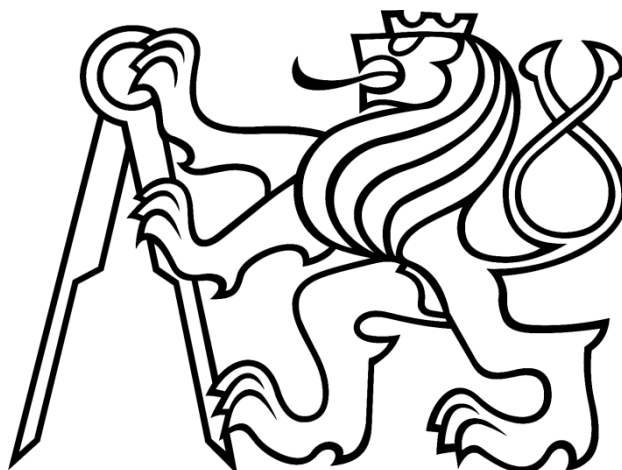


České vysoké učení technické v Praze
Fakulta Elektrotechnická



Bakalářská práce

Call centrum pro soukromou firmu

Filip Šamánek

Vedoucí práce: Ing. Martin Klíma, Ph.D.

Studijní program: Otevřená informatika (Bakalářský)

Obor: Informatika a počítačové vědy

Datum 22.5.201

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip Š a m á n e k
Studijní program: Otevřená informatika (bakalářský)
Obor: Informatika a počítačové vědy
Název tématu: Call centrum pro soukromou firmu

Pokyny pro vypracování:

Navrhněte a implementujte část funkcionality call centra pro soukromou firmu. Aplikaci navrhněte jako webovou nad technologií PHP/JavaScript.

Aplikace umožní:

- současnou práci několika operátorů,
- jejich kontrolu a možnost sledování plnění týdenního plánu,
- stahování a přípravu kontaktů pro obvolávání,
- export nasbíraných dat pro další zpracování,
- náhled do získaných dat zákazníků.

Specificky se zaměřte na tyto oblasti:

- frontování kontaktů,
- analýzu stavu hovoru,
- modul pro operátory,
- modul pro zákazníky,
- modul pro supervizory.

Práci koordinujte a spolupracujte s panem Martinem Dendisem, který pracuje na podobném tématu v rámci své bakalářské práce.

Seznam odborné literatury:

- [1] Aaron Saray - Professional PHP Design Patterns – srpen 2009 - ISBN: 0470496703.
[2] Baron Schwartz, Peter Zaitsev, Vadim Tkachenko – High Performance MySQL - březen 2012 - ISBN: 1449314287.

Vedoucí bakalářské práce: Ing. Martin Klíma, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Poděkování

Děkuji všem, kteří mě podporovali při tvorbě této práce, především vedoucímu práce, panu Ing. Martinu Klímovi, Ph.D. za poskytnuté rady. Také bych rád poděkoval mým rodičům a mé přítelkyni za podporu v průběhu celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

Podpis autora práce

Abstrakt

Tato bakalářská práce se zabývá vytvořením rozhraní pro uživatele call centra. Toto call centrum se zabývá obvoláváním zákazníků s cílem zjistit jejich spokojenost s poskytovanými službami autorizovaných partnerů nadnárodní společnosti. Aplikace je naprogramovaná v jazyce PHP za pomoci frameworku Nette a využívá databázi MySQL.

Klíčová slova

Call centrum, VoIP, Daktela, Nette, PHP, MySQL, Nette Tester, Webová aplikace

Abstract

This bachelor thesis deals with creating interface for call center users. This call center deals with research to determine customer's satisfaction with the services by authorized partners of multinational company. Application is programmed in PHP using Nette framework and using MySQL database.

Keywords

Call center, VoIP, Daktela, Nette, PHP, MySQL, Nette Tester, Web application

Obsah

1	Úvod.....	1
1.1	Cíle práce.....	1
2	Funkční požadavky	2
2.1	Uživatelské role	2
2.1.1	Operátor.....	3
2.1.2	Supervizor	4
2.1.3	Uživatel portálu.....	5
2.2	Obvolávání kontaktů a dostupnost dat.....	6
2.3	Správa uživatelů	6
2.4	Schéma zpracování dat	6
3	Nefunkční požadavky.....	11
3.1	Hardware	11
3.2	Software.....	11
3.3	Zálohování.....	12
3.4	Ostatní požadavky	12
4	Analýza.....	13
4.1	Analýza stavů hovoru	13
4.1.1	Rozdělení stavů	13
4.2	Technologie	14
4.2.1	PHP.....	15
4.2.2	MySQL.....	15
4.2.3	JavaScript	15
4.2.4	HTML.....	16
4.2.5	Daktela	16
4.2.6	Framework	16
4.2.7	Závěr.....	19
4.2.8	jQuery.....	19
4.2.9	DataTable	19
4.2.10	Databázová Vrstva	20
5	Návrh řešení	21
5.1	Návrh databáze	21
5.2	Zálohování a záložní server.....	23

5.3	Přihlašování	23
5.4	Práce s dotazníkem	23
5.5	Fronta kontaktů připravených k obvolání.....	23
5.5.1	Datová reprezentace fronty kontaktů	23
5.5.2	Frontování kontaktů	24
5.6	Hovory	25
5.6.1	Vytvoření hovoru	25
5.6.2	Zobrazení dotazníku	25
5.6.3	Uložení informací o hovoru	25
5.6.4	Nahrávání hovoru a uchovávání nahrávek	26
5.7	Kontrola operátorů.....	26
5.8	Výpis provedených hovorů.....	26
5.9	Statistiky	26
5.9.1	Online statistiky.....	27
5.9.2	Statistiky ke stažení	27
5.10	Soubory pro portál	27
5.11	Správa uživatelů	27
6	Implementace	28
6.1	Logické rozdělení aplikace	29
6.1.1	Presentery	29
6.1.2	Modely	31
6.1.3	Komponenty	32
7	Testování	33
7.1	Testování bez uživatele	33
7.1.1	Testování	33
7.1.2	Testování presenterů	33
7.2	Testování s uživatelem	33
8	Závěr.....	35
	Seznam obrázků	36
	Zdroje	37

1 Úvod

Tato bakalářská práce vznikla pro společnost FirmaA zprostředkovávající služby v oblasti marketingu, výzkumu a monitoringu trhu pro mezinárodní společnosti.

Mimo jiné zprostředkovává osobní rozhovory v podobě call center, které slouží pro výzkum trhu a spokojenosti zákazníků s produkty a službami poskytovanými těmito společnostmi.

Tato bakalářské práce se zabývá návrhem a implementací rozhraní call centra a bude použita pro obvolávání zákazníků s cílem zjistit jejich spokojenost se službami autorizovaných partnerů jednoho z jejich zákazníků (FirmaB).

Výsledky výzkumu budou předávány do společnosti FirmaB a do FirmaC, která vytvoří podrobné statistiky jednotlivým partnerům.

Tato bakalářské práce bude vycházet z funkčních požadavků, které byly obdrženy od FirmaA.

1.1 Cíle práce

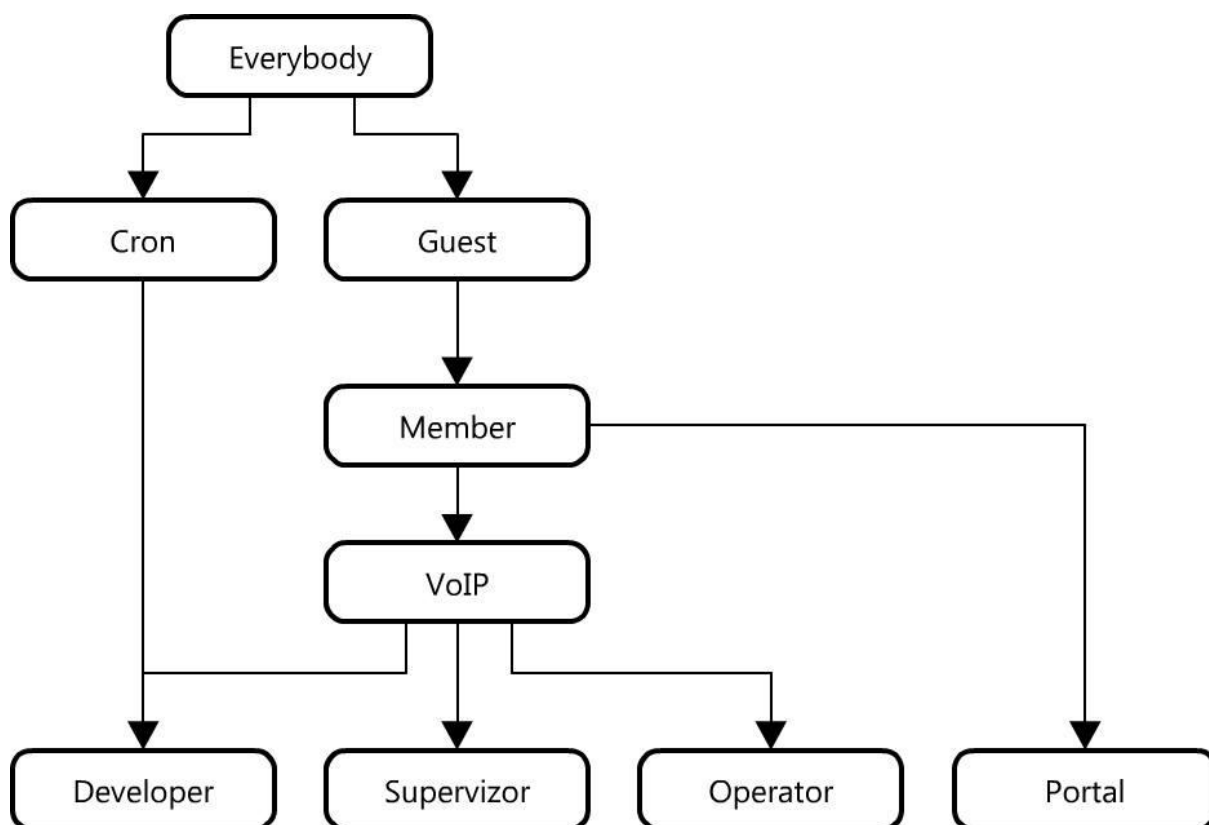
Cílem této práce je navrhnout a naimplementovat část aplikace call centra, které bude splňovat specifikované požadavky.

2 Funkční požadavky

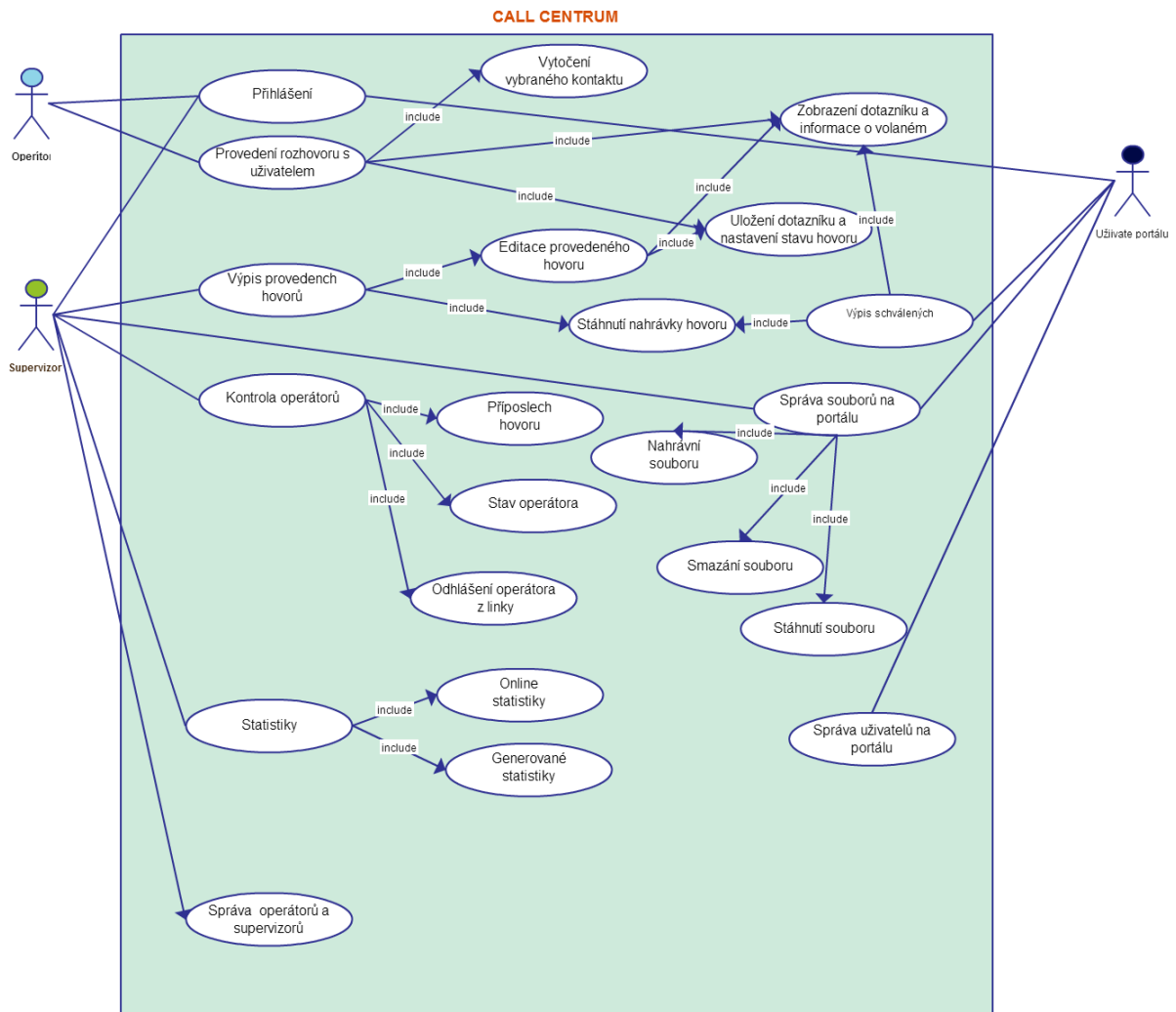
Nejprve budou popsány jednotlivé uživatelské role a jejich oprávnění. Dále budou specifikovány požadavky netýkající se uživatelských rolí a nakonec bude popsáno schéma zpracování dat.

2.1 Uživatelské role

K aplikaci budou přistupovat tři role uživatelů - operátor, supervizor a uživatel portálu. Jednotlivé role jsou znázorněny na Obrázek 1. Operátor má na starosti obvolávání kontaktů a vyplňování dotazníků s volaným zákazníkem. Supervizor dohlíží na práci operátorů, kontroluje statistiky a plnění plánu. Portál slouží pro přístup pracovníků FirmaB. Jednotlivé oprávnění jednotlivých rolí je znázorněno na Obrázek 2.



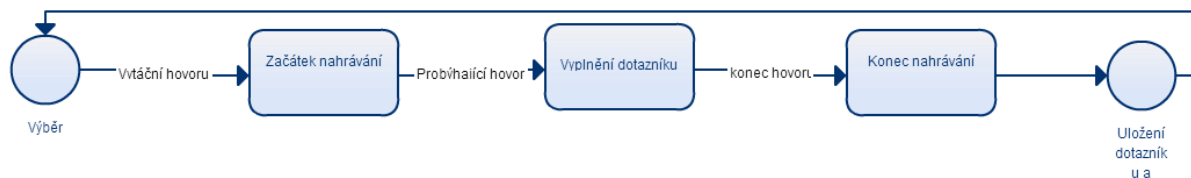
Obrázek 1. Role. Zdroj [11]



Obrázek 2. Znázornění oprávnění rolí uživatelů

2.1.1 Operátor

- Provádí osobní rozhovory s připravenými kontakty.
- Každý operátor má svoji vlastní frontu kontaktů, která je mu automaticky doplňována systémem.
- Operátor vybere jeden kontakt z vlastní fronty a zahájí s ním hovor.
- Po vytvoření hovoru je operátorovi zobrazen dotazník s informacemi o volaném klientovi.
- Celý hovor se zákazníkem je nahráván a ukládán pro pozdější kontrolu hovoru.
- Po vyplnění dotazníku zadá operátor stav, se kterým hovor skončil, eventuelně čas dohodnutého dalšího hovoru.



Obrázek 3. Schéma práce operátora

2.1.2 Supervizor

Supervizor má na starosti monitorování práce jednotlivých operátorů, ruční kontrolování vyplněných dotazníků, kontrolování stavu obvolávání aktuální vlny, kontrolu statistik a nahrávání souborů na portál.

2.1.2.1 Statistika

- Globální statistiky uvádí, kolik je připravených kontaktů k obvolání a s jakými stavy skončily provedené hovory.
- Podrobné statistiky uvádějí, kolik kontaktů je obvoláno a v jakém stavu hovory skončily. Dále uvádějí přehled plnění kvót¹ pro jednotlivé partnery (servis nebo prodejce). Podrobné statistiky jsou rozdělené podle zdroje, z jakého pocházejí kontakty (prodej/servis).
- Automaticky generované reporty call centra. Tyto reporty zpracovává ve své bakalářské práci Martin Dendis.

2.1.2.2 Monitoring operátorů

- Supervizor má možnost příposlechu aktuálně probíhajících hovorů, bez možnosti do hovoru jakkoliv zasáhnout.
- Má možnost zobrazení stavu operátorů a v případě potřeby možnost odhlásit operátora z ústředny.
- Má také možnost editace operátorů. (možnost změnit jméno, příjmení a heslo operátora).
- Provádí ruční kontrolu všech provedených hovorů a jejich dotazníků. Supervizor k tomu má k dispozici tyto prostředky:
 - Výpis všech provedených hovorů.
 - Informace o stavu, délce a času začátku hovoru, o tom s kým byl hovor prováděn a kdo hovor prováděl.

¹ Kvóta – stanovený počet obvolávaných kontaktů pro každý servis

- Možnost zobrazit a editovat vyplněný dotazník.
 - Audio nahrávku hovoru.
- Zkontrolované hovory má možnost schválit pro zobrazení na portálu nebo odebrat z generovaných reportů.

2.1.2.3 Nahrávání souborů na portál

- Automaticky generované reporty (eventuelně i jiné soubory) může nahrát ke stažení na portál.
- Při nahrání souboru může zadat emailovou adresu, na kterou přijde upozornění o nahraném souboru.

2.1.3 Uživatel portálu

Slouží pro přístup pracovníků FirmaB a jejich partnerů. Tito uživatelé si mohou projít schválené hovory a soubory nahrané na portál.

2.1.3.1 Schválené hovory

- Může projít dokončené hovory, které jsou schválené pro zobrazení na portálu.
- U všech hovorů má k dispozici informace o tom, kdy a s jakým zákazníkem byl hovor proveden, a má možnost zobrazit si dotazník k danému hovoru.

2.1.3.2 Soubory na portálu

- Má přístup k souborům nahraných na portál.
- Soubory může přidávat nebo je mazat.
- Při nahrání souboru může stejně jako supervizor zadat emailovou adresu, na kterou přijde upozornění o nahraném souboru.

2.1.3.3 Správa uživatelů

- Každý si může změnit své uživatelské údaje (jméno a heslo).
- Uživatel s rozšířeným oprávněním může editovat ostatní uživatele portálu (měnit oprávnění, jméno a heslo uživatele).

2.2 Obvolávání kontaktů a dostupnost dat

- Přednostně jsou obvolávány kontakty pocházející z prodeje. U kontaktů ze servisu bude obvoláno jen tolik kontaktů, aby byly splněny kvóty. Kvóty jsou určeny pro každý servis zvlášť. Nedovolané kvóty se přenášejí do další vlny.
- Všechny provedené hovory budou nahrávány. Nahrávky budou dostupné minimálně půl roku online přes webové rozhraní call centra a dva roky na vyžádání do pěti pracovních dnů.

2.3 Správa uživatelů

- Přidávání a odebrání uživatelů bude mít na starosti administrátor aplikace, protože tyto úkony souvisí se založením účtu a jeho nastavením v ústředně.
- Není potřeba grafického rozhraní.

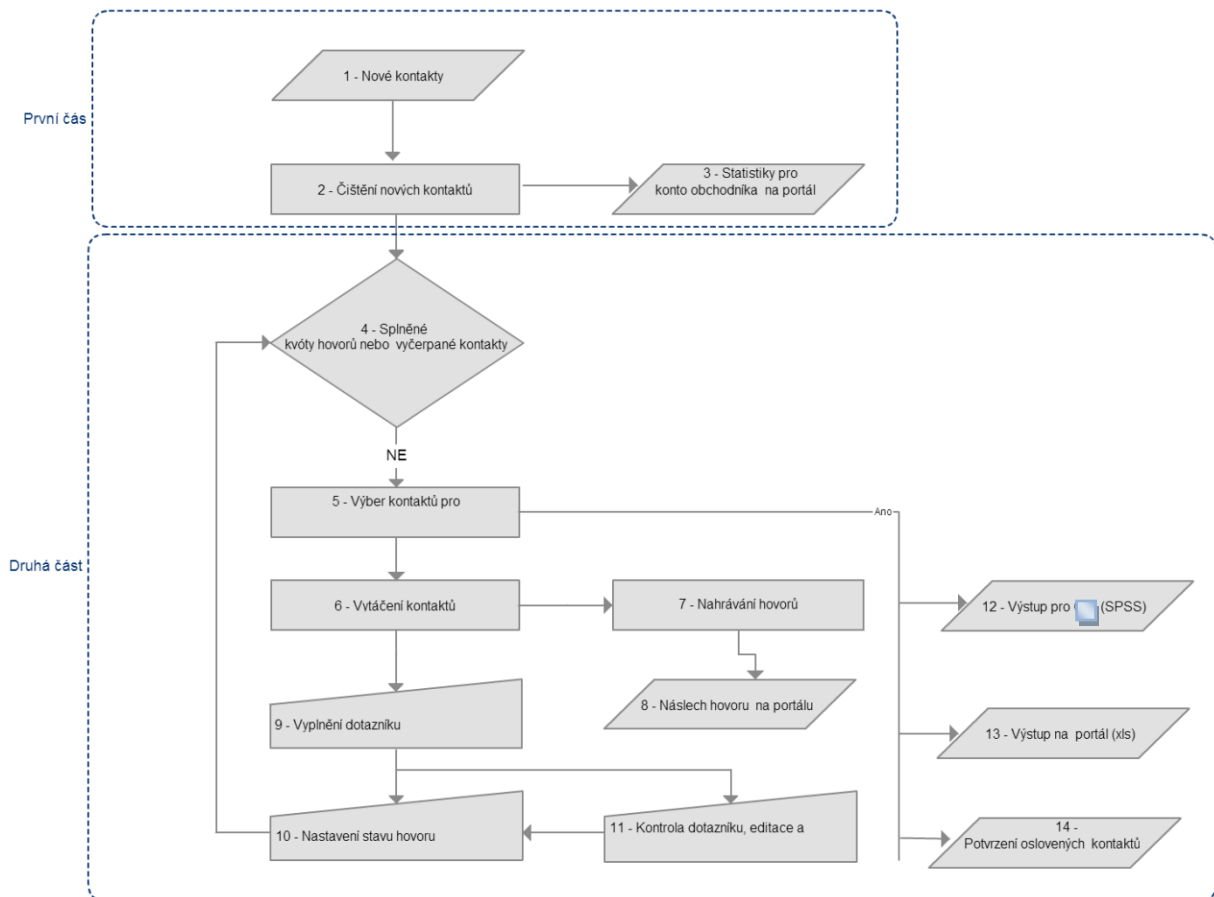
2.4 Schéma zpracování dat

Následující schéma popisuje proces zpracování získaných dat v rámci jedné vlny². Záměrně je zde vynechána část, která popisuje získávání kontaktů pro obvolávání, jelikož není realizována pomocí této aplikace.

Schéma je rozděleno do dvou částí - příprava a obvolávání kontaktů. Každá část trvá jeden týden. Procesy přípravy a obvolávání kontaktů probíhají paralelně, tzn. když je obvolávána jedna vlna, paralelně jsou čištěny kontakty z další vlny.

Proces zpracování dat popisuje Obrázek 4. Jednotlivé body znázorněné na obrázku jsou dále podrobněji popsány.

² Vlna – jeden cyklus obvolávání kontaktů



Obrázek 4. Schéma zpracování dat

1) *Nové kontakty*

- Jedná se o kontakty zákazníků, kteří navštívili servis nebo zakoupili nové auto u dealera ve sledovaném období. V každém období přibude maximálně 10 000 kontaktů ze servisu a 1 000 kontaktů z prodeje.
- Kontakty budou rozděleny podle původu (servis / prodej) a podle konkrétního partnera, kterého navštívily.
- Software call centra si bude aktivně stahovat nové kontakty.
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

2) *Čištění nových kontaktů*

- Ze stažených kontaktů je potřeba odstranit duplicitní kontakty a kontakty s neúplnými nebo neplatnými informacemi.
- Čištění bude probíhat automaticky po stažení kontaktů.
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

3) *Statistiky pro konto obchodníka na portál*

- Po vyčištění kontaktů se generují statistiky počtu přijatých a odstraněných kontaktů a důvodu jejich odstranění. Tyto statistiky budou dostupné na portále ke stažení.
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

4) *Splněné kvóty hovorů nebo vyčerpané kontakty*

- Pokud nejsou splněny kvóty, pokračuje obvolávání kontaktů. Nominované kontakty jsou dále doplňovány (pokud existují kontakty pro doplnění), aby bylo možné kvóty dovolat.

5) *Výběr kontaktů pro obvolání*

- Z načtených a vyčištěných kontaktů je potřeba vybrat kontakty pro obvolávání podle specifických pravidel tak, aby byly splněny kvóty počtu telefonátů pro kontakty ze servisů.
- *Pravidla pro výběr kontaktů pro obvolávání:*
 - Kontakty z prodeje se obvolávají všechny.
 - Kontakty ze servisů se seskupí podle servisu a obvolává se pouze tolik kontaktů, kolik je potřeba ke splnění kvóty daného servisu. Počítají se pouze dokončené hovory.
 - Každý kontakt může mít maximálně 3 pokusy o dovolání.
 - Jednotlivé pokusy o dovolání se jednomu kontaktu musí mít mezi sebou pauzu minimálně 24 hodin. Výjimkou je, pokud se operátor dohodne se zákazníkem na konkrétním čase rozhovoru.
 - Pokud se v dané vlně neobvolá u některého servisu kvóta, je v další vlně navýšena o počet nedovolaných kontaktů.
- *Proces nominace kontaktů:*
 - Na začátku týdne se nominují všechny kontakty z prodeje a kontakty ze servisů se seskupí podle jednotlivých servisů. Z každé skupiny kontaktů ze servisů je následně nominován takový počet kontaktů, aby byla splněna kvóta pro daný servis.
 - Postupně bude docházet k doplňování nominovaných kontaktů. Kontakty z prodeje se budou nominovat automaticky zpět, pokud uplyne 24 hodin od posledního pokusu o obvolání. Kontakty ze servisů se budou doplňovat tak, aby byla splněna kvóta pro daný servis a aby nebyl nominovaný kontakt, který byl obvoláván v

posledních 24 hodinách. Přednost mají kontakty s nejnižším počtem pokusů o obvolání.

6) Vytáčení kontaktů

- Nominované kontakty jsou obvolávány paralelně až dvaceti operátory. Každý operátor má svoji malou frontu kontaktů.
- Kontakty jsou obvolávány podle priorit (od nejvyšší po nejnižší):
 - Kontakty s domluveným časem rozhovoru
 - Kontakty z prodeje
 - Kontakty s nejmenším počtem pokusů o provedení hovoru
- Pokud má více kontaktů stejnou prioritu, jsou obvolávány náhodně.
- Operátor ručně vybere kontakt z vlastní fronty, který bude vytočen.

7) Nahrávání hovorů

- Všechny hovory budou nahrávány a ukládány pro pozdější poslech.

8) Náslech hovoru na portálu

- Schválené hovory budou dostupné k náslechu na portále.

9) Vyplnění dotazníku

- Operátor během telefonického hovoru vyplní dotazník. Při vyplňování bude mít k dispozici potřebné informace o volaném zákazníkovi.
- Dotazníky budou dostupné v několika verzích podle toho, zda jde o kontakt z prodeje, ze servisu nebo ze servisu se servisním poradcem.

10) Kontrola a editace dotazníku, náslech hovoru

- Supervizor call centra bude mít možnost zkontrolovat a opravit nebo doplnit vyplněný dotazník. V případě chyby či nejasností má možnost si přehrát nahrávku hovoru.

11) Nastavení stavu hovoru

- Před uložením hovoru musí nastavit operátor stav, s jakým skončil hovor. Podle zvoleného stavu bude poté kontakt zpracován - přeplánován na jiný čas, vyřazen, zařazen mezi dokončené, atp. (viz Analýza stavů hovoru).

12) Výstup pro firmu zpracovávající data (SPSS)

- Po ukončení obvolávání jedné vlny se vygeneruje a odešle report ve formátu SPSS³
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

13) Výstup na portál (xls)

- Po ukončení obvolávání jedné vlny se vygeneruje report také ve formátu xls, který bude moci supervizor zkontrolovat a následně nahrát na portál.
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

14) Potvrzení oslovených kontaktů

- Po ukončení obvolávání jedné vlny se dále odešle potvrzení o obvolaných kontaktech, čímž bude zajištěno, že jednou kontaktovaní zákazníci nebudou zařazeni do dalších probíhajících průzkumů.
- Tuto část má podrobněji zpracovanou ve své práci kolega Martin Dendis.

³ SPSS – Datový formát od IBM

3 Nefunkční požadavky

Většina nefunkčních požadavků se vztahuje k nárokům na hardware, software, zálohování a obnovu chodu při výpadku callcentra.

3.1 Hardware

Hardware a jeho konfigurace bude zajištěna externí firmou. V návrhu aplikace je potřeba počítat s maximálním dostupným výkonem.

- K dispozici budou dva servery (hlavní a záložní) umístěné v budově callcentra a dvacet stanic pro operátory.
- *Minimální požadavky na server*
 - CPU: Intel/AMD, 4 jádra, frekvence 2.0 GHz, cache 4MB
 - RAM: DDR3, 8GB
 - 1x LAN
 - OS: Linux
- *Minimální požadavky na stanici pro operátory*
 - All-in-one PC
 - CPU: Intel/AMD, 2 jádra, frekvence 2.0 GHz, cache 2MB
 - RAM: DDR3, 2GB
 - 1x LAN
 - OS: Linux
 - Příslušenství (klávesnice, myš, sluchátka s mikrofonem)

3.2 Software

- Call centrum bude implementováno v jazyce PHP⁴.
- Jednotlivé moduly, např. pro frontování kontaktů a generování reportů, mohou být implementovány i v jiných jazycích např. C/C++⁵, Java⁶ atd.
- Pro účely callcentra bude použita databáze MySQL⁷.
- Aplikace poběží na některé Linuxové distribuci.

⁴ PHP – Hypertext Preprocessor, skriptovací jazyk interpretovaný na straně serveru

⁵ C/C++ – nízko úroňový, kompilovaný programovací jazyk

⁶ Java – objektivě orientovaný programovací jazyk, kompilovaný do bitecodu

⁷ MySQL – SQL

3.3 Zálohování

- Jednou denně bude probíhat kompletní záloha aplikace (databáze a souborový systém).
- V případě výpadku serveru bude k dispozici záložní server, který bude dostupný do dvou hodin. Databáze z hlavního serveru bude online replikovaná na záložní server a souborový systém bude synchronizovaný jednou denně.

3.4 Ostatní požadavky

- Aplikace bude spolupracovat s virtuální ústřednou Daktela.
- Aplikace bude přístupná přes webové rozhraní.
- Aplikace bude umožňovat práci až 20 operátorů současně.
- Každý týden přibude až 10 000 kontaktů ze servisu a 1 000 kontaktů z prodeje.
- Kontakty a záznamy o hovorech se budou uchovávat minimálně dva roky.

4 Analýza

4.1 Analýza stavů hovoru

Pro správné zpracování provedeného hovoru je potřeba přiřadit hovoru stav, podle kterého bude hovor následně zpracován. Nejjednodušší stavy mohou být například, *nebyl zvednut telefon* nebo *telefon byl obsazen* a podobně.

V praxi je potřeba rozlišovat stavy podrobněji a zahrnout do nich i výsledek rozhovoru se zákazníkem, například že byl vyplněn celý dotazník nebo si zákazník objednal zboží. Podle stavu s jakým hovor skončil je kontakt zařazen k dalšímu zpracování, například potvrzení objednávky, dokončení přerušovaného hovoru nebo také vyřazení kontaktu z databáze, pokud si zákazník nepřeje být dál kontaktován.

4.1.1 Rozdělení stavů

V této analýze stavů se budeme zabývat pouze stavy, které jsou relevantní pro tuto aplikaci (jedná se o aplikaci na obvolávání zákazníků s cílem zjistit jejich spokojenost se službami partnerů zákazníka callcentra). V jiných call centrech můžeme mít stavy týkající se například dokončení objednávky.

Stavy jsou nejprve rozděleny do tří základních kategorií a mají přidělenou sadu číselných kódů. Jedná se o stav *úspěšně dokončeno*, *přeplánování hovoru* a *stav vyřazení kontaktu z připravených kontaktů*. Číselné kódy byly přidělovány s ohledem na budoucí rozšíření nebo modifikaci pro jiné zákazníky.

4.1.1.1 Úspěšně dokončeno

Stavům *úspěšně dokončeno* jsou přidělené číselné kódy od 1 do 999. Kontakty, u kterých hovor skončí tímto stavem, už nebudou dále obvolávány a označí se jako splněné. V této aplikaci se jedná pouze o jeden možný stav a to, že byl dotazník úspěšně dokončen s kódem 1.

4.1.1.2 Přeplánování

Stavům *přeplánování* jsou přidělené číselné kódy od 1001 do 1999. Kontakty, u kterých hovor skončil tímto stavem, budou zařazeny znovu k obvolání. Stavy jsou rozdělené podle důvodu jejich přeplánování a následující akce.

- 1001 až 1009 jsou stavy přeplánování z důvodu dohody s volaným na jiný čas

- 1001 – Přepínat se stejným operátorem
- 1002 – Přepínat s jakýmkoliv operátorem
- *1011 až 1019 je přepínání z důvodu nezvednutého volaného telefonu*
 - 1011 – Volat okamžitě znovu (se stejným operátorem)
 - 1012 – Volat v dalším kole (s jakýmkoliv operátorem)
- *1021 až 1029 jsou stavy, kdy je obsazený telefon*
 - 1021 – Volat okamžitě znovu (se stejným operátorem)
 - 1022 – Volat v dalším kole (s jakýmkoliv operátorem)
- *1031 až 1039 jsou stavy, kdy se hovor rozpojil*
 - 1031 – Přepínat hovor na konkrétní čas (se stejným operátorem)
 - 1032 – Přepínat hovor na konkrétní čas (s jakýmkoliv operátorem)
 - 1033 – Volat okamžitě znovu (se stejným operátorem)
 - 1034 – Volat v dalším kole (s jakýmkoliv operátorem)

4.1.1.3 Vyřazení hovoru

Stavy *vyřazení hovoru* mají přidělené kódy od 2001 do 2999. Jedná se o stavy hovoru, kdy bude kontakt vyřazen jako nesplněný. Stavy jsou rozdělené podle důvodu vyřazení.

- *2001 až 2009 zákazník si nepřeje být kontaktován*
 - 2001 – Zákazník si už nepřeje být nikdy kontaktován (bude zařazen mezi Robinsony)
 - 2002 – Zákazník si nepřeje vyplnit tento dotazník
 - 2003 – Příliš mnoho pokusů o dovolání (zákazník nezvedá telefon)
- *2011 až 2019 špatný kontakt*
 - 2011 – Telefonní číslo neexistuje
 - 2012 – Omyl (telefonní číslo existuje, ale nejedná se o uvedeného zákazníka)
 - 2013 – Zákazník nebyl v servisu ani si nekoupil nové auto
 - 2014 – Zaměstnanec servisu/prodejny

4.2 Technologie

Výběr technologií probíhal na základě nefunkčních požadavků (aplikace poběží na PHP a MySQL), předchozích zkušeností z předchozích projektů a srovnání s konkurenčními technologiemi.

4.2.1 PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk interpretovaný na straně serveru. Je určený především pro tvorbu dynamických webových aplikací. Umožňuje jak objektový, tak procedurální přístup k psaní programu.

PHP bylo zvoleno na základě nefunkčního požadavku.

4.2.2 MySQL

MySQL je rozšířená multiplatformní relační databáze, s kterou komunikace probíhá pomocí dotazovacího jazyka SQL.

MySQL bylo zvoleno na základě funkčních požadavků.

4.2.3 JavaScript

JavaScript je objektově orientovaný skriptovací jazyk interpretovaný v prohlížeči na straně klienta. Slouží primárně pro tvorbu dynamických webových aplikací, kde umožňuje zpracovávat události na straně klienta bez znovunačtení stránky, manipulaci s HTML objekty a komunikaci se serverem na pozadí.

4.2.3.1 Ajax

Ajax (Asynchronous JavaScript and XML⁸) je JavaScriptová technologie pro asynchronní komunikaci se serverem bez nutnosti načtení celého obsahu stránky. Komunikace se serverem probíhá na pozadí stránky bez uživatelského vědomí. Data ze serveru jsou vracena nejčastěji ve formátu XML (jak název napovídá) nebo JSON⁹. Tyto formáty nejsou však podmínkou pro použití této technologie, záleží pouze na programátorovi.

4.2.3.2 Alternativy

Jako alternativy k JavaScriptu byly zvažovány technologie Flash, Silverlight a Java. Tyto alternativy byly zamítnuty, jelikož vyžadují pro svůj běh nainstalované doplňky do prohlížeče a v současné době jsou na ústupu.

⁸ XML - Extensible Markup Language, obecný značkovací jazyk

⁹ JSON - JavaScript Object Notation, datový formát

4.2.4 HTML

HTML (HyperText Markup Language) je značkovací jazyk pro tvorbu webových stránek a aplikací.

4.2.5 Daktela

Virtuální telefonní ústředna. Komunikace probíhá pomocí API¹⁰ přes http¹¹ protokol. Pomocí tohoto API lze pracovat s hovory, monitorovat operátory a přihlašovat uživatele do telefonní ústředny. API vrací data ve formátu JSON nebo XML.

Kompletní dokumentaci lze najít na adrese <http://daktela.com/api/api.html>

Virtuální ústředna daktela byla zvolena z důvodu nefunkčních požadavků na její využití.

4.2.6 Framework¹²

Pro usnadnění vývoje, zbavení se řešení základních operací (routování, autoloading, validace formulářů, atd.) a možnosti soustředit se na programování samotné funkcionality callcentra bylo rozhodnuto o použití některého php frameworku. Níže jsou uvedené zvažované alternativy.

4.2.6.1 Nette framework

Nette framework je objektově orientovaný open source¹³ framework pro tvorbu aplikací v PHP 5. Nette je nabízený pod licencemi GNU GPL¹⁴ a licencí Nette. Původním autorem tohoto frameworku je David Grudl.

Vlastnosti:

- Šablonovací systém Latte
- Ladící nástroj Laděnka, který v ladícím módu vypisuje chyby graficky viz Obrázek 2. V produkčním módu vygeneruje log, což je HTML soubor ve stejném formátu.

¹⁰ API - Application Programming Interface, rozhraní pro komunikaci s aplikací

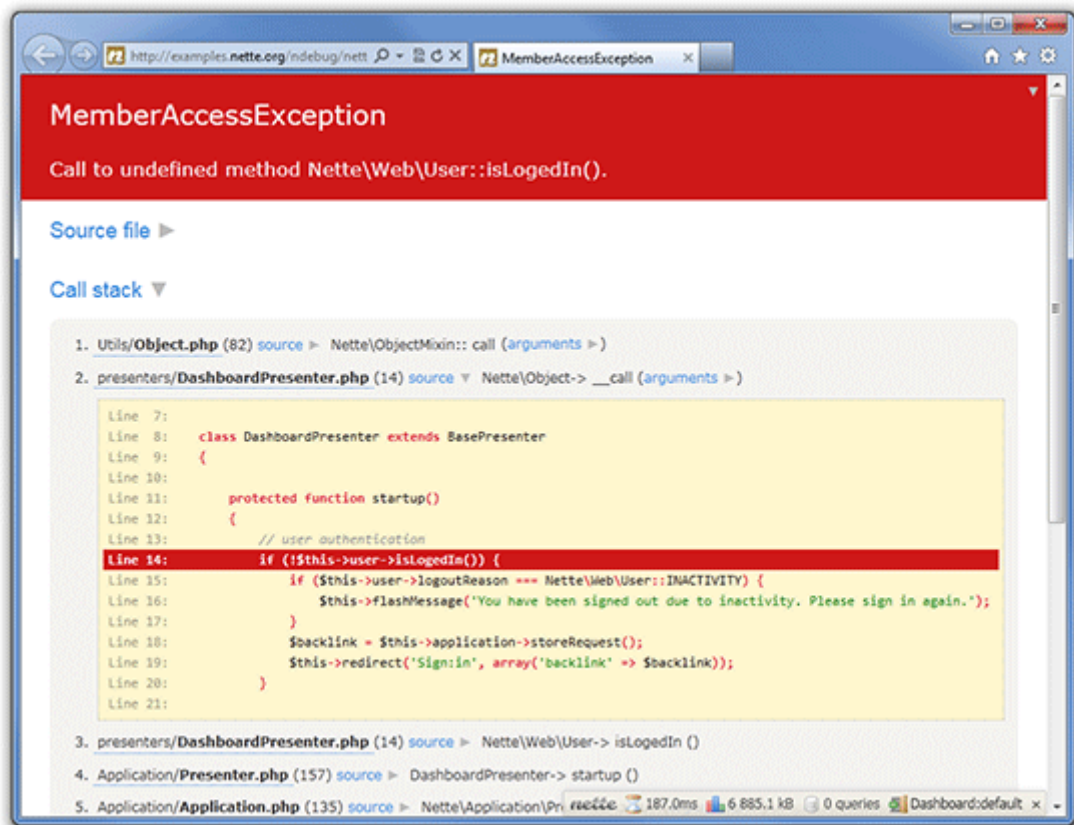
¹¹ http - Hypertext Transfer Protocol, internetový protokol

¹² Framework – softwarová struktura, která slouží jako podpora při vývoji softwarových projektů

¹³ Open source – svobodný software

¹⁴ GNU GPL- GNU General Public License, GNU GPL = všeobecná veřejná licence GNU

- Rozšíření jazyka PHP pomocí třídy Nette\Object.
- MVC (model view controler) architektura aplikací. Přesněji její novější podoba MVP (model view presenter).
- Eliminuje většinu nejčastějších bezpečnostních děr, jako jsou XSS¹⁵, CSRF¹⁶
- Velká česká komunita



Obrázek 5. Nette Framework, chybová hláška. Zdroj [3]

4.2.6.2 Symfony 2

Symfony 2 je objektově orientovaný open source framework pro tvorbu aplikací v PHP 5. Symfony 2 je nabízený pod licenci BSD¹⁷ License. Autorem tohoto frameworku je společnost Yii Software LLC.

¹⁵ XSS – Cross Site Scripting, metoda útoku na webové aplikace

¹⁶ CSRF - Cross-site request foery, metoda útoku na webové aplikace

¹⁷ BSD – licence pro svobodný software

Vlastnosti

- Velká podpora komunity programátorů
- Využívá objektové programování
- Využívá velké množství nástrojů
- Možnost rozšíření
- Vysoká stabilita

4.2.6.3 *Yii framework*

Yii je objektově orientovaný open source framework pro tvorbu aplikací v PHP 5. Yii je nabízený pod licencí BSD License. Autorem tohoto frameworku je společnost Yii Software LLC.

Vlastnosti

- Abstraktní a objektový přístup k databázi
- Validace uživatelských vstupů
- ACL¹⁸
- i18n a l10n¹⁹
- Cache vrstva²⁰
- Podpora komponent jQueryUI
- Podpora zhlédů
- Zcaffolding²¹ (abstraktní popis databáze) pro automatické generování kódu
- Velká mezinárodní komunita

¹⁸ ACL - access control list, práce s oprávněními uživatelů

¹⁹ I18n a l10n - lokalizace (překlady stránek)

²⁰ Cache – mezipaměť pro zrychlení aplikace

²¹ Zcaffolding - abstraktní popis databáze

Zend framework

Zend framework je objektově orientovaný open source framework pro tvorbu aplikací v PHP 5. Zend je nabízený pod licenci BSD License. Autorem tohoto frameworku je společnost Zend Technologies Ltd.

Vlastnosti

- Podporuje databáze MySQL, MSSQL, SQLite, Oracle, PostgreSQL, IBM DB2.
- Součástí je mnoho modulů, např. ověřování práv (Zend_Acl), posílání emailů (Zend_Mail).
- Pohledy můžou být klasické soubory tvořeny kombinací HTML a PHP mající příponu phtml, nebo lze využít i šablonovací systém SMARTY.
- Obsahuje nadstandardní validační metody, jako např. ověření, zda jde o: IP adresu, hexadecimální číslo nebo datum. Nabízí také možnost zadání regulárního výrazu.
- MVC (Model View Controller) architektura.
- Velká mezinárodní komunita

4.2.7 Závěr

Pro tvorbu této aplikace byl zvolen Framework Nette, a to hlavně z důvodu MVP architektury, dobrého zabezpečení, velké české komunity a dobré zkušenosti z předchozích projektů.

4.2.8 jQuery

jQuery je JavaScriptová knihovna umožňující manipulaci s HTML objekty, zpracování událostí, animace a Ajax. Její výhodou je jednoduché API, které funguje napříč všemi prohlížeči.

4.2.9 DataTable

DataTable je plug-in pro jQuery knihovnu. Umožňuje jednoduché vytváření datových tabulek a práci s nimi (vyhledávání, řazení podle sloupců, stránkování, atd.).

4.2.10 Databázová Vrstva

Výběr vhodné databázové vrstvy zpracovává ve své bakalářské práci Martin Dendis.

4.2.10.1 Dibi

Dibi je databázová vrstva napsaná v PHP, která se snaží zjednodušit zápis SQL příkazů a ulehčit rutiny, se kterými se programátor běžně setkává. Podporuje 8 typů databází – MySQL, MySQLi, PostgreSQL, SQLite, ODBC a experimentálně MS SQL, Oracle a PDO.

5 Návrh řešení

Aplikace bude postavená na technologiích PHP, MySQL, HTML, JavaScript a bude napsaná ve frameworku Nette. Pro komunikaci s databází bude použita knihovna Dibi. Tato kombinace byla zvolená tak, aby se jednalo o webovou aplikaci, aby splňovala nároky na výkon a velikost zpracovaných dat a na základě dobrých zkušeností z dřívějších projektů.

5.1 Návrh databáze

Návrh databáze je znázorněn na Obrázek 6.

5.2 Zálohování a záložní server

Zálohování a správu záložního serveru zajišťuje správce serveru. Aplikace se o tuto část nemusí starat.

5.3 Přihlašování

Do aplikace mají přístup pouze přihlášení uživatelé. Proces přihlašování a ověřování uživatele zpracoval podrobněji kolega Martin Dendis ve své bakalářské práci.

5.4 Práce s dotazníkem

O vykreslení dotazníku a jeho uložení se stará komponenta od kolegy Martina Dendise, které je předán typ dotazníku, informace o volaném operátorovi, callback na uložení dotazníku a volitelně id dotazníku (pokud se má vykreslit už existující dotazník).

Při ukládání dotazníku je zavolána zvolená metoda, která předá komponentě data z dotazníku. V této metodě je získán z dotazníku stav, s jakým byl hovor dokončen a id dotazníku. Dále může být v této metodě zrušeno ukládání dotazníku, pokud na to nemá uživatel oprávnění.

5.5 Fronta kontaktů připravených k obvolání

Aby nedocházelo k vytočení jednoho kontaktu více operátory současně, má každý z operátorů svoji vlastní malou frontu o pěti kontaktech a kontaktech, u kterých má naplánovaný konkrétní čas volání.

5.5.1 Datová reprezentace fronty kontaktů

Fronta bude řešena dvěma tabulkami - tabulka připravené kontakty (*contacts_ready*) a kontakty ve frontě (*contacts_queue*).

Tabulka připravené kontakty obsahuje všechny kontakty připravené k volání a informaci o čase posledního volání a stavu kontaktu:

- Ve frontě (*in_queue*)
- Nepoužitý (*unused*)
- Dokončený (*done*)

- Vyřazený (*termination*)
- Stará vlna (*old_wave*)

Z tabulky připravených kontaktů bude do tabulky fronty přesunuto jen tolik kontaktů, aby byly splněny kvóty. Tabulka fronty obsahuje informaci o přiřazeném operátorovi a čase naplánovaného volání kontaktu.

5.5.2 Frontování kontaktů

Frontování kontaktů má na starosti přesun kontaktů k z tabulky *contacts_ready* do tabulky *contacts_queue* a jejich přiřazení k operátorovi.

5.5.2.1 Přesun kontaktů do fronty

Přesun kontaktů z *contacts_ready* do *contacts_queue* je potřeba rozdělit do dvou částí, a to přesun kontaktů ze servisu a přesun kontaktů z prodeje.

Kontakty z prodeje, které nebyly volány v posledních 24 hodinách, budou přesunuty všechny.

U přesunu kontaktů ze servisu je potřeba nejprve vypočítat, kolik a jakých kontaktů se má přesunout, aby byly splněny kvóty. Výpočet se provádí tak, že se z tabulky *owners_quota* zjistí, kolik je potřeba obvolat kontaktů pro každý servis, a od toho se odečte, kolik kontaktů z daného servisu je připraveno v tabulce *contacts_queue*. Výsledek vyjadřuje, kolik kontaktů u každého servisu je potřeba doplnit z tabulky *contacts_ready*. Pokud v tabulce *contacts_ready* je na výběr více kontaktů než je potřeba (berou se v potaz pouze kontakty, které nebyly kontaktovány déle než 24 hodin), proběhne výběr kontaktů náhodně.

5.5.2.2 Přiřazení kontaktů k operátorovi

Kontakty jsou do fronty operátora přiřazovány na základě jejich priorit (viz Schéma zpracování datcallcentru). Každému operátorovi se přiřadí tolik kontaktů, aby měl ve frontě pět kontaktů bez naplánovaného času volání.

Při odhlášení operátora budou kontakty, které nemají naplánovaný čas odebrány operátorovi a přiřazeny jinému. Pokud má s kontaktem domluvený čas volání a není přihlášen do callcentra 5 minut po plánovaném začátku hovoru, bude kontakt přeřazen jinému operátorovi.

5.5.2.3 Spouštění frontování

Jelikož php nemá prostředky pro permanentní běh skriptu, je zde využit cron, který bude volat tuto část aplikace v intervalu jedné minuty. Script může běžet maximálně půl minuty (zajištěno nastavením v PHP), nemůže proto dojít k jeho spuštění dvakrát najednou.

Využitím cronu je zajištěno, že nedojde ke konfliktům a přiřazení jednoho kontaktu více operátorům. Je ale potřeba ošetřit, aby při předčasném ukončení skriptu zůstala zachována konzistence dat. Proto bude přenos každého kontaktu uzavřen do transakce. Uzavřít celou operaci frontování do transakce by nebylo vhodné, protože pokud by skript nestihl doběhnout, do fronty by se nepřesunul žádný kontakt. Při tomto uzavření operací do transakcí se sice může přesunout jen část potřebných kontaktů, ale zbytek se přesune při dalším spuštění skriptu.

5.6 Hovory

5.6.1 Vytvoření hovoru

Pro vytvoření hovoru bude poslán do ústředny požadavek k vytvoření hovoru. Id hovoru, které bude vráceno ústřednou, bude uloženo do databáze (tabulka *calls*) spolu se záznamem o začátku hovoru.

5.6.2 Zobrazení dotazníku

Dotazník obsahuje kromě dotazníku samotného také informace o volaném. Typ dotazníku se zvolí podle toho, z jakého zdroje pochází kontakt (servis/prodej). Pokud byl už s daným kontaktem proveden hovor, bude načten uložený dotazník. Data o provedených hovorech se budou získávat z tabulky *calls*.

5.6.3 Uložení informací o hovoru

Po dokončení hovoru bude k uloženému záznamu o hovoru (v tabulce *calls*) přidán uložený dotazník. Kontakt bude na základě výsledku (stavu) hovoru:

- **Volán v dalším kole.** Kontakt bude vyřazen z fronty (odebrán z tabulky *contacts_queue* a v tabulce *contacts_ready* bude přidán kontaktu čas volání a stav se změní na *unused*).

- **Přeplánován na jiný čas/volán okamžitě znovu.** U kontaktu se v tabulce *contacts_queue* doplní čas plánovaného volání (pokud má být volán okamžitě znovu, bude uveden aktuální čas). Pokud je zvoleno s jakýmkoliv operátorem, odebere se přiřazený operátor od daného kontaktu.
- **Dokončení hovoru.** Kontakt bude odebrán z tabulky *contacts_queue* a v tabulce *contact_ready* bude změněn stav hovoru na *done*.
- **Vyřazení kontaktu.** Kontakt bude odebrán z tabulky *contacts_queue* a v tabulce *contact_ready* bude změněn stav hovoru na *termination*.

5.6.4 Nahrávání hovoru a uchovávání nahrávek

Nahrávání hovorů a dostupnost nahrávek bude mít na starosti virtuální ústředna Daktela, se kterou bude aplikace komunikovat pomocí jejího API.

5.7 Kontrola operátorů

Aktuální činnost operátora bude získávána z ústředny pomocí jejího API. Příposlech nebo odhlášení operátora bude také řešit ústředna a aplikace pouze zašle pomocí jejího API požadavek o provedení dané akce.

5.8 Výpis provedených hovorů

Jedná se o výpis dat z tabulky *calls* s informací o operátorovi, který hovor provedl, a informací o kontaktu. Stáhnutí nahrávky je řešeno pomocí API dotazu na ústřednu.

Dotazníky jednotlivých hovorů je možné zobrazit, a pokud má uživatel dostatečná oprávnění (role *supervisor*) tak i smazat ze statistik nebo chválit pro zobrazení na portálu (tyto akce budou řešeny pomocí dvou sloupců v tabulce *calls*).

5.9 Statistiky

Statistiky jsou rozdělené do dvou částí. Online statistiky, které se generují vždy pro aktuální vlnu a nelze je stáhnout, a statistiky ke stažení, které jsou automaticky generované.

5.9.1 Online statistiky

Online statistiky se počítají pro aktuální vlnu. Počítají se z tabulek *contact_new*, *calls*, *contacts_ready*, *owners_quota* a *contacts_done*.

Z tabulky *contact_new* se počítá, kolik kontaktů bylo pro danou vlnu staženo a kolik jich neprošlo filtrováním.

Z ostatních tabulek se počítá počet provedených hovorů, v jakém stavu v jakém skončily a podrobné statistiky kvót pro jednotlivé servisy.

5.9.2 Statistiky ke stažení

Ty jsou generovány automaticky a jejich generování má na starosti část aplikace od kolegy Martina Dendise.

5.10 Soubory pro portál

Výpis dostupných souborů je výpis všech souborů ve složce. Při stahování souborů není generován přímo odkaz do složky. Přístup k souborům je pomocí php skriptu, aby bylo možné kontrolovat, zda má uživatel oprávnění stahovat tyto soubory.

5.11 Správa uživatelů

Správa uživatelů je rozdělena zvlášť pro portál a supervizory či operátory. V rámci správy uživatelů lze měnit heslo uživatele a jeho jméno a příjmení. U uživatelů portálu lze změnit i jeho oprávnění do přístupu k jednotlivým částem aplikace.

6 Implementace

Aplikace poběží na serveru s operačním systémem *Ubuntu 3.11* s nainstalovaným *Apache 2.2.22*, *MySQL 5.5.37* a *PHP 5.3.10*. Jednotlivé technologie jsou popsány v kapitole Technologie.

Pro implementaci aplikace byly zvoleny technologie:

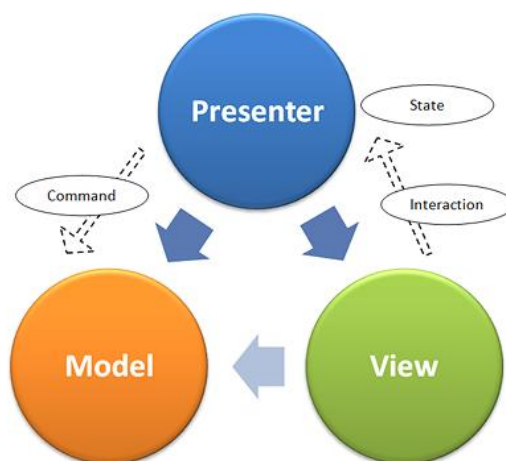
- PHP, verze 5.3.10
- MySQL, verze 5.5.37
- JavaScript
- HTML 5

Aplikace využívá tyto frameworky/knihovny:

- Nette, verze 2.0.10
- Dibi, verze 2.0
- jQuery, verze 1.9.1
- DataTable, verze 1.9.4

Aplikace je postavená na architektuře MVP (Model View Pre), kterou ilustruje Obrázek 7. Tato architektura vychází ze staršího typu MVC (Model View Controller). Jejich hlavní rozdíl je v chápání controlleru, který je v MVP nahrazen presenterem a přebírá na sebe více povinností. Presenter se stará o presentaci dat z modelu do šablony a udržuje stav aplikace (pomocí perzistentních proměnných) a zpracovává tři hlavní události:

- Změna pohledu (přechod na jinou stránku)
- Změna stavu (změna perzistentní proměnné)
- Příkaz modelu (aktualizace dat)



Obrázek 7. Ilustrace architektury MVP, zdroj [2]

6.1 Logické rozdělení aplikace

Aplikace je na základě MVP architektury a logiky Nette frameworku rozdělená do několika částí.

6.1.1 Presentery

Mají na starosti zpracovávání přichozích požadavků, předávání dat modelu, získávání dat z modelu a jejich předávání do view a udržování stavu aplikace. Každý presenter má vlastní akce a k těm přiřazené šablony. Dále presenter obsahuje továrnu na komponenty, ve které si vytvoří komponentu a předá jí šabloně (viz 6.1.3 Komponenty, str. 32).

Každý presenter má na starosti jednu část aplikace - úvod, přihlašování, operátorskou část, supervizorskou část a frontování kontaktů. Všechny presentery dědí od BasePresenteru.

6.1.1.1 BasePresenter

BasePresenter je předek všech presenterů. Má na starosti společné části aplikace jako jsou překlady, kontrola oprávnění, navigaci a nastavování debugmodu. Debugmode umožňuje zobrazování ladících informací a výpisů a je aktivovaný pouze pro uživatele s rolí developer.

Tuto část aplikace podrobněji zpracovává ve své práci kolega Martin Dendis.

6.1.1.2 Úvodní strana

Úvodní stranu zpracovává *homepagePresenter*, který přesměruje uživatele na úvodní stranu presenteru podle role uživatele. Role uživatele jsou:

- **Operátor** – Přesměruje uživatele na úvodní stranu operátorské části.
- **Supervizor** – Přesměruje uživatele na úvodní stranu supervizorské části.
- **Portál** – Přesměruje uživatele na úvodní stranu portálu.

6.1.1.3 Operátorská část

Tuto část zpracovává *operatorPresenter* a obsahuje tyto části:

- **Úvodní strana** (action *default*) – Zobrazí operátorovu frontu hovorů a výpis posledních hovorů.
- **Dotazník** (action *call*) – Informace o volaném kontaktu a dotazník, který bude operátor vyplňovat. Dotazník má na starosti komponenta *Call*.
- **Začátek hovoru** (action *startCall*) – Vytvoří záznam o začátku hovoru do databáze.

6.1.1.4 Supervizorská část

Tuto část má na starosti *SupervisorPresenter* a řeší rozhraní pro supervizora call centra.

- **Celková statistika hovorů** (action *default*) – Informace o tom, kolik bylo staženo kontaktů v této vlně a kolik kontaktů prošlo filtrováním. Dále informace o tom, kolik se má provést hovorů pro servis a prodej, a v jakém stavu jsou provedené hovory. Data jsou získávána pomocí modelu *Statistic*.
- **Statistika hovorů servis** (action *statisticServis*) – Informace o tom, kolik se má obvolat kontaktů, kolik kontaktů je k dispozici, kolik hovorů bylo už provedeno a v jakém stavu skončili provedené hovory. Dále jsou zde informace o všech servisech, kolik bylo celkově obvoláno kontaktů, kolik kontaktů se má ještě obvolat a kolik kontaktů je k dispozici. Data jsou získávána pomocí modelu *Statistic*.
- **Statistika hovorů prodej** (action *statisticSeller*) – Stejně jako statistiky hovorů servis akorát pro prodej. Data jsou získávána pomocí modelu *Statistic*.
- **Výpis provedených hovorů** (action *calls*) – Výpis všech provedených hovorů, možnost editace dotazníků, stažení nahrávky a schválení hovoru pro portál nebo jeho odstranění ze statistik. Má na starosti komponenta *Call*.
- **Stav operátorů** (action *operators*) – Zobrazení stavu operátorů, možnost jejich odhlášení a náslechu jejich hovorů. Data o stavu operátorů jsou získávána z telefonní ústředny *Daktela*.
- **Soubory pro portál** (action *files*) – Možnost nahrát soubory ke stažení na portálu (dále výpis těchto souborů a možnost smazání). V souborech lze vyhledávat a řadit je podle jednotlivých sloupců tabulky javascriptové komponenty *DataTable*.

- **Statistiky ke stažení** (action *download*) – Výpis statistik ke stažení. Jedná se o statistiky, které jsou generovány automaticky. Ve statistikách lze vyhledávat a řadit je podle jednotlivých sloupců pomocí tabulky javascriptové komponenty *DataTable*.

6.1.1.5 *Portál*

Tuto část zpracovává *portalPresenter* a řeší rozhraní portálu.

- **Výpis provedených hovorů** (action *calls*) – Výpis hovorů schválených pro zobrazení na portálu. Uživatel může pustit nahrávku hovoru, a zobrazit dotazník hovoru. Řeší komponenta *Call*.
- **Správa uživatelů portálu** (action *manageUser*) – Umožňuje editaci, mazání a vytváření nových uživatelů portálu, dále umožňuje nastavování oprávnění.
- **Soubory ke stažení** (action *files*) – Možnost nahrát soubory ke stažení na portálu, dále výpis těchto souborů a možnost smazání. V souborech lze vyhledávat a řadit je podle jednotlivých sloupců tabulky pomocí javascriptové komponenty *DataTable*.

6.1.1.6 *Frontování kontaktů*

Tuto část má na starosti *serverSidePresenter*. Akce tohoto presenteru jsou volány pomocí cronu v zadaném intervalu a nejsou přístupné z vnější sítě.

- **Nahrávání kontaktů do globální fronty** (action *updateQueue*) – Je volán každou minutu.
- **Nahrávání kontaktů do fronty operátora** (action *updateOperatorQueue*) – Odstraní kontakty odhlášených uživatelů a nahraje potřebný počet kontaktů do fronty operátora. Je volán každou minutu.
- **Nahrání vyčištěných kontaktů** (action *ContactUpdate*) – Spouští se jedenkrát týdně, a to vždy v sobotu večer.

6.1.2 *Modely*

Mají na starosti zpracování dat, jejich aktualizaci v databázi a načtení na základě požadavků z presenteru.

6.1.2.1 *OperatorState*

Získává data o operátorovi z databáze a z ústředny *Daktela*.

6.1.2.2 *AddContactToQueue*

Stará se o přesun připravených kontaktů do globální fronty (doplňování kontaktů do fronty podle kvót).

6.1.2.3 *ContactToOperator*

Stará se o přesun kontaktů z globální fronty do front operátorů.

6.1.2.4 *ContactsToDatabase*

Překopíruje vyčištěné kontakty do tabulky *contacts* a předzpracuje jejich data.

6.1.3 Komponenty

Znovupoužitelná část aplikace. Nejčastěji vizuální část stránky, ale může mít na starosti i část práce s daty a komunikaci s modely. Mnou byla implementována pouze komponenta Calls.

6.1.3.1 *Calls*

Tato komponenta má na starosti výpis provedených hovorů, možnost stáhnout si nahrávku hovoru a zobrazení dotazníku s následnou editací. V kontraktoru této komponenty se předává role uživatele. Tím je zajištěno, že například uživatel portálu nemůže ukládat dotazník.

7 Testování

Testování je rozděleno do dvou částí - testování bez uživatele a testování s uživatelem.

7.1 Testování bez uživatele

Pro testování *bez uživatele* byl zvolen testovací framework Nette Tester, který je napsaný v PHP a je od stejného autora jako Nette. Tento testovací Framework byl vybrán z několika důvodů. Hlavní důvodem je, že celý Nette framework je testovaný pomocí tohoto nástroje a mají společného autora, což zjednodušuje testování aplikací postavených na Nette frameworku. Další jeho výhodou je, že je napsaný v PHP, což umožňuje spouštět testy na kterémkoliv serveru s PHP a není potřeba další rozšíření, jako například rozšířené PHPUnit.

7.1.1 Testování

Každý test je z pohledu Nette Tester samostatný script, což umožňuje spouštět každý test samostatně. Při hromadném spouštění testů, jsou všechny testy spuštěny současně, ale každý ve svém vlákně, aby nedocházelo ke konfliktům mezi testy. Nette tester také umožňuje serializaci testů, která se hodí například ve chvíli, kdy by více testů pracovalo s databází najednou a navzájem se mohly ovlivňovat.

7.1.2 Testování presenterů

Testování presenterů umožňuje testovat aplikaci jako celek. Presentery byly testovány pomocí simulace http požadavku. Vzhledem k tomu, že jednotlivé akce presenteru generují HTML kód, Nette Tester nabízí knihovnu *Tester\DomQuery*, pomocí které je možné selektovat jednotlivé prvky HTML dokumentu. *Tester\DomQuery* používá syntax CSS selektorů.

7.2 Testování s uživatelem

Testování s uživatelem probíhalo týden při testovacím provozu call centra. Při testování byl běh call centra stejný jako při normální provozu. Aplikaci používalo 18 operátorů, kteří obvolávali připravené kontakty. Supervizoři sledovali statistiky a kontrolovali práci operátorů. Kromě funkčnosti bylo testováno uživatelské prostředí i součinnost všech částí call centra (naší aplikace, rozhraní pro stahování kontaktů, ústředny, softwarových telefonů). Součástí

testu byla také kontrola, která měla zjistit, zda jsou při reálném běhu aplikace kontakty správně stahovány a frontovány.

Při testování nebyly odhaleny žádné zásadní chyby. Odhaleno bylo pouze několik drobných chyb, které byly většinou způsobeny prací několika operátorů najednou a pomalejší odezvou ústředny nebo špatně popsaným rozhraním pro komunikaci s jinou částí call centra. Tyto chyby se při testování *bez uživatele* neprojevíly.

8 Závěr

Aplikace splňuje všechny funkční i nefunkční požadavky.

První verze aplikace byla spuštěna 1. 7. 2013. A od té doby bez závažných problémů funguje. Aplikace je postupně vylepšována, především po stránce zabezpečení a "hezčího kódu". Aplikaci používá dvanáct až dvacet operátorů současně. Každý týden je obvoláno pomocí této aplikace 3 400 kontaktů. Celkově od začátku běhu aplikace bylo obvoláno 152 251 kontaktů a celkově bylo provedeno 216 190 hovorů.

Aplikace je dále postupně vylepšována. Mezi hlavní plánované vylepšení patří lepší možnost správy uživatelů s možností přidávání a odebrání uživatelů. Je uvažováno o přepsání logiky frontování kontaktů do JAVY, což by umožnilo nepřetržité doplňování kontaktů do fronty. Mezi další možná rozšíření patří možnost zpracování příchozích hovorů a s tím spojené provázání s některou CRM aplikací (aplikace pro řízení vztahů se zákazníky).

Seznam obrázků

Obrázek 1. Role. Zdroj [11]	2
Obrázek 2. Znázornění oprávnění rolí uživatelů	3
Obrázek 3. Schéma práce operátora	4
Obrázek 4. Schéma zpracování dat	7
Obrázek 5. Nette Framework, chybová hláška. Zdroj [3]	17
Obrázek 6. Návrh databáze	22
Obrázek 7. Ilustrace architektury MVP, zdroj [2]	29

Zdroje

- [1] Nette Framework [online]. [cit. 16.5.2014]
Dostupné z: <<http://www.nette.org>>
- [2] Nette Framework: MVC & MVP [online]. [cit. 17.5.2014]
Dostupné z: <<http://www.zdrojak.cz/clanky/nette-framework-mvc-mvp/>>
- [3] Nette Framework, Debugování a zpracování chyb [online]. [cit. 17.5.2014]
Dostupné z: <<http://doc.nette.org/cs/2.1/debugging>>
- [4] Yii mieša karty PHP frameworkov [online]. [cit. 4.4.2014]
Dostupné z: <<http://www.zdrojak.cz/clanky/yii-miesa-karty-php-frameworkov/>>
- [5] The Fast, Secure and Professional PHP Framework [online]. [cit. 4.4.2014]
Dostupné z: <<http://www.yiiframework.com/>>
- [6] Why use Zend Framework? [online]. [cit. 4.4.2014]
Dostupné z: <<http://framework.zend.com/>>
- [7] Velký test PHP frameworků: Zend, Nette, PHP a RoR [online]. [cit. 4.4.2014] Dostupné z:
<<http://www.root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror/>>
- [8] VELESTRUČNÉ TESTOVÁNÍ PRESENTERŮ V NETTE [online]. [cit. 22.5.2014]
Dostupné z: <<http://phpfashion.com/velestrucne-testovani-presenteru-v-nette>>
- [9] Daktela API [online]. [cit. 20.8.2013] Dostupné z:<<http://daktela.com/api/api.html>>
- [10] Jak si zjednodušit práci s PHP projektem pomocí příkazové řádky [online]. [cit. 10.5.2014] Dostupné z:
<<http://www.zdrojak.cz/clanky/jak-si-zjednodusit-praci-s-php-projektem-pomoci-prikazove-radky/>>
- [11] PHP [online]. [cit.20.5.2014] Dostupné z: <<http://www.php.net/>>
- [12] Dibi is Database Abstraction Library for PHP 5. [online]. [cit. 22.5.2014] Dostupné z:
<<http://dibiphp.com/>>
- [13] SARAY, Aaron H. Professional php design patterns. 1st ed. Indianapolis, IN: Wiley Pub., inc, 2009, p. cm. ISBN 04-704-9670-3.
- [14] SCHWARTZ, Baron, Peter ZAITSEV a Vadim TKACHENKO. High performance MySQL. 3rd ed. Cambridge [Mass.]: O'Reilly, c2012, xxviii, 793 p. ISBN 14-493-1428-7.

Příloha A

Obsah příloženého CD

BP.pdf – Eletronická podoba této práce

openminder.zip – Zdrojové kódy této aplikace

install.txt - popis instalace aplikace