



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

**ANALÝZA, NÁVRH A TVORBA SPRÁVCOVSKÉHO
INFORMAČNÍHO SYSTÉMU**

ANALYSIS, DESIGN AND CREATION OF INFORMATION SYSTEM FOR ADMINISTRATORS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Matúš Dobrovodský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2019

Zadání diplomové práce

Ústav: Ústav informatiky
Student: **Bc. Matúš Dobrovodský**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **Ing. Petr Dydowicz, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Analýza, návrh a tvorba správcovského informačního systému

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrh řešení, přínos práce
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem mé práce je zanalyzovat současný stav informačních systémů společnosti, navrhnout procesy a vytvořit samostatný informační systém určen pro administrátory systémů. Analýza bude zaměřená na současný stav databáze a procesů stávajících informačních systémů. Na základě analýzy vypracuji návrh nového informačního systému, což zahrnuje návrh a tvorbu databáze a také procesů probíhajících v informačním systému.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy. Podnik v informační společnosti. Praha: Grada, 2008. 283 s. ISBN 978-80-247-2279-5.

MOLNÁR, Z. Automatizované informační systémy. Praha: Strojní fakulta ČVUT, 2000. 126 s. ISBN 80-01-02269-2.

MOLNÁR, Z. Efektivnost informačních systémů. Praha: Grada Publishing, 2000. 142 s. ISBN 80-716--410-X.

ŘEPA, V. Analýza a návrh informačních systémů. Praha: Ekopress, 1999. 403 s. ISBN 80-86119--3-0.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

doc. RNDr. Bedřich Půža, CSc.

ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.

děkan

Abstrakt

Diplomová práca sa zameriava na analýzu, návrh a tvorbu informačného systému vytvoreného pre administrátorov. Pomocou informačného systému bude možné spravovať všetky webové aplikácie spoločnosti z centrálného miesta.

Abstract

The master's thesis focuses on analysis, design and creation of an information system designed for administrators. With this information system will be possible to manage all company's web applications from a central location.

Klíčové slová

informačný systém, databáza, proces, vývojový diagram, SQL, MySQL, PHP, HTML

Key words

information system, database, process, flowchart, SQL, MySQL, PHP, HTML

Bibliografická citácia

DOBROVODSKÝ, Matúš. *Analýza, návrh a tvorba správcovského informačného systému* [online]. Brno, 2019 [cit. 2019-04-29]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/118397>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Petr Dydowicz.

Čestné prehlásenie

Prehlasujem, že predložená diplomová práca je pôvodná a spracoval som ju samostatne.
Prehlasujem, že citácia použitých prameňov je úplná, že som vo svojej práci neporušil autorské práva (v zmysle Zákona č. 121/2000 Sb., o právu autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 6. mája 2019

.....

podpis autora

Pod'akovanie

Za poskytnutie odbornej pomoci, poskytnutých cenných rád a pomoci pri tvorbe diplomovej práce by som chcel poďakovať svojmu vedúcemu práce Ing. Petrovi Dydowiczovi Ph.D.

Obsah

ÚVOD.....	11
1. TEORETICKÉ VÝCHODISKÁ PRÁCE.....	14
1.1 Vymedzenie základných pojmov	14
1.1.1 Informácie, dáta a znalosti	14
1.1.2 Informačný systém	15
1.2 Budovanie informačného systému	15
1.3 Životný cyklus IS	16
1.4 Dátový pohľad na informačné systémy	16
1.4.1 Základné vrstvy informačného systému v organizácií.....	16
1.4.2 Hlavné dáta používané v podnikových IS.....	17
1.5 Databázové systémy	18
1.5.1 Systém riadenia databáze	19
1.5.2 Dátové modely	20
1.6 Relačný model	22
1.6.1 Typy kľúčov	22
1.6.2 Zložené kľúče.....	23
1.6.3 Kandidátne a primárne kľúče.....	23
1.6.4 Náhradné kľúče	23
1.6.5 Cudzie kľúče a referenčná integrita	24
1.6.6 Normálne formy	24
1.7 Funkčné modelovanie.....	26
1.7.1 Slovný popis funkčného modelu.....	26
1.7.2 Stavový diagram.....	27
1.7.3 Diagram toku dát.....	27
1.7.4 Vývojový diagram.....	28
1.7.5 Rozhodovacia tabuľka.....	28
1.8 Jazyk SQL	28
1.8.1 Príkazy jazyka SQL.....	29
1.9 MySQL databáza	29
1.10 Apache server	30
1.11 Jazyk HTML a CSS3	31
1.11.1 Základné HTML tagy	32
1.11.2 Kaskádové štýly.....	33

1.12	Jazyk PHP	33
1.13	JavaScript.....	35
1.14	Bootstrap.....	36
2.	ANALÝZA SÚČASNÉHO STAVU	37
2.1	Základné informácie o spoločnosti	37
2.2	Organizačná štruktúra spoločnosti	38
2.3	Produkt WifiMedia.....	38
2.4	Informačný systém pre zákazníkov	39
2.4.1	Moduly informačného systému.....	40
2.4.2	Databáza informačného systému.....	41
2.5	SWOT analýza informačného systému	49
2.6	Požiadavky na informačný systém pre administrátorov	50
2.7	Oprávnenia administrátorov	51
2.8	Procesy informačného systému	52
2.8.1	Založenie účtu podnikom a inzerentom.....	52
2.8.2	Založenie administrátorského účtu	53
2.8.3	Náhľad a editácia údajov o užívateľovi IS.....	54
2.8.4	Navýšenie kreditu.....	56
2.8.5	Procesy spojené s aktívnymi prvkami	57
2.8.6	Náhľad na počet prihlásených užívateľov.....	59
2.8.7	Založenie reklamnej kampane.....	60
2.8.8	Odstránenie reklamnej kampane	61
2.8.9	Zablokovanie užívateľského účtu	62
2.9	Zhodnotenie analýzy súčasného stavu.....	63
3.	VLASTNÉ NÁVRHY RIEŠENIA	65
3.1	Príprava prostredia.....	65
3.1.1	Inštalácia MySQL, Apache a PHP	65
3.1.2	Príprava databáze	66
3.2	Prihlasovanie užívateľov do IS.....	67
3.3	Úvodná stránka informačného systému.....	70
3.3.1	Menu systému	71
3.3.2	Widgety na úvodnej stránke.....	72
3.4	Routery	74
3.4.1	Úvodná stránka komponenty routerov	75
3.4.2	Vytvorenie nového routeru	76

3.4.3	Deaktivácia routeru	78
3.4.4	Editácia routeru	79
3.5	Zákazníci	80
3.5.1	Úvodná stránka komponenty podnikov	81
3.5.2	Vytvorenie nového podniku.....	82
3.5.3	Editácia podnikov.....	84
3.5.4	Zmena hesla účtu podnikov	85
3.5.5	Úprava kreditu podniku.....	86
3.5.6	Deaktivácia účtu podniku.....	87
3.6	Inzerenti	88
3.6.1	Úvodná stránka komponenty.....	89
3.6.2	Vytvorenie nového účtu inzerentom	90
3.6.3	Dobitie kreditu a nastavenie limitu	91
3.7	Reklamné kampane	92
3.7.1	Úvodná stránka komponenty.....	93
3.7.2	Vytvorenie reklamnej kampane	94
3.7.3	Editácia reklamnej kampane	95
3.7.4	Zablokovanie reklamnej kampane	96
3.8	Správa užívateľských účtov.....	97
3.8.1	Založenie nového účtu	97
3.8.2	Deaktivácia správcovského účtu	98
3.8.3	Editácia administrátorského účtu	99
3.9	Náhľad na štatistiky prihlásení	99
3.9.1	Úvodná stránka komponenty.....	100
3.10	Ekonomické zhodnotenie.....	101
3.11	Prínosy práce.....	101
	ZÁVER	102
	ZOZNAM POUŽITÝCH ZDROJOV	103
	ZOZNAM POUŽITÝCH OBRÁZKOV	106
	ZOZNAM POŽITÝCH TABULIEK.....	110

ÚVOD

V súčasnom svete sú informačné systémy všade okolo nás. Je takmer nemožné, aby sa akákoľvek spoločnosť na trhu zaobišla bez informačného systému. Informačné systémy nám pomáhajú ušetriť čas a peniaze, preto sú pre spoločnosti veľmi prínosné. Nie je to inak ani vo firme, ktorú som si vybral pre svoju diplomovú prácu.

Spoločnosť podniká v oblasti, v ktorej jej informačné systémy využívajú zákazníci. Na mieste je teda otázka, akým spôsobom má spoločnosť spravovať tieto systémy. Dostávame sa do obdobia, kedy sú potrebné informačné systémy, pomocou ktorých je možné spravovať množinu informačných systémov.

Predstavme si situáciu, že existuje jedna centrálna databáza, nad ktorou sú 3 informačné systémy. Je rozumné mať v každom jednom systéme vlastné administrátorské rozhranie, pomocou ktorého bude možné konkrétny systém spravovať? Čo ak nastane situácia, kedy pribudne ďalší informačný systém? Bude opäť rozumné, aby obsahoval ďalšie vlastné rozhranie pre správu? V takýchto prípadoch by sa musel administrátor prihlasovať do každého systému zvlášť a spravovať systémy jednotlivo.

Vo svojej diplomovej práci som sa rozhodol vyvinúť pre spoločnosť jednotné správcovské rozhranie (centrálne miesto), pomocou ktorého budú môcť administrátori spravovať všetky informačné systémy. Samotná tvorba informačného systému potrebuje dôkladne spracovanú analýzu a návrh. V kapitole analýzy súčasného stavu som sa teda venoval analýze centrálnej databáze a návrhu procesov, ktoré by malo správcovské rozhranie obsahovať.

CIELE PRÁCE

V tejto kapitole popíšem a jasne vytyčím ciele, ktoré má moja diplomová práca dosiahnuť. Ide o dôležitú kapitolu, na základe ktorej sa v závere poukáže na fakt, či táto práca mala takú úroveň, aby tieto ciele naplnila.

Tak, ako v úvode píšem, práca je rozdelená na tri hlavné časti, ktorými sú teoretické východiská práce, analytická a praktická časť, no nemôžem vynechať aj kapitolu s ekonomickým vyhodnotením navrhovaného riešenia. Každá táto časť, respektíve kapitola práce musí spĺňať požiadavky, ktoré sa premietajú do cieľov v práci.

V teoretickej časti práce som vytyčil nasledujúci cieľ. Táto **kapitola musí zachytávať v teoretickej rovine problematiku, ktorou sa moja práca zaoberá**. Dá sa pod tým predstaviť to, že ak čitateľ v práci narazí na terminológiu, s ktorou nie je oboznámený, tak táto problematika musí byť obsiahnutá v teoretických východiskách. Samozrejme, že táto časť práce je veľmi dôležitá, aj keď sa to na prvý pohľad nemusí zdať, keďže v tejto kapitole sa nič nové nevymýšľa ani netvorí, ale len akýmsi spôsobom sa zoskupujú už známe informácie.

Keďže samotná návrhová časť sa opiera o tú analytickú, tak mojím cieľom v analytickej časti práce je **identifikovať a popísať všetky procesy, ktoré budú začlenené do informačného systému**. Keďže samotný produkt už má svoj súčasný stav, tak je dôležité aby som v tejto časti práci **zanalyzoval súčasný stav informačných systémov spoločnosti a taktiež súčasný stav databáze**. Ako som písal vyššie, jedným z cieľov tejto kapitoly je identifikovať procesy, no taktiež je dôležité určiť, **kto bude mať prístup k daným procesom**, teda je potreba určiť právo prístupu užívateľov ku konkrétnym procesom.

Na základe týchto vytyčených cieľov môžem prejsť ku samotnému návrhu riešenia. V návrhovej časti sa odzrkadľuje úroveň spracovania ako analytickej časti práce, tak aj teoretickej časti práce. Je preto dôležité všetky ciele, ktoré sú vytyčené v týchto kapitolách splniť, aby aj samotný návrh riešenie bol spracovaný kvalitne. Preto som si určil, že v návrhovej časti práce je mojím cieľom **na základe spracovanej analýzy vytvoriť informačný systém, ktorý bude schopný v plnej miere vykonávať všetky**

procesy zanalyzované v danej kapitole. Znamená to, že užívateľ bude môcť vykonať všetky požiadavky, ktoré sú na informačný systém a vyplývajú z analýzy.

Pri vytýčení cieľov nemôžem zabudnúť ani na ekonomické zhodnotenie navrhovaného riešenia a preto som sa rozhodol, že cieľom tohto zhodnotenia je **prehľadne a jednoznačne určiť náklady plynúce na tvorbu tohto informačného systému.**

1. TEORETICKÉ VÝCHODISKÁ PRÁCE

V tejto časti mojej diplomovej práci sa budem venovať teoretickým východiskám. Táto časť práce je dôležitá, aj keď pre mnohých príliš nezaujímavá. V ďalších častiach práce budem zo získaných teoretických poznatkov vychádzať, či už v oblasti analýzy a návrhu, tak i pri samotnej tvorbe informačného systému.

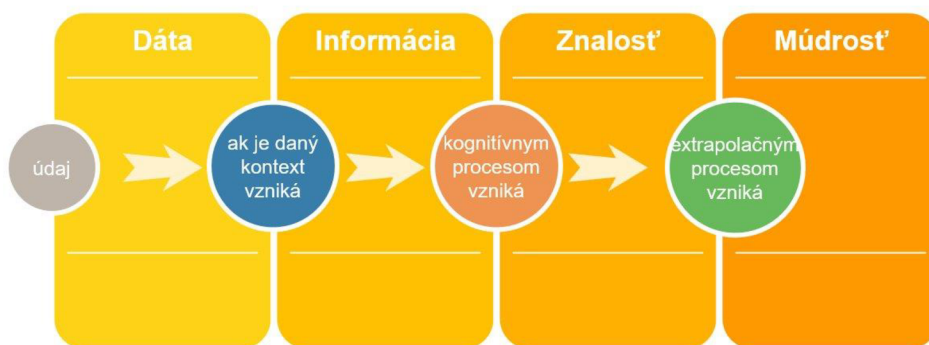
1.1 Vymedzenie základných pojmov

Na úvod tejto kapitoly je potreba si vymedziť základné pojmy, ktoré sa v práci budú veľmi často vyskytovať. Pôjde najmä o jasné definovanie toho, čo sú dáta, informácie, čo je informačný systém a podobne.

1.1.1 Informácie, dáta a znalosti

Dáta, informácie a znalosti sú navzájom prepojené. Samotné dáta sú len údaje, ktoré sú často vo forme nejakého čísla, znaku, prípadne symbolu. Nemajú žiaden vonkajší kontext a nedávajú žiaden zmysel. Môže sa zdať, že nemajú žiadnu hodnotu. Ak však dáta vložíme do kontextu, tak v tomto momente začnú nadobúdať zmysel a stávajú sa tak informáciou, ktorej človek rozumie a vie ju využiť. Informácia odpovedá na otázku, kto, čo, kedy a kde (1).

Informácia, ktorá má štruktúru a je organizovaná ako výsledok kognitívneho procesu, stáva sa znalosťou. To znamená, že znalosť je teda informácia, na základe ktorej vie človek uskutočniť dané rozhodnutie, prípadne ju využiť (1).



Obrázok č. 1: Dáta, informácie, znalosti a múdrosť (1)

1.1.2 Informačný systém

Na začiatok je dobré vysvetliť, čo je informačný systém.

V teórii systémov sa rozumie pod pojmom systém usporiadaná množina prvkov, ktoré majú spoločné vlastnosti a prvky majú medzi sebou vzťahy. Definíciou informačného systému je samozrejme viacero, no najčastejšou definíciou je, že IS je súbor ľudí, technických prostriedkov a metód, zabezpečujúcich zber, prenos, spracovanie, uchovanie dát za účelom prezentácie informácií pre potreby užívateľov (2).

1.2 Budovanie informačného systému

Prvou vecou pri budovaní informačného systému je potrebné si položiť otázku, čo očakávame od nasadenia nového alebo inovácie stávajúceho informačného systému. Odpoveďou na túto otázku determinujú dva hlavné prístupy. Ten prvý, ktorý sa uplatňuje u väčšiny organizácií je, že organizácie chápe IS ako podporný nástroj pre riadenie. Hlavnými požiadavkami na taký systém sú:

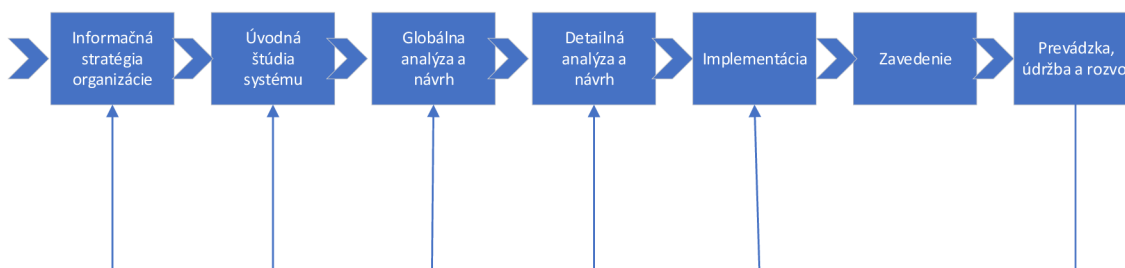
- podpora automatizácie každodennej rutiny,
- dostupnosť informácií pre rozhodovanie,
- jednotná verzia pravdy prítomná vo všetkých výstupoch systému (4).

Druhý prístup, ktorý smeruje ku maximálne výhodnému pomeru cena/kvalita/pridaná hodnota celého riešenia sa opiera o požiadavky smerujúcich do oblastí, ktoré nesúvisia iba s vlastnosťami informačného systému, ale zohľadňujú taktiež:

- zmeny nutné v organizačnej štruktúre a riadení spoločnosti,
- štandardizácia podnikových procesov a pracovných návykov,
- zdieľanie najlepších praktík so zmluvnými partnermi a konkurenciou,
- poskytnutie pohľadu na fungovanie organizácie,
- zaistenie podpory manažérskeho rozhodovania až po strategickú úroveň,
- zvyšovanie výkonnosti a podpora konkurencieschopnosti podniku (4).

1.3 Životný cyklus IS

Je dôležité poznamenať, že aj informačný systém má svoj životný cyklus. Tento cyklus zaznamenáva nasledujúci obrázok:



Obrázok č. 2: Životný cyklus IS (Upravené podľa 3, s. 18)

1.4 Dátový pohľad na informačné systémy

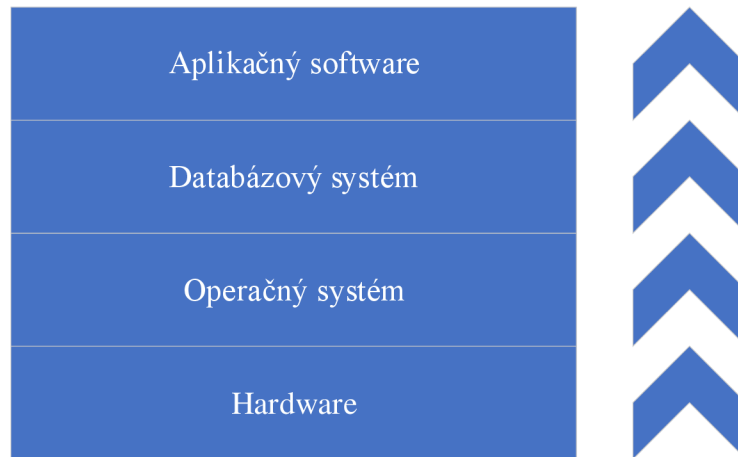
Dátové zjednotenie rôznych aplikácií informačných systémov prostredníctvom spoločnej databáze predstavuje jeden z principiálnych fenoménov úspešnosti a rozvoja informačných systémov v organizácii. Kdekoľvek a odkiaľkoľvek prístupné dáta, ktoré sú uložené v spoločnej databáze predstavovala technologickú zmenu. Relačné databáze napomohli zjednoteniu podnikových dát a on-line dostupnosti ku zníženiu nákladov na materiálové zásoby, skráteniu času realizácie zákazky i presnejšiemu a rýchlejšej podpore rozhodovania (5).

1.4.1 Základné vrstvy informačného systému v organizácii

Využitie dát pri dekompozícii podnikových IS patrí medzi možné, ale z hľadiska početnosti častejšie prístupy. Dátový pohľad je taktiež dôležitou súčasťou architektúry IS podniku. Jeho využitie je napríklad tými, ktorí programujú v databázových prostrediach vlastné aplikácie a prostredníctvom vlastného programu následne realizujú aplikačné riešenie spolu s užívateľským prostredím i dokumentáciou (5).

Tento prístup chápania IS je možné označiť svojim spôsobom za technologický. Je možné ho znázorniť pomocou na seba nadväzujúcich vrstiev, kde základ tvorí hardware a ďalšie

vrstvy smerujú postupne skrz operačný systém a databázové prostredie ku aplikačnému softwaru (5).



Obrázok č. 3: Technologický model podnikového IS (Upravené podľa 5, s. 100)

1.4.2 Hlavné dáta používané v podnikových IS

Pripravenosť, správnosť a úplnosť dát a spôsob ich prenosu do softwarovej aplikácie informačného systému významne ovplyvní kvalitu implementácie a taktiež následnú efektivitu využívania IS (5).

Z hľadiska používania dát vnútri IS existuje päť základných skupín:

- **číselníky** – používajú sa pre identifikáciu položiek, pracovísk, skladových miest, nákladových stredísk, referentov, dodávateľov, zákazníkov a podobne,
- **kmeňové dáta** – obsahujú najmä údaje o:
 - výrobkoch,
 - spôsobu realizácie výrobku,
 - výrobnej základne,
 - dodávateľoch,
 - zákazníkoch.
- **zákazkové dáta** – údaje o zákazke pre konkrétneho zákazníka s väzbou na požadované termíny, množstve, štruktúre a uskutočnení výrobku,
- **archívne dáta,**

- **parametre** – predstavujú posledný typ dát, ktoré obsahujú hodnoty pre nastavenie optimálneho systému a jeho jednotlivých moduloch v konkrétnych podmienkach (5).

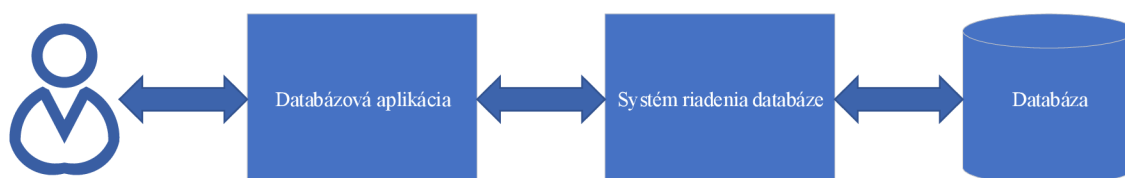
1.5 Databázové systémy

Vďaka databázovým systémom máme možnosť zhromažďovať špecifické informácie, ktoré sú pre nás dôležité. Veľkou výhodou je, že tieto informácie umožňuje ukladať a udržiavať na jednom centrálnom mieste (6).

Znalosti databázových technológií dnes neustále získavajú na význame. Databáze sa používajú všade a sú kľúčovým prvkom elektronického obchodovania a iných webových aplikácií. Spočívajú v jadre aplikácií na riadenie podnikov a podporu rozhodovania. K databázam tiež pristupujú členovia tisícok pracovných skupín a milióny jednotlivcov. Odhaduje sa, že v súčasnosti na svete existuje viac než 10 miliónov aktívnych databáz (7).

Obecne sa každý databázový systém skladá z:

- **systém riadenia báze dát** – program, ktorý organizuje a udržiava nazhromaždené informácie,
- **databázová aplikácia** - programu, ktorý umožňuje vyberať, prehliadať a aktualizovať informácie,
- **databáza** – uložené dáta,
- **užívatelia** – pomocou databázovej aplikácie udržiavajú prehľad o určitých skutočnostiach. Pomocou formulárov načítavajú a zadávajú dáta a dopytujú sa na ne a vytvárajú zostavy (7).



Obrázok č. 4: Komponenty databázového systému (Upravené podľa 7, s. 30)

1.5.1 Systém riadenia databáze

Úlohou systému riadenia databáze je vytvárať, spracovávať a spravovať databáze. Systém riadenia databáze je veľký a zložitý produkt, ktorý sa takmer vždy vyhotovuje formou licencie od dodávateľa softwaru. Jedným z príkladom systému riadenia databáze je Microsoft Access. K ďalším komerčným produktom tohto typu patrí:

- Microsoft SQL Server,
- MySQL od Oracle Corporation,
- Oracle Database od Oracle Corporation,
- DB2 od IBM (7).

Funkcie systému riadenia databáze:

- vytvorenie databáze,
- vytvorenie tabuliek,
- vytvorenie podporných štruktúr (napríklad indexy),
- čítanie dát z databáze,
- úpravy databázových dát:
 - vkladanie,
 - aktualizovanie,
 - odstránenie.
- údržba databázových štruktúr,
- vynucovanie pravidiel,
- kontrola súbežnosti,
- zaistenie bezpečnosti,
- zálohovanie a obnovenie (7).

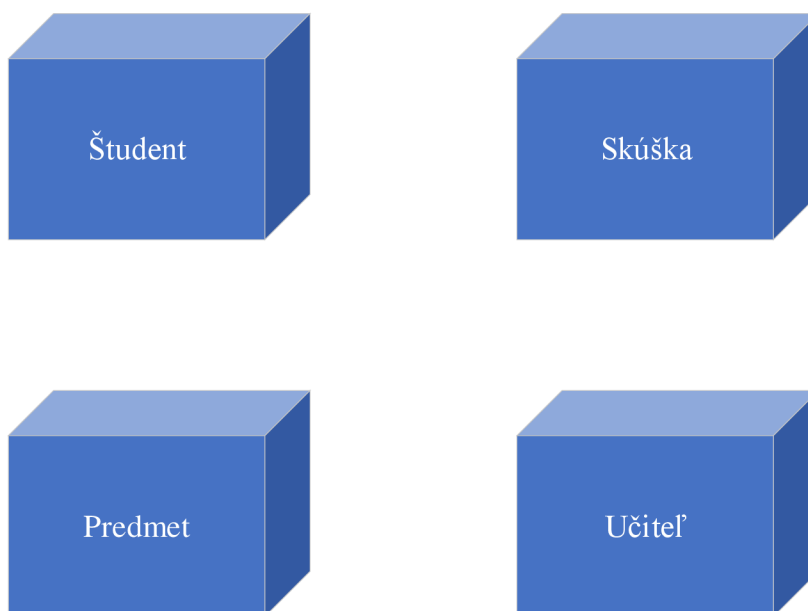
Databáza, ktorá slúži ako centralizované úložisko dát, predstavuje veľmi cenné organizačné aktívum. Vzhľadom k tomu, ako môžu byť databáze dôležité, je potreba uskutočniť bezpečnostné opatrenie, aby v prípade chýb (hardwarových či softwarových) nedošlo ku strate žiadnych dát (7).

1.5.2 Dátové modely

Pri vytváraní informačných systémov spravidla nevystačíme s jedinou štruktúrou vety. Pre každý typ dátového objektu musíme navrhnuť samostatnú dátovú štruktúru. V realite, ktorú poznáme objekty spolu súvisia a preto v IS musíme vytvoriť odpovedajúci obraz reality, tak aby dáta vložené do systému tejto realite odpovedali (8).

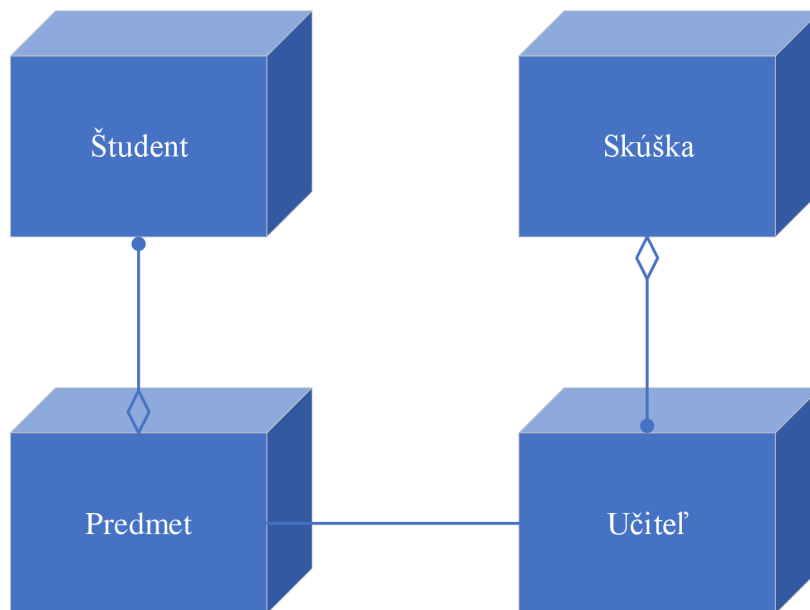
V súčasnej dobe máme pri projektovaní informačných systémov tri možné typy dátových modelov, ktoré môžeme použiť a to:

- lineárny dátový model,
- relačný dátový model,
- objektový dátový model (8).



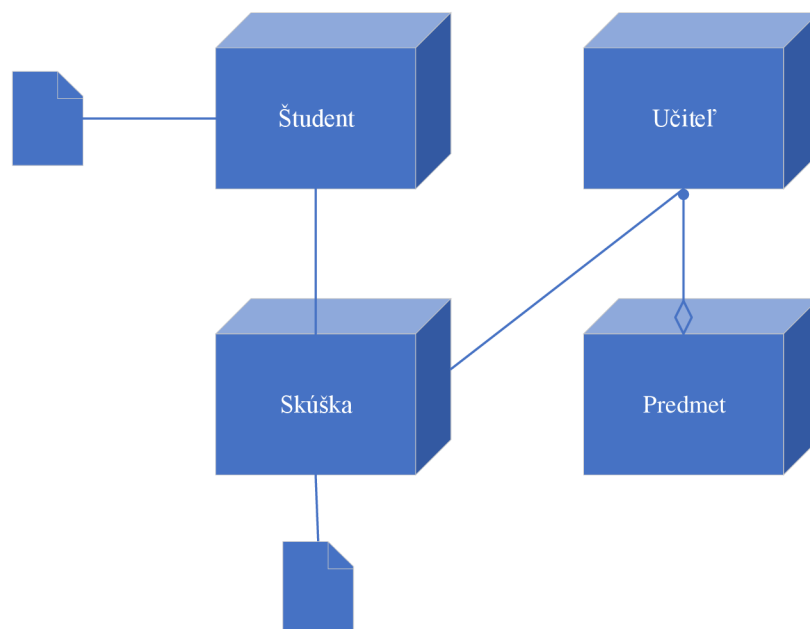
Obrázok č. 5: Lineárny dátový model (Upravené podľa 8, s. 20)

Na obrázku, ktorý je zobrazený vyššie, je zachytený lineárny dátový model. Ak pri realizácii tohto modelu použijeme databázový systém, tak každý obdĺžnik predstavuje jednu tabuľku v databáze. Nevýhoda lineárnych modelov je, že medzi jednotlivými skupinami nie je žiadna väzba, teda nemohli by sme presne určiť, ktorý študent mal z ktorého predmetu skúšku (8).



Obrázok č. 6: Relačný dátový model (Upravené podľa 8, s. 21)

Obrázok vyššie zachytáva relačný dátový model. Relačný dátový model patrí v súčasnosti k najpoužívanejším. Vzniká z niekoľkých lineárnych modelov spojených dohromady prostredníctvom položiek, ktoré sa nazývajú relačný kľúč. Toto spojenie nie je trvalé, ale vzniká v okamžiku, kedy potrebujeme mať spoločne k dispozícii dáta zo všetkých spojených tabuliek a zaniká v okamžiku, kedy prácu s modelom ukončíme (8).



Obrázok č. 7: Objektový dátový model (Upravené podľa 8, s. 21)

K najnovším dátovým modelom v súčasnosti je objektový dátový model, ktorý je zachytený na obrázku vyššie. Objektové dátové modely sú vystavené na základnom prvku, a to objekte, kde tento objekt má okrem svojich atribútov aj definované metódy, ktoré určujú chovanie objektu (8).

1.6 Relačný model

V nasledujúcich riadkoch je popísaný relačný model, čo je zďaleka najdôležitejší štandard súčasného databázového odboru. Tento model vznikol v roku 1970 a vyvinul ho a publikoval Edgar Frank Codd. Tento model je založený na základe relačnej algebry (7).

Na začiatok je potreba definovať základné pojmy. V prvom rade to sú **entity**, ktoré definujeme ako niečo, čo je pre užívateľa dôležité a čo je potreba reprezentovať v databázy. **Reláciu** môžeme definovať ako dvojrozmernú tabuľku, ktorá sa skladá z riadkov a stĺpcov a vyznačuje sa nasledujúcimi vlastnosťami:

- každý riadok tabuľky obsahuje dáta, ktoré sa týkajú nejakej entity alebo časti tejto entity,
- každý stĺpec tabuľky obsahuje dáta, ktoré reprezentujú atribúty entity,
- bunky v tabuľke musia uchovávať jedinú hodnotu, nemôžu teda obsahovať opakujúce sa prvky,
- všetky položky v každom stĺpci musia byť rovnakého druhu,
- všetky stĺpce musia mať jedinečný názov,
- na poradí stĺpcov v tabuľke nezáleží,
- nezáleží ani na poradí riadkov,
- sada dátových hodnôt musí byť na každom riadku jedinečná, čo znamená, že žiadne dva riadky v tabuľke nesmú obsahovať identické sady dátových hodnôt (7).

1.6.1 Typy kľúčov

Kľúč je jeden alebo viac stĺpcov v relácii, ktorý umožňuje identifikovať riadok. Kľúč môže byť jedinečný alebo nejedinečný. Pod jedinečným kľúčom si môžeme predstaviť napríklad rodné číslo nejakej osoby. Vieme, že každé rodné číslo je unikátne a v tomto prípade ide o jedinečný kľúč. Naopak ako nejedinečný kľúč si môžeme predstaviť

napríklad oddelenia vo firme. Každý zamestnanec patrí do určitého oddelenia, teda aj oddelenie nám umožní identifikovať riadok. Avšak v každom oddelení je viacero zamestnancov, pretože ide o nejedinečný kľúč (7).

1.6.2 Zložené kľúče

Kľúč, ktorý obsahuje dva alebo viacero atribútov sa nazýva zložený kľúč. Dá sa to vysvetliť na jednoduchom prípade. Užívatelia nám oznámili, že hodnoty u priezviska síce nie sú jedinečné, ale v spojení s oddelením vo firme sa neopakujú. Ak by toto pravidlo teda platilo, tak kombinácia priezviska a oddelenia bude predstavovať jedinečný zložený kľúč. Samozrejme, užívatelia môžu vedieť, že táto kombinácia nie je jedinečná, ale v spojení s menom, priezviskom a oddelením už je. V takom prípade bude zložený kľúč s tromi atribútmi (7).

1.6.3 Kandidátne a primárne kľúče

Kandidátne kľúče sú kľúče, ktoré jedinečne identifikujú každý riadok v relácií. Tieto kľúče môžu byť založené na jedinom stĺpce, prípadne sa môže jednať o zložené kľúče. Primárny kľúč je kandidátny kľúč, ktorý bol zvolený ako kľúč, podľa ktorého systém riadenia databáze identifikuje všetky riadky v relácií. Primárny kľúč je dôležitý nielen preto, že umožňuje identifikovať jedinečné riadky, ale taktiež z toho dôvodu, že je možné pomocou neho reprezentovať riadky v relácií. Dá sa to reprezentovať na jednoduchom príklade, kedy máme tabuľku zákazníkov a primárny kľúč je identifikačné číslo zákazníka. Následne máme tabuľku nákup, kde je umiestnený stĺpec s identifikačným číslom zákazníka, čím sme vytvorili prepojenie oboch tabuliek (7).

1.6.4 Náhradné kľúče

Náhradný kľúč je stĺpec s jedinečným identifikátorom, ktorý priradzuje systém riadenia databáze. Tento stĺpec je doplnený do tabuľky preto, aby plnil úlohu primárneho kľúča. Systém riadenia databáze prideluje unikátne hodnoty náhradného kľúča pri každom vytvorení riadku a tieto hodnoty sa nikdy nemenia. Ideálny primárny kľúč je krátky, číselný a jeho hodnoty sa nikdy nemenia. Náhradné kľúče sú krátke, číselné a ich hodnoty zostávajú stále rovnaké, takže sa jedná o ideálny primárne kľúče. Vzhľadom k tomu, že

hodnoty náhradného primárneho kľúča nemajú pre užívateľa žiadny zmysel, často sú vo formulároch, výsledkoch dotazu a zostavách skryté (7).

1.6.5 Cudzie kľúče a referenčná integrita

V kapitole 1.6.3 je vysvetlené, že ak chceme reprezentovať vzťah, umiestnime hodnoty z jednej relácie do druhej relácie. Používame pritom hodnoty primárneho kľúča prvej relácie. Atribút druhej relácie, ktorý uchováva tieto hodnoty, sa v tomto prípade označuje ako cudzí kľúč. Dôležitý poznatok je to, že primárny kľúč a cudzí kľúč nemusia mať nutne rovnaký názov stĺpca, ale jediná požiadavka je, že musia mať rovnakú sadu hodnôt. Vo väčšine prípadov je potrebné zaistiť, aby každá hodnota cudzieho kľúča odpovedala hodnote primárneho kľúča. Takéto pravidlo sa nazýva **obmedzenie referenčnej integrity** (7).

1.6.6 Normálne formy

Tento pojem sa používa v spojitosti s dobre navrhnutými tabuľkami. Správne vytvorené tabuľky splňujú 4 základné formy (6).

1. normálna forma – najjednoduchšia forma, ktorá hovorí, že všetky atribúty sú atomické, čo znamená, že sa nedajú ďalej deliť. Obecne by sme sa mali snažiť, aby obsahom jednej databázovej položky bola práve jedna hodnota (6).

2. normálna forma – tabuľka splňuje túto formu, ak splňuje predchádzajúcu a navyše každý atribút, ktorý nie je primárnym kľúčom je na primárnom kľúči úplne závislý. Z toho vyplýva, že sa nesmie v riadku tabuľky objaviť položka, ktorá by bola závislá iba na časti primárneho kľúča. Z definície vyplýva, že problém tejto formy sa týka iba tabuliek, kde volíme za primárny kľúč viacej položiek než jednu (6).

3. normálna forma – relačné tabuľky splňujú túto formu, ak splňujú predchádzajúce dve formy a zároveň, že žiadny atribút, ktorý nie je primárnym kľúčom, nie je tranzitívne závislý na žiadnom kľúči (6).

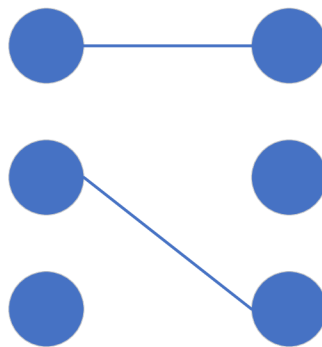
Boyce-Coddova normálna forma – tabuľka splňuje túto formu práve vtedy, keď pre dve množiny atribútov A a B platí: $A \rightarrow B$ a súčasne B nie je podmnožinou A, potom množina A obsahuje primárny kľúč tabuľky. Táto forma zjednodušuje prácu s tabuľkami a obecne,

ak sme dobre postupovali pri tvorbe tabuliek podľa prvej až tretej formy, Boyce-Coddova forma je splnená (6).

1.6.7 Integritné obmedzenia pre vzťahy

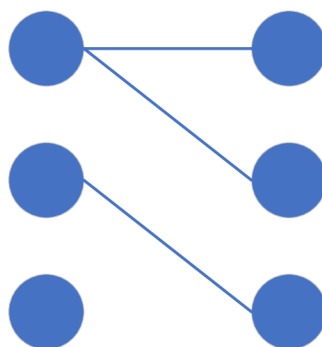
Integritné obmedzenie pre vzťahy obmedzuje kardinalitu vzťahu na pomery:

- 1 : 1,
- 1 : N,
- N : 1,
- N : M (8).



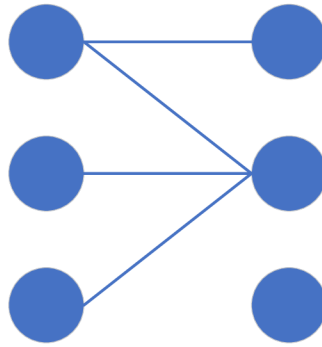
Obrázok č. 8: Vzťah 1 : 1 (Upravené podľa 8, s. 30)

Vzťah 1 : 1 nám hovorí, že vždy ku jednej n-tici relácie odpovedá jedna alebo žiadna n-tica inej relácie. Príkladom môže byť vzťah medzi dátovým objektom človek a objektom vodičský preukaz. Jeden človek môže vlastniť jeden alebo žiadny vodičský preukaz (8).



Obrázok č. 9: Vzťah 1 : N (Upravené podľa 8, s. 30)

Vzťah 1 : N nám hovorí, že vždy jednej n-tici relácie odpovedá jedna alebo viac n-tic inej relácie. Príkladom môže byť vzťah medzi entitami študent a skúška. Jeden študent môže vykonať viacej skúšok (8).



Obrázok č. 10: Vzťah N : M (Upravené podľa 8, s. 30)

Vzťah N : M nám hovorí, že obecně niekoľkým n-ticiam relácie odpovedá jedna alebo viac n-tic inej relácie. Príkladom môže byť vzťah medzi entitami študent a predmet, kedy si študent vyberie viacero predmetov, ale jeden predmet navštevuje viacero študentov. Riešením tohto vzťahu je vytvorenie novej entity, ktorá sa nazýva prieniková entita (8).

1.7 Funkčné modelovanie

V predchádzajúcich kapitolách som sa venoval dátovému modelovaniu, čo predstavuje problematiku dát, ktoré potrebujeme pre informačný systém. Funkčné modelovanie sa zaoberá skúmaním a algoritmizáciou činností a procesov, ktoré v IS prebiehajú. Pri popisu činností v IS môžeme uskutočniť hierarchický rozklad funkcií od najobecnejších až po elementárne funkcie, ktoré majú užívatelia k dispozícii (8).

Príklady funkčného modelovania sú:

- slovný popis funkčného modelu,
- stavový diagram,
- diagram toku dát,
- vývojový diagram,
- rozhodovacia tabuľka (8).

1.7.1 Slovný popis funkčného modelu

Táto metóda patrí k najpoužívanejším pri riešení úloh menšieho rozsahu a pre komunikáciou vnútri pracovného analytického tímu. Z dôvodu menšej prehľadnosti sa nepoužíva v dokumentácii informačných systémov (8).

1.7.2 Stavový diagram

V metóde stavového diagramu skúmame možné stavy objektov, ktoré môžu nastať a snažíme sa popísať, čo tieto zmeny prechodu medzi stavmi vyvolá. U tohto diagramu je potrebné si uvedomiť, že ho vždy kreslíme pro určitú entitu, teda objekt. V elipsách sú zakreslené možné stavy. Šípky následne určujú, za akej podmienky prechádza systém z jedného stavu do druhého, pričom u šípok rozpisujeme príslušné podmienky (8).

1.7.3 Diagram toku dát

Diagram toku dát je jedna z najpoužívanějších metód funkčného modelovania. Môžeme z nej čítať náväznosť jednotlivých činností v rámci úlohy, aké dátové vstupy a výstupy sa v úlohe objavujú a kto jednotlivé činnosti uskutočňuje (8).

V diagrame toku dát sa pracuje s nasledujúcimi symbolmi:



Obrázok č. 11: Symboly DFD diagramu (Upravené podľa 8, s.84)

Proces – činnosť, transformácia vstupných dát na výstupné. Názov by mal vyjadrovať podstatu transformácie. Každý proces je buď špecifikovaný alebo reprezentovaný iným DFD (8).

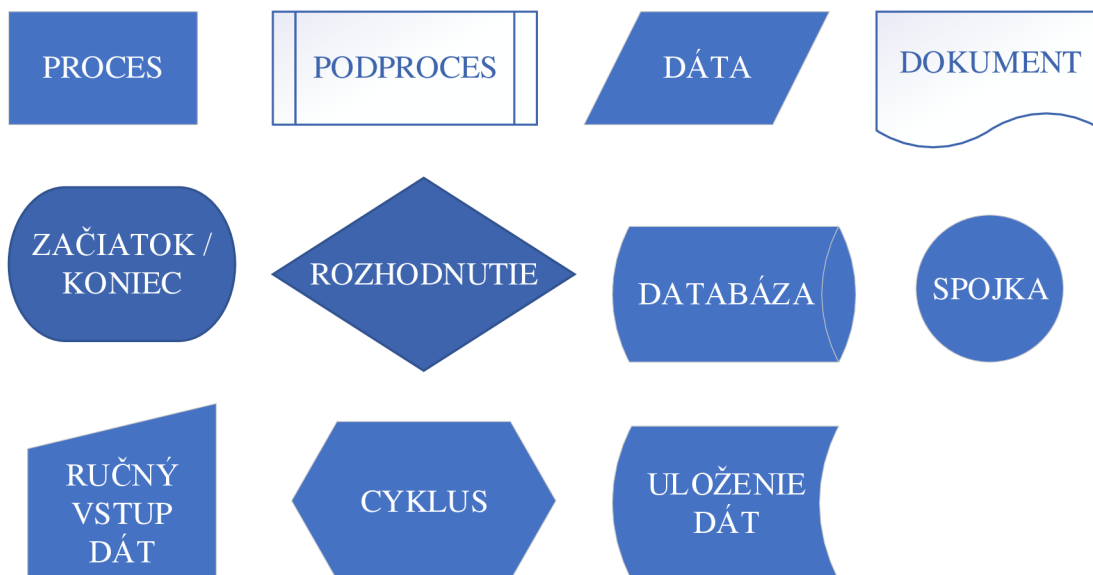
Entita – objekt v okolí systému, s ktorým proces komunikuje. Môže to byť napríklad užívateľ alebo organizačné miesto (8).

Uloženie dát – dátový súbor, doklad, zostava. Dátová pamäť je pasívny objekt pre uloženie dát na neskoršie spracovanie (8).

Dátový tok – presun dát z jednej časti systému do druhej. Reprezentuje prechodné hodnoty v priebehu spracovania. Rovnaké dáta môžu mať rozdielny význam (8).

1.7.4 Vývojový diagram

Vývojový diagram patrí spoločne s DFD diagramom k najpoužívanejším. Jeho nespornou výhodou je možnosť zachytiť veľmi dobre vetvenie spracovania podľa splnenia či nesplnenia požadovaných podmienok (8).



Obrázok č. 12: Symboly vývojového diagramu (Upravené podľa 8, s. 90)

1.7.5 Rozhodovacia tabuľka

Táto metóda sa používa k popisu výberu riešenia v závislosti na parametrizácii rozhodovacieho procesu. Rozhodovacia tabuľka je rozdelená na štyri kvadranty – v prvom sú formulované podmienky. V druhom sú uvedené možné varianty odpovedí na tieto dotazy. Tretí kvadrant určuje množinu prípustných riešení a štvrtý priradzuje riešenie k parametrom odpovedí (8).

1.8 Jazyk SQL

Structured Query Language (ďalej len SQL), je štruktúrovaný dopytovací jazyk. SQL je takzvaný jazyk deklaratívny. Zatiaľ čo u imperatívnych jazykoch počítaču hovoríme krok po kroku čo má urobiť, u jazykov deklaratívnych len hovoríme, čo má byť výsledkom a vôbec nás nezaujíma, ako tohto výsledku počítač dosiahne (9).

Vďaka tomu sú databázové dotazy zjednodušené na príkaz typu „Vráť mi 15 užívateľov s najvyšším hodnotením“. Databáze takýto dotaz pochopí, rozloží si ho na svoje inštrukcie a tak ho spracuje. Nám potom len vráti výsledok bez toho, aby sme tušili ako k nemu prišla (9).

Jazyk SQL sa pôvodne volal SQUEL (z anglického Structured English Query Language) a vznikol vďaka spoločnosti IBM s cieľom vytvoriť jazyk, ktorým by sa dalo komunikovať s databázou jednoduchou angličtinou. SQL databáze sa následne rozširovali a ujali. V dnešnej dobe sa prakticky nič iné nepoužíva a i keď má tento jazyk v objektovom programovaní značné nevýhody, firmám sa nechce prechádzať na iný jazyk, aj keď existujú alternatívne riešenia (9).

1.8.1 Príkazy jazyka SQL

Príkazy jazyka SQL by sa dali rozdeliť do týchto skupín:

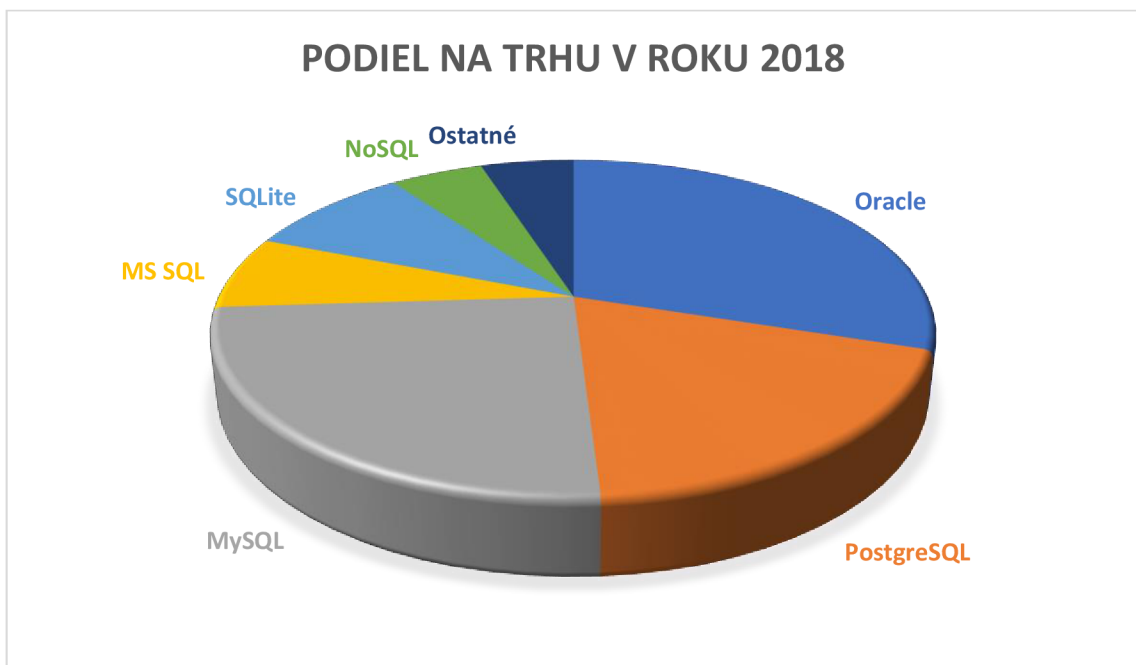
- **DDL** (data definition language) – príkazy patriace do tejto skupiny vytvárajú alebo upravujú štruktúru databáze. Príklady: CREATE, ALTER, DROP atď.,
- **DML** (data manipulation language) – príkazy, ktoré slúžia k získavaniu, ukladaniu a odstráneniu dát v databáze. Príklady: SELECT, INSERT, UPDATE, DELETE atď.,
- **DCL** (data control language) – príkazy pre správu užívateľských rolí a práv. Príklady: GRANT, REVOKE atď.,
- **TCL** (transactional control language) – príkazy pre správu databázových transakcií. Príklady: BEGIN, COMMIT, ROLLBACK atď. (10).

1.9 MySQL databáza

MySQL je relačný databázový systém typu DBMS (Database Management System), ktorý je vlastnený spoločnosťou Oracle. Každá databáza MySQL je tvorená z jednej alebo viacerých tabuliek, ktoré majú riadky a stĺpce. V riadkoch zaznamenávame jednotlivé záznamy. Stĺpce majú meno a uvádzajú dátový typ jednotlivých polí záznamov. Práca s touto databázou, tabuľkami a dátami sa uskutočňuje pomocou dotazov jazyka SQL (11).

System MySQL je použiteľný v:

- ADO.NET,
- C,
- C++,
- JDBC,
- ODBC,
- PERL,
- PHP,
- Python,
- Ruby (11).



Obrázok č. 13: Databázové systémy - podiel na trhu (Upravené podľa 12)

1.10 Apache server

Apache je softwarový server, program, ktorý beží na hardwarovom stroji pripojenom do internetu a zaisťuje obsluhu prehliadačov jednotlivých návštevníkov. Medzi najväčšie výhody Apache serveru patrí najmä dostupnosť pre všetky platformy, ako sú Windows, Linux a podobne a taktiež je vyvíjaný ako open source, čo znamená, že je k dispozícii zadarmo (13).

Spoločne s PHP a MySQL patrí Apache ku takzvanej triáde, teda trojici programov najčastejšie používaných k vytváraniu dynamických stránok. Svoju úlohu v tom hrá i fakt, že inštalácia Apache serveru na domácich počítačoch je pomerne jednoduchá, čím je možné zdarma vytvoriť veľmi kvalitné vývojové prostredie (13).

Okrem rôznych programovacích jazykoch a databázach podporuje Apache server najrôznejšie formy autentizácie, čo umožňuje využívať externé moduly a dokáže vytvárať logy návštevnosti, ktoré sa používajú v pokročilej webovej analýze (13).

1.11 Jazyk HTML a CSS3

Vytvorenie HTML stránky je viac ako jednoduché, ide vlastne iba o textový dokument. HTML sa skladá zo značiek, ktoré sa nazývajú tagy. Tento jazyk umožňuje hlavne dodávať prvkom (elementom) na stránke určitý význam a to je jeho hlavná úloha. V minulosti sa používal i na grafické štylizovanie stránok, ale keďže vzniknuté web stránky boli neprehľadné, bol obmedzený iba na webový obsah (14).

Tagy slúžia k tomu, aby sme nimi mohli obaliť text a tak mu dať určitý význam. Tagy sa píšú do ostrých (lomených) zátvoriek. HTML súbor má určitú štruktúru. Na začiatok každého HTML súboru sa vkladá `<!DOCTYPE html>`, čím sa textový súbor definuje ako HTML dokument (14).

Ďalej sa definuje samotný HTML dokument, ktorý je rozdelený do dvoch častí:

- **hlavička** – obsahuje informácie pre prehliadač a vyhľadávače,
- **telo** – samotný obsah webovej stránky (14).

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>
```

Obrázok č. 14: Štruktúra HTML dokumentu (Upravené podľa 14)

1.11.1 Základné HTML tagy

Medzi základné HTML tagy môžeme zaradiť:

- **odstavce** – ide o párový obalovací tag, ktorý má vo vnútri odstavec `<p>` `</p>`,
- **zvýraznenie textu** – ide hlavne o párové tagy `` a ``,
- **podtrhnutie** – k podtrhnutiu existuje párový tag `<u>`,
- **preškrtnutie** – na tento úkon slúži párový tag `<s>`,
- **optické zvýraznenie** – `<mark>`, slúži k zvýrazneniu časti textu,
- **nadpisy** – nadpis najvyššej úrovne sa zapisuje tagom `<h1>` a nadpis najnižšej úrovne sa zapíše tagom `<h2>` (15).

Medzi ďalšie základné tagy v HTML patria napríklad aj ``, ktorý slúži na vkladanie obrázkov alebo `<a>`, ktorý slúži na pridanie odkazu. Tieto tagy sa ale od predchádzajúcich odlišujú, pretože vyžadujú takzvané atribúty. Atribút v HTML je doplňujúca informácia k tagu. U tagu `` sa jedná o atribút `src`, ktorý definuje cestu k obrázku a `alt`, čo je popis daného obrázku (16).

Často sa stáva, že na stránkach potrebujeme vložiť tabuľku. Tabuľka nám umožňuje vkladať do jej buniek elementy, ktoré sú následne usporiadané. Na rozdiel od odstavcov, ktoré sa vždy skladajú pod seba, v tabuľke môžeme mať text vedľa seba. V bunkách samozrejme môžu byť aj obrázky a ďalšie rôzne elementy. Zaujímavosťou je aj fakt, že v minulosti sa rozloženie, teda usporiadanie (v praxi nazývané ako layout) na webových stránkach tvoril práve pomocou tabuliek. Tento princíp je samozrejme v dnešnej dobe už dávno prekonaný, no určite sa nájde ešte veľké množstvo stránok, ktoré tento princíp používajú (17).

Medzi najdôležitejší HTML tag patrí párový tag `<div>`, ktorý má však iba jeden jediný účel a to obaliť iné HTML tagy. Jeho použitie je úzko spojené s textom vyššie, kedy nastáva problém, ako vytvoriť viacero stĺpcov v obsahu. Najjednoduchšie riešenie je tieto stĺpce zabaliť pomocou tagu `<div>` a použiť CSS štýly. Samozrejme v dnešnej dobe sa na túto problematiku využívajú rôzne knižnice, ako napríklad Bootstrap alebo Material Design (18).

1.11.2 Kaskádové štýly

Kaskádové štýly poznáme pod skratkou CSS (z anglického Cascading Style Sheets). Ide o moderný jazyk, ktorý nám umožňuje účinné formátovanie webových stránok napísaných v jazykoch HTML, XHTML či XML. Charakteristická vlastnosť CSS je, že jednotlivé pravidlá štýlov sa môžu vzájomne prekrývať, čo zvyšuje ich efektívnosť (19).

Ak sa kaskádové štýly používajú správne, umožnia nám celkové oddelenie vzhľadu dokumentu od jeho obsahu (ide o takzvaný bez tabuľkový layout). Toto oddelenie oboch vrstiev zvyšuje prístupnosť webovej stránky a práve v ňom spočíva hlavný rozdiel oproti formátovania pomocou atribútov, čo sa používalo v minulosti (19).

Výhody kaskádových štýlov:

- väčšia možnosť formátovania,
- ľahšia správa väčších prezentácií (CSS šablóny),
- rýchlejšie načítavanie stránky,
- menšia záťaž na servery (19).

1.12 Jazyk PHP

PHP je programovací jazyk, ktorý sa používa prevažne na budovanie webových stránok. Program PHP obvykle nebeží na desktopových počítačoch, ktorý používa len jedna osoba, ale typicky beží na webovom serveri a prostredníctvom webového prehliadača k nemu prístupujú ľudia (20).

Priebeh komunikácie klient-server bez PHP:

- do políčka pre adresu v prehliadači sa napíše `www.example.com/catalog.html`,
- prehliadač cez internet odošle správu počítaču s názvom `www.example.com` a požiada ho o obsah na `/catalog.html`,
- http server Apache dostane túto správu a prečíta súbor `catalog.html` z diskovej jednotky,
- http server Apache odošle obsah požadovaného súboru späť k vášmu počítaču (odpoveď na požiadavku od prehliadača),

- prehliadač spracuje html značky obsiahnuté v zdrojovom kóde stránky a následne ju zobrazí (20).

Priebeh komunikácie klient-server s PHP:

- do políčka pre adresu v prehliadači sa napíše `www.example.com/catalog/yak.php`,
- prehliadač cez internet odošle správu počítaču s názvom `www.example.com` a požiada ho o stránku `catalog/yak.php`,
- http server Apache dostane túto správu a položí enginu PHP (čo je program, ktorý beží na `www.example.com`) otázku, ako vypadá stránka `catalog/yak.php`,
- PHP engine prečíta súbor `yak.php` z diskovej jednotky,
- PHP engine uskutoční príkazy uvedené v `yak.php` (popríklad môže prísť k tomu, že si vymení dáta s nejakým databázovým programom, ako napríklad MySQL),
- PHP engine zoberie výstup z `yak.php` a odošle ho späť http serveru Apache ako odpoveď na jeho požiadavku,
- http server Apache odošle obsah stránky, ktorý dostal od PHP enginu späť vášmu počítaču,
- webový prehliadač spracuje html značky obsiahnuté v zdrojovom kóde stránky a následne ju zobrazí (20).

Výhody jazyka PHP:

- PHP je zdarma (pokiaľ ide o peniaze),
- PHP je zdarma (pokiaľ ide o jazyk),
- PHP je multiplatformový (cross-platform),
- PHP je široko používané,
- PHP je zrodený pre webové programovanie (20).

PHP bez problémov zvládne:

- prácu s textom a číslami,
- logika – rozhodovanie a smyčky,
- zoskupovanie dát – práca s poliami,
- zoskupovanie logiky – funkcie a súbory,
- prácu s objektami,

- webové formuláre,
- uchovávanie informácií – databáze,
- prácu zo súbormi,
- zapamätávanie užívateľov – cookie a relácie,
- komunikáciu s inými weby a zo službami,
- testovanie
- a mnohé ďalšie možnosti (20).

1.13 JavaScript

Vznik tohto jazyka je úzko spojený so spoločnosťou Sun, v ktorej v roku 1992 vznikol jazyk Java. Jeho cieľom je tvoriť prenosné programy medzi rôznymi operačnými systémami a dokonca i rôznymi architektúrami. Na multiplatformový jazyk mal byť taktiež prívetivý syntaxou podobnou jazyku C a mal obsahovať veci, ktoré C++ nemá (21).

Sun videl na webe príležitosť, ale súčasné prehliadače v tej dobe dokázali zobrazovať len statický HTML obsah. Nebolo možné na web vkladať žiadne interaktívne prvky. Sun začal preťažovať JavaApplety pre prehliadače, ktoré umožňovali vkladať do stránok Java aplikácie, bežiace na strane klienta. Spoločnosť Netscape prišla s väčšou interaktívnosťou pomocou jazyka LiveScript, ktorý bol v podstate spojený z jazykov Java, Scheme a Self. Sun chcel, aby sa jazyk volal JavaScript, čo bol čisto obchodný ťah a mal symbolizovať akúsi jednoduchšiu verziu Javy. V podstate ale išlo o úplne nový jazyk, ktorý prebral len časť syntaxi Java, ale jej filozofiu už vôbec nie (21).

Dôležité je si uvedomiť, že JavaScript beží na strane klienta, všetky tieto aplikácie sú teda spúšťané v prehliadači užívateľa. To je obrovský rozdiel oproti serverovým jazykom, ako je napríklad PHP. Pomocou JavaScriptu teda môžeme meniť obsah webovej stránky u užívateľa, napríklad tvorbou dynamických menu, rôznych kontajnerov, ktoré umožňujú šetriť miesto na stránke keď sú zatvorené a na nejakú konkrétnu udalosť sa otvoria. JavaScript je skvelý k formátovaniu textu, pomocou neho si môžeme formátovať text, tak ako je to vo Word (21).

JavaScript sa taktiež používa napríklad na validáciu webových formulárov, pretože odoslanie formuláru na server a čakanie na následnú odpoveď serveru je pomalé. Ak by

sme napríklad napísali zle e-mail, webová stránka nás na to upozorní a nie je potreba načítať stránku znova. Je však potreba si uvedomiť, že keďže JavaScript beží na klientovi, môže si ho užívateľ vypnúť alebo prepísať a preto je potrebné, aby sme túto validáciu uskutočnili aj na serveri (21).

JavaScript je jazyk interpretovaný, je teda prekladaný za behu a vykonávaný podľa svojho zdrojového kódu. Syntax je podobná jazyku C a jazyk je dynamicky typovaný, obsahuje iba jeden numerický typ a to **number**, typ **string** pre text, **boolean** pre pravdivostnú hodnotu a **object** pre čokoľvek iné (21).

1.14 Bootstrap

Bootstrap bol vytvorený spoločnosťou Twitter v roku 2011 a dnes je známy ako najpopulárnejší CSS framework na svete. Bol vytvorený na základe nekonzistencie rôznych aplikácií vo firme Twitter, ich vzhľad bol odlišný a bola nutná znalosť konkrétneho štýlu k ich úprave. Preto začali pracovať na univerzálnom CSS frameworku, ktorý spoločnosť nakoniec uvoľnila ako open-source, čo znamená, že sa môže voľne používať i ku komerčným účelom a nemusí sa to nikde uvádzať (22).

Bootstrap momentálne zaručuje moderný a dobre fungujúci vzhľad pre web, kde je použitý. To najlepšie je však obrovské množstvo voľne dostupných šablón, ktoré sú nápadité, vzhľadovo rozmanité a po menšej úprave slúžia ako dobrý fungujúci základ pre daný web (22).

Bootstrap kopíruje najmodernejšie trendy v oblasti web designu. Medzi najdôležitejšie rozhodne patria:

- **responzivita** – štýly Bootstrap-u sú dokonale prispôbené pre mobilné zariadenia,
- **mobile-first** – framework bol vo verzii 3 kompletne prepísaný, aby podporoval mobile-first prístup,
- **grid** – obsahuje dvanásť stĺpcový grid systém,
- **je zadarmo** – i pre komerčné účely (22).

2. ANALÝZA SÚČASNÉHO STAVU

Cieľom tejto kapitoly bude zanalyzovať súčasný stav spoločnosti a produktu, ktorým sa moja diplomová práca zaoberá. Analýza bude slúžiť ako podklad, z ktorého budem vychádzať pri tvorbe vlastného riešenia, takže je dôležité aby bola dôkladne spracovaná. V prvej časti analýzy popíšem spoločnosť, čo zahŕňa jej základné informácie, organizačnú štruktúru a postavenie na trhu a následne sa bude venovať hĺbkovej analýze produktu spoločnosti.

2.1 Základné informácie o spoločnosti

Tabuľka č. 1: Základné informácie o spoločnosti (Zdroj: vlastné spracovanie)

Obchodné meno	P&T Design Agency, s.r.o.
Sídlo	Nové Mesto nad Váhom
Právna forma	Spoločnosť s ručením obmedzeným
Deň zápisu	26.04.2013

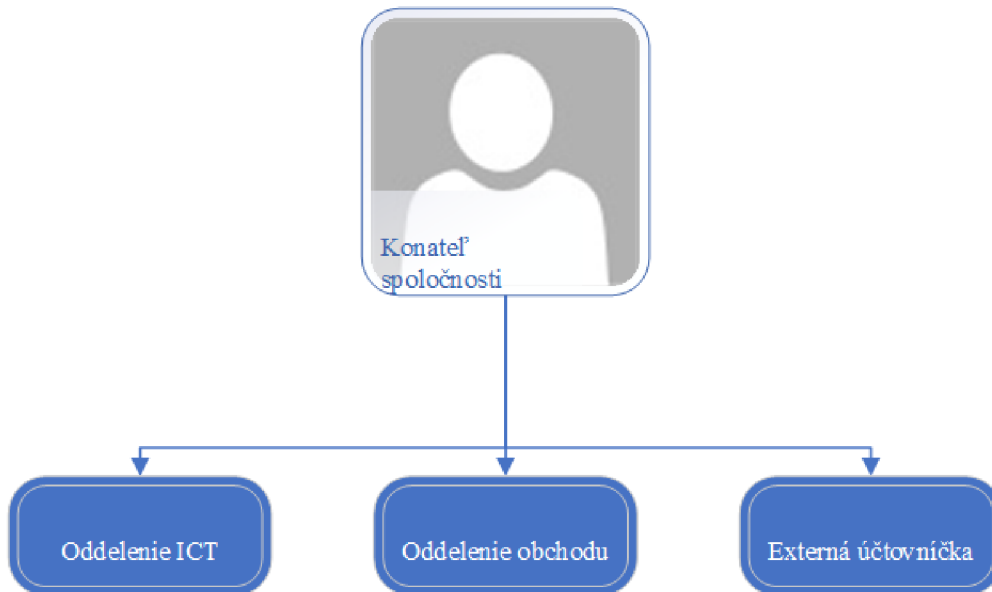
Spoločnosť od roku 2016 podniká v oblasti služieb, konkrétne v **poskytovaní reklamných plôch za využitia bezdrôtových sietí**. Jej zákazníci sa členia do dvoch základných skupín a to podniky, ktoré využívajú wifi routery spoločnosti a inzerenti, ktorí si na týchto routeroch platia reklamu. Jej podiel na trhu je najvýraznejší najmä na západnom Slovensku, kde je minimum spoločností, ktoré by poskytovali obdobné služby. Potrebné je taktiež spomenúť, že v spoločnosti pracuje v súčasnosti 8 zamestnancov plus konateľ a práve títo zamestnanci sú vo väčšine prípadov študentami posledných ročníkov vysokých škôl so zameraním na informatiku, respektíve obchod a služby.

Medzi zákazníkov, ako som vyššie spomínal patria najmä právnické osoby. Podniky, ktoré využívajú služby spoločnosti, teda wifi routery, sú prevažne reštaurácie, bary, kaviarne, športové centrá a mnohé ďalšie podniky, kde sa **združuje väčšie množstvo ľudí**. Naopak medzi inzerentov patria spoločnosti, ktoré majú rozličnú pôsobnosť podnikania.

Ak by som sa pozrel na čísla, tak služby spoločnosti **využíva celkovo 27 zákazníkov**, pričom **16 je inzerentov**.

2.2 Organizačná štruktúra spoločnosti

Organizačná štruktúra je zobrazené na nasledujúcom obrázku:



Obrázok č. 15: Organizačná štruktúra spoločnosti (Zdroj: vlastné spracovanie)

Z obrázku je vidieť, že spoločnosť tvoria dve oddelenia a to konkrétne ICT oddelenie, kde pracuje celkovo 8 zamestnancov, ktorých náplňou práce je vývoj informačného systému spoločnosti a správa týchto IS. Druhé oddelenie tvorí obchodná časť spoločnosti, ktorých náplňou je získavanie nových zákazníkov, komunikácia a udržiavanie kontaktu so súčasnými zákazníkmi.

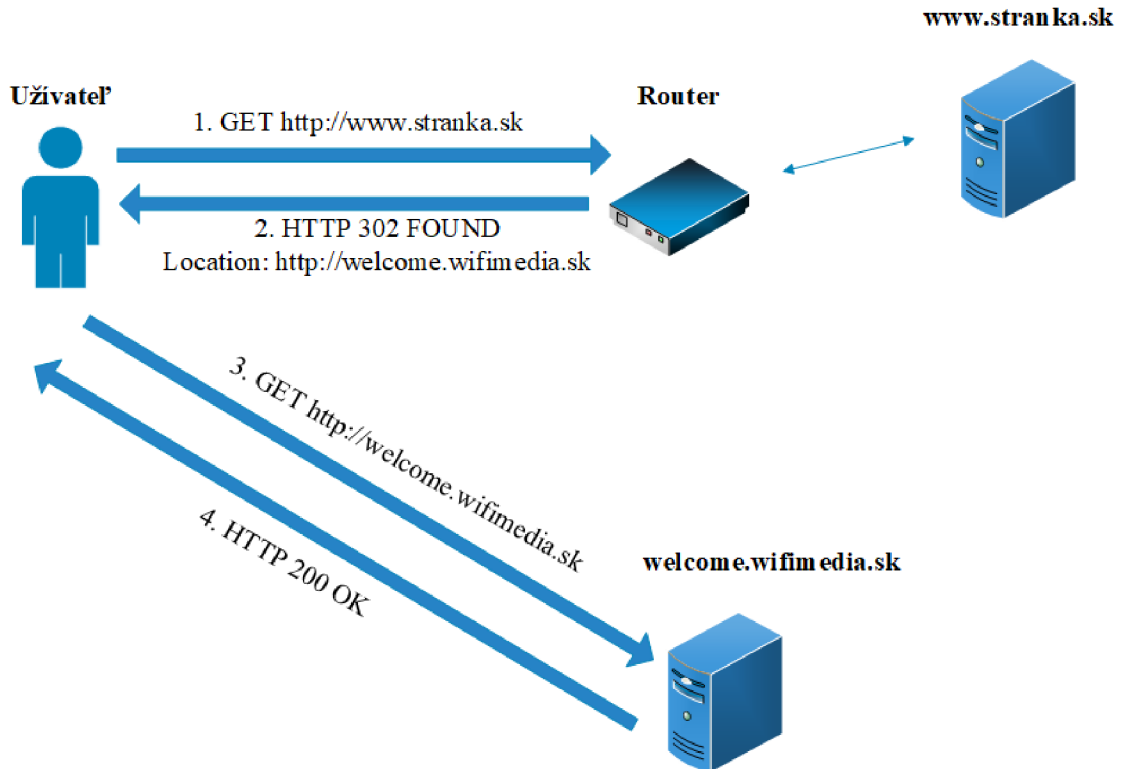
2.3 Produkt WifiMedia

Už z názvu kapitoly je vidieť, že názov produktu, ktorý spoločnosť ponúka je WifiMedia. Využitie tohto produktu je v oblasti poskytovania reklamných plôch za využitia bezdrôtových sietí. Na to, aby bolo možné využiť bezdrôtové siete na reklamné účely je potreba mať wifi router a server, na ktorých beží takzvaný kaptívny portál. Tento portál zabezpečí, aby sa každý užívateľ prihlásil do systému. Bez samotného prihlásenia nie je užívateľovi prístupné prehliadanie internetu.

Užívateľovi, ktorý sa pripojí na wifi sieť sa po otvorení prehliadača zobrazí úvodná stránka, takzvaná welcome page, kde sa mu zobrazí reklama inzerenta. Užívateľ sa môže prihlásiť do systému kliknutím na tlačidlo, ktoré ho presmeruje na takzvanú landing page,

ktorú si zvolí inzerent, vo väčšine prípadov sa jedná o ich domovské stránky, prípadne stránky na Facebooku.

Nasledujúci obrázok zobrazuje komunikáciu medzi užívateľom, routerom a serverom:



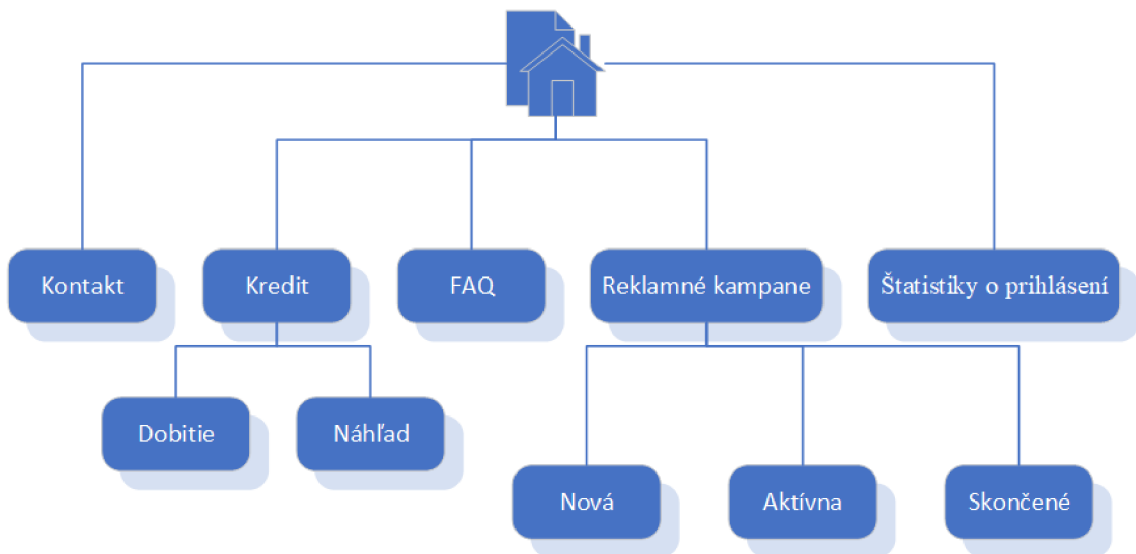
Obrázok č. 16: Komunikácia užívateľa, routeru a serveru (Zdroj: vlastné spracovanie)

Ako som už vyššie spomenul, aby bolo možné využiť bezdrôtové siete na reklamné účely, je potrebný wifi router a server. Samozrejme, že je potreba sprostredkovať informácie zákazníkom, teda podnikom a inzerentom.

2.4 Informačný systém pre zákazníkov

Informačný systém (ďalej len IS), je napísaný v jazyku PHP a využíva databázu MySQL. Prihlásenie prebieha štandardne a to na základe užívateľského mena a hesla. Ku každému užívateľovi je priradený typ účtu, na základe ktorého má prístup k rôznym modulom IS, to znamená, že užívateľ s typom účtu, ktorý je určený pre inzerenta má prístup k iným modulom ako užívateľ, ktorý má typ účtu určený pre podniky.

Hlavným účelom IS je poskytovať zákazníkom informácie, respektíve štatistiky o aktuálne prihlásených užívateľoch, celkových prihlásení, priebehu reklamnej kampane, počte kreditov a samozrejme možnosť založenia nových kampaní.



Obrázok č. 17: Mapa informačného systému (Zdroj: vlastné spracovanie)

2.4.1 Moduly informačného systému

Z predchádzajúceho obrázku sú vidieť všetky aktuálne funkčné moduly, ktoré IS poskytuje svojim zákazníkom.

Modul Kontakt a FAQ – ide o jednoduché moduly, ktoré poskytujú informácie o kontaktoch na našu spoločnosť (hlavne telefón a mail zamestnancov) a o častých dotazoch zo strany zákazníkov.

Modul Kredit – tento modul umožňuje zákazníkovi zobrazit' aktuálny stav kreditu, ktorý má k dispozícii na reklamné kampane, detailne môže užívateľ vidieť v čase ako mu bol kredit strhávaný a samozrejme ďalšou možnosťou je dobitie kreditu, čo môže uskutočniť platobnou kartou prípadne prevodom na účet. Samotné pripísanie kreditu záleží na voľbe, ktorú si užívateľ zvolí, u platobnej karty je to však do pár minút, čo pre zákazníka predstavuje, že pri dobití môže pracovať s kreditom prakticky ihneď.

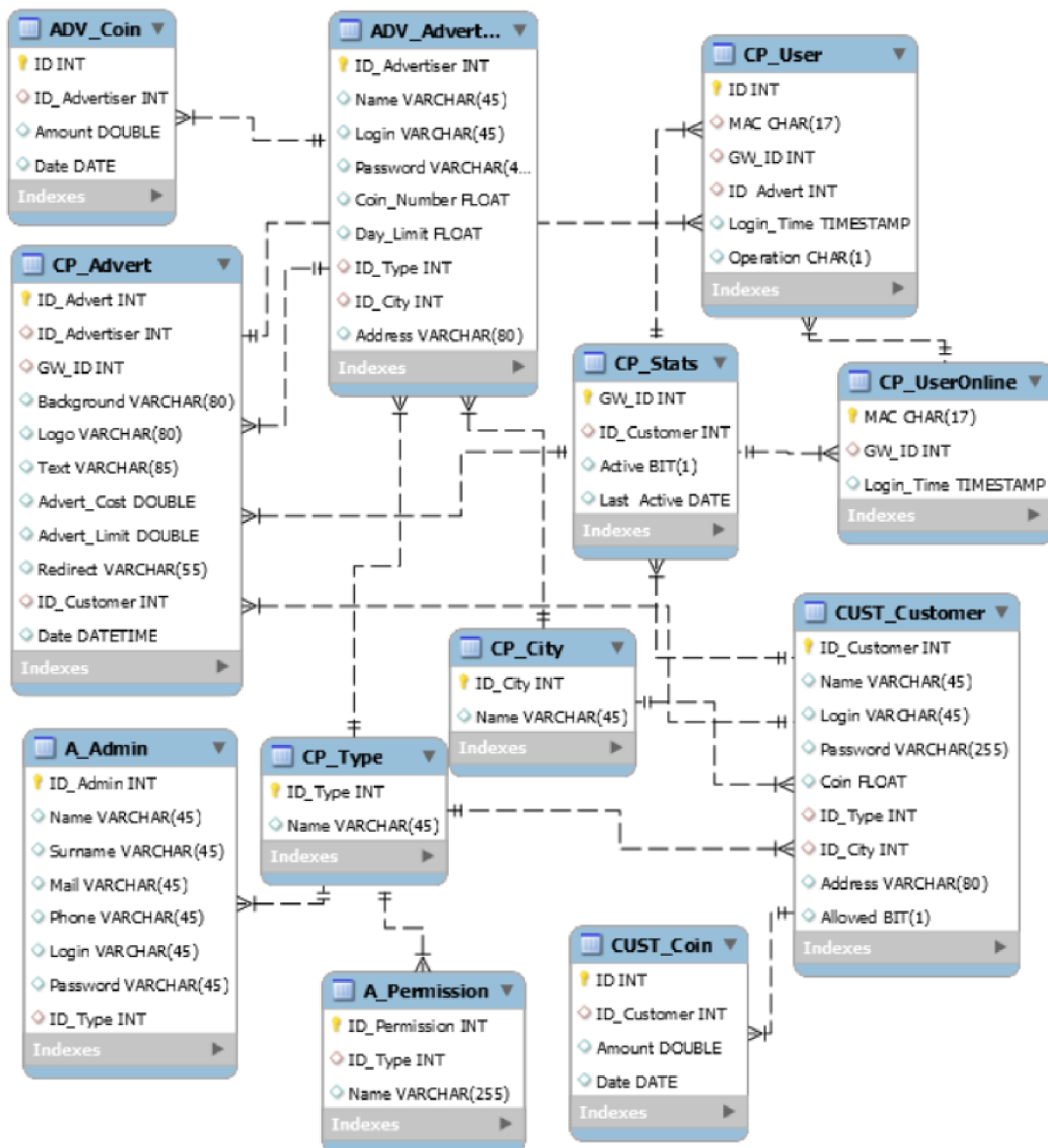
Modul Reklamné kampane – užívateľovi je umožnené prostredníctvom tohto modulu reklamné kampane vytvárať, spravovať si svoje aktívne kampane a v poslednom rade nahliadať na ukončené kampane. Založenie novej kampane je pre zákazníka naozaj

jednoduché, ide o vyplnenie formuláru, kde zadá údaje o kampani ako je text, ktorý sa má zobrazovať na uvítacej stránke, logo a pozadie. Následne musí užívateľ zvoliť podnik, v ktorom sa bude reklama zobrazovať, dátum do kedy sa bude reklama zobrazovať, odkaz na stránku kde bude užívateľ presmerovaný po prihlásení do systému a samozrejme kredit, ktorý sa mu bude strhávať po každom vzhliadnutí uvítacej stránky a prihlásení do systému. Ďalšou možnosťou, ktorú som vyššie spomenul je možnosť náhľadu na aktuálne prebiehajúce kampane. U každej kampane vidí všetky informácie, či už o počte prihlásených užívateľov, počte vzhliadnutí uvítacej stránky a počte kreditu, ktorý ostáva vyčerpať u danej kampane. Samozrejme užívateľovi je umožnené u každej prebiehajúcej kampani si doplniť kredit, prípadne zmeniť podnik, kde sa má reklama zobrazovať a taktiež zmeniť text, logo a prípadne pozadie uvítacej stránky. Poslednou funkciou u tohto modelu je náhľad na ukončené kampane, ktoré môžu byť z dôvodu vypršania termínu alebo vyčerpania kreditu. Užívateľ vidí celkový počet prihlásení do systému a celkový počet vzhliadnutí uvítacej stránky. U každej ukončenej kampane môže užívateľ zvoliť že takto nastavenú kampaň chce spustiť znovu, čo mu je umožnené kliknutím na jedno tlačidlo a následne si len zvolí počet kreditu a termín do kedy má táto kampaň bežať.

Modul Štatistiky o prihlásení – tento modul poskytuje zákazníkovi náhľad na detailné štatistiky všetkých jeho kampaní. Tieto štatistiky sú rozdelené na základe jednotlivých kampaní, ale užívateľ má možnosť vidieť aj celkové počty prihlásení a vzhliadnutia uvítacej stránky naprieč všetkými kampaňami. Jednotlivé prihlásenia sú taktiež rozdelené podľa času a miesta, teda môže vidieť na základe hodín prípadne mesta, ako bola jeho kampaň úspešná.

2.4.2 Databáza informačného systému

Nasledujúce riadky sa budem venovať analýze súčasného stavu databáze informačného systému. Ako som vyššie spomínal tento IS využíva databázu MySQL, ktorá je vhodná pre tento systém.



Obrázok č. 18: E-R diagram databáze (Zdroj: vlastné spracovanie)

Na obrázku vyššie sú vidieť všetky tabuľky, ktoré IS využíva a ich relácie medzi sebou. Každá tabuľka má určitý prefix, aby bolo hneď na prvý pohľad možné rozoznať, ktorá tabuľka ku čomu patrí. Prefix **CP** (z anglického Captive Portal) značí, že tabuľku využíva kaptívny portál, **ADV** (z anglického advertiser) poukazuje, že tabuľka sa týka údajov o inzerentovi, **CUST** (z anglického customer) zasa ukazuje, že sa týka údajov o zákazníkovi, v tomto pojatí ide o podniky, kde sa nachádzajú wifi routery. Posledným prefixom je **A** (z anglického administrator). Tieto tabuľky ešte IS nevyužíva, keďže v súčasnosti nie je vytvorený informačný systém, ktorý by spravoval celý portál. Nasledujúce riadky budem venovať popisu jednotlivých tabuliek a ich atribútov.

Tabuľka CUST_Customer – slúži na uloženie dát o zákazníkovi, teda podniku v ktorom sa nachádza wifi router, teda reklamná plocha.

Tabuľka č. 2: Tabuľka CUST_Customer (Zdroj: vlastné spracovanie)

ID_Customer	INT
Name	VARCHAR(45)
Login	VARCHAR(45)
Password	VARCHAR(45)
Coin	FLOAT
ID_Type	INT
ID_City	INT
Address	VARCHAR(80)
Allowed	BIT(1)

Primárny kľúč v tejto tabuľke je ID_Customer, čo je identifikačné číslo zákazníka, teda podniku. Ide o celé číslo a využíva sa IČO daného podniku. Názov podniku, prihlasovacie meno a heslo sa do databáze ukladajú ako text, ktorý je obmedzený na 45 znakov. Atribút Coin slúži na uloženie kreditu podniku a má nastavený dátový typ na float, čo je desatinné číslo. ID_Type, tento atribút je vlastne cudzí kľúč a spája sa s tabuľkou CP_Type. Ide o celé číslo, ktoré užívateľovi priradí typ účtu. Atribút ID_City ako cudzí kľúč odkazuje na tabuľku CP_City, ide o číselník slovenských miest. Posledným atribútom tejto tabuľky je Address, ukladá sa sem adresa podniku. Keďže si podnik, ktorý využíva router spoločnosti môže určiť, že ho chce využívať výslovne iba pre svoju propagáciu, tak je potreba rozlíšiť podniky, v ktorých môže inzerovať svoju reklamu inzerent, preto je vytvorený atribút Allowed, ktorý má dátový typ BIT a môže nadobudnúť hodnoty 0 alebo 1. Na základe tejto hodnoty sa podnik inzerentovi pri zakladaní reklamnej kampane zobrazí alebo nie.

Tabuľka ADV_Coin – je potreba sledovať u inzerenta jeho dobíjanie kreditu a na to slúži táto tabuľka.

Tabuľka č. 3: Tabuľka ADV_Coin (Zdroj: vlastné spracovanie)

ID	INT
ID_Advertiser	INT
Amount	DOUBLE
Date	DATE

Primárnym kľúčom tejto tabuľky je atribút **ID**, čo je celé číslo a jedná sa o záznamové číslo. Atribút **ID_Advertiser** je cudzím kľúčom a k tomuto záznamu jednoznačne priradí konkrétneho inzerenta. Odkazuje na tabuľku **ADV_Advertiser**. Atribút **Amount** je desatinné číslo a uloží sa sem čiastka za ktorú si inzerent dobil kredit. Samozrejme je dôležité k tomu priradiť aj dátum a na to slúži atribút **Date**, ktorého dátový typ je dátum.

Tabuľka CUST_Coin – keďže systém dovolí podniku využiť reklamnú plochu aj na propagáciu seba samého, je potrebné uchovávať záznamy o dobíí kreditu podniku. Atribúty sú totožné s tabuľkou **ADV_Coin**, rozdiel je samozrejme len v cudzom kľúči, kedy sa priradí k záznamu konkrétny podnik.

Tabuľka ADV_Advertiser – tabuľka slúži na uloženie dát o inzerentovi.

Tabuľka č. 4: Tabuľka ADV_Advertiser (Zdroj: vlastné spracovanie)

ID_Advertiser	INT
Name	VARCHAR(45)
Login	VARCHAR(45)
Password	VARCHAR(45)
Coin_Number	FLOAT
Day_Limit	FLOAT
ID_Type	INT
ID_City	INT
Address	INT

Tak ako aj u predošlej tabuľke, primárny kľúč je identifikátor inzerenta, opäť jeho dátový typ je celé číslo a ukladá sa tu IČO inzerenta. Názov, prihlasovacie meno a heslo je opäť textový reťazec a je obmedzený na 45 znakov. Atribút **Coin_Number** slúži na uloženie celkového počtu kreditu inzerentov. Inzerent si taktiež môže nastaviť denný limit, ktorý je ochotný minúť na všetky reklamné kampane. Pre tento prípad slúži atribút **Day_Limit**, čo je desatinné číslo. Opäť ako u podniku aj u inzerentov je cudzím kľúčom **ID_Type**, ktorý sa odkazuje na tabuľku **CP_Type** a určuje o aký typ užívateľa, respektíve užívateľského účtu ide a taktiež atribúty **ID_City** a **Address** majú rovnakú funkciu ako u tabuľky **CUST_Customer**.

Tabuľka A_Admin – slúži na uloženie dát o administrátoroch, táto tabuľka pribudla len nedávno, bude slúžiť pri návrhu informačného systému pre správcov portálu.

Tabuľka č. 5: Tabuľka A_Admin (Zdroj: vlastné spracovanie)

ID_Admin	INT
Name	VARCHAR(45)
Surname	VARCHAR(45)
Mail	VARCHAR(45)
Phone	VARCHAR(45)
Login	VARCHAR(45)
Password	VARCHAR(45)
ID_Type	INT

Primárnym kľúčom tejto tabuľky je atribút ID_Admin, ide o identifikátor administrátora a dátový typ je celé číslo. Základné informácie o administrátorovi a prihlasovacie údaje sú opäť textové reťazce obmedzené na 45 znakov. Taktiež je potrebné administrátorovi určiť typ účtu, na čo slúži atribút ID_Type. Ide o cudzí kľúč odkazujúci na tabuľku CP_Type.

Tabuľka CP_Advert – najdôležitejšia funkcia pre zákazníkov je vytvorenie reklamnej kampane. Táto kampaň samozrejme so sebou nesie dáta, ktoré je potrebné niekam uložiť. A práve tieto dáta sa ukladajú do tabuľky CP_Advert.

Tabuľka č. 6: Tabuľka CP_Advert (Zdroj: vlastné spracovanie)

ID_Advert	INT
ID_Advertiser	INT
GW_ID	INT
Background	VARCHAR(80)
Logo	VARCHAR(80)
Text	VARCHAR(85)
Advert_Cost	DOUBLE
Advert_Limit	DOUBLE
Redirect	VARCHAR(55)
ID_Customer	INT
Date	DATETIME

Primárny kľúč tejto tabuľky je nastavený na atribút ID_Advert, čo je číselný identifikátor reklamnej kampane. Cudzí kľúč ID_Advertiser odkazuje na tabuľku ADV_Advertiser, čím prepojí konkrétneho inzerenta s konkrétnou kampaňou. GW_ID je atribút, ktorý jednoznačne identifikuje router, ide taktiež o cudzí kľúč, ktorý odkazuje na tabuľku CP_Stats. Atribúty Background a Logo majú dátový typ varchar, uloží sa sem odkaz, kde

sa nachádza pozadie a logo. Taktiež je potrebné určiť sumu, z ktorej sa bude strhávať za prihlásenie do portálu, respektíve za vzhliadnutie reklamy. Na to slúži atribút `Advert_Cost`. Ďalšou možnosťou inzerenta si nastaviť denný limit, za ktorý je ochotný denne zaplatiť reklamu. Na uloženie tohto limitu poslúži atribút `Advert_Limit`. Ako som vyššie spomínal portál sa skladá z úvodnej stránky a po prihlásení užívateľa presmeruje na takzvanú landing stránku. Url adresa tejto stránky uchováva v sebe atribút `Redirect`. Ak nastane situácia, že podnik nepovolí reklamu inzerentovi, teda sa rozhodne propagovať sám seba, tak v tom prípade sa ku reklamnej kampane nepriradí ID inzerenta, ale uloží sa ID podniku. Pre tento prípad slúži atribút `ID_Customer`, samozrejme že sa jedná o cudzí kľúč, ktorý odkazuje na tabuľku `CUST_Customer`. Taktiež je nutné nastaviť dátum ukončenia reklamnej kampane. Tento dátum sa ukladá do tejto tabuľky a slúži na to atribút `Date`, ktorý má dátový typ `datetime`, čiže je nutné nastaviť aj presný čas ukončenia kampane.

Tabuľka CP_Stats – tak ako o podnikoch a inzerentoch sa ukladajú dáta, tak je potrebné ich uložiť aj o wifi routeroch.

Tabuľka č. 7: Tabuľka CP_Stats (Zdroj: vlastné spracovanie)

GW_ID	INT
ID_Customer	INT
Active	BIT(1)
Last_Active	DATE

Každému routeru je potrebné priradiť identifikátor, aby ich bolo možné od seba odlišiť. Tento identifikátor sa ukladá pod názvom `GW_ID` a jedná sa opäť o číselný identifikátor. Samozrejme je potrebné mať aj prehľad, ktorý router sa nachádza u ktorého zákazníka a na to slúži prepojenie tabuliek na základe atribútu `ID_Customer`. Nutnosťou pre správu je samozrejme vedomosť, či daný router je aktívny alebo neaktívny. Ak je router neaktívny, tak u konkrétneho záznamu sa u atribútu `Active` objaví hodnota 0, v opačnom prípade sa objaví hodnota 1. Dôležité je aj vedieť pri neaktívnom prvku vedieť, kedy bol naposledy aktívny a na to slúži atribút `Last_Active`, ktorý uchováva dátum poslednej komunikácie.

Tabuľka CP_City – jedná sa o jednoduchý číselník, ktorý uchováva číselný identifikátor mesta a jeho úplný názov.

Tabuľka č. 8: Tabuľka CP_City (Zdroj: vlastné spracovanie)

ID_City	INT
Name	VARCHAR(45)

O každom meste na Slovensku si uchováваме jeho úplný názov, pre ktorý slúži atribút Name. Ide o textový reťazec, ktorý je obmedzený na 45 znakov. Ku každému mestu je priradený celočíselný identifikátor.

Tabuľka CP_Type – u každého užívateľa je potrebné si viesť záznam o aký typ účtu ide. Jedná sa opäť o klasický číselník, ktorý obsahuje dva atribúty.

Tabuľka č. 9: Tabuľka CP_Type (Zdroj: vlastné spracovanie)

ID_Type	INT
Name	VARCHAR(45)

Každý typ účtu sa identifikuje na základe celého čísla, atribút ID_Type je teda primárnym kľúčom tejto tabuľky. Aby bolo jasne definovaný typ, je potreba uložiť textovú hodnotu. Preto je vytvorený atribút Name, ktorý ukladá názov účtu a je obmedzený na 45 znakov.

Tabuľka A_Permission – pri administrátorských účtoch je potrebné rozlišovať, či sa jedná o bežného správcu alebo ide o hlavný administrátorský účet. Každý z týchto typov účtov má nejaké privilégia, respektíve právomoci. Aby boli jasne definované, založila sa tabuľka A_Permission.

Tabuľka č. 10: Tabuľka A_Permission (Zdroj: vlastné spracovanie)

ID_Permission	INT
ID_Type	INT
Name	VARCHAR(255)

Každé oprávnenie je opäť nutné jednoznačne identifikovať. Na to slúži atribút ID_Permission, ktorý je primárnym kľúčom tabuľky a na základe dátového typu je vidieť, že sa jedná o celočíselný identifikátor. Ďalej je potreba konkrétne oprávnenie priradiť ku konkrétnemu typu užívateľského účtu. Pre tento prípad je vytvorený atribút ID_Type, cudzí kľúč odkazujúci na tabuľku CP_Type. Ako som vyššie spomínal každé oprávnenie

musí mať jasne definovaný názov. Tento názov sa uloží ku každému záznamu a slúži na to atribút Name.

Tabuľka CP_User – aby bolo možné strhávať kredit za každé vzhliadnutie reklamy a prihlásenie do portálu je potrebné uchovávať dáta o pripojených užívateľoch, v tomto prípade sa pod týmto pojmom myslí zariadenie.

Tabuľka č. 11: Tabuľka CP_User (Zdroj: vlastné spracovanie)

ID	INT
MAC	CHAR(17)
GW_ID	INT
ID_Advert	INT
Login_Time	TIMESTAMP
Operation	CHAR(1)

Primárnym kľúčom tejto tabuľky je ID záznamu a jedná sa o celé číslo. Každé zariadenie je identifikované pomocou MAC adresy, ktorá sa uloží do tabuľky. Pre uchovanie tohto údaju sa využíva atribút MAC a ide o textový reťazec. Tento atribút je cudzím kľúčom a odkazuje na tabuľku CP_UserOnline. Ďalej je potreba zistiť, na ktorý router sa zariadenie pripojilo a tento údaj uložiť do tabuľky. Každý router je identifikovaný podľa celočíselného identifikátora, ktorý sa uloží do atribútu GW_ID. Ide taktiež o cudzí kľúč, ktorý sa vzťahuje na tabuľku CP_Stats. Ak už vieme, ktoré zariadenie sa pripojilo ku ktorému routeru, je potrebné taktiež viesť záznam o tom, ktorá reklama sa mu zobrazila, aby bolo možné strhnúť kredit tejto reklame. Atribút ID_Advert spája tabuľku s CP_Advert, teda je jednoznačne zistiteľná reklamná kampaň. Atribút Login_Time uchováva dátum a čas, kedy sa zariadenie pripojilo. Posledný atribút s názvom Operation identifikuje, či sa užívateľ prihlásil do portálu alebo len vzhliadol welcome stránku.

Tabuľka CP_UserOnline – uchováva záznam o zariadeniach, ktoré sú aktuálne pripojené do portálu.

Tabuľka č. 12: Tabuľka CP_UserOnline (Zdroj: vlastné spracovanie)

MAC	CHAR(17)
GW_ID	INT
Login_Time	TIMESTAMP

Po každom vzhliadnutí reklamy a prihlásení do portálu sa zariadenie uloží do tabuľky CP_User. Je potrebné, aby sa uchovával aj záznam o aktuálne prihlásených užívateľoch. Primárnym kľúčom je MAC adresa zariadenia. Taktiež na základe hodnoty u atribútu GW_ID je možné zistiť, v ktorom podniku je prihlásený užívateľ. Poslednou hodnotou, ktorá sa uloží do tabuľky je čas prihlásenia. Portál sa „pýta“ zariadenia, či je stále aktívne, ak mu nepríde žiadna odpoveď, záznam sa z tabuľky CP_UserOnline zmaže.

2.5 SWOT analýza informačného systému

Tabuľka č. 13: SWOT analýza informačného systému (Zdroj: Vlastné spracovanie)

SILNÉ STRÁNKY	SLABÉ STRÁNKY
<ul style="list-style-type: none"> • Jednoduchosť používania • Unikátnosť na trhu • Dostupnosť • Spoľahlivosť systému • Náklady na správu systému 	<ul style="list-style-type: none"> • Spravovanie systému • Zabezpečenie • Nedôvera užívateľov otvoreným wifi sieťam • Geografické pokrytie
PRÍLEŽITOSTI	HROZBY
<ul style="list-style-type: none"> • Expanzia na trhu • Zvýšenie počtu zákazníkov • Rast tržieb 	<ul style="list-style-type: none"> • Počítačová gramotnosť užívateľov • Posun v oblasti technológií • Príchod konkurencie na trh • Vyššie náklady na mzdy zamestnancov

Medzi silné stránky informačného systému určite patrí jednoduchosť používania, čo je pre užívateľov systému veľmi priaznivé. Ďalšou výhodou je dostupnosť informačného systému. Užívateľ sa môže pripojiť odkiaľkoľvek, samozrejme pokiaľ má prístup k internetu a taktiež sa môže pripojiť akýmkoľvek zariadením. Výhodou je takisto, že na trhu momentálne nie je riešenie, ktoré by ohrozovalo spoločnosť. Nesporné výhody plynú aj z toho, že náklady na správu systému nie sú vysoké. Za posledné obdobie sa nezaznamenalo, že by informačný systém vykazoval chyby, ktoré by užívateľa odradili od používania. Naopak medzi najslabšie stránky systému v súčasnosti patrí určite spravovanie systému, keďže nie je vytvorená webová aplikácia pre administrátorov. Taktiež tu určite je treba spomenúť nedôveru užívateľov k otvoreným wifi sieťam a s tým plynúce bezpečnostné riziká. Určite za zmienku stojí aj geografické pokrytie v rámci trhu.

Naopak sa tu ale otvára príležitosť expanzie do iných regiónov s čím súvisí zvýšenie počtu zákazníkov a rast tržieb. Treba brať v úvahu ale aj hrozby. Posun v oblasti technológií je enormný a nikto nevie zaručiť do budúcnosti využitie portálu. Určite spoločnosť bude musieť rátať aj so zvýšenými nákladmi na mzdy, pokiaľ študenti zostanú pracovať v spoločnosti aj po ukončení štúdií. V neposlednom rade počítačová gramotnosť užívateľov je nie vždy na dostatočnej úrovni.

2.6 Požiadavky na informačný systém pre administrátorov

Medzi najväčší problém spoločnosti je v súčasnosti určite náročné spravovanie portálu ako celku. Informačný systém, ktorý by zabezpečil úplnú správu portálu, čo zahŕňa informačný systém podnikov a inzerentov a taktiež kaptívny portál, ktorý riadi komunikáciu. Pred samotným návrhom systému je však potrebné dôkladne poznať procesy, ktoré by mal informačný systém zvládať a na základe toho si určiť požiadavky.

V prvom rade by informačný systém mal dokázať **pracovať s dvoma typmi účtov**. Jeden typ účtu by využívali **klasickí radoví administrátori** a druhý typ účtu by bol určený pre **hlavných správcov**. Dôvodom je to, že aj pri spravovaní systému by nemalo byť povolené každému administrátorovi vykonať všetky úkony. Taktiež je nutné, aby do informačného systému by bolo možné **vytvoriť nový účet pre správcu**, keďže klasická registrácia tu nepripadá v úvahu. Čo sa týka podnikov a inzerentov, správca by mal byť schopný **založiť nový účet pre oba subjekty**.

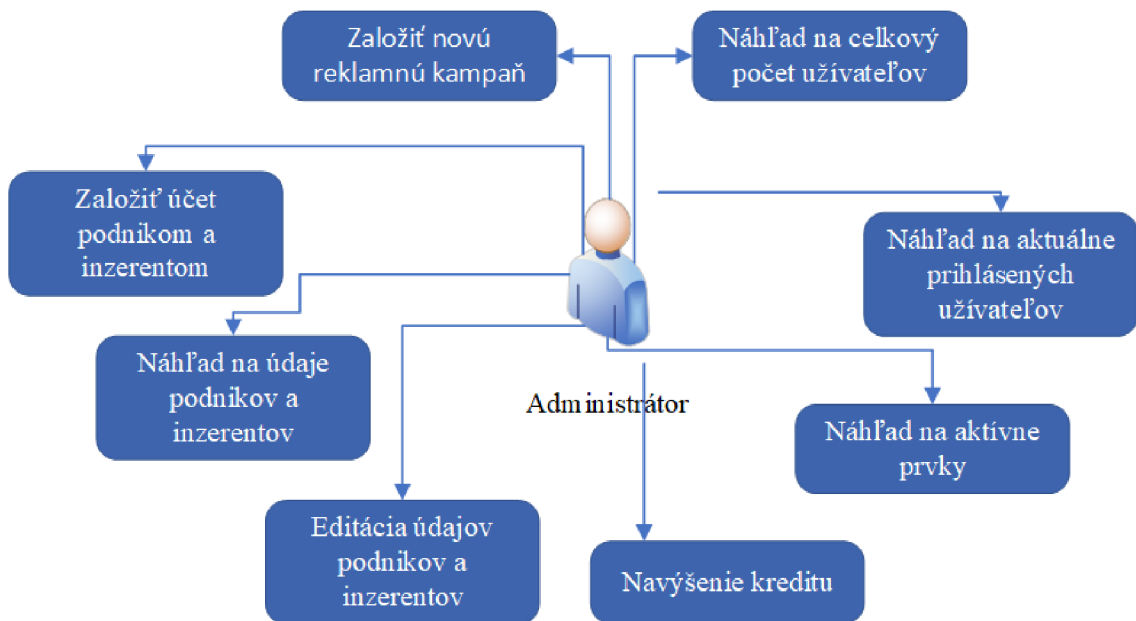
Taktiež je dôležité, aby mal administrátor možnosť **náhľadu a editácie všetkých údajov** o podnikoch a inzerentoch. Správca portálu musí byť schopný pri komplikáciách s dobíjaním kreditu manuálne **zvýšiť kredit** obom subjektom. Dôležitý je aj náhľad na aktívny prvok. Administrátor musí byť schopný vidieť, či **daný wifi router je aktívny alebo nie**. Taktiež je nutné, aby administrátor mohol **zadat' nový wifi router** do portálu a **priradiť ho ku konkrétnemu podniku**.

Administrátor musí taktiež vedieť o počte **aktuálne prihlásených užívateľov** a celkovom počte prihlásených užívateľov. Dôležitá je aj prípadná pomoc inzerentovi, respektíve podniku pri zakladaní reklamnej kampane. Administrátor musí mať náhľad nad **všetkými** vytvorenými **kampanami** a musí byť schopný **založiť novú reklamnú kampaň**. V prípade vyskytnutia problémov, musí byť správca schopný **zablokovať**

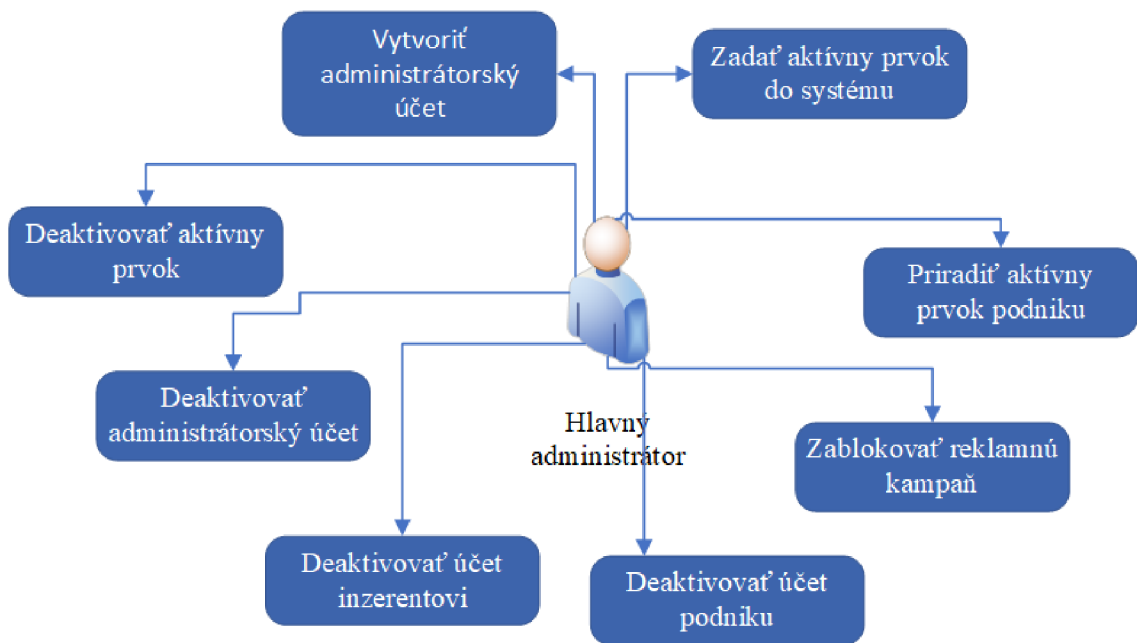
reklamné kampane prípadne deaktivovať účet podnikom, inzerentom, administrátorom a deaktivovať aktívny prvok v podniku.

2.7 Oprávnenia administrátorov

Z predchádzajúcej kapitoly vyplynuli požiadavky na informačný systém administrátorov. Ako som vyššie spomínal, nie každý administrátor bude schopný vykonať všetky typy úkonov, respektíve procesov, ktoré informačný systém ponúka. Nasledujúce dva obrázky ukazujú oprávnenia administrátorov na jednotlivé procesy.



Obrázok č. 19: Oprávnenie administrátorského účtu (Zdroj: vlastné spracovanie)



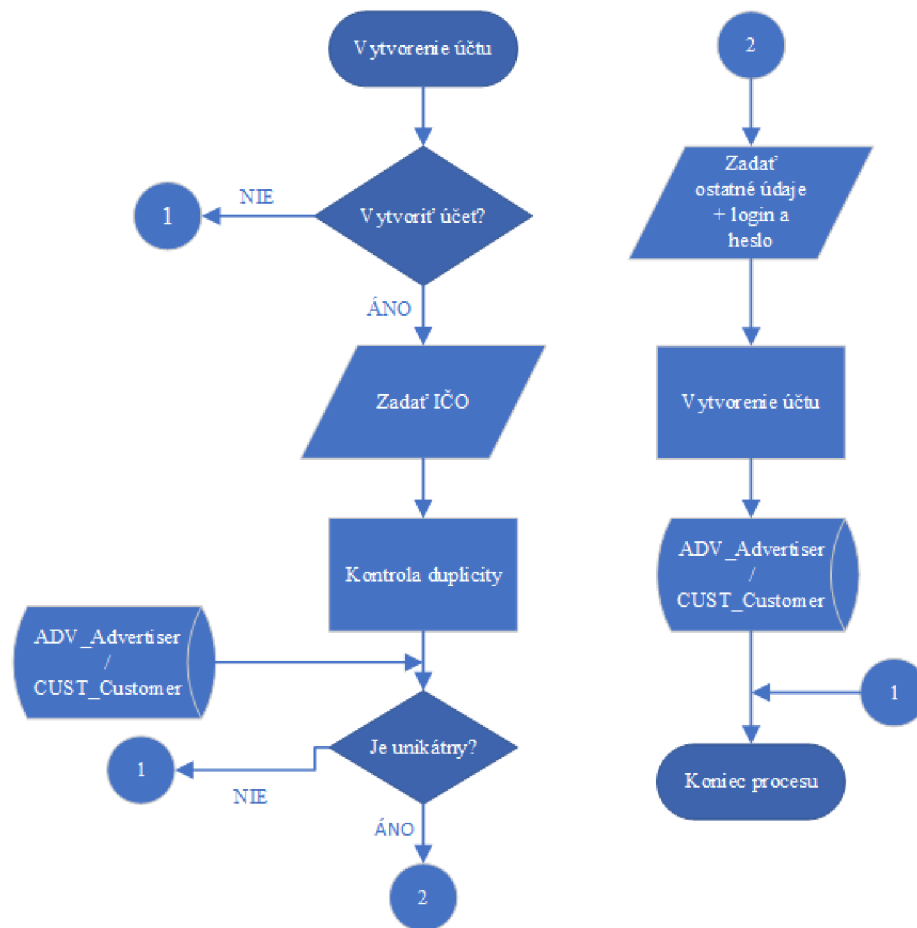
Obrázok č. 20: Oprávnenie účtu hlavných administrátorov (Zdroj: vlastné spracovanie)

2.8 Procesy informačného systému

Z predchádzajúcich kapitol vyplynuli požiadavky na informačný systém pre administrátorov a taktiež ich oprávnenia. Poslednou dôležitou úlohou pred samotnou tvorbou informačného systému, je každý jeden proces detailne rozpracovať, najlepšie vizuálnou formou. Táto úloha je dôležitá pre programátora, aby vedel, čo daný proces predstavuje. Jedna z najlepších metód je spracovať procesy prostredníctvom diagramov.

2.8.1 Založenie účtu podnikom a inzerentom

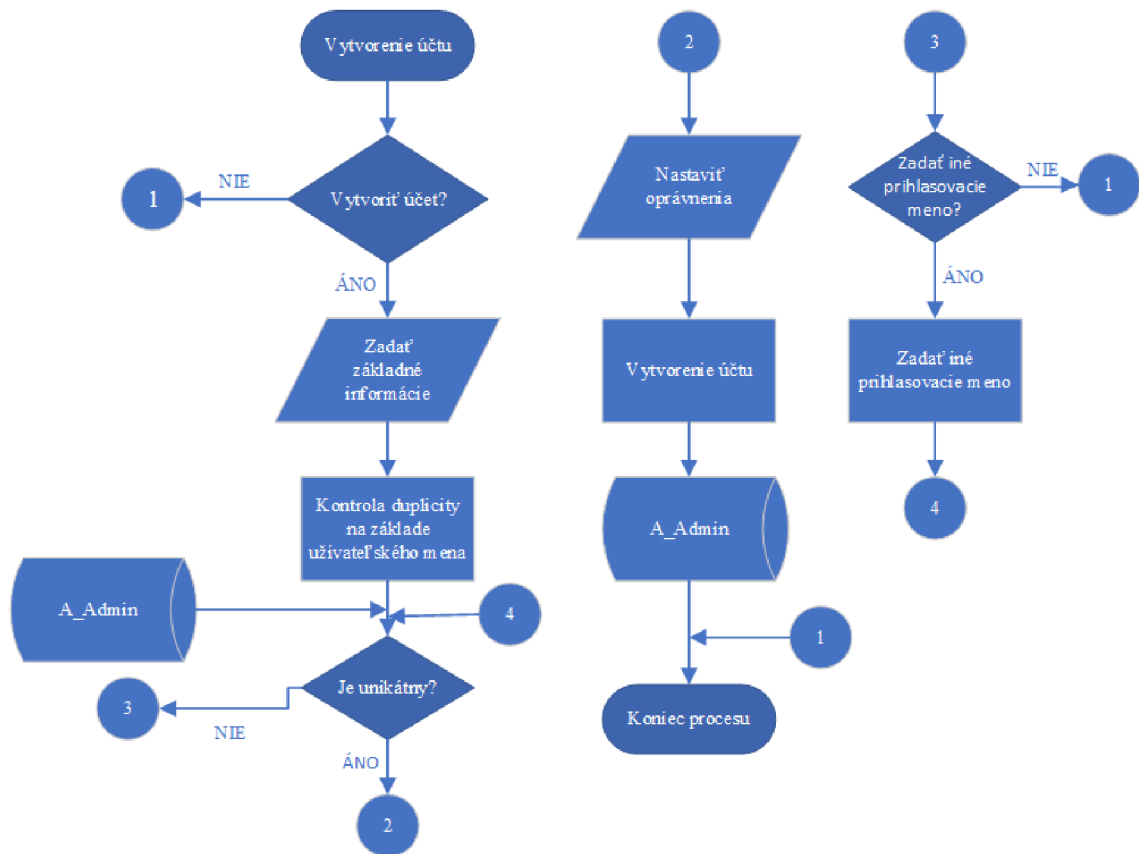
Aby mohol užívateľ vôbec mať prístup k informačnému systému, je dôležité zistiť základné údaje o subjekte a založiť mu účet. Tento proces je zachytený prostredníctvom vývojového diagramu na obrázku nižšie. Z diagramu je vidieť, že u prvku vývojového diagramu, ktorý značí databázu sú zapísané dve tabuľky. Kontrola duplicity a uloženie dát prebehne na základe typu účtu do príslušnej tabuľky.



Obrázok č. 21: Vytvorenie účtu zákazníkovi (Zdroj: Vlastné spracovanie)

2.8.2 Založenie administrátorského účtu

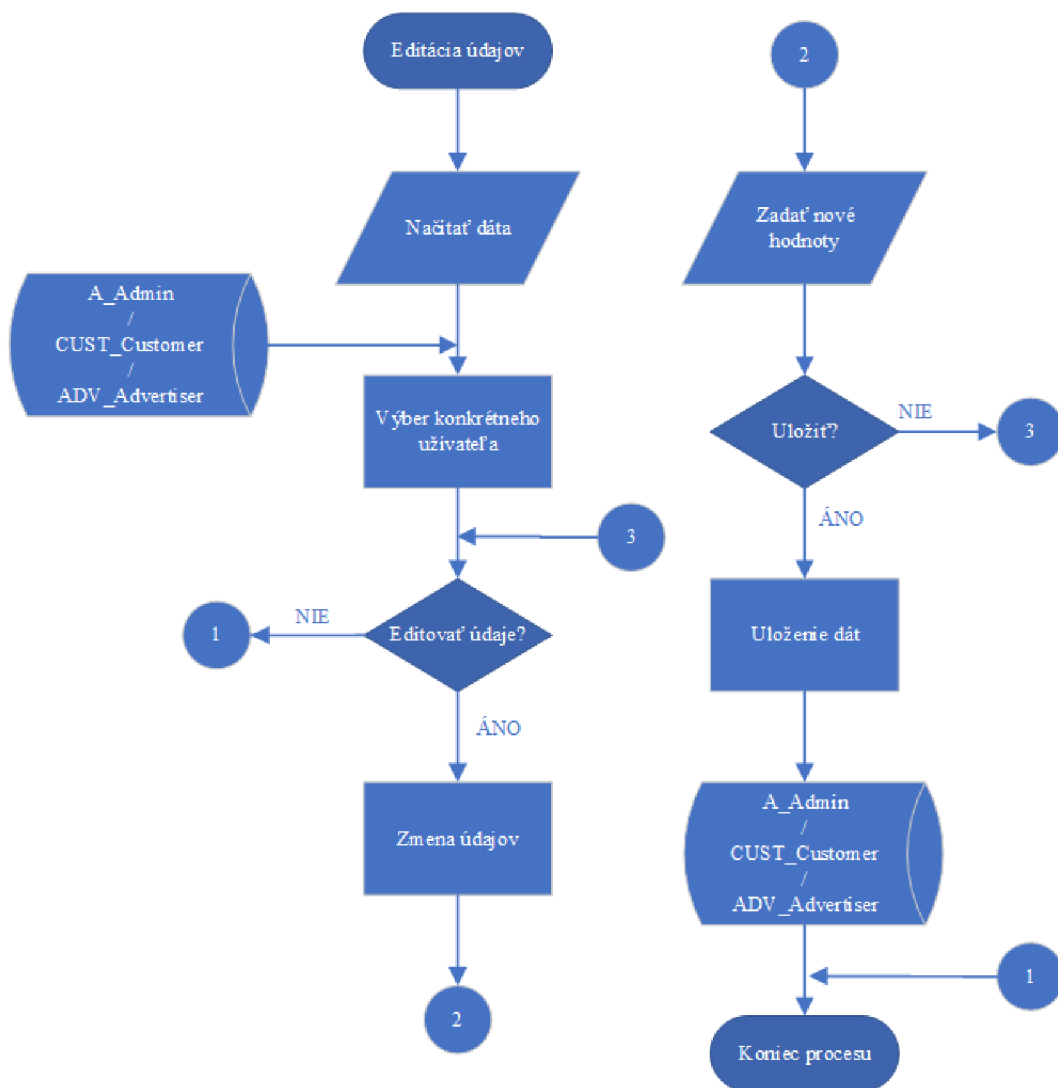
Tak ako je potreba na začiatok založiť účet zákazníkovi, taktiež pri príchode nového administrátora je potrebné mu založiť účet do informačného systému a priradiť mu oprávnenia.



Obrázok č. 22: Vytvorenie administrátorského účtu (Zdroj: Vlastné spracovanie)

2.8.3 Náhľad a editácia údajov o užívateľovi IS

Ďalším dôležitým procesom v informačnom systéme je náhľad a editácia údajov založeného užívateľského alebo administrátorského účtu. V bežnej praxi nastávajú situácie, kedy je potrebné zmeniť základné informácie, prípadne zmeniť prihlasovacie meno, respektíve heslo. Administrátor musí byť schopný tieto úkony uskutočniť.

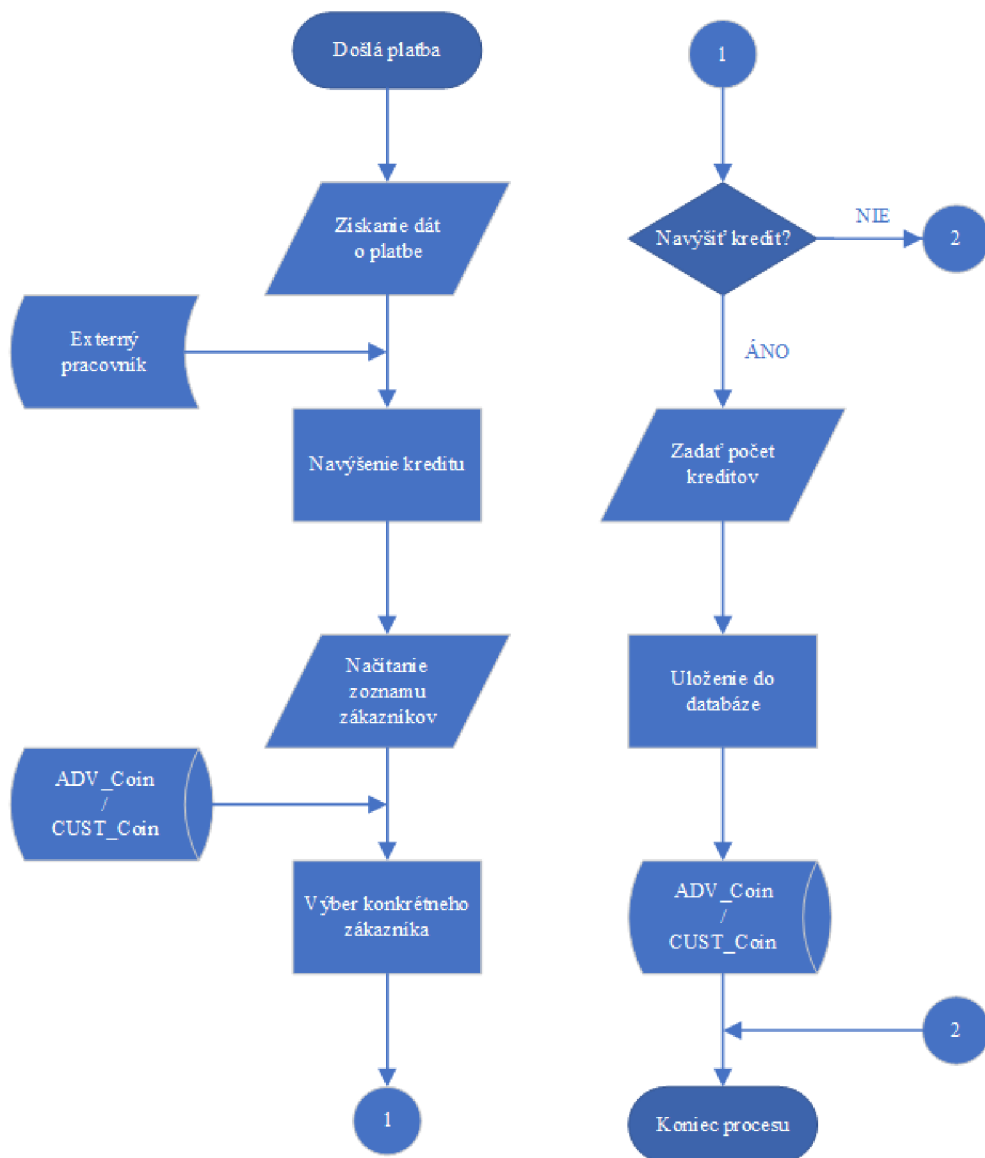


Obrázok č. 23: Náhľad a editácia údajov (Zdroj: Vlastné spracovanie)

Z predchádzajúceho obrázku je vidieť, že som proces náhľadu na dáta o užívateľoch informačného systému (podniky, inzerenti a administrátori) a proces editácie týchto údajov zjednotil do jedného vývojového diagramu. Je logické, že je možné tieto dva procesy zakresliť v jednom, keďže na samotnú editáciu dát je potrebné, aby prebehol proces zobrazenia dát o konkrétnom užívateľovi. Taktiež u prvku, ktorý značí databázu vo vývojovom diagrame sú vidieť celkovo tri tabuľky. Samozrejme, že načítanie a následné uloženie neprebehne vo všetkých troch tabuľkách, ale na základe typu užívateľa sa dáta načítajú a uložia do príslušnej databáze. Je zbytočné, aby tieto procesy boli rozdelené do troch, teda pre každý typ užívateľa, keďže samotné procesy sú rovnaké.

2.8.4 Navýšenie kreditu

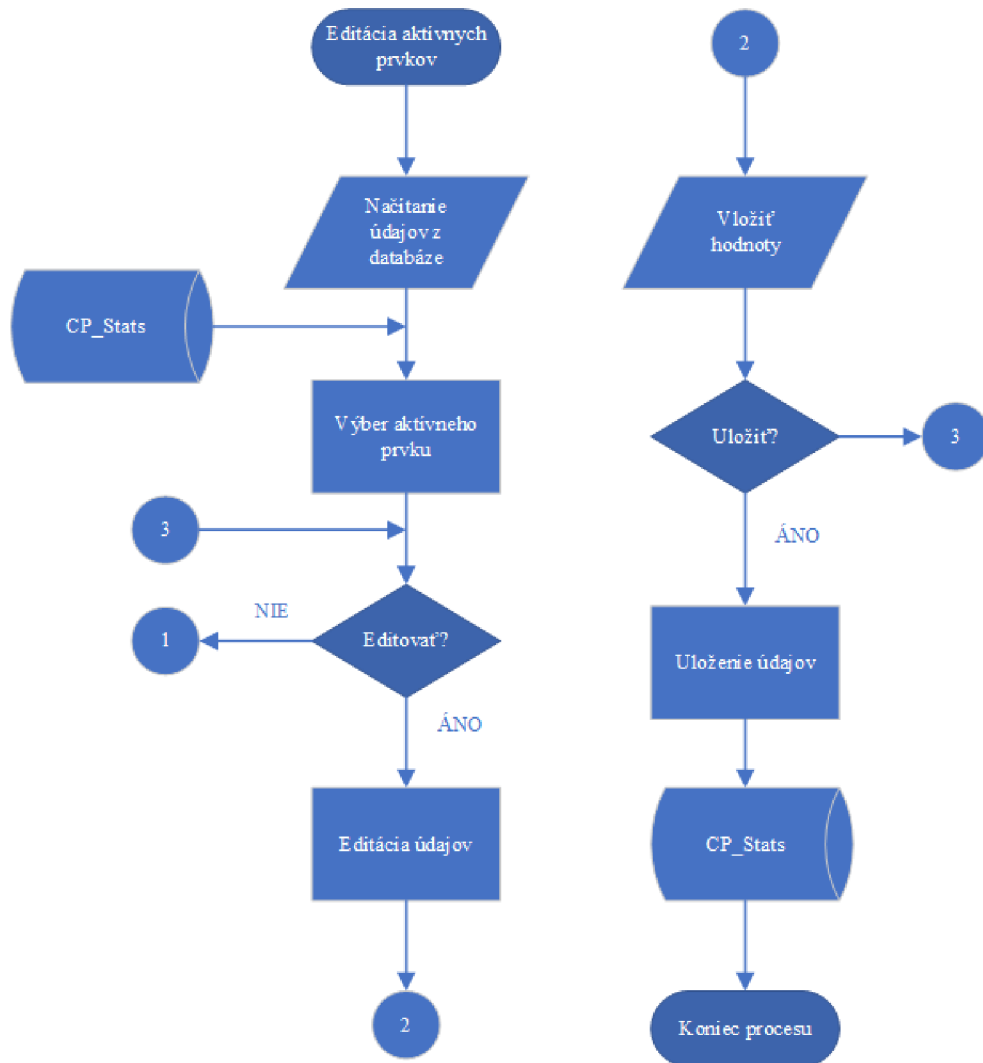
Vo väčšine prípadov si zákazníci pri dobití kreditu zvolia možnosť platby kartou. Informačný systém na základe prebehnutej online platby kartou dokáže pripísať kredit konkrétnemu zákazníkovi. Malé percento zákazníkov si však pri platení vyberá voľbu prevod na účet. V tomto prípade je potrebné pri došlých peniazoch na účet, zákazníkovi navýšiť kredit. Nasledujúci obrázok zachytáva práve tento proces.



Obrázok č. 24: Navýšenie kreditu (Zdroj: Vlastné spracovanie)

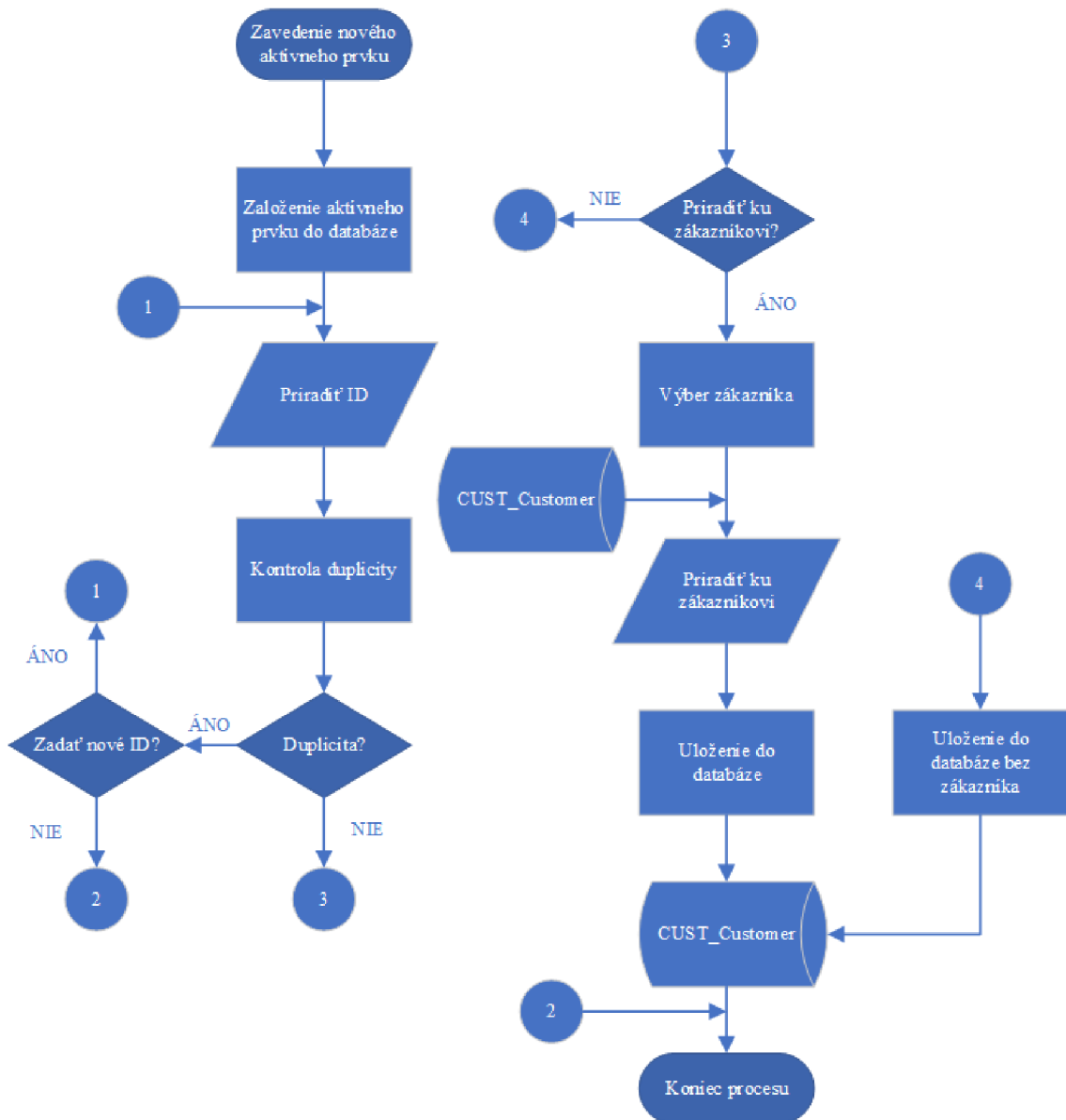
2.8.5 Procesy spojené s aktívnymi prvkami

Z požiadaviek na informačný systém pre administrátorov vyplýva, že je potreba udržiavať aktuálny stav aktívnych prvkov, teda sledovať ich aktivitu, priradiť ku konkrétnemu podniku a prípadne zaviesť do systému nový prvok, respektíve ho odobrať zo systému. Na nasledujúcom obrázku, respektíve diagrame je zakreslený proces náhľadu a editácie aktívneho prvku.



Obrázok č. 25: Náhľad a editácia aktívnych prvkov (Zdroj: Vlastné spracovanie)

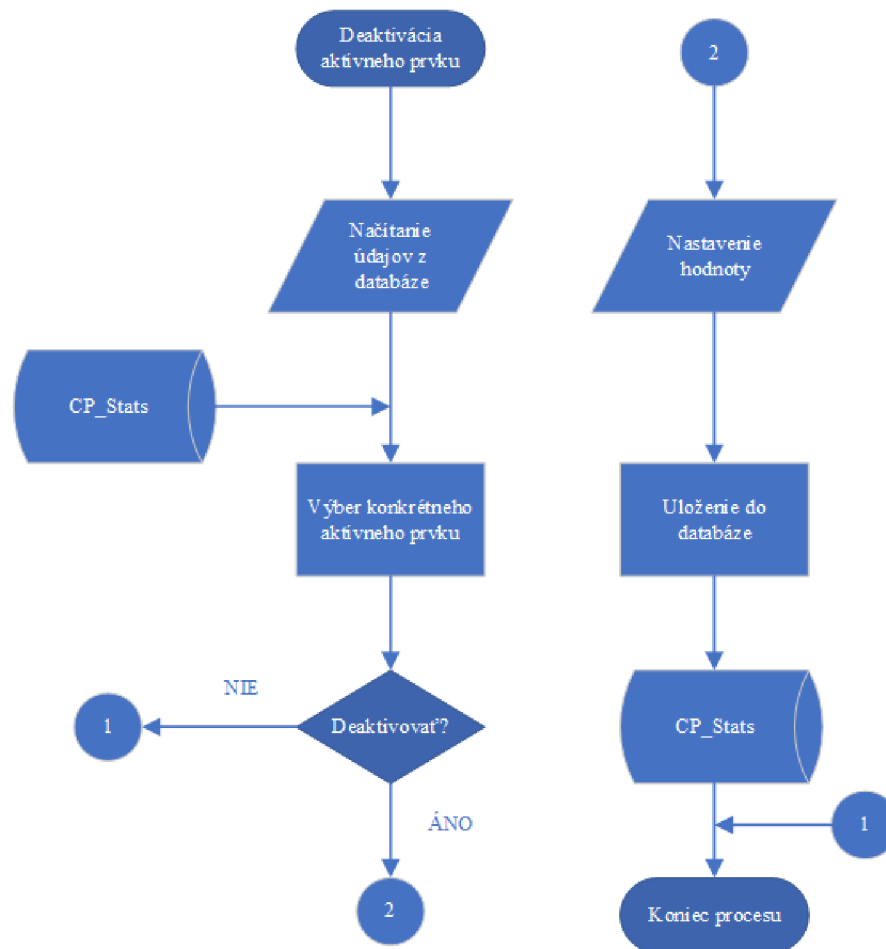
U aktívnych prvkov je však potrebné nie len ich editovať a prípadne uskutočniť náhľad, ale pri nových routeroch ich zaviesť do systému a prípadne priradiť ku konkrétnemu podniku. Tento proces zachytáva nasledujúci diagram.



Obrázok č. 26: Zavedenie aktívneho prvku do systému (Zdroj: Vlastné spracovanie)

Posledným procesom, ktorý sa týka aktívnych prvkov je jeho prípadná deaktivácia. Pokiaľ má aktívny prvok v databáze u atribútu Active hodnotu 1, kaptívny portál na takomto wifi routery zobrazuje reklamu. Ak nastane situácia, že má hodnotu 0, kaptívny portál s prvkom vôbec nekomunikuje. Aby mal aktívny prvok hodnotu u tohto atribútu

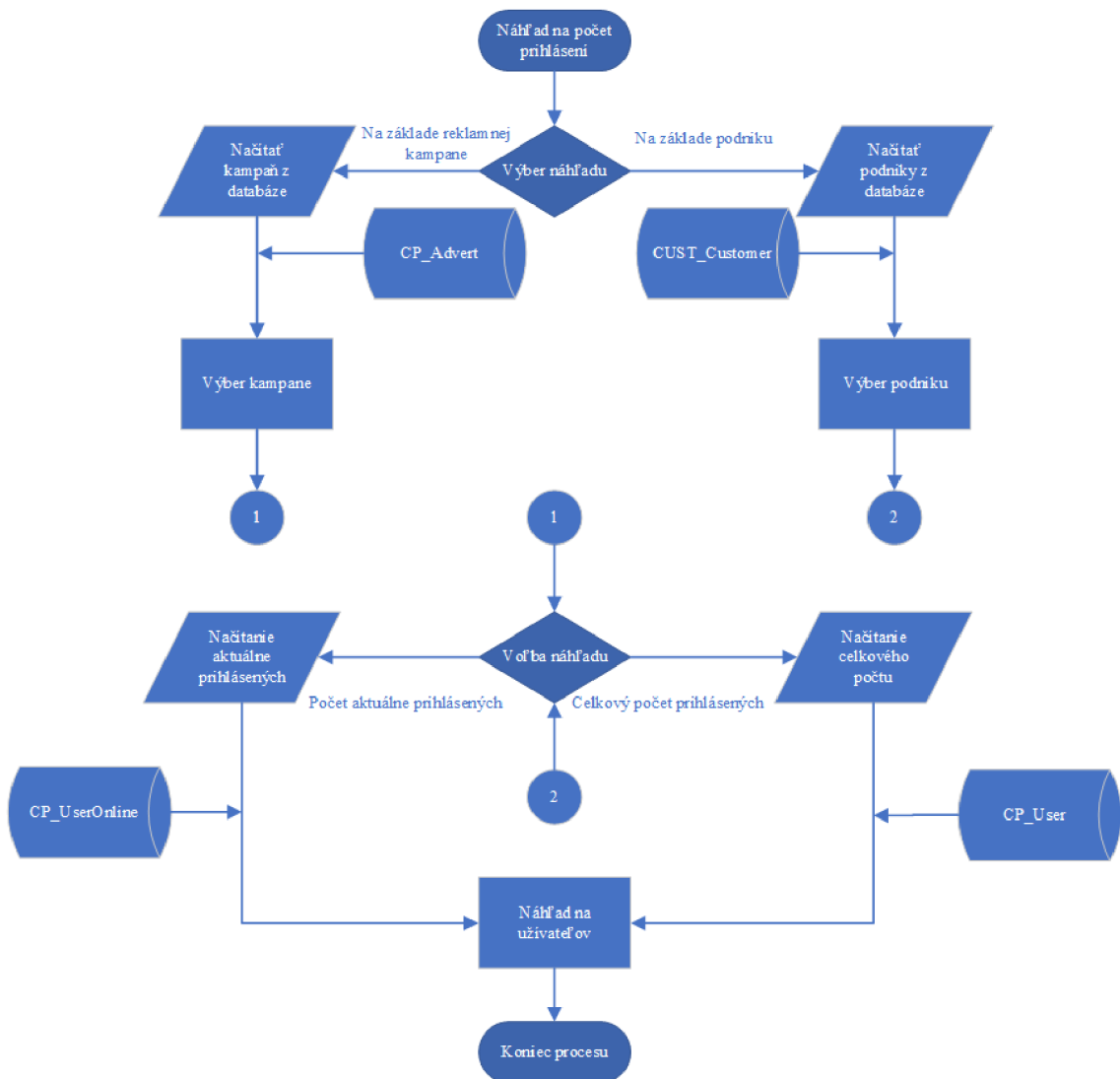
rovnú 0 musí byť buď aktívny prvok buď vypnutý alebo administrátorom deaktivovaný. Samotný proces deaktivácie zobrazuje nasledujúci vývojový diagram.



Obrázok č. 27: Deaktivácia aktívneho prvku (Zdroj: Vlastné spracovanie)

2.8.6 Náhľad na počet prihlásených užívateľov

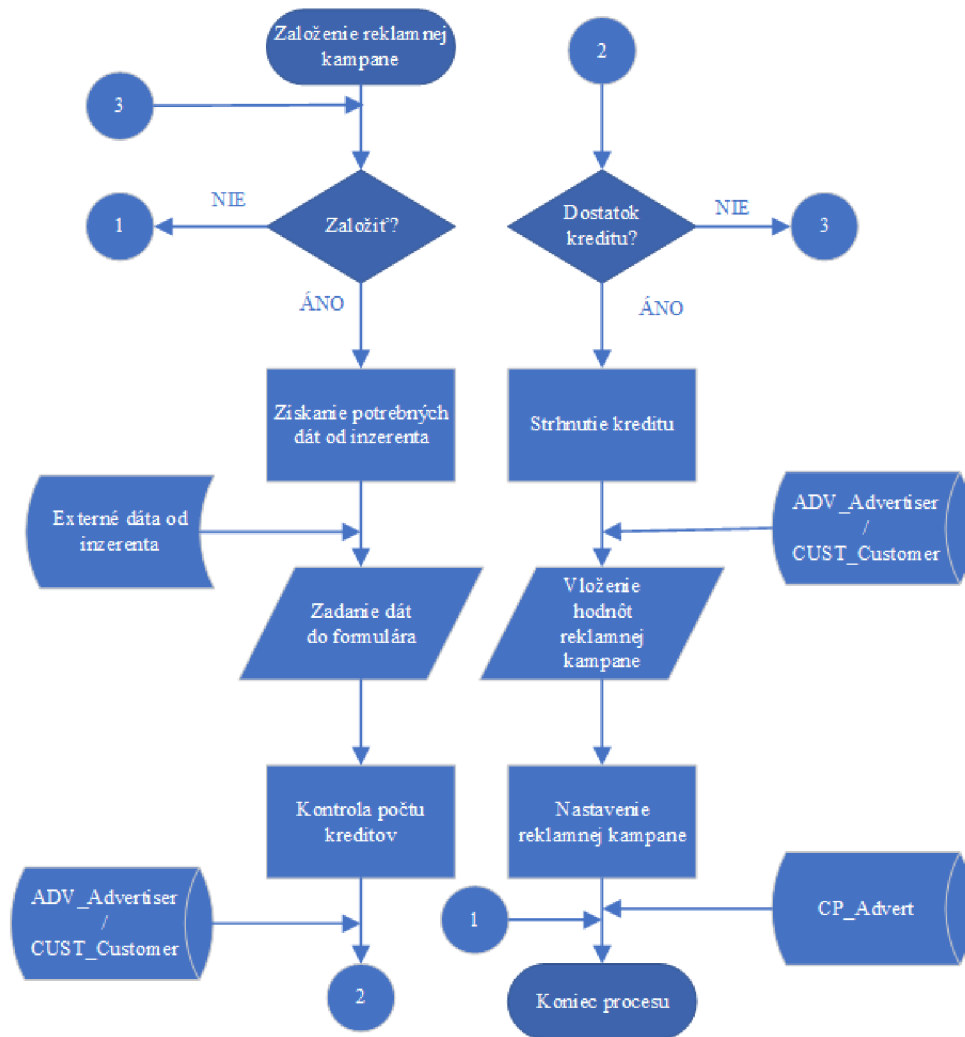
Správca systému by mal mať prístup ku využívaniu reklamných kampaní z čoho vyplýva, že by mal mať možnosť náhľadu nad celkovo prihlásenými užívateľmi portálu, teda nad užívateľmi, ktorý vzhľadli uvítaciu stránku kampane konkrétneho inzerenta. Najlepšou voľbou bude, ak si správca na základe podniku zobrazí celkový počet užívateľov portálu. V tomto procese má možnosť si správca pozrieť celkový počet užívateľov alebo aktuálny stav prihlásených užívateľov v konkrétnom podniku.



Obrázok č. 28: Sledovanie počtu prihlásení (Zdroj: Vlastné spracovanie)

2.8.7 Založenie reklamnej kampane

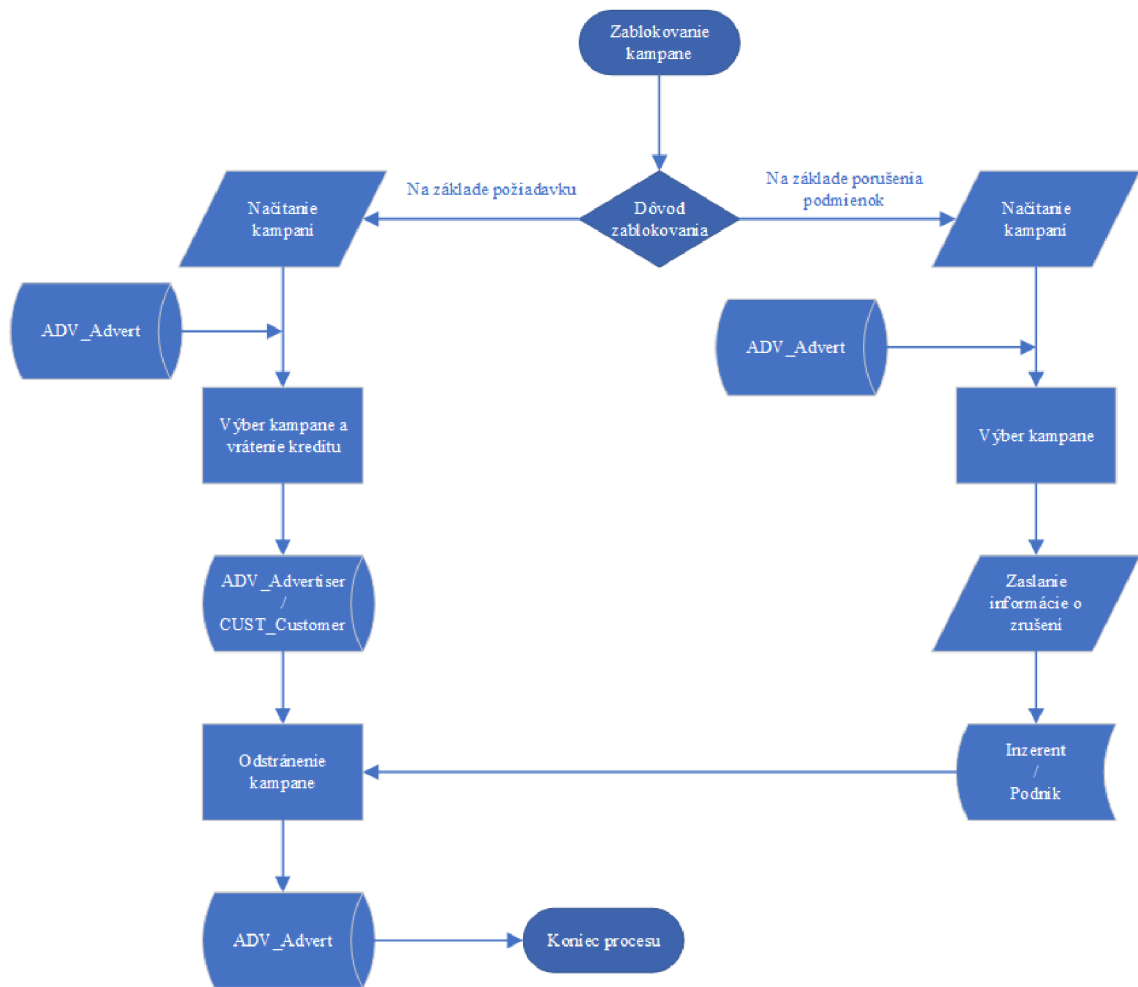
V určitých prípadoch pri určitých inzerentoch musí byť administrátor schopný založiť na základe dát, ktoré mu poskytnú reklamnú kampaň. Ide o vyplnenie jednoduchého formulára, ktorý zaberie pár minút. V súčasnosti sa stále stretávame, že u niektorých zákazníkoch je bariéra v ich počítačovej gramotnosti. Pre to musí byť tento proces zapracovaný do informačného systému pre administrátorov. Proces zachytáva nasledujúci vývojový diagram.



Obrázok č. 29: Založenie reklamnej kampane (Zdroj: Vlastné spracovanie)

2.8.8 Odstránenie reklamnej kampane

Z požiadavkou na informačný systém vyplýva ešte jeden proces spojený s reklamnou kampaňou a to je zablokovanie, respektíve odstránenie reklamnej kampane. Toto zablokovanie môže byť buď na základe požiadavku od inzerenta, respektíve od podniku alebo môže sa vyskytnúť pri porušení obchodných podmienok. Administrátor musí na túto požiadavku, prípadne porušenie podmienok reagovať aktívne zablokovaním reklamnej kampane.



Obrázok č. 30: Zablokovanie reklamnej kampane (Zdroj: Vlastné spracovanie)

2.8.9 Zablokovanie užívateľského účtu

Tak ako v predchádzajúcom prípade, kedy administrátor musí mať prístup ku zablokovaniu reklamnej kampane, to isté platí aj pri užívateľských účtoch. Administrátor na základe porušenie podmienok alebo žiadosti od inzerenta, respektíve podniku musí byť schopný ich účet zablokovať a to tak, aby nebolo možné prihlásenie do informačného systému. Toto zablokovanie však platí nie len u inzerentov a podnikov, ale aj u samotného účtu administrátora. Hlavný administrátor, ako správca s úplným prístupom do informačného systému musí byť schopný zablokovať účet bežnému správcovi. Čo sa týka zablokovaniu účtu hlavného administrátora, tento proces nebude možný vykonať ani z pozície účtu hlavného administrátora. Na takýto úkon bude potrebné zablokovanie uskutočniť priamo v databáze informačného systému, kam ani jeden správca nemá

prístup na manuálne odstránenie záznamu z tabuľky A_Admin. Zablokovanie ostatných účtov zachytáva nasledujúci obrázok.



Obrázok č. 31: Zablokovanie užívateľského účtu (Zdroj: Vlastné spracovanie)

2.9 Zhodnotenie analýzy súčasného stavu

Tak, ako som na začiatku tejto kapitoly avizoval, analýzu som rozčlenil do dvoch hlavných častí. V prvej časti som sa venoval samotnej spoločnosti a v druhej časti zasa jej produktu.

Pri analýze spoločnosti som sa venoval jej základným údajom, zakreslil som jej organizačnú štruktúru a venoval som sa jej postavení na trhu. Z týchto informácií vyplýva, že sa jedná o spoločnosť, ktorej zamestnanci sú tvorení z oblasti študentov, čo má svoje výhody, ale aj nevýhody. Medzi výhody samozrejme patria najmä nižšie náklady na mzdy no na druhej strane som spomenul, že sa jedná prevažne o študentov posledných ročníkov vysokej školy, takže spoločnosť musí rátať, že ak sa niektorí rozhodnú v spoločnosti naďalej pracovať, ich mzdové náklady sa budú zvyšovať. Čo sa týka postaveniu spoločnosti na trhu, tak v regiónoch, v ktorých ponúka svoj produkt nemá konkurenciu, ktorej by sa mala obávať.

V druhej časti som sa venoval samotnému produktu spoločnosti. Zakreslil som komunikáciu kaptívneho portálu so serverom a užívateľom, ďalej som sa venoval modulom, ktorými disponuje v súčasnosti informačný systém, popísal som jednotlivé databázové tabuľky, ich atribúty a príslušné dátové typy. V ďalšej časti som definoval základné požiadavky na informačný systém pre administrátorov, ktorý je v súčasnosti veľmi potrebný a žiaduci. Pred samotnou tvorbou informačného systému je však mať detailný pohľad na jednotlivé procesy, ktoré vyplynuli z požiadaviek. Každý jeden tento proces som spracoval na základe vývojových diagramov.

Z analýzy jednoznačne vyplýva, že informačný systém pre správcov je potrebné spracovať dôkladne z jedného miesta a snažiť náklady stiahnuť na minimum. Na základe analýzy požiadaviek a procesov je možné v návrhovej časti práce vytvoriť informačný systém pre administrátorov portálu.

3. VLASTNÉ NÁVRHY RIEŠENIA

Nasledujúca kapitola sa zaoberá problematikou vlastného návrhu riešenia tvorby informačného systému pre administrátorov. Pri tvorbe návrhu informačného systému budem vychádzať zo spracovanej analýzy, kde využijem už navrhnutú databázu a jednotlivé požiadavky na procesy zapracujem do informačného systému.

3.1 Príprava prostredia

Prvým krokom pri tvorbe informačného systému je príprava prostredia. Informačný systém budem vyvíjať na lokálnom počítači (ďalej používaný výraz localhost), čo vyžaduje inštaláciu softvéru potrebného a využívaného pri tvorbe.

Ako som už v teoretickej časti písal, je potrebné pre vývoj mať nasledujúci softvér:

- Apache server,
- MySQL databázu,
- PHP engine.

3.1.1 Inštalácia MySQL, Apache a PHP

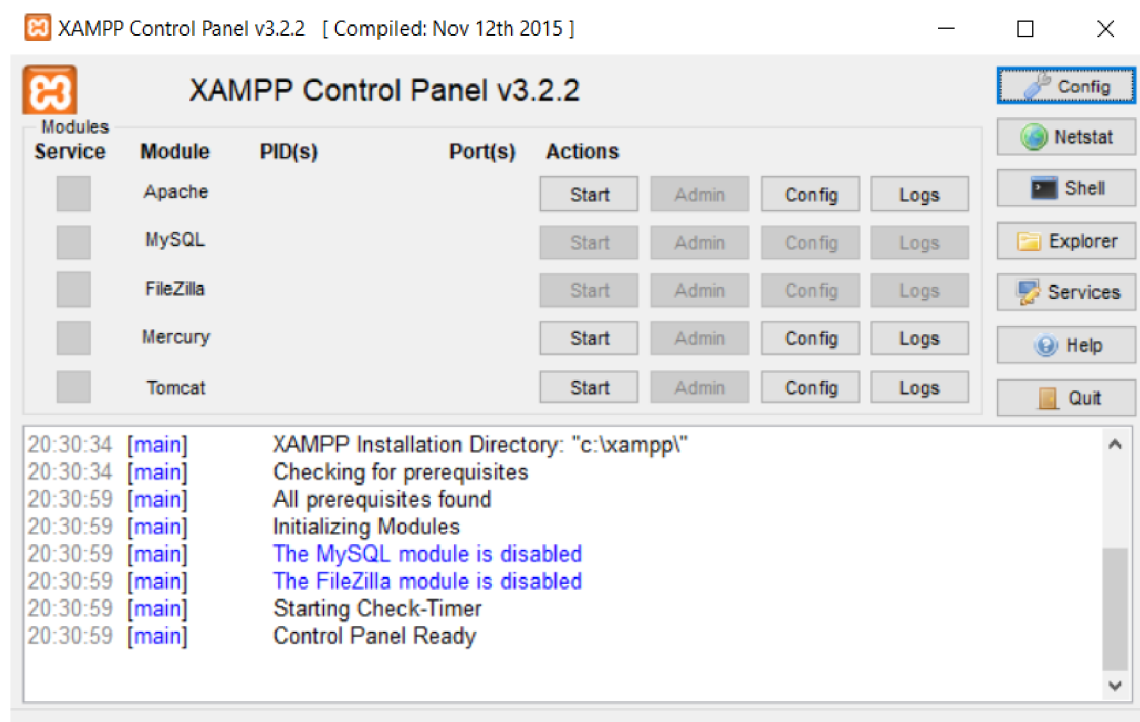
MySQL server je voľne dostupný a dá sa jednoducho stiahnuť z oficiálnych stránok. Aktuálna verzia a verzia, ktorú budem v tomto projekte používať je MySQL 8.0.15. Samotná inštalácia prebieha v pár krokoch a je v celku jednoduchá. Po výbere určených produktov prebehne inštalácia a nasleduje konfigurácia databázového serveru. Vo väčšine prípadov pri konfigurácii ponechám predvolené nastavenia. Novinka, ktorá je v konfigurácii oproti starším verziám databázového serveru je výber autentizačnej metódy, kedy server ponúka dve možnosti:

- Strong Password Encryption (odporúčané),
- Legacy Authentication Method (pozostatok zo starších verzií).

Keďže projekt budem vyvíjať na localhost-u, tak mi postačí klasická autentizačná metóda, no do produkčného prostredia navrhujem určite použiť prvú a odporúčanú metódu. Rozdiel pri týchto metódach je, už ako názov napovedá v šifrovaní hesla, kedy prvá metóda má už zabudované šifrovanie na algoritme SHA 256.

Posledným krokom pri konfigurácii je nastavenie hesla koreňovému (root) užívateľovi. Pre prístup k databáze budem využívať softvér HeidiSQL, ktorý je taktiež voľne dostupný.

Následne pristúpim k inštalácii XAMPP balíčku, ktorý obsahuje Apache server, MariaDB (databázový server), PHP a Perl. Pre potreby tohto projektu z tohto balíčka budem inštalovať iba Apache server a PHP. Inštalácia je taktiež pomerne jednoduchá, po výbere produktov z balíčka sa nastaví cesta, respektíve adresár, odkiaľ sa bude informačný systém spúšťať. Po inštalácii sa pracuje s Apache serverom veľmi jednoducho, otvorí sa takzvaný control panel, z ktorého sa http server spustí.



Obrázok č. 32: XAMPP Control Panel (Zdroj: vlastné spracovanie)

3.1.2 Príprava databáze

Po inštalácii potrebných produktov môžem pristúpiť ku nahratiu poslednej zálohy databáze. Ako prvé spustím skript, ktorý mi vytvorí všetky tabuľky v databáze a následne druhá skript, ktorý mi databázu naplní dátami. Po konzultácii s firmou, sme sa rozhodli, že nebudem uvádzať skutočné názvy inzerentov a zákazníkov, takže poslednou prípravou v databáze je skript, ktorý mi zmení mená u zákazníkov a inzerentov. V projekte ich budem používať pod názvami svetových miest.

Table Name	Size (KiB)
wifimedia	528,0
adv_advertiser	48,0
adv_coin	32,0
a_admin	32,0
a_permission	32,0
cp_advert	48,0
cp_city	128,0
cp_stats	32,0
cp_type	16,0
cp_user	48,0
cp_useronline	32,0
cust_coin	32,0
cust_customer	48,0

Obrázok č. 33: Štruktúra tabuliek v databáze (Zdroj: vlastné spracovanie)

Keďže mám databázu vytvorenú, môžem prísť k samotnej tvorbe informačného systému.

3.2 Prihlasovanie užívateľov do IS

Z praxe poznáme viacero prístupov, ako vytvoriť prihlasovanie užívateľov. Samozrejme, že sa odporúča viac stupňové overovanie, čo znamená, že ku klasickému užívateľskému menu a heslu sa pridáva ešte niečo navyše, napríklad overenie prostredníctvom sms a podobne. Z požiadaviek na prihlasovanie nevyplývala nutnosť mať dvoj, prípadne viac stupňové overenie, takže overenie užívateľa prebehne na úrovni užívateľského mena a hesla. Samozrejme je potreba, aby tento typ prihlasovania bol čo najbezpečnejší.

Ku prihláseniu užívateľa, je potreba vytvoriť formulár, ktorý sa bude odosielať na server spolu s užívateľským menom a zašifrovaným heslom, kde prebehne dotaz do databáze vo forme SELECT-U. Následne prebehne vyhodnotenie, v prípade, že dotaz do databáze vráti 0 nájdených záznamov, tak užívateľ s danými parametrami neexistuje, alebo dotaz vráti 1 nájdený záznam a získam všetky potrebné údaje o užívateľovi.

Čo sa týka formuláru, tak využijem práve v teoretických východiskách spomínaný framework Bootstrap, ktorý ponúka preddefinované šablóny. Výsledný formulár vyzerá nasledovne:



Obrázok č. 34: Prihlasovací formulár (Zdroj: vlastné spracovanie)

Najdôležitejší HTML tag pri tomto formulári je samozrejme `<form>`, kde sa definuje trieda, ktorá sa využije v kaskádových štýloch, metóda, ktorou sa odošle na server, v tomto prípade ide o metódu POST a posledný atribút je `action`, ktorý definuje, v ktorom súbore sa bude spracovávať overenie užívateľa.

```
<form class="login-form" method="POST" action="login.php">
```

Obrázok č. 35: HTML tag formulára (Zdroj: vlastné spracovanie)

Po kliknutí na tlačidlo „Prihlásiť“, sa prejde do súboru `login.php`, kde si odchytním užívateľské meno a heslo do premenných. Pre zabezpečenie tohto typu prihlasovania, využijem spôsob, kedy ku užívateľskému heslu pripojím takzvaný **salt**, čo je textový reťazec a následne celý tento textový reťazec, ktorý obsahuje heslo a salt zahashujem pomocou algoritmu SHA256.

```
// SALT
$salt = '5600715F42BF51C40DC330D750CD996F58FEAD4DDEA56466CE7498D17801B3A5';
$pw_salt = $_POST['password'].$salt;
$user_pwd = strtoupper(hash('sha256', $pw_salt));
```

Obrázok č. 36: Pripojenie salt ku heslu (Zdroj: vlastné spracovanie)

Následne prebehne dotaz do databáze so zadaným užívateľským menom a heslom. Pokiaľ sa vráti nula výsledkov, prihlásenie neprebehlo v poriadku, v inom prípade sa vráti jeden výsledok, ktorý zodpovedá dotazu a tým je užívateľ.

Samozrejme, pred samotným dotazom do databáze, je potreba vytvoriť pripojenie do nej. Keďže toto pripojenie sa v systéme bude vyskytovať často, nemá význam zakaždým samotný kód písať, ale stačí vytvoriť funkciu, ktorú si v prípade potreby zavolám. Samotnú funkciu zobrazuje nasledujúci obrázok.

```
function DatabaseConn() {  
    $servername = "localhost";  
    $username = "root";  
    $password = "123456";  
    $dbname = "wifimedia";  
  
    // Vytvorenie propojenia  
    $conn = new mysqli($servername, $username, $password, $dbname);  
  
    // Kontrola pripojenia  
    if ($conn->connect_error) {  
        die("Pripojenie zlyhalo: " . $conn->connect_error);  
    }  
}
```

Obrázok č. 37: Funkcia na pripojenie do databáze (Zdroj: vlastné spracovanie)

Aby sa po celú dobu prihlásenia zachovali informácie o užívateľovi, je potrebné ich vložiť do session, čo je technológia, ktorá slúži práve pre uloženie užívateľských dát v priebehu komunikácie so serverom.

Po prihlásení užívateľa ho presmeruje na úvodnú stránku informačného systému, ktorá je popísaná v ďalšej kapitole.

Odhlásenie užívateľa prebehne po kliknutí na odkaz v menu, kedy sa spustí PHP kód, ktorú túto session zničí a prebehne presmerovanie na úvodnú stránku.

```
session_destroy();  
header("Location: login_template.php");
```

Obrázok č. 38: Zničenie session pri odhlásení (Zdroj: vlastné spracovanie)

3.3 Úvodná stránka informačného systému

Ako som už spomenul v predchádzajúcej kapitole, po prihlásení presmeruje užívateľa na úvodnú stránku informačného systému. Na úvodnej stránke prihlásený užívateľ vidí menu, informácie o jeho účte a ďalšie informácie.

Medzi tieto informácie patrí:

- Zobrazenie neaktívnych routerov,
- Inzerenti a zákazníci, ktorý majú nízky kredit,
- Celkový počet prihlásení užívateľov do kaptívneho portálu (štatistiky),
- Počet prihlásených užívateľoch.



Obrázok č. 39: Údaje o prihlásenom užívateľovi (Zdroj: vlastné spracovanie)

Údaje o prihlásenom užívateľovi sa zobrazujú na vrchu úvodnej stránky a ukazujú prihlasovacie meno, e-mail a typ účtu prihláseného užívateľa. Ako som už spomínal, v IS je potreba definovať aspoň dva typy užívateľských účtov, a to konkrétne administrátor a hlavný administrátor. Na výpis, o aký typ účtu ide na úvodnej stránke, som napísal jednoduchú funkciu.

```
// Funkcia na vypísanie typu účtu
function GetAdminType($type){
    global $AdminType;
    $AdminType = 0;
    if ($type == 1) {
        $AdminType = 'Administrator';
    }
    elseif ($type == 2) {
        $AdminType = 'Head Administrator';
    }
    else {
        $AdminType = 'Common User';
    }
}
```

Obrázok č. 40: Typ účtu užívateľa (Zdroj: vlastné spracovanie)

3.3.1 Menu systému

Ďalej, ako som spomínal, tak na úvodnej stránke (taktiež na každej) sa zobrazuje menu. Toto menu nie je potreba, aby sa zobrazovalo stále, takže sa užívateľovi zobrazí po kliknutí na html tag ``.



Obrázok č. 41: Predvolené zobrazenie menu (Zdroj: vlastné spracovanie)

Po kliknutí na tento span, sa spustí JavaScriptová funkcia, ktorá menu zobrazí. Samozrejme je viacero možností, ako túto funkcionality zabezpečiť, no najjednoduchším je iba meniť šírku menu.

```
function openNav() {  
    document.getElementById("mySidenav").style.width = "220px";  
}
```

Obrázok č. 42: Javascriptová funkcia na zmenu šírky menu (Zdroj: vlastné spracovanie)

Nasledujúci obrázok zobrazuje menu informačného systému.



Obrázok č. 43: Menu informačného systému (Zdroj: vlastné spracovanie)

Ako je z obrázku vidieť, menu sa skladá z nasledujúcich komponentov:

- routery,
- zákazníci,
- inzerenti,
- reklamné kampane,
- užívateľské účty,
- štatistiky,
- odhlásenie užívateľa.

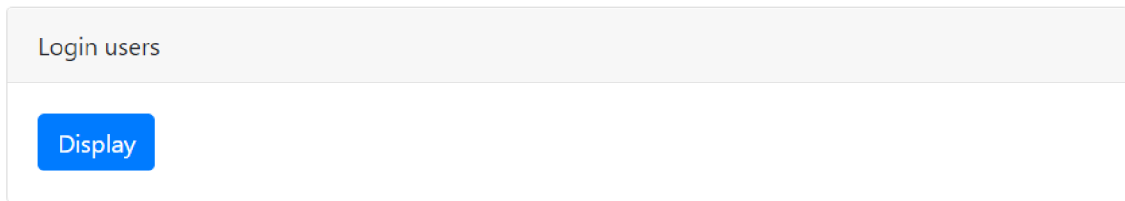
Taktiež je na obrázku vidieť „krížik“, ide o ďalší html tag, ktorý po kliknutí spustí javascriptovú funkciu, ktorá vráti menu do pôvodného zobrazenia, teda zmení šírku danému elementu na 0px.

```
function closeNav() {  
    document.getElementById("mySidenav").style.width = "0";  
}
```

Obrázok č. 44: Vrátenie menu do pôvodného nastavenia (Zdroj: vlastné spracovanie)

3.3.2 Widgety na úvodnej stránke

Pre zobrazenie neaktívnych routerov, inzerentov, ktorý majú nízky kredit, počet celkových prihlásení a počet aktuálne prihlásených užívateľov opäť využijem framework Bootstrap, ktorý ponúka možnosť zobrazenia informácií pomocou takzvaných kariet (v predošlej verzii panelov), ktoré obsahujú hlavičku, telo a v mojom prípade ešte tlačidlo. Samozrejme, že tieto karty (div elementy) sú responzívne, čo znamená, že sa zobrazujú pri rôznom rozlíšení inak. Táto responzivita u tohto frameworku sa volá grid. Nejde o nič iné, ako o stĺpcový layout. Pre zobrazenie na veľmi malých zariadeniach, ako sú mobily, tak sa tieto elementy zobrazujú v jedностĺpcovom layout-e, čo znamená, že sa skladajú pod seba. Pre ostatné rozlíšenia sa zobrazujú v dvojstĺpcovom layout-e, teda dva elementy vedľa seba. Dátami nenaplnený element zobrazuje nasledujúci obrázok.



Obrázok č. 45: Bootstrap card (Zdroj: vlastné spracovanie)

Pre zobrazenie neaktívnych routerov, je potrebné napísať **select** do databáze, ktorý mi vráti dáta o neaktívnych routeroch.

```
SELECT GW_ID FROM cp_stats c, cust_customer cu WHERE  
active=0 AND c.ID_Customer = cu.ID_Customer;
```

Obrázok č. 46: Výber neaktívnych routerov (Zdroj: vlastné spracovanie)

Následne to, čo mi tento dotaz vráti spracujem v PHP. Ak select vráti 0 výsledkov, znamená to, že žiaden router nie je neaktívny. Ak mi vráti viacero výsledkov, tieto výsledky spracujem a zobrazím v konkrétnej Bootstrap-ovej komponente. Na zobrazenie opäť využijem Bootstrap-ové CSS triedy. Výsledky zobrazím pomocou tagov `` a ``, čo naznačuje, že sa bude jednať o zoznam.

```
<ul class="list-group list-group-flush">  
<li class="list-group-item"></li>  
</ul>
```

Obrázok č. 47: Bootstrap zoznam (Zdroj: vlastné spracovanie)

Do vnútra zoznam, na základe počtu vrátených výsledkov select-u vytvorím cyklus, ktorý pre každý jeden záznam vytvorí jeden tag ``, do ktorého vloží záznam z databáze vo forme ID routeru a meno zákazníka, kde sa router nachádza. Tento algoritmus zobrazenia dát na úvodnej stránke budem opakovať taktiež u zobrazenia celkových a aktuálnych prihlásených užívateľov a inzerentoch, ktorý majú nízky kredit. Rozdiel bude samozrejme iba v SQL Query (dopyt). Jednotlivé SQL dotazy sú zobrazené na obrázku nižšie.

```

-- Dotaz na získanie inzerentov s nízkym kreditom
SELECT Name, Coin_number FROM adv_advertiser
WHERE Coin_number < 50;

-- Dotaz na zobrazenie celkových prihlásení
SELECT a.Name, COUNT(u.ID) FROM cp_user u, adv_advertiser a, cp_advert ad WHERE
a.ID_Advertiser = ad.ID_Advertiser AND
ad.ID_Advert = u.ID_Advert AND
u.Operation = 'R';

-- Dotaz na zobrazenie aktuálne prihlásených
SELECT a.Name, COUNT(uo.ID) FROM cp_useronline uo, adv_advertiser a, cp_advert ad WHERE
uo.ID_Advert = ad.ID_Advert AND
ad.ID_Advertiser = a.ID_Advertiser;

```

Obrázok č. 48: SQL dotazy na zobrazenie údajov na úvodnej stránke (Zdroj: vlastné spracovanie)

Výsledné naplnené elementy dátami z databáze vyzerajú nasledovne:

All sign-ins	
Antonio : 3412	Phoenix : 811
Tuscon : 1967	Dallas : 607
Omaha : 1907	Honolulu : 577
Riverside : 1275	Mesa : 345
Seattle : 1047	Detroit : 195

Display

Obrázok č. 49: Prehľad celkových prihlásení (Zdroj: vlastné spracovanie)

3.4 Routery

Ďalšia komponenta v informačnom systéme je určená pre prácu s routermi. Na túto komponentu sa užívateľ dostane buď cez menu, kde klikne na „Routers“ alebo pomocou úvodnej stránky, kde je odkaz na túto stránku nastavený na tlačidlo „Display“ pod Bootstrap kartou s názvom „Inactive routers“.

Pre prácu s routermi, je potrebné k ním pristupovať ako ku objektu, teda vytvoriť PHP triedu, ktorá bude obsahovať tri vlastnosti, a to jeho identifikátor, meno zákazníka a indikátor, či je aktívny. Vytvorenie objektu je znázornené na obrázku nižšie:

```
class routers {
    public $id;
    public $customer;
    public $active;
}
```

Obrázok č. 50: Objekt router (Zdroj: vlastné spracovanie)

V prípade vytvorenie nového routeru, je potrebné priradiť mu hodnoty do identifikátora, zákazníka a indikátor, či je aktívny. V prostredí PHP je jednoduchá syntax na vytvorenie novej inštancie. Príklad vytvorenia novej inštancie je zobrazený na obrázku nižšie:

```
$Ankara = new routers;
$Ankara->id = "IDENTIFIKATOR";
$Ankara->customer = "MENO_ZAKAZNIKA";
$Ankara->active = 1;
```

Obrázok č. 51: Vytvorenie novej inštancie (Zdroj: vlastné spracovanie)

Z obrázku vyššie, je vidieť príklad vytvorenia inštancie s názvom Ankara a priradenie hodnôt do jej vlastností, teda premenných, ktoré sú definované v triede.

3.4.1 Úvodná stránka komponenty routerov

Na tejto stránke (routery), sa užívateľovi budú zobrazovať opäť neaktívne routery a zoznam práve všetkých routerov a informácie k nim, teda u akého sú zákazníka a či sú aktívne. Na zobrazenie tohto zoznamu využijem opäť Bootstrap a opäť sa bude jednať o kartu. Do tohto elementu vypíšem všetky inštancie z triedy routers a ich vlastnosti. Tento prvok Bootstrapu vyzerá ako klasická tabuľka, no nejedná sa o tabuľku. V záhlaví karty sa zobrazí jej názov. Následne prvý prvok tela karty je celkový počet výskytu routerov, ktoré sú priradené ku zákazníkovi a za tým nasleduje zoznam routerov, kde je uvedený ich identifikátor, meno zákazníka a či je daný router aktívny. Pod zoznamom sa nachádzajú tri tlačidlá, pomocou ktorých je možné router editovať, pridať nový alebo deaktivovať. Nad jednotlivými inštanciami objektu je možné vykonávať CRUD operácie, čo je skratka z anglických slov create, read, update a delete. Celkový zoznam routerov a priradených podnikov je zobrazený na obrázku nižšie:

Routers - Customers

Total Count: 9

GW_ID: 2244 **Customer:** Astana **Active:** 1

GW_ID: 1156 **Customer:** Ankara **Active:** 1

GW_ID: 2536 **Customer:** Milan **Active:** 1

GW_ID: 3347 **Customer:** Brno **Active:** 1

GW_ID: 5536 **Customer:** Bagdad **Active:** 1

GW_ID: 5986 **Customer:** Paris **Active:** 1

GW_ID: 6641 **Customer:** Hanoj **Active:** 1

GW_ID: 7415 **Customer:** Tokyo **Active:** 1

GW_ID: 7485 **Customer:** London **Active:** 1

[Edit](#) [New](#) [Delete](#)

Obrázok č. 52: Zoznam všetkých routerov u zákazníka (Zdroj: vlastné spracovanie)

3.4.2 Vytvorenie nového routeru

Pre vytvorenie nového routeru do systému, je potreba skontrolovať, či má užívateľ informačného systému prístupové práva na tento úkon. Pokiaľ ich nebude mať, nepovolí sa mu prístup. Táto kontrola prebieha na základe typu účtu užívateľa. Pokiaľ má typ účtu hlavný administrátor, môže prejsť na danú stránku, pokiaľ má však účet bežného administrátora, presmeruje ho to na univerzálnu stránku, ktorá ho informuje o tom, že na danú operáciu nie je možné vykonať. Po kliknutí na tlačidlo „New“, sa otvorí jednoduchý formulár pre zadanie nového routeru do systému. Formulár je zobrazený na obrázku nižšie a obsahuje tri prvky:

- GW ID – textový typ,
- Customer – select (výber z možností),
- Active – checkbox (zaškrávané pole).

GW ID

Customer

Active

Obrázok č. 53: Vytvorenie nového routeru do systému (Zdroj: vlastné spracovanie)

Do prvku formulára, kde je výber zákazníka, je potreba samozrejme načítať všetkých zákazníkov. Toto načítanie je jednoduché, ide o SQL dotaz, ktorý vráti názvy všetkých zákazníkov.

```
SELECT Name FROM cust_customer;
```

Obrázok č. 54: Načítanie názvu podnikov (Zdroj: vlastné spracovanie)

Prvok formulára, ako aj celý formulár je napísaný v HTML za využitia frameworku Bootstrap. Kód tohto prvku je zobrazený nižšie:

```
<div class="form-group">
  <label for="inputState">Customer</label>
  <select name="Customer" id="Customer" class="form-control">
    <option selected>Choose...</option>
    <option></option>
  </select>
</div>
```

Obrázok č. 55: Prvok formulára – Customer (Zdroj: vlastné spracovanie)

Zoznam všetkých názvov podnikov je potrebné do tohto kódu dostať. Využijem opäť jazyk PHP, kde mi select vráti názvy a počet vrátených hodnôt. Sú dve možnosti na zobrazenie týchto názvov. Jedným je využiť cyklus for, kde je potrebné vedieť počet podnikov alebo druhou možnosťou je cyklus foreach. Do tohto cyklu sa vloží html tag <option></option>, do ktorého sa vždy vloží jeden názov podniku.

Následne som s formulárom už nerobil žiadne úpravy, po jeho vyplnení a odoslani (po kliknutí na „Save“) sa odošle na server. Na serveru je potrebné vyhodnotiť, či je router v systéme vedený ako aktívny alebo nie. Toto vyhodnotenie je pomocou podmienky, kde zistím, či tento checkbox bol zaškrtnutý alebo nie. Toto vyhodnotenie je zobrazené nižšie:

```
if (isset($_POST["Active"])) {  
    $Active = 1;  
}  
else {  
    $Active = 0;  
}
```

Obrázok č. 56: Vyhodnotenie aktivity routeru (Zdroj: vlastné spracovanie)

Toto vyhodnotenie je jednoduché, ide vlastne o podmienku, kde sa vyhodnotí, či tento checkbox bol zaškrtnutý, Ak bol, nastaví hodnotu premennej Active na 1, v prípade že tento formulárový prvok nebol zaškrtnutý, nastaví hodnotu na 0;

Následne sa uskutoční pripojenie do databáze zavolaním funkcie DatabaseConn(), ktorá je popísaná vyššie a vloží sa nový záznam do databáze, konkrétne do tabuľky cp_stats.

Odstránenie alebo deaktivovanie routeru, môže opäť uskutočniť iba hlavný administrátor, takže aj tu platí rovnaká kontrola, teda overí sa na strane serveru, aký typ účtu má prihlásený užívateľ.

3.4.3 Deaktivácia routeru

Odstránenie routeru prebieha opäť pomocou formulára. Formulár je zobrazený nižšie:

Choose Router:

Choose... ▼

Delete

Obrázok č. 57: Deaktivácia routeru (Zdroj: vlastné spracovanie)

Formulár obsahuje dva prvky, jeden je určený pre výber routeru na základe jeho identifikátora a druhý prvok je zaškrťavacie tlačidlo, ktoré informuje o tom, že router

nemá byť deaktivovaný, ale aj odstránený zo systému. Pokiaľ toto tlačidlo nie je zaškrtnuté, router sa iba deaktivuje.

Po odoslaní formuláru opäť prebieha kontrola na strane serveru, kde zistí, či má byť router aj odstránená alebo len deaktivovaný. Toto rozhodnutie sa vyhodnocuje na základe podmienky, či užívateľ zaškrtnol možnosť odstránenia routeru zo systému. Pokiaľ tento checkbox nebol zaškrtnutý, prebehne len update do databáze na základe identifikátora routeru, pokiaľ zaškrtnutý bol, prebehne delete v databáze.

3.4.4 Editácia routeru

Editáciu routerov môže uskutočniť aj bežný administrátor, takže pri tejto operácii neprebíha žiadne overenie prístupu. Po kliknutí na „Edit“ sa užívateľovi otvorí checkbox, kde zvolí router, ktorý má byť editovaný, na základe jeho identifikátora.

Choose Router:

Choose... ▼

Edit

Obrázok č. 58: Výber routeru na editáciu (Zdroj: vlastné spracovanie)

Užívateľ vyberie zo zoznamu router, ktorý chce editovať. Na základe vybraného routeru prebehne do databáze dotaz (select), ktorý vráti všetky potrebné dáta k tomuto routeru a vyplní ich do formulára. Toto naplnenie som uskutočnil tak, že do konkrétnych HTML inputov som pridal atribút value, do ktorého som naplnil hodnoty, ktoré mi vrátila databáza. Ak si užívateľ vyberie router s ID napríklad 2244, tak dotaz bude vyzerať nasledovne:

```
SELECT s.GW_ID, c.Name, s.Active FROM cp_stats s, cust_customer c WHERE  
s.ID_Customer = c.ID_Customer AND GW_ID = 2244;
```

Obrázok č. 59: SQL dotaz na získanie dát o routeru (Zdroj: vlastné spracovanie)

Samozrejme, že v PHP kóde sa do podmienky v tomto dotaze generuje GW_ID na základe toho, čo užívateľ zadal, teda pomocou metódy POST po odoslaní formulára si túto hodnotu odchyším. Nasledujúci obrázok zobrazuje naplnenie hodnoty do formulára:

```
<div class="form-group" >
  <label for="Active">Active:</label>
  <input type="Active" class="form-control" name="Active" id="Active" value="<?php print $Active ?>">
</div>
```

Obrázok č. 60: Naplnenie editačného formulára (Zdroj: vlastné spracovanie)

Z obrázku je vidieť, že do atribútu value sa vkladá hodnota, ktorá je uložená v premennej s názvom active, do ktorej sa priradila hodnota z databáze. Takto som postupoval u všetkých prvkov formulára.

Následne opäť využijem dostupný formulár z Bootstrapu:



The image shows a Bootstrap form with three input fields and an 'Update' button. The first field is labeled 'GW ID:' and contains the value '2244'. The second field is labeled 'Customer:' and contains the value 'Astana'. The third field is labeled 'Active:' and contains the value '1'. Below the fields is a blue 'Update' button.

Obrázok č. 61: Vyplnený editačný formulár routeru (Zdroj: vlastné spracovanie)

Administrátor po ukončení editácie (ktorá je primárne určená na zmenu podniku, v ktorom sa konkrétny router nachádza), klikne na tlačidlo „Update“, ktorý pošle hodnoty pomocou POST-u na server, kde ich odchytiť rovnakým spôsobom ako u predošlom posielaní pomocou tejto metódy. Následne na uloženie do databáze je potreba napísať SQL dotaz vo forme update, ktorý aktualizuje hodnoty v databáze.

3.5 Zákazníci

V tejto kapitole sa budem venovať práci so zákazníkmi, teda podnikmi, v ktorých sú routery spoločnosti, na ktorých beží reklama inzerentov. Na základe analýzy, je potreba aby správcovia mohli založiť nový účet zákazníkom, mať možnosť náhľadu nad tento účet a dáta s tým spojené, editovať údaje o zákazníkovi, priradiť ku zákazníkovi router (opačná operácia ako u komponenty routerov) a v poslednom rade mať možnosť odstrániť účet zo systému.

Tak, ako aj u routerov, tak aj u zákazníkov je potrebné k nim v informačnom systéme pristupovať ako ku objektu. Ako každý objekt, aj objekt zákazník má nejaké vlastnosti, čo je vlastne premenná, ktorá je definovaná v triede. Vytvorenie triedy a definícia vlastností je zobrazená na obrázku nižšie:

```
class Customers {  
    public $id;  
    public $name;  
    public $login;  
    public $password;  
    public $coin;  
    public $type;  
    public $allowed;  
}
```

Obrázok č. 62: Trieda zákazníkov (Zdroj: vlastné spracovanie)

3.5.1 Úvodná stránka komponenty podnikov

Na úvodnej stránke tejto komponenty budem zobrazovať zoznam podnikov, ktoré sú vedené v informačnom systéme a všetky potrebné informácie o tomto objekte. Pre zobrazenie všetkých zákazníkov využijem jednoduchú HTML tabuľku od Bootstrapu. V hlavičke sa bude zobrazovať identifikátor podniku, názov, prihlasovacie meno, počet kreditu, typ podniku, a či má daný podnik povolenú reklamu od inzerenta alebo nie. Telo tabuľky obsahuje jednotlivé objekty, teda podniky.

Pod tabuľkou sa opäť nachádzajú tlačidlá, pomocou ktorých sa dá vytvoriť nový objekt, editácia stávajúceho, zablokovanie účtu alebo nastavenie nového hesla. Na obrázku nižšie je zobrazenie podnikov v tabuľke a jednotlivé tlačidlá, s ktorými je možné uskutočniť CRUD (vytvoriť, čítať, aktualizovať, vymazať) operácie nad konkrétnym objektom:

ID	Name	Login	Coin	Type	Allowed
12258712	Bagdad	Bagdad	0	Restaurant	1
13258965	Baku	Baku	0	Restaurant	1
15965852	Ankara	Ankara	0	Pub	1
32545965	Astana	Astana	0	Caffe	1
36985412	Paris	Paris	0	Restaurant	1
39872365	Brno	Brno	0	Shop	1
45878520	Milan	Milan	0	Gym	1
47858965	Tokyo	Tokyo	0	Bar	1
65234852	Peking	Peking	0	Caffe	1
65896585	Soul	Soul	50.5	Bar	0

[Edit](#)
[New](#)
[Set Password](#)
[Deactivate](#)
[Coin](#)

Obrázok č. 63: Zoznam všetkých podnikov v IS (Zdroj: vlastné spracovanie)

3.5.2 Vytvorenie nového podniku

Založenie nového podniku bude opäť prebiehať jednoduchým formulárom, ktorý vyplní správca informačného systému. Oprávnenie na tento úkon majú oba typy administrátorských účtov, takže nie je potrebné uskutočniť kontrolu účtu. Formulár sa skladá z deviatich prvkov:

- identifikátor podniku,
- názov podniku,
- prihlasovacie meno,
- heslo,
- zopakovať heslo (pre overenie správnosti),
- typ podniku (výber zo zoznamu),
- mesto (výber zo zoznamu, možnosť filtrácie),
- adresa,
- povolenie pre reklamu (výber áno/nie).

Formulár sa zobrazí po kliknutí na tlačidlo „New“ a vyzerá nasledovne:

ID_Customer:

Name:

Login:

Password:

Repeat password:

Choose type:

Choose city:

Address:

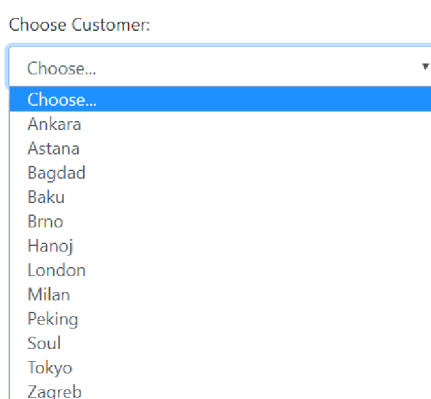
Allowed:

Obrázok č. 64: Formulár pre pridanie nového podniku (Zdroj: vlastné spracovanie)

Po odoslaní na server prebieha niekoľko kontrol. Samozrejme, že prvou kontrolou je zistenie, či sa nejedná o duplicitu. Táto kontrola sa uskutoční pomocou porovnania identifikátora prijatého na server a identifikátora aktuálnych objektov.. Ak je táto kontrola v poriadku, neznamená to, že sa nejedná o duplicitu podniku, preto sa porovnáva ešte názov a adresa podniku prijatého na server s ďalšími objektami. Ak táto kontrola prebehne v poriadku, prejdem ku poslednej kontrole a to, porovnanie zadaných hesiel. Táto kontrola samozrejme prebieha aj na klientovi, ale je potrebné ju z bezpečnostných dôvodov uskutočniť aj na serveri. Až keď tieto kontroly prebehnú v poriadku, môžem jednoduchým SQL insert-om vložiť záznam do databáze.

3.5.3 Editácia podnikov

Editáciu na úrovni podnikov, môže uskutočniť každý užívateľ informačného systému, takže nie je potreba uskutočniť overenie užívateľského účtu. Ako v predchádzajúcom prípade, užívateľ klikne na tlačidlo „Edit“, vyberie zákazníka, respektíve podnik, ktorý chce editovať a pristúpi ku samotnej editácii. Výber podniku som uskutočnil pomocou klasického výberu, kde sa načítajú všetky názvy objektu customer. Zoradia sa abecedne a je možné pomocou kliknutia na konkrétny názov alebo prípadne pomocou písania na klávesnici vybrať konkrétny podnik. Tento výber je zobrazený na obrázku nižšie:



Obrázok č. 65: Výber podniku ku editácií (Zdroj: vlastné spracovanie)

Na editačnom formulári sa zobrazia nasledujúce prvky:

- identifikátor podniku,
- názov,
- prihlasovacie meno,
- typ účtu,
- mesto,
- adresa,
- povolenie pre reklamu.

Všetky tieto údaje sú pred vyplnené vo formulári na základe toho podniku, ktorý som vybral v predchádzajúcom kroku. Editačný formulár je zobrazený nižšie:

ID_Customer:

Name:

Login:

Choose type:

Choose city:

Address:

Allowed:

Obrázok č. 66: Editačný formulár podniku (Zdroj: vlastné spracovanie)

Administrátor môže zmeniť ľubovoľný počet formulárových prvkov a odoslať údaje na server. Samozrejme, ako pri vytváraní nového, je potreba skontrolovať, či už neexistuje objekt s daným identifikátorom, názvom a adresou. Pokiaľ kontrola prebehne v poriadku, prebehne jednoduchý SQL update do databáze. Keďže, ale administrátor môže upraviť aj identifikátor, tak je potrebné si uchovať pôvodný identifikátor, aby som mohol v podmienke SQL update jednoznačne určiť, ktorý záznam sa má aktualizovať.

3.5.4 Zmena hesla účtu podnikov

Z analýzy, ktorú som uskutočnil v predchádzajúcej kapitole, tak tento proces nebol spracovaný, no rozhodol som sa ho tam zakomponovať, keďže nie je potrebné pri zmene hesla, aby administrátor sa za každým pripojil na databázu a manuálne zmenil heslo. Za prvé je to práca na vyše a z bezpečnostných dôvodov to neodporúčam. Táto zmena hesla je jednoduchá, užívateľ na formulári vyberie účet, zadá dva krát heslo a odošle formulár na server. Formulár je zobrazený na nasledujúcom obrázku:

Choose customer:

Password:

Repeat password:

Obrázok č. 67: Formulár na zmenu hesla zákazníkovi (Zdroj: vlastné spracovanie)

Z hľadiska bezpečnosti, opäť využijem šifrovací algoritmus SHA 256. Tento úkon je potrebné spraviť pre oba formulárové prvky (heslo, zopakovať heslo). Následne zašifrované heslá porovnam, či sa zhodujú, pokiaľ áno, aktualizujem heslo v databáze u konkrétneho zákazníka.

```
// Hash prvého prvku
$pw1_salt = $_POST['PWD'].$salt;
$user_pwd1 = strtoupper(hash('sha256', $pw1_salt));

// Hash druhého prvku
$pw2_salt = $_POST['RPWD'].$salt;
$user_pwd2 = strtoupper(hash('sha256', $pw2_salt));
```

Obrázok č. 68: Šifrovanie hesla pri zmene (Zdroj: vlastné spracovanie)

Predchádzajúci obrázok zobrazuje zašifrovanie hesla spolu so salt (preddefinovaný textový reťazec). Následne prebehne podmienka na to, či sa tieto dve premenné zhodujú. Ak sa zhodujú, pripojím sa prostredníctvom PHP na databázu a pomocou SQL update heslo aktualizujem, ak sa nezhodujú, administrátorovi sa zobrazí chyba.

3.5.5 Úprava kreditu podniku

Ako som spomínal v analýze, podniky môžu využiť možnosť propagácie svojho podniku. Táto propagácia samozrejme niečo stojí a je potreba v istých prípadoch, kedy neprejde pripísanie kreditu na účet podniku, pripísať kredit na účet. Opäť ide o jednoduchý formulár, kedy administrátor vyberie účet podniku a nastaví sumu, ktorá sa má pripísať.

Výsledný formulár vyzerá nasledovne:

Choose customer:

Set amount:

Give

Obrázok č. 69: Formulár pre dobitie kreditu podniku (Zdroj: vlastné spracovanie)

Administrátor môže vybrať ako kladnú, tak aj zápornú čiastku. Preto túto čiastku napíše do textového políčka. Pri odobratí kreditu jednoducho zadá čiastku s mínusovým znamienkom. Formulár užívateľ odošle na server, kde prebehne SQL insert do tabuľky cust_coin. Aby som nemusel riešiť na serveri ešte aktualizovanie kreditu v tabuľke inzerentov, napíšem jednoduchý trigger na tabuľku cust_coin, ktorý spustí update po vložení hodnôt do tejto tabuľky.

```
CREATE TRIGGER Customer_coin
AFTER INSERT
ON cust_coin FOR EACH ROW
BEGIN
UPDATE cust_customer SET Coin = Coin + new.Amount WHERE ID_Customer = new.ID_Customer;
END;
```

Obrázok č. 70: Trigger po dobití kreditu (Zdroj: vlastné spracovanie)

3.5.6 Deaktivácia účtu podniku

Posledná možnosť nad komponentov zákazníkov (podnikov) je deaktivácia účtu. Oprávnenie na deaktiváciu má iba hlavný administrátor, preto využijem kontrolu, ktorú som použil pri komponente routerov, teda porovnam u aktuálneho administrátora jeho typ účtu. Pokiaľ sa jedná o hlavného administrátora, vyberie opäť zo zoznamu podnik a účet sa deaktivuje. V opačnom prípade ho presmeruje na preddefinovanú stránku, ktorá užívateľa informuje o tom, že nemôže danú operáciu uskutočniť, pretože nemá oprávnenie. Formulár pre deaktiváciu účtu je zobrazený nižšie:

Choose customer:

Deactivate

Obrázok č. 71: Výber podniku na deaktiváciu účtu (Zdroj: vlastné spracovanie)

Po kliknutí na tlačidlo „Deactivate“, sa odošle na server identifikátor podniku, ktorému sa má účet deaktivovať. Informačný systém podnikov využíva pre prihlasovanie do svojej aplikácie tabuľku `cust_customer`. Prihlásenie do tohto oddeleného IS funguje na princípe (ako aj v tomto prípade), že ak pri prihlásení nie je vyplnené užívateľské meno alebo heslo, tak sa prihlásenie neuskutoční. Takže nemá zmysel, aby sa účet zmažal fyzicky z databázy, ale len pomocou SQL update sa vymažú údaje u atribútov `login` a `password`. Výhodou to má aj v tom prípade, ak by sa podnik opäť zavádzal do systému, nie je potreba mu zakladať znova účet, ale len nastaviť meno a heslo a všetky relácie, ktoré bola v databáze na tento účet budú archivované a zachované.

3.6 Inzerenti

Táto komponenta je určená inzerentom, teda zákazníkmi, ktorí inzerujú reklamu prostredníctvom portálu v podnikoch. Samozrejme, aj u inzerentoch budem ku nim pristupovať ako ku objektu, teda je potreba na začiatku vytvoriť triedu a vlastnosti tejto triedy.

```
class Advertisers {
    public $id_advertiser;
    public $name;
    public $login;
    public $coin_number;
    public $day_limit;
    public $type;
    public $city;
    public $address;
}
```

Obrázok č. 72: Vytvorenie triedy inzerentov (Zdroj: vlastné spracovanie)

3.6.1 Úvodná stránka komponenty

Takisto ako u podnikov, aj komponenta inzerentov má úvodnú stránku, kde budem zobrazovať zoznam všetkých inzerentov vedených v informačnom systéme. Opäť využijem na zobrazenie tabuľku od Bootstrapu, ktorú naplním jednotlivými inštanciami objektov inzerentov. V záhlaví tabuľky sú názvy stĺpcov (identifikátor, názov inzerenta, prihlasovacie meno, počet kreditu, denný limit).

ID	Name	Login	Coin	Limit
14524965	Antonio	Antonio	56	10
21459820	Tuscon	Tuscon	42	5
23368522	Omaha	Omaha	98	10
23552398	Riverside	Riverside	169	20
36541258	Seattle	Seattle	12	10
48522452	Witchita	Witchita	131	20

Edit New Set Password Deactivate Coin

Obrázok č. 73: Úvodná stránka inzerentov (Zdroj: vlastné spracovanie)

Nad jednotlivými entitami bude administrátor môcť uskutočniť:

- editáciu,
- vytvoriť nový účet,
- nastaviť heslo,
- deaktivovať,
- dobiť kredit,
- nastaviť denný limit.

Nastavenie hesla, deaktivácia a editácia bude prebiehať rovnako, ako u zákazníkov, takže využijem už napísaný kód, kedy len upravím samotný update do databáze.

3.6.2 Vytvorenie nového účtu inzerentom

Vytvorenie účtu inzerentom má oprávnenie ako hlavný administrátor, tak bežný administrátor, nie je teda potrebná kontrola užívateľských práv. Vytvorenie účtu bude prebiehať podobne, ako u inzerenta, prostredníctvom formulára, ktorý administrátor vyplní. Položky formulára sa však budú o niečo líšiť. Formulár sa užívateľovi zobrazí po kliknutí na tlačidlo „New“ a vyzerá nasledovne:

ID Advertiser:

Name:

Login:

Password:

Repeat password:

Day limit:

Choose type:

Choose city:

Address:

Obrázok č. 74: Formulár pre vytvorenie účtu inzerentovi (Zdroj: vlastné spracovanie)

Po tom, ako užívateľ vyplní formulár, pomocou tlačidla „Save“ odošle údaje na server, kde ich spracujem. Porovnam, či položky hesiel sa zhodujú a takisto, či už neexistuje

inzerent s rovnakým identifikátorom. Pokiaľ kontrola prebehne v poriadku (v oboch prípadoch podmienka vráti true), údaje sa vložia do databáze pomocou SQL insertu.

3.6.3 Dobitie kreditu a nastavenie limitu

Tak, ako v prípade zákazníkov, môže v istých situáciách nastať, že administrátor musí manuálne dobiť kredit inzerentovi, prípadne mu nastaviť denný limit, ktorý je ochotný minúť na inzerciu. Dobitie kreditu prebehne podobne, ako už zákazníka, teda prebehne insert do tabuľky adv_coin. Opäť napíšem trigger, ktorý po vložení dát do tejto tabuľky aktualizuje kredit aj v tabuľke adv_advertiser.



Choose advertiser:

Choose...

Set amount:

Give

Obrázok č. 75: Formulár pre dobitie kreditu inzerentom (Zdroj: vlastné spracovanie)

Trigger bude potom vyzerat' nasledovne:

```
CREATE TRIGGER Advertiser_coin
AFTER INSERT
ON adv_coin FOR EACH ROW
BEGIN
    UPDATE adv_advertiser SET Coin_Number = Coin_Number + new.Amount WHERE ID_Advertiser = new.ID_Advertiser;
END;
```

Obrázok č. 76: Trigger pre dobitie kreditu inzerenta (Zdroj: vlastné spracovanie)

Pre nastavenie limitu, pridám na úvodnú stránku komponenty ďalšie tlačidlo. Po kliknutí naň, sa užívateľovi otvorí formulár, kde vyberie inzerenta a nastaví mu denný limit, ktorý je ochotný minúť. Do formulárového prvku, ktorý je určený na vloženie limitu, sa zobrazí užívateľovi aktuálne nastavený denný limit inzerenta. Na strane serveru potom musí prebehnúť kontrola, či užívateľ náhodou nezadal záporné číslo, keďže záporný limit nedáva žiadnu logiku. Pokiaľ opäť prejde kontrola v poriadku, prebehne aktualizácia dát do databáze.

```
function DayLimit ($limit) {
    if ($limit > 0) {
        return true;
    }
    else {
        return false;
    }
}
```

Obrázok č. 77: Funkcia na kontrolu zadaného limitu (Zdroj: Vlastné spracovanie)

Ide o jednoduchú funkciu, ktorá má jednu vstupnú hodnotu a to limit. Následne prebehne kontrola, či táto vstupná hodnota je väčšia ako hodnota 0. Pokiaľ áno, funkcia vráti true, pokiaľ je hodnota 0 alebo záporná, funkcia vráti false.

3.7 Reklamné kampane

Prostredníctvom tejto komponenty bude môcť administrátor zakladať prípadne editovať súčasne vytvorené reklamné kampane inzerentov, prípadne samotných podnikov. Vytvorenie bude opäť prebiehať prostredníctvom formulára, kde administrátor vyplní povinné položky a odošle dáta na server. Ku reklamnej kampani opäť budem pristupovať objektovo, nasledujúci obrázok zachytáva triedu reklamnej kampane.

```
class advert {
    public $id;
    public $id_advertiser;
    public $id_customer;
    public $id_router;
    public $background;
    public $logo;
    public $text;
    public $advert_cost;
    public $advert_limit;
    public $redirect;
    public $date;
}
```

Obrázok č. 78: Trieda reklamnej kampane (Zdroj: vlastné spracovanie)

3.7.1 Úvodná stránka komponenty

Úvodná stránka komponenty nebude vyzerat' úplne rovnako, ako v predchádzajúcich prípadoch. Nie je potrebné, aby mal administrátor náhľad nad všetkými reklamnými kampaňami. Na úvodnej stránke si administrátor vyfiltruje reklamné kampane, ktoré bude chcieť, aby sa mu zobrazili. Filtrácia bude možná buď podľa inzerenta, zákazníka prípadne konkrétneho routeru. Užívateľ si teda vyberie konkrétneho inzerenta alebo podnik a zobrazia sa mu len reklamné kampane, ktoré budú zodpovedať filtru. Následné zobrazenie výsledkov, ktoré zodpovedali filtrácii sa zobrazia pomocou Bootstrap-ovej tabuľky. Nad touto entitou bude môcť administrátor uskutočniť:

- editáciu reklamnej kampane,
- deaktiváciu reklamnej kampane.

Čo sa týka editácie reklamnej kampane, administrátor bude môcť zmeniť celkovú sumu, denný limit, redirect, odkaz na pozadie, odkaz na logo a text. Ak užívateľ zvolí deaktiváciu, prepíše sa v databáze dátum u reklamnej kampane.

Choose router:

Obrázok č. 79: Príklad filtrácie podľa routeru (Zdroj: vlastné spracovanie)

Po vyfiltrovaní, sa zobrazia užívateľovi konkrétne výsledky:

ID	Advertiser	Router	Cost	Limit	Redirect	Customer
1	Detroit	7485	100	10	https://google.com	Bagdad
7	Detroit	5536	250	15	https://google.com	Paris
14	Detroit	3347	150	10	https://google.com	Ankara
24	Detroit	2536	100	20	https://google.com	Tokyo
33	Detroit	5986	300	10	https://google.com	Hanoj
37	Detroit	2244	130	15	https://google.com	London

Obrázok č. 80: Zobrazenie výsledkov filtrácie (Zdroj: vlastné spracovanie)

3.7.2 Vytvorenie reklamnej kampane

Práva na vytvorenie reklamnej kampane majú oba typy správcovských účtov, takže nie je potrebná kontrola práv na strane serveru. Vytvorenie novej kampane má niekoľko krokov, ktoré je možné uskutočniť pomocou formulára. Formulár vyzerá nasledovne:

Choose advertiser:

Choose router:

Set background:

Set logo:

Set text:

Set advert cost:

Set advert limit:

Set redirect:

Set datetime:

Obrázok č. 81: Vytvorenie reklamnej kampane (Zdroj: vlastné spracovanie)

3.7.3 Editácia reklamnej kampane

Na editáciu konkrétnej kampane sa užívateľ dostane pomocou tlačidla „Edit“ na úvodnej stránke komponenty. Obrazovka editácie sa skladá z dvoch častí, ide teda o dvojstĺpcový layout, vytvorený pomocou Bootstrap frameworku. Na pravej časti sa nachádza editačný formulár, s vyplnenými hodnotami v jednotlivých formulárových prvkoch:

Set background:

Set logo:

Set text:

Set advert cost:

Set advert limit:

Set redirect:

Obrázok č. 82: Editáčný formulár reklamnej kampane (Zdroj: vlastné spracovanie)

Keďže z analýzy vyplynulo, aby správca systému mal náhľad aj nad štatistikami prihlásených užívateľov, rozhodol som sa to zakomponovať do konkrétnej reklamnej kampane. Na ľavej časti obrazovky má teda administrátor systému detailnejší náhľad na konkrétnu reklamnú kampaň, čo obsahuje teda aj informácie o prihlásených užívateľoch.

ID	Advertiser	Router	Cost	Limit	Redirect	Customer
1	Detroit	7485	100	10	https://google.com	Bagdad

January	February	March	April	May	June	July	August	September	October	November	December
93	67	72	34	45	30	7	0	0	0	0	0

Currently logged in: 2

Obrázok č. 83: Detail reklamnej kampane (Zdroj: vlastné spracovanie)

3.7.4 Zablokovanie reklamnej kampane

Správca by mal mať možnosť zablokovať reklamnú kampaň inzerentovi, či už na základe jeho vlastnej žiadosti, prípadne za porušenie podmienok. Algoritmus výberu reklamnej kampane zistí, či je reklamná kampaň aktívna. To vykoná tak, že sa pozrie do databáze na tabuľku cp_advert a konkrétne na atribút date. Pokiaľ je dátum a čas menší alebo rovný skutočnému, tak túto kampaň odignoruje. Preto je pre administrátora postačujúce, aby sa na konkrétnej kampani, ktorú chce zablokovať, zmenil dátum a čas. Správca klikne na úvodnej stránke komponenty na tlačidlo „Deactivate“, vyberie konkrétnu reklamnú kampaň a tá sa deaktivuje. Na strane serveru opäť prebieha kontrola, či sa jedná o typ účtu, ktorý prináleží hlavnému administrátorovi. Pokiaľ išlo o deaktiváciu na základe žiadosti od inzerenta, administrátor zistí počet kreditu a vráti ho na účet inzerentovi.

Choose advert:

Choose...

Obrázok č. 84: Výber kampane na deaktiváciu (Zdroj: vlastné spracovanie)

3.8 Správa užívateľských účtov

Hlavný administrátor informačného systému má taktiež možnosť založiť nový účet administrátorovi. Na úvodnej stránke tejto komponenty sa užívateľovi zobrazia všetci aktuálni správcovia systému. Ide opäť o tabuľku, ktorá je zobrazená nižšie:

ID	Name	Surname	Mail	Phone	Login	Type
1	Admin	Admin	admin@wifimedia.com	none	admin	standard
2	Head	Head	headadmin@wifimedia.com	none	headadmin	main

[New](#) [Edit](#) [Deactivate](#)

Obrázok č. 85: Zobrazenie všetkých užívateľských účtov administrátorov

(Zdroj: vlastné spracovanie)

3.8.1 Založenie nového účtu

Založenie nového účtu prebieha štandardne, prostredníctvom formulára. Je potrebné, aby pri založení nového účtu, deaktivácia alebo prípadne editácie účtu prebehla kontrola na strane serveru, či má užívateľ práva na tieto operácie, keďže na tieto operácie je potrebný užívateľský účet s typom hlavný administrátor.

Užívateľ klikne na tlačidlo nový, prebehne kontrola na strane serveru a pokiaľ je mu táto operácia umožnená, zobrazí sa mu nasledujúci formulár. Po uložení prebehne kontrola duplicity na základe užívateľského mena. Pokiaľ je kontrola v poriadku, záznam sa uloží do tabuľky a_admin. Ak sa našla duplicita, záznam sa do databáze neuloží.

Name:

Surname:

Mail:

Phone:

Login:

Password:

Repeat password:

Choose type:

Obrázok č. 86: Založenie nového účtu administrátorovi (Zdroj: vlastné spracovanie)

3.8.2 Deaktivácia správcovského účtu

Ako som vyššie spomínal, aj na túto operáciu je potreba mať práva. Samotná deaktivácia bude prebiehať tak, že si užívateľ vyberie konkrétny účet a klikne na tlačidlo. Spoločnosť vyžaduje, aby sa účet, ktorý bol deaktivovaný bolo možné späťne aktivovať, preto je potrebné vytvoriť kópiu tabuľky, kde sa presunú neaktívne účty.

```
drop table if exists D_Admin;  
create table D_Admin (  
  ID_Admin INT AUTO_INCREMENT NOT NULL,  
  Name VARCHAR(45),  
  Surname VARCHAR(45),  
  Mail VARCHAR(45),  
  Phone VARCHAR(45),  
  Login VARCHAR(45) NOT NULL,  
  ID_Type INT NOT NULL,  
  PRIMARY KEY(ID_Admin)  
);
```

Obrázok č. 87: Tabuľka pre deaktivované účty (Zdroj: vlastné spracovanie)

Následne je potreba vytvoriť trigger do databáze, ktorý vymazané dáta z tabuľky a_admin presunie do tabuľky d_admin.

```
CREATE TRIGGER Deactivate_adminn
AFTER DELETE
ON a_admin FOR EACH ROW
BEGIN

INSERT INTO d_admin (ID_Admin, Name, Surname, Mail, Phone, Login, ID_Type)
VALUES
(OLD.ID_Admin, OLD.Name, OLD.Surname, OLD.Mail, OLD.Phone, OLD.Login, OLD.ID_Type);

END;
```

Obrázok č. 88: Trigger pre deaktiváciu administrátorského účtu (Zdroj: vlastné spracovanie)

3.8.3 Editácia administrátorského účtu

Editácia funguje na rovnakom princípe ktorejkoľvek editácie v informačnom systéme. Hlavný administrátor si vyberie konkrétny účet správcu systému a pomocou formulára zmení nastavenie účtu:

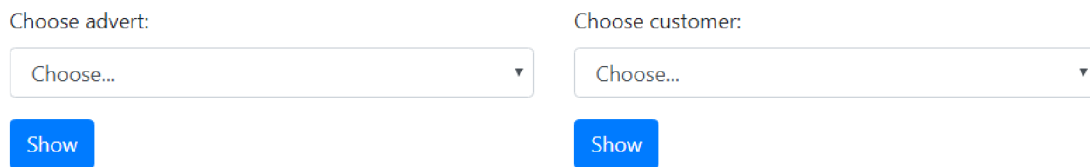
- meno správcu,
- priezvisko správcu,
- emailovú adresu,
- telefónne číslo,
- prihlasovacie meno,
- heslo,
- typ účtu.

3.9 Náhľad na štatistiky prihlásení

Správca informačného systému má náhľad na prihlásených užívateľov, či už v rámci konkrétnej reklamnej kampane alebo konkrétneho podniku. Tieto štatistiky sa zobrazujú pre daný mesiac v konkrétnom roku. Ide o tabuľku, ktorej hlavičku tvoria práve názvy mesiacov. Pre každý rok je vytvorená daná tabuľka a užívateľ pomocou tlačidiel si jednotlivé roky môže prepínať.

3.9.1 Úvodná stránka komponenty

Na úvodnej stránke komponenty si užívateľ vyberie, či chce zobrazit' štatistiky prihlásených užívateľov na konkrétnej reklamnej kampani alebo v rámci daného podniku. Ide o jednoduché filtrovanie dát, keďže tabuľka cp_user obsahuje cudzie kľúče do tabuľky kampaní a na konkrétny router, ktorý je previazaný zasa na konkrétny podnik. Užívateľ si vyberie formu filtrácie a zobrazia sa mu dáta, ktoré odpovedajú filtru. Príklad filtrácie je na obrázku nižšie:



Choose advert: Choose customer:

Obrázok č. 89: Výber náhľadu nad prihlásenými užívateľmi (Zdroj: vlastné spracovanie)

Podľa zvolenej filtrácie, prebehne select do databáze, prvá časť selectu je rovnaká, líši sa len podmienka za klauzulou where. Následne sa užívateľovi zobrazia dáta, ktoré odpovedajú selectu v tabuľke:

Year	January	February	March	April	May	June	July	August	September	October	November	December	Overall
2017	198	147	219	245	201	199	174	215	244	279	283	217	2621
2018	211	193	201	234	204	179	183	190	214	301	289	274	2673
2019	201	179	187										567

Overall: 5861

Currently logged in: 3

Obrázok č. 90: Štatistiky prihlásení (Zdroj: vlastné spracovanie)

3.10 Ekonomické zhodnotenie

Náklady, ktoré sú spojené s tvorbou tohto rozhrania vyplývajú z časovej náročnosti. Nasledujúca tabuľka zobrazuje rozpad úloh (činností), ich časovú náročnosť a náklady na ňu. Na jednu hodinu pracovníka pripadá 15 €.

Tabuľka č. 14: Ekonomické zhodnotenie (Zdroj: Vlastné spracovanie)

Činnosť	Časová náročnosť	Suma
Rozdelenie činností a zodpovednosti	10	150 €
Výber technológií a príprava prostredia	15	225 €
Návrh a zmena dátového modelu	10	150 €
Naplnenie databáze testovacími dátami	3	45 €
Programovanie back-end	80	1 200 €
Grafický návrh a programovanie front-end	70	1 050 €
Testovanie rozhrania	50	750 €
Implementácia	20	300 €
Školenie administrátorov	15	225 €
Celkové náklady	273	4 095 €

3.11 Prínosy práce

Nasledujúci zoznam zobrazuje prínosy mojej práce:

- **možnosť spravovať všetky informačné systémy spoločnosti,**
- **možnosť spravovať IS z jedného centrálného miesta,**
- **zjednotenie procesu spravovania pre administrátorov,**
- **zníženie rizika straty dát,**
- **jednoduchá správa užívateľských účtov.**

Medzi najväčšie prínosy diplomovej práce radím najmä možnosť spravovať všetky systémy z jedného centrálného miesta, čím sa zjednotil proces pre správu systémov. S tým spojené je aj zníženie rizika straty dát, keďže administrátori nebudú môcť zasahovať priamo do centrálnej databáze, iba s pomocou jednotného rozhrania. Ďalším dôležitým prínosom je správa užívateľských účtov podnikov a inzerentov.

ZÁVER

Vo svojej diplomovej práci som sa venoval analýze, návrhu a tvorbe informačného systému pre administrátorov. Prostredníctvom tohto systému je možné spravovať všetky informačné systémy spoločnosti z jedného centrálného miesta, čo ušetrí prácu administrátorom a zníži rizika, ktoré boli spojené s doterajším procesom spravovania.

V prvej časti práce som sa venoval teoretickým východiskám práce, z ktorých som následne čerpal či už v analytickej časti alebo v návrhovej časti. V teoretických východiskách som sa venoval spracovaniu problematiky informačných systémov, databázových systémov, dátovému modelovaniu a technológiám, ktoré sa v súčasnosti používajú pri tvorbe informačných a databázových systémov.

Pred samotným návrhom a tvorbou informačného systému som dôkladne zmapoval súčasný stav informačných systémov a databáze, taktiež procesov, ktoré by mali prebiehať v centrálnom rozhraní pre administrátorov. Na začiatku kapitoly analýzy súčasného stavu som sa venoval analýze spoločnosti a jej aktuálnych produktov. Pomocou E-R diagramu som zachytil súčasný stav databáze a následne som popísal jednotlivé tabuľky a ich atribúty. Následne som prešiel ku zmapovaniu procesov, ktoré by mal správcovní informačný systém zvládnuť. Tieto procesy som zachytil pomocou vývojových diagramov, čo slúžilo ako podklad pre samotné programovanie, teda návrhovú časť práce.

V návrhovej časti diplomovej práce som sa venoval samotnému návrhu a tvorbe informačného systému. V úvode kapitoly som popísal priebeh prípravy prostredia, čo zahŕňalo inštaláciu potrebného softvéru. Následne som sa venoval každej komponente, ktorá sa nachádza v informačnom systéme. Dôležitou časťou tejto kapitoly nie je len ukážka samotnej tvorby informačného systému, ale aj ekonomické zhodnotenie môjho návrhu a prínosy mojej diplomovej práce.

ZOZNAM POUŽITÝCH ZDROJOV

- (1) MATLAB FEI TUKE. *Dáta, informácie, znalosti – Kybernetika – TUKE*. *Matlab.fei.tuke.sk* [online]. © 2018 [cit. 2019-02-21]. Dostupné z: http://matlab.fei.tuke.sk/wiki/index.php?title=D%C3%A1ta,_inform%C3%A1cie,_znalosti
- (2) MOLNÁR, Z. *Efektivnost informačních systémů*. Praha: Grada Publishing, 2000. ISBN 80-7169-410-X.
- (3) ŘEPA, V. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. ISBN 80-86119-13-0.
- (4) SODOMKA, P., H. KLČOVÁ. *Informační systémy v podnikové praxi*. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- (5) BASL, J., R. BLAŽÍČEK. *Podnikové informační systémy. Podnik v informační společnosti*. Praha: Grada, 2008. ISBN 978-80-247-2279-5.
- (6) KŘÍŽ, Jiří a Petr DOSTÁL. *Databázové systémy*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8
- (7) KROENKE, David, David J AUER a Jakub GONER. *Databáze*. Brno: Computer Press, 2015, 496 s. : il., grafy, tab. ISBN 978-80-251-4352-0.
- (8) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.
- (9) ITNETWORK. *MySQL krok za krokem: Vytvoření databáze a tabulky*. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-07]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-vytvoreni-databaze-a-tabulky>
- (10) VOHO. *SQL*. *Voho.eu* [online]. ©2008-2019 [cit. 2019-03-07]. Dostupné z: <http://voho.eu/wiki/sql/>
- (11) JUNEXT. *MySQL*. *Junext.net* [online]. ©2002-2017 [cit. 2019-03-04]. Dostupné z: <https://www.junext.net/mysql/>

- (12) QUORA. What is the future of commercially available databases like SQL Server and Oracle since we see so many options which are more efficient and reasonable being available in the market? *Quora.com* [online]. ©2019 [cit. 2019-03-04]. Dostupné z: <https://www.quora.com/What-is-the-future-of-commercially-available-databases-like-SQL-Server-and-Oracle-since-we-see-so-many-options-which-are-more-efficient-and-reasonable-being-available-in-the-market>
- (13) ADAPTIC. Apache server. *Adaptic.cz* [online]. ©2005-2019 [cit. 2019-03-04]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/apache-server/>
- (14) ITNETWORK. Jak psát moderní web. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-08]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-uvod-do-html>
- (15) ITNETWORK. Základní HTML tagy. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-09]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-zakladni-tagy>
- (16) ITNETWORK. Obrázky a odkazy v HTML. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-10]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-obrazky-odkazy>
- (17) ITNETWORK. Tabulky v HTML. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-10]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-tabulky-a-seznamy>
- (18) HELPMARK. Html značka div. *Helpmark.cz* [online]. ©2019 [cit. 2019-03-10]. Dostupné z: <https://www.helpmark.cz/navody/html-css/html-znacky/279-html-znacka-div>
- (19) ADAPTIC. Kaskádové styly. *Adaptic.cz* [online]. ©2005-2019 [cit. 2019-03-11]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/kaskadove-styly/>

- (20) SKLAR, David a Jan POKORNÝ. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Brno: Zoner press, 2018. ISBN 978-80-7413-363-3.
- (21) ITNETWORK. Úvod do JavaScriptu. *ITnetwork.cz* [online]. ©2019 [cit. 2019-03-11]. Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- (22) ITNETWORK. Úvod do CSS frameworku Bootstrap. *ITnetwork.cz*. [online]. ©2019 [cit. 2019-03-11]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>

ZOZNAM POUŽITÝCH OBRÁZKOV

Obrázok č. 1: Dáta, informácie, znalosti a múdrosť (1)	14
Obrázok č. 2: Životný cyklus IS (Upravené podľa 3, s. 18).....	16
Obrázok č. 3: Technologický model podnikového IS (Upravené podľa 5, s. 100)	17
Obrázok č. 4: Komponenty databázového systému (Upravené podľa 7, s. 30).....	18
Obrázok č. 5: Lineárny dátový model (Upravené podľa 8, s. 20)	20
Obrázok č. 6: Relačný dátový model (Upravené podľa 8, s. 21).....	21
Obrázok č. 7: Objektový dátový model (Upravené podľa 8, s. 21).....	21
Obrázok č. 8: Vzťah 1 : 1 (Upravené podľa 8, s. 30)	25
Obrázok č. 9: Vzťah 1 : N (Upravené podľa 8, s. 30)	25
Obrázok č. 10: Vzťah N : M (Upravené podľa 8, s. 30).....	26
Obrázok č. 11: Symboly DFD diagramu (Upravené podľa 8, s.84)	27
Obrázok č. 12: Symboly vývojového diagramu (Upravené podľa 8, s. 90)	28
Obrázok č. 13: Databázové systémy - podiel na trhu (Upravené podľa 12).....	30
Obrázok č. 14: Štruktúra HTML dokumentu (Upravené podľa 14).....	31
Obrázok č. 15: Organizačná štruktúra spoločnosti (Zdroj: vlastné spracovanie)	38
Obrázok č. 16: Komunikácia užívateľa, routeru a serveru (Zdroj: vlastné spracovanie)	39
Obrázok č. 17: Mapa informačného systému (Zdroj: vlastné spracovanie)	40
Obrázok č. 18: E-R diagram databáze (Zdroj: vlastné spracovanie)	42
Obrázok č. 19: Oprávnenie administrátorského účtu (Zdroj: vlastné spracovanie)	51
Obrázok č. 20: Oprávnenie účtu hlavných administrátorov (Zdroj: vlastné spracovanie)	52
Obrázok č. 21: Vytvorenie účtu zákazníkovi (Zdroj: Vlastné spracovanie).....	53
Obrázok č. 22: Vytvorenie administrátorského účtu (Zdroj: Vlastné spracovanie)	54
Obrázok č. 23: Náhľad a editácia údajov (Zdroj: Vlastné spracovanie).....	55
Obrázok č. 24: Navýšenie kreditu (Zdroj: Vlastné spracovanie).....	56
Obrázok č. 25: Náhľad a editácia aktívnych prvkov (Zdroj: Vlastné spracovanie)	57
Obrázok č. 26: Zavedenie aktívneho prvku do systému (Zdroj: Vlastné spracovanie)..	58
Obrázok č. 27: Deaktivácia aktívneho prvku (Zdroj: Vlastné spracovanie).....	59
Obrázok č. 28: Sledovanie počtu prihlásení (Zdroj: Vlastné spracovanie)	60
Obrázok č. 29: Založenie reklamnej kampane (Zdroj: Vlastné spracovanie).....	61

Obrázok č. 30: Zablokovanie reklamnej kampane (Zdroj: Vlastné spracovanie)	62
Obrázok č. 31: Zablokovanie užívateľského účtu (Zdroj: Vlastné spracovanie)	63
Obrázok č. 32: XAMPP Control Panel (Zdroj: vlastné spracovanie).....	66
Obrázok č. 33: Štruktúra tabuliek v databáze (Zdroj: vlastné spracovanie).....	67
Obrázok č. 34: Prihlasovací formulár (Zdroj: vlastné spracovanie).....	68
Obrázok č. 35: HTML tag formulára (Zdroj: vlastné spracovanie).....	68
Obrázok č. 36: Pripojenie salt ku heslu (Zdroj: vlastné spracovanie)	68
Obrázok č. 37: Funkcia na pripojenie do databáze (Zdroj: vlastné spracovanie).....	69
Obrázok č. 38: Zničenie session pri odhlásení (Zdroj: vlastné spracovanie)	69
Obrázok č. 39: Údaje o prihlásenom užívateľovi (Zdroj: vlastné spracovanie)	70
Obrázok č. 40: Typ účtu užívateľa (Zdroj: vlastné spracovanie)	70
Obrázok č. 41: Predvolené zobrazenie menu (Zdroj: vlastné spracovanie)	71
Obrázok č. 42: Javascriptová funkcia na zmenu šírky menu (Zdroj: vlastné spracovanie)	71
Obrázok č. 43: Menu informačného systému (Zdroj: vlastné spracovanie).....	71
Obrázok č. 44: Vrátanie menu do pôvodného nastavenia (Zdroj: vlastné spracovanie) 72	
Obrázok č. 45: Bootstrap card (Zdroj: vlastné spracovanie)	73
Obrázok č. 46: Výber neaktívnych routerov (Zdroj: vlastné spracovanie).....	73
Obrázok č. 47: Bootstrap zoznam (Zdroj: vlastné spracovanie).....	73
Obrázok č. 48: SQL dotazy na zobrazenie údajov na úvodnej stránke (Zdroj: vlastné spracovanie)	74
Obrázok č. 49: Prehľad celkových prihlásení (Zdroj: vlastné spracovanie).....	74
Obrázok č. 50: Objekt router (Zdroj: vlastné spracovanie)	75
Obrázok č. 51: Vytvorenie novej inštancie (Zdroj: vlastné spracovanie).....	75
Obrázok č. 52: Zoznam všetkých routerov u zákazníka (Zdroj: vlastné spracovanie)...	76
Obrázok č. 53: Vytvorenie nového routeru do systému (Zdroj: vlastné spracovanie) ...	77
Obrázok č. 54: Načítanie názvu podnikov (Zdroj: vlastné spracovanie).....	77
Obrázok č. 55: Prvok formulára – Customer (Zdroj: vlastné spracovanie).....	77
Obrázok č. 56: Vyhodnotenie aktivity routeru (Zdroj: vlastné spracovanie)	78
Obrázok č. 57: Deaktivácia routeru (Zdroj: vlastné spracovanie)	78
Obrázok č. 58: Výber routeru na editáciu (Zdroj: vlastné spracovanie).....	79
Obrázok č. 59: SQL dotaz na získanie dát o routeru (Zdroj: vlastné spracovanie)	79

Obrázok č. 60: Naplnenie editačného formulára (Zdroj: vlastné spracovanie)	80
Obrázok č. 61: Vyplnený editačný formulár routeru (Zdroj: vlastné spracovanie).....	80
Obrázok č. 62: Trieda zákazníkov (Zdroj: vlastné spracovanie)	81
Obrázok č. 63: Zoznam všetkých podnikov v IS (Zdroj: vlastné spracovanie).....	82
Obrázok č. 64: Formulár pre pridanie nového podniku (Zdroj: vlastné spracovanie)....	83
Obrázok č. 65: Výber podniku ku editácií (Zdroj: vlastné spracovanie).....	84
Obrázok č. 66: Editačný formulár podniku (Zdroj: vlastné spracovanie)	85
Obrázok č. 67: Formulár na zmenu hesla zákazníkovi (Zdroj: vlastné spracovanie).....	86
Obrázok č. 68: Šifrovanie hesla pri zmene (Zdroj: vlastné spracovanie).....	86
Obrázok č. 69: Formulár pre dobitie kreditu podniku (Zdroj: vlastné spracovanie)	87
Obrázok č. 70: Trigger po dobití kreditu (Zdroj: vlastné spracovanie)	87
Obrázok č. 71: Výber podniku na deaktiváciu účtu (Zdroj: vlastné spracovanie)	88
Obrázok č. 72: Vytvorenie triedy inzerentov (Zdroj: vlastné spracovanie)	88
Obrázok č. 73: Úvodná stránka inzerentov (Zdroj: vlastné spracovanie).....	89
Obrázok č. 74: Formulár pre vytvorenie účtu inzerentovi (Zdroj: vlastné spracovanie)	90
Obrázok č. 75: Formulár pre dobitie kreditu inzerentom (Zdroj: vlastné spracovanie) .	91
Obrázok č. 76: Trigger pre dobitie kreditu inzerenta (Zdroj: vlastné spracovanie)	91
Obrázok č. 77: Funkcia na kontrolu zadaného limitu (Zdroj: Vlastné spracovanie).....	92
Obrázok č. 78: Trieda reklamnej kampane (Zdroj: vlastné spracovanie).....	92
Obrázok č. 79: Príklad filtrácie podľa routeru (Zdroj: vlastné spracovanie).....	93
Obrázok č. 80: Zobrazenie výsledkov filtrácie (Zdroj: vlastné spracovanie).....	93
Obrázok č. 81: Vytvorenie reklamnej kampane (Zdroj: vlastné spracovanie)	94
Obrázok č. 82: Editačný formulár reklamnej kampane (Zdroj: vlastné spracovanie)	95
Obrázok č. 83: Detail reklamnej kampane (Zdroj: vlastné spracovanie)	96
Obrázok č. 84: Výber kampane na deaktiváciu (Zdroj: vlastné spracovanie).....	96
Obrázok č. 85: Zobrazenie všetkých užívateľských účtov administrátorov (Zdroj: vlastné spracovanie).....	97
Obrázok č. 86: Založenie nového účtu administrátorovi (Zdroj: vlastné spracovanie)..	98
Obrázok č. 87: Tabuľka pre deaktivované účty (Zdroj: vlastné spracovanie).....	98
Obrázok č. 88: Trigger pre deaktiváciu administrátorského účtu (Zdroj: vlastné spracovanie)	99

Obrázok č. 89: Výber náhľadu nad prihlásenými užívateľmi (Zdroj: vlastné spracovanie)	100
Obrázok č. 90: Štatistiky prihlásení (Zdroj: vlastné spracovanie).....	100

ZOZNAM POŽITÝCH TABULIEK

Tabuľka č. 1: Základné informácie o spoločnosti (Zdroj: vlastné spracovanie)	37
Tabuľka č. 2: Tabuľka CUST_Customer (Zdroj: vlastné spracovanie)	43
Tabuľka č. 3: Tabuľka ADV_Coin (Zdroj: vlastné spracovanie).....	43
Tabuľka č. 4: Tabuľka ADV_Advertiser (Zdroj: vlastné spracovanie).....	44
Tabuľka č. 5: Tabuľka A_Admin (Zdroj: vlastné spracovanie)	45
Tabuľka č. 6: Tabuľka CP_Advert (Zdroj: vlastné spracovanie)	45
Tabuľka č. 7: Tabuľka CP_Stats (Zdroj: vlastné spracovanie).....	46
Tabuľka č. 8: Tabuľka CP_City (Zdroj: vlastné spracovanie)	47
Tabuľka č. 9: Tabuľka CP_Type (Zdroj: vlastné spracovanie)	47
Tabuľka č. 10: Tabuľka A_Permission (Zdroj: vlastné spracovanie).....	47
Tabuľka č. 11: Tabuľka CP_User (Zdroj: vlastné spracovanie).....	48
Tabuľka č. 12: Tabuľka CP_UserOnline (Zdroj: vlastné spracovanie).....	48
Tabuľka č. 13: SWOT analýza informačného systému (Zdroj: Vlastné spracovanie) ...	49
Tabuľka č. 14: Ekonomické zhodnotenie (Zdroj: Vlastné spracovanie)	101