



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

**WEBOVÁ APLIKACE PRO TVORBU PREDIKCÍ CALL
CENTRA**
WEB APPLICATION FOR MAKING PREDICTIONS OF A CALL CENTRE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

David Mička

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Vladimír Bartík Ph.D.

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Mička David**

Obor: Informační technologie

Téma: **Webová aplikace pro tvorbu predikcí call centra**

Web Application for Making Predictions of a Call Centre

Kategorie: Informační systémy

Pokyny:

1. Seznamte se s principy tvorby webových aplikací a možnými vývojovými prostředími.
2. Seznamte se se současným stavem v oblasti nástrojů pro predikci call center a existujícím algoritmem Erlang. Analyzujte slabé stránky těchto existujících řešení.
3. Analyzujte požadavky na informační systém pro predikci vytíženosti call centra, který bude nahrazovat současné řešení (Excel) a odstraní nedostatky zjištěné analýzou v bodu 2.
4. Navrhněte informační systém dle požadavků a implementujte jej ve zvoleném vývojovém prostředí.
5. Ověřte funkčnost systému na vhodném vzorku dat a porovnejte úspěšnost predikce s existujícími řešeními.
6. Zhodnoťte dosažené výsledky a další možné pokračování tohoto projektu.

Literatura:

- Erlang: The foremost site for call center forecasting tools, download the software you require today. Dostupné na: <http://www.erlang.co.uk/default.htm>
- Welling, L., Thomsonová, L.: PHP a MySQL: rozvoj webových aplikací. Vyd. 1. Praha: SoftPress, 2003, 910 s. ISBN 80-86497-60-7.
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015. ISBN: 978-80-251-4573-9

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Cílem práce je vytvořit aplikaci pro tvorbu predikcí call centra, která by mohla nahradit stávající řešení predikcí v prostředí firmy Kiwi.com s.r.o. Výsledná aplikace by měla být intuitivnější a snazší na používání než systém Verint, či tabulkové řešení predikcí a zároveň by měla poskytovat dostatečné množství použití pro tvorbu taktických predikcí, díky kterým lze reagovat na nastávající situace a nastavovat reálná očekávání pro management zákaznického centra.

Abstract

The goal of this thesis is to create a web application for creation of call centre predictions. The app should be able to replace current solutions that are in use in the daily operation of Kiwi.com s.r.o. The app should be more intuitive and easier to use and maintain than Verint or the spreadsheet solution of doing predictions. It should also have enough options for creation of tactical forecasts that allow the company to react on upcoming situations and should help set realistic expectations for the management of our customer centre.

Klíčová slova

Predikce, odhad, call centra, zákaznická centra, taktické predikce, informační systémy, webová aplikace, úroveň servisu, Kiwi.com, PHP, Nette, MySQL

Keywords

Prediction, forecast, call centres, information systems, tactical forecast, web applications, service levels, Kiwi.com, PHP, Nette, MySQL

Citace

MIČKA, David. Webová aplikace pro tvorbu predikcí call centra. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Bartík Ph.D.

Webová aplikace pro tvorbu predikcí call centra

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka Ph.D. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Mička

17. 5. 2017

Poděkování

Tímto bych chtěl poděkovat panu Ing. Vladimíru Bartíkovi Ph.D., Bc. Ladislavu Vaškovi a Ing. Jakubovi Bártovi za vedení práce, facilitaci práce ve firmě Kiwi.com a konzultace týkajících se zachování struktury aplikace používané ve firmě.

© David Mička, 2018

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	3
1.1 Kiwi.com s.r.o.....	3
1.2 Motivace	3
1.3 Cíl práce.....	4
2 Principy fungování call center	5
2.1 Obecné pojmy a metriky call center	5
2.1.1 IVR	5
2.1.2 ASA	5
2.1.3 ACD.....	5
2.1.4 SLA.....	5
2.2 Hlavní metriky použité v algoritmu.....	7
2.2.1 Predikce množství (Volume forecast).....	7
2.2.2 Množství agentů.....	7
2.2.3 Plánovací období	7
2.2.4 Absence.....	7
2.2.5 Obsazenost (Occupancy)	8
2.2.6 Kontakty za hodinu	8
2.2.7 Denní křivka (Intraday curve).....	8
2.3 Taktické predikce.....	9
3 Použité technologie.....	10
3.1 Server.....	10
3.2 HTML.....	10
3.3 CSS	10
3.4 PHP.....	10
3.5 Framework.....	11
3.6 Databáze	11
3.7 Phinx.....	11
3.8 Bootstrap v4.1	12
4 Srovnání dostupných řešení	13
4.1 Problém malého množství hovorů	13
4.2 Erlang.....	13
4.3 Verint	14
5 Specifikace požadavků.....	15

5.1	Firemní požadavky	15
6	Implementace aplikace	17
6.1	Návrh databáze	17
6.2	Rozvržení webu	18
6.3	Popis jednotlivých stránek	19
6.3.1	Přihlášení	19
6.3.2	Homepage	20
6.3.3	Predictor	22
6.3.4	Upload data	24
6.3.5	Upload CSV files	25
7	Testování	26
7.1	Běžný provoz	26
7.2	Nárůst absence	28
7.3	Zvýšení množství hovorů v intervalu	29
7.4	Shrnutí testování	31
8	Závěr	32

1 Úvod

Za poslední desetiletí rapidně vzrostlo množství webových služeb, s čímž přichází i potřeba zákaznické podpory těchto služeb. Poskytování co nejkvalitnějšího servisu je dnes klíčovým faktorem pro vytvoření základny stálých zákazníků, bez kterých je pro konkurenci relativně snadné naklonit váhy jejich směrem a generovat zisk na úkor ostatních.

Pro to, aby však zákaznická podpora dosahovala co nejlepších výsledků, zákazníci nečekali dlouho na linkách, nebo se nemohli dovolat v případech nouze, je nutné mít optimální množství agentů zákaznické podpory ve správný čas a zároveň udržovat jistou úroveň produktivity těchto agentů.

K takovým účelům se v zákaznických centrech používají predikce (forecasts), díky kterým je možnost předpovídat, jaká situace s daným počtem agentů v daný čas může nastat. Tato práce se bude zabývat především taktickými predikcemi, tedy krátkodobými predikcemi v momentě, kdy jsou již agenti na směny naplánováni. Za pomoci těchto predikcí lze reagovat na aktuální situace, či trendy ovlivňující provoz, nebo například na změny v množství agentů a nastavit tak reálná očekávání pro management.

Tato práce je zpracována pro firmu Kiwi.com s.r.o.

1.1 Kiwi.com s.r.o.

Firma Kiwi.com s.r.o. [1] je firma působící v cestovním ruchu již od roku 2012, zabývající se prodejem letenek a propojováním nabídek letů různých leteckých společností, vytvářející tak široké spektrum letových kombinací. Jedním z hlavních benefitů našeho produktu je Kiwi.com guarantee, což je záruka náhrady letů v případě různých cestovních komplikací. Jako jeden z příkladů bych uvedl zpoždění jednoho z letů, v jehož důsledku je následující let zmeškán. V této situaci je pak ze strany Kiwi.com zákazníkovi nabídnut náhradní let zcela zdarma, či za zvýhodněných podmínek.

Kiwi.com také disponuje velmi uživatelsky přívětivým vyhledáváním, které umožňuje zákazníkům například vyhledávání letů z letišť v daném okruhu z jakéhokoliv bodu na mapě, či vyhledávání podle zemí, měst, nebo vyhledání letů bez upřesnění destinace, umožňující tak objevování levých letů do zajímavých míst světa, například pro spontánní dovolené.

Kiwi.com v dnešní době zaměstnává více než 1500 zaměstnanců ve více než devíti zemích světa, z čehož je velký podíl zaměstnán v zákaznické podpoře, ať už v manažerských či agentských pozicích.

S takovýmto množstvím se zvyšuje nutnost plánování a vytváření predikcí, aby bylo zaručeno co nejoptimálnější pokrytí a tím také co nejlepší a nejrychlejší servis pro zákazníky.

1.2 Motivace

Motivací pro zvolení tohoto tématu, bylo mé praktické zapojení do tvorby predikcí v rámci mých každodenních pracovních povinností. I přes to, že je hlavním nástrojem oddělení aplikace Verint, která tvorbu predikcí obsahuje, není oddělením pro tvorbu predikcí používána. Využívá se v rámci oddělení

především na sestavování plánů směn zaměstnanců na základě predikce, nahrané do systému po vytvoření mimo program Verint.

V rámci oddělení je používán především algoritmus Erlang jako doplněk programu Microsoft Excel 2016. Po zkušenostech s tímto algoritmem jsem se v minulém roce rozhodl, vytvořit vlastní kalkulaci, která by více seděla na provoz zákaznického centra firmy Kiwi.com, které se od běžných zákaznických center výrazně liší množstvím aktivit probíhajících mimo samotné kontakty se zákazníky. To je dáno především produktem, pro který je podpora poskytována. U leteckého průmyslu je v mnoha případech po kontaktu se zákazníkem nutné kontaktovat aerolinku, či provádět úpravy rezervace letenek dle přání zákazníka. Mnou vytvořený algoritmus se v minulém roce osvědčil a běžně dosahoval odchylky 5 % pod, či 5 % nad reálným výsledkem. Tento algoritmus byl však také v prostředí Microsoft Excel, čímž se stával náročnějším a méně pohodlným pro obsluhu a údržbu.

1.3 Cíl práce

Cílem práce je tedy vytvořit webovou aplikaci, která by usnadňovala každodenní používání algoritmu, rozšířila jeho možnosti o tvorbu taktických predikcí (dříve sloužil pouze pro tvorbu přesných predikcí) a zároveň přidala vizualizace sloužící ke snazšímu srovnání statistik o jednotlivých dnech při každodenních revizích výsledků. Přidání taktických predikcí umožní oddělení Workforce Management snáze reagovat na nečekané situace, které jsou v letectví téměř na denním, či týdenním pořádku.

V příštích kapitolách bude nejprve vysvětlena základní problematika call center a termíny s nimi spojenými, popsány použité technologie, srovnány stávající řešení používané ve firmě Kiwi.com a nakonec struktura, implementace a testování samotné aplikace.

2 Principy fungování call center

Call centra v dnešní době tvoří velkou součást webových služeb, jako jsou e-shopy, či různé aplikace. S jejich rozvojem je spojena nutnost tvorby predikcí objemu kontaktů, na základě kterých se poté tvoří kapacitní plány, tedy plány rozložení zaměstnanců do jednotlivých intervalů, ať už to jsou měsíce, týdny, dny, či hodiny.

V této kapitole se budu zabývat principy fungování call center jak v obecné rovině, tak v rámci Kiwi.com, ve kterém výsledná aplikace bude testována. Vysvětleny zde budou jak nejdůležitější metriky a pojmy využívané pro výslednou predikci z hlediska provozu, tak i z hlediska Workforce, tedy oddělení, které plánuje kolik agentů bude v daný interval v práci. Nazvěme tyto dvě kategorie např. *Obecné metriky* a *Metriky použité v algoritmu*.

2.1 Obecné pojmy a metriky call center

Pro snazší pochopení fungování call centra je nutné vysvětlit důležité technické pojmy s nimi spojené, týkající se jak technických aspektů, tak různých metrik určených pro analýzu výkonnosti a efektivnosti call centra.

2.1.1 IVR

IVR neboli Interactive voice response [2] je technologie umožňující komunikaci počítače a člověka pomocí tónové volby. V systému zavedeném v Kiwi.com je IVR definované tzv. Call flows, které dle vstupů od volajícího určují, do které fronty (queue) bude příchozí kontakt přiřazen, a tedy i kým bude obsluhován.

2.1.2 ASA

Průměrná rychlost odpovědi (Average Speed of Answer) je definována jako čas od příchodu kontaktu do příslušné fronty do zvednutí kontaktu agentem. Zahrnuje v sobě čas, po který kontakt čekal ve frontě a dobu směrování kontaktu agentovi. Routing probíhá pomocí ACD.

2.1.3 ACD

Automatic Call Distribution [3] systém, tedy automatický systém distribuce hovorů, je systém směřující kontakty agentům na základě předem definovaných pravidel. Pravidla mohou být definována mnoha způsoby, například pomocí definování dovedností agenta a jeho úrovních v těchto dovednostech. Pak je tedy možné kontakt s danými vlastnostmi směřovat nejvhodnějšímu agentovi. Pro fungování aplikace předpokládáme, že všichni agenti mají stejné dovednosti a směrování probíhá na základě doby čekání na kontakt, tedy na principu FIFO fronty.

2.1.4 SLA

SLA neboli Service Level Agreement (někdy také Service Level, SVL) je metrika známá ze síťové problematiky. Stejně jako u sítí znamená určitou úroveň servisu mezi poskytovatelem a zákazníkem

(klientem). U telefonie se však oproti síti, kde se SLA cíle stanovují například na uptime síť, nastavují na procentuální garanci přijmutí hovoru do určitého časového intervalu. SLA může být u call center, jak smluvně garantováno s třetí stranou, jež provádí náhodné testy kontaktováním podpory, tak interně pro nastavení firemních cílů, a tím nastavovat interní cíle pro výkonnost celé zákaznické podpory.

SLA může být u call center vypočítáno různými způsoby. Některé z výpočtů uvádím v tabulce 2.1 a v nich použité výrazy v odrážkách pod tabulkou.

Název	Výpočet
SLA _A	$SLA_A = \frac{Calls_{InSLA}}{Calls_{Handled} + Calls_{Abandoned}}$
SLA _B	$SLA_B = \frac{Calls_{InSLA}}{Calls_{Handled}}$
SLA _C	$SLA_C = \frac{Calls_{InSLA}}{Calls_{Handled} + Calls_{AbandonedOutSLA}}$

Tabulka 2.1 Přehled možných výpočtů Service Level Agreement

- *Calls_{InSLA}* – Hovory přijaté v čase definovaném SLA, v případě Kiwi.com tedy 30 vteřin od příchodu do fronty
- *Calls_{Handled}* – Celkové množství přijatých (zvednutých) hovorů za evaluovaný interval
- *Calls_{Abandoned}* – Celkové množství hovorů, které zákazník položil, než mohly být zvednuty agentem
- *Calls_{AbandonedOutSLA}* – Celkové množství hovorů, které zákazník položil, než mohly být zvednuty agentem, s časem stráveným ve frontě delším, než je čas definovaný SLA, tedy v Kiwi.com opět 30 vteřin. Touto modifikací metriky zaniká možnost spekulace, že by některý z agentů kontakt mohl v rámci 30 vteřin zvednout.

V Kiwi.com je využíván výpočet odpovídající SLA_B z tabulky 2.1. Tento způsob je vhodný v případech, kdy má firma nastavené samostatné cíle pro *Service Level* a *Abandon rate*. Pokud by *Abandon rate* neměla v rámci oddělení samostatně nastavené cíle, bylo by vhodnější použít výpočtu A, či C, kde je jsou již položené hovory zahrnuty. *Abandon rate*, tedy míra položených hovorů, je metrika říkající, jaké procento kontaktů volající položí dříve, než je přijme agent. Vzorec pro *Abandon rate* je tedy následovný.

$$AR = \frac{Calls_{Abandoned}}{Calls_{Queued}}$$

Calls queued, jsou veškeré hovory, které se přes IVR dostaly do fronty kontaktů. Výraz je možné nahradit zápisem *Calls_{Handled} + Calls_{Abandoned}*, tedy součtem hovorů přijatých (odbavených) agentem a položených zákazníkem.

2.2 Hlavní metriky použité v algoritmu

Tato kapitola popisuje metriky využívané pro samotnou kalkulaci výsledného SLA, nebo s použitými metrikami úzce souvisí a jsou důležité pro porozumění problematice. Byť jsou tyto metriky běžně používané ve většině call center, mohou mít rozdílné definice, či vzorce pro kalkulaci. To je zapříčiněno odlišností produktů firem majících zákaznická centra. Zde jsou popsány metriky dle využití v Kiwi.com.

2.2.1 Predikce množství (Volume forecast)

Výsledná aplikace předpokládá dostupnost předpovědi příchozího objemu hovorů. Tento údaj je nejdůležitější, neboť je hlavním údajem, dle kterého je výsledná predikce udělána. Tato metrika specifikuje, kolik kontaktů přijde v daném časovém období do fronty, tedy od momentu, kdy volající/kontakt projde IVR a je vytvořena jeho instance v příslušné frontě. V aplikaci je toto množství rozdělené po hodinových intervalech.

2.2.2 Množství agentů

Druhou nejdůležitější metrikou pro tuto aplikaci je množství agentů pro daný interval, tedy v tomto případě hodina, či den. Tato metrika udává, kolik agentů je v daný interval v práci, tedy kolik jich bude k dispozici pro přijímání hovorů. Tato metrika není konečné číslo využitě pro samotnou predikci výsledků a je ovlivňována dalšími metrikami popsány níže. Množství agentů je určováno předem pro dané plánovací období.

2.2.3 Plánovací období

Je interval, pro který je stanovováno množství agentů na směny a je z pravidla dlouhé jeden měsíc a déle. Po zveřejnění množství agentů pro dané plánovací období se již většinou množství agentů neupravuje. Ve výjimečných případech se množství agentů může změnit, avšak tímto se není třeba zajímat, neb pro potřebu aplikace je toto číslo definované uživatelem, tedy v případě změny se předpokládá s upravením hodnoty v databázi a zároveň jedním z cílů aplikace je tuto nutnost odhalit.

2.2.4 Absence

Absence, jak již název napovídá je procento agentů, u kterých se počítá s jejich nepřítomností na směně. Tento údaj lze rozdělovat do dvou podkategorií. Plánovaná a neplánovaná absence.

- **Plánovaná absence** – Do plánované absence se počítají události jako dovolené, neplacená volna a je s ní počítáno již při plánování množství agentů, kteří mají být v daný interval v práci. Pro daný interval je tedy za ideální situace množství agentů vyšší, než kolik by jich mělo být potřeba.
- **Neplánovaná absence** – Neplánovanou absencí může být například nepřítomnost z důvodu neschopnosti práce, nemoc, neomluvená nepřítomnost, či jakýkoliv jiný důvod nepřítomnosti, který nelze předpovídat dostatečně dlouho předem, a tedy s ním nelze počítat při stanovování množství agentů pro dané plánovací období.

2.2.5 Obsazenost (Occupancy)

Tento údaj značí, kolik procent času je agent přítomný u svého počítače, tedy připravený přijímat příchozí hovory. Pro potřeby aplikace není třeba tento údaj nutně brát jako procento z doby směny, ale stačí ho uvažovat jako procento z daného intervalu (zde tedy hodina), neboť se aplikace nezabývá rozložením pauz v rámci směny, ale uvažuje rovnoměrné rozložení pauz.

Obsazenost lze zadávat buď jako předpokládanou, tedy jakýsi cíl, či kvótu která je po agentech očekávána, nebo jako průměrnou z výsledků za určité období před dnem tvoření predikce. Vzhledem k tomu, že aplikace má sloužit pro predikci výsledků v krátkém časovém horizontu (dva týdny až měsíc), je vhodnější používat průměrnou obsazenost za například poslední dva týdny, neboť nejvíce reflektuje aktuální situaci.

2.2.6 Kontakty za hodinu

Kontakty za hodinu vyjadřují množství kontaktů odbavených agentem během horizontu jedné hodiny. Tento údaj lze rozdělit na příchozí a odchozí hovory za hodinu. Pro účely této aplikace jsou důležité především příchozí kontakty za hodinu, avšak i odchozí kontakty agenty na nějakou dobu vytěžují, tedy je musíme brát také v úvahu. Stejně jako u obsazenosti lze tento údaj uvažovat buď jako předpokládaný, nebo jako reálnou hodnotu z posledních dat, a i zde je nejvhodnější využívat průměru za poslední dva týdny.

2.2.7 Denní křivka (Intraday curve)

Postupným shromažďováním dat o příchozích hovorech za určité časové období (například jeden rok) s rozpadem po hodinách vznikají denní křivky. Denní křivky vyjadřují procentuální podíl příchozích hovorů za danou hodinu oproti celému dni. Pokud tedy analýzou historických dat zjistíme, že během první hodiny dne přijmeme v průměru 100 kontaktů a celkový denní objem je v průměru 1000 kontaktů, bude první hodina v denní křivce ohodnocena deseti procenty. Tento údaj aplikace využívá při vkládání nových dat, kdy po zadání celkového množství hovorů rozpočítá a uloží objem hovorů pro každou hodinu dne.

2.3 Taktické predikce

Aplikace nemá sloužit zcela jen pro tvoření běžné predikce výsledků, ale hlavně pro tvoření taktických předpovědí. Taktickou předpovědí uvažujeme v podstatě simulaci výsledku, pokud nastane nějaká nepředpokládaná událost, či se v daném období podstatně změní některý ze vstupních parametrů.

Jako příklad lze uvést například chřipkovou epidemii, která zásadně ovlivní absenci, tudíž i výsledné pokrytí hovorů. V aplikaci je možné tedy tento údaj při vypuknutí oné epidemie navýšit, čímž analytik získá přehled o tom, jak může například dalších 7 dní vypadat pokrytí směn a předat dále impuls k rychlé úpravě.

Dalším z případů, vycházejícím z provozu Kiwi.com, mohou být situace ovlivňující specifickou skupinu lidí, například sněhová bouře ve Spojených Státech Amerických ovlivňující provoz některých letišť. V takové situaci lze očekávat nárůst množství hovorů v časech odpovídajících nejrušnějším hodinám dne – mezi pátou hodinou odpolední a půlnocí středoevropského času, tedy během denních hodin v USA, během kterých létá velké množství letů.

Obecně mají taktické předpovědi sloužit buď ke krátkodobé úpravě směn či činnosti jednotlivých agentů, nebo pro vytvoření realistického očekávání pro vyšší management, tedy management předem informovat o situacích, které mohou mít na celkové výsledky oddělení negativní dopad.

3 Použité technologie

Před zahájením tvorby aplikace je nutné se zaměřit na technologie, které budou při samotné tvorbě využívány. Ne každý jazyk, framework, či IDE je vhodné k tvorbě dané aplikace, proto je velice důležité zvážit požadavky na tvorbu aplikace, její údržbu, jazyk databáze apod.

3.1 Server

Pro práci na aplikaci jsem se rozhodl využít aplikace Wamp server v3.0 [4], která se při své instalaci postará o nainstalování PHP ve zvolené verzi, serveru Apache ve verzi 2.4.27, aplikace PhpMyAdmin a Admineru. Díky této aplikaci je nastavení serveru a majority potřebných nástrojů pro psaní webových aplikací otázkou minut. Aplikaci momentálně vyvíjím pouze na localhost, ale převedení na firemní server je v plánu zhruba v polovině června 2018.

3.2 HTML

Základním jazykem pro tvorbu webů je *Hypertext Markup Language* [5], což je značkovací jazyk vyvinutý v roce 1991, využívající takzvaných HTML tagů pro označení jednotlivých elementů stránky. Tyto tagy jsou poté prohlížečem využívány pro vykreslení stránky, samy se však v prohlížeči nevykreslí. Tento jazyk byl postupem času vylepšován a oproti prvotním verzím je pomocí něho dnes ve verzi 5 možné přímo psát webové aplikace, bez nutnosti využití Javy apod.

3.3 CSS

Jazyk CSS (Cascading Style Sheets) [6] určuje jakým způsobem budou jednotlivé elementy webové stránky zobrazeny a jeho zavedením zmizela nutnost tyto vlastnosti definovat přímo v HTML tazích, což vedlo k zpřehlednění kódu a separaci obsahu a vzhledu stránky. Díky CSS je velice snadné pomocí tříd definovat různé zobrazení stejných elementů.

3.4 PHP

Při výběru vhodného programovacího jazyku pro mě byla rozhodujícím kritériem možnost integrovat výslednou aplikaci do interního systému firmy. Z tohoto důvodu bylo mou volbou PHP [7]. Je to velice často používaný skriptovací jazyk, jehož nespornou výhodou je jednoduchost integrace s HTML a množství dostupných rozšíření a frameworků z něj dělá nástroj, pomocí něhož lze webovou aplikaci vyvinout za zlomek času ve srovnání s jinými jazyky.

Další nespornou výhodou PHP je jeho nezávislost na platformě [8], podpora veškerých běžných serverů a prohlížečů a možnost nasazení na majoritě databází. PHP je opensource, což také přispělo k jeho rozvoji a velké popularitě.

PHP využívám ve verzi 5.6.31 a jako IDE jsem si zvolil program PhpStorm [9] od firmy JetBrains, který je vytvořen tak, aby co nejnadhěji pracoval s velkým množstvím frameworků, včetně mnou využívaným Nette [10]. Program PhpStorm využívám se vzdělávací licencí, udělenou na školní email. Tento program je také využíván v rámci PHP týmu firmy Kiwi.com s.r.o.

3.5 Framework

Nespornou výhodou PHP je množství frameworků, které pro tento jazyk byly vyvinuty. Jejich využitím vývojářům odpadá nutnost zabývat se mnoha aspekty, o které se stará samotný framework, jako například low level security. PHP frameworky také rozdělují datový model, řídicí logiku a uživatelské rozhraní, tedy následují architekturu MVC. Mezi nejznámější frameworky [11] patří například Laravel, který je v dnešní době pravděpodobně nejpoužívanějším, či Symfony nebo v České republice vyvinutý framework Nette.

Jak bylo již zmíněno, klíčová pro mě byla možnost nasazení výsledné aplikace do interního systému jako doplňující nástroj pro oddělení Workforce Management. Z tohoto důvodu jsem se rozhodl využít framework Nette ve verzi 2.4. Mezi jednu z hlavních výhod Nette je jeho důraz na bezpečnost, o kterou se vývojář do určité míry nemusí starat, obsahuje výborné ladící nástroje jako je Tracy [12], která vývojáři poskytne veškeré údaje nutné pro odstranění chyb a díky Nette je možné Tracy zobrazovat pouze na produkční verzi a uživatelům zobrazovat uživatelsky přívětivé chybové hlášky.

3.6 Databáze

Při výběru programu pro tvorbu a údržbu databáze jsem uvažoval nad prací s Microsoft SQL Server, ale nakonec jsem se rozhodl pro systém řízení báze dat MySQL a správu v PhpMyAdmin [13], který je volně dostupný a mám s ním již předchozí zkušenosti ze školních projektů. Zároveň byla tato volba ovlivněna konzultací s vývojáři interního systému, aby byla zaručena snadná možnost zakomponování aplikace do firemního systému. Kvůli snadné integraci jsem se rozhodl použít i nástroje Phinx.

3.7 Phinx

Při tvorbě aplikace jsem se rozhodl použít open-source nástroje Phinx [14] pro tvorbu a správu migrací. Pomocí Phinx migrací je vytvoření tabulek databáze velice snadné a vývojář si může vybrat, zda je psát v SQL, nebo pouze v PHP. Využitím možnosti psaní v PHP je výslednou databází možné snadno převést na jiného poskytovatele databází, tedy by bylo otázkou sekund přejít například na Microsoft SQL Server.

Každá migrace obsahuje třídu *Up*, *Down*, či u první migrace funkci *Change*. Za pomoci těchto funkcí se definují akce, které budou při migraci (*Up*), či rollbacku (*Down*) provedeny. Phinx také dokáže k funkci *Up* automaticky vytvořit funkci *Down*, takže se o to vývojář nemusí starat, pokud ovšem nechce při rollbacku vykonat jinou akci než reversi funkce *Up*.

3.8 Bootstrap v4.1

V aplikaci jsem se rozhodl využít i front-end frameworku Bootstrap [15] ve verzi 4.1. Tento framework nabízí vývojáři velké množství CSS stylů, či rozšíření v Javascriptu, které bez nutnosti zásahu řeší například rozložení stránky na mobilních zařízeních škálováním velikosti jednotlivých prvků.

Bootstrap využívá systému mřížky, dle které lze snadno určovat velikost elementů. Maximální počet sloupců mřížky je 24. Díky tomu lze například určit šířku jakéhokoliv elementu pomocí CSS třídy *col*, volitelně následovanou pomlčkou a číslem od 1 do 24. Toto číslo specifikuje, kolik sloupců má daný element zabírat. Při použití pouze třídy *col* využívají elementy rovnoměrného rozdělení, tedy v případě dvou elementů `<div>` umístěných vedle sebe, budou oba elementy zabírat stejně širokou oblast rodičovského elementu, či stránky.

Kromě definice stylů je Bootstrap také výborným pomocníkem pro nastolování celé stránky, použitím buď placených, či volně dostupných šablon. Pro potřebu aplikace jsem zvolil volně dostupnou šablonu *Dashboard* [16].

4 Srovnání dostupných řešení

V této kapitole se budu zabývat problémem malého množství hovorů, jedním z nejpoužívanějších algoritmů pro výpočet SLA a aplikací pro tvorbu predikcí aktuálně používanou ve firmě Kiwi.com.

4.1 Problém malého množství hovorů

U majority stávajících řešení se setkáváme s problémem malého množství hovorů. Tento problém vzniká při pokusech o vytvoření predikce výsledku u telefonních linek s malým denním počtem hovorů. Vhodným příkladem jsou zákaznická centra, či linky, specifikující se pouze na podporu v denních hodinách s denním množstvím okolo 100 kontaktů.

Při menším množství hovorů pak hraje velkou roli dostupnost agentů v době příchodu hovoru, tedy optimalizace jejich rozvrhu tak, aby například agenti neměli obědovou pauzu ve stejný čas.

Pro linku o cca 100 kontaktech během 8 hodinové otevírací doby může stačit na pokrytí cca 7 agentů (pokud počítáme i s plánovanou absencí a produktivitou okolo 3 kontaktů za hodinu. Majorita dnes dostupných řešení při odhadu kapacit uvádí pro stejně množství hovorů cca 15 agentů.

Tato aplikace tento problém řeší díky kalkulaci, která neřeší plánování více agentů pro případy vysoké absence, ale počítá rovnou s počty naplánovaných agentů, tedy je plánování více agentů prováděno člověkem, nikoliv algoritmem.

4.2 Erlang

Algoritmus Erlang [17] byl vymyšlen Agner Krarup Erlangem, narozeným v roce 1878 v dánském Lonborgu, který byl průkopníkem ve studiu telekomunikace. Nejdůležitější publikací, která obsahovala kalkulaci pro ztrátu hovorů a čekací dobu (později na jeho počest nazvanou Erlang), vydal Erlang v roce 1917. Kalkulace je poměrně komplexní a pracuje s pravděpodobností příchodů kontaktů v rámci daných intervalů, ale zabývá se pouze pokrytím příchozích kontaktů.

Výpočet Erlang pro výpočet Service Levelu potřebuje jako vstupní data množství agentů pro daný interval, průměrnou dobu obsluhy, očekávané množství příchozích kontaktů a požadovanou dobu zvednutí hovoru, tedy cílový čas pro Service Level.

Tento výpočet je dostatečně komplexní, aby dosáhl vysoké přesnosti za předpokladu, že agenti odbavují pouze příchozí kontakty. V případě Kiwi.com ale agenti v mnoha případech musí ještě kontaktovat aerolinky, či odepsat zákazníkovi, takže výpočet za pomoci množství příchozích kontaktů a doby obsluhy není dostačující.

Proto jsem pro potřeby Kiwi.com zvolil údaj *Cases per hour*, který je interním měřítkem produktivity. Tento údaj značí průměrný počet příchozích kontaktů za hodinu práce agenta, takže v něm jsou nepřímou zahrnuty i odchozí kontakty, čímž se tento údaj stává ideálním pro výpočet pokrytí příchozích kontaktů.

Doplněk Erlang pro Microsoft Excel, či Google Sheets obsahuje více než 20 funkcí, díky kterým je možné počítat různé metriky jako například průměrný počet kontaktů ve frontě, očekávaný počet položených kontaktů, průměrnou dobu do zvednutí telefonu, kapacitu na počet zvednutých hovorů, či procento času, které agenti stráví na telefonu [18]. Pro potřeby predikce se v rámci Kiwi.com

používá především funkce pro výpočet možného SLA, na základě vstupních parametrů zmíněných výše. Na úrovni vedení oddělení je Erlang také využíván pro výpočet potřebného počtu agentů na daný počet hovorů, což však není konečné číslo, a proto vzniká nutnost tvorby predikcí.

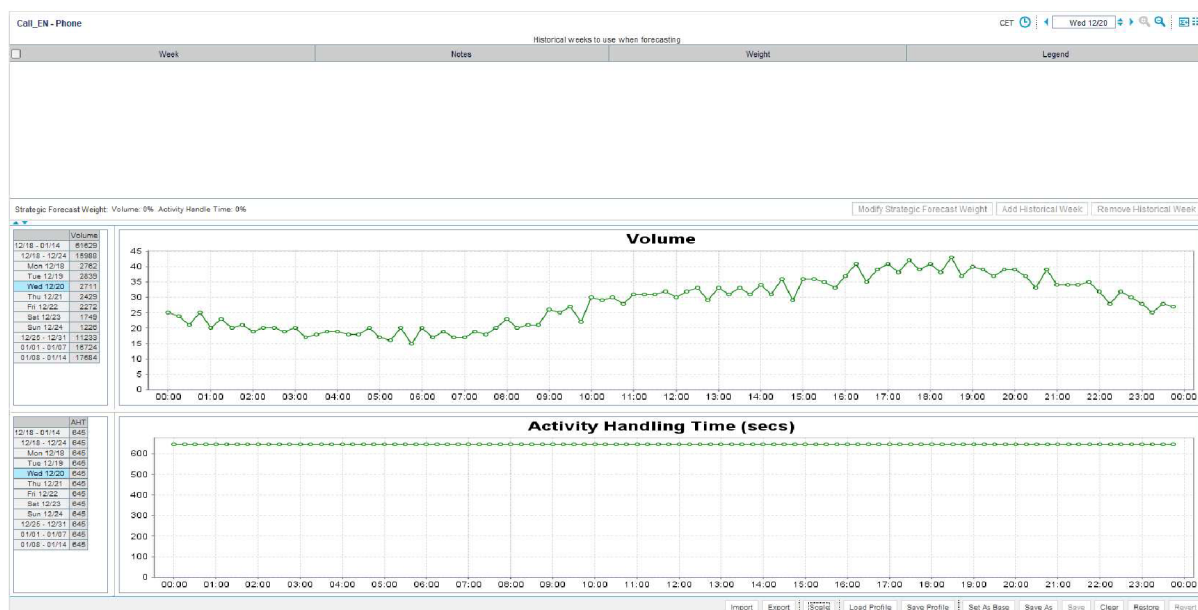
4.3 Verint

Verint [19] je komplexní workforce management systém, vyvinutý firmou Verint Systems, obsahující nástroje od monitorování systému, přes správu organizace (databáze zaměstnanců), automatického plánování směn dle zadaných parametrů, trackingu některých metrik důležitých pro Workforce Management, reportingu, či již zmíněného forecastingu (tvorby predikcí).

Tato komplexnost ale dělá práci s jeho taktickým plánováním méně uživatelsky přívětivou. Kupříkladu je možné pouze zvýšit celodenní objem, nikoliv jednotlivé hodiny, což v podstatě odstraňuje možnost taktického forecastingu v rámci dne. Srovnání s touto aplikací je tedy téměř nemožné, protože má mnou vytvořená aplikace naprosto jiné případy užití.

Další z nevýhod Verintu je kompatibilita, či spíše nekompatibilita. Verint pracuje pouze v prohlížeči Internet Explorer a k plnému využití aplikace je nutné mít nainstalovanou aktuální verzi Java Runtime Environment. Verint pracuje i v prohlížeči Mozilla Firefox, ale pouze v omezené míře, takže v tomto prohlížeči nepůjdou spustit reporty, či právě forecasting. Mozilla je v tomto ohledu vhodná pouze na prohlížení dat a jakékoliv úpravy nejsou možné.

Jako další z nevýhod bych uvedl jeho dostupnost pouze z IP adresy přidělené firmě Kiwi.com. Tohle by mnohými nemuselo být chápáno jako nevýhoda, neboť je možné tohle omezení obejít pomocí VPN, ale v dnešní době, kdy často zaměstnanci vyjíždí na pracovní cesty, je to komplikace, bránící, jakkoliv pracovat s tímto systémem, neboť VPN opět umožní pouze částečný přístup.



Obrázek 4.1 - Nástroj pro tvorbu taktických forecastů obsažený v aplikaci Verint

5 Specifikace požadavků

Aplikace nemá sloužit zcela jen pro tvoření běžné predikce výsledků, ale hlavně pro tvoření taktických předpovědí. Taktickou předpovědí uvažujeme v podstatě simulaci výsledku, pokud nastane nějaká nepředpokládaná událost, či se v daném období podstatně změní některý ze vstupních parametrů.

Jako příklad lze uvést například chřipkovou epidemii, která zásadně ovlivní absenci, tudíž i výsledné pokrytí hovorů. V aplikaci je možné tedy tento údaj při vypuknutí oné epidemie navýšit, čímž analytik získá přehled o tom, jak může například dalších 7 dní vypadat pokrytí směn a předat dále impuls k rychlé úpravě.

Dalším z případů, vycházejícím z provozu Kiwi.com, mohou být situace ovlivňující specifickou skupinu lidí, například sněhové bouře ve Spojených Státech Amerických ovlivňující provoz některých letišť. V takových situacích lze očekávat nárůst množství hovorů v časech odpovídajících nejrušnějším hodinám dne – mezi pátou hodinou odpolední a půlnocí středoevropského času, tedy během denních hodin v USA, během kterých létá velké množství letů.

Třetím ze základních případů užití je predikce výsledků při snížení obsazenosti. To může být využito kupříkladu při nutnosti proškolení agentů na změnu v procesu apod. Výstupem z takové analýzy může být opět impuls ke změně rozložení množství agentů v ovlivněných hodinách,

Čtvrtým a posledním zde zmíněných případů užití je jako zdroj dat při nastavování cílů produktivity agentů. Máme-li v databázi například reálné výsledky za poslední týden a k dispozici hodnotu počtu odbavených kontaktů za hodinu v průměru jedním agentem, můžeme na stejných datech zjistit jaké by bylo pokrytí a service level v případě nárůstu produktivity o 10 %.

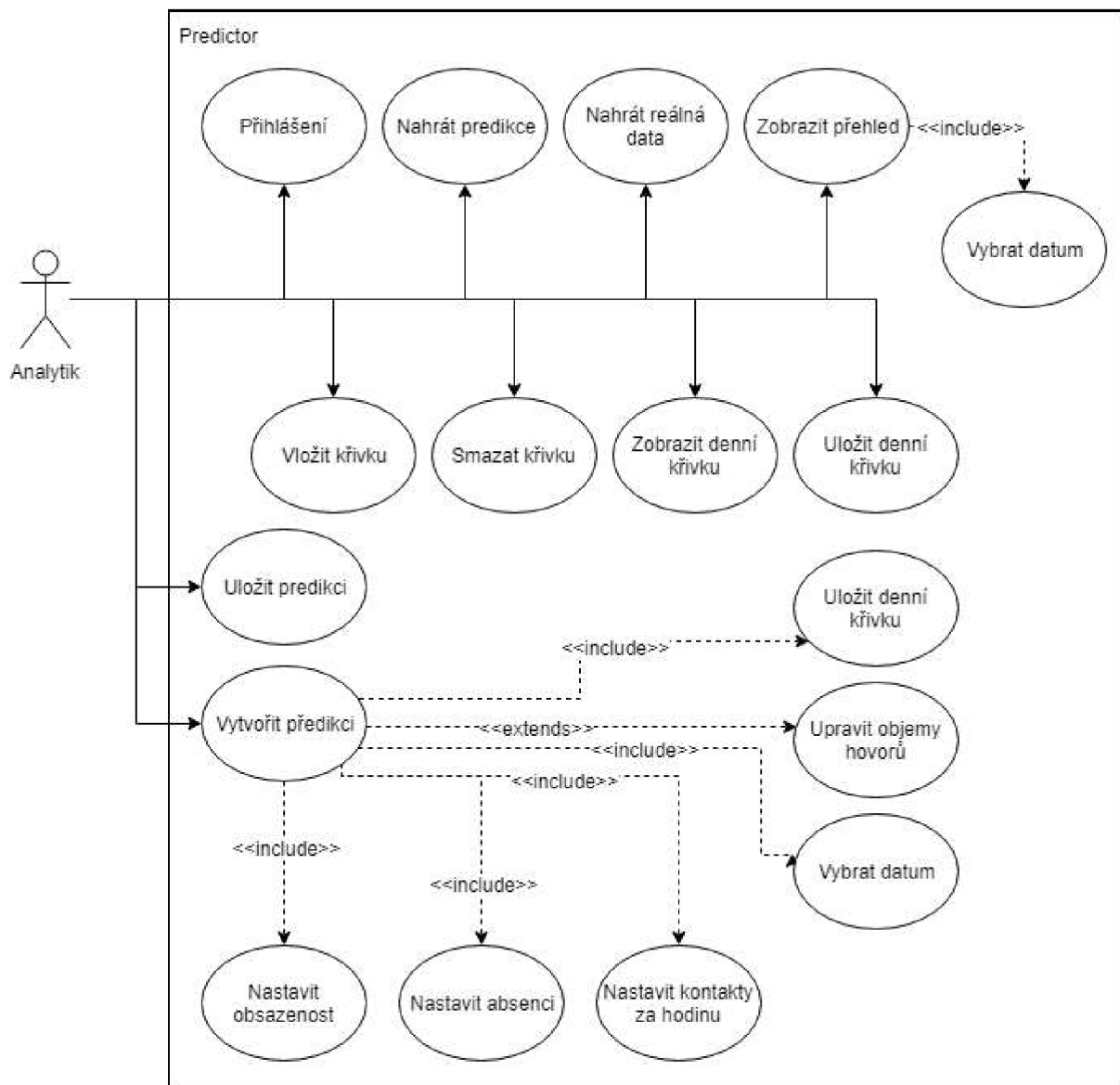
Obecné případy užití jsou popsány v obrázku 5.1.

5.1 Firemní požadavky

Na základně předchozích zkušeností firmy s nástroji pro tvorbu predikcí, byla jedním z hlavních požadavků jednoduchost používání, přívětivé UI a možnost nahrávat data pomocí CSV souborů ve formátu, který odpovídá formátu reportů, běžně používaných oddělením Workforce Management v Kiwi.com.

Dalším z požadavků bylo vytvoření dashboardu s přehlednou vizualizací uložené predikce a reálných dat s rozdělením po hodinách, stejně tak jako zobrazení pouze predikce. Zobrazení samostatné predikce slouží, jak již bylo zmíněno, k vytvoření reálných očekávání pro management.

V praxi to v Kiwi.com znamená, že je každou středu v týdnu prezentován výhled na další týden, rozdělený po hodinách, aby bylo možné identifikovat problematická místa (tzv. hotspots). Na stejné schůzi je také prezentováno srovnání predikce vytvořené pro předchozí týden s doplněnými reálnými daty.



Obrázek 5.1 - Diagram případů užití aplikace

6 Implementace aplikace

V této kapitole se budu zabývat samotnou implementací aplikace a popisem jednotlivých stránek a jejich funkcí.

6.1 Návrh databáze

Při návrhu databáze jsem se zaměřoval především na jednoduchost databáze, neboť to pro účely mé aplikace je nejvhodnějším řešením. Tabulky v databázi jsem původně měl tři. Tabulky *Hour*, *Day* a *Curve*. Později v průběhu samotné implementace aplikace jsem došel k závěru, že přítomnost tabulky *Day* s sebou přinášela pouze více komplexnosti jednotlivých kalkulací a vkládání dat. Tabulka *Day* obsahovala atributy popsané v tabulce 7.1.

Atribut	Popis
ID	Primární klíč s automatickou inkrementací
Date	Primární klíč, unikátní datum reprezentující jednotlivé dny
Forecast_volume	Celkové očekávané množství příchozích kontaktů
Forecast_sla	Celkový očekávaný <i>Service level</i>
Real_volume	Reálný objem kontaktů
Real_sla	Reálně dosažený <i>Service level</i>

Tabulka 6.1 Popis atributů tabulky Day

Jedním z důvodů, proč byla tabulka odstraněna je vkládání dat. Při vkládání dat do aplikace pomocí uploadu souboru je žádoucí i nahrávání dat pro jednotlivé hodiny. V aplikaci by to poté ale znamenalo, že při vložení, byť jen jednoho řádku do tabulky *Hour*, by bylo nutné přepočítat celkový počet kontaktů pro daný den (ať už očekávaných či reálných), ale také by v tu chvíli přestala být platná hodnota *Forecast SLA*, pokud by součástí nahrávaných dat byla i hodnota *Forecast volume*. To mě vedlo k zamyšlení, zda je tabulka opravdu nutná. Pro odstranění jsem se rozhodl na základě následujících bodů:

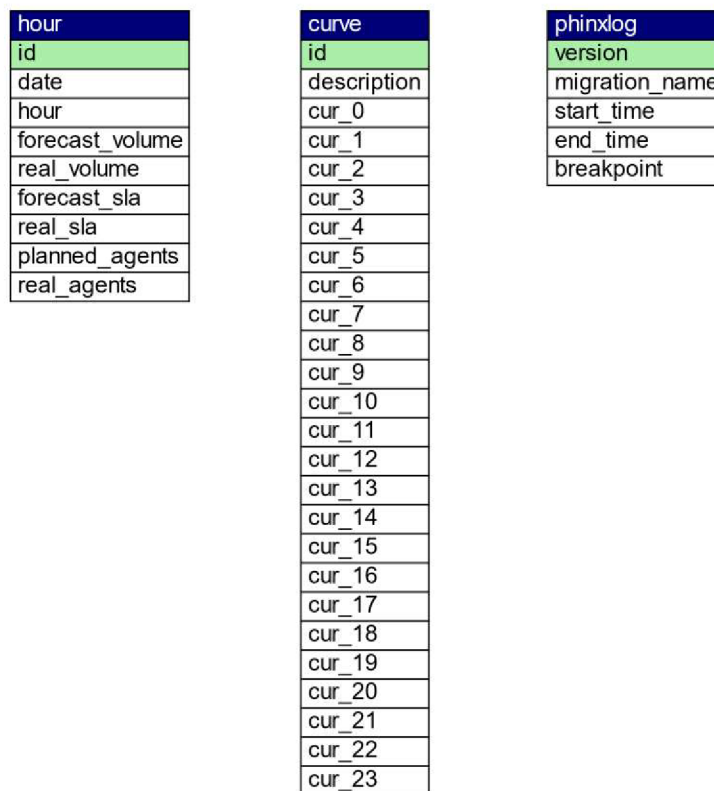
- Celkové součty objemů je možné snadno vypočítat až v momentě, kdy je potřeba je zobrazovat na front-endu jednoduchým sečtením všech řádků z tabulky *Hour* obsahujících shodnou hodnotu pro atribut *Date*.
- Celkové SLA, ať už předpokládané či reálné je také snadné vypočítat ze záznamů tabulky *Hour* pomocí výpočtu uvedeného v podkapitole 6.3.2.

Tabulka *Hour* obsahuje stejné atributy jako tabulka *Day*, zmíněná výše, obohacená atributem *Hour* obsahující číslo hodiny v daném dni, nabývající hodnot 0–23 a atributy *Planned_agents* a *Real_agents*. Atribut *Planned_agents* obsahuje počet agentů, kteří jsou v danou hodinu na směně, tedy

se očekává s jejich účastí. Atribut *Real_agents* udává reálný počet agentů, kteří v danou hodinu pracovali. Tento údaj je důležitý pro odůvodnění horších výsledků v případě vyšší absence na směně.

Tabulka *Curve* se skládá z primárního klíče *ID*, volitelného popisu dané křivky a dvaceti čtyřmi atributy pro jednotlivé hodiny, označené spojením řetězce „*Cur_*“ a čísla hodiny, podobně jako u tabulky *Hour*, tedy 0–23. Tyto atributy označují procentuální podíl kontaktů, který je v danou hodinu očekáván, zapsaný v podobě desetinného čísla. Součet těchto hodnot může překročit číslo 1, a to z důvodů simulace situací, kdy se může v daných hodinách zvýšit množství hovorů, například z důvodu vyššího množství hovorů z jiné časové zóny. V tu chvíli lze o součtu uvažovat jako o potenciálním násobku očekávaného množství hovorů.

Tabulka *phinxlog* je tabulkou automaticky vytvořenou nástrojem Phinx.



Obrázek 6.1 - ER diagram databáze

6.2 Rozvržení webu

Vzhled aplikace je, jak jsem již zmínil určen šablonou *Dashboard* z kolekce frameworku Bootstrap. Celá aplikace je rozdělena do čtyř stránek, nepočítáme-li přihlašovací stránku. Jednotlivé stránky jsou vytvořeny podle různých případů užití.

6.3 Popis jednotlivých stránek

V této kapitole jsou popsány veškeré stránky v aplikaci, jak z hlediska vzhledu, tak i implementace a popisu důležitých částí kódu.

6.3.1 Přihlášení

Při prvním zobrazení webové stránky je automaticky zobrazena přihlašovací stránka s formulářem. Bez přihlášení do systému samozřejmě nelze zobrazit žádnou jinou stránku. Tohoto je dosaženo vytvořením třídy *SecurePresenter*, která rozšiřuje třídu *Presenters*. Každý další presenter v aplikaci rozšiřuje *SecurePresenter*, takže při každém pokusu o otevření jakékoliv stránky dojde ke kontrole, zda je nějaký uživatel přihlášený. Toto probíhá pomocí následujícího kódu.

```
abstract class SecurePresenter extends Presenter
{
    /**
     * @throws \Nette\Application\AbortException
     */
    public function startup()
    {
        parent::startup();
        if (!$this->getUser()->isLoggedIn()) {
            $this->redirect('Sign:in');
        }
    }
}
```

K čemu tedy dochází v případě, že není žádný uživatel přihlášený je, že dojde k přesměrování na přihlašovací stránku.

Co se šablony stránky týče, u této stránky jsem ponechal vykreslení formuláře na rendereru bootstrapu, neboť obsahuje pouze dva elementy, u kterých není důležité umístění na stránce.

V aplikaci mám momentálně pouze uživatele *Admin*, neboť bude aplikace zprvu používána pouze lidmi, kteří mohou predikce tvořit. Po zakomponování do firemního systému, budou přístupy k jednotlivým stránkám řešeny pomocí rolí, které už v interním systému firmy jsou zavedené.

6.3.2 Homepage

Homepage je v podstatě hlavní stránkou, která bude v praxi prezentována. Na stránce se nachází 6 grafů implementovaných pomocí open source knihovny Charts.js, která je velice jednoduchá na používání a umožňuje například překreslování dle velikosti okna.

Z této knihovny jsem na stránce *Homepage* používal dva typy grafů. Sloupcový a spojnicový (čárový). V každém z grafů jsou jako sloupce uvedeny reálné hodnoty a jako čára potom hodnoty predikované.

První sloupec grafů obsahuje grafy pro konkrétní zvolený den (výchozí hodnotou je aktuální den) s daty zobrazenými po jednotlivých hodinách. Druhý sloupec pak přehledem týdne, v kterém se daný den nachází s celkovými výsledky pro daný den. Na stránce se nachází následující grafy.

- SLA Comparison – Day – tento graf zobrazuje srovnání Service Levels pro jednotlivé hodiny. Graf nabývá hodnot od 0 po 100.
- Volume Comparison – Day – zde jsou vidět rozdíly mezi objemy kontaktů. Tyto data mohou posloužit jako první zdroj informací, proč měla daná hodina jiný výsledek, než bylo očekáváno. Kupříkladu v případě neobvykle vysokého objemu hovorů pro danou hodinu je důvod horšího SLA jasný. Graf má minimum v nule a maximum automaticky určené knihovnou Charts.js dle zobrazených hodnot.
- Staffing Comparison – Day – tento graf podobně jako graf předchozí slouží k zjištění deviací od očekávaného SLA. Na tomto grafu můžeme detekovat vysokou neúčast, ať už způsobenou vyšší neplánovanou absencí, nebo například schválením dovolené nad rámec povoleného procenta. Graf má minimum i maximum určené knihovnou, dle obsažených hodnot.
- SLA Comparison – Week – totožné jako SLA Comparison – Day pouze s agregací po dni
- Volume Comparison – Week – totožné jako Volume Comparison – Day s agregací po dni
- Staffing Comparison – Week – tento graf se poněkud liší, neboť obsahuje součet počtů agentů z jednotlivých hodin, což ovšem neznamená, že tolik agentů bylo v daný den na směně. Graf slouží spíše k vytvoření obrazu o tom, jak velký byl rozdíl mezi naplánovanými agenty a skutečnou účastí a jde spíše o pozorování trendu absence.

Jak jsem již zmínil v kapitole o návrhu databáze, data pro jednotlivé dny nemají vlastní tabulku s celkovými výsledky, ale jsou vypočítávány na back-endu při odeslání formuláře. Celodenní SLA je počítáno následujícím vzorcem. Volume i SLA je buď predikované, nebo skutečné, dle toho, jaký údaj počítáme.

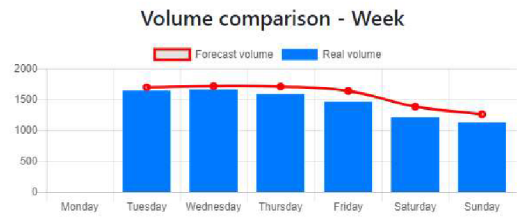
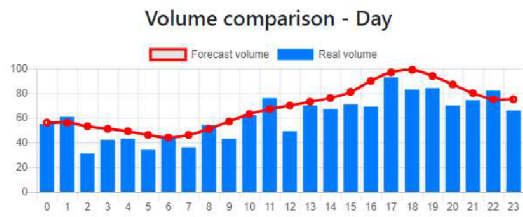
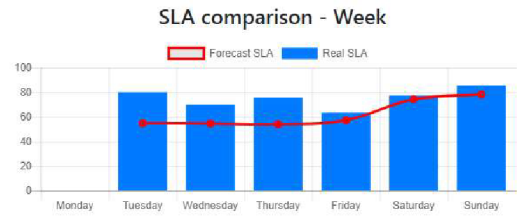
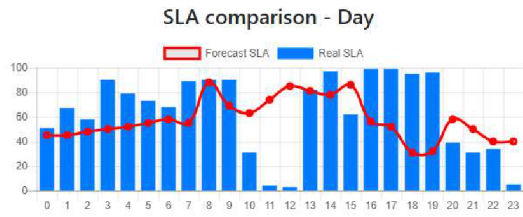
$$\sum_{i=0}^{23} SLA_i \times Volume_i$$

V případě chybějících dat v databázi není pro danou hodinu, či den zobrazena žádná hodnota chybějící metriky (viz. obrázek 6.2).

Homepage

04/05/2018

Load values



Obrázek 6.2 - Náhled úvodní stránky aplikace

6.3.3 Predictor

Predictor je nejkompexnější stránkou celé aplikace, na které probíhají veškeré výpočty. Po načtení stránky jsou implicitně zobrazena data pro aktuální den. Formulář na stránce obsahuje elementy popsané v následující tabulce.

Název elementu	Popis
Date	Datum, pro které budou predikce vytvářeny
Occupancy(Obsazenost)	Průměrná procentuální přítomnost agenta u počítače. Viz. kapitola 2.2.7
Absence	Procentuální absence agentů pro daný den. Viz. kapitola 2.2.4
Cases per hour	Aktuální produktivita agentů na příchozích kontaktech. Tato hodnota by měla být v době tvorby predikce nastavena z co nejaktuálnějších dat. Viz. kapitola 2.2.8
Curve description	Toto pole slouží k popsání křivky, kterou uživatel nastaví před tvorbou predikce pomocí range sliderů. Nutné zadávat pouze pokud chce uživatel danou křivku uložit do databáze.
Load values	Po kliknutí na tlačítko jsou načteny hodnoty pro zvolený den
Calculate	Tlačítko sloužící k provedení kalkulace predikce na základě zadaných hodnot
Defaults	Vrátí hodnoty sliderů na původní hodnoty pro daný den
Save forecast	Tlačítko sloužící k uložení vypočítané predikce
Save curve	Tlačítko sloužící k uložení křivky nastavené pomocí sliderů

Tabulka 6.2 - Popis elementů na stránce

Na stránce se také nachází 24 sliderů, kde každý z nich je nastaven na hodnotu očekávaného množství příchozích hovorů pro danou hodinu. Slidery jsou vytvořeny za pomoci knihovny *rangeslider.js*, která přidává mnoho druhů sliderů s nastavitelnými parametry jako například nastavení orientace na vertikální, což HTML element `<input type="range">` neumožňuje.

Na stránce je opět zobrazen graf z knihovny *Charts.js*, který zobrazuje vypočítané, či z databáze načtené predikované SLA. Slouží jako vizuální reprezentace predikce a umožňuje uživateli snadno získat přehled o potenciální situaci.

Predikce výsledků, jak jsem již zmínil v předchozích kapitolách, nenásleduje konvenční výpočty. Pro výpočet neuvažuji běžnou dobu obsluhy, ale využívám údaje *Kontakty za hodinu*, který je v rámci firmy Kiwi.com hodnotou reprezentující celkovou agentovu produktivitu. Pro účely aplikace používáme pouze počet příchozích odbavených kontaktů za hodinu, neboť nepřímo zahrnuje i odchozí kontakty, kterých se v rámci Kiwi.com uskutečňuje mnoho.

V případě, že bych použil celkové množství kontaktů za hodinu, by mohlo dojít k vytvoření nereálného očekávání pro management, neboť by zahrnovalo hodnotu odchozích kontaktů, což bývá v průměru cca 1 za hodinu. Samotný výpočet tedy využívá následujícího vzorce.

$$SLA_{hour} = \frac{Agents \times (1 - Absence) \times Occupancy \times Cases_per_hour}{Forecast_volume}$$

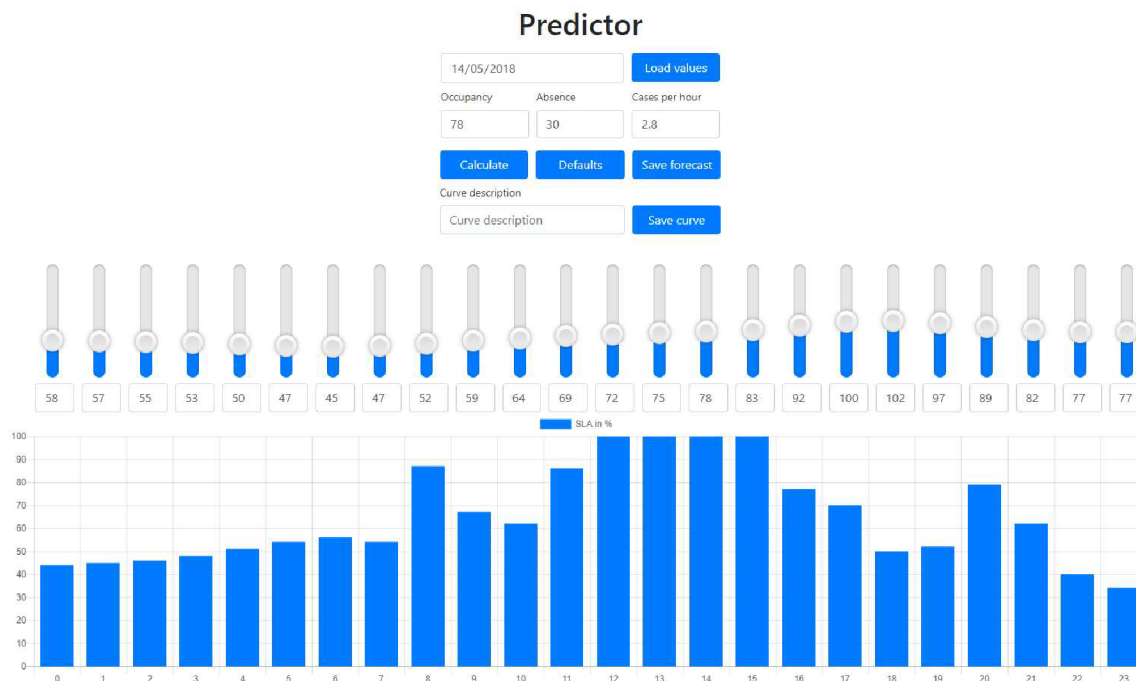
Popis výrazů ve vzorci:

- Agents – počet agentů naplánovaných na danou hodinu, hodnota dotazem vybraná z databáze
- Absence – procentuální předpokládaná absence, zadaná z formuláře na stránce Predictor
- Occupancy – obsazenost (viz. podkapitola 2.2.5) zadaná z formuláře na stránce Predictor
- Cases per hour – počet příchozích kontaktů odbavených agentem za hodinu, zadaná jako průměrná hodnota ve formuláři na stránce Predictor
- Forecast volume – očekávaný počet příchozích kontaktů

Čítel tohoto vzorce vypočítává, kolik příchozích kontaktů je daný počet agentů schopný odbavit v závislosti na nastavené parametry. Toto číslo může být vyšší než samotné očekávané množství hovorů pro daný interval, což v realitě značí, že je v danou hodinu přítomno více agentů, než je na dané množství hovorů potřeba. Po vydělení čitatele jmenovatelem je jakýkoliv výsledek vyšší než 1 zaokrouhlen na 1.

Tento výpočet je v kódu prováděn funkcí *calculatePrediction* a poté při ukládání funkcí *savePrediction*. Obě funkce jako parametr dostávají element typu *Nette\Forms\Controls\SubmitButton*.

Při vytváření predikce pro konkrétní situaci, kdy je očekáváno zvýšení množství kontaktů v daném intervalu, je možné po nastavení sliderů provést uložení podílů kontaktů jednotlivých hodin jako křivky se zvoleným popisem pro použití v podobných situacích, či při vkládání nové predikce objemu.



Obrázek 6.3 - Vzhled stránky Predictor, obsahující nástroje pro vytváření predikcí

6.3.4 Upload data

Tato stránka slouží pro vkládání očekávaného množství hovorů, na základě uvedení celkového objemu pro daný den a zvolení jedné z denních křivek, uložených v databázi. Zvolenou křivku je možné zobrazit v grafu, popřípadě ji vymazat, pokud již není aktuální.

Po kliknutí na tlačítko Insert, či Update, je celková hodnota očekávaného objemu hovorů rozpočítána do jednotlivých hodin, vynásobením celkového objemu hodnotou denní křivky odpovídající vypočítávané hodině.

Druhou funkcí této stránky je vkládání nových denních křivek. Jednotlivé podíly objemu hovorů lze vepsat do formuláře na konci stránky. Podíly musí být zapisovány v procentech ve formátu desetinného čísla tak, že jednotka zadaného čísla je jednotka procenta, tedy například po zadání 8.3 bude hodnota aplikací interpretována jako 8.3 %.

Po uložení nové křivky je daná křivka zobrazena v již zmíněném grafu pro kontrolu podílů. Pokud uživatel udělal při zápisu chybu, je možné křivku hned odstranit a vložit křivku novou.

Upload data

Date: Forecast volume:

Intraday curves:

Intraday curve

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

Curve creation

Please enter the values in percent, e.g. 8.5 will be 8.5%

Curve description:

Obrázek 6.4 - Vzhled stránky Upload data

6.3.5 Upload CSV files

Stránka *Upload CSV files* obsahuje jednoduchý formulář pro nahrání souboru CSV. Součástí formuláře je také přepínač určující, zda se jedná o soubor s predikcí (objemu, SLA, agentů), či soubor s reálnými daty.

Po zvolení souboru a odeslání formuláře je soubor nahrán do dočasné paměti, ze které je potom otevřen pomocí funkce *fopen*, s parametrem *\$values->getTemporaryFile()* a parametrem „r“ určujícím, že se bude ze souboru pouze číst, nikoliv do něj zapisovat. Otevřený soubor je následně čten řádek po řádku s tím, že první řádek s nadpisy sloupců je ignorován. Nahrávaný soubor musí mít strukturu popsanou v tabulce 6.3.

date	hour	volume	sla	agents
6/30/2018	4	150	45	15
6/29/2018	3	100	63	12

Tabulka 6.3 - Příklad struktury souboru CSV

V tabulce 6.3 neuvádím výrazy *forecast*, či *real*, které by ale u sloupců *volume*, *sla* a *agents* měly být, v závislosti na to, zda se jedná o soubor s predikcí či reálnými daty. Soubor nemusí nutně obsahovat 24 záznamů pro daný den, aplikace umožňuje upload libovolného počtu hodin z daného dne, pro účely korekce dat.

V případě, že dojde při procesování nějakého z řádků souboru k chybě, zůstávají všechny předchozí řádky již uloženy v databázi a operace se přeruší s chybovou hláškou. K chybám může dojít například při neuvedení čísla hodiny, nebo data. V případě chybějícího objemu hovorů, SLA, či počtu agentů, je do databáze uložena hodnota *null*. Toto chování je chtěné především z důvodu nahrávání samostatné predikce objemu, či počtu agentů. Je důležité si ale uvědomit, že při uvedení například jen počtu agentů, dojde k přepsání dat ve sloupcích s objemem i SLA, tudíž lze dělat korekce celých záznamů, nikoliv pouze jednoho sloupce.

Upload files

forecast No file chosen
 real data

Please note, that the CSV file has to follow this format.

Date	Hour	Volume	SLA	Agents
mm/dd/yyyy	X	Y	Z	A
Date in month/date/year format	Number of hour from 0 to 23	Either forecast or real volume, based on the selected button	Either forecast or real SLA, based on the selected button	Number of planned or real agents in said hour, planned = forecast
Required	Required	Optional - empty value stored in database if empty	Optional - empty value stored in database if empty	Optional - empty value stored in database if empty

Obrázek 6.5 - Náhled stránky Upload CSV files

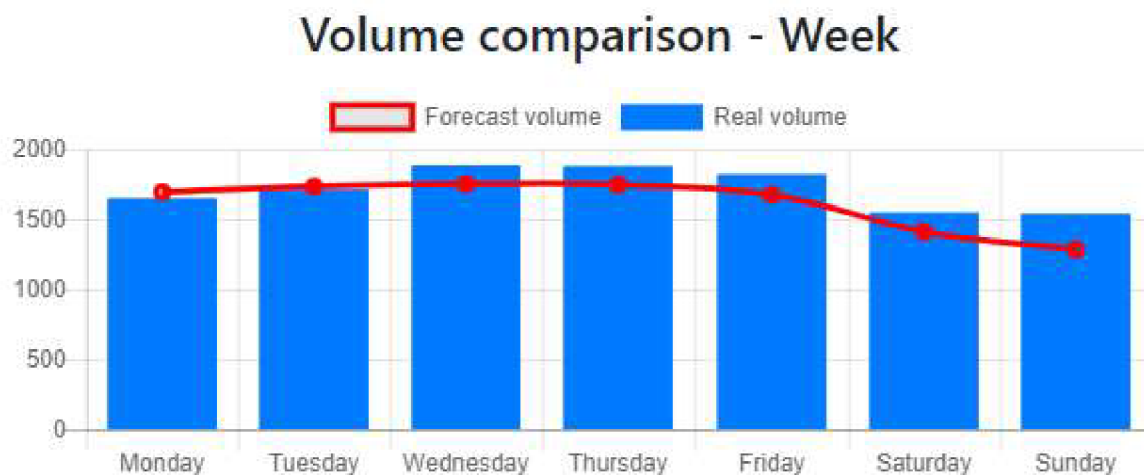
7 Testování

Pro testování jsem použil data z jednoho měsíce provozu Kiwi.com v roce 2017. V daném vzorku se vyskytují jak dny, kdy nenastaly žádné větší problémy, tak dny, kde bylo množství příchozích hovorů ovlivněno například hurikány v Asii. V následujících podkapitolách budou shrnuty výsledky, kterých jsem použitím aplikace dosáhl a budou srovnány s reálnými daty z daného časového období.

Data ze zvoleného měsíce byly namapovány na měsíc květen tohoto roku, aby nebyla zveřejněna konkrétní data pro konkrétní dny z provozu firmy a zároveň byla data rychleji dostupná při práci a předvádění výsledné aplikace.

7.1 Běžný provoz

Jako vzorek dat pro běžný provoz bez výrazných výkyvů byly zvoleny dny od 21. května do 27. května. Během těchto dnů nedocházelo k výkyvům v celkovém množství příchozích hovorů, viz. obrázek 7.1.



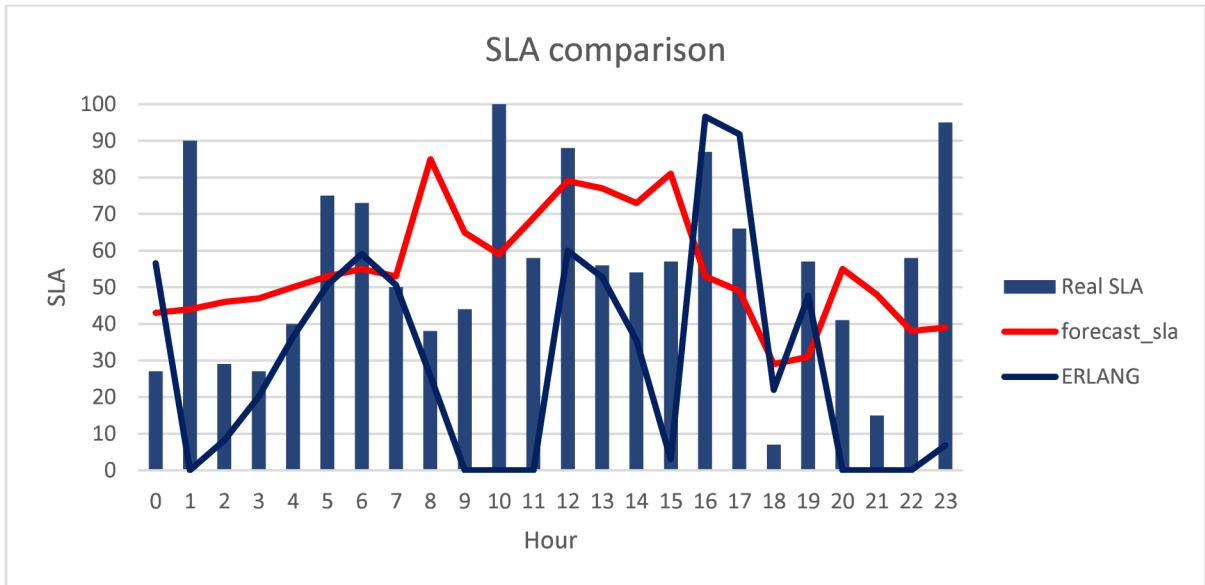
Obrázek 7.1 - Srovnání předpokládaného a reálného množství hovorů v období 21. - 27. května

Ve zvoleném období se pohybovala produktivita mezi 3,2 a 3,3 příchozími kontakty za hodinu, obsazenost byla kolem 80 % a absence kolem 10 %. Vytvořením predikce se zvolenými hodnotami jsem se dostal velice blízko k reálným hodnotám daného týdne.

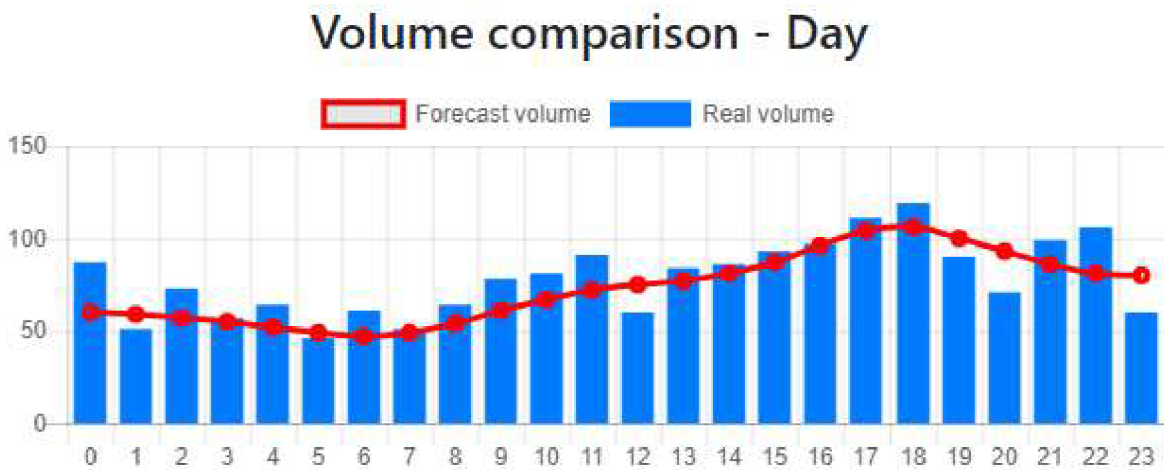
Průměrná odchylka od predikce byla pro daný týden 8,88 %, a to především kvůli datu 27. května. Toho dne přišlo o 19,8 % více kontaktů a byla i zvýšená absence. V tento den jsem testováním se stejnými hodnotami dosáhl odchylky 21,88 %, ale je nutné vzít v úvahu ono zvýšené množství hovorů. Bez započtení 27. května byla průměrná odchylka 6,7 %.

Algoritmus Erlang dosáhl obecně větší odchylky, a to v průměru 17,15 %. Tím, že výpočet Erlang vytváří Erlang z průměrné doby obsluhy jednoho kontaktu, jsem tento údaj musel zpětně dopočítat. Vezmu-li v úvahu, že v té době se pohyboval počet odchozích kontaktů za hodinu kolem 1,2, dostal jsem se na průměrnou dobu obsluhy 13,3 minut. Počet agentů jsem stejně jako u mé kalkulace vynásobil průměrnou účastí a poté obsazeností. U algoritmu Erlang je velkou nevýhodou, že v mnoha

případech je jeho výsledek při výpočtu SLA buď velice blízko nule, nebo naopak blízko sto procentům. Srovnání hodinového SLA vypočítaného mým algoritmem a algoritmem Erlang je zobrazeno na obrázku 7.2. Data v něm jsou z 24. května. Je z něj jasně čitelné, že algoritmus Erlang často fluktuuje, kdežto můj algoritmus, ač výpočtem jednodušší, má mnohem konzistentnější hodnoty. Pro srovnání je v obrázku 7.3 ukázán i rozdíl očekávaného a reálného objemu příchozích hovorů.



Obrázek 7.2 - Srovnání SLA vypočítaných oběma algoritmy s SLA reálným z 24. května



Obrázek 7.3 - Srovnání očekávaného a reálného příchozího množství hovorů z 24. května

7.2 Nárůst absence

Pro testování aplikace pro simulaci nárůstu absence jsem si zvolil data za 31. květen. Tento den jsem si zvolil především kvůli faktu, že během odpoledních hodin v práci skutečně bylo v průměru o jednu třetinu méně agentů, než bylo očekáváno. Vzhledem k tomu, že výpočet jednotlivých SLA bere v potaz pouze data z dané hodiny, je možné testovat nárůst absence v některých hodinách, nastavením vyšší absence pro celý den a soustředit se při tom pouze na požadované hodiny.

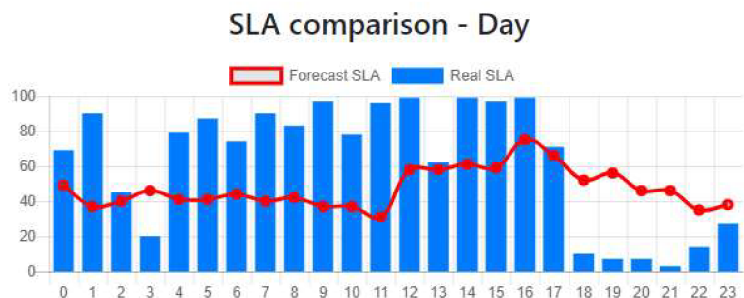


Obrázek 7.4 - Data zobrazující rozdíl v počtu agentů pro den 31. května

Jak je vidět v obrázku 7.4, výše zmíněný nárůst v absenci začal od 12:00, kdy místo 36 agentů na směně bylo pouze agentů 25. To činí absenci o hodnotě 30,6 %, což je nárůst v absenci o 20 % oproti standartnímu limitu absence, který je v Kiwi.com nastaven na 10 %. Po zjištění takovýchto údajů jsou běžně zkoumané důvody této absence, aby bylo jasné, zda se jednalo o absenci jednodenní, nebo se jedná o trend. Pokud by se jednalo o trend, byla by v danou situaci využita tato aplikace pro přepočítání očekávaného SLA pro další dny. Pro otestování aplikace jsem tedy nastavil 30 % absenci. V obrázcích 7.5 a 7.6 je možné vidět srovnání predikovaného SLA při absenci 10 % a při absenci 30 %.



Obrázek 7.5 - Predikované SLA při absenci 10 %



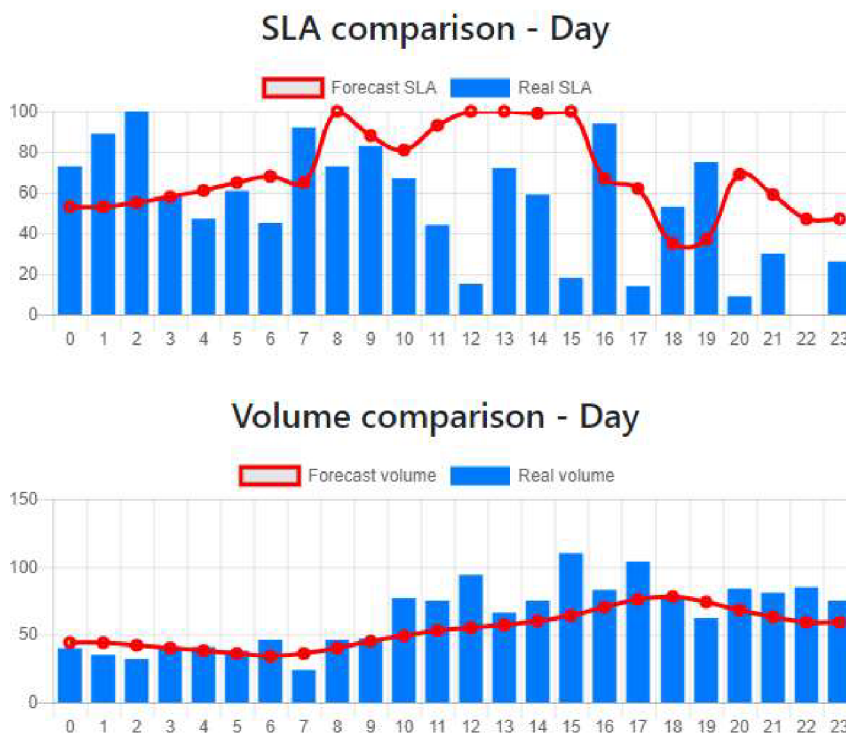
Obrázek 7.6 - Predikované SLA při absenci 30 %

Jak je z obrázků vidět, zvýšení absence o 20 % snížilo predikované SLA o zhruba 30 %, ale rozdíl s reálným SLA je stále velký. Důvodem může být fakt, že při menším počtu lidí je mnohem důležitější následování předem předepsaných pauz jednotlivými agenty a časy příchodů jednotlivých kontaktů. Aplikace předpokládá rovnoměrné rozložení kontaktů v hodině, což v praxi nebývá pravdou. Pro tento konkrétní den byl faktorem i nárůst počtu kontaktů právě v hodinách se zvýšenou absencí, kdy se počet příchozích kontaktů zvýšil v průměru o 14 % (o 10 až 20 kontaktů v jednotlivých hodinách). V momentě, kdy takováto situace nastane, se často vytvoří delší fronta čekajících kontaktů, což při sníženém počtu agentů znamená, že agenti obsluhují nejdéle čekající kontakty (fronta FIFO), takže nemohou přijímat nově příchozí kontakty v čase stanoveném SLA. Proto bývá pokles v hodinovém SLA dosti markantní.

Je nutné podotknout, že taktické predikce nemají za cíl se co nejpřesněji přiblížit k reálnému SLA, ale spíše určit rozmezí nově vytvořené predikce, či procentuální rozdíl mezi novou a původní predikcí. Výsledkem taktické predikce tedy může být například rozhodnutí o povolení přesčasů pro problematický časový interval, což by v tomto případě byly odpolední a večerní hodiny.

7.3 Zvýšení množství hovorů v intervalu

V případech, kdy lze s předstihem očekávat zvýšené množství hovorů je vhodné provést predikci znovu a nastavit zvýšení dle aktuálních dat, například pomocí koeficientu, kterým pronásobíme některé hodiny dne. Zvýšení kontaktů je obtížné testovat s konkrétním výsledkem, neboť v případě, že taková situace nastane, je linka okamžitě posílena zdroji z jiné fronty. Proto se stává, že pokud po hodině s nadměrným objemem hovorů tento objem klesne, linka dosáhne dobrého výsledku, protože je v tu chvíli na nižší množství kontaktů více lidí. Tento jev je dobře vidět na obrázku 7.7.

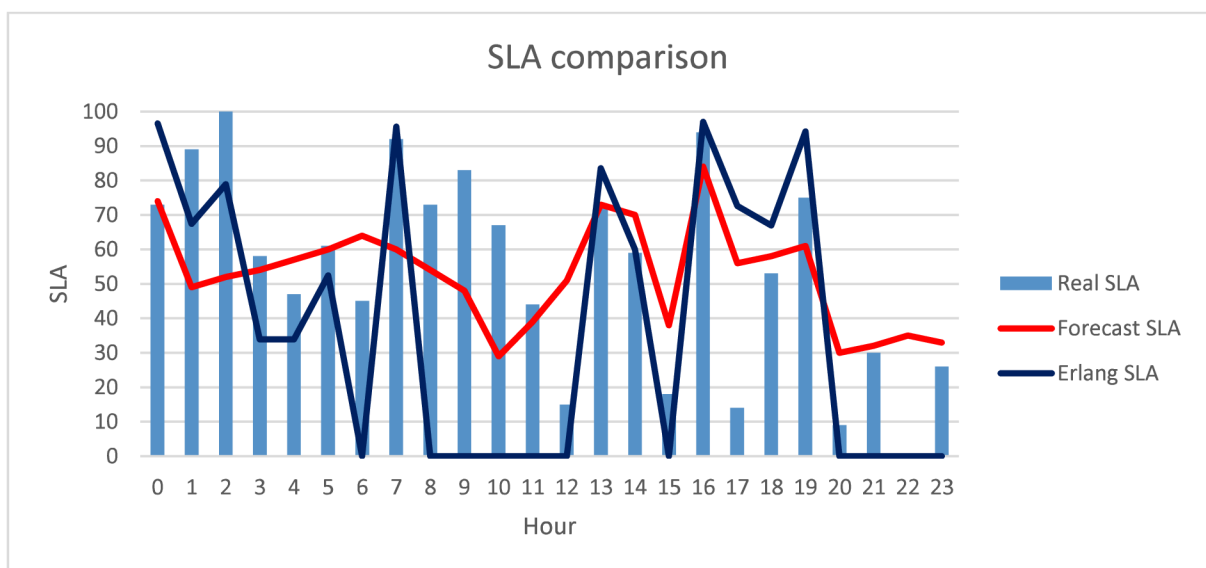


Obrázek 7.7 - Přehled objemu a SLA pro 27. května

Pro testování jsem zvolil stejný den jako v obrázku, 27. května. Je u něj snadno vidět výše zmíněný jen, především u hodin 13, 14, 16, 18 a 19. Po upravení hodnot na reálné hodnoty z tohoto dne se křivka potenciálního SLA posunula blíže k reálným hodnotám z tohoto dne. Výsledek této predikce je vidět na obrázku 7.8.



Obrázek 7.8 - Hodnoty z 27. května po zvýšení hodinového objemu



Obrázek 7.9 - Srovnání výsledků algoritmu Erlang a výsledku predikce pomocí algoritmu použitého v aplikaci

Při opětovném srovnání výsledků algoritmu Erlang a algoritmu použitého v aplikaci, lze snadno vidět, že hodnota mého algoritmu se hodnotě reálné blíží více, neboť algoritmus Erlang opět fluktuuje mezi extrémy. Budeme-li mluvit v číslech, můj algoritmus se v průměru lišil o 19 %, oproti Erlangu, který se v průměru lišil o 26 %.

Jak jsem již zmínil dříve v této podkapitole, aproximovat výsledek hodiny při větším nárůstu a stejném (či menším) počtu agentů je velice obtížné, protože sebemenší vlna příchozích kontaktů může vytvořit větší frontu, které se zbavuje v horizontu hodiny velice těžko.

7.4 Shrnutí testování

Krom výše zmíněných dat jsem zkoumal ještě průměrnou hodinovou odchylku za celý zkoumaný měsíc, ve kterém jsem dosáhl následujících výsledků.

Algoritmus	Průměrná odchylka za 31 dní
Erlang	35,91 %
Algoritmus aplikace	27,12 %

Tabulka 7.1 - Srovnání celkové odchylky obou algoritmů

Porovnáním obou výsledků jsem se dostal k závěru, že je mnou vytvořený algoritmus méně náchylný k výkyvům a dosahuje celkově menší odchylky, což je výhodnější za klidnějších situací call centra. V případech kdy nastane velký výkyv v objemu hovorů na něj Erlang reaguje větším poklesem v hodinovém SLA.

Data zvoleného měsíce byla charakteristická velkým počtem výkyvů, což mi dovolilo otestovat různé situace oběma algoritmy, byť u některých dnů byly výkyvy jak v objemu hovorů, tak v počtu agentů, kteří byli reálně přítomni na směně, což komplikovalo testování nutností zaměřovat se na více faktorů zároveň.

To vše má za důsledek celkově větší odchylku obou algoritmů. I přes to ovšem považuji testování za úspěšné, protože se na testovaných situacích vždy po nastavení hodnot dle reálného objemu a agentů povedlo, mnou vytvořeným algoritmem, přiblížit křivce reálných SLA.

8 Závěr

Výsledkem této bakalářské práce je webová aplikace sloužící pro tvorbu přesných i taktických predikcí v prostředí call center. Aplikace byla vytvářena tak, aby odstranila problémy, které má aktuální způsob tvorby těchto predikcí ve firmě Kiwi.com. Aplikace obsahuje grafy s přehledným srovnáním třech nejdůležitějších faktorů pro pokrytí příchozích kontaktů. Předpokládaný versus reálný objem příchozích hovorů, naplánovaný versus reálný počet agentů pro dané hodiny a nejdůležitější srovnání predikovaného a reálného SLA.

Pouhým pohledem na tyto grafy lze snadno odhalit důvody výsledného SLA. Grafy jsou srovnány pod sebou, takže jsou veškerá relevantní data pro danou hodinu pod sebou, přispívající k lepší čitelnosti. Přehled také obsahuje týdenní srovnání stejných metrik, agregované po dnech. To je užitečné pro celkovou evaluaci pokrytí, především vyšším managementem, který je zaměřený více na denní než hodinové výsledky.

Testováním taktických predikcí jsem dospěl k závěru, že algoritmus funguje, neboť se i v extrémních případech přiblížit reálným výsledkům, tedy je v těchto situacích vhodným nástrojem pro evaluaci situace a předání výsledků managementu, aby se na základě těchto dat mohlo udělat potřebné rozhodnutí pro potenciální předejití špatné situaci, a tím i nespokojenosti zákazníků.

Aplikace byla ve firmě testována oddělením Workforce Management a momentálně je v jednání možné datum integrace této aplikace do interního systému a její rozšíření o více typů uživatelů a rolí. Aplikace byla přijata pozitivně, neboť odstraňuje nedostatky aktuálního tabulkového řešení. Hlavním zlepšením je rychlost práce s daty, neboť je aplikace ovládána z větší míry pouze myší a není tedy potřeba velké množství hodnot kopírovat, či dopisovat do různých polí, jako tomu je v aktuálním řešení.

Nahrávání dat do aplikace je implementováno nahráním souboru CSV, který je na straně aplikace zpracován a data vložena do databáze. Struktura onoho CSV souboru byla nastavena tak, aby odpovídala reportu ze systému Verint, který obsahuje jak predikci počtu hovorů, tak počty lidí pro jednotlivé intervaly.

Celkově považuji aplikaci za úspěch, neboť splňuje veškeré požadavky, které pro ni byly před zahájením implementace vytvořeny.

Literatura

- [1] Kiwi.com: O nás. *Kiwi.com* [online]. Brno, 2018 [cit. 2018-05-10]. Dostupné z: <https://www.kiwi.com/cz/pages/content/about>
- [2] Techopedia Inc.: Interactive Voice Response (IVR). *Techopedia Inc.* [online]. 2018 [cit. 2018-05-10]. Dostupné z: <https://www.techopedia.com/definition/1525/interactive-voice-response-ivr>
- [3] VoIP-Info.org. *Automatic Call Distributor* [online]. 2018 [cit. 2018-05-10]. Dostupné z: <https://www.voip-info.org/automatic-call-distributor/>
- [4] WampServer. *A Windows web development environment* [online]. [cit. 2018-05-08]. Dostupné z: <http://www.wampserver.com/en/>
- [5] Tutorials, W. O. W. *HTML Introduction* [online]. [cit. 2018-04-08]. Dostupné z: https://www.w3schools.com/html/html_intro.asp
- [6] Tutorials, W. O. W. *CSS Introduction* [online]. [cit. 2018-04-08]. Dostupné z: https://www.w3schools.com/css/css_intro.asp
- [7] PHP Group. *What is PHP* [online]. [cit. 2018-03-17]. Dostupné z: <http://php.net/manual/en/intro-what-is.php>
- [8] ISquare Technologies. *Benefits of PHP in Web Development* [online]. [cit. 2018-03-17]. Dostupné z: <https://www.isquaretechnologies.com/isquare-blog/benefits-php-web-development/>
- [9] JetBrains s.r.o. *PhpStorm - Lightning-smart PHP IDE* [online]. [cit. 2018-03-17]. Dostupné z: <https://www.jetbrains.com/phpstorm/specials/phpstorm/phpstorm.html>
- [10] Nette Foundation. *Seznámení s Nette Frameworkem* [online]. [cit. 2018-03-20]. Dostupné z: <https://doc.nette.org/cs/2.4/getting-started>
- [11] Coderseye.com. *11 Best PHP Frameworks for Modern Web Developers in 2018* [online]. [cit. 2018-03-19]. Dostupné z: <https://coderseye.com/best-php-frameworks-for-web-developers/>
- [12] Nette Foundation. *Tracy* [online]. [cit. 2018-03-20]. Dostupné z: <https://tracy.nette.org/>
- [13] PhpMyAdmin. *Bringing MySQL to the web* [online]. [cit. 2018-04-01]. Dostupné z: <https://www.phpmyadmin.net/>
- [14] CakePHP. *Phinx Migrations* [online]. [cit. 2018-03-24]. Dostupné z: <https://book.cakephp.org/3.0/en/phinx.html>
- [15] *Introduction to Bootstrap* [online]. [cit. 2018-03-25]. Dostupné z: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

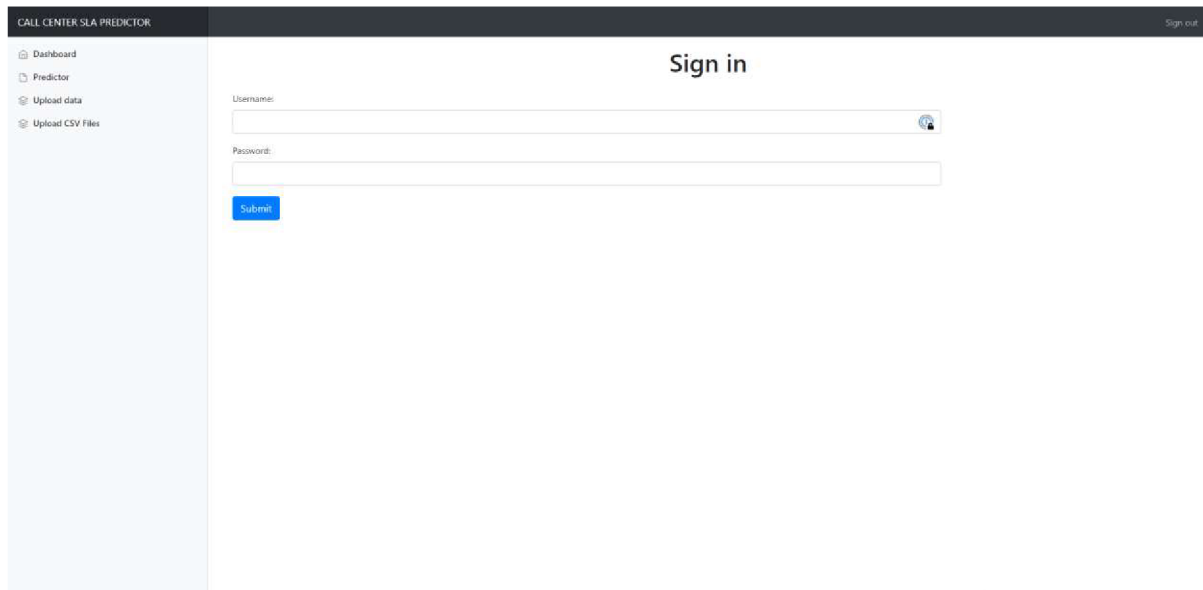
- [16] *Bootstrap - Dashboard layout* [online]. [cit. 2018-03-25]. Dostupné z: <https://getbootstrap.com/docs/4.1/examples/dashboard/>
- [17] Westbay Engineers Limited. *What is Erlang* [online]. [cit. 2018-03-25]. Dostupné z: <http://www.erlang.com/whatis.html>
- [18] Erlang: *The foremost site for call center forecasting tools*. Dostupné na: <http://www.erlang.co.uk/default.htm>
- [19] Verint Systems Inc. *Verint® Branch Workforce Management™ Solution Brief* [online]. [cit. 2018-04-20]. Dostupné z: <https://www.verint.com/Assets/resources/resource-types/brief/branch-workforce-management-solution-brief-english-us.pdf>

Přílohy

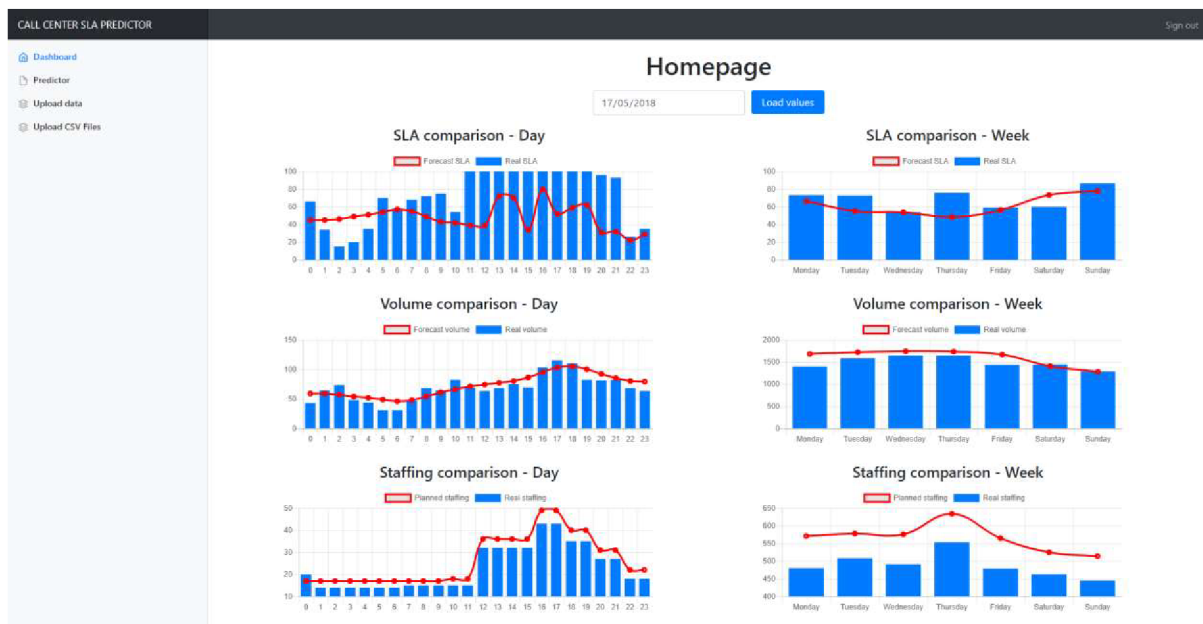
Příloha A – Obsah CD

- /Predictor
 - /app
 - /config
 - config.local.neon – lokální nastavení konfiguračního souboru
 - config.neon – nastavení konfiguračního souboru
 - /presenters – zdrojové soubory .php
 - /templates – zdrojové soubory .latte
 - HomepagePresenter.php
 - PredictorPresenter.php
 - SecurePresenter.php
 - SignInPresenter.php
 - UploadFilePresenter.php
 - UploadPresenter.php
 - bootstrap.php
 - /db/migrations – soubory s Phinx migracemi
 - Readme.txt – readme se strukturou složek

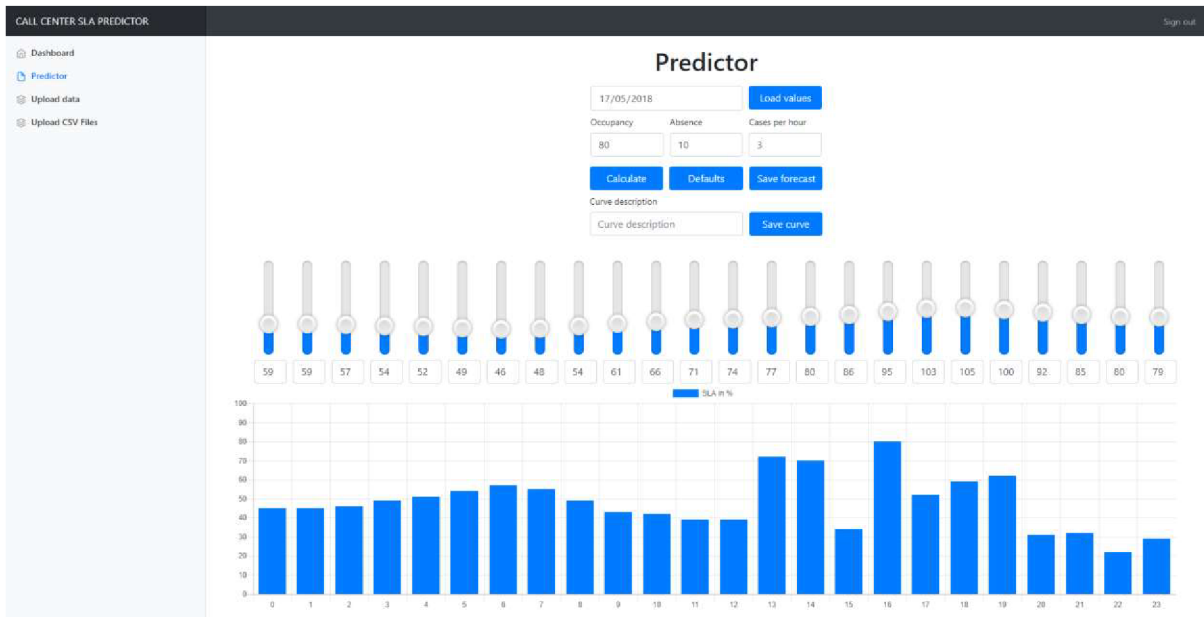
Příloha B – Ukázky aplikace



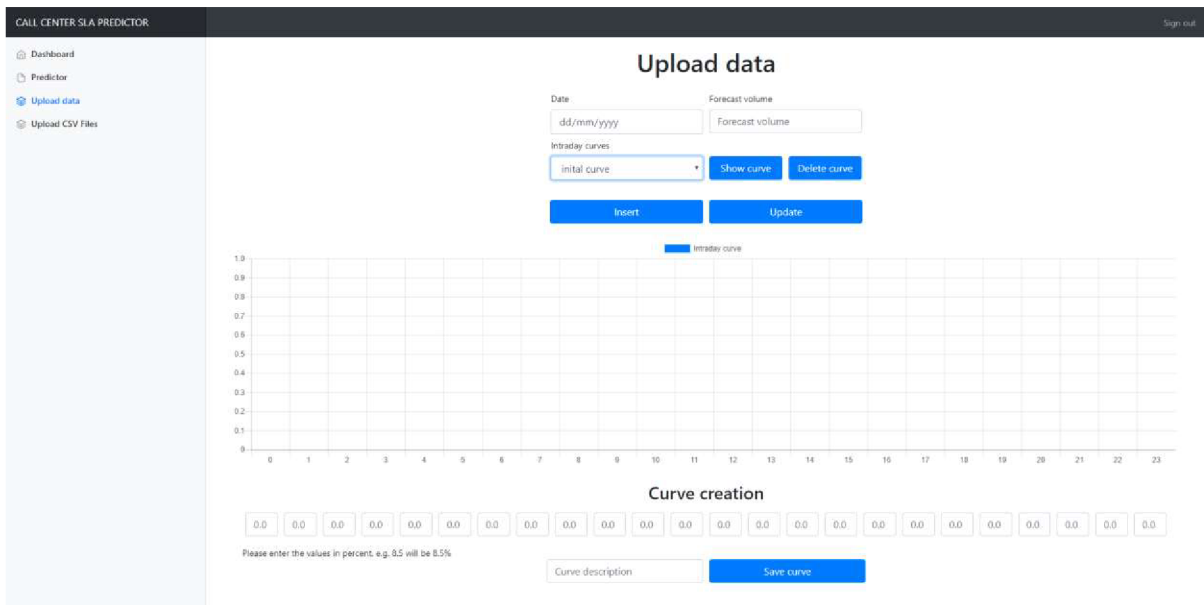
Obrázek 8.1 - Přihlašovací stránka



Obrázek 8.2 - Úvodní stránka s přehledem dat



Obrázek 8.3 - Stránka s nástrojem pro tvorbu predikcí



Obrázek 8.4 - Nahrávání dat pomocí denních křivek s možností vytvoření či smazání křivek

CALL CENTER SLA PREDIKTOR Sign out

Dashboard
 Predictor
 Upload data
 Upload CSV Files

Upload files

forecast No file chosen
 real data

Please note, that the CSV file has to follow this format.

Date	Hour	Volume	SLA	Agents
mm/dd/yyyy	X	Y	Z	A
Date in month/date/year format	Number of hour from 0 to 23	Either forecast or real volume, based on the selected button	Either forecast or real SLA, based on the selected button	Number of planned or real agents in said hour, planned + forecast
Required	Required	Optional - empty value stored in database if empty	Optional - empty value stored in database if empty	Optional - empty value stored in database if empty

Obrázek 8.5 - Stránka s formulářem pro upload souborů CSV s predikcí, či reálnými daty a popisem předeepsané struktury CSV souboru s vysvětlením povinných dat

Příloha C – Vyjádření Kiwi.com s.r.o.

Web Application for Making Predictions of a Call Centre – company's acknowledgement

Kiwi.com hereby confirms that David's application proved to be a useful application for the creation of predictions and has a potential to be used in our daily operations. David Mička consulted the structure of the app with our PHP team to follow the same structure and tools as they use in our internal system Nest so that the integration of the app to the system would be possible.

Kiwi.com tímto potvrzuje, že Davidova aplikace se osvědčila být užitečnou aplikací pro tvorbu predikcí výsledků a má potenciál, být používána v denim provozu. David Mička konzultoval strukturu aplikace s naším PHP týmem, aby zajistil, že aplikace bude mít stejnou strukturu a použité nástroje, jako je tomu u našeho interního nástroje Nest. Díky tomu je zajištěna možnost budoucí integrace aplikace do tohoto systému.

14.5.2018 in Brno



.....
Kamil Jagodzinski
Chief Customer Officer of Kiwi.com s.r.o.