



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Informa ní systém personální agentury
Student:	Michal Majer
Vedoucí:	Ing. Dana Vynikarová, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informa ní systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Cílem bakalá ské práce je nasazení informa ního systému do personální agentury AWPartners s.r.o., který nahradí sou asný nevyhovující informa ní systém. Jádro systému bude tvo it modul evidence uchaze o zam stnání a generování dotazník .

1. Analyzujte stávající informa ní systém personální agentury AWPartners s.r.o.
2. Analyzujte business procesy agentury AWPartners s.r.o.
3. Na základ zjišt ných informací definujte uživatelské požadavky na nový informa ní systém.
4. Navrhn te vhodnou metodiku vývoje informa ního systému (vodopád, agilní metodiky).
5. Navrhn te vlastní ešení informa ního systému.
6. Vyberte vhodné vývojové prostředí a technologie.
7. Pomocí vybraných technologií implementujte vámi navržené ešení informa ního systému ve zvoleném vývojovém prostředí.
8. Informa ní systém ádn otestujte a zdokumentujte.
9. Analyzujte rizika a ekonomické dopady nasazení informa ního systému do agentury AWPartners s.r.o.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
řídící kan

V Praze dne 27. listopadu 2015

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



Bakalářská práce

Informační systém personální agentury

Michal Majer

Vedoucí práce: Ing. Dana Vynikarová, Ph.D.

16. května 2016

Poděkování

Chtěl bych poděkovat především vedoucí práce Ing. Daně Vynikarové Ph.D. a zástupcům agentury AW Partners s.r.o za konstruktivní a příjemnou spolupráci. Taktéž děkuji mé rodině, přátelům a přítelkyni za to, že mě podporovali a měli pochopení pro mé časové vytížení.

Prohlášení

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

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

V Praze dne 16. května 2016

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2016 Michal Majer. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Majer, Michal. *Informační systém personální agentury*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato bakalářská práce se zabývá analýzou a vývojem informačního systému pro personální agenturu AW Partners s.r.o. Při analýze požadavků a návrhu architektury systému vychází autor práce z dostupné vědecké literatury zaměřené na tematiku e-recruitment procesů. Pro implementaci byl vybrán PHP Framework Symfony 3.0 a uživatelské rozhraní je realizováno s využitím šablony AdminLTE. Výsledkem práce je fungující informační systém, který je možné využít pro podporu firemních procesů personální agentury. Část práce je věnována procesu testování výsledného informačního systému a nechybí analýza rizik a možných ekonomických dopadů na fungování personální agentury.

Klíčová slova informační systém, personální agentura, e-recruitment, Symfony framework, systém pro evidenci uchazečů

Abstract

The subject of this thesis is the analysis and development of an information system for employment agency AW Partners s.r.o. Scientific literature concerned with the topic of e-recruitment has been applied for the design of system requirements and architecture. The PHP Framework Symfony 3.0 was chosen for the implementation of the information system and the user interface was based on AdminLTE template. A working information system supporting the business processes of an employment agency was produced as a result of this thesis. A part of the work is also designated for the process of testing the system and analysing risks and potential economic consequences of implementation for e-recruitment agency.

Keywords information system, employment agency, e-recruitment, Symfony framework, applicant tracking system

Obsah

Úvod	1
1 Cíl práce	3
2 Popis činnosti personální agentury AW Partners s.r.o	5
2.1 Recruitment	5
2.2 E-recruitment	7
2.3 Zaměstnanecké dotazníky a průzkumy	9
3 Analýza současného systému	11
3.1 Základní informace	11
3.2 Detailní popis funkčnosti	12
3.3 Použité technologie a analýza kódu	14
3.4 Podpora firemních procesů	15
3.5 Shrnutí	16
4 Uživatelské požadavky	19
4.1 Proces definice požadavků	19
4.2 Požadavky investorů	21
4.3 Funkční požadavky	22
4.4 Nefunkční požadavky	23
5 Analýza možných řešení	25
5.1 Rozšíření stávajícího systému	25
5.2 Existující řešení nabízené jako služby	26
5.3 Vývoj nového informačního systému	28
5.4 Závěr analýzy možných řešení	28
6 Návrh a organizace systému	29
6.1 Metodika vývoje	29

6.2	Výběr technologií	31
6.3	Architektura systému	32
7	Realizace iterace 1 - Jádru systému a komponenta přehled	37
7.1	Případy užití	37
7.2	Implementace	37
7.3	Testování	40
7.4	Ověření splnění požadavků	42
8	Realizace iterace 2 - Komponenty databáze uchazečů a nabídka pozic	43
8.1	Případy užití	43
8.2	Implementace	44
8.3	Testování	50
8.4	Ověření splnění požadavků	51
9	Realizace iterace 3 - Komponenta analytika	53
9.1	Případy užití	53
9.2	Implementace	53
9.3	Testování	54
9.4	Ověření splnění požadavků	55
10	Realizace iterace 4 - Komponenta dotazníky	57
10.1	Případy užití	57
10.2	Implementace	57
10.3	Testování	61
10.4	Ověření splnění požadavků	62
11	Analýza rizik a ekonomických dopadů nasazení nového IS	63
11.1	Analýza rizik spojených s nasazením IS	63
11.2	Ekonomické dopady vývoje IS	64
	Závěr	71
	Literatura	73
	A Seznam použitých zkratk	77
	B Ukázka nejdůležitějších funkcí systému	79
	C Obsah příloženého CD	89

Seznam obrázků

2.1	Proces nábor z externích zdrojů	6
2.2	Registrace uchazeče na webových stránkách AW Partners s.r.o	8
2.3	Diagram procesu dotazníkových průzkumů	9
3.1	Editace uchazeče v původním systému AW Partners s.r.o	13
4.1	Proces nábor z externích zdrojů	20
6.1	Komponenty nového systému	33
6.2	Rozdělení funkcí komponent do balíčku Symfony	35
6.3	Umístění balíčkové struktury v rámci projektu Symfony	36
6.4	Struktura obecného balíčku ve složce /src	36
7.1	Přidání nového upozornění do systému	38
7.2	Získání aktivních úkolů daného typu	39
7.3	PHPUnit testy jádra systému	41
8.1	Struktura balíčku ATSBundle	44
8.2	Diagram tříd tvořících datový model uchazeče	45
8.3	Diagram tříd tvořících funkce vyhledávání	46
8.4	Diagram vykreslení vyhledávacího pole v šabloně	47
8.5	Diagram stavů evidovaného kandidáta na pozici	49
8.6	Změna stavu kandidáta na pozici	50
8.7	Struktura PHPUnit testů komponent databáze uchazečů a pozic	51
9.1	Graf vykreslený javascriptovou knihovnou ChartJS	54
10.1	Diagram tříd definujících chování dotazníkových formulářů	58
10.2	Zpracování respondentem vyplněného dotazníku	59
10.3	Zpracování respondentem vyplněného dotazníku	60
10.4	Struktura PHPUnit testů komponenty dotazníků	61

11.1	Porovnání ceny vývoje nového IS oproti SaaS řešení	69
B.1	Úvodní obrazovka nového systému	80
B.2	Vyhledávání v databázi uchazečů	81
B.3	Přidání záznamu nového uchazeče	82
B.4	Karta uchazeče z databáze	83
B.5	Tvorba formuláře dotazníkového průzkumu	84
B.6	Zobrazení výsledků dotazníkového průzkumu	85
B.7	Zobrazení grafů analytické komponenty	86
B.8	Přidání účtu nového uživatele systému	87
B.9	Nástroje pro migraci dat z původní databáze	88

Seznam tabulek

3.1	Dotazník k původnímu IS agentury AW Partners s.r.o	11
7.1	Mapování případů užití z 1. iterace na požadavky	42
8.1	Mapování případů užití ze 2. iterace na požadavky	52
9.1	Mapování případů užití ze 3. iterace na požadavky	55
10.1	Mapování případů užití ze 4. iterace na požadavky	62
11.1	Čas strávený aktivitami procesu softwarového vývoje	65
11.2	Použité hodnoty hodinových sazeb za služby	66
11.3	Celkové náklady na vývoj nového IS	68

Úvod

Tato práce se zabývá analýzou, implementací a nasazením nového informačního systému (IS) s podporou e-recruitment procesů do personální agentury. Toto téma v současné době nabývá na důležitosti, neboť v oboru agenturního zaměstnávání bude brzy docházet k velkým změnám. Tyto plánované změny budou mít podle předběžných informací dramatické dopady na personální agentury a jejich klienty. Aby agentury zvládly do budoucna ustát plánované změny a konkurenční boj, potřebují efektivnější a bezpečnější nástroje pro podporu svých firemních procesů.

Existuje mnoho literárních zdrojů zabývajících se problematikou využití e-recruitment procesů pro zvýšení produktivity při náboru zaměstnanců. Jen velmi málo z těchto prací se ovšem zaměřuje na implementaci těchto poznatků v praxi. Protože vývoj technologií postupuje mílovými kroky, ztratily ale i tyto dřívější pokusy aktuálnost a staly se v dnešních podmínkách nepoužitelnými. Právě absence literatury popisující realizaci informačních systémů pro personální agentury s využitím nově dostupných technologií se mi stala motivací pro zpracování tohoto tématu.

Analýzu a implementaci řešení jsem se rozhodl demonstrovat na případu agentury AW Partners s.r.o, která nyní pro podporu firemních procesů využívá již z mnoha důvodů zastaralý informační systém. Analýza původního systému agentury poukázala na mnoho kritických nedostatků, které jsou v této práci dále analyzovány a je navržen způsob jejich řešení.

Struktura práce se dělí na dvě hlavní části. Prvních 6 kapitol je orientováno na analýzu problémové domény, sběr uživatelských požadavků, analýzu možných řešení a návrh vlastního systému. Právě kapitola 6 zaměřená na návrh tvoří přechod mezi analytickou částí a částí implementační, jenž vychází z dříve zjištěných poznatků a demonstruje, jakým způsobem je možné jich v praxi využít. Kapitoly zaměřené na praktickou realizaci požadavků jsou sdružené do logických celků, které vychází z organizace projektu do iterací. Každá tato kapitola obsahuje seznam případů užití vycházejících z požadavků na systém, ukázkou nejdůležitějších implementačních částí iterace a sekci

věnovanou testování vyvinuté funkcionality. V závěrečné kapitole poskytnu manažersko-ekonomické shrnutí a finanční analýzu mnou navrženého řešení.

Při návrhu architektury nového systému vycházím z několika studií pojednávajících o návrhu architektury holistického systému pro e-recruitment, které publikoval roku 2007 profesor In Lee, Ph.D. z Western Illinois University.

Cíl práce

Cílem této bakalářské práce je analyzovat vnitřní fungování personální agentury AW Partners a na základě zjištěných informací navrhnout nasazení nového informačního systému, jenž nahradí současný nevyhovující systém.

Teoretická část práce bude zaměřena na analýzu a dokumentaci firemních procesů agentury, využití současného systému pro jejich podporu a zjištění uživatelských požadavků na systém nový. Tyto poznatky budou dále využity při návrhu vhodného řešení informačního systému. Následně bude vybrána vhodná vývojová metodika, technologie a vývojové prostředí pro implementaci.

Výsledkem praktické části bude implementace zvoleného řešení pomocí dříve vybraných technologií. K implementaci bude vypracována dokumentace kódu a nápověda k použití nového systému pro zaměstnance personální agentury. Funkce i zdrojový kód systému budou řádně otestovány.

Posledním dílčím cílem práce je vytvořit analýzu rizik a ekonomických dopadů nasazení informačního systému do personální agentury.

Popis činnosti personální agentury AW Partners s.r.o

Personální agentura AW Partners s.r.o nabízí rozmanitou škálu služeb v oblasti řízení lidských zdrojů. Mezi tyto služby se řadí například outsourcing personálních a mzdových služeb, konzultační a poradenská činnost nebo implementace procesů řízení lidských zdrojů do společností. Těmi nejdůležitějšími nabízenými službami z pohledu této bakalářské práce jsou ovšem vyhledání a výběr vhodných kandidátů na pozice (recruitment) a organizace zaměstnaneckých dotazníků a průzkumů. [1]

2.1 Recruitment

Společnosti se mohou obrátit na agenturu AW Partners s.r.o s žádostí o vyhledání a výběr vhodných kandidátů na volné pozice, neboli recruitment. V rámci této služby dojde k vyhledání vhodného kandidáta, posouzení způsobilosti a kompetence pro danou pozici a prověření klíčových znalostí potřebných pro výkon práce.

2.1.1 Proces náboru zaměstnanců

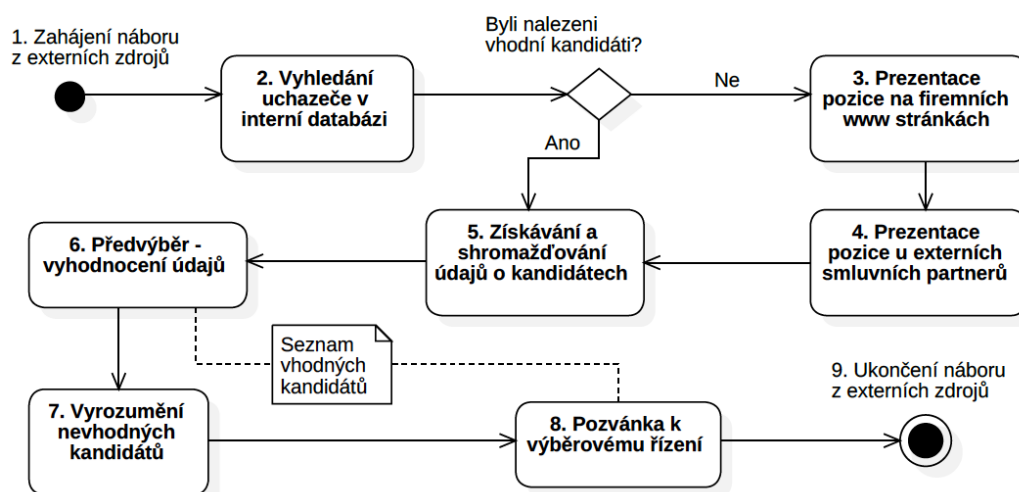
V rámci vykonávání služby náboru zaměstnanců dochází uvnitř agentury k několika navazujícím procesům, které jsou přesně definovány interními předpisy:

1. **Nábor z interních zdrojů** - Hledání vhodného uchazeče na volnou pozici nejprve probíhá uvnitř firmy, která pozici vypisuje.
2. **Nábor z externích zdrojů** - Kandidát se vyhledává v interním systému uchazečů AW Partners, na internetových portálech či skrze inzerci na vlastních webových stránkách.

- Výběr zaměstnanců** - V rámci tohoto procesu dojde k organizačnímu a administrativnímu zabezpečení výběrového řízení, absolvování pohovoru a výběru vhodného kandidáta na pozici.

Pro podporu procesu 1. Nábor z interních zdrojů jsou využívány vždy rozdílné nástroje dle typu zakázky a není možné tyto systémy propojit s interním systémem agentury, mimo jiné také kvůli požadavkům zákona o ochraně osobních údajů. Pro proces 3. Výběr zaměstnanců je postačující využití běžných kancelářských nástrojů. Proto se v rámci této práce se zaměřuji převážně na podporu procesu 3. Nábor z externích zdrojů.

2.1.1.1 Proces náboru z externích zdrojů



Obrázek 2.1: Proces náboru z externích zdrojů

1. Zahájení náboru z externích zdrojů

Proces náboru z externích zdrojů je zahájen v případě nenalezení vhodného uchazeče při procesu náboru z interních zdrojů.

2. Vyhledání v interní databázi uchazečů

Dojde k pokusu o vyhledání uchazeče v interním informačním systému agentury. V případě nalezení vhodných uchazečů se pokračuje krokem 5. Získávání a shromažďování údajů o kandidátech.

3. Prezentace pozice na firemních webových stránkách

Pokud v předešlém kroku nebyli nalezeni vhodní uchazeči na pozici, je pozice zveřejněna na webových stránkách agentury a potenciální kandidáti mohou agenturu kontaktovat.

4. **Prezentace pozice u externích smluvních partnerů**

Po zvážení a s ohledem na maximální efektivitu a náklady je pozice prezentována na serverech smluvních partnerů (např. Jobs.cz, atd.). Potenciální kandidáti se skrze tyto servery mohou hlásit na pozici.

5. **Získání a shromažďování údajů o kandidátech**

Informace o všech uchazečích z kroku 2 a kandidátech získaných z předešlých dvou kroků jsou shromážděny na jednom místě.

6. **Předvýběr - vyhodnocení údajů**

V souladu se strukturovaným profilem pozice dojde k vyhodnocení získaných údajů a posouzení kandidátů na danou pracovní pozici. Výsledkem tohoto kroku je seznam vhodných kandidátů na pozici.

7. **Vyrozumění nevhodných kandidátů**

Kandidáti, kteří nebyli v předešlém kroku vybráni jako vhodní na pozici, jsou informováni e-mailem.

8. **Pozvánka k výběrovému řízení**

Osobám ze seznamu vhodných kandidátů je zaslána pozvánka k výběrovému řízení.

9. **Ukončení náboru z externích zdrojů**

Tímto krokem končí proces 2. Nábor z externích zdrojů, na který přímo navazuje proces 3. Výběr zaměstnanců.

2.2 E-recruitment

E-recruitment lze definovat jako proces náboru zaměstnanců s využitím různých druhů elektronických prostředků a technologií. Převážně se jedná o internetové technologie napomáhající zefektivnit proces náboru automatizací některých jeho činností a poskytováním informací pro podporu strategických rozhodnutí. [2]

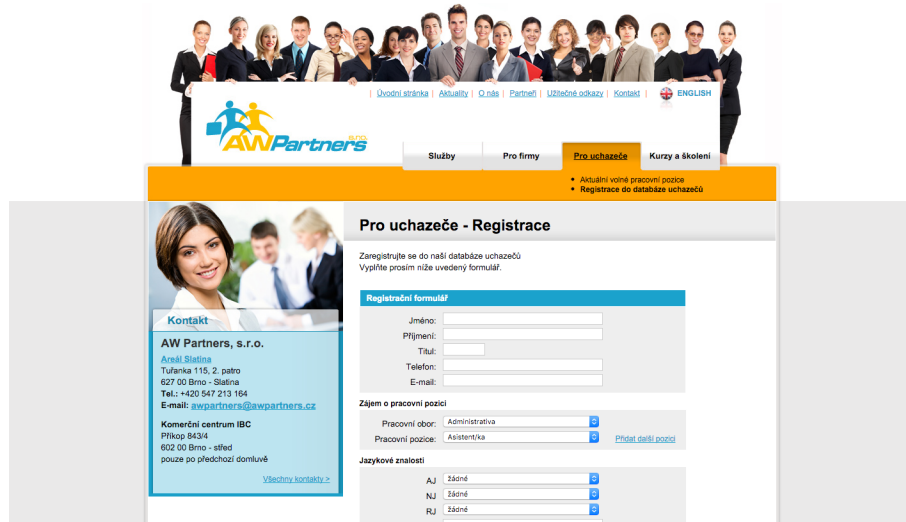
V dnešní době, kdy je na pracovním trhu stále větší nedostatek kvalifikovaných uchazečů, nabývá e-recruitment na důležitosti. Stefan Lang a jeho tým zmiňují využití e-recruitment procesů jako obzvláště důležité pro zvýšení schopnosti společností nabírat kvalifikované zaměstnance. [3]

2.2.1 Využití E-recruitment procesů agenturou AW Partners

Agentura využívá e-recruitment procesů za pomoci stávajícího informačního systému a webových stránek. Webové stránky plní jednak účel jednoduché prezentace nabízených pozic a registrovaných uchazečů o zaměstnání. Co je

2. POPIS ČINNOSTI PERSONÁLNÍ AGENTURY AW PARTNERS S.R.O

ovšem nejpodstatnější, uchazeči mají možnost registrovat se do systému. Na obrázku 2.2 je zobrazen formulář, po jehož vyplnění informacemi o své osobě jsou uchazeči zařazeni do interní databáze agentury. Zaměstnanci agentury pak mohou informovat uchazeče o vhodné pozici, naleznou-li jejich profil v IS při vyhledávání mezi relevantními záznamy.



Obrázek 2.2: Registrace uchazeče na webových stránkách AW Partners s.r.o

2.2.2 Přínosy a rizika zavedení E-recruitment procesů

V publikaci “Drivers, Challenges and Consequences of E-Recruiting” zmiňují Stefan Lang a jeho tým hlavní dopady a výzvy při implementaci e-recruitment systémů. Tyto body budou v následujících kapitolách využity pro ohodnocení současného IS agentury. Níže uvedu body relevantní pro e-recruitment systém agentury AW Partners s.r.o. Záměrně neuvádím body, které hodnotí využití firmami pro nábor do vlastních pozic, čímž nejsou relevantní pro potřeby personální agentury. [3]

2.2.2.1 Pozitivní dopady:

1. snížení nákladů na nábor a selekci,
2. zvýšení počtu vhodných kandidátů,
3. úspora času pro firmu i uchazeče.

2.2.2.2 Negativní dopady:

1. Drahé náklady na vývoj a údržbu systému,

2. potenciální nebezpečí zneužití osobních dat uchazečů,
3. vnímaná ztráta individuality ze strany uchazečů.

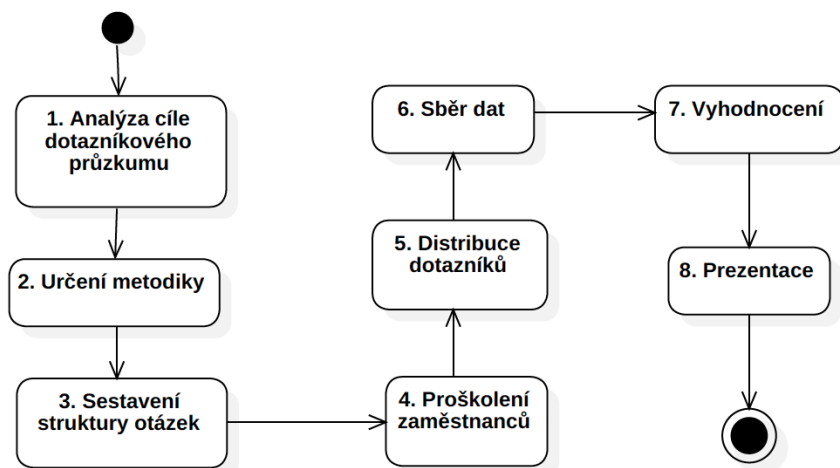
2.2.2.3 Výzvy:

1. Implementace efektivních metod pro vyhledávání vhodných kandidátů,
2. identifikace všech potřebných parametrů pro ohodnocení uchazeče,
3. dostatečné zabezpečení systému.

2.3 Zaměstnanecké dotazníky a průzkumy

Firmy se mohou na agenturu AW Partners s.r.o obrátit s požadavkem na zjištění názorů svých zaměstnanců. Agentura se zaměřuje na průzkum spokojenosti zaměstnanců, zjištění zájmu o zaměstnanecké benefity a identifikace vzdělávacích potřeb zaměstnanců. Hlavní předností této služby je anonymita garantovaná ze strany agentury. Osoby vyplňující dotazník si proto mohou být jisté, že se jejich konkrétní odpovědi nedostanou k vedení firmy.

2.3.1 Proces dotazníkových průzkumů



Obrázek 2.3: Diagram procesu dotazníkových průzkumů

Na obrázku 2.3 je znázorněn firemní proces, jímž se agentura řídí při poskytování služby dotazníkových průzkumů. Body 3 a 5-8 jsou podporovány současným informačním systémem, jenž se stará o distribuci a zpracování výsledků rozeslaných dotazníků.

Analýza současného systému

Společnost AW Partners nyní využívá pro podporu e-recruitment procesů a procesu dotazníkových průzkumů jednoduchý webový informační systém vyvinutý na zakázku v roce 2011. Následující podsekcce se zaměřují na analýzu systému z pohledu uživatelů, zdrojového kódu a firemních procesů. Závěrem je systém zhodnocen na základě kritérií pro e-recruitment systémy, tak jak byly definovány v předešlé kapitole.

3.1 Základní informace

Pro zjištění základních informací o parametrech IS vyplnili zástupci společnosti AW Partners jednoduchý dotazník. Tabulka 3.1 shrnuje výstup dotazníku. Požadavky agentury na změny systému jsou dále specifikovány v sekci 4.2 Požadavky investorů.

Tabulka 3.1: Dotazník k původnímu IS

Počet souběžně aktivních uživatelů systému	10-15
Uživatelské role	Nejsou. Uživatelům jsou operativně nastavována přístupová práva.
Doba využívání systému	Po-Ne, 5hod. ráno do 2hod. v noci
Použitelnost a přehlednost systému (hodnocení 1-5)	3
Plánované změny ve fungování agentury, které by mohly ovlivnit užívání systému	Nejsou do budoucna plánovány.
Systémová podpora dalších služeb nabízených agenturou	Není vyžadována, jsou pro tento účel využívány jiné nástroje.

3.2 Detailní popis funkčnosti

Systém je řazen do čtyř hlavních logických komponent (modulů) a ty jsou dále rozděleny na submoduly plnící konkrétní funkce podporující e-recruitment a dotazníkové průzkumy.

3.2.1 Uchazeči

Následující submoduly jsou určeny pro práci s databází uchazečů o zaměstnání. Jedná se o nejdůležitější a také nejvyužívanější část celého systému.

1. **Databáze uchazečů** - Tento submodul umožňuje vyhledávat, přidávat, upravovat a mazat záznamy uchazečů. Ve výpisu se mimo jiné zobrazuje informace o tom, za kolik dní musí uchazeč znovu udělit souhlas se zpracováním osobních údajů. Dále je zde funkce pro zobrazení e-mailových adres vybraných uchazečů, export karty uchazeče do PDF (Portable Document Format) souboru a možnost zobrazit záznam na webových stránkách. Informace zaznamenávané o konkrétním uchazeči jsou: jméno, příjmení, titul, pozice, obor, telefon, e-mail, město, kraj, země, dodatečné informace, klíčová slova, mzdové požadavky, jazykové dovednosti (angličtina, němčina, ruština a jiné), vzdělání, praxe, certifikace, pc gramotnost a přiložené soubory. Na obrázku 3.1 je vidět stránka systému umožňující editaci zmíněných údajů o uchazeči. Za zmínku stojí především uchovávání informací o dříve nabídnutých pozicích tomuto uchazeči, jimiž jsou datum, zaměstnavatel, nabízená pozice a výsledek této nabídky.
2. **Seznam oborů (číselník)** - Submodul pro správu oborů, které je možné přiřazovat k uchazečům. Jediným parametrem je název oboru.
3. **Seznam pozic (číselník)** - Submodul umožňující správu pozic, které je možné přiřazovat k uchazečům. Parametry jsou název pozice a nadřazený obor.
4. **Povolené země (číselník)** - Submodul umožňující správu povolených zemí, které je možné k uchazečům přiřadit.

3.2.2 Dotazníky

Komponenta dotazníky podporuje tvorbu, úpravu, mazání a rozesílání dotazníkových průzkumů.

1. **Seznam dotazníků** - Submodul umožňující správu vybraných dotazníků. Do dotazníku lze přidat otázky a předvyplnit možné odpovědi, které jsou vždy zobrazeny ve formátu přepínače. Ke každému dotazníku je nutné nastavit název, úvodní text a případně vybrat logo společnosti, pro kterou je dotazník tvořen. Submodul umožňuje buďto nastavit počet

3.2. Detailní popis funkčnosti

Administrace stránek Uchazeči Dotazníky Nástroje Přihlášen: elhcyr Odhlásit se

Uchazeči - V této sekci se nacházejí veškeré nástroje pro práci s databází uchazečů.

Databáze uchazečů Seznam pozic Povolené země Seznam oborů

Úprava uchazeče Úprava vybraného uchazeče z databáze uchazečů.

Zpět Zobrazit na webu Nabídka pozic

Editace položky

Jméno: Michal

Příjmení: Majer

Obor: Administrativa

Město: Brno

Pozice
Neobsahuje žádná klíčová slova.

Kraj: Jihomoravský kraj

Nabídnuté pozice:

Datum	Zaměstnavatel	Pozice	Výsledek
25/1/2016		Sales/Marketing Koordinator s NJ	nevhodná profilem

Jazykové dovednosti:

Angličtina: pokročilý

Němčina: Nevyplněno

Ruština: pokročilý

Jiné jazykové znalosti:

Vzdělání: Vysokoškolské - univerzitní

Praxe

Přiložené soubory

Přidání souboru: Choose File No file chosen

Již přidáné:
Nebyl nahrán žádný soubor.
Datum posledního souhlasu se zpracováním OU
02/10/2016

Klíčová slova
Neobsahuje žádná klíčová slova.

Mzdové požadavky
30000

Certifikace

PC Gramotnost
MS Office

Obrázek 3.1: Editace uchazeče v původním systému AW Partners s.r.o

respondentů a vygenerovat společný hypertextový odkaz na formulář, nebo je možné vyplnit e-mailové adresy respondentů a tím je zajištěna věrohodnost výsledků. Lze sledovat stav vyplnění dotazníků a průběžné výsledky, které je také možné tisknout. V neposlední řadě je zde velmi specifická funkce pro spojení výsledků několika otázek do jedné kategorie, čímž dojde k agregaci odpovědí na několik otázek pod tuto kategorii.

3.2.3 Administrace stránek

Tato komponenta obsahuje submoduly pro správu informací zobrazených na webových stránkách.

1. **Volné pozice** - Správa aktuálně nabízených pozic. Vyplněný název a popis pozice se zobrazí na webových stránkách.

2. **Práci hledají** - Správa uchazečů zobrazených na webových stránkách. K vyplnění jsou zde pole jméno (nepovinné, nikde se nezobrazí) a obsah (obsahuje základní údaje o uchazeči).

3.2.4 Nástroje

1. **Přístup** - Submodul pro správu uživatelů systému a udělování přístupových práv. Kromě nastavení uživatelského jména a hesla je možné operativně povolovat či omezovat uživatelům přístup do vybraných submodulů informačního systému.
2. **Nápověda** - Submodul obsahující popis fungování některých částí systému.

3.3 Použité technologie a analýza kódu

V současnosti využívaný informační systém agentury je implementován jako webová aplikace za použití následujících technologií a knihoven:

- PHP 5.5/MySQL,
- HTML/CSS/JS (jQuery 1.3.1),
- CKEditor 3.1.

Systém není napojen na žádný populární PHP (Hypertext Preprocessor) framework a je navržen v čistém PHP kódu za pomoci několika tříd zamezujících zbytečné redundanci zdrojového kódu. Nevyužití frameworku ovšem v případě tohoto systému znamená absenci podpory standardních návrhových vzorů, zhoršené podmínky pro rozšiřitelnost a testování. V neposlední řadě se pak zvyšují bezpečnostní rizika, jelikož systém neobsahuje podporu pro automatické testování.

3.3.1 Nalezené vady zdrojového kódu

Po důkladné analýze zdrojového kódu a fungování systému se podařilo nalézt následující závady:

1. **Veřejně přístupné dokumenty** - Nejzávažnějším nalezeným prohřeškem systému je možnost veřejného přístupu k dokumentům uloženým v systému. Odkazy na tyto dokumenty jsou sice přístupné pouze uživatelům systému, se znalostí přesné adresy jsou ovšem přístupné i po odhlášení uživatele ze systému. Stejná chyba se vyskytuje u všech exportů do formátu PDF, díky čemuž je systém potenciálně zneužitelný kteroukoli osobou, která v minulosti se systémem pracovala.

2. **Souběžný přístup k záznamům** - V celém systému, při úpravě kteréhokoli záznamu může dojít k situaci, kdy k tomuto stejnému záznamu přistupuje několik uživatelů současně. Vzájemný přístup ovšem není systémem nijak ošetřen a pokud upravují dva uživatelé tentýž záznam, přepíše se úpravy prvního uživatele úpravami toho, kdo uložil záznam jako poslední.
3. **Ošetření logických závislostí** - Systém nedefinuje žádnou akci pro případ, odstraní-li uživatel hodnotu z některého z číselníků (např. pozice, obory, atd...) a tato hodnota je přitom v systému napojena na jiné aktivní záznamy. Tento nedostatek může vést k nepředpokládanému chování systému.
4. **Implementace kritických operací pomocí odkazů** - Operace pro smazání záznamů v systému (např. smazání uchazeče z databáze) jsou implementovány pomocí hypertextových odkazů. Případný útočník by mohl této znalosti zneužít a podstrčit uživateli systému tento odkaz, čímž dojde k nechtěnému smazání záznamu.
5. **SSL (Secure Sockets Layer) certifikát** - Při komunikaci se systémem není využíváno zabezpečeného připojení. Aby bylo připojení šifrované, je nutné zakoupit SSL certifikát a následně přistupovat primárně přes HTTPS (Hypertext Transfer Protocol Secure) protokol.

3.4 Podpora firemních procesů

Informační systém je agenturou využíván pro podporu procesu 2. Nábory z externích zdrojů a pro uchovávání výstupních dokumentů procesu 3. Výběr zaměstnanců. Dále poskytuje potřebné nástroje pro všechny kroky procesu dotazníkových průzkumů.

Po zahájení procesu nábory z externích zdrojů provádí zaměstnanci agentury vyhledání potenciálních uchazečů o zaměstnání. K vyhledání jsou použity nástroje submodulu Databáze uchazečů, které umožňují filtraci výsledků vyhledávání podle všech parametrů souvisejících se záznamem uchazeče. V případě nenalezení vhodných uchazečů je využit submodul nabízené pozice pro zveřejnění nabídky na webových stránkách. V případě nabídnutí pozice uchazeči z databáze je dále u záznamu tohoto uchazeče zaznamenávána jeho/její reakce a výsledek výběrového řízení, byl-li do něj kandidát připuštěn.

Následující body představují mé návrhy na změny systému pro lepší podporu procesu nábory, které byly schváleny pro implementaci po diskuzi s agenturou AW Partners.

1. Umožnit registraci uchazečů na vybranou pozici, která byla zveřejněna na webových stránkách v kroku 3 procesu. Jako následek tohoto kroku

3. ANALÝZA SOUČASNÉHO SYSTÉMU

dojde ke zjednodušení shromažďování informací v 5. kroku procesu, jelikož bude možné v systému zobrazit seznam kandidátů na pozici.

2. Přidat funkci pro nabídnutí pracovní pozice uchazeči z databáze, čímž dojde k rozšíření seznamu z předešlého bodu a taktéž se zjednoduší administrativní práce související s krokem 5.
3. Pro podporu kroku 6. předvýběr přidat možnost měnit stav předvýběru (např. vhodný, zamítnut, atd...) u seznamu zavedeném předešlými dvěma body.
4. Implementovat možnost vygenerování e-mailu zamítnutým kandidátům při změně stavu z předešlého kroku. Tím dojde k snížení administrativní zátěže při kroku 7. vyrozumění nevhodných kandidátů.
5. Umožnit po obsazení pozice automatický zápis výsledku a důvodu pro odmítnutí ke všem kandidátům, kteří byli v seznamu uchazečů.

3.5 Shrnutí

V následujících podsekcích shrnuji kladné a záporné stránky současného informačního systému a využiji přitom poznatků z předešlé sekce 2.2.2 Přínosy a rizika zavedení e-recruitment procesů.

3.5.1 Kladné stránky

Stávající informační systém agentury AW Partners sebou bezesporu přinesl všechny 3 pozitivní dopady zavedení e-recruitment systému do společnosti podle sekce 2.2.2. Jelikož agentura dříve žádný systém nepoužívala, byly zavedením systému sníženy jak finanční, tak časové náklady na výběr uchazeče a s možností registrace na stránkách se zvýšil počet vhodných kandidátů.

3.5.2 Záporné stránky

Přestože systém přinesl zmíněné kladné dopady pro agenturu, nepodařilo se zavedením původního systému předejít těm negativním. Kvůli návrhu architektury, která se neřídí žádnými běžně používanými standardy je údržba systému velmi složitá a finančně náročná. Analýzou zdrojových kódů jsem našel mnoho bezpečnostních rizik, která mohou vést ke zneužití dat a žádným opatřením není ošetřena vnímaná ztráta individuality ze strany uchazečů.

Především jako důsledek slabého zabezpečení lze prohlásit, že nebyly zvládnuty výzvy na vývoj e-recruitment systému podle sekce 2.2.2.3. Nové požadavky ze strany agentury na zlepšení vyhledávacích nástrojů a změnu parametrů pro ohodnocení uchazeče zmíněné v sekci 4.3.1 kapitoly požadavků pak znamenají, že ani zbylé body nebyly naplněny.

Zmíněné bezpečnostní nedostatky představují závažný problém, který je nutné neprodleně řešit. Ministerstvo práce a sociálních věcí vyšlo vstříc odborovým organizacím a začalo počátkem roku 2016 plánovat razantní změny, která mají přinést vyšší míru regulace a zavedení přísnějších kontrol firem v tomto oboru. Pokud má být jakákoliv personální agentura připravena na tyto změny, které podle mnohých budou pro obor agenturního zaměstnávání velmi bolestné, musí být její nástroje nejen efektivní, ale také dostatečně zabezpečené. [4]

Uživatelské požadavky

Dne 9. února 2016 jsem se sešel se zástupci agentury AW Partners s.r.o., abychom společně zahájili proces sběru požadavků na úpravy informačního systému. V této kapitole vysvětlím použitý proces definice požadavků, následně představím výstupy zmíněné schůze a provedu jejich analýzu.

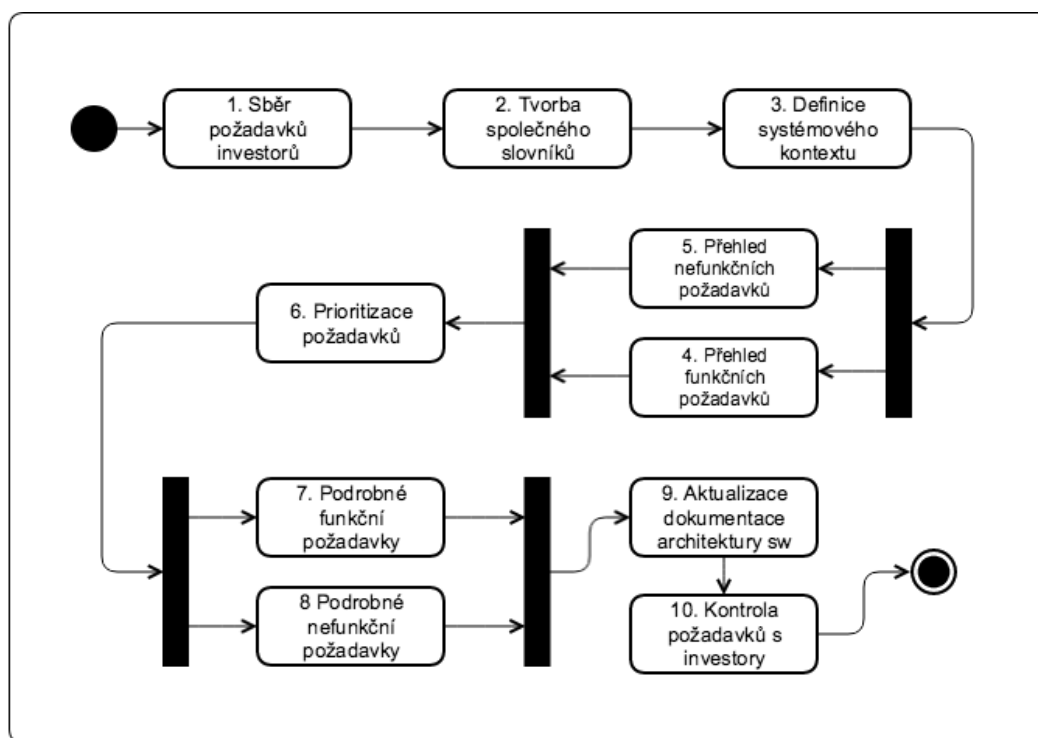
Aby tato práce zachovávala kontinuitu s realizací projektu a zároveň samotná tato kapitola nepřesáhla počtem stran požadovaný rozsah bakalářské práce, je dokument specifikace požadavků SRS (Structured requirements specification) přiložen na CD ve složce Documentation. Tento dokument je vypracován v souladu se standardem organizace pro standardizaci softwarového procesu IEEE 29148-2011. Peter Eeles ve své knize Architektura softwaru zmiňuje, že *“... definice požadavků není ničím, co by proběhlo pouze jednou, protože obvykle není možné předem plně porozumět všem požadavkům a zdokumentovat je”* [5]. To znamená, že se v průběhu projektu i dokument požadavků průběžně rozšiřuje, a proto se na něj budu v práci odkazovat nejenom v této kapitole, ale také v kapitolách věnovaných různým fázím vývoje.

4.1 Proces definice požadavků

Pro sběr požadavků jsem využil proces popsany Peterem Eeeseem, jenž je zobrazen na obr 4.1 a popisuje činnosti prováděné v rámci jedné iterace sběru požadavků. Způsob, jakým jsem plnil jednotlivé kroky tohoto procesu, popisují následující body.

1. **Sběr požadavků investorů** - Požadavky investorů byly specifikovány na společné schůzi se zástupci agentury AW Partners a jsou uvedeny v následující sekci 4.2 požadavky investorů.
2. **Tvorba společného slovníku** - Časté obchodní a technické pojmy byly uvedeny v dokumentu specifikace požadavků v sekci 1.3 Slovníček, aby si obě strany při komunikaci rozuměli.

4. UŽIVATELSKÉ POŽADAVKY



Obrázek 4.1: Proces nábor z externích zdrojů

3. **Definice systémového kontextu** - Rozsah zamýšlených změn v informačním systému byl na schůzi definován a následně byla vytvořena vize budoucí logické struktury IS (viz sekce 6.3 Architektura Systému).
4. **Přehled funkčních požadavků** - Po společné schůzi byla vytvořena první verze dokumentu požadavků s přehledem funkčních požadavků. Klíčové funkční požadavky jsou zmíněny v následující sekci 4.3 Funkční požadavky.
5. **Přehled nefunkčních požadavků** - Podobně jako v případě předchozího bodě jsou nefunkční požadavky zmíněny v následující sekci 4.4 Nefunkční požadavky.
6. **Prioritizace požadavků** - Po diskuzi se zástupci agentury byla přiřazena požadavkům priorita na škále 1 (Nezbytné) až 3 (Nice-to-have).
7. **Podrobné funkční požadavky** - Před každou fází implementace došlo k opětovné kontrole specifikace požadavků a jejich finální verze byla zaznamenána v příloženém dokumentu SRS.
8. **Podrobné nefunkční požadavky** - viz předchozí bod.

9. **Aktualizace dokumentace architektury systému** - Návrh architektury systému byl v průběhu celého projektu uchovávan aktuální a jeho finální verze je popsána v kapitole 6 Návrh a organizace systému.
10. **Kontrola požadavků s investory** - Ihned po zhotovení první verze byl SRS dokument zaslán agentuře AW Partners a dne 15.2. 2016 došlo k jeho schválení. Dále před každou fází vývoje docházelo k doplnění a schválení dokumentu specifikace.

4.2 Požadavky investorů

Společné konzultace s vedením agentury, má analýza původního systému a odhalení bezpečnostních chyb vedly ke stanovení následujících požadavků ze strany AW Partners s.r.o.

1. Dojde k odstranění všech nalezených bezpečnostních chyb.
2. Nástroje pro vyhledávání v databázi uchazečů budou upraveny tak, aby byly snadněji použitelné a obsahovaly nově specifikované parametry.
3. Karta uchazeče bude přepracována s důrazem na větší přehlednost, použitelnost a některé parametry budou upraveny. Také bude přidána možnost nabídky pozice uchazeči.
4. Uživatelské rozhraní pro rozesílání dotazníkových průzkumů bude intuitivnější.
5. Budou přidány nástroje pro sledování a grafickou vizualizaci některých vybraných metrik.
6. Získávání souhlasu se zpracováním OÚ (osobních údajů) od uchazečů z databáze bude automatizováno.
7. Systém bude podporovat registraci uchazečů z webových stránek na konkrétní nabízené pozice.
8. Všechny nezmíněné funkce původního systému budou ponechány.
9. Všechny výše provedené změny budou implementovány s důrazem na možný budoucí rozvoj.

Funkční a nefunkční požadavky zmíněné v následujících sekcích jsou detailní specifikací požadavků investorů z této sekce.

4.3 Funkční požadavky

Kompletní seznam funkčních požadavků lze nalézt v sekci 3.3 přílohy SRS. Zde jsou nyní detailně rozepsány především požadavky měnící fungování původního systému.

4.3.1 Vyhledávání v databázi uchazečů

V databázi uchazečů bude možné vyhledávat přes agenturou specifikované parametry. Oproti původní verzi budou některá pole vyhledávání odebrána a jiná naopak přidána. Nově tak půjde vyhledávat pomocí polí: Příjmení, e-mail, obor, pozice, dosažené vzdělání, místo výkonu práce, místo bydliště, jazykové dovednosti, titul, telefon, mzdové požadavky, praxe a ostatní informace. Mimo jiné přibude pole pro vyhledání přes všechny parametry záznamu uchazeče a také přes dříve nabídnuté pozice tomuto uchazeči. Vyhledávač bude podporovat spojování vyhledávacích příkazů pomocí logických operací konjunkce a disjunkce.

4.3.2 Správa uchazečů

Pro pohodlnější práci s databází uchazečů bude zobrazení uchazeče rozděleno na dvě obrazovky. První obrazovka bude zobrazovat údaje takovým způsobem, aby na klasickém monitoru (od 1200px x 900px) byly viditelné všechny potřebné údaje bez nutnosti posunu kolečkem myši. Druhá obrazovka bude sloužit pro editaci záznamů, podobným způsobem jako v původním systému. Oproti původnímu systému však budou efektivněji implementovány formulářová pole pro výběr několika hodnot současně.

4.3.3 Rozesílání dotazníků

Způsob rozesílání dotazníků bude oproti původnímu systému zjednodušen tím, že uživatelé nebudou mít kontrolu nad fungováním funkcí pro zajištění unikátnosti vyplnění dotazníku jednou osobou. Systém bude po uživatelích vyžadovat pouze zadání e-mailových adres příjemců průzkumu, na které se rozešle předpřipravený e-mail s unikátním odkazem na dotazník.

4.3.4 Sledování vybraných metrik

Systém bude uživatelům poskytovat grafické vizualizace vybraných metrik. Sledovanými metrikami budou:

- počet nově registrovaných uchazečů v časovém období,
- změny velikosti databáze uchazečů v časovém období,
- distribuce úrovně vzdělání mezi uchazeči,

- nejpočetněji zastoupené obory, pozice, města a země mezi uchazeči,
- počet nově přidáných volných pozic v časovém období.

4.3.5 Automatická obnova souhlasu ze zpracováním OÚ

Systém umožní bez nutnosti obsluhy uživatelem zajištění obnovy souhlasu se zpracováním osobních údajů uchazečů z databáze. V časovém intervalu deseti dní před vypršením původního souhlasu bude uchazeči odeslán automaticky vygenerovaný e-mail s možností kliknutí na odkaz obnovující souhlas se zpracováním OÚ agenturou AW Partners. Pokud nebude udělen souhlas, zobrazí se záznam uchazeče v seznamu uchazečů pro odebrání z databáze.

4.3.6 Správa nabízených pozic

Původní funkce pro správu nabízených pozic zobrazených na webových stránkách bude rozšířena o možnost zpracování registrace kandidátů na tyto konkrétní pozice. Registrovaní kandidáti budou zobrazeni v seznamu a uživatel systému bude mít možnost měnit u jejich záznamů stav (akceptován, odmítnut, atd..). Dále bude zobrazen seznam osob z databáze uchazečů, jímž byla pozice nabídnuta. U záznamů uchazečů bude taktéž možné měnit stav (přijal nabídku, odmítl nabídku, atd..). Po ukončení nábory na pozici bude mít uživatel systému možnost vybrat přijaté a zamítnuté kandidáty a přidat k výsledkům vlastní poznámku.

4.4 Nefunkční požadavky

Kompletní seznam nefunkčních požadavků lze nalézt v sekci 3.4 přílohy SRS. Zde jsou detailně rozepsány především požadavky měnící chování původního systému.

4.4.1 Zvýšení zabezpečení

Všechny nalezené bezpečnostní chyby původního systému zmíněné v sekci 3.3.1 Nalezené vady zdrojového kódu budou ošetřeny. Především bude přístup do systému povolen pouze přes zabezpečené připojení pomocí HTTPS protokolu. Dále bude ošetřen souběžný přístup ke stejnému záznamu. Pouze přihlášení uživatelé budou moci přistupovat k interním dokumentům. Při mazání hodnot z číselníků bude uživatel upozorněn na možné logické závislosti a důsledky této akce. Kritické akce (např. mazání záznamů) budou implementovány pomocí formulářových tlačítek, aby nebylo možné podstrčit nebezpečný odkaz uživatelům systému.

4.4.2 Rozšiřitelnost

Všechny výše zmíněné funkční požadavky budou provedeny s důrazem na možnou rozšiřitelnost do budoucna. Například půjde rozšiřovat vyhledávání o nová kritéria či záznamy uchazečů a pozic o další parametry.

Analýza možných řešení

Pro uspokojení všech požadavků z předešlé kapitoly lze zvažovat tři možná řešení - rozšíření stávajícího systému, využití existujících řešení nebo návrh nového informačního systému. V této kapitole analyzuji všechny zmíněné možnosti a na základě zjištěných informací určím tu nejvhodnější variantu.

5.1 Rozšíření stávajícího systému

Jelikož je většina uživatelských požadavků charakteru úprav původního informačního systému agentury, je řešení rozšířením původního systému první zvažovanou variantou.

5.1.1 Výhody rozšíření stávajícího systému

- Je možné se soustředit převážně na implementaci požadovaných změn.
- Zaměstnanci firmy jsou po více než pěti letech zvyklí na používání původního systému, což ulehčí náklady na zaškolení a uživatelskou dokumentaci.

5.1.2 Nevýhody rozšíření stávajícího systému

- I přes nápravu všech nalezených bezpečnostních nedostatků nemusí být systém bezpodmínečně bezpečný.
- Časová a technická náročnost zajištění úprav bude podobná času potřebnému na vývoj nového systému kvůli špatnému návrhu architektury původního systému.
- Složitě zajištění požadavku na snadnou rozšiřitelnost implementovaných funkcí kvůli chybějícím podpurným nástrojům v důsledku toho, že není využit PHP framework.

- Bez dostupnosti standardních návrhových vzorů v implementaci architektury nebude snadné zajistit kvalitní objektový návrh prováděných úprav a nové funkcionality.
- Systém neposkytuje žádnou podporu pro provádění automatického testování.

5.1.3 Hodnocení

Přestože by se mohlo zdát na první pohled řešení rozšířením stávajícího systému jako logické, byly identifikovány vážné nedostatky původního systému, které budou případnou implementaci značně komplikovat. Výhody této možnosti jsou v porovnání s nevýhodami velmi zanedbatelné.

5.2 Existující řešení nabízené jako služby

Na internetu je k nalezení mnoho systémů pro podporu e-recruitment procesů, které jsou poskytovány formou služby za měsíční poplatek. Z těchto SaaS (Software as a service) řešení se mi podařilo nalézt tři zástupce, které nejlépe odpovídají svou funkčností požadavkům agentury. Následující seznam obsahuje detailnější popis těchto systémů:

- **iKrut.com**
E-recruitment online systém iKrut v základní zdarma dostupné verzi poskytuje systém pro přihlašování kandidátů na nabízené pozice, správu uchazečů i pozic, základní statistiky, hromadné rozesílání e-mailů odmítnutým kandidátům a mnoho dalších funkcí. Pro naplnění některých požadavků by ovšem bylo zapotřebí zakoupit alespoň verzi Premium a používání systému by se tím výrazně prodražilo. [6]
- **Recruiterbox.com**
Online systém zjednodušující proces náboru zaměstnanců. Umožňuje spravovat a zveřejňovat pracovní pozice, uchovávat informace o uchazečích a kandidátech a navíc silně podporuje týmovou spolupráci uživatelů. V základní verzi je systém poskytován zdarma, pro potřeby agentury AW Partners by ovšem bylo nutné zakoupení verze PRO v roční ceně 145 tisíc Kč. [7]
- **Qandidate.com**
Náborový systém Qandidate je funkcemi velmi podobný oběma předšlým řešením, je ovšem v plné verzi plně zdarma a jako zdroj financování využívají autoři projektu zpoplatněné výukové semináře. Oproti konkurenci je navíc Qandidate velmi uživatelsky přívětivý a disponuje propracovanými nástroji pro vyhledávání databáze uchazečů. [8]

Každý z výše zmíněných systémů je mírně specifický svou implementací funkcionality pro podporu e-recruitment procesů, výhody a nevýhody využití existujícího řešení poskytovaného formou služby však zůstávají pro všechny tyto systémy stejné.

5.2.1 Výhody existujících řešení

- Všechny výše zmíněné e-recruitment systémy je možné zřídit zdarma.
- Během několika minut může být systém agenturou využíván.
- Zabezpečení systému je zajišťováno vývojovými týmy, které systém nepřetržitě zdokonalují.
- Je dostupných mnoho užitečných funkcí, které je nepraktické implementovat do vlastního systému pro jejich málo časté využití.

5.2.2 Nevýhody existujících řešení

- Systémy nelze volně upravovat a proto nepůjde uspokojit některé z požadavků AW Partners s.r.o, např. možnost implementace komplexních vyhledávacích nástrojů odrážející potřeby zaměstnanců agentury.
- Systémy nelze využít pro podporu dalších poskytovaných služeb nesouvisejících s e-recruitmentem.
- Napojení nabízených pozic a zveřejňování profilů uchazečů na vlastních webových stránkách bude vyžadovat implementaci funkcí pro import dat z těchto systémů.
- Agentura nezíská vlastnická práva k systému.
- Využití všech potřebných funkcionalit prvních dvou systémů vyjde z dlouhodobého hlediska zcela jistě draž než zřízení vlastního systému.
- Žádný ze systémů nenabízí podporu českého jazyka a jejich využití proto bude zhoršovat použitelnost v českém prostředí.

5.2.3 Hodnocení

Přestože se může z krátkodobého hlediska zdát využití online e-recruitment systémů jako finančně a časově výhodné řešení, z dlouhodobého pohledu tomu tak být nemusí. Nemožnost přizpůsobení softwaru nabízeného formou služby je častý problém, který vede firmy k rozhodnutí vyvíjet vlastní systémy. [9] Tyto i ostatní zmíněné nevýhody těchto systémů proto v případě řešení systému agentury AW Partners znemožňují jejich využití.

5.3 Vývoj nového informačního systému

Jako poslední analyzuji variantu vývoje nového informačního systému. Již bylo zmíněno, že není možné využít řešení zakoupení systému jako služby, a proto je nyní důležité porovnat vývoj nového systému oproti rozšíření původního IS.

5.3.1 Výhody vývoje nového systému

- Použitím PHP frameworku a následováním standardního vývoje ve zvoleném prostředí bude snadným způsobem zajištěna požadovaná bezpečnost a testovatelnost.
- Implementace úprav bude daleko snazší, protože s nimi bude od začátku návrhu počítáno.
- Kromě částí systému podléhajících úpravám se zlepší použitelnost celého systému.
- Budoucí rozšíření systému bude možné provést s menšími časovými a finančními náklady.

5.3.2 Nevýhody vývoje nového systému

- Vyšší finanční a časové náklady na vývoj v krátkodobém časovém horizontu.
- Vyšší časové náklady na zaškolení zaměstnanců agentury po zavedení nového systému.

5.3.3 Hodnocení

Varianta vývoje nového informačního systému je sice z počátku nákladnější, do budoucna ovšem přináší větší stabilitu a menší náklady na případné úpravy a rozšíření. Tím se liší od ostatních možností a proto je jako jediná schopna uspokojit všechny požadavky investorů ze sekce 4.2 Požadavky investorů.

5.4 Závěr analýzy možných řešení

Jak ukázala kapitola 3. Analýza současného systému, většina bezpečnostních závad původního systému je způsobena špatným návrhem architektury. Přitom právě kvalitní návrh architektury je podle In Leeho kritický pro úspěch e-recruitment systému. [10] Nutnost přeprogramování původního systému od úplných základů v případě rozšiřování původního systému, spolu s ostatními argumenty zmíněnými v této kapitole, vedly k rozhodnutí vytvořit nový informační systém agentury AW Partners s.r.o.

Návrh a organizace systému

Analýza z předešlé kapitoly určila návrh a implementaci nového informačního systému jako nejvhodnější možné řešení pro splnění všech agenturou definovaných požadavků. V této kapitole nejprve volím vhodnou metodiku vývoje. Následuje výběr vhodných technologií a návrh architektury nového systému.

6.1 Metodika vývoje

Před samotným začátkem návrhu a implementace je zapotřebí zvolit správnou metodiku vývoje. Tato zvolená metodika představuje v obecném smyslu soubor metod a postupů pro realizaci určitého úkolu. Konkrétně se pak bavíme o metodice pro budování IS, která “... *definuje principy, procesy, praktiky, role, techniky, nástroje a produkty používané při vývoji, údržbě a provozu informačního systému, a to jak z hlediska softwarově inženýrského, tak z hlediska řízení*” [11, str. 110].

Metodiky dělíme podle modelu vývoje na rigorózní nebo agilní. Rigorózní metodiky vycházejí z přesvědčení, že vývoj softwaru lze plánovat, řídit a měřit. [12, str. 22] Jsou proto vhodné pro projekty s jasně definovanými požadavky. Agilní metodiky naopak vynikají v případech, kdy všechny požadavky předem nejsou známy a tudíž mezi hlavní principy patří častá komunikace se zákazníkem a flexibilita během implementace. [11, str. 116]

Jelikož jsou v případě vývoje informačního systému AW Partners uživatelské požadavky pro velkou část systému relativně jasně definované, nabízí se využití rigorózní metodiky. Například pro službu dotazníkových průzkumů byla však agenturou přislíbena další specifikace požadavků až za průběhu implementace. Analýza původního systému dále představila možnost rozdělení funkcí systému do samostatných komponent. Tyto aspekty vedly k rozhodnutí výběru metodiky Rational Unified Process (RUP), která kombinuje podporu standardního softwarového procesu vývoje a iterativní přístup.

6.1.1 RUP - Rational Unified Process

Tvůrci metodiky Rational Unified Process ve stejnojmenné publikaci zaměřené na osvědčené postupy využití metodiky zmiňují 6 nejdůležitějších pravidel. Následující seznam popisuje, jak se tyto postupy vztahují na vývoj informačního systému AW Partners popsany v následujících kapitolách. [13]

1. Vyvíjet software iterativně

Iterační vývoj je základem metodiky RUP, jelikož umožňuje rozdělit projekt na menší části, nad kterými je možné provádět celý cyklus procesu softwarového vývoje. Proto je dále realizace rozdělena do kapitol zaměřených na konkrétní iterace.

2. Spravovat uživatelské požadavky během projektu

Požadavky a jejich změny je třeba neustále zaznamenávat ve speciálním dokumentu, čehož je v této práci dosaženo vedením dokumentu SRS, který je uložen na příloženém CD ve složce Documentation. Z požadavků je dále vycházeno při tvorbě případů užití, jež jsou uvedeny na začátku kapitol zabývajících se implementací.

3. Používat architekturu založenou na komponentách

Komponentami jsou míněny netriviální moduly nebo subsystémy plnící jasnou funkci. Následující sekce je věnována návrhu logické architektury, která analyzuje komponenty v rámci IS AW Partners.

4. Vizuálně modelovat software

Metodika RUP je silně závislá na modelech v jazyce Unified Modeling Language (UML). Pomocí UML modelů lze přehledně navrhnout strukturu aplikace pro následnou implementaci. Kapitoly zaměřené na realizaci využívají právě modelů UML pro prezentaci implementačních detailů.

5. Ověřovat kvalitu softwaru

Pro dosažení vyšší stability a spolehlivosti systému je během každé iterace prováděno automatické i uživatelské testování.

6. Kontrolovat požadavky na změny softwaru

Jelikož jsem jediným vývojářem, tak není nutné tvořit speciální dokumentaci pro sdílení informací o změnách mezi členy týmu. Pro zachycení změn týkajících se požadavků mi slouží revize dokumentu SRS.

6.1.2 Testování v rámci iterací

Proces testování je při vývoje softwarových produktů často z důvodů nedostatku času zanedbáván. Protože odhalení chyb v rané fázi vývoje může naopak čas krátit, rozhodl jsem se v rámci iterací aplikovat tzv. V-model. V-model popisuje pořadí prováděných aktivit softwarového vývoje v průběhu iterace a ke každé aktivitě přiřazuje příslušný typ testování. [14, str. 330]

Následující kapitoly 7-10, které jsou zaměřené na realizaci jednotlivých iterací, jsou proto strukturovány s využitím tohoto modelu. Postupuji v nich od analýzy a návrhu řešení uživatelských požadavků směrem ke konkrétním ukázkám implementace. Dále jsou demonstrovány unit testy, následně testy systémové a uživatelské a závěrem je představena tabulka validace a verifikace případů užití vzhledem k dříve definovaným požadavkům.

6.2 Výběr technologií

Pro realizaci e-recruitment systémů se běžně využívají webové technologie a mnou navržený systém v tomto ohledu nebude výjimkou, jelikož je funkčnost systému úzce závislá na propojení s webovými stránkami agentury. Projekt jsem se rozhodl realizovat v jazyce PHP, což je velmi populární skriptovací programovací jazyk pro tvorbu dynamických internetových stránek. [15]

Rovněž využiji framework Symfony 3.0 Standard edition, který je dnes jedním z nejpobulárnějších a s největší garancí bezpečnosti. Bezpečnost Symfony je garantována mimo jiné tím, že jeho tvůrci nechali provést bezpečnostní audit nezávislou bezpečnostní agenturou. [16]

Kromě Symfony jsem před zahájením implementace vybral následující knihovny a technologie dostupné pod MIT či jinou open-source licencí. Některé další využití nástroje jsou zmíněny v kapitolách věnovaných realizaci.

- **AdminLTE 2.3**

AdminLTE je populární volně dostupná šablona pro webové aplikace. Kromě realizace responzivního uživatelského rozhraní shromažďuje šablona mnoho dalších velmi užitečných doplňků a knihoven, např. jQuery, Bootstrap 3 nebo jQuery Datatables. Šablona je navržena pro modulární aplikace, a proto je velmi vhodná pro využití při implementaci informačního systému personální agentury. [17]

- **jQuery 2.1.4**

jQuery je JavaScript knihovna, která zjednodušuje manipulaci s prvky na webových stránkách a poskytuje rozhraní pro tvorbu jednoduchých efektů a animací. [18]

- **Bootstrap 3.3.5**

Bootstrap je v současné době nepoužívanější HTML, CSS a JS framework pro tvorbu responzivních webů, tj. webů, jejichž rozhraní se přizpůsobí velikosti obrazovky uživatele. [19]

- **DataTables 1.10.9**

DataTables je plug-in pro knihovnu jQuery, který umožňuje z běžných tabulek v jazyce HTML za pomoci několika konfiguračních příkazů vytvořit interaktivní tabulky podporující stránkování, vyhledávání, řazení a mnoho dalších funkcí. [20]

Při implementaci jsem využil následujících vývojových prostředí a nástroje:

- **PhpStorm IDE** - Vývojové prostředí pro práci v jazyce PHP,
- **Selenium IDE** - Nástroj pro automatizaci uživatelských testů,
- **PHPUnit** - Nástroj pro tvorbu automatizovaných testů zdrojového kódu,
- **UML Star** - Aplikace pro tvorbu UML modelů,
- **Source Tree** - Nástroj pro práci s technologií GIT,
- **Trello** - Online nástroj pro tvorbu seznamů a poznámek.

6.3 Architektura systému

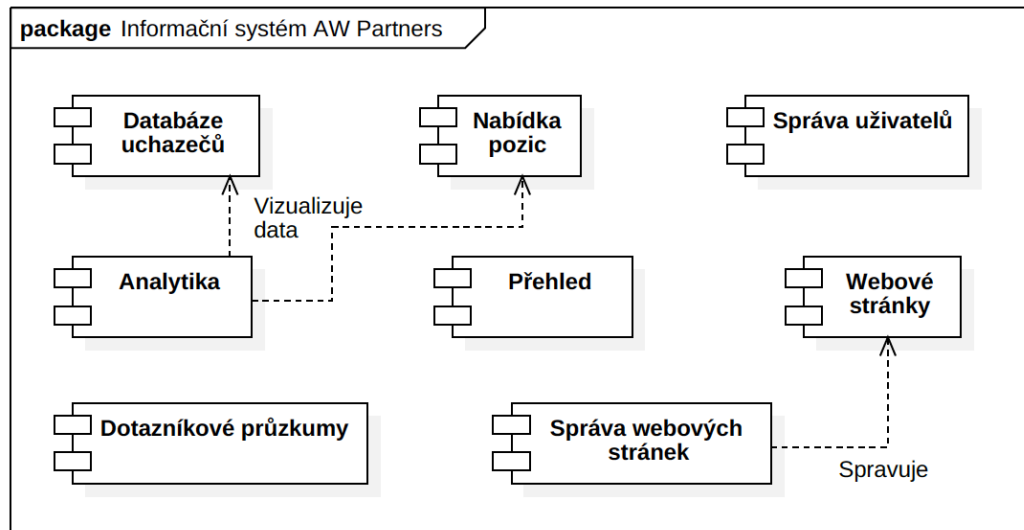
Tato sekce obsahuje logický a technologický pohled na architekturu, což jsou dva z pohledů definovaných Philippem Krutchenem roku 1995 v rámci známého 4+1 pohledu na architekturu. Zbylé pohledy, jako například fyzický nebo sekvenční, úzce souvisí s implementací frameworku Symfony a ne přímo s mojí prací. Rozhodl jsem se proto je zde neuvádět. [21]

6.3.1 Logický pohled

Ve studii Architektura pro holistický e-recruitment systém nové generace z roku 2007 definoval In Lee, Ph.D. seznam komponent, které by měl obsahovat komplexní e-recruitment systém tak, aby podporoval činnosti v rámci celého procesu nábory zaměstnanců. V závěru své práce ovšem In Lee pojednává o nutnosti analyzování potřeby konkrétní společnosti a přizpůsobení systému především jejímu internímu fungování. [10]

Při návrhu architektury nového informačního systému agentury jsem se rozhodl využít analýzy pana In Leea a přizpůsobit strukturu komponent holistického e-recruitment systému požadavkům společnosti AW Partners s.r.o.

Obrázek 6.1 znázorňuje logickou strukturu nového systému. Následující seznam nabízí popis fungování jednotlivých komponent a také analyzuje jejich funkčnost z pohledu holistického e-recruitingového systému podle návrhu In Leeho.



Obrázek 6.1: Komponenty nového systému

6.3.1.1 Databáze uchazečů

Databáze uchazečů je úplným jádrem celého systému, protože shromažďuje pro agenturu nesmírně cenné informace o všech registrovaných zájemcích o zaměstnání. Součástí komponenty musí být podle studie In Leeho výkonný vyhledávací nástroj umožňující nalezení kompetentní osoby pro nabízené pozice. V zahraniční literatuře je pro tuto komponentu často využíván název “System pro sledování uchazečů” (Applicant Tracking System - ATS).

6.3.1.2 Nabídka pozic

Mnou navržená komponenta nabídka pozic plní funkci subsystému pro správu nabízených pozic z pohledu holistického e-recruitment systému. Důležitou funkcí této komponenty je přidávání nových volných pozic a jejich export na webové stránky, kde bude umožněna registrace případným kandidátům. Komponenta bude také poskytovat nástroje pro práci s nově přihlášenými uchazeči na pozici touto cestou.

6.3.1.3 Přehled

Pouze v omezené podobě komponenta přehled plní některé funkce In Leem definovaného subsystému pracovního postupu. In Lee navrhuje automatizovat některé činnosti související s přihlašování kandidátů, mnou navržený systém tyto aktivity ovšem ponechá převážně v kompetenci zaměstnanců agentury. Komponenta přehled pak bude zaměstnancům nápomocná, protože bude informovat o nové aktivitě v rámci systému a agregovat úkoly, jenž je nutné provést.

6.3.1.4 Analytika

In Lee zmiňuje subsystém pro analýzu výkonnosti systému jako nedílnou součást pro zlepšení efektivity celého procesu náboru. Je důležité, aby management firmy měl představu o výkonnosti systému a na základě informací dostupných z komponenty analytiky mohl provádět strategická rozhodnutí.

6.3.1.5 Webové stránky a jejich správa

Korporátní webové stránky musí podporovat agendu e-recruitment systému, a proto jsou součástí systému i nástroje pro úpravu webových stránek. Jelikož se jedná o vedlejší produkt z pohledu zaměření této práce, nebude o těchto nástrojích více pojednáno, ale zdrojové kódy této komponenty a webových nástrojů budou uloženy na přiloženém CD.

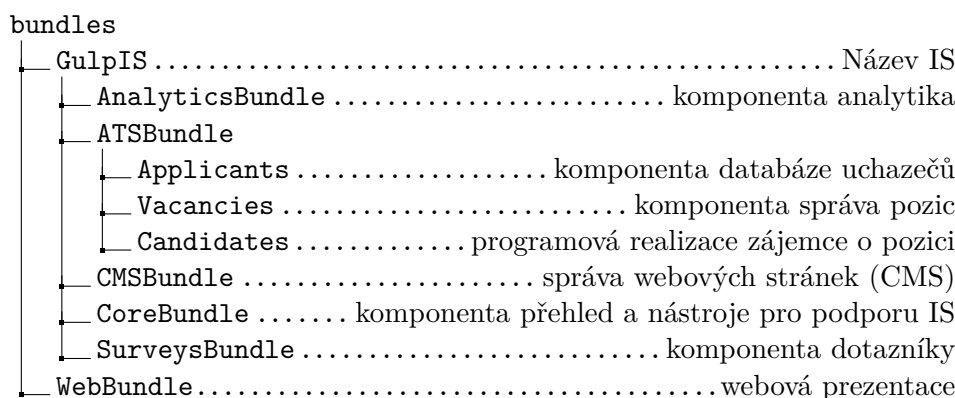
6.3.1.6 Dotazníky

Nástroj pro tvorbu dotazníkových průzkumů sice není součástí klasického e-recruitment systému z pohledu In Leeho, jedná se ovšem o jednu z klíčových služeb agentury. Komponenta bude plnit všechny funkce související s uživatelskými požadavky na funkčnost dotazníkových průzkumů.

6.3.2 Vývojový pohled

Logický pohled na architekturu je vhodný pro vysvětlení funkcionality uživateli systému. Při programování je ovšem nutné vzít v potaz závislosti mezi jednotlivými komponenty, a proto se vývojový pohled mírně liší od pohledu logického.

Struktura 6.2 zobrazuje rozdělení programových částí jednotlivých komponent do balíčků Symphony, tzv. *bundles*. Především je vidět, že jsem se rozhodl vše související s webovou prezentací umístit do balíčku *WebBundle*. Samotný systém jsem pojmenoval GulpIS (tento název nemá žádný hlubší význam) a do složky s tímto názvem jsem umístil systémové balíčky. Jelikož funkce komponenty databáze uchazečů a nabídky pozic jsou velmi propojené, je vhodné umístit je v rámci stejného balíčku *ATSBundle*. Dále jsem přidal mezivrstvu



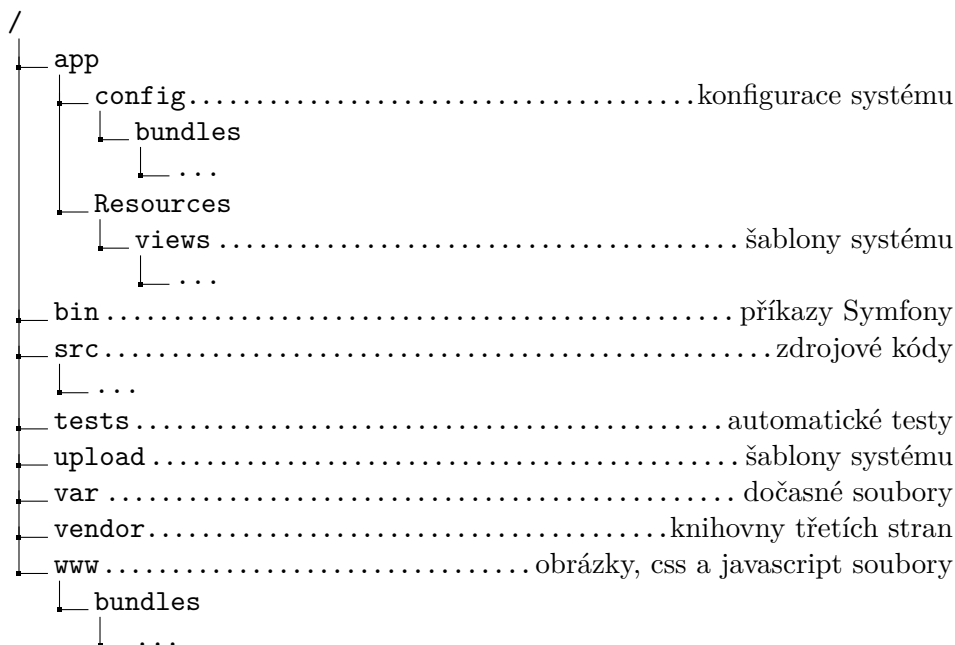
Obrázek 6.2: Rozdělení funkcí komponent do balíčku Symfony

mezi uchazeči a pozicemi v podobně složky pro funkce kandidátů. Slovem kandidát rozlišuji uchazeče, který je evidován jako zájemce o určitou pozici.

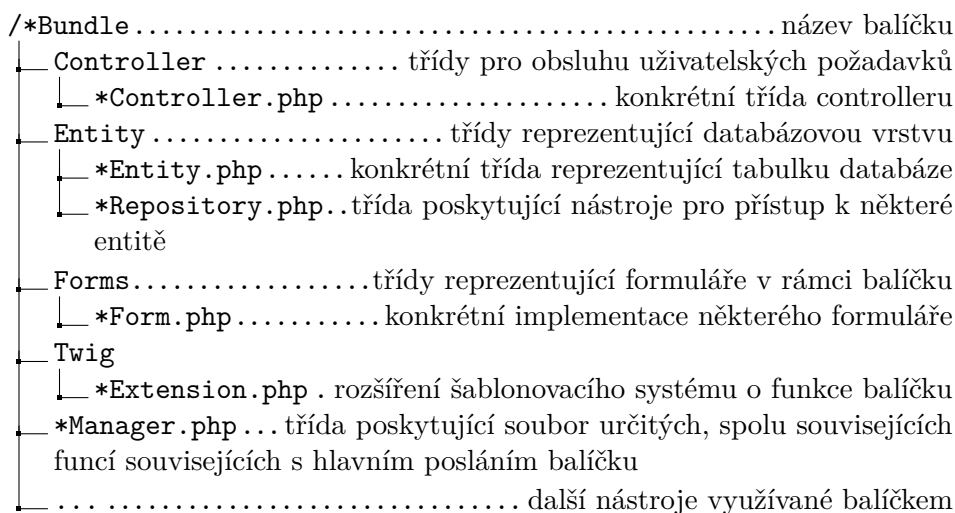
Každý balíček je koncipován jako nezávislý kus kódu a z tohoto důvodu je nutné uchovávat rozděleně jak zdrojové kódy, tak grafické šablony, konfiguraci a CSS/JS soubory. Výše uvedenou strukturu balíčku proto aplikuji na několika místech ve stromové struktuře Symfony projektu. V následujícím grafu 6.3, všude na místě označeném třemi tečkami, je aplikována struktura balíčků zmíněna výše a jednotlivé složky balíčků obsahují soubory související s funkcionalitou patřičné komponenty.

Každý balíček ve složce */src* má v principu podobnou strukturu a dále se liší pouze implementací této struktury. Tato struktura je znázorněna na obrázku 6.4. Základem jsou třídy reprezentující tabulky databáze a tzv. repozitáře poskytující přístup k těmto objektům, jenž jsou vždy ve složce *Entity*. Složka *Controller* obsahuje třídy implementující chování třídy *Controller*, tj. objektu zachycujícího uživatelské požadavky a reagující odesláním příslušné odpovědi. V závislosti na konkrétním balíčku pak může být přítomna složka *Forms*, kde jsou umístěny třídy všech formulářů v rámci balíčku a nebo *Twig*, kde jsou umístěny rozšíření šablonovacího systému Twig. Balíčky mohou dále obsahovat jakékoliv další pomocné třídy, velmi často se jedná o třídu s příponou *Manager*, která má na starost určitou činnost v rámci balíčku.

Navržená struktura projektu 6.3 dodržuje doporučení pro nejlepší praktiky v rámci Symfony projektu podle [22].



Obrázek 6.3: Umístění balíčkové struktury v rámci projektu Symfony



Obrázek 6.4: Struktura obecného balíčku ve složce /src

Realizace iterace 1 - Jádro systému a komponenta přehled

V rámci první iterace jsem se rozhodl vytvořit nástroje, které budou využívány skrze všechny následně implementované komponenty systému. Mezi hlavní zástupce takových podpůrných nástrojů patří submoduly notifikace, úkoly, menu a správa uživatelů.

7.1 Případy užití

Zde zmíněné případy užití jsou realizovány funkcemi obsaženými v rámci první iterace.

- UC1: Přihlásit se
- UC2: Odhlásit se
- UC3: Upravit vlastní přihlašovací údaje
- UC4: Přidat uživatele
- UC5: Upravit uživatele
- UC6: Smazat uživatele

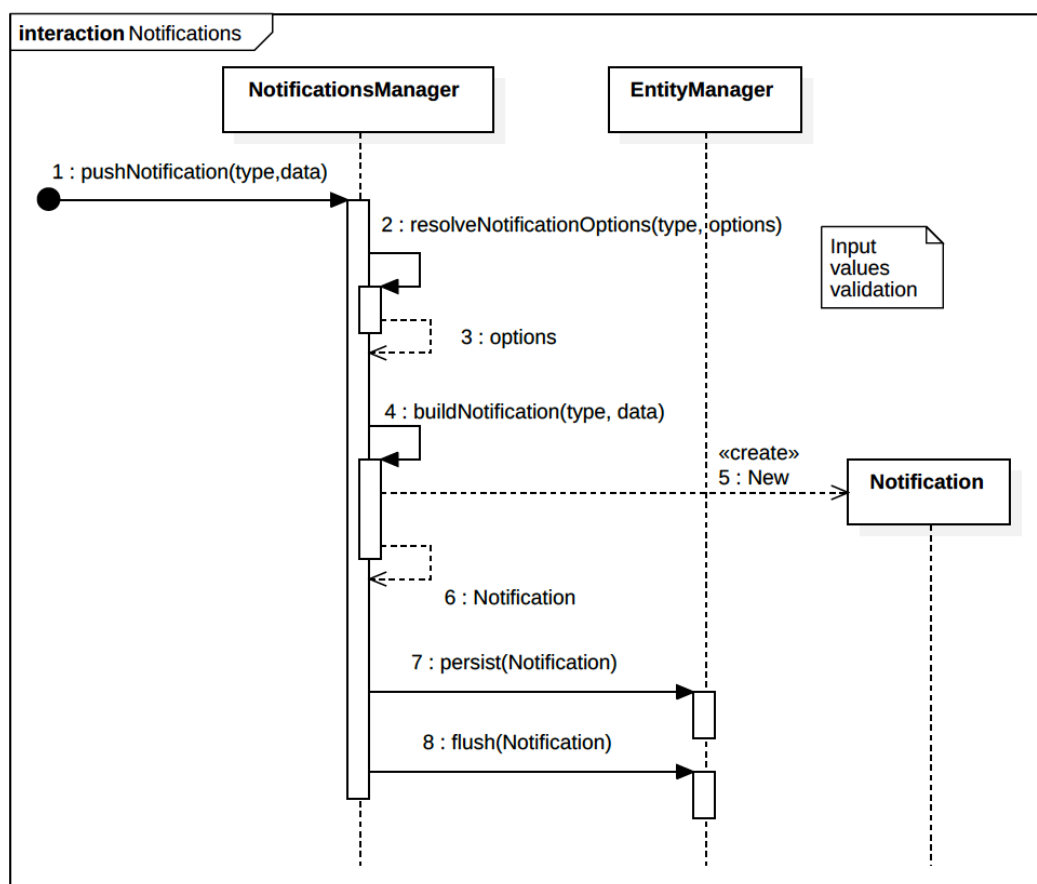
7.2 Implementace

Základem iterace je balíček *CoreBundle*, kde jsou koncentrovány všechny submoduly plnící konkrétní funkce.

7.2.1 Modul upozornění

Protože systém přijímá data nezávisle na činnosti zaměstnanců agentury, musí být uživatelé systému upozorněni, pokud nastala v systému významná změna. Takovou změnou může být například registrace nového uchazeče, přihlášení kandidáta na vypsanou pozici nebo vyplnění dotazníku respondenty. Aby o všech těchto změnách měli zaměstnanci přehled, umístil jsem do horního menu systému ikonku, po jejímž rozkliknutí je možné zobrazit všechna nová upozornění.

Protože komponenta upozornění musí být už z principu rozšiřitelná, přidal jsem do hlavního konfigurační složky systému soubor *notifications.yml*, kde lze připsáním několika řádků deklarovat novou položku upozornění. Nové upozornění je možné definovat pomocí názvu, grafického stylu (barva pozadí a ikonka), povolených uživatelských rolí, nadpisu, textového obsahu a odkazu.



Obrázek 7.1: Přidání nového upozornění do systému

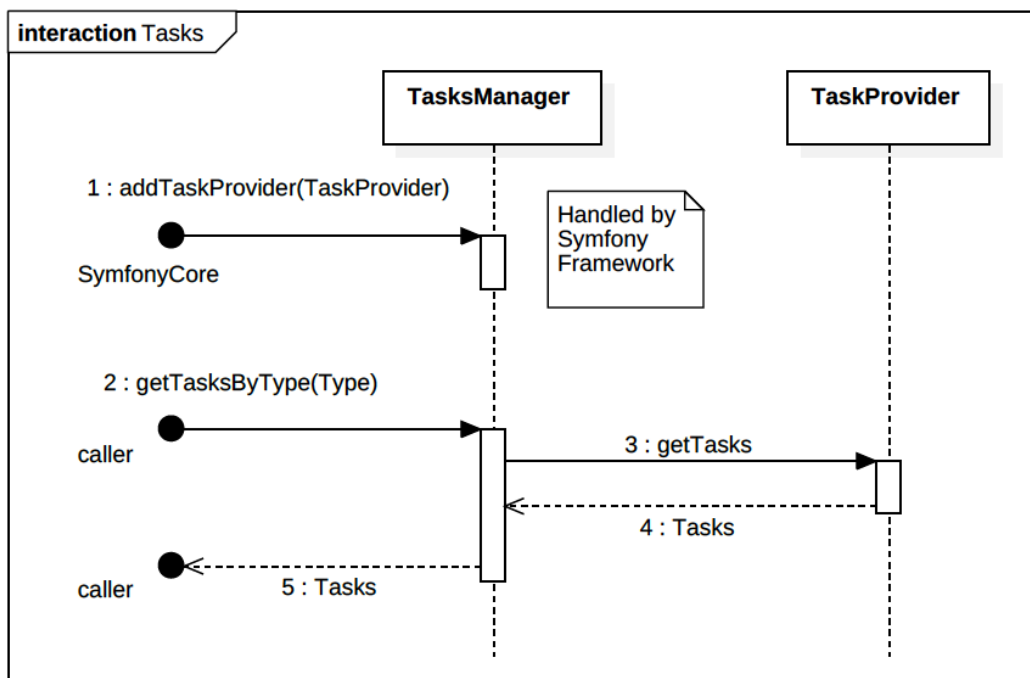
O přidání nového a získání aktivních upozornění se stará třída *Notification-Manager*, jenž je registrována jako služba a tudíž k ní mohou všechny ostatní

komponenty přistupovat pomocí sdíleného kontejneru. Na obrázku 7.1 je znázorněno, jak třída vnitřně zpracuje požadavek na přidání nového upozornění. Je k tomu využita metoda `pushNotification(type, data)`, která obdrží typ notifikace definovaný v konfiguračním souboru a data, která definují strukturu notifikace. Metoda nejprve provede validaci obdržených dat na jejichž základě vytvoří novou notifikaci a tu následně uloží do databáze.

7.2.2 Modul úkoly

Funkce modulu úkoly, ačkoliv může připomínat funkci submodule upozornění, je v jednom aspektu zásadně rozdílná. Pokud například vyprší souhlas se zpracováním osobních údajů uchazeče, je úkol odstranit tohoto uchazeče z databáze aktuální nezávisle na čase, kdy tato událost nastala. Vytvořil jsem proto v systému službu úkoly, jenž umožňuje kterékoliv komponentě vytvořit třídu implementující definované rozhraní `TaskProvider`. Pokud je tato třída registrována jako služba se značkou `gulpis.task_provider`, bude zachycena komponentou úkoly a zpracována.

Zjednodušeně řečeno, stačí vytvořit třídu, která implementuje metody pro vrácení aktivních úkolů, celkového počtu úkolů a seznam uživatelských rolí, kterým přísluší zobrazovat tyto úkoly. Na obrázku 7.2 je znázorněno, jak třída `TasksManager` pomocí takto popsaného rozhraní zpracuje požadavek na vrácení aktivních úkolů.



Obrázek 7.2: Získání aktivních úkolů daného typu

7.2.3 Menu

Pro realizaci navigace v systému jsem využil volně dostupný *KnplMenuBundle 2.0* balíček. S pomocí této knihovny jsem vytvořil třídu *MenuBuilder* a naprogramoval načítání obsahu menu z konfiguračního souboru *menu.yml*, který jsem umístil do hlavní složky konfigurace systému. Takto jsem dosáhl velmi jednoduché správy položek menu bez nutnosti zásahů do zdrojových souborů šablony. [23]

7.2.4 Přihlašování uživatelů

Stejně jako v případě navigace, i pro správu uživatelů má Symfony Framework předpřipravený balíček zvaný *FOSUserBundle*. Tato knihovna se stará o přihlašování, správu uživatelů a jejich ukládání do databáze. Standardním způsobem popsaným na oficiálních webových stránkách Symfony Frameworku jsem balíček implementoval a definoval v rámci systému následující práva, která je možné uživatelům nastavit: [24]

- povolení zobrazení databáze uchazečů,
- povolení zobrazení databáze pozic,
- povolení zobrazení kontaktních formulářů z webu,
- povolení úpravy webových stránek,
- povolení úpravy databáze uchazečů,
- povolení úpravy databáze pozic,
- přístup do komponenty dotazníků,
- přístup do komponenty analytika,
- přístup k nástrojům pro správu systému.

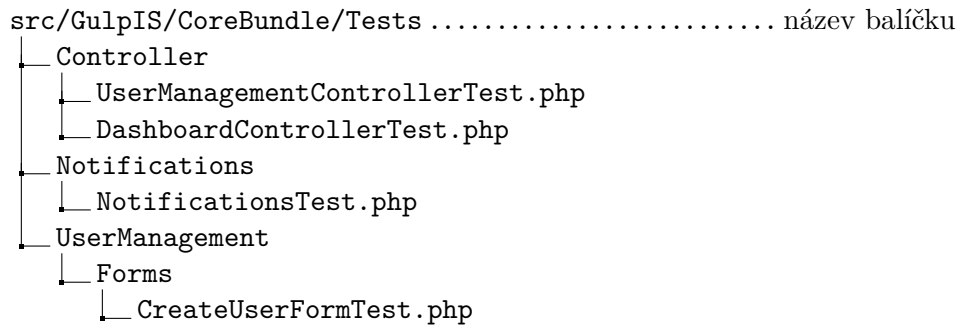
Taktéž je možné přiřadit uživateli roli administrátora, čímž získá uživatel veškerá výše zmíněná práva.

7.3 Testování

Testování v rámci první iterace se skládalo z kombinace PHPUnit testů, které mají za cíl kontrolovat správné fungování jednotlivých tříd v PHP kódu aplikace a Selenium testů, které simulují akce uživatele. Pro správu uživatelů byl využit balíček *FOSUserBundle*, který již obsahuje testy kódu a tak bylo testování funkcionality spojené se správou uživatelů testováno převážně pomocí Selenium testů.

7.3.1 PHPUnit testy

Všechny automatizované PHPUnit testy pro první iteraci jsou umístěné ve složce `src/GulpIS/CoreBundle/Tests`, jejíž obsah je vidět na obrázku struktury 7.3.



Obrázek 7.3: PHPUnit testy jádra systému

Složka *Controller* obsahuje tzv. *smoke testy*, které testují, zda-li je možné úspěšně zavolat všechny stránky související s touto komponentou. Dále jsou testovací třídy řazeny do složek podle struktury balíčku *CoreBundle*.

7.3.2 Selenium testy

Selenium testy simulují akce uživatele s využitím prohlížeče a následující scénáře jsem vytvořil pro automatizování základních operací souvisejících s uživatelskými účty, což je po dokončení první iterace jediná tímto způsobem testovatelná funkcionality.

- **Vytvoření, přihlášení, odhlášení a odebrání nového uživatele**

Tento automatizovaný test nejprve přidá nového uživatele do systému a nastaví mu přihlašovací údaje a přístupová práva. Následně se test pokusí s údaji tohoto uživatele přihlásit do systému. Toto přihlášení ovšem selže, jelikož uživatelský účet nebyl aktivován. Test proto aktivuje účet a opakuje proces, tentokrát ovšem se špatnými přihlašovacími údaji, aby ověřil nemožnost takového přístupu. Konečně test ověří úspěšné přihlášení uživatele a odebere uživatele ze systému. Poslední akce tohoto testu je ujištění, že pod účtem odebraného uživatele není možné přihlásit se do systému.

- **Změna profilu uživatele**

Testem je ověřeno, že jsou funkční nástroje pro změnu uživatelského jména a mailu aktuálně přihlášeného uživatele. Test ověří, že je možné uživatele s takto změněnými údaji do systému přihlásit.

- **Úprava hesla v nastavení profilu uživatele**

Test ověřuje fungování nástrojů pro změnu hesla aktuálně přihlášeného uživatele. Se změněným heslem se test pokusí do systému přihlásit.

- **Kontrola práv uživatele (povoleno pouze CMS)**

Test ověřuje, že uživatel s právy pro správu webových stránek má přístup k nástrojům pro úpravu webových stránek a zároveň nemá možnost přistupovat do jiných částí systému.

- **Kontrola práv uživatele (povolena pouze databáze uchazečů)**

Test ověřuje, zda-li uživatel s nastavenými právy pro přístup k databázi uchazečů může k nástrojům databáze uchazečů přistupovat a zároveň nemá možnost přistupovat do jiných částí systému.

- **Kontrola práv uživatele (vše zamítnuto)**

Testem se ověřuje, zda-li uživateli, kterému byly odebrány veškerá přístupová práva, bude zamítnut přístup do všech sekcí informačního systému.

- **Kontrola práv uživatele (povoleny databáze uchazečů a pozic)**

Test ověřuje, zda-li uživatel s nastavenými právy pro přístup k databázi uchazečů a pozic může k příslušným nástrojům přistupovat a zároveň nemá možnost přistupovat do jiných částí systému.

7.4 Ověření splnění požadavků

Tabulka 7.1 znázorňuje míru splnění funkčních požadavků z dokumentu SRS případy užití realizovanými v rámci první iterace.

Tabulka 7.1: Mapování případů užití z 1. iterace na požadavky

Uživatelské požadavky	Případy užití
3.3.5.1. Přihlášení do systému	UC1, UC2
3.3.5.2. Správa uživatelů	UC3, UC4, UC4, UC5
3.3.5.3. Nastavení přístupových práv uživatelům	UC4, UC5

V rámci první iterace byl splněn nefunkční požadavek 3.4.1.1 na šifrovanou komunikaci pomocí SSL certifikátu. Symfony projekt byl nakonfigurován tak, aby všechny přístupové požadavky na dokumenty s předponou */admin* povinně využívaly zabezpečený HTTPS protokol.

Realizace iterace 2 - Komponenty databáze uchazečů a nabídka pozic

Komponenty databáze uchazečů a nabídka pozic jsou mezi sebou velmi provázané a bylo proto přirozené vyvíjet jejich funkcionalitu v rámci jedné iterace.

8.1 Případy užití

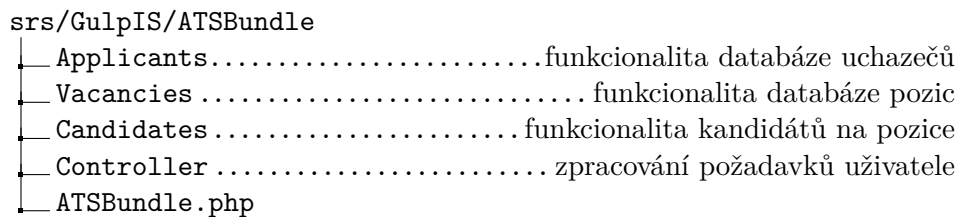
Zde jsou vypsány případy užití, jenž jsou pokryty implementací komponent z této kapitoly.

- UC1: Vyhledat uchazeče
- UC2: Spravovat uchazeče (přidat, zobrazit, upravit, smazat)
- UC3: Navrhnout uchazeče na pozici
- UC4: Zveřejnit profil uchazeče na webových stránkách
- UC5: Přidat doporučení k uchazeči
- UC6: Rozeslat e-mail vybraným uchazečům
- UC7: Spravovat souhlas se zpracováním osobních údajů uchazeče
- UC8: Vyhledat pozici
- UC9: Spravovat pozice (přidat, zobrazit, upravit, smazat)
- UC10: Zveřejnit pozici na webových stránkách
- UC11: Přihlášení na pozici z webových stránek

- UC12: Správa kandidátů přihlášených z webových stránek
- UC13: Správa ručně navržených kandidátů na pozici
- UC14: Ukončit nábor na pozici
- UC15: Spravovat číselníky databáze uchazečů (pozice, obory, města, země, pc gramotnost)
- UC16: Spravovat číselníky databáze pozic (Firmy)

8.2 Implementace

Všechna funkcionalita z této kapitoly je zastřešena nově vytvořeným balíčkem *ATSBundle*, nebo-li balíčkem pro sledování uchazečů (viz obr. 8.1). Ten se dále dělí na 3 složky určené pro funkcionality související s databází uchazečů (Applicants), databází pozic (Vacancies) a kandidáty (Candidates), což je označení pro uchazeče hlásícího se na některou z nabízených pozic.



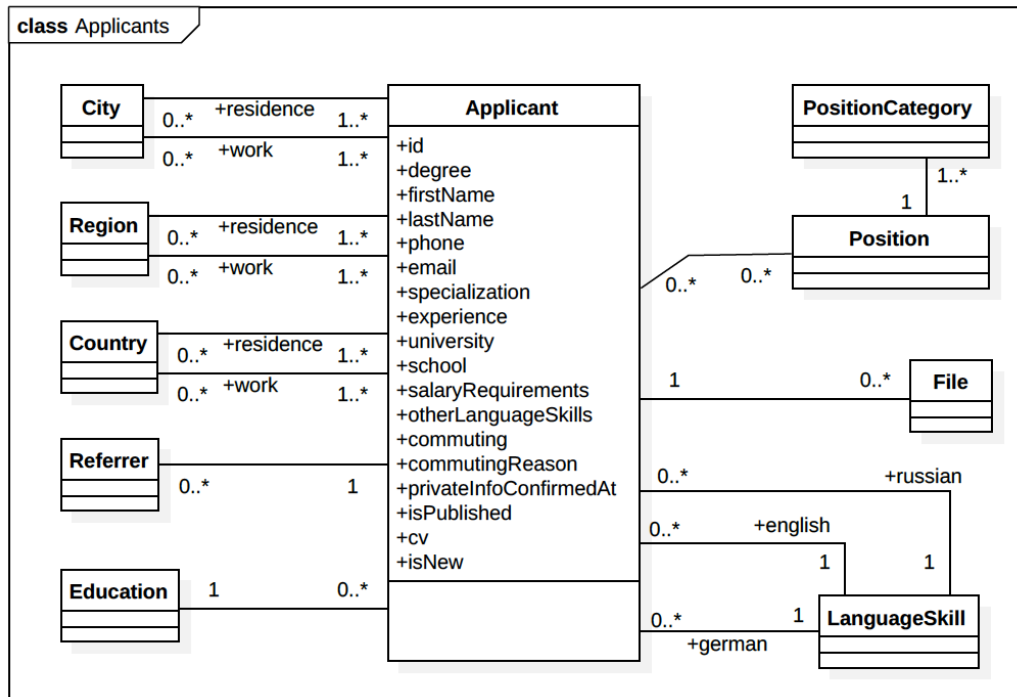
Obrázek 8.1: Struktura balíčku ATSBundle

8.2.1 Databáze uchazečů

Jak již název napovídá, z velké části má komponenta na starost uchování dat, a proto jsou jejich správné uložení a vztahy mezi objekty úplným základem, na kterém je možné dále budovat. Zjednodušený model datových tříd je zobrazen na obr 8.2. Jelikož vypsání všech soukromých atributů a přístupových metod by vyplnilo několik stran a nepřineslo žádnou přidanou hodnotu, rozhodl jsem se uvést pouze ty nejzákladnější atributy datového modelu. Pokud nemá třída v diagramu žádné parametry, má ve skutečnosti pouze identifikátor a jméno.

8.2.1.1 Vyhledávání uchazečů

V databázi uchazečů se vyskytují tisíce záznamů a bez nástrojů pro podporu efektivního vyhledávání se při práci se systémem zaměstnanci agentury neobejdou.



Obrázek 8.2: Diagram tříd tvořících datový model uchazeče

Pro realizaci uživatelsky přívětivého vyhledávače, který má možnost spojovat dotazy za pomoci logických spojek konjunkce a disjunkce, jsem využil volně dostupný jQuery plugin *jQuery QueryBuilder v2.3.2*. Tato knihovna ovšem zajišťuje pouze tvorbu dotazů na straně klienta, a proto bylo nutné vytvořit vlastní mezivrstvu pro zpracování dat zaslaných knihovnou a navrácení příslušných výsledků z databáze. Níže jsou popsány nejdůležitější třídy tvořící tuto mezivrstvu a jejich společné fungování při vykreslení vyhledávacího pole do šablony představuje diagram na obrázku 8.4. [25]

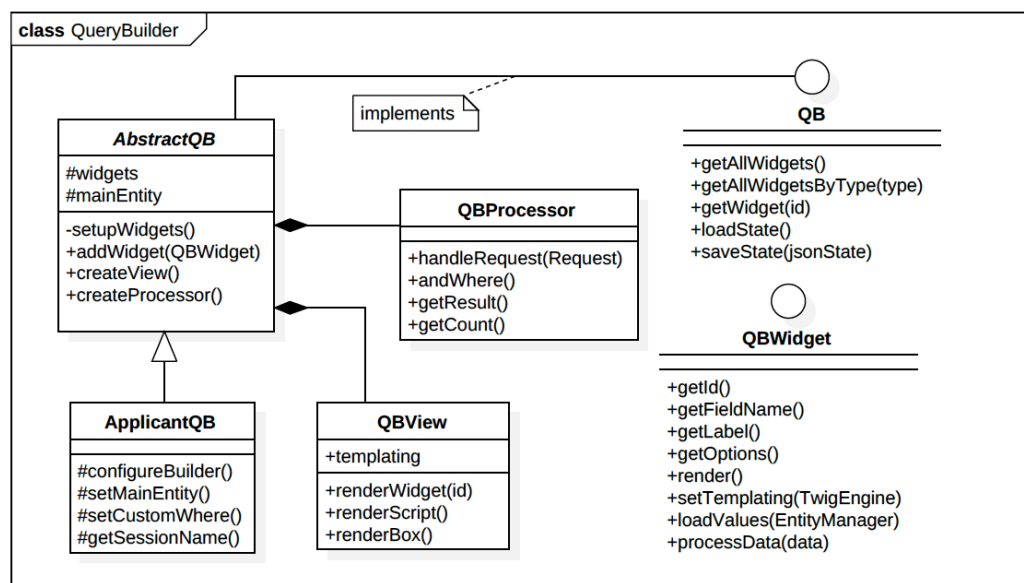
- **AbstractQB**

Tato abstraktní třída se stará především o načtení všech vyhledávacích polí, jež jsou reprezentovány třídami implementujícími *AbstractQBWidget* rozhraní. Pro přístup k těmto vyhledávacím polím je vyhrazeno několik metod, které vracejí všechna nebo pouze některá, parametrem určená pole. Třída dále definuje základní chování a předepisuje metody, které musí implementovat konkrétní vyhledávač, např. *ApplicantQB*.

- **ApplicantQB**

Třída reprezentuje konkrétní vyhledávač a za pomoci metod zděděných od abstraktní třídy *AbstractQB* jednoduchým způsobem definuje vyhle-

8. REALIZACE ITERACE 2 - KOMPONENTY DATABÁZE UCHAZEČŮ A NABÍDKA POZIC



Obrázek 8.3: Diagram tříd tvořících funkce vyhledávání

dávací pole tohoto vyhledávače. V metodě *configureBuilder()* přidá třída konkrétní implementace polí implementující rozhraní *QBWidget*.

- **QBView**

Objekt této třídy je vložen přes *controller* do šablony a pomocí speciálních metod objektu může tato šablona vykreslovat html a javascript kód daného vyhledávače.

- **QBProcessor**

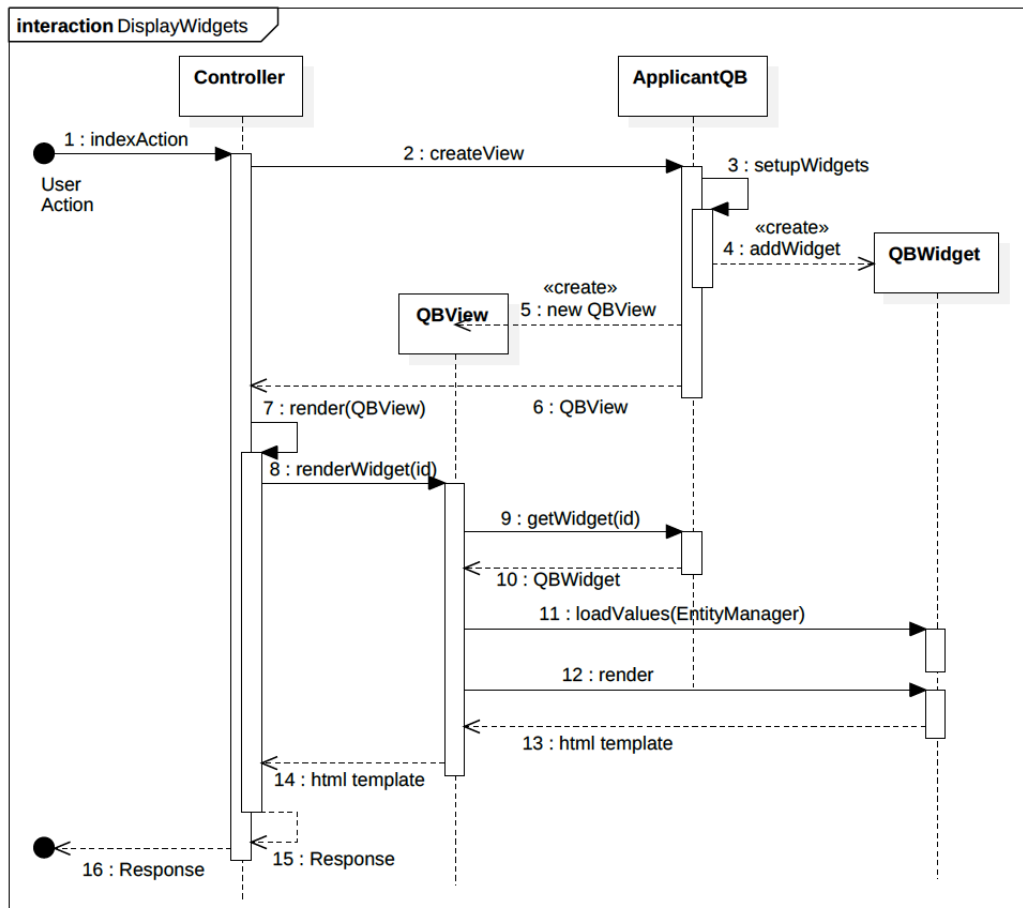
Třída *QBProcessor* je využívána pro zpracování a validaci dat obdržných od uživatele po provedení vyhledávání. Objekt třídy přijímá celý HTTP požadavek, ve kterém rozezná data zasláná vyhledávačem a ty zpracuje. Jakmile dojde ke zpracování, je možné zasílat objektu požadavky na získání dat z databáze, které odpovídají dříve provedené konfiguraci vyhledávače.

- **AbstractQBWidget**

Různé druhy vyhledávacích polí se mohou lišit. Například do pole pro vyhledání podle jména uchazeče stačí vepsat hodnotu, ale pro vyhledávání podle pozic, oborů nebo dosaženého vzdělání je nutné vybrat z existujících hodnot v databázi. Pro všechny tyto konkrétní implementace vyhledávacích polí poskytuje abstraktní třída *AbstractQBWidget* základní programové vybavení a také definuje povinná pole pro dodržení implementačních požadavků rozhraní *QBWidget*.

- *QBWidget

Konkrétními implementacemi, jež dědí třídu *AbstractQBWidget*, jsou například *TextQBWidget* nebo *SelectQBWidget* a další. Třída *TextQBWidget* uplatňuje úplné minimum konfigurace a pouze nastavuje příslušnou šablonu pro vykreslení. V případě *SelectQBWidget*, jež je určen pro výběr existujících hodnot, je nejprve potřeba tyto hodnoty načíst z databáze, a proto je součástí implementace také definice pravidel obstarávajících toho načítání.



Obrázek 8.4: Diagram vykreslení vyhledávacího pole v šabloně

8.2.1.2 Bezpečné ukládání souborů

Pro ukládání souborů souvisejících s položkami uchazečů v databázi jsem využil Symfony balíček *VichUploaderBundle*. S využitím tohoto balíčku stačí upravit třídu reprezentující uchazeče a do konfiguračního souboru uložit cestu pro ukládání nově nahraných souborů. Tato cesta nesmí být veřejně přístupná, aby nemohl kdokoliv získat k souborům přístup, není-li přihlášen do systému. Složku pro nahrané soubory jsem proto umístil za veřejně přístupný adresář */www* a pro stahování souboru jsem vytvořil zabezpečenou metodu *download-FileAction()* ve třídě *ViewController*, která využívá metodu *downloadObject()* poskytovanou výše zmíněným balíčkem. [26]

8.2.2 Nabídka pozic

Funkcionalita komponenty databáze nabídky pozic je rozdělena do složek *Vacancies* (pozice) a *Candidates* (kandidáti) v rámci balíčku *ATSBundle*. Složka *Vacancies* obsahuje velmi obdobnou strukturu a obecné chování, jaké bylo popsáno v kapitole návrhu v sekci 6.3.2.

Pokud se uchazeč sám přihlásí na pozici skrze webové stránky nebo je-li přiřazen ručně zaměstnancem agentury, pak se může nacházet v některém z mnoha předdefinovaných stavů. Kandidáti mění své stavy až do doby, než-li je pozice označena jako uzavřená a jsou vybráni úspěšní kandidáti. Stavový diagram možných průchodů mezi stavy je zobrazen na obrázku 8.5

Nejzajímavější je právě obsah složky *Candidates*, kde se ukrývají následující třídy podporující správu přiřazených kandidátů na pozice.

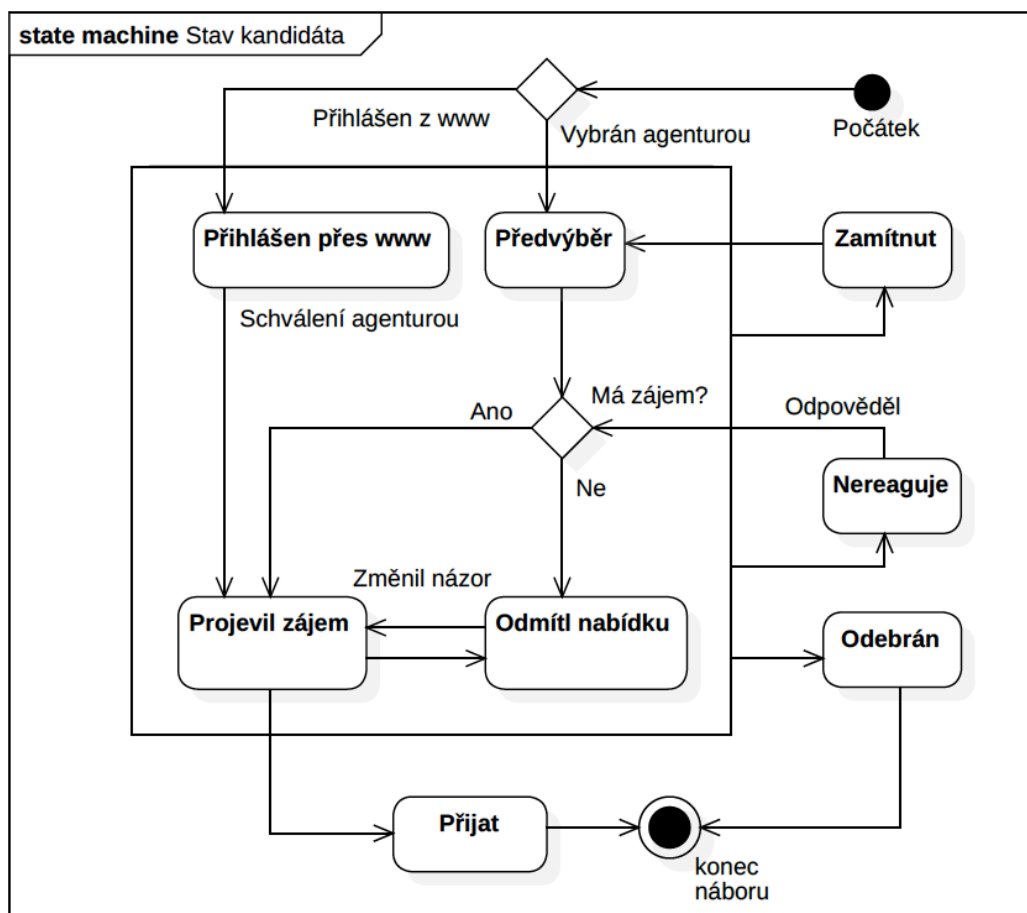
- **CandidateManager**

Na tuto třídu, jenž je registrována jako služba, se *controller* obrací při zpracování požadavků uživatele. Mezi nejdůležitější metody této třídy patří přiřazení kandidáta na pozici nebo ukončení náboru, přičemž *CandidateManager* zajistí provedení programové logiky, která s danou akcí souvisí. Ukázkou takového volání je např. změna stavu kandidáta, viz obrázek 8.6.

- **AbstractState**

Abstraktní třída *AbstractState* implementuje společné chování všech stavů a definuje povinnosti pro potomky dědicí z této třídy. Těmito povinnostmi je implementovat metody *configureOptions()* a volitelně pak *generateLogMsg()*. První ze zmíněných metod nastaví jméno, styl a chování při změně stavu. Pro všechny potomky třída obstarává načítání tohoto nastavení a poskytuje k němu přístupové metody. Dále *AbstractState* zajišťuje, aby byly implementovány všechny veřejné metody rozhraní *State*.

- **StateOptions**



Obrázek 8.5: Diagram stavů evidovaného kandidáta na pozici

Jedinou funkcí této třídy je zajistit správné načtení konfigurace stavu a případně jí vyplnit standardními hodnotami. K tomuto účelu využívá třída Symfony komponentu OptionsResolver.

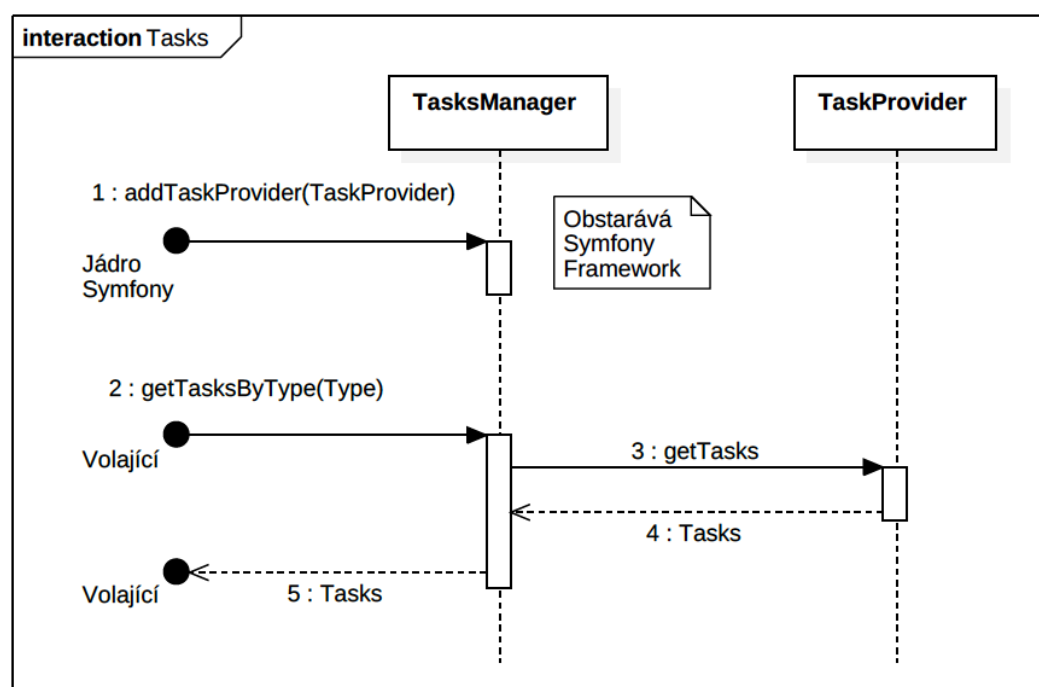
- **States**

Tato třída uchovává informace o všech dostupných stavech a skládá se pouze z konstant, jejichž hodnota je identifikátorem daných stavů.

- **StateAction**

V grafickém rozhraní má uživatel možnost provádět nad záznamy kandidátů akce, jež následně vedou ke změně stavu. Tyto akce a jejich příslušné stavy (např. zamítnout kandidáta) jsou definovány v konfiguračním souboru *Actions.yml*. Třída StateAction je reprezentací jedné takovéto akce a uchovává informace o svém textu, css třídě, ikoně a odkazu.

8. REALIZACE ITERACE 2 - KOMPONENTY DATABÁZE UCHAZEČŮ A NABÍDKA POZIC



Obrázek 8.6: Změna stavu kandidáta na pozici

8.3 Testování

Fáze testování druhé iterace je opět rozdělena na PHPUnit testování, jenž ověřují převážně přístupnost všech stránek a také základní funkce databáze uchazečů a Selenium testy simulující interakci uživatele.

8.3.1 PHPUnit testy

Na obrázku 8.7 je znázorněna struktura automatických PHPUnit testů v rámci druhé iterace. Ve složce *Controller* jsou rozděleny testy ověřující základní funkčnost stránek na testy databáze uchazečů (*Applicants*) a databáze pozic *Vacancies*. Soubor *ApplicantCrudTest.php* obsahuje test pro ověření funkčnosti správy uchazečů a *ApplicantRegisterTest.php* ověřuje celý proces registrace uchazeče na pozici.

8.3.2 Selenium testy

- **Přidání, správa a odebrání uchazeče**

Testem je ověřováno úspěšné přidání nového uchazeče po zadání korektních údajů. Následně je využit vyhledávač k zobrazení karty přidaného uchazeče a je ověřena funkcionality spojená se správou uchazeče (přidání

```

src/GulpIS/ATSBundle/Tests.....název balíčku
├── Controller
│   ├── Applicants
│   │   ├── EditControllerTest.php
│   │   ├── IndexControllerTest.php
│   │   ├── ListsControllerTest.php
│   │   ├── ReferrersControllerTest.php
│   │   └── ViewControllerTest.php
│   └── Vacancies
│       ├── IndexControllerTest.php
│       ├── ListsControllerTest.php
│       └── ViewControllerTest.php

```

Obrázek 8.7: Struktura PHPUnit testů komponent databáze uchazečů a pozic

reference, úprava záznamu CV, zobrazení záznamu na webových stránkách). Na závěr je uchazeč odebrán ze systému a je ověřeno, že není možné nalézt jeho záznam v databázi.

- **Validace formuláře pro přidání nového uchazeče**

Test postupně zkouší přidat nového uživatele zadáním nevalidních údajů do formuláře pro přidání uchazeče. Test je úspěšný, pokud se přes všechny pokusy nepodaří záznam uchazeče uložit.

- **Správa číselníků databáze uchazečů**

Test ověřuje vyhledání, přidání, úpravu a odebrání záznamu z tabulky pozic, oborů, měst a zemí.

- **Přidání a odebrání pozice**

Jednoduchý test pro ověření funkčnosti nástroje pro přidání, úpravu a odebrání pozice.

- **Správa pozice**

Tento test je obdobný předešlému testu, zajišťuje navíc ověření zveřejnění pozice na webových stránkách agentury.

- **Správa číselníku firem**

Testem je ověřováno fungování vyhledávání, přidání, úprav a odebrání záznamu z tabulky firem.

8.4 Ověření splnění požadavků

Tabulka 8.1 obsahuje funkční požadavky z dokumentu SRS a příslušné případy užití z této kapitoly, které tyto požadavky uspokojují.

8. REALIZACE ITERACE 2 - KOMPONENTY DATABÁZE UCHAZEČŮ A NABÍDKA POZIC

Tabulka 8.1: Mapování případů užití ze 2. iterace na požadavky

Uživatelské požadavky	Případy užití
3.3.1.1. Vyhledávání uchazečů	UC1
3.3.1.1.1. Vyhledávání na základě dříve nabízených pozic	UC1
3.3.1.1.2. Vyhledávání v přiložených dokumentech	UC1
3.3.1.1.3. Vyhledávání mezi referenčními osobami	UC1
3.3.1.2. Správa uchazečů	UC2
3.3.1.2.2. Správa referenčních osob	UC5
3.3.1.2.2. Zveřejnění profilu uchazeče na WWW stránkách	UC4
3.3.1.3. Získání e-mailových adres vybraných uchazečů	UC6
3.3.2.1. Správa pozic	UC9
3.3.2.1.1. Zobrazení uchazečů spojených s pozicí	UC9
3.3.2.1.2. Přijetí uchazeče na pozici	UC9
3.3.2.1.3. Odmítnutí uchazeče	UC9
3.3.2.1.4. Odstranění ze seznamu	UC9
3.3.2.1.5. Zobrazení uchazečů přihlášených na pozici přes webové stránky	UC9
3.3.2.1.6. Akceptace uchazeče přihlášeného na pozici přes webové stránky	UC9
3.3.2.1.7. Odstranění uchazeče přihlášeného na pozici přes webové stránky	UC9
3.3.2.2. Vyhledávání v databázi pozic	UC8

Sekce 8.2.1.2 pojednávala o realizaci bezpečného přístupu k interním dokumentům, čímž byl naplněn nefunkční požadavek na omezení veřejného přístupu ke všem dokumentům zmiňovaných v sekci 3.4.1.2 z dokumentu SRS.

Dále bylo všemi nástroji dodrženo pravidlo pro odebrání záznamu ze systému pouze přes HTTP požadavek s pomocí metody POST, čímž je splněn nefunkční požadavek 3.4.1.3 dokumentu SRS.

Jelikož byla při návrhu objektového modelu popsáném v této kapitole brána v potaz rozšiřitelnost, byl rovněž naplněn nefunkční požadavek 3.4.4.

Realizace iterace 3 - Komponenta analytika

Informační systém uchovává velké množství dat, s kterými je možné pracovat skrze nástroje komponent. Cílem modulu Analytika je shromážděné informace prezentovat uživateli způsobem, který umožní získat komplexní přehled o datovém obsahu systému.

9.1 Případy užití

- UC1: Zobrazit graf vybrané metriky

Systém zobrazí uživateli graf z nabídky, jejíž položky odpovídají metrikám definovaným uživatelskými požadavky (např. počet nově přidávaných uchazečů do databáze v závislosti na čase).

9.2 Implementace

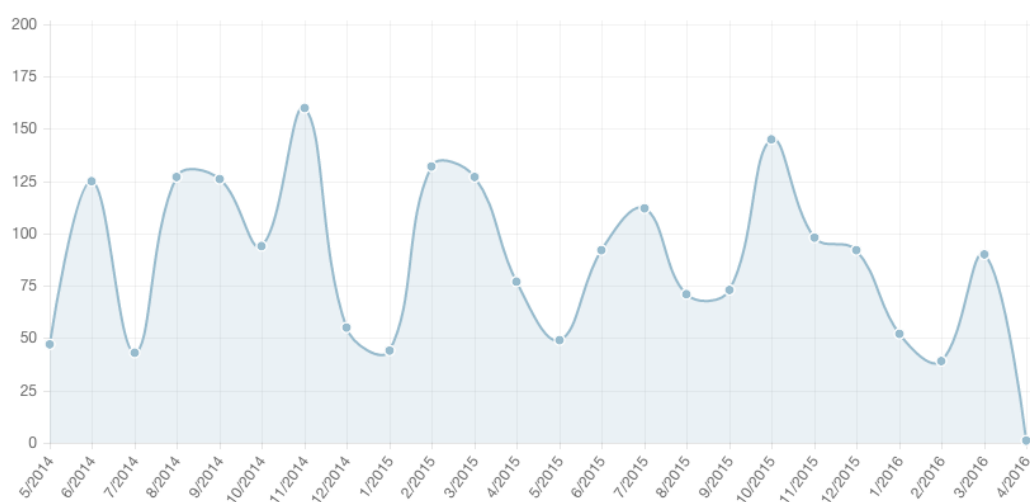
Implementace této komponenty se dá shrnout do dvou hlavních částí. První je nalezení vhodného způsobu vizuální prezentace dat uživateli, neboli vykreslení grafů. Druhým bodem je vyplnění těchto grafů správnými daty. Pro zdrojové kódy tvořící komponentu jsem vytvořil balíček *AnalyticsBundle* ve stejnojmenné složce.

9.2.1 Vykreslení grafů

Pro vizuální prezentaci dat uživateli jsem se rozhodl využít JS knihovnu ChartJS. Na výběr bylo mnoho alternativních knihoven podporující stejné funkce. Pro ChartJS jsem se rozhodl především pro jeho jednoduchou implementaci, velmi obsáhlou dokumentaci a přizpůsobivost grafů velikosti obrazovky. Obrázek 9.1 demonstruje využití grafu generovaného knihovnou ChartJS pro vi-

zualizaci počtu nových uchazečů v závislosti na čase. Na horizontální ose je promítnut čas a vertikálně se zobrazuje počet uchazečů, kteří byli v konkrétní dobu přidání do databáze. [27]

Počet nových uchazečů v databázi



Obrázek 9.1: Graf vykreslený javascriptovou knihovnou ChartJS

9.2.2 Zdroj dat

Poskytovatelem dat pro vybrané typy grafů je mnou navržená třída *ChartManager*. Objekt této třídy, jenž je možné získat ze sdíleného kontejneru, poskytuje rozhraní pro získání dat k naplnění grafů. Jednotlivé metody třídy volají repozitáře z ostatních komponent a získaná data formátují do podoby použitelné v šabloně grafu.

9.2.3 Rozšíření Doctrine

Jednou z hlavních překážek při získávání dat je nemožnost použití agregačních funkcí v databázových dotazech prováděných pomocí knihovny Doctrine. Doctrine neposkytuje funkce pro agregaci dat za pomoci časových intervalů, např. měsíce či roku. Tuto překážku jsem překonal využitím rozšíření pro knihovnu Doctrine navrženého programátory Rafaellem Kassnerem a Sarjono Mukti Aji, zveřejněného pod MIT licencí. [28]

9.3 Testování

Přestože komponenta nenabízí žádné konkrétní nástroje pro použití uživatelem, je vhodné ověřit korektnost zobrazených dat oproti dostupným zázna-

mům v databázi. Testování tedy spočívá v ručním porovnání dat vykreslených v grafech s výsledky vyhledávání v databázi uchazečů za použití měřené metricky při vyhledávání. Pro vyloučení chyby jsem při testování využil jak novou komponentu databáze uchazečů, tak vyhledávání pomocí nástrojů původního systému.

9.4 Ověření splnění požadavků

Komponenta analytika je z pohledu uživatele velmi jednoduchá a jediným funkčním požadavkem je zobrazení grafu pro vybranou metriku. Abych udržel v rámci kapitol jednotný styl zaznamenávání pokrytí požadavků, znázorňuje tabulka 9.1 pokrytí požadavku jediným případem užitím realizovaným touto iterací.

Tabulka 9.1: Mapování případů užití ze 3. iterace na požadavky

Uživatelské požadavky	Případy užití
3.3.4.1. Zobrazení grafu pro sledovanou metriku	UC1

Realizace iterace 4 - Komponenta dotazníky

Poslední realizovanou komponentou byl modul pro tvorbu dotazníkových průzkumů. Z pohledu požadovaných funkcí se nová komponenta zásadně neliší od komponenty původního systému. Hlavním požadavkem agentury však bylo vyvinout přehledné uživatelské nástroje pro tvorbu dotazníků a zjednodušit proces jejich rozesílání na e-mailové adresy respondentů.

10.1 Případy užití

Následující seznam případů užití vychází z analýzy fungování původního informačního systému.

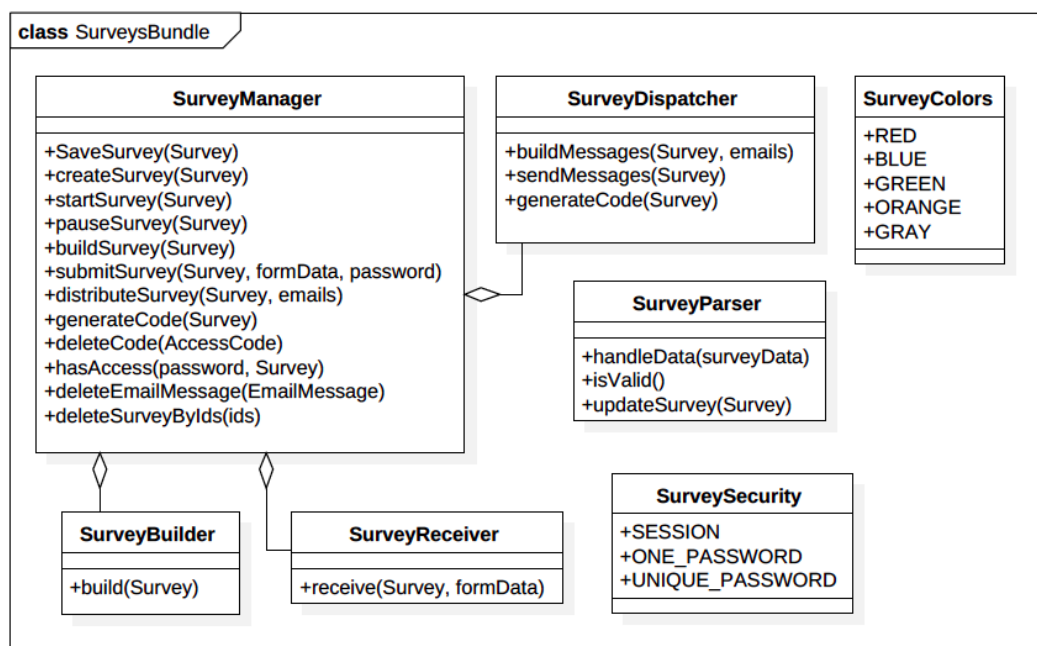
- UC1: Spravovat dotazník (přidat, upravit, smazat)
- UC2: Zveřejnit dotazník
- UC3: Rozeslat dotazník e-mailem
- UC4: Generovat unikátní hesla k dotazníkům
- UC5: Zobrazit výsledky
- UC6: Tisknout výsledky
- UC7: Exportovat výsledky do PDF souboru

10.2 Implementace

Protože byl velký důraz ze strany agentury kladen na jednoduché použití, rozhodl jsem se poskytnout uživatelům nástroje, jenž budou podporovat tvorbu

dotazníků za pomoci přetahování předdefinovaných polí formuláře. K tomuto účelu mi posloužila jQuery knihovna *FormBuilder*, kterou jsem nakonfiguroval pro potřeby komponenty dotazníků. Hlavními implementačními výzvami pak bylo zpracovat výstup knihovny *FormBuilder* na straně serveru, z těchto dat postavit objektový model dotazníku a ten distribuovat budoucím respondentům. [29]

Mnou navržený objektový model pro realizaci zmíněných cílů je možné vidět na obrázku 10.1. Stejně jako v ostatních balíčcích, třída s koncovkou *Manager* má zodpovědnost za akce prováděné nad dotazníky. *SurveyManager* má přístup ke třídě *SurveyDispatcher*, která má na starost rozeslání dotazníků respondentům. Třída *SurveyParser*, jak již název naznačuje, se stará o zpracování výstupu jQuery knihovny *FormBuilder* a její výsledná data využívá objekt třídy *SurveyBuilder*, jenž má na starost převod těchto dat do databázového modelu.



Obrázek 10.1: Diagram tříd definujících chování dotazníkových formulářů

Níže je do většího detailu popsáno fungování zmíněných tříd.

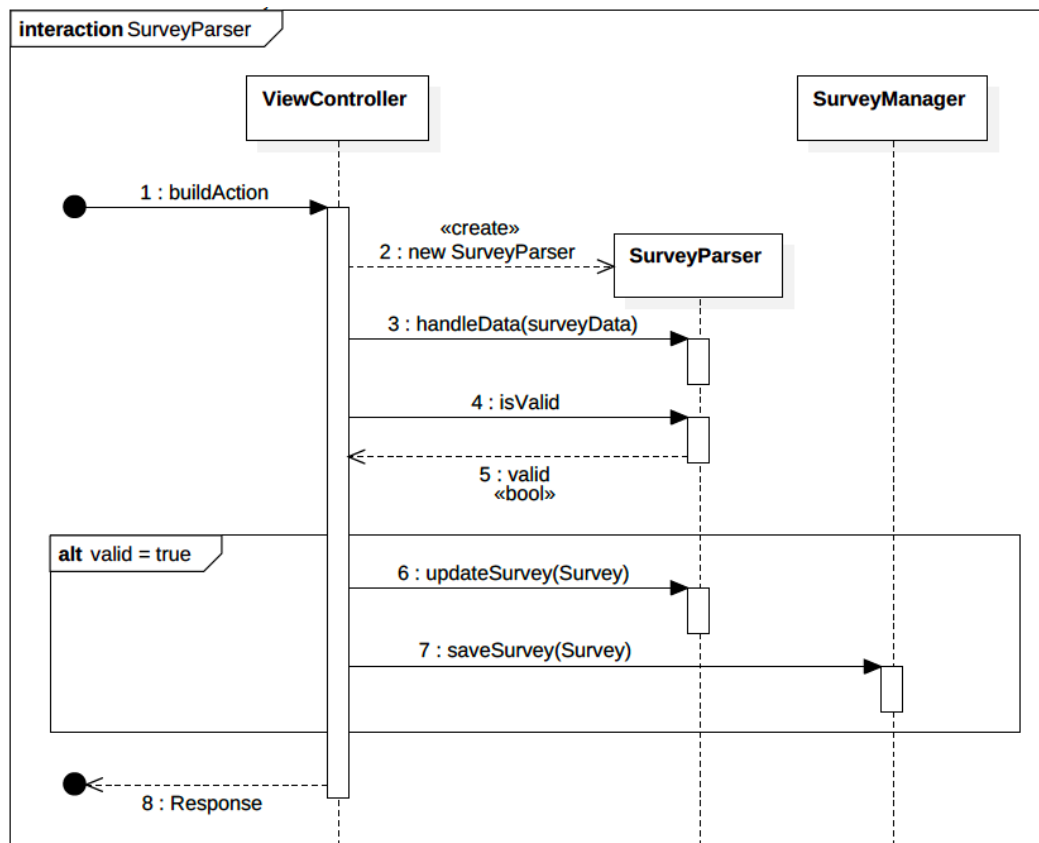
- **SurveyManager**

Skrze objekt této třídy má *controller* možnost provádět funkce související se správou dotazníků. Třída má například metody pro tvorbu, uložení, zveřejnění, rozeslání nebo zpracování výsledků dotazníkových průzkumů. Metody obstarávající tuto funkcionalitu jsou většinou velice

štíhlé, jelikož pro složitější operace využívá *SurveyManager* speciální třídy zaměřené na konkrétní problematiku, viz např. *SurveyDispatcher*.

- **SurveyParser**

Pro zpracování dat formuláře vytvořeného uživatelem slouží třída *SurveyParser*. Rozhraní této třídy je velmi jednoduché, připomíná práci s formuláři v Symfony a detailněji je interakce s ostatními třídami modelu zobrazena na obrázku 10.2. Nejprve *controller* zavolá metodu *handleData()*, které přijme uživatelem vyplněná data popisující nový dotazníkový formulář. Třída si z parametru vybere příslušná data a ty zkontroluje. Zavoláním metody *isValid()* je dále možné zjistit, zda-li byla data úspěšně přijata a je-li tomu tak, může se metodou *updateSurvey(Survey)* uložit struktura dotazníkového formuláře do objektu reprezentujícího dotazníkový průzkum.



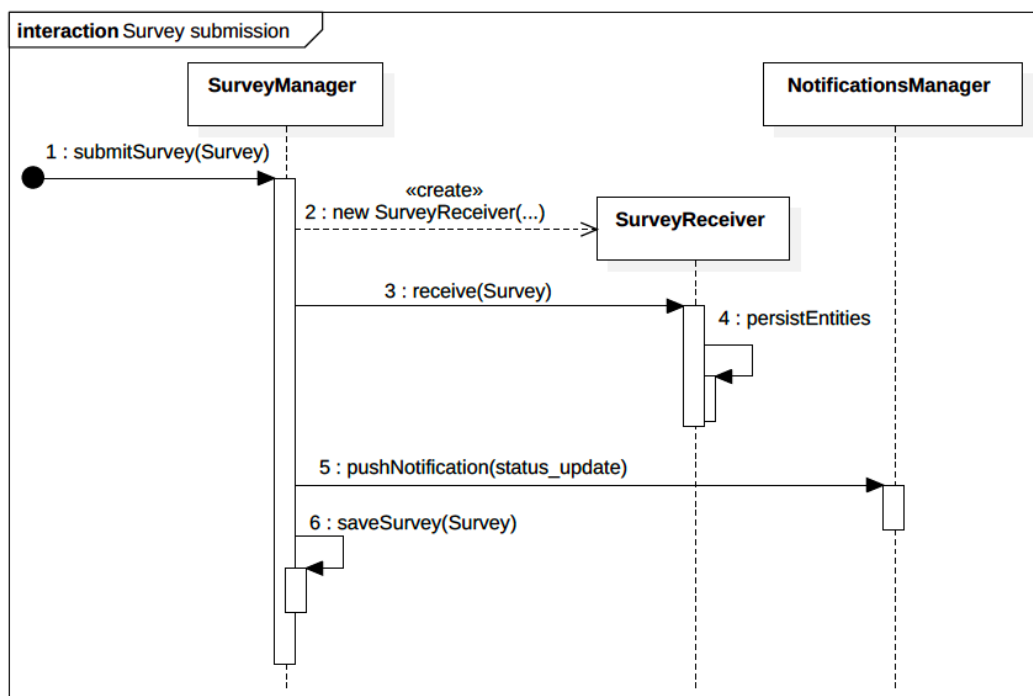
Obrázek 10.2: Zpracování respondentem vyplněného dotazníku

- **SurveyDispatcher**

Distribuce dotazníků je poměrně složitá operace a *SurveyManager* proto k této činnosti využívá třídu *SurveyDispatcher*. Tato třída poskytuje nástroje pro vygenerování e-mailových zpráv na základě zadaných adres a objektu dotazníku. Z objektu dotazníku získá *SurveyDispatcher* šablonu pro tělo e-mailu a informaci, jakým typem zabezpečení je dotazník chráněn. Pokud je dotazník přístupný po zadání hesla, musí *SurveyDispatcher* zajistit přidání tohoto hesla do e-mailu tak, aby měl každý respondent přístup pod svým unikátním kódem.

- **SurveyReceiver**

SurveyReceiver je využit třídou *SurveyManager* pro kontrolu a uložení zasláných dat z formuláře poté, co respondent vyplní dotazník. Jedinou veřejnou metodou je *Receive(Survey, FormData)*, jejíž fungování v rámci třídy *SurveyManager* znázorňuje diagram na obrázku 10.3. Metoda uloží výsledky dotazníku a v případě selhání validace je vyhozena výjimka, jež může *controller* zachytit a informovat respondenta o vzniklé situaci.



Obrázek 10.3: Zpracování respondentem vyplněného dotazníku

- **SurveyColors a SurveySecurity**

Tyto dvě třídy nemají na starost žádnou funkcionalitu a jejich jediným posláním je uchovávat konstanty, jež jsou využívány v metodách dříve

zmíněných tříd. *SurveyColors* obsahuje seznam barev, jenž může uživatel použít při vykreslení grafu s výsledky. *SurveySecurity* definuje tři typy zabezpečení dotazníkového průzkumu. Prvním typem je veřejný přístup všem uživatelům internetu. Dalšími možnostmi jsou povolení vstupu po zadání hesla, které může být společné nebo unikátní pro každého potenciálního respondenta.

10.3 Testování

Stejně jako ve většině předchozích kapitol věnovaným realizaci bylo testování komponenty dotazníku prováděno kombinací nástrojů PHPUnit a Selenium IDE.

10.3.1 PHPUnit testy

Na obrázku 10.4 je možné vidět všechny PHPUnit testovací soubory komponenty dotazníkových průzkumů. Ve složce *Controller* jsou pak uloženy automatické testy pro ověření funkčnosti stránek systému souvisejících s dotazníky.

```
src/GulpIS/SuveysBundle/Tests ..... název balíčku
├── Controller
│   ├── FrontControllerTest.php
│   ├── IndexControllerTest.php
│   └── ViewControllerTest.php
```

Obrázek 10.4: Struktura PHPUnit testů komponenty dotazníků

10.3.2 Selenium testy

Automatizované Selenium testy kopírující chování uživatele systému pokrývají funkcionalitu skrze celý proces dotazníkového průzkumu.

- **Přidání a odebrání dotazníku**

Tímto testem je ověřena základní funkcionalita zajišťující vytvoření a odebrání dotazníku ze systému.

- **Vytvoření formuláře, distribuce a vyplnění dotazníku respondentem**

Tento automatizovaný test oproti předchozímu testu navíc vytvoří formulářový dotazník (zadá otázky a odpovědi), dotazníkový průzkum spustí a vygeneruje přístupové kódy. S těmito přístupovými kódy následně vyplní dotazník a ověří, že není možné ke stejnému dotazníku přistoupit s kódem, který byl již jednou použit. Test dále ověřuje, že výsledky

dotazníku odpovídají vyplněným hodnotám a že není možné dotazník vyplnit, jakmile překročil počet respondentů určený limit.

10.4 Ověření splnění požadavků

V tabulce 10.1 jsou znázorněny funkční požadavky z dokumentu SRS a příslušné případy užití z této iterace, jež dané požadavky uspokojují.

Tabulka 10.1: Mapování případů užití ze 4. iterace na požadavky

Uživatelské požadavky	Případy užití
3.3.3.1. Vytvoření a správa nového dotazníkového průzkumu	UC1
3.3.3.2. Distribuce dotazníkového průzkumu	UC2, UC3, UC4
3.3.3.3. Zobrazení výsledků dotazníku	UC5
3.3.3.4. Export a tisk dotazníku	UC6, UC7

S použitím HTTPS protokolu pro přístup k formulářům dotazníkových průzkumů byl uspokojen nefunkční požadavek na bezpečnost 3.4.1.1.

Dále sekce věnované implementaci v této kapitole ukázaly, že při návrhu byl kladen důraz na rozšiřitelnost a tudíž byl naplněn i nefunkční požadavek 3.4.4 z dokumentu požadavků SRS.

Analýza rizik a ekonomických dopadů nasazení nového IS

V této kapitole nabízím manažersko-ekonomický pohled na nasazení nového informačního systému, s nímž se pojí potenciální rizika a finanční zátěž pro agenturu. Nejprve zmiňuji možná rizika a mnou navržený způsob jejich mitigace. Následuje ekonomická analýza na základě časové náročnosti zjištěné během implementace a porovnání výhodnosti mého řešení s možnostmi zmíněnými v kapitole 5. Analýza možných řešení.

11.1 Analýza rizik spojených s nasazením IS

Byl identifikován možný výskyt následujících tří rizik provázející proces nasazení. Tato rizika mají potenciální právní a ekonomické důsledky pro fungování personální agentury a je proto nutné jim věnovat pozornost. Mnoho častých rizik spojených s vývojem softwaru se podařilo zmírnit využitím populárních knihoven a programů, čímž se zmenšil objem nově vyvíjeného zdrojového kódu a tím i možných chyb. [21]

11.1.1 Komplikace zaměstnanců agentury při přechodu na nový IS

Protože byl původní systém agenturou AW Partners roky využíván, jsou nyní její zaměstnanci zvyklí na původní uživatelské rozhraní. Přestože mnoho funkcí bylo zachováno nebo prošlo pouze drobnými změnami, z pohledu uživatelského rozhraní docházelo k dramatickým změnám v rámci všech komponent. Pro minimalizaci případných komplikací zaměstnanců agentury při zahájení práce s novým systémem jsem vypracoval náповědu ve formě několika textových dokumentů k hlavním funkcím systému. Tyto dokumenty je možné nalézt na CD přiloženém k této práci ve složce *documentation*. Rovněž jsem na někte-

rých místech v systému umístil malé otazníky, které po kliknutí tlačítkem myši zobrazují nápovědu k dané funkcionalitě.

11.1.2 Komplikace při migraci dat z původní databáze

Ačkoliv by se mohla migrace dat jevit jako čistě technická záležitost, je třeba si uvědomit, že manipulace s osobními údaji je velmi riziková záležitost s potenciálně vážnými důsledky. Po celou dobu vývoje systému je třeba dbát na dodržování zákona č. 101/2000 Sb., o ochraně osobních údajů. Zároveň je ale potřeba zajistit bezproblémový přesun všech informací, přičemž výskyt chyby by znamenal nejen možné poškození agentury, ale především také osoby s jejíž daty je manipulováno. Problematiku migrace dat jsem vyřešil vytvořením samostatného balíčku (bundle), který slouží pouze k podpoře migrace dat při nasazení a definuje všechna potřebná pravidla pro přesun dat mezi novou a původní databází. Na obrázku B.9 je vidět uživatelské rozhraní této pomocné komponenty. Tlačítka na obrazovce slouží k zahájení přesunu vybraných tabulek z databáze nebo souborů přidružených k uchazečům. Tímto způsobem je migrace dat exaktně definovaná a případné komplikace vyžadující přemazání celé databáze se stávají záležitostmi vyžadující opravu minimálně jednoho řádku kódu. [30]

11.1.3 Nedostupnost systému neprodleně po nasazení

V kapitole popisující fungování původního systému již bylo zmíněno, že je systém využíván agenturou každý den včetně víkendů. Je proto naprosto kritické, aby při nasazení nového systému nedošlo k žádným dlouhodobým výpadkům, které by mohly narušit pracovní činnost zaměstnanců. Abych zajistil správné fungování a dostupnost systému od prvního dne jeho nasazení, domluvil jsem s agenturou zakoupení testovací domény a její umístění na stejném serveru, kde bude nový IS umístěn. Na této dočasné doméně jsem následně spustil testovací verzi systému, do které byl zástupcům agentury umožněn přístup pro provedení akceptačního testování. Po odsouhlasení přesunu systému na hlavní doménu stačí pouze zaslat požadavek na přenos domény hostingové společnosti. Tímto se mi podařilo zajistit, že termín výpadku spojeného s přesunem domény lze dopředu naplánovat na dohodnutou dobu a nebude ohroženo běžné fungování společnosti.

11.2 Ekonomické dopady vývoje IS

Ještě před započítáním implementace, při analýze možných řešení pro uspokojení uživatelských požadavků, jsem došel k závěru, že vývoj nového informačního systému pro podporu firemních procesů agentury je po finanční stránce nejvýhodnější možné řešení. V této sekci se k tomuto tvrzení vracím a s vy-

užitím dat získaných při realizaci systému analyzuji celkové časové a finanční náklady na vývoj informačního systému agentury AW Partners.

Přestože bylo analýzou vyloučeno využití řešení nabízeného formou služby (SaaS) z důvodů nevhodnosti pro podporu konkrétních požadavků agentury AW Partners, nejsou tyto závěry obecné pro potřeby všech personálních agentur. Z tohoto důvodu je vhodné analyzovat rozdíl mezi využitím některé ze zmíněných služeb a vývojem vlastního systému.

11.2.1 Časová náročnost

Při realizaci nového informačního systému jsem pomocí nástroje Toggl Desktop zaznamenával čas strávený různými částmi procesu softwarového vývoje. V následující tabulce 11.1 uvádím naměřená data, jejichž detailní výpis je možné najít na přiloženém CD ve složce *documentation*. Do analýzy nezahrnuji čas strávený psaním textu bakalářské práce. [31]

Tabulka 11.1: Čas strávený aktivitami procesu softwarového vývoje

Typ činnosti	Naměřený čas [hod.]
Sběr požadavků	27
Návrh systému	17
Implementace	305
Testování	49
Nasazení	26
Komunikace s agenturou	6
Ostatní	9
Celkem	439

Je nutné zmínit, že ačkoliv tyto údaje velmi přesně vystihují časovou náročnost mnou implementovaného systému, jsou závislé na mých schopnostech a jiný vývojář či firma by zajisté naměřili mírně odlišné výsledky. Celkový naměřený čas 439 hodin budu každopádně dále využívat při analýze celkových nákladů, jelikož je to nejpřesnější možný odhad, který mám k dispozici.

11.2.2 Faktory ovlivňující celkové náklady

Kromě nákladů na programátorské služby dále do analýzy zahrnuji následující faktory ovlivňující celkovou cenu vývoje nového IS:

- **Hostingové služby**

Pronajmutí serverového prostoru a registraci doménového jména je nutné do ceny vývoje zahrnout nejen v prvním roce, ale také přičíst ke každému následujícímu roku fungování systému. Pro uspokojení všech nároků systému se mi podařilo nalézt hostingové služby v ceně 79 Kč za měsíc bez

DPH a roční poplatek 148 Kč za správu domény u společnosti Blueboard s.r.o. Celkové roční náklady proto činí 1090 Kč bez DPH. [32]

- **Čas zaměstnanců agentury**

Je také potřeba do celkové ceny zahrnout čas, jenž zástupci vedení agentury AW Partners poskytli při procesu zjišťování požadavků na systém. Pro stanovení tohoto údaje vycházím z délky trvání společných schůzek (6 hodin), které násobím průměrnou cenou zaměstnance. Pro zjištění této ceny využívám statistik ISPV (Informační systém o průměrném výdělků) pro 4. čtvrtletí 2015 v Jihomoravském kraji, kde má agentura sídlo. Průměrná hrubá mzda specialisty v oboru personálního řízení činila 263,1 Kč za hodinu. Po vynásobení hrubé mzdy číslem 1.34 získávám tzv. super hrubou hodinovou mzdu 316.4 Kč, což je náklad agentury na zaměstnance dále využitý pro účely analýzy. [33]

- **Cena údržby**

Při stanovení ceny za údržbu vycházím z nákladů na servisní služby předešlého systému. V průměru lze očekávat, že bude nutné vynaložit 10 hodin ročně na drobné úpravy či přidání nové funkcionality. Dále předpokládám, že v prvním roce po spuštění systému bude tato částka dvojnásobná a až od druhého roku používání systému se ustálí na zmíněných 10 hodinách za rok.

11.2.3 Náklady na vývoj nového IS

Pokud bych nyní pro zjištění celkových nákladů použil veškerý vynaložený čas, ten vynásobil hodinovou sazbou a připočetl výše zmíněné položky ovlivňující cenu, dopustil bych se zásadní chyby. Již jsem dokázal, že je možné IS implementovat jednotlivcem a lze proto předpokládat, že práci by mohl vykonat živnostník. Pokud by ovšem byla oslovena firma, bude cena zajisté vyšší dle velikosti oslovené firmy. Rozhodl jsem se proto brát jako předpoklad 3 různé hodinové sazby dle tabulky 11.2 pro případ využití služeb živnostníka, malé či velké firmy. Zajisté se najdou živnostníci, kteří se mohou cenami pohybovat v cenové hladině mnou přiřazené velkým firmám, jsou proto důležité vybrané hodnoty a jejich označení pouze reflektuje fakt, že drobní živnostníci bývají zpravidla levnější než velké firmy.

Tabulka 11.2: Použité hodnoty hodinových sazeb za služby

Poskytovatel služeb	Hodinová sazba
Živnostník	400 Kč
Malá firma	700 Kč
Velká firma	1000 Kč

S použitím hodnot z tabulky 11.2 nyní mohu vyčíslit konečnou cenu pro zmíněné 3 cenové kategorie. Pro výpočet nákladů ($Cena_z$) vynaložených na zřízení informačního systému používám následující vzorec 11.1.

$$Cena_z = (Sazba * Čas_z) + Cena_a \quad (11.1)$$

- $Cena_z$ - zřizovací náklady
- $Sazba$ - hodinová sazba služeb programování podle tabulky 11.2
- $Čas_z$ - dříve naměřená hodnota 439 hodin strávených vývojem systému
- $Cena_a$ - výdaje agentury spojené s časem vydaným zástupci na společné schůze (1900 Kč)

Následující vzorec 11.2 využiji pro výpočet nákladů ($Cena_u(i)$) spojených s údržbou systému v i -tém roce.

$$Cena_u(i) = (Sazba * Čas(i)) + Cena_h \quad (11.2)$$

- $Cena_u(i)$ - cena údržby v roce i
- $Čas(i)$ - odhad času stráveného údržbou v roce i
- $Cena_h$ - cena hostingových služeb (1090 Kč)

Pro zjištění nákladů ($Cena_{Xlet}$) spojených s využíváním informačního systému po dobu X let provedu součet výsledků předešlých vzorců.

$$Cena_{Xlet} = Cena_z + (Cena_u(1) + Cena_u(2) * (X - 1)) \quad (11.3)$$

- $Cena_{Xlet}$ - náklady na realizaci a údržbu systému po dobu X let
- $Cena_u(1)$ - náklady na údržbu v prvním roce
- $Cena_u(2)$ - náklady na údržbu ve druhém roce (stejně pro všechny následující roky)

Výše uvedené vzorce jsem použil při výpočtu hodnot vypsanych v tabulce 11.3, kde jsou uvedeny náklady na zřízení a údržbu systému po dobu až 10 let.

Tabulka 11.3: Celkové náklady na vývoj nového IS

	Živnostník	Malá firma	Velká firma
Zřízovací náklady	177 500 Kč	309 200 Kč	440 900 Kč
Údržba: 1. rok	9 090 Kč	15 090 Kč	21 090 Kč
Údržba: 2. rok a déle	5 090 Kč	8 090 Kč	11 090 Kč
Celkové náklady za 5 let	206 950 Kč	356 650 Kč	506 350 Kč
Celkové náklady za 10 let	232 400 Kč	397 100 Kč	561 800 Kč

11.2.4 Porovnání ceny vývoje nového IS se službami SaaS

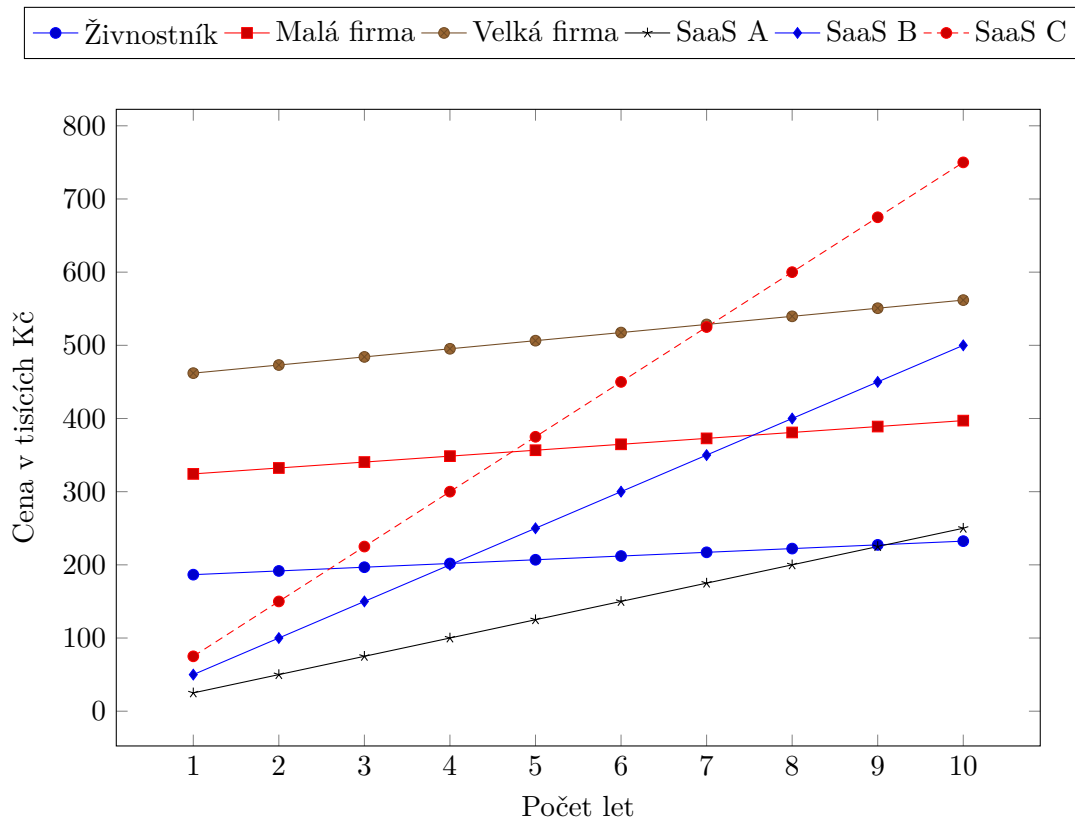
Na trhu je k dispozici nepřeberné množství e-recruitment systémů poskytovaných formou služby, jejichž funkce i cena se dramaticky liší. O velké dynamice tohoto segmentu trhu svědčí například i fakt, že během psaní této práce změnil dříve analyzovaný systém iKrut ceník svých služeb. Různí poskytovatelé používají jiné způsoby výpočtu ceny, např. se může výše platby odrážet od počtu uživatelů systému nebo počtu agenturou nabízených pozic. [34]

Z tohoto důvodu nebudu nyní analyzovat výhodnost konkrétního řešení, ale obecně výši ceny vzhledem k předchozím výpočtům nákladů na vývoj nového systému. Pro zjednodušení předpokládám fixní roční sazby v hodnotách 30 (A), 60 (B) a 90 (C) tisíc Kč za rok. Graf 11.1 porovnává dříve získané informace o ceně mnou vyvinutého systému s navrženými sazbami SaaS řešení v horizontu 10 let. Z grafu lze vyčíst 3 zajímavé informace, které mohou být vodítkem při rozhodování, zda-li zakoupit systém formou SaaS nebo vyvíjet vlastní.

1. Při zvolení vhodného SaaS řešení, které bude splňovat požadavky agentury, za roční cenu 25 tisíc Kč (SaaS A) a méně bude tato varianta v horizontu do 9 let lepší investicí než vývoj vlastního informačního systému a to i v případě oslovení živnostníka. Naopak, má-li být předpokládána životnost delší než 9 let a zvažované SaaS řešení stojí ročně přes 25 tisíc Kč, pak se v případě oslovení živnostníka vyplatí implementovat vlastní řešení IS.
2. Je-li roční cena vyhovujícího SaaS řešení 50 tisíc Kč a méně, pak v případě předpokládané doby životnosti 10 let nemá smysl oslovovat s žádostí o vývoj nového systému velkou firmu. V opačném případě, pokud by byla cena SaaS řešení vyšší, pak by oslovení velké firmy dávalo z ekonomického hlediska smysl.
3. Je-li roční cena zvažovaného SaaS řešení více jak 75 tisíc Kč, pak má vývoj vlastního systému velkou firmou z ekonomického hlediska smysl v případě předpokládané životnosti více jak 7 let. Má-li být využito služeb malé firmy, pak by předpokládaná životnost musela být alespoň necelých 5 let a přes 2 a půl roku v případě živnostníka.

Ze zmíněných bodů vyplývá, že není možné s absolutní přesností říci, kdy je zvažované SaaS řešení ekonomicky výhodnější než vývoj vlastního systému a vždy bude záležet minimálně na předpokládané životnosti IS. Dalším argumentem pro vývoj vlastního systému v případě dlouhé předpokládané životnosti je riziko výskytu nových požadavků, které nebude zvažované SaaS řešení splňovat. Možnost přizpůsobení systému vlastním potřebám je ovšem u SaaS systémů faktorem zvyšujícím cenu. [9]

Konečně je pak nutné připomenout, že analýza v této kapitole je velmi závislá na konkrétních požadavcích agentury AW Partners, a proto není možné aplikovat zmíněné závěry na případ všech personálních agentur.



Obrázek 11.1: Porovnání ceny vývoje nového IS oproti SaaS řešení

11.2.5 Podpora firemních procesů

Nově vytvořený informační systém pro agenturu AW Partners byl navržen pro lepší podporu firemního procesu náboru popsaného v první kapitole. Oproti původnímu systému se ovšem nejedná o tak významné změny, aby je bylo

11. ANALÝZA RIZIK A EKONOMICKÝCH DOPADŮ NAsAZENÍ NOVÉHO IS

možné zahrnout do ekonomické analýzy v předešlých sekcích této kapitoly. Jak ovšem ukázala analýza uživatelských požadavků, nebylo zvýšení zisku zkvalitněním podpory firemních procesů novým IS primárním cílem jeho nasazení. Těmi hlavními cíli bylo navrhnout uživatelsky přívětivější a lépe zabezpečený systém. Přínosem nového řešení z pohledu vyššího zabezpečení je pak zamezení možných ztrát způsobených únikem citlivých informací.

Závěr

Cílem práce bylo analyzovat, navrhnout, implementovat, otestovat a nasadit informační systém personální agentury AW Partners s.r.o.

Na základě analýzy fungování firemních procesů, původního systému a požadavků agentury byly zváženy varianty řešení formou vylepšení původního systému, využití SaaS řešení nebo vývoje nového informačního systému. Převažně kvůli požadavkům týkajících se bezpečnosti a rozširitelnosti byla vybrána možnost poslední, tj. realizace nového informačního systému.

Architekturu aplikace jsem navrhl na základě výzkumu profesora In Leeho, Ph.D. s důrazem na konkrétní požadavky agentury. Pro realizaci jsem se rozhodl využít PHP framework Symfony ve verzi 3.0 a jako základ uživatelského rozhraní mi posloužila šablona AdminLTE. Protože byla aplikace navržena modulárně a uživatelské požadavky byly dopředu dostatečně jasné, využil jsem pro vývoj metodiku využívající iterační přístup Rational Unified Process.

Výsledkem implementační části práce je fungující informační systém splňující všechny agenturou definované požadavky s důrazem na bezpečnost a rozširitelnost. V rámci jednotlivých iterací byl systém řádně uživatelsky i strojově otestován za pomoci nástrojů PHPUnit a Selenium IDE. Každá iterace byla rovněž zakončena verifikací navržené funkcionality oproti požadavkům agentury. Po skončení realizace byla výsledná aplikace úspěšně nasazena na produkční server a agentura obdržela uživatelskou dokumentaci popisující fungování hlavních komponent systému.

V závěrečné kapitole jsem popsal tři potenciální rizika provázející nasazení nového systému a mnou zvolené řešení jejich mitigace. Následně byla provedena analýza ekonomických dopadů vývoje nového systému a výsledná cena byla porovnána se sazbami SaaS řešení.

Profesor In Lee, Ph.D. v závěru svého pojednání o návrhu holistického e-recruitment informačního systému zdůrazňuje fakt, že komponenty jím popsaného systému nemusí být nutně implementovány všechny, ale každá firma musí určit ty, které jsou pro její činnost prioritní. V tomto duchu byl navržen

ZÁVĚR

také informační systém společnosti AW Partners popsany v této bakalářské práci. Do budoucna vidím jako reálnou možnost implementovat některé další komponenty podle In Leeho, především pak modul pro strojové ohodnocení přihlášených kandidátů na pozici a modul pro podporu udržení dlouhodobých vztahů s uchazeči z databáze.

Literatura

- [1] Služby - AWPpartners.cz [online]. 2009. Dostupné z: <http://www.awpartners.cz/sluzby>
- [2] Lee, I.: Modeling the benefit of e-recruiting process integration [online]. *Decision Support Systems*, ročník vol. 51, č. issue 1, 2011: s. 230–239, ISSN 01679236, doi:10.1016/j.dss.2010.12.011. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0167923611000029>
- [3] Lang, S.; Laumer, S.; Maier, C.; aj.: Drivers, challenges and consequences of E-recruiting. *Proceedings of the 49th SIGMIS annual conference on Computer personnel research - SIGMIS-CPR '11 [online]*, 2011: s. 26–35, doi:10.1145/1982143.1982152. Dostupné z: <http://portal.acm.org/citation.cfm?doid=1982143.1982152>
- [4] Čechová, L.; Haisová, E.: SOUMRAK AGENTURNÍHO ZAMĚSTNÁVÁNÍ? [online]. 2016. Dostupné z: <http://www.epravo.cz/top/clanky/soumrak-agenturniho-zamestnavani-98014.html>
- [5] Eeles, P.; Cripps, P.: *Architektura softwaru*. Brno: Computer Press, vyd. 1. vydání, 2011, ISBN 978-80-251-3036-0.
- [6] IKrut - Online Recruitment System by Zodo [online]. 2012. Dostupné z: <https://ikrut.com>
- [7] Recruiting Software and Applicant Tracking System - Recruiterbox.com [online]. 2016. Dostupné z: <http://recruiterbox.com/>
- [8] Qandidate.com Your Free Full Blown Recruitment System [online]. 2016. Dostupné z: <http://qandidate.com/>
- [9] Sun, W.; Zhang, X.; Guo, C. J.; aj.: Software as a Service. *2008 IEEE Congress on Services Part II (services-2 2008)*, 2008: s. 18–25, doi:10.1109/SERVICES-2.2008.29. Dostupné z: <http://>

[//ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4700495](http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4700495)

- [10] Lee, I.: An architecture for a next-generation holistic e-recruiting system. *Communications of the ACM [online]*, ročník vol. 50, č. issue 7, 2007-07-01: s. 81–85, ISSN 00010782, doi:10.1145/1272516.1272518. Dostupné z: <http://portal.acm.org/citation.cfm?doid=1272516.1272518>
- [11] Bruckner, T.: *Tvorba informačních systémů*. Praha: Grada, první vydání, 2012, ISBN 978-80-247-4153-6.
- [12] Buchalcevoá, A.: *Metodiky vývoje a údržby informačních systémů*. Praha: Grada, první vydání, 2005, ISBN 80-247-1075-7.
- [13] Rational Unified Process Best Practices for Software Development Teams [online]. 1998. Dostupné z: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- [14] Wiegers, K. E.; Beatty, J.: *Software requirements*. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation, third edition. vydání, [2013], ISBN 07-356-7966-5.
- [15] PHP 5 Tutorial [online]. 2016. Dostupné z: <http://www.w3schools.com/php/default.asp>
- [16] Potencier, F.: Symfony2 Security Audit [online]. 2011. Dostupné z: <http://symfony.com/blog/symfony2-security-audit>
- [17] AdminLTE - Almsaeed Studio [online]. 2016. Dostupné z: <https://almsaeedstudio.com/preview>
- [18] W3Schools: jQuery Introduction [online]. 2016. Dostupné z: http://www.w3schools.com/jquery/jquery_intro.asp
- [19] W3Schools: Bootstrap 3 Tutorial [online]. 2016. Dostupné z: <http://www.w3schools.com/bootstrap/>
- [20] DataTables | Table plug-in for jQuery [online]. 2016. Dostupné z: <https://www.datatables.net/>
- [21] Sommerville, I.: *Softwarové inženýrství*. Brno: Computer Press, první vydání, 2013, ISBN 978-80-251-3826-7.
- [22] The Best Practices Book [online]. 2016. Dostupné z: http://symfony.com/pdf/Symfony_best_practices_3.0.pdf?v=4
- [23] KnpMenuBundle [online]. 2011. Dostupné z: <https://github.com/Knplabs/KnpMenuBundle>

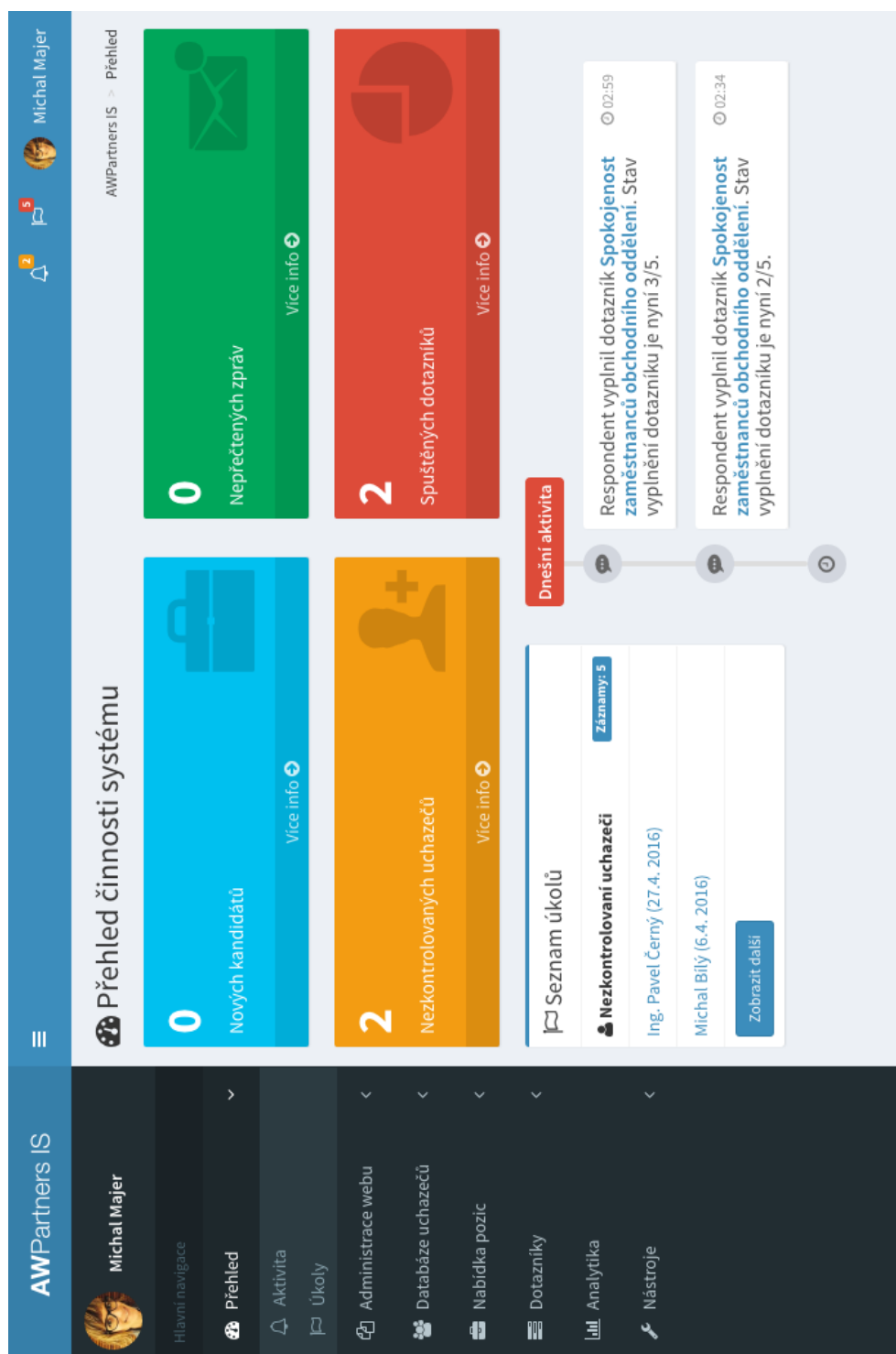
-
- [24] FOSUserBundle [online]. 2011. Dostupné z: <https://github.com/FriendsOfSymfony/FOSUserBundle>
- [25] JQuery-QueryBuilder [online]. 2014-2015. Dostupné z: <https://github.com/mistic100/jQuery-QueryBuilder>
- [26] VichUploaderBundle [online]. 2011. Dostupné z: <https://github.com/dustin10/VichUploaderBundle/blob/master/Resources/doc/index.md>
- [27] ChartJS [online]. 2013-2016. Dostupné z: <https://github.com/chartjs/Chart.js>
- [28] Knihovny rozříšení Doctrine (Day, Month, Year) [online]. 2012. Dostupné z: <https://github.com/wiredmedia/doctrine-extensions/tree/master/lib/DoctrineExtensions/Query/Mysql>
- [29] FormBuilder [online]. 2016. Dostupné z: <http://formbuilder.readthedocs.io/>
- [30] Data migration risks [online]. 2015. Dostupné z: <http://www.data-migrations.com/risks.html>
- [31] Toggl - Free Time Tracking Software [online]. 2016. Dostupné z: <https://toggl.com/>
- [32] Webhosting - Blueboard.cz [online]. 2016. Dostupné z: <https://hosting.blueboard.cz/hosting/>
- [33] Regionální statistika ceny práce [online]. 2016. Dostupné z: <http://www.ispv.cz/cz/Vysledky-setreni/Aktualni.aspx>
- [34] What Does Recruiting Software Cost? [online]. 2014. Dostupné z: <http://blog.capterra.com/recruiting-software-cost/>

Seznam použitých zkratek

- CMS** Content Management System
- CSS** Cascading Style Sheets
- HTML** HyperText Markup Language
- HTTPS** Hypertext Transfer Protocol Secure
- IS** Informační systém
- JS** JavaScript
- MySQL** My Structured Query Language
- OÚ** Osobní údaje
- PDF** Portable Document Format
- PHP** Hypertext Preprocessor
- SaaS** Software as a Service
- SSL** Secure Sockets Layer
- SRS** Structured requirements specification
- RUP** Rational Unified Process
- UML** Unified Modeling Language

Ukázka nejdůležitějších funkcí systému

B. UKÁZKA NEJDŮLEŽITĚJŠÍCH FUNKCÍ SYSTÉMU



Obrázek B.1: Úvodní obrazovka nového systému

AWPartners IS > Databáze uchazečů > Vyhledávání

Michal Majer

Databáze uchazečů

Vyhledávání

A Zároveň Nebo

Obor je jedním z **Administrativa**

Město (výkon práce) je jedním z **Praha**

+ Nové pravidlo Nová skupina

X Odebrat X Odebrat

Vyhledat

Zobraz záznamů 15 Hledat:

Jméno	Pozice	Obor	Město	Datum přidání
Petr Červený	Asistent/ka	Administrativa	Praha	22.8. 2014
Bc. Michal Zlatý	Key Account Manager	Administrativa	Praha	10.1. 2013
Ing. Tomáš Bílý	Administrativní pracovník/pracovnice	Administrativa	Praha, Brno	16.9. 2013
Ing. Lucie Černá	Asistent/ka	Administrativa	Praha	12.11. 2013
Igor Modrý	Administrativní pracovník/pracovnice, Obchodní manažer/ka	Administrativa, Prodej – obchod	Praha	19.3. 2015
Jan Hrnědý	Administrativní pracovník/pracovnice, Key	Administrativa, Prodej –	Praha,	15.8. 2014

Obrázek B.2: Vyhledávání v databázi uchazečů

B. UKÁZKA NEJDŮLEŽITĚJŠÍCH FUNKCÍ SYSTÉMU

AWPartners IS > Databáze uchazečů > Přidat uchazeče

Michal Majer

Přidání nového uchazeče

Vše

- Základní info
- Vzdělání
- Lokalita
- Jazykové dovednosti
- Ostatní

Uložit uchazeče

Titul

Ing.

Jméno

Příjmení

E-mail

Telefon

Pozice

Specializace

Mzdové požadavky

Dosažené vzdělání

Vzdělání VŠ

Obrázek B.3: Přidání záznamu nového uchazeče

AWPartners IS > Databáze uchazečů > Vyhledávání

Michal Majer

798

2

Ing. Pavel Černý - karta uchazeče

Info CV Nabídnuté pozice Doporučení Akce

Dosažené vzdělání: Vysokoškolské - univerzitní

Vzdělání VŠ: Informační technologie

Obor: Informační technologie

Specializace: Informační systémy a management

Rok ukončení: 2016

Vzdělání SŠ: Informační technologie

Obor: Informační technologie

Specializace: Počítačové sítě

Rok ukončení: 2012

Místo výkonu práce: Praha
Hlavní město Praha
Česká republika

IT

Pozice: Programátor

E-mail: pavel@cerny.test

Telefon: 777999000

WWW: Offline

Přidání: 27.4.2016

Souhlas se zpracováním OÚ: 27.4.2016

Dne: 27.4.2016

Zbývá dní: 723

Přidat soubor

Obrázek B.4: Karta uchazeče z databáze

B. UKÁZKA NEJDŮLEŽITĚJŠÍCH FUNKCÍ SYSTÉMU

The screenshot displays a web application interface for creating a survey form. The interface is in Czech and features a sidebar on the left with navigation options: Přehled, Administrace webu, Data báze uchazečů, Nabídka pozic, Dotazníky, Analytika, and Nástroje. The main content area is titled "Tvorba dotazníku: test" and contains a survey form titled "Dotazník spokojenosti zaměstnanců". The survey includes a greeting, a question about satisfaction with the company, and a question about the relationship with the company. The right-hand panel contains form controls such as "Vyčistit" and "Uložit formulář", a color palette for responses, and a "Přidat možnost" button.

AWPartners IS

Michael Majer

788

2

H Nadpis

Odstavec textu

Otazka

Dostupné barvy odpovědí:

- red
- blue
- green
- orange
- gray

Vyčistit

Uložit formulář

Tvorba dotazníku: test

Dotazník spokojenosti zaměstnanců

Dobrý den, rádi bychom Vám položili následující otázky.

Jste spokojeni ve svém zaměstnání?

Ano

Spíše ano

Spíše ne

Ne

Máte dobrý vztah se svými nadřízenými?

Text

Máte dobrý vztah se svými nadřízenými?

Možnosti

Ano green

Ne red

Přidat možnost

Zavřít

Obrázek B.5: Tvorba formuláře dotazníkového průzkumu

AWPartners IS > Dotazníky > Seznam dotazníků

Michal Majer 798 2

Obecné údaje Distribuce Výsledky

[Tisknout výsledky](#)

Spokojenost zaměstnanců obchodního oddělení Aktivní

Status: 3/5

Stav vyplnění: 3/5

Typ zabezpečení: Unikátní heslo

Datum přidání: 15.4.2016

[Pozastavit dotazník](#)

[Zobrazit dotazník](#)

Jaký máte pocit ze svých nadřízených?

Dobry (0)
Spíše dobry (1)
Spíše špatny (2)
Špatny (0)

Pohádal/a jste se někdy se svým nadřízeným?

Ne (2)
Ano (1)

Mají vztahy s nadřízenými vliv na vaši produktivitu?

Ano, rozhodně (0)
Spíše ano (0)
Spíše ne (1)
Rozhodně ne (2)

AWPartners IS

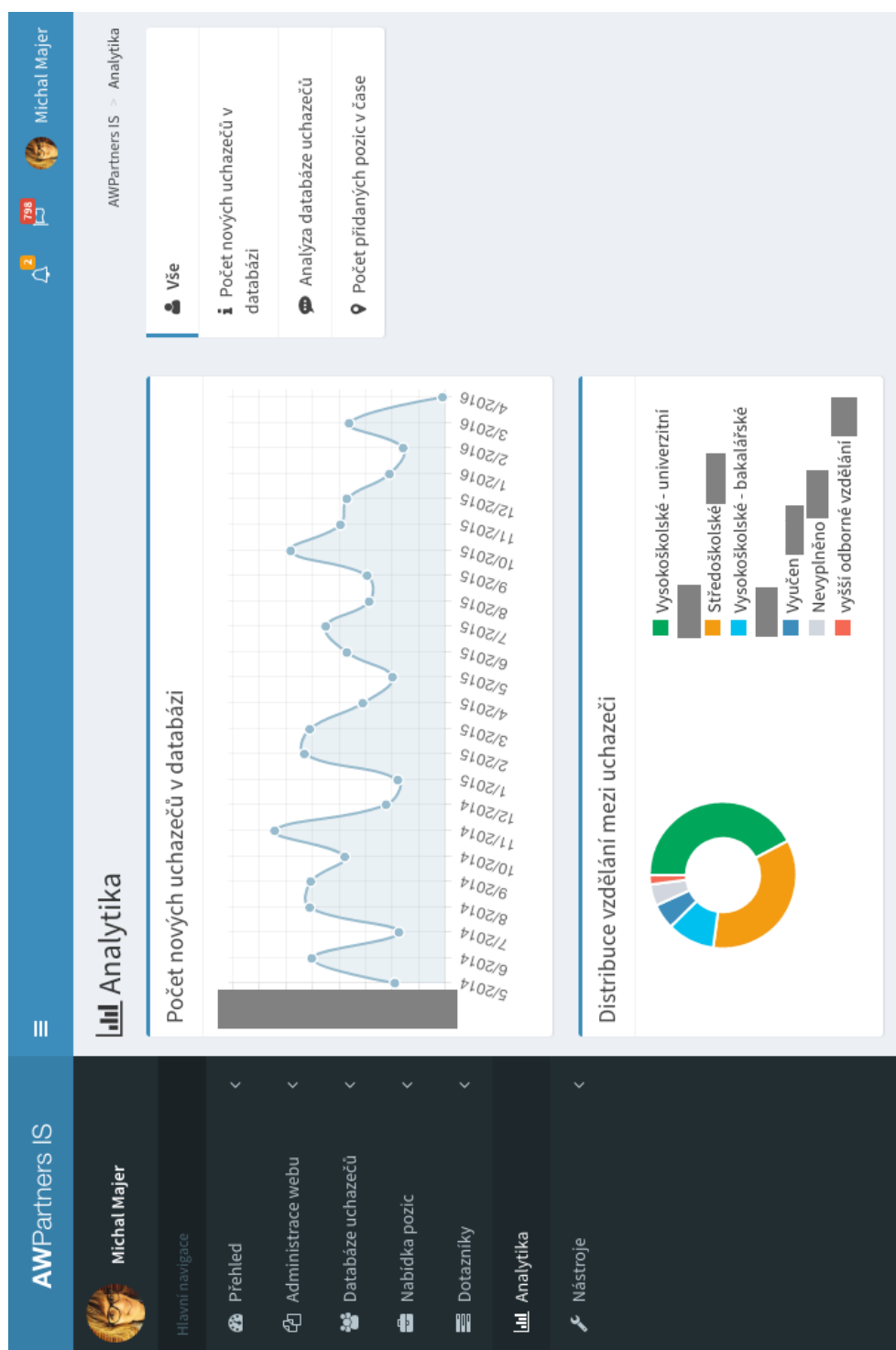
Michal Majer

Hlavní navigace

- Přehled
- Administrace webu
- Databáze uchazečů
- Nabídka pozic
- Dotazníky
- Seznam dotazníků
- Vytvořit nový
- Analytika
- Nástroje

Obrázek B.6: Zobrazení výsledků dotazníkového průzkumu

B. UKÁZKA NEJDŮLEŽITĚJŠÍCH FUNKCÍ SYSTÉMU



Obrázek B.7: Zobrazení grafů analytické komponenty

AWPartners IS > Nástroje > Správa uživatelů

Michal Majer

798

2

Přidání uživatele

Údaje o uživateli

Uživatelské jméno
zamestnanec1

Skutečné jméno
Pavel Černý

E-mail
pevel@cerny.test

Heslo
.....

Kontrola hesla
.....

Typ účtu
Běžný uživatelský účet

Role

- Zobrazení databáze uchazečů
- Zobrazení databáze pozic
- Zobrazení kontaktních formulářů z webu
- Úprava webových stránek
- Úprava databáze uchazečů
- Úprava databáze pozic
- Komponenta dotazníků
- Komponenta analytika
- Nástroje pro správu systému

Obrázek B.8: Přidání účtu nového uživatele systému

B. UKÁZKA NEJDŮLEŽITĚJŠÍCH FUNKCÍ SYSTÉMU

Migrate data

Tabulky pro migraci

Před migrací se prosím ujistěte, že jsou všechny tabulky prázdné a následně pokračujte v importech dat podle následujícího pořadí.

- Obory - 39 Záznamů. [Načíst](#) [Smazat](#)
- Pozice - 191 Záznamů. [Načíst](#) [Smazat](#)
- Uchazeči - 3263 Záznamů. [Načíst](#) [Přidat pozice](#) [Přidat obory](#) [Přidat země](#) [Přidat kraje](#) [Přidat města](#)
- Načíst data k online uchazečům [Smazat](#)
- Nabídky - 1137 Záznamů. [Smazat](#) [Načíst firmy](#) [Načíst nabídnuté pozice k uchazečům](#) [Načíst aktivní pozice k uchazečům](#)
- Formuláře - 60 Záznamů. [Načíst](#) [Smazat](#)
- Přílohy uchazečů [Načíst](#) [Smazat](#)

AWPartners IS

Michal Majer

Hlavní navigace

- Přehled
- Administrace webu
- Databáze uchazečů
- Nabídka pozic
- Dotazníky
- Analýtika
- Nástroje

Obrázek B.9: Nástroje pro migraci dat z původní databáze

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
documentation.....	adresář dokumentace
├─ SRS.....	dokument specifikace požadavků
├─ toggl_report.pdf.....	dokumentace vynaložených časových nákladů
├─ user_documentation.....	uživatelské příručky
├─ ats.pdf.....	nápověda k databázi uchazečů
├─ surveys.pdf.....	nápověda k dotazníkům
src	
├─ impl.....	zdrojové kódy implementace
├─ selenium.....	zdrojové soubory SeleniumIDE testů
├─ thesis.....	zdrojová forma práce ve formátu L ^A T _E X
text.....	text práce
├─ thesis.pdf.....	text práce ve formátu PDF