinak. Na obrázku číslo 23 jsou zobrazeny aktivity firemního procesu *Komunikace s účetní firmou*.



Obr. 23 Komunikace s účetní firmou

Zdroj: vlastní zpracování

Posledním velkým firemním procesem je ukončení pracovního poměru zaměstnance, kdy je opět nutné provést větší počet aktivit. Aktivity tohoto firemního procesu jsou na obrázku číslo 24.



Obr. 24 Ukončení pracovního poměru zaměstnance

Zdroj: vlastní zpracování

7.2 Collaboration view

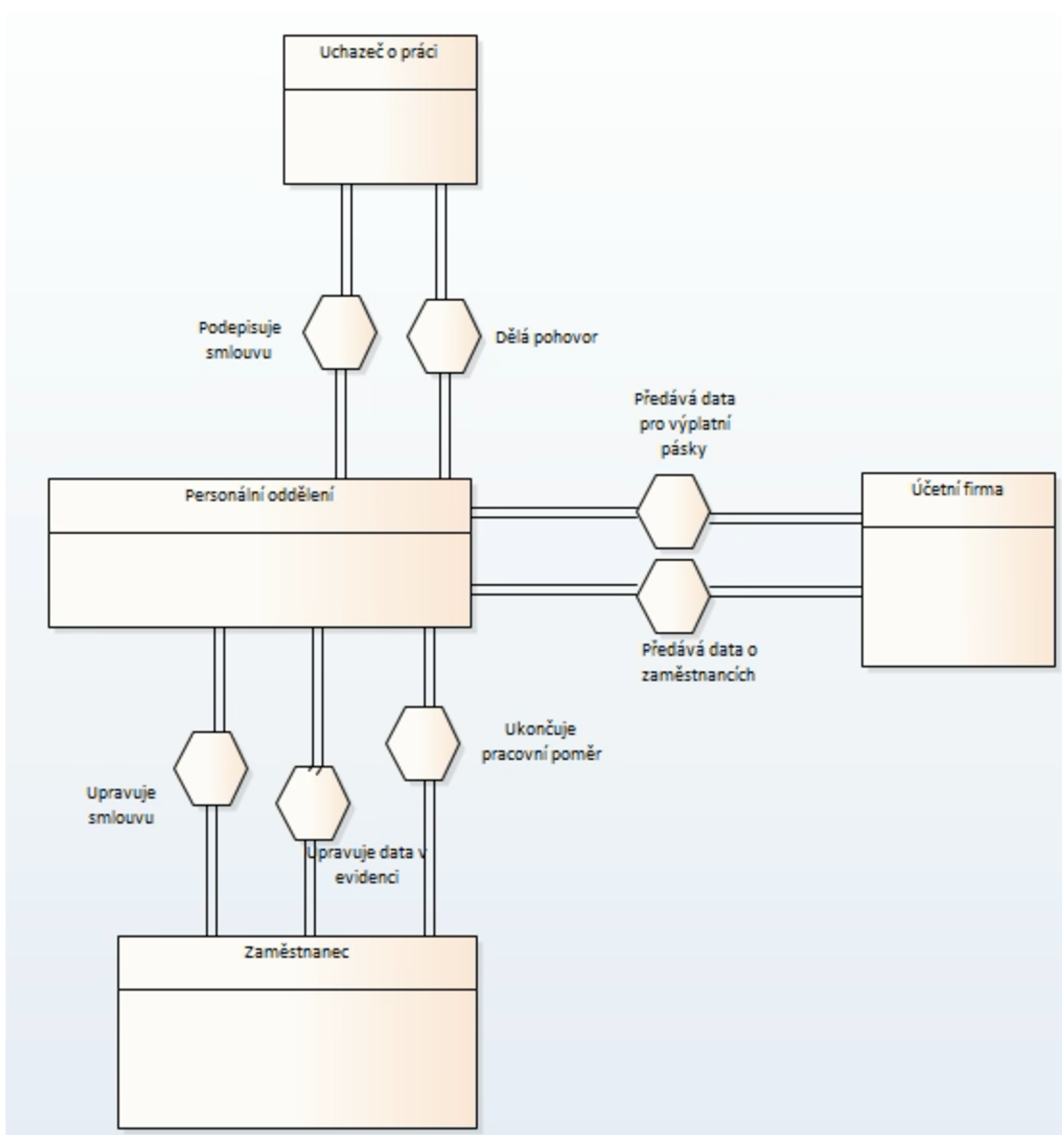
Tento model slouží k detailnímu popisu firemních procesů. V našem případě jsou důležité dva procesy, *Nástup zaměstnance* a *Výstup zaměstnance*. Pro programátora dané aplikace je zde důležité zobrazit daný proces krok za krokem, aby se opravdu na nic nezapomnělo a nemusely se zadávat změnové požadavky na funkční SW.

Obrázková příloha číslo 1 popisuje firemní proces *Nástup zaměstnance*. Celý tento proces začíná reakcí uchazeče a pohovorem, pro naše zkoumání je ale začátek až v momentě, kdy má uchazeč veškeré potřebné dokumenty pro podpis pracovní smlouvy. Během procesu je třeba domluvit datum podpisu smlouvy, po příchodu zkontrolovat dané dokumenty, jestli vše sedí. Pokud ano, pustit se do přidání zaměstnance do evidence a podpisu smlouvy. Pokud zaměstnanec smlouvu podepíše, stává se naším zaměstnancem.

V příloze číslo 2 je zobrazen opačný proces, tedy proces *Ukončení pracovního poměru zaměstnance*. Celý proces má 3 možnosti svého vzniku. Zaměstnanec mohl dostat smlouvu na dobu určitou, mohl se sám rozhodnout ukončit pracovní poměr, případně se rozhodl zaměstnavatel ukončit zaměstnancův pracovní poměr. Pokud obě strany souhlasí s podmínkami výpovědi, proběhne odebrání zaměstnance z evidence a jeho archivace. Tímto je pracovní poměr ukončen.

7.3 Conversation view

Tento diagram zobrazuje jednotlivé aktéry a jejich interakce. Opět je to důležité, neboť je poté patrné, kdo za co odpovídá, kdo co řeší. Na obrázku číslo 25 jsou zobrazeny 4 účastníci, tedy zaměstnanec, personální oddělení, uchazeč a účetní firma, kteří spolu musí řešit aktivity pojmenované v šestiúhelnících.



Obr. 25 Conversation view

Zdroj: vlastní zpracování

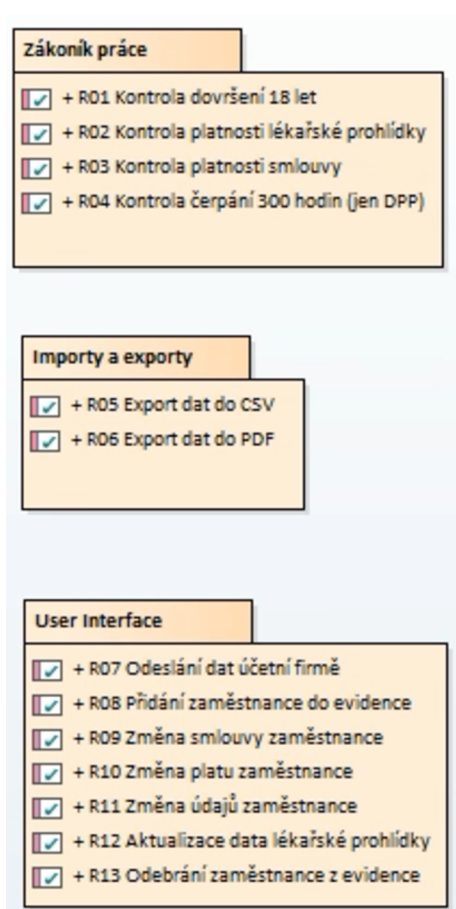
7.4 Požadavky na systém

Požadavky na systém lze rozdělit do dvou kategorií: funkční a nefunkční. Funkční požadavky jsou požadavky na funkčnost daného systému. Naopak nefunkční požadavky jsou požadavky na podmínky / omezení systému. Funkční

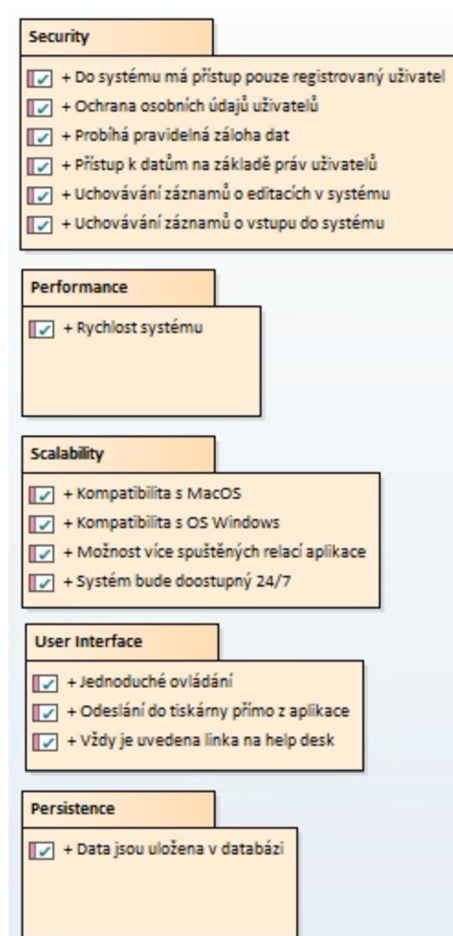
požadavky vyplývají hlavně z předcházejících diagramů. Nefunkční požadavky jsou dané organizací, pro kterou se systém tvoří a svou povahou nemohou vycházet z předcházejících diagramů.

Tyto požadavky vzešly z rozpravy s mou kolegyní, která této problematice věnovala posledních 5 let a přesně tedy věděla, co by takový systém měl obsahovat. Za cenné rady ji moc děkuji.

Obrázky číslo 26 a 27 zobrazují všechny tyto požadavky. Pro snazší členění jsou ještě logicky rozděleny do sekcí dle svých vlastností.



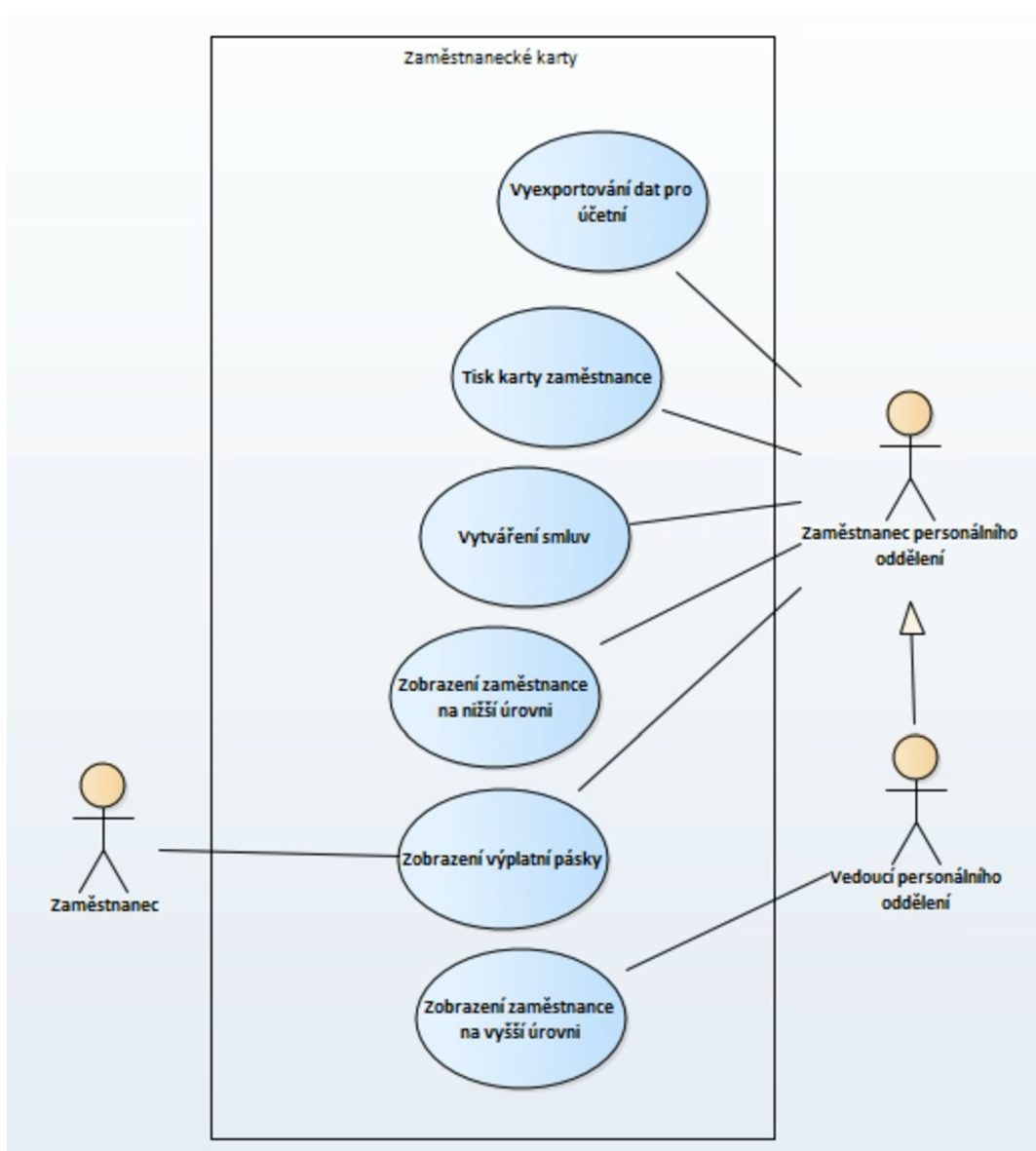
Obr. 26 Funkční požadavky
Zdroj: vlastní zpracování



Obr. 27 Nefunkční požadavky
Zdroj: vlastní zpracování

7.5 Use case model

Neboli diagram případů užití popisuje to, co může uživatel (role) v systému dělat. Vzhledem k tomu, že do systému mohou přistupovat pouze firemní zaměstnanci, jsou jednotlivé role pozicemi ve firmě. Konkrétně se jedná o role *Zaměstnanec*, *Zaměstnanec personálního oddělení* a *Vedoucí pracovník*. Na obrázku číslo 28 jsou popsány jednotlivé případy užití, které opět vycházejí z jednotlivých procesů a aktivit.



Obr. 28 Případy užití
Zdroj: vlastní zpracování

7.6 Class model

Diagramy tříd je posledními modely našeho zkoumání. Tyto modely budou ze všech nejdůležitější, neboť se zde modeluje struktura celého systému. Postupnými kroky jsme se přesunuli od procesů přes aktivity až k požadavkům, rolím a následně modelům tříd. Vše spolu souvisí, každý diagram vychází z toho předcházejícího.

Vždy se začíná modelovat od modelu analytického, který je dost primitivní oproti návrhovému. V analytickém modelu nejsou do velké míry řešeny relace, datové typy a další. Příloha číslo 3 obsahuje vygenerovaný model z nástroje Enterprise Architect, který zobrazuje použité třídy, jejich atributy a metody a nastiňuje tak základní funkčnost aplikace. Jak již bylo několikrát zmíněno, množství dat, které je potřeba uchovávat, je opravdu velké, z toho důvodu je i mnoho atributů u třídy *Zaměstnanec*, neboť každá její instance bude tyto atributy obsahovat.

Příloha číslo 4 je modelem návrhovým, který je již oproti předcházejícímu modelu zpřesněn, jsou zobrazeny datové typy atributů a návratové hodnoty operací. U relací jsou doplněny role a některé asociace jsou zpřesněny na kompozice a agregace. Všechny třídy jsou doplněny o gettery a settery, které umožňují získávání a zápis hodnot a dále o konstruktory, které umožňují vznik instance dané třídy.

Poslední přílohou s číslem 5 je ukázka vygenerovaného java kódu s třídou *Zaměstnanec*, kde jsou navrženy namodelované atributy a operace. Pro funkčnost celého systému by bylo třeba naimplementovat funkce a spustitelné uživatelské rozhraní.

Datová příloha číslo 1 uzavírá celou práci s vygenerovanými Java kódy a současně zdrojovým souborem Enterprise Architectu, kde je celý projekt vytvořen.

8 Shrnutí výsledků

Hlavním cíle práce bylo prozkoumání dostupných personálních systémů a návrh nového řešení. Během přípravy práce bylo ale zjištěno, že bez zakoupení licencí k systémům není možné systémy srovnat, neboť dokumentace není veřejná. Tento cíl práce tedy nebyl splněn.

Z toho důvodu vychází praktická část primárně ze zkušeností autora, které načerpal při tvorbě excelového řešení pro personálně informační agendu. V praktické části byl postupně vytvořen UML diagram tříd, který by v budoucnu měl posloužit jako základ pro nový personální informační systém.

9 Závěry a doporučení

Hlavním cílem práce bylo prozkoumat aktuální situaci mezi personálními informačními systémy, jejich historií a zaměřit se na problematiku implementace nových systémů. V teoretické části byl kladen důraz na poznání cyklů v tvorbě aplikace a dále pochopení BPMN a UML notace, část byla věnovaná i samotné personální problematice a historickému vývoji personálních informačních systémů. Bohužel srovnání dostupných systémů nebylo úspěšné, neboť není možné stáhnout jakoukoliv dokumentace placených systémů.

V praktické části byly představeny jednotlivé diagramy, které při tvorbě diagramu tříd hrály velkou roli. Diagramy byly vytvořeny v Enterprise Architectu, který je pro potřeby modelování aplikací velkým přínosem. Výsledný návrhový model byl užít k vygenerování struktury v jazyce Java obsahující atributy a operace jednotlivých tříd. Tento model bohužel nebylo možné srovnat s existujícími možnostmi, neboť z dostupných zdrojů není možné získat o aktuálních systémech žádnou informaci.

V budoucnu bych rád svůj projekt dotáhl k implementaci a nasazení v reálných provozovnách, kterým by daná aplikace rozhodně ulehčila práci, jak to mají systémy ve zvyku. Současně není problém systém využít i v dalších odvětvích a případně jen přidávat nové drobné funkce.

10 Seznam použité literatury

1. BRUCKNER, Tomáš., VOŘÍŠEK, Jiří a BUCHALCEVOVÁ, Alena. *Tvorba informačních systémů: principy, metodiky, architektury*. Praha: : Grada,, 2012. Sv. Managment v informační společnosti. 978-80-247-4153-6.
2. TAPSCOTT, Don. *Grown up digital*. místo neznámé : McGraw-Hill Professional, 2009. 97800715089636.
3. GÁLA, Libor, POUR, Jan a ŠEDIVÁ, Zuzana. *Počítačové aplikace v podnikové a mezipodnikové praxi*. 3. vydání. Praha 7 : Grada Publishing, a.s., 2015. str. 240. ISBN 978-80-247-9918-6.
4. Vema. *Mzdový a personální software*. [Online] 10. 01 2023. www.vema.cz.
5. Ministerstvo vnitra české republiky. *Ministerstvo vnitra české republiky*. [Online] 10. 01 2023. <https://www.mvcr.cz/gdpr/clanek/co-je-gdpr.aspx>.
6. Seyfor. [Online] <https://www.seyfor.com/cs-cz/hr-systemy#4-verejny-sektor>.
7. Vema. [Online] <https://www.vema.cz/cs/reference>.
8. *Sloneek*. [Online] https://www.sloneek.cz/?utm_term=personální%20informační%20systémy&utm_campaign=%5BCZ%5D%20S%20%7C%20HR%20systém%20%5Bcz%5D&utm_source=google&utm_medium=cpc&hsa_acc=1165394791&hsa_cam=14064662591&hsa_grp=137879985592&hsa_ad=601676390837&hsa_src=g&hsa_tgt=.
9. Okbase. [Online] <https://www.okbase.cz/cs/personalistika>.
10. Anet. [Online] <https://anet.eu/cz/dochazkove-systemy/>.
11. *Aptien*. [Online] https://aptien.com/cs/personalistika?gad=1&gclid=Cj0KCQjwldKmBhCCARISAP-0rfyWarF9j_KuGGWngsdMkX4xnEN0FW5Ko5DDV8Wjc2Egtgb_swfCt4aApc_EALw_wcB.
12. DUMAS, M., M.L. ROSA, J. MENDLING, H.A. REIJERS. *Fundamentals od Business Process Management*. místo neznámé : Springer Berlin Heidelberg, 2013. 978-3-662-58585-6.

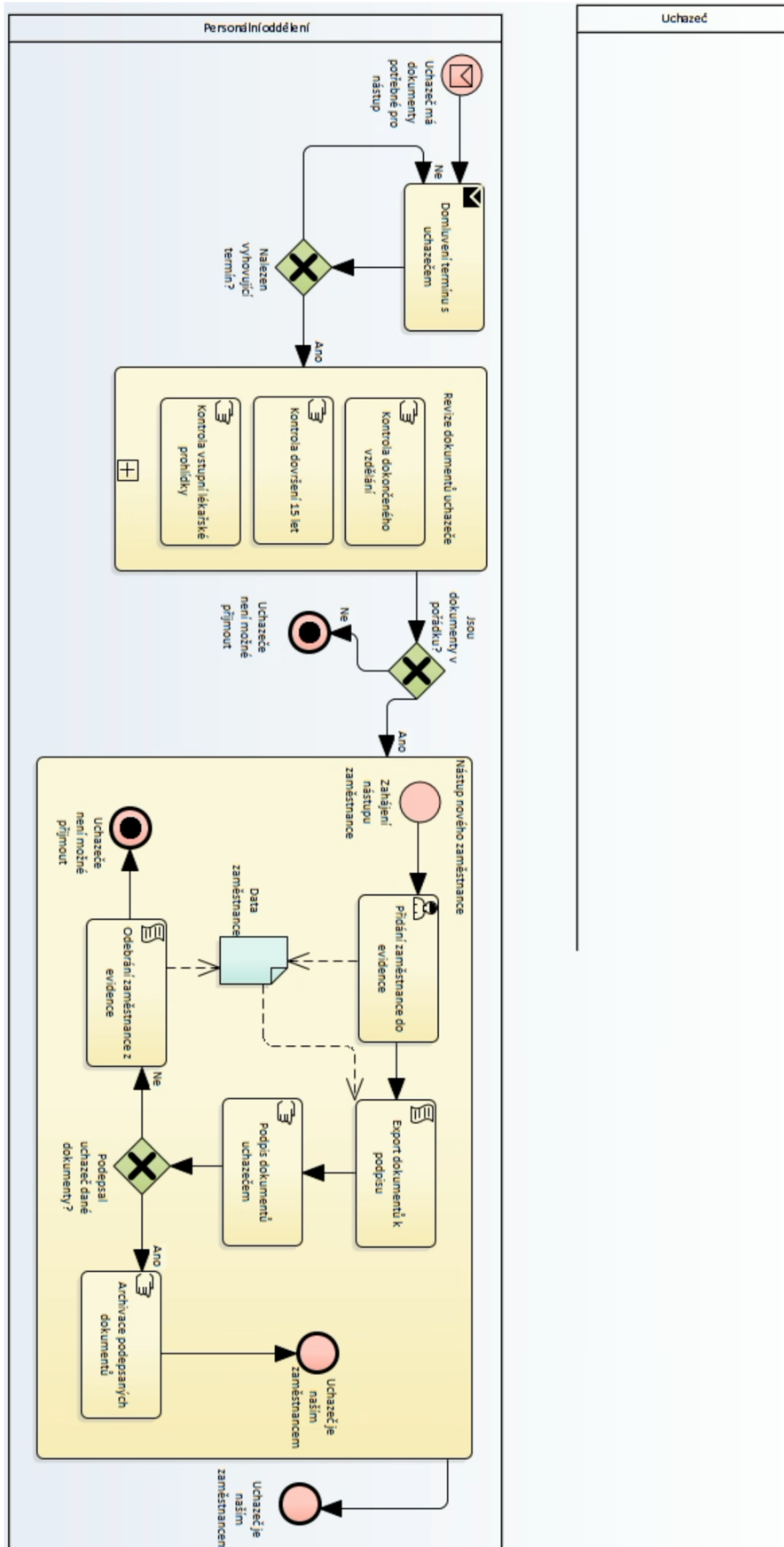
13. ARLOW, Jim a NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací*. Brno : Computer Press, a.s., 2007. 978-80-251-1503-9.
14. *Gliffy*. [Online] 06. 03 2023. <https://www.gliffy.com/blog/guide-to-bpmn-symbols>.
15. *Gliffy*. [Online] 06. 03 2023. <https://help.gliffy.com/confluence/Content/GliffyCommon/Adding-swimlanes.htm>.
16. *Visual Paradigm*. [Online] 06. 03 2023. <https://www.visual-paradigm.com/guide/bpmn/bpmn-activity-types-explained/>.
17. *Wondershare*. [Online] 06. 03 2023. <https://www.edrawsoft.com/how-to-use-bpmn-gateways.html>.
18. *Wondershare*. [Online] 06. 03 2023. <https://www.edrawsoft.com/bpmn-symbols.html>.

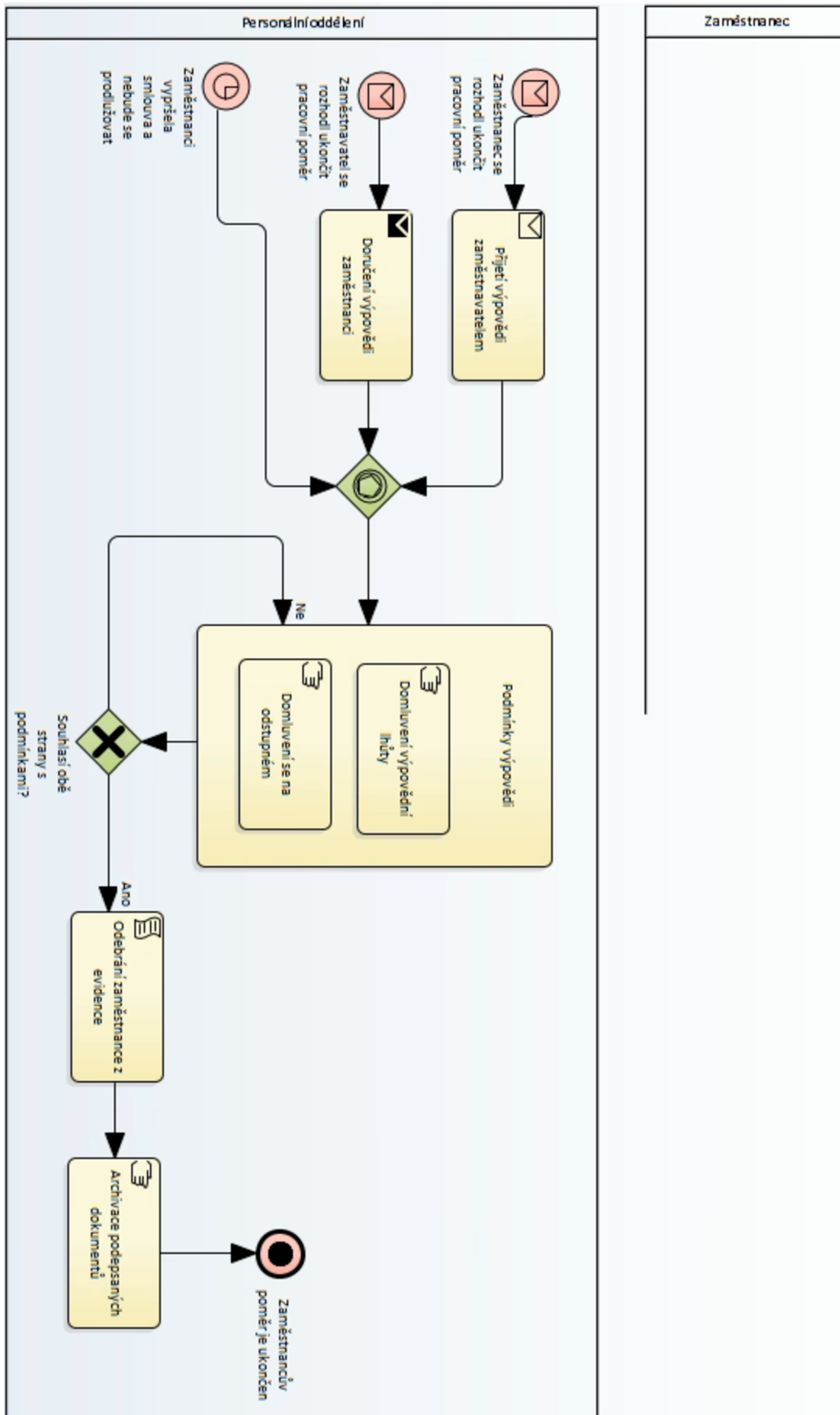
11 Přílohy

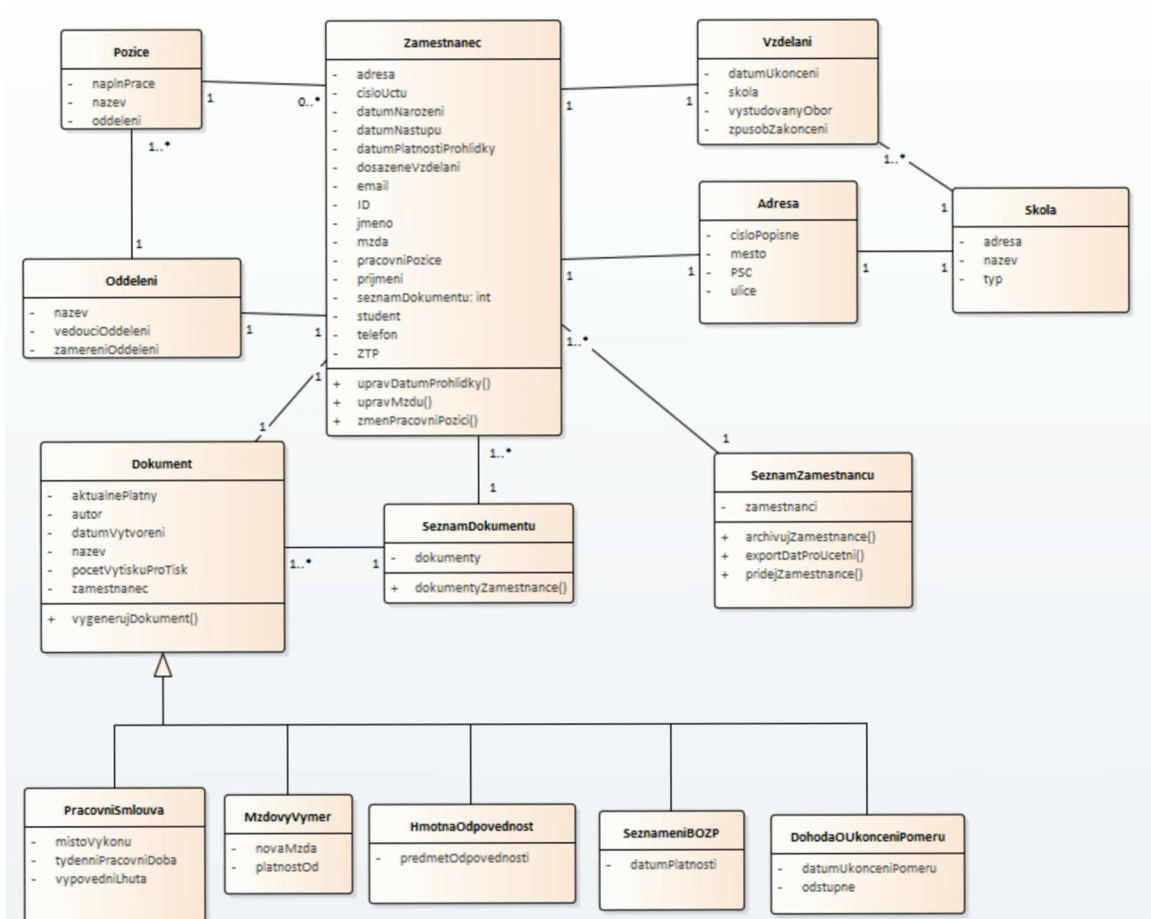
- 1) Proces nástupu zaměstnance
- 2) Proces výstupu zaměstnance
- 3) Analytický model systému
- 4) Návrhový model systému
- 5) Java kód třídy *Zaměstnanec*

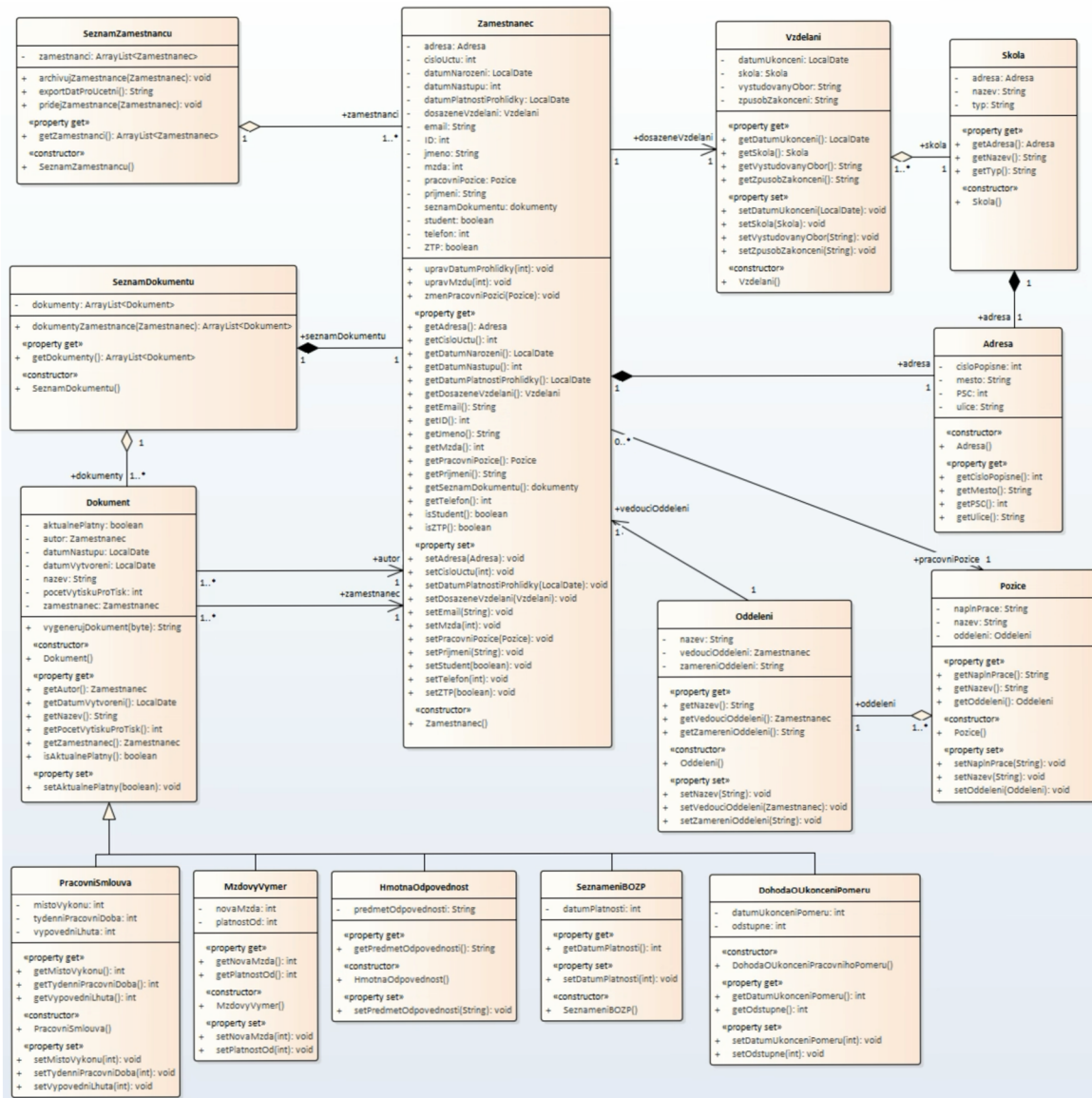
12 Datové přílohy

- 1) Zdrojový soubor Enterprise Architectu (PersInfSys)
- 2) Vygenerované Java kódy pro jednotlivé třídy









```
/**
 * @author lanzhva1
 * @version 1.0
 * @created 23-dub-2023 23:56:13
 */
public class Zamestnanec {

    private Adresa adresa;
    private int cisloUctu;
    private LocalDate datumNarozeni;
    private int datumNastupu;
    private LocalDate datumPlatnostiProhlidky;
    private Vzdelani dosazeneVzdelani;
    private String email;
    private int ID;
    private String jmeno;
    private int mzda;
    private Pozice pracovniPozice;
    private String prijmeni;
    private dokumenty seznamDokumentu;
    private boolean student;
    private int telefon;
    private boolean ZTP;

    public void finalize() throws Throwable {

    }
    public Zamestnanec(){

    }

    public Adresa getAdresa(){
        return adresa;
    }

    public int getCisloUctu(){
        return cisloUctu;
    }

    public LocalDate getDatumNarozeni(){
        return datumNarozeni;
    }

    public int getDatumNastupu(){
        return datumNastupu;
    }

    public LocalDate getDatumPlatnostiProhlidky(){
        return datumPlatnostiProhlidky;
    }

    public Vzdelani getDosazeneVzdelani(){
```

```
        return dosazeneVzdelani;
    }

    public String getEmail(){
        return email;
    }

    public int getID(){
        return ID;
    }

    public String getJmeno(){
        return jmeno;
    }

    public int getMzda(){
        return mzda;
    }

    public Pozice getPracovniPozice(){
        return pracovniPozice;
    }

    public String getPrijmeni(){
        return prijmeni;
    }

    public dokumenty getSeznamDokumentu(){
        return seznamDokumentu;
    }

    public int getTelefon(){
        return telefon;
    }

    public boolean isStudent(){
        return student;
    }

    public boolean isZTP(){
        return ZTP;
    }

    /**
     *
     * @param newVal
     */
    public void setAdresa(Adresa newVal){
        adresa = newVal;
    }

    /**
     *
     * @param newVal
```

```
*/
public void setCisloUctu(int newVal){
    cisloUctu = newVal;
}

/**
 *
 * @param newVal
 */
public void setDatumPlatnostiProhlidky(LocalDate newVal){
    datumPlatnostiProhlidky = newVal;
}

/**
 *
 * @param newVal
 */
public void setDosazeneVzdelani(Vzdelani newVal){
    dosazeneVzdelani = newVal;
}

/**
 *
 * @param newVal
 */
public void setEmail(String newVal){
    email = newVal;
}

/**
 *
 * @param newVal
 */
public void setMzda(int newVal){
    mzda = newVal;
}

/**
 *
 * @param newVal
 */
public void setPracovniPozice(Pozice newVal){
    pracovniPozice = newVal;
}

/**
 *
 * @param newVal
 */
public void setPrijmeni(String newVal){
    prijmeni = newVal;
}

/**
```

```
*
* @param newVal
*/
public void setStudent(boolean newVal){
    student = newVal;
}

/**
*
* @param newVal
*/
public void setTelefon(int newVal){
    telefon = newVal;
}

/**
*
* @param newVal
*/
public void setZTP(boolean newVal){
    ZTP = newVal;
}

/**
*
* @param newVal
*/
public void upravDatumProhlidky(int newVal){

}

/**
*
* @param newVal
*/
public void upravMzdu(int newVal){

}

/**
*
* @param pozice
*/
public void zmenPracovniPozici(Pozice pozice){

}
} //end Zamestnanec
```


Zadání bakalářské práce

Autor:	Václav Lanžhotský
Studium:	I1900123
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Personální informační systémy
Název bakalářské práce AJ:	Human resources information systems

Cíl, metody, literatura, předpoklady:

Cílem bakalářské práce je zmapovat situaci na trhu personálně informačních systémů, zjistit, zda-li aktuální možnosti vyhovují koncovým uživatelům a navrhnout nové, efektivnější softwarové řešení.

Osnova

1. Úvod
2. Dostupné personálně informační systémy
3. Srovnání personálně informačních systémů
4. Popis koncového uživatele a jeho potřeby
5. Výstupy z dotazníku mezi koncovými uživateli
6. Návrh vlastního softwaru
7. Shrnutí výsledků
8. Závěr

Tvorba informačních systémů - Tomáš Bruckner, Jiří Voříšek, Alena Buchalcevoová

Zadávací pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Ing. Tomáš Nacházel, Ph.D.
Datum zadání závěrečné práce:	15.10.2021