Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Databázový systém, mobilní aplikace a business plán pro aplikaci Partyboard

Plzeň, 2016 Bc. Jan Novák

Originální zadání

Vložení zadání s kulatým červeným razítkem.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 11. května 2016

Bc. Jan Novák

Poděkování

Chtěl bych poděkovat doc. Ing. Přemyslovi Bradovi, MSc. Ph.D. za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Děkuji také své rodině a přítelkyni za podporu při studiu.

Abstrakt

Cílem této práce je seznámit čtenáře s projektem Partyboard, jehož primární myšlenkou je realizace komplexního systému, který pomocí zpráv přes internet nebo SMS zpráv umožňuje „sblížení“ návštěvníků v klubech. Komunikace probíhá mezi návštěvníky, v případě potřeby i mezi majitelem (manažerem) klubu a na základě tohoto systém poskytuje využití programu pro hosty s různými nápady pro vytváření dalších variant zábavy.

V této práci se čtenář dočte o metodách tvorby business a marketingového plánu a jeho realizaci projektu Partyboard, o navrženém relačním datovém modelu a také o implementaci mobilní aplikace pro tento systém.

S tímto projektem je svázána diplomová práce Bc. Antonína Neumanna s názvem **Jádro aplikace, rozhraní webových služeb a prezentační vrstva pro systém PartyBoard**, která pojednává o realizaci webového serveru a webových stránek.

Abstract

The aim of this thesis is to introduce a project called Partyboard and the primary idea of realization of a complex system, which allows visitors of night clubs and bars to approach and get to know each other via messages over the Internet and text messages. The communication takes place among the visitors and, if wanted, between them and the owner (manager) of the club who can provide and create other options of entertainment for the guests on the basis of this system.

In this paper, the reader can learn about the methods of making a business and a marketing plan and how these are applied in the PartyBoard project, about the proposed relational data model and also about the implementation of mobile applications for the system. The thesis of Bc. Antonin Neumann named **Application core, web-service interface and presentation layer for the PartyBoard system**, which deals with the creation and implementation of web server and websites, is related to this project.

Obsah

1 Úvod

Myšlenka projektu Partyboard vznikla na základě mých vlastních zkušeností při práci v nočních klubech, které mě přivedly k nápadu vytvořit aplikaci umožňující jak „sblížení“ návštěvníků klubu mezi sebou, tak také mezi majitelem (manažerem) klubu a na základě toho vytvořit další varianty zábavy.

Pro realizaci tohoto nápadu jsme se rozhodli s Bc. Antonínem Neumannem vytvořit komplexní projekt Partyboard, jehož primární funkcí by měl být příjem a zobrazování zpráv od zákazníků nočních klubů na televizních obrazovkách a jiných zobrazovacích zařízeních umístěných v klubu. Na základě toho je umožněna konverzace v klubu, uskutečnění soutěží (např. každá 10. zpráva vyhrává), možnost hlasování v anketách (např. DJ na příští akci) aj.

Mojí úlohou v tomto projektu (diplomové práci) je prostudovat business a marketingový plán a na základě analýzy požadavků potenciálních zákazníků navrhnout a realizovat relační datový model pro uchování dat ve zvoleném systému řízení báze dat[1] a dále naimplementovat mobilní aplikaci s funkčností pro běžného uživatele i administrátora.

S projektem Partyboard je ještě svázána diplomová práce Bc. Antonína Neumanna s názvem **Jádro aplikace, rozhraní webových služeb a prezentační vrstva pro systém PartyBoard**, který má za úkol vytvořit korporátní stránky pro tento projekt, vypracovat návrh a naimplementovat rozhraní, které bude využívat mnou realizovaný datový model a zřídit samotné stránky pro zobrazování zpráv. Výsledný projekt si klade za cíl vytvořit komplexní systém s názvem Partyboard, který by měl nabídnout nový druh zábavy do nočních klubů a který je schopný obstát na trhu díky navrženému business a marketingovému plánu.

Obsahem této diplomové práce je popis možností tvorby business a marketingového plánu, jeho samotné sepsání, popis mobilních platforem, technologií a vývoje mobilních aplikací, návrh a popis architektury systému, výběr databázového systému a návrh datového modelu s jeho realizací a také testování aplikace v rámci celého systému. V přílohách je umístěn kompletní realizovaný datový model, včetně jeho detailního technického popisu.

[1] SŘBD Systém řízení báze dat

2 Business plán

Tato kapitola se zabývá teoretickou částí o možnostech tvorby business a marketingového plánu.

Business plán [1, 2, 3, 4] je základní plán, který se zaměřuje na vytvoření podmínek realizace podnikatelského záměru až po jeho zhodnocení a zrevidování. Jedná se o „motto“, které máme neustále před očima a které se přibližuje našemu ideálu, ale zároveň je to dobře promyšlená strategie, která nám umožní ideál realizovat. Stejně tak může být podnikatelský záměr nejprve pouhou myšlenkou nebo inspirací, později je to však důležitý dokument v písemné podobě [3].

2.1 Myšlenka projektu

Není důležitý pouze nápad v čem budeme podnikat, ale stejně tak je důležité vědět, proč vlastně plánujeme podnikat a musíme znát náš cíl. Abychom se mohli rozhodnout, zda podnikat, musíme si odpovědět na otázky: Co je na našem projektu originálního a jaké jsou jeho nejzajímavější vlastnosti? Jde nám jen o to, abychom vydělávali velké množství peněz nebo také o dobrou věc, která nás bude bavit a přinese nám radost? Je naším snem být nezávislí, bez diktátu nadřízených? Budeme podnikat v oboru, ve kterém uplatníme své znalosti a schopnosti? Budeme mít společníky nebo zaměstnance? Jak se budeme případně se společníky dělit o kompetence a zisk? Jak bude naše myšlenka převedena do podoby zisku? Kolik budeme potřebovat zaměstnanců a jakou mají mít kvalifikaci? Kdo nám bude dělat účetnictví? Je psán podnikatelský plán jen pro nás nebo i pro případného investora, banku? Budeme náš plán konzultovat s jinými podnikateli s podobnými projekty nebo budeme jednat v utajení, aby nám naši myšlenku někdo nevzal? Budeme mít v něčem konkurenční výhodu? Co uděláme se ziskem? Investujeme peníze dál a budeme rozšiřovat projekt nebo si je necháme pro vlastní potřebu? V neposlední řadě musíme také vědět, jaké povinnosti pro naši osobu plynou ze zákona, jakou společnost založit, jaké dokumenty k tomu budeme potřebovat, jaká osvědčení a schválení jsou potřebná a jak moc finančně náročné založení firmy bude.

2.2 Metody tvorby business plánu

Dříve, než budeme tvořit business plán, musíme znát náš podnikatelský záměr, který rozložíme do několika částí, z nichž všechny části jsou stejně důležité. Na začátku je vždy myšlenka a teprve od ní se odvíjí popis podnikatelské činnosti, tj. příprava realizace a rozhodnutí, zda je plán uskutečnitelný a udržitelný na trhu. „Firma by měla dělat jenom takové věci, pro které se může nadchnout a zároveň musí pochopit, v čem může a v čem nemůže být nejlepší“ [5]. Musíme vědět a mít jasno kdo, co, komu a jak. Potom následuje realizace, testování a ladění. Pokud až do tohoto okamžiku vše probíhá dobře a potvrdí se, že je tato cesta správná, lze projekt nabízet. V této fázi již dochází k trvalému provozu služby. Aby všechna předchozí práce nepřišla vniveč a výrobek nebo nabízená služba byla schopná obstát se na trhu, zůstává realizátorovi ještě neméně důležitá fáze a to je údržba a rozšiřování produktu až do jeho ukončení. Stejně tak je důležité vědět, že náš business plán (nebo alespoň jeho nejdůležitější části) musí být chráněny a utajeny před konkurencí.

Ideální vzor business plánu neexistuje, přesto je dobré ho sepsat, i když bude vytvořen pouze pro sebe. Dá se sestavovat z různého úhlu pohledu a je nutné, aby byl psán v souladu se skutečností, protože ho musíme chápat jako prostředek k uskutečnění naší vize. Proto ho můžeme podat tak, jak cítíme, použít selský rozum, napsat ho vlastní formou a prakticky. Měl by být především konkrétní a kompletní. Výhoda je, že už při jeho sestavování se vše důkladně promyslí, můžeme se průběžně k němu vracet a hodnotit výsledky, přehodnocovat a doplňovat, popřípadě najít schůdnější cestu k realizaci. Již sepsání podnikatelského plánu nám napoví, zda je projekt realizovatelný a životaschopný. Pokud dojdeme k závěru, že není, projekt neuskutečníme. Business plán je živý dokument, který lze s ohledem na měnící se situace během jeho realizace upravovat a aktualizovat.

Business plán můžeme sepsat velmi jednoduše nebo zvolit složitější variantu. Při vytváření jednoduché varianty (např. u malých projektů do 20 000,- Kč) se analýza zaměřuje pouze na nejdůležitější body. U projektů složitějších, kde je mnohem větší riziko (např. nad 100 000,- Kč), je nutná podrobnější analýza, která je už finančně náročná. Důkladná analýza odhalí chyby a usnadňuje v závěru správné zhodnocení proveditelnosti.

Kromě základního plánu se v business plánu mohou vytvořit i jeho různé varianty dle průběžných analýz nebo změn nepředvídatelného zákona trhu.

Pokud se vytváří business plán pro banku nebo plánujeme získat společníky, měl by je business plán zaujmout. I stručně napsaný by měl být poutavý a emočně zapůsobit na potenciálního investora, společníka či banku. Jeho cílem je zanechat dojem, aby je přesvědčil!

V plánu nesmí chybět jméno autora, název firmy, popř. popis společnosti, místo podnikání, účel podnikatelského plánu, popis výrobku nebo služby, vize a poslání, konkurenční srovnání, silné a slabé články, analýza trhu, dodavatelů a zákazníků, konkurence, strategie a komunikace, marketing a prodejní strategie, nutná investice a finanční plán. Důležité je popsat tým pracovníků, jejich kvalifikaci a kompetence. Na závěr je nutné doložit studie, analýzy, smlouvy, reklamní letáky apod.

2.2.1 Příprava realizace

Cíl – SMART

Slovo SMART je zkratka složená z prvních písmen anglických slov pro mnemotechnickou pomůcku používanou v projektovém řízení ve fázích stanovení cílů. Jedná se o způsob jak hodnotit kvalitu projektových cílů nebo cílů osobního rozvoje [6].

SMART představuje následující kritéria:

- S = specific (jednoznačné)
- M = measurable (měřitelné)
- A = achievable (realizovatelné, akceptovatelné a aktivní)
- R= realistic (reálné)
- T = timed (termínované, časově ohraničené)

Znalost námi vyměřeného cíle je jednou z nejdůležitějších částí projektu a k jeho uskutečnění vede časový plán. O úspěchu a neúspěchu projektu rozhodují především sami podnikatelé a odborné schopnosti managementu (často je důležitější dobrý management než produkt sám). Cíl musí být společný všem, i když každý pracovník má jinou motivaci.

2.2.2 Podnikatelský záměr

V podkapitole podnikatelského záměru nacházíme popis toho, jaká myšlenka nebo zkušenost přinesla nápad, proč zrovna tento produkt uvést na trh a následně vysvětlení, o jaký produkt se jedná a v jaké oblasti se může využívat (jednotlivé osoby, jiné firmy, organizace apod.). Jestliže jsou již nějaké kroky učiněny, je nutné popsat, v jaké fázi se záměr nachází. Aby byl podnikatelský plán v cíli úspěšný, musí z něj mít zákazník užitek a musí být lepší než podobný konkurenční produkt. Pokud jsou k dispozici důkazy, je dobré je doložit.

Konkrétní údaje o firmě

Tak jako je důležité popsat manažerský tým, stejně tak je pro investory důležitá i informace o historii, vývoji a směřování podniku.

Neméně důležitá je též informace, zda již firma existuje, nebo se teprve zakládá. Pokud je již firma založena, musí být uveden její název, právní forma, sídlo a kontakty na kompetentní osoby. Stejně tak nesmí chybět popis předmětu podnikání, strategie, cíle a musí zde být uvedeni i společníci. Je dobré doložit, že firma má všechny zákonem stanovené dokumenty, povolení, popř. patenty.

Pro investory je důležité mít informace o klíčových lidech, kteří se budou na realizaci podnikatelského plánu podílet, jaké pozice budou zastávat, jejich kvalifikaci, zkušenosti, vzdělání, popř. jejich úspěchy z jiných projektů. Popis týmu musí být takový, aby z informací vyplynulo, že je schopen podnikatelský plán úspěšně realizovat.

Produkt

Pokud chceme prezentovat nějaký produkt, je nutné popsat, o jaký výrobek nebo službu se jedná a v čem je originální. Do popisu také patří vlastnosti služby, jak funguje, vysvětlení, v čem je lepší než konkurenční nabídky a proč si zákazník má vybrat právě tento produkt (předpokládá se naše výborná znalost podnikatelského prostředí v daném okruhu služeb), výhody této služby nebo výrobku, servisní podpora a poradenství (vlastní nebo spolupráce s partnerem) a výsledná cena pro zákazníka. U produktu nebo u poskytované služby je nutné znát cenu na trhu (porovnání cen s konkurencí), za jakou cenu se bude prodávat, jakou hodnotu má pro zákazníka, jaký servis

bude poskytnut a jak se zajistí, aby firma byla rychle schopná reagovat na změny, požadavky a přání zákazníka.

Výrobní plán

Je dobré popsat celou strukturu procesu výroby i s časovým harmonogramem. Do plánu zahrnujeme časové údaje - kdy se výrobek bude užívat (existují i sezónní výrobky), kolik čeho bude zapotřebí a kolik čeho bude vyráběno, tzn. znalost o nákupu materiálu, jaké vybavení bude potřeba a předpokládaný vývoj produkce. Časový harmonogram se skládá z několika milníků, které zahrnují vytvoření a realizaci projektu, kroky, které s tím souvisejí a termíny k jejich dosažení. Kvalitní výrobní plán obsahuje i údaje o personálu, který zajišťuje jednotlivé úkony a informace o požadavcích na údržbu.

Organizační plán

Struktura podniku je závislá na velikosti firmy. U větších firem je dobré popsat jednotlivé pozice a to včetně požadavků na zaměstnance. V případě nedostatku zaměstnanců je vhodné mít plán a vědět, při jakém minimálním počtu zaměstnanců je ještě podnik schopen zajistit řádný chod firmy, jaká je možná zástupnost jednotlivých pozic a jak dlouho může trvat, aby neohrozila stabilitu firmy. Podle jedné americké studie velmi záleží na osobnostech nejen ve vedení firmy, ale i ostatních zaměstnanců na všech úrovních. Může zde být popsán způsob náboru, výhodné nabídky a benefity pro zaměstnance, jejich možnost profesního růstu ve firmě a politika odměňování [5]. U malých firem takové nebezpečí většinou nehrozí, pokud je podnikatel schopen (alespoň krátkodobě) vše zajistit sám.

Finanční plán

Po sepsání předchozích bodů a důkladném zhodnocení se přechází k finančnímu plánu. Zde se ukáže, zda je produkt životaschopný a plán z finančního pohledu uskutečnitelný. Zhodnotí se a doloží podklady zisků a ztrát (hodnocení rizik), náklady a výnosy za určité období (rentabilita), plán toku peněz a efektivnost investic. Je důležité rozlišit zisk a tok peněz, kdy tok znamená předpokládané příjmy i výdaje související s činností, zatímco zisk je rozdíl mezi výnosy a náklady. K finančnímu plánu náleží i body jako jsou propagace, prodej, finanční zdroje a dluhy v danou chvíli a analýza vývoje. Dalším bodem je vyčíslení nákladů před spuštěním projektu a během jeho fungování, zajištění vedení účetnictví, výběr banky (plánovaný průběh platby převodem a platby v hotovosti), předpoklad příjmů a finanční zajištění. Současně se při

psaní plánu rozhodne, zda se přizve ke spolupráci investor.

Podle J. L. Hesketta je ve službách finanční plán prvotním zdrojem informací o projektu. Data jsou nejcennější „surovinou“, jakou mnohé servisní firmy vlastní, přesto se to nikdy neobjeví v rozvaze [7].

Cenová politika je základem úspěšného marketingu [8], a proto je k úspěšnému sestavení finančního plánu správné stanovení ceny. Při jejím stanovování je důležité vědět, jaké jsou ceny na trhu a zároveň znát cíle cenové politiky, náklady, poptávku a fáze životního cyklu produktu. To vše musí být podřízeno k co nejlepšímu zisku. Může se v krátkodobém intervalu vyskytnout i ztráta, ovšem nemělo by to vést k pouhému trvalému přežívání.

Dále do finančního plánu musíme zahrnout následující položky:

- Prodejní cena – budou naše ceny v porovnání s konkurencí průměrné nebo mírně podprůměrné?
- Bonusy, cenová zvýhodnění, kupónový prodej, věrnostní programy, spotřebitelské soutěže, soutěže pro obchodní partnery aj.

Marketingový plán

Osou marketingových nástrojů je plánování a musí být i součástí podnikatelského plánu. V podstatě se hodnotí, jak jsou využité finance, které se do něj vloží. Pokud nebudou odběratelé, podnikatelský záměr ztroskotá. Záleží na tom, aby zákazník o produktu věděl (nejlépe již v předstihu) a na produkt se těšil. Důležité je stanovení ceny, marketingový rozvrh ceny při vstupu, udržitelnost, bonusy, množstevní slevy, výhodné balíčky, stanovení, jak se bude k propagaci využívat reklama a jakými médii. Dále pak jak bude probíhat distribuce, zda je produkt balený a v jakém obalu, kdo ho navrhne, jak bude probíhat rozvoz, zda budou využívány některé konkrétní prodejny a podrobně popsat cestu distribuce až k určenému zákazníkovi. Rozsah i obsah marketingového plánu závisí na tom, zda se bude jednat o výrobek nebo službu a zda se jedná o firmu „tradiční“ nebo „internetovou“. Správně vyhotovený marketingový plán pomáhá zkvalitnit a zefektivnit celý proces a tím přináší i větší zisky. Pokud se firma pravidelně vrací ke svému plánu a hodnotí dosažené výsledky, může rychleji reagovat a odstraňovat případné problémy.

K sestavení marketingového plánu je důležité uvědomit si jednotlivé nástroje

marketingového mixu. Podle doc. Petera Stoličného [8] rozlišujeme dva marketingové mixy:

- Marketingový mix 4P (product, price, place, propagation – výrobek, cena, distribuce, propagace. Někde je možno setkat se s modelem 5P (+ people – lidé).
- Marketingový mix 4C (customer, cost, convenience, communications – řešení potřeb zákazníka, náklady, které zákazníkovi vznikají, pohodlná dostupnost, komunikace).

Z toho vyplývá, že by firma měla více myslet z pohledu zákazníka a MM (marketingový mix) by se měl přizpůsobit povaze nabízeného výrobku nebo službě.

Zákazník se ale o firmě ani o produktu nedozví bez správné propagace [9]. Proto je nutné vědět, jak a čím zaujmout, jakým způsobem bude propagace probíhat a zvolit její efektivní způsob, vytvořit pozitivní postoj u spotřebitelů a zaměřit se na budování dobrých vztahů se zákazníky. Osobní prodej je nejefektivnější (lze při tom reagovat na náladu i postoj potenciálního zákazníka a přizpůsobit nabídku jeho přáním), ale je finančně nejnáročnější. Vedle osobního prodeje je další možností reklama, u které je však negativní stránkou chybějící zpětná vazba a klesá u ní naléhavost a přesvědčivost. Podle Adriana Payne [2] má být reklama dominantním nástrojem propagace, obchodní jednání může být formálního nebo neformálního charakteru. Problém ale může nastat při nabídce služby, která není hmatatelná. Při inzerci je důležité slíbit jen to, co je možné splnit a vysvětlit službu tak, aby byla pochopena.

Podle Martiny Blažkové [1] v komunikační strategii nesmí chybět zhodnocení současné situace, pozice služby nebo výrobku na trhu, musíme vědět, čeho chceme dosáhnout (např. zvýšit povědomí o firmě), na jakou cílovou skupinu se zaměříme a jakými prostředky, znát její životní styl, jaká média využijeme, co vše budeme říkat a rozhodnout, kdy bude nejlepší čas a jak dlouho v předstihu poslat veřejnosti správné informace a posléze do kdy bude účelné informace podávat. Pokud je cílová skupina složená z mladších osob, je vhodné využívat pro vzájemnou komunikaci internetové prostředí. Důležité je ověřovat si zpětnou vazbu, reakci veřejnosti na naši nabídku a zhodnotit, zda jsme dosáhli našeho cíle. Stejně tak je důležité stanovit si rozpočet, jaké budou náklady, kdo to vše udělá a kdo za celou oblast komunikační strategie bude zodpovídat.

SWOT analýza

Zkratka SWOT je odvozena z prvních písmen anglických názvů a jedná se o metodu, díky které je možno identifikovat silné a slabé stránky, příležitosti a hrozby, které jsou spojeny s určitým projektem, typem podnikání, podnikatelským záměrem apod. Výstupem SWOT analýzy by mělo být zjištění, jak pomoci při výběru vhodné strategie a chování společnosti, která maximalizuje přednosti a příležitosti a minimalizuje své nedostatky a hrozby.

- S = strengths (silné stránky - klady)
- W= weaknesses (slabé stránky - zápory)
- O = opportunities (příležitosti)
- T = threats (hrozby)

Mezi silné stránky patří zkušený management, kvalifikovaný servis, originalita produktu, popis, v čem a zda je zde časový náskok před konkurencí apod. Pokud jsou známé některé slabé stránky, tak je nutné navrhnout řešení, jak je překonat. Příležitosti značí atraktivnost produktu, možnosti dalšího rozšiřování a dalšího zisku nebo úspor.

Podle doc. Petera Stoličného [8] dvojice „klady - zápory“ vyjadřuje vnitřní charakteristiku podniku, v čem jsme silní a v čem slabí. Dvojice „příležitosti – hrozby“ vyjadřuje vnější prostředí, ve kterém se podnik nachází. Nejdůležitějším znakem SWOT má být objektivita, proto je lepší, aby SWOT provedla profesionální firma, která nemá přímé vazby s podnikem. To umožňuje najít hrozby a úskalí z okolí podnikání, zamyšlení, jak tomu předejít a najít řešení dopředu. Je dobré je uvést, aby nedošlo během realizace k nečekaným zádrhelům. „Uznejte nepříjemná fakta a neřiďte se stylem pokus – omyl a naučte se hledat shodu“ [10].

Analýza rizik

Analýza rizik umožňuje dva pohledy na projekt – pravděpodobnost jejich výskytu a intenzitu negativního vlivu. Uvědoměním si rizik můžeme připravit plán a v případě, když nastanou problémy víme, jakým způsobem situaci řešit, popřípadě jim předejít. Víme, která rizika ovlivnit můžeme (vnitřní) a která ovlivnit nemůžeme (vnější). Vnější rizika se zjišťují většinou pouze odhadem – např. síla konkurentů, jejich silné a slabé stránky.

Na základě analýz se mohou udělat opatření k minimalizaci rizik, připravit strategie preventivních opatření k jejich snížení, např. pojištění, dělení rizik, uzavírání dlouhodobých smluv apod.

Může se stát, že spolehlivé podklady k některým analýzám nejsou k dispozici, v tom případě se dělá odhad. Odhad je důležité vytvořit alespoň na spolehlivých základech, aby byl logický a porovnatelný s podobnými produkty.

K uskutečnění cíle je dobré mít analýzy:

- trhu - je nutné znát mezery na trhu v dané oblasti podnikání, jaký je potenciál dostat se na trh, překážky při vstupu a zájem budoucích zákazníků (stačí ty nejdůležitější údaje, celková analýza trhu je velmi nákladná);
- konkurentů (jmenovitě) - potřebné je zjištění jejich konkurenční síly a naší pozice mezi nimi (přednosti a nedostatky, dostupnost a šíři okruhu jejich podnikání). Konkurence se nesmí podceňovat, v dnešní době je konkurence téměř u všech produktů, je to velmi důležitý bod;
- dodavatelů - je dobré vyjmenovat a vysvětlit, proč právě tito dodavatelé byli vybráni a popsat stupeň závislosti na jejich dodávkách, jejich spolehlivost a cenu;
- odběratelů (zákazníků) - určit, kteří jsou klíčoví (člověk na různé úrovni svého postavení má také různé požadavky na své potřeby), znát jejich problémy, odhad tržní velikosti a růstový potenciál. Je dobré se vžít do jejich vnímání [11].

Ostatní informace jsou stejně důležité

Je nutné rozhodnout, zda bude použita obchodní značka, logo, jaký zvolit design, zda využívat služeb poradců, jak bude smluvně zajištěna mlčenlivost zaměstnanců (ochrana duševního vlastnictví), jasně stanovit podíly společníků při vstupu při založení firmy i během fungování, aby nedošlo k pozdějším sporům o vlastnictví apod.

2.2.3 Testování a ladění: typy testování

Je důležité si uvědomit a popsat, jakou cestou se bude produkt testovat. Některý výrobek se zkouší již během vývoje ve firmě, jiný se musí dát k testování veřejnosti. Musíme určit, jakému okruhu spotřebitelů produkt k testování poskytnout a za jakých podmínek, aby byla zpětná vazba. Povědomí o spokojenosti zákazníků je důležité k následnému vyhledaní chyb a zajištění vylepšení. Produkt může být poskytnut zdarma nebo za určitou nižší částku, můžeme využít dotazníkovou formu po telefonu, prostřednictvím fyzicky vyplněného dotazníku nebo dotazníku vyplněného přes internet, lze svolat přes Facebook setkání a účastníky testování zaplatit, pohostit apod.

2.2.4 Trvalé provozování služby a rozšiřování produktu

Aby mohlo dojít k trvalému provozu výroby nebo stále dodávat popř. rozšiřovat službu, je důležité mít trvalé odběratele. Získávat stále nové zákazníky je nákladné, proto je velmi potřebné si již získané zákazníky udržet [12]. Znamená to nejen odvádět dobrou práci, ale také být se zákazníky v kontaktu, vzbuzovat důvěru, všímat si nových trendů a nabízet inovaci. Spokojení zákazníci jsou zdrojem referencí [1]. Každý zákazník velmi ocení nabídnutý servis, který obstará buď přímo prodávající firma, nebo bude alespoň v roli zprostředkovatele. Zákazníkovi potom stačí jedna adresa nebo jedno telefonní číslo k vyřízení svých přání. Zároveň má prodejce zpětnou vazbu se zákazníkem a může reagovat na jeho další návrhy a potřeby.

3 Potenciální potřeby zákazníků a návrh business plánu

Na základě obecných poznatků o tvorbě business plánu popsaného v předchozí kapitole je vysvětleno, jak je vytvořený a co obsahuje plán pro realizaci projektu Partyboard.

3.1 Myšlenka a cíl projektu Partyboard

Myšlenkou projektu je příjem a zobrazování jak internetových, tak SMS zpráv od uživatelů nočních klubů na televizních obrazovkách nebo jiných zobrazovacích zařízeních. Na základě toho bude umožněna konverzace v klubu, uskutečnění různých soutěží (např. každá 10. zpráva vyhrává), možnost hlasování v anketách (např. téma/DJ na příští akci) atp.

Primárním cílem projektu Partyboard je poskytnutí nového druhu zábavy do nočních klubů, jeho samotná realizace za účelem zisku na základě jeho poskytování a možnosti zobrazování reklam od různých společností.

3.2 Příprava realizace projektu Partyboard

3.2.1 Podnikatelský záměr

Myšlenka aplikace Partyboard vznikla z mé vlastní zkušenosti při práci v klubech. Mnoho let jsem se pohyboval v prostředí diskoték a nočních klubů jako DJ. Nešlo však jen čistě o práci DJe, podílel jsem se také na pořádání a realizování akcí, u kterých jsem se snažil vždy vymyslet něco originálního, co by návštěvníky bavilo a přivedlo na danou akci. A právě tyto moje výše zmíněné zkušenosti „z terénu“ mě přivedly k nápadu vytvořit aplikaci, která by umožňovala „sblížit“ se s návštěvníky v klubu a vytvořit další varianty zábavy.

Primárně aplikace slouží pro příjem a zobrazování zpráv od uživatelů na mul-

timediálních zařízeních jako jsou např. televizní obrazovky, plátna, mobilní telefony a jiné. Jejich prostřednictvím je možné realizovat konverzaci v klubu a to jak mezi samotnými zákazníky, tak také samozřejmě s majitelem klubu (případně manažerem, DJem atd.) a navázat tak kontakt mezi všemi.

Partyboard je jedinečný systém zábavy, který se s tímto druhem zábavy momentálně na českém trhu nevyskytuje. Potenciální zákazníci jsou majitelé nočních klubů, kterým tato aplikace nabízí nový druh zábavy a tím se stává pro hosty jejich podniků přitažlivější.

Proč mít Partyboard:

- originalita klubu
- zabavení hostů
- nepřeberné množství soutěží
- možnost zobrazení reklam, plakátů a pozvánek na další akce
- možnost komunikace s hosty
- propagace registrovaných klubů na stránkách služby Partyboard
- náskok oproti konkurenci

Předmětem podnikání je poskytování služby Partyboard a s ním spojená podpora a rozšiřování aplikace.

Konkrétní údaje o firmě

V tuto chvíli není firma založena, mělo by však v nejbližší době dojít k založení společnosti s ručením omezeným s názvem „Partyboard s.r.o.“.

Tato společnost by měla mít sídlo v Českých Budějovicích a jejími kompetentními osobami jsou výše uvedení Bc. Jan Novák a Bc. Antonín Neumann. Postavení společníků by mělo vyplynout z podílu firmy na základě základního kapitálu, který je stanoven na 10 000 Kč. Mělo by se jednat o poměr 51% (5 100 Kč) ku 49% (4 900 Kč) pro J. Nováka vůči A. Neumannovi, jelikož J. Novák je autorem myšlenky projektu.

Zakladatelem projektu je Bc. Jan Novák a Bc. Antonín Neumann. Oba jsou

studenty informatiky v posledním ročnímu Magisterského studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Bc. Jan Novák má praxi s návrhem a realizací ERA modelů a zkušenosti s implementací mobilních aplikací. Momentálně pracuje ve společnosti Datalite spol s.r.o. na pozici Java programátora. V tomto projektu má na starosti návrh a realizaci databáze, implementaci mobilních aplikací a samotné manažerské povinnosti pro realizaci a provoz projektu Partyboard.

Bc. Antonín Neumann má dlouholetou praxi v realizaci webových stránek a momentálně pracuje na pozici programátora ve firmě Socialbakers a.s. Na tomto projektu má na starosti návrh a realizaci aplikační rozhraní aplikace (server) a realizaci korporátních webových stránek a webových stránek pro jednotlivé Partyboardy.

Toto rozdělení úkolů by mělo zůstat i při provozu služby.

Popis služby

Partyboard je jedinečný systém, který je primárně určen pro kluby (diskotéky, bary a jiné noční podniky) a slouží pro příjem a zobrazování zpráv od uživatelů na multimediálních zařízeních (televizních obrazovkách, plátnech, mobilních telefonech) nebo jiných zobrazovacích zařízeních. Jeho prostřednictvím je možné provozovat konverzaci v klubu a to jak mezi samotnými zákazníky, tak také samozřejmě s majitelem klubu (případně manažerem, DJem atd.) a navázat tak kontakt s návštěvníky nočního klubu.

Potenciální zákazník (ve většině případů majitel klubu) kontaktuje provozovatele Partyboardu pomocí online registrace a zašle žádost na zřízení služby. Následně dojde k vyřízení žádosti a pak už jen stačí otevřít si zaregistrovaný Partyboard v nočním klubu a tímto krokem může započít konverzace návštěvníků klubu pomocí zpráv.

Jedná se o aplikaci, která se bude stále vyvíjet a umožňovat vytvářet různé druhy zábavy, např. bude mít schopnost uspořádat přes tuto službu soutěže. S aplikací Partyboard se nápadům opravdu meze nekladou, může být dokonalým společníkem na party a jeho prostřednictvím se mohou dělat nezapomenutelné akce.

Partyboard umožnuje vytvořit jedinečnou konverzaci v klubu, do které může vstoupit i majitel klubu (případně manažer, DJ atd.). Správci aplikací Par-

tyboard v jednotlivých klubech mohou samozřejmě také přispívat do zpráv mezi návštěvníky, ale také budou moci nevhodné zprávy či nevyžádané reklamy mazat. Hodí se dokonale pro akce jako jsou Seznamkové party, Love chat nebo jako výherní portál, na kterém bude možné určitými SMS zprávami vyhrát některou z cen. Může také fungovat ale i jako playlist pro celý večer. Dále umožňuje vkládání systémových zpráv, které se automaticky zobrazují po určité době, jako např. zveřejnění vlastní reklamy v grafickém formátu. Díky tomu lze návštěvníkům podávat informace o všem, co si správce bude přát.

Servisní podpora a poradenství
Servisní podporou a poradenstvím se myslí pomoc při nastavení služby a případných problémech s jejím fungováním. Při nefunkčnosti služby dojde k přezkoumání problému a zjištění, na které straně došlo k chybě. Pokud došlo k problémům na straně poskytovatelů služby, dojde k analýze chyby a následné opravě.

Tyto služby budou zajištěny pomocí elektronické pošty či telefonicky samotnými zakladateli (realizátory) projektu a tím bude nejlépe zajištěna odpovídající podpora a poradenství vyplývající z dokonalé znalosti služby. Časy podpory jsou popsány v jednotlivých verzích produktu v podkapitole **Finanční plán a cena produktu**. Při velkém zájmu o tuto službu by samozřejmě došlo k přijetí nových zaměstnanců na tuto pozici a k jejich následným zaškolením.

Cena služby
Cena služby se bude odvíjet dle druhu zvolených nabízených balíčků poskytovateli, viz bod Finanční plán a cena produktu. Bude se jednat o paušální typ služby, na pronajímat po jednotlivé měsíce.

Reakce na změny
Reakce na požadavky od zákazníků by měla být co nejrychlejší, jelikož je aplikace postavena na principu pravidelného a dlouhodobého užívání a tím pádem pro ně musí být co nejpřívětivější, aby docházelo k jejímu častému využívání.

Výrobní plán (dílčí kroky pro realizaci projektu)

Dílčími kroky pro realizaci projektu je sepsání analýzy projektu, návrh datové

vrstvy, návrh API[1], samotná implementace datové vrstvy a serveru podle aplikačního rozhraní, vývoj korporátních stránek a samotné stránky pro jednotlivé Partyboardy, vývoj mobilní aplikace a testování systému. Dále do výčtu kroků patří také založení společnosti s ručením omezeným, získání zákazníků a provoz služby, rozšiřování a podpora služby.

Organizační plán

Struktura podniku je složená ze dvou osob. Bc. Jan Novák má primárně na starosti marketing a podporu služby Partyboard a Bc. Antonín Neumann má na starosti technickou stránku služby Partyboard. Tímto je stanoven i nejmenší počet osob, aby nedošlo k ohrožení stability firmy. O účetnictví se bude starat najímaná externí firma.

Při velkém zájmu o tuto službu ze strany klubů by se počet zaměstnanců navýšil a to především na pozici konzultant (prodejce), podpora systému (helpdesk), případně na pozici PHP a JavaScriptového programátora.

Finanční plán a cena produktu

Výnosy firmy budou převážne z prodeje služby Partyboard. Kromě nákladů na založení firmy a realizace projektu bude mít firma také pravidelné náklady na provoz, mezi které patří pronájem hostingu domény, serveru, GSM SMS brány atd.

Služba Partyboard bude provozována v základních 6 balíčcích, ze kterých si zákazník bude moci vybrat pouze jeden (jednotlivé balíčky nelze kombinovat). Balíčky jsou navzájem od sebe odlišené množstvím, druhem poskytovaných služeb a paušální cenou. Smyslem je navázat dlouhodobé vztahy se zákazníky, což by přineslo samozřejmě i dlouhodobé zisky.

Balíček Free
Jedná se o testovací balíček platný po dobu jednoho měsíce, který je zcela zdarma a na jednoho registrovaného uživatele lze uplatnit tento balíček pouze jedenkrát. Zákazník má však minimální možnosti pro svoje nastavení Partyboardu:

- nelze přidávat vlastní reklamy

[1] API Application Programming Interface

- lze měnit pozadí pouze z předvolených možností
- nelze změnit úvodní logo
- nelze zpětně procházet příchozí zprávy
- lze využívat soutěží
- reklamy vkládají pouze administrátoři systému
- podnik není propagován na webových stránkách Partyboardu
- podpora telefonicky i přes e-mail Po-So 10:00 – 15:00 hod.
- možno vytvořit pouze jeden Partyboard pro klub

Balíček Basic
Jedná se o základní balíček v hodnotě 500,- Kč/měsíc.

Možnosti nastavení:

- lze přidat 1 vlastní reklamu
- lze měnit pozadí pouze z předvolených možností
- lze změnit úvodní logo
- lze zpětně procházet příchozí zprávy
- lze využívat soutěží
- ostatní reklamy vkládají administrátoři systému
- podnik není propagován na webových stránkách Partyboardu
- podpora telefonicky i přes e-mail Po-So 10:00 – 18:00 hod.
- možno vytvořit pouze jeden Partyboard pro klub

Balíček Medium
Jedná se o střední balíček v hodnotě 800,- Kč/měsíc.

Možnosti nastavení:

- lze přidat až 3 vlastní reklamy
- lze měnit pozadí pouze z předvolených možností
- lze změnit úvodní logo
- lze zpětně procházet příchozí zprávy
- lze využívat soutěží
- ostatní reklamy vkládají administrátoři systému
- podnik je propagován na webových stránkách Partyboardu
- podpora telefonicky i přes e-mail Po-So 10:00 – 22:00 hod.
- možno vytvořit až 2 Partyboardy pro klub

Balíček Full
Jedná se o plný balíček v hodnotě 1500,- Kč/měsíc.

Možnosti nastavení:

- lze přidávat až 5 svých reklam
- lze upravit pozadí dle požadavků zákazníka
- lze změnit úvodní logo
- lze zpětně procházet příchozí zprávy
- lze využívat soutěží
- pouze 1 vložená reklama administrátory
- podnik je propagován na stránkách webových Partyboardu
- podpora telefonicky i přes e-mail Po-So 10:00 – 22:00 hod.
- možno vytvořit až 5 Partyboardů pro klub

Balíček Diamant

Jedná se o speciální balíček, kdy bude Partyboard vytvořen přímo na míru dle požadavků zákazníka s 24 hodinovou podporou. Cena dohodou.

Balíček DJ
Jedná se o základní balíček v hodnotě 350,- Kč/měsíc. Tento balíček je jako jediný možné přenášet mezi kluby a díky tomu může být DJ žádanější než DJ bez Partyboardu, jelikož může do klubu majiteli přivést více lidí díky vlastnictví Partyboardu.

Možnosti nastavení:

- nelze přidávat vlastní reklamy
- lze měnit pozadí pouze z předvolených možností
- lze změnit úvodní logo
- nelze zpětně procházet příchozí zprávy
- lze využívat soutěží
- reklamy vkládají pouze administrátoři systému
- podnik/Dj není propagován na webových stránkách Partyboardu
- podpora telefonicky i přes e-mail Po-So 10:00 – 18:00 hod.
- možno vytvořit pouze jeden Partyboard pro klub

Marketingový plán

Celá myšlenka projektu Partyboard je postavena na tom, že majitelé klubů vlastnící tuto službu budou mít náskok oproti konkurenci a to díky tomu, že jim tato služba umožní nový druh zábavy a tím si obohatí program ve svém klubu. Netradiční zábava znamená přísun nových návštěvníků a tím pádem větší zisk pro majitele klubu. Čím více podniků bude vlastnit službu Partyboard, tím více lidí ji bude využívat a čím více lidí jí bude využívat, tím bude větší výnos za cílenou reklamu a tím se budou zvyšovat zisky i majitelů Partyboardu.

Komunikace se zákazníkem (majitelem klubu) bude primárně probíhat pomocí korporátních stránek a osobním kontaktem jak už ze strany potenciálních zákazníků, tak i kontaktováním potenciálních zákazníků ze strany provozovatele služby Partyboardu.

Podnětem ke zřízení služby je požadavek ze strany potenciálního zákazníka a následně jejího internetového (elektronického) zřízení ze strany provozovatele Partyboardu. V případě zájmu je možný i osobní servis či zaškolení přímo samotného zákazníka.

Abychom mohli správně stanovit způsob, kterým budeme produkt nabízet, musíme určit, na jaký sektor populace budeme cílit. Služba je primárně určena pro Českou republiku, avšak při jejím velkém úspěchu by se mohla rozšířit i do ostatních státu Evropy. Potenciální zákazníci jsou rozděleni do segmentů podle typu (kvality) klubů. Toto rozdělení je účelné pro lepší pochopení zákazníků v konkrétních segmentech a nabízí jim právě to, co by jim mohlo pomoci v rozvoji svých podniků.

Rozdělení klubů do segmentů

1. Obyčejné kluby většinou na okraji měst - Těmto klubům by měla být primárně nabízena nejlevnější verze produktu s možností využívání soutěží, protože tyto kluby mají převážně „lidovější“ ceny a dá se předpokládat, že nevydělávají tolik, jako například luxusní kluby v centru měst. Do těchto klubů se stahuje většinou sociálně slabší populace, která nemá srovnatelné příjmy s lidmi, kteří navštěvují kluby luxusní. Ovšem vidina výhry za poslání neplacené zprávy by je mohla zaujmout a tím i přilákat více hostů do klubu.

2. Střední kluby – Těmto klubům by měla být především nabízena střední verze produktu a to z toho důvodu, že kluby jsou většinou multižánrové a tím pádem do nich chodí lidé, kteří si vybírají zábavu podle typu akce a zpravidla mají větší příjem než populace návštěvující kluby obyčejné.

3. Luxusní kluby – Těmto klubům by měla být primárně nabízena jedna z nejvyšších verzí produktu. Majitelé těchto klubů nemají většinou problém investovat do něčeho nového, co je zviditelní a dá jim náskok a ještě větší originalitu oproti ostatním klubům. Podobně je tomu i s návštěvníky, kteří mohou utratit větší finanční obnos za placené zprávy.

4. Tematicky založené kluby a kluby studentské – Těmto klubům by měla být primárně nabízena střední verze produktu, avšak s tím, že by se pro

tyto kluby připravovaly tematické soutěže. Majitelé těchto klubů nemají většinou nouzi o návštěvníky. Příkladem nám může být např. karaoke bar se soutěží o nejlepší zpěvačku večera, kterou si svými hlasy zvolí návštěvníci karaoke pomocí služby Partyboard. Tento druh hlasování zajistí náskok před podobně laděnými kluby, kde by se podobné soutěže realizovaly velmi těžko.

Popis jednotlivých verzí produktu je v bodu **Finanční plán a cena produktu**. Na základě osobních zkušeností z tohoto prostředí jsem získal kontakty na majitele klubů z jednotlivých segmentů po celé republice, a proto bych přednostně začal s nabízením tohoto produktu právě u nich.

SWOT analýza

Tato SWOT analýza je vytvořena pro projekt Partyboard z pohledu trhu.

Obrázek 3.1: SWOT analýza systému Partyboard

Analýza k uskutečnění cíle projektu a rizik

V dnešní době není na trhu podobná služba, jako je navrhovaná aplikace Partyboard, což dává velkou příležitost se s ní prosadit. Doposud není známa žádná přímá konkurence mimo reklamních televizí umist'ovaných v klubech a systému řetězce „The PUB“, který zobrazuje množství vypitých piv v jejich jednotlivých podnicích. Otázkou zůstává, zda se neobjeví konkurenční nabídka při vypuštění této služby do „světa“.

Hlavní překážkou při provozu ze strany zákazníků (návštěvníků klubů) by mohla být snad pouze stydlivost či neochota využívat tuto službu. Avšak tuto překážku ve velké míře může ovlivnit nevhodné využívání služby správcem aplikace v klubu. Celý koncept aplikace je postaven na principu poskytnutí nového druhu zábavy do nočních klubů, avšak jak bude tato aplikace v konkrétním klubu využívaná, je už pouze na aktivitě správce a zapojení návštěvníků do konverzace.

Dodavatelem pro tuto službu je zároveň sám poskytovatel serveru (hostingu) a poskytovatel GSM SMS brány. Pro účely testování a vyvíjení aplikace je aktuálním poskytovatelem školní server Západočeské univerzity v Plzni a aktuálně využívaná GSM SMS brána od společnosti ProfiSMS s.r.o. Před zavedením služby do ostrého provozu dojde k analýze a výběru stabilního poskytovatele hostingu s dlouholetou zkušeností, který zároveň bude splňovat požadavky na plynulý provoz služby. To samé platí i pro poskytovatele GSM SMS brány.

Primárním odběratelem jsou majitelé nočních klubů, kteří musí „bojovat“ o své zákazníky ve velké konkurenci. Na základě zkušeností z tohoto odvětví vím, že není lehké vymýšlet akce na každý týden tak, aby majitel klubu dosáhl zisku a zároveň motivoval návštěvníky k další návštěvě. Tato služba jim poskytne možnost pořádat mnoho originálních akcí, při kterých mohou navázat kontakt s hosty a vyplnit tak jejich aktuální tužby (např. během večera jim připravit soutěž na nápoje zdarma), což například komunikace přes sociální sítě a jiné komunikační portály neumožňují.

Faktickými náklady na tento projekt je strávený čas nad jeho realizací a placený hosting s doménou pro korporátní stránku Partyboardu. Při nasazení aplikace do ostrého provozu náklady vzrostou o placení serveru pro provozování aplikace, poplatek za založení společnosti s ručením omezeným, sepsání právních podmínek pro provoz aplikace a finanční odměny za odvedenou práci. Údržba je brána jako podpora služby a její další rozšíření funkčnosti.

Celkové snížení možnosti výskytu rizik by mělo být eliminováno na základě výše popsané analýzy celé problematiky projektu.

Vnější rizika

Za vnější riziko se považuje v našem případě konkurence. Vzhledem k tomu, že si nejsem v současné době vědom žádné podobné aplikace na trhu, tak je toto riziko velmi malé. Momentálně jediná známá konkurence je umist'ování reklamních televizí v podnicích a systém řetězce „The PUB“.

Za vnitřní rizika lze považovat:

- výpadek či pomalá odezva serveru při velkém množství zpráv;
- chyba v logice programu;
- výpadek GSM SMS brány.

Riziko výpadku a pomalé odezvy serveru lze minimalizovat zajištěním kvalitního poskytovatele serveru s dobrým připojením. Stejné řešení platí i pro GSM SMS bránu pro příjem a odesílání SMS zpráv. Riziko chyby v logice programu lze minimalizovat na základě provedení detailních testů, viz kapitola Nasazení a testování systému Partyboard.

Ostatní informace ke službě Partyboard

Tento projekt bude na veřejnost vstupovat pod názvem Partyboard a bude disponovat logy v různém barevném provedení. Před nasazením služby do provozu dojde k založení společnosti s ručením omezeným, kde podíl společnosti bude rozdělen na 51% pro Bc. Jana Nováka a 49% pro Bc. Antonína Neumanna. Mlčenlivost zaměstnanců bude zajištěna právně sepsanou smlouvou o mlčenlivosti.

3.2.2 Testování a ladění služby Partyboard

Před samotným vypuštěním služby do ostrého provozu by mělo dojít ke 3 typům testování.

Jako první by mělo dojít k otestování API, které by mělo ověřit správnou

funkčnost serveru a datového modelu. Toto testování funguje podobně jako jednotkové testy. Jedná se o zaslání požadavku na server a následně zkontrolování odpovědi ze serveru.

Dalším testem by měly být již uživatelské testy přes naimplementované aplikace. Tyto testy budou prováděny samotnými vývojáři.

Posledním testem by mělo být samotné pilotní testování v některém z vybraných podniků, kterým bude služba poskytnuta za jejich report po jejím vyzkoušení.

Na základě těchto testů, vyhodnocení a opravení chyb by mělo dojít k nasazení aplikace do ostrého provozu.

3.2.3 Trvalý provoz, údržba a rozšiřování systému Partyboard

Nasazením aplikace do ostrého provozu samozřejmě projekt nekončí. Dalším důležitým bodem je jeho samotný provoz, údržba (nenarušit stávající funkčnost) a také jeho rozšiřování, aby obstál na trhu i z dlouhodobého hlediska. Rozšiřování Partyboardu bude primárně vycházet z požadavků a připomínek zákazníků. Dále budou také probíhat analýzy získaných dat pro zjištění důležitých informací, jako např. počet příchozích zpráv na konkrétní Partyboard, počet registrovaných uživatelů, počet příchozích zpráv ke konkrétním soutěžím atd. Tyto informace nám nabídnou pohled na službu Partyboard, díky kterému budeme moci reagovat na požadavky zákazníků a zlepšovat naši službu.

4 Výběr relačního SŘBD

V této kapitole bych rád zmínil informace o jednotlivých SŘBD (systém řízení báze dat). Pro tuto aplikaci je z počátku zbytečné používat některý ze zpoplatněných systémů jako je například SQL Developer od společnosti Oracle. Na základě toho jsem hledal nějaký vhodný open source databázový systém a jako hlavní kandidáti z mého pohledu na SŘBD připadaly v úvahu MySQL, PostgreSQL a Firebird.

4.1 Analýza jednotlivých SŘBD systémů

MySQL

Prvopočátky tohoto databázového systému sahají do roku 1995. Byl vyvinut švédskou společností MySQL AB a jeho hlavními autory jsou D. Axmark a M. Widenius [13]. Velice zajímavá je historie vzniku názvu tohoto systému - byl složen ze jména dcery („My“) jednoho ze spoluzakladatelů a zkratky SQL[1], která charakterizuje dotazovací jazyk, který tento systém využívá [14]. V lednu roku 2010 tento systém odkoupila společnost Oracle [15]. MySQL prošel od svého vzniku velkými změnami a v současné době je aktuální verze 5.7.11 [16].

V dnešní době patří MySQL mezi nejrozšířenější open source databázové systémy, který využívá dotazovací jazyk SQL a je považován za základní software webových serverů [17]. Další nezanedbatelnou výhodou tohoto SŘBD je také podpora pohledů, cizích klíčů a transakcí [18]. Tento systém je dnes nabízen ve více verzích pro různé systémy jako je např. Linux, MacOS, Windows – 32bit, Windows – 64bit atd [19].

Hlavní funkce systému:

- View – jedná se o databázový objekt, který uživateli poskytuje data ve stejné podobě jako tabulka, avšak v pohledu nejsou data přímo uložena v databázi, jako je tomu u tabulky. Obsahují předpis, jakým způsobem se mají data získat z tabulky a jiných pohledů [20].

[1] SQL Structured Query Language

- Partitioning - „umožňuje rozdělit jednu tabulku na více menších, které je pak možné uložit každou v jiném uložišti, což má za následek urychlení provádění dotazů nad danou tabulkou“ [21].
- Triggery - jedná se o objekty, které se mají provést v případě konkrétní události nad databázovou tabulkou. Například před smazáním, vložením nebo přepsáním hodnoty v tabulce dojde k vykonaní příkazu mezi BEGIN a END [21]. Jsou například vhodné pro ověřování hodnot ve sloupci před uložením dat.
- Procedura, funkce - jedná se o bloky programu, které jsou jasně oddělené od svého okolí a má rozhraní pro komunikaci s jinými moduly programu. Funkce mohou mít vlastní lokální proměnné, které jsou neviditelné pro ostatní části programu. Proceduru i funkce jsou uložené v databázi a neobsahují žádná data. Lze je vytvořit, upravit či smazat pomocí příkazů dotazovacího jazyka databáze. Rozdíl mezi procedurou a funkcí je v tom, že procedura nemá žádnou návratovou hodnotu [22].

PostgreSQL

Prvopočátky systému PostgreSQL se datují od roku 1986, vznikl na University of California at Beckeley a je spojen se jmény L. Row a M. Stonebraker [23]. Jak uvádí Jakub Král ve své práci: „V roce 1995 byl zaveden jazyk SQL namísto starého QUELu[2] a další vývoj již probíhal mimo univerzitu. O dva roky později došlo ke změně názvu na PostgreSQL. Vůbec první verze byla vypuštěna roku 1997 a nesla název PostgreSQL 6.0 [21].“ V současné době je aktuální verze tohoto systému 9.5.2 [24].

PostgreSQL, stejně jako MySQL, patří mezi nejrozšířenější open source databázové systémy. Tento systém nabízí velké uplatnění v oblasti geografických informačních systémů a to díky rozsáhlému množství datových typů, které lze využívat pro uchování geografických dat [25]. PostgreSQL v dnešní době také nabízí více verzí pro různé operační systémy jako je např. Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), Windows atd. [26].

[2]QUEL Relational database query language

Hlavní funkce systému:

- Indexy – jedná se o databázovou konstrukci, která slouží pro rychlejší vyhledávání v případě velkého množství dat nebo pro optimalizaci full-textového vyhledávání.
- Triggery – tato funkce se neliší od triggerů systému MySQL. Využívají se např. také pro ověřování hodnot ve sloupci a před uložením dat.
- Funkce – jsou stejné jako funkce u systému MySQL, ale dokáží pracovat s vlastními datovými typy a podporují vracení řádků. Výstupem funkce je pak množina hodnot, se kterou lze dále pracovat jako s tabulkou [27].
- MVCC[3] - tato metoda se stará o souběžné připojení více uživatelů ke stejným datům. Díky tomu je umožněno provádět změny, aniž by je ostatní uživatelé před potvrzením transakce viděli [28].
- Inheritance - značí dědičnost, která pochází z objektově orientovaných databází [29]. Tato vlastnost umožňuje dědění sloupců z rodičovské tabulky do tabulky potomka [30].
- ACID[4]
  - Atomičnost – v rámci transakce dojde buď k provedení všech změn, nebo nebude provedena žádná.
  - Konzistence – ať už je transakce úspěšně dokončená nebo není, data se v databázi nachází ve stavu, který splňuje všechna integritní omezení.
  - Izolace –žádná transakce není ovlivněna souběžnými transakcemi. Pokud však tyto transakce provádějí určité změny, nesmí vědět o změnách ostatních transakcí. Jednotlivé dočasné mezivýsledky musí být skryty před ostatními transakcemi.
  - Trvanlivost – tato podmínka říká, že pokud jsou všechny změny dat trvalé (do databáze zapsané), tak není možné, aby došlo jakýmkoliv způsobem ke změně. Všechny provedené změny jsou uchovány i při výpadku systému [31].

[3]MVCC Multi-Version Concurrency Control
[4]ACID Atomicity, Consistency, Isolation, Durability

Firebird

Systém Firebird je nejmladší ze všech výše zmíněných systémů. Byl odvozen od komerčního databázového systému InterBase 6.0, jehož vlastníkem byla firma Inprise (dnes Borland Software Corp.). V roce 2000 se firma Borland rozhodla zdarma zveřejnit zdrojové kódy systému InterBase 6.0 internetové komunitě vývojářů a ti se na základě toho rozhodli založit samostatný open source projekt pod názvem Firebird [25]. Tento systém prošel od svého vzniku velkými změnami a v současné době je aktuální verze 2.5.5 [32].

Tento systém není příliš známý mezi open source databázovými platformami, i přesto jde ale o relační databázi nabízející mnoho standardních funkcí ANSI SQL, která je také schopná být provozována na různých operačních systémech jako je Linux, MacOS, Windows a mnoho dalších OS založených na unixových platformách. Firebird nabízí vysoký výkon, velkou jazykovou podporu pro vytváření procedur, triggerů a také umožňuje souběžný přístup uživatelů. Zpravidla se jedná o plnohodnotný SŘBD.

Tento systém má samozřejmě mnoho stejných funkcí jako předchozí dva systémy, ale za zmínku stojí tyto dvě vlastnosti:

- MGA[5] - jedná se o klíčovou vlastnost, která umožňuje vývoj a podporu hybridního OLTP (On-line Transaction Processing) a OLAP (On-line Analytical Processing) aplikací. Díky tomu je tento databázový systém schopný sloužit současně jako analytické a zároveň operativní úložiště dat, protože čtenáři neblokují přístup zapisovatelů na stejná data [33].
- Logging and monitoring - Firebird poskytuje Trace API a bohatou sadu monitorovacích tabulek (MON $), které umožňují i monitorování v reálném čase. Dále je součástí systému SQL ladění pro optimalizaci dotazů a Audit, který má na starosti správu událostí a částečné nebo úplné protokolování prostřednictvím vzdáleného připojení [33].

4.2 Shrnutí a výběr relačního SŘBD systému

V dnešní době jsou si všechny tyto zmiňované open source databázové systémy navzájem velkou konkurencí, ale mají i mnoho společných vlastností

[5]MGA Multi-generation architecture

a funkcí jako např. pohledy, procedury, funkce, triggery, indexy, partitioning, MVCC (MGA), ACID atd. Nicméně systémy se liší podporou různých datových typů (např. formát JSON) či systémovými funkcemi, které usnadňují programátorům práci s daty. PostgreSQL a Firebird umožnují při vkládání dat do databáze rovnou vrátit uložený záznam (INSERT INTO TOWNS (NAME) VALUES („PRAHA“) RETURNING *;), PostgreSQL také nabízí novou vlastnost GROUPING SETS.

Z analýzy jednotlivých SŘBD vyplynulo, že všechny tyto systémy jsou z hlediska jejich zásadních vlastností pro tento projekt postačující. Mezi dalšími hlavními podmínkami při výběru databázového systému byla pro náš projekt jeho dostupnost na webových hostinzích ve spojení s podporou frameworku Node.js a také existence ovladače pro podporu vzájemné komunikace. Pro všechny uvedené systémy existují ovladače pro komunikaci se zmiňovaným frameworkem, avšak jediný systém MySQL se zmiňuje o tomto spojení na svých oficiálních stránkách [34]. Obecně je dnes problém získat na českém trhu webový hosting s podporou pro Nodejs a také s databázovým systémem Firebird. Největší podporu na trhu má systém MySQL, přičemž v poslední době se k němu přibližuje PostgreSQL. Při hledání hostingu, který by podporoval framework Nodejs v kombinaci s jedním databázovým systémem jsem nalezl pouze jeden hosting s názvem Roští, který podporuje Node.js a PostgreSQL [35].

V neposlední řadě o výběru SŘBD rozhodovala jeho uživatelská přívětivost při práci s ním. Systém Firebird prakticky nabízí po instalaci ovládání pouze z příkazové řádky. Tento způsob ovládání může být pro některé uživatele plně postačující, avšak pro mě to nepředstavovalo optimální variantu. Narozdíl od tohoto systému, PostgreSQL a MySQL disponují po instalaci uživatelským rozhraním pdAdmin III (PostgreSQL) a webovým rozhraním phpMyAdmin (MySQL).

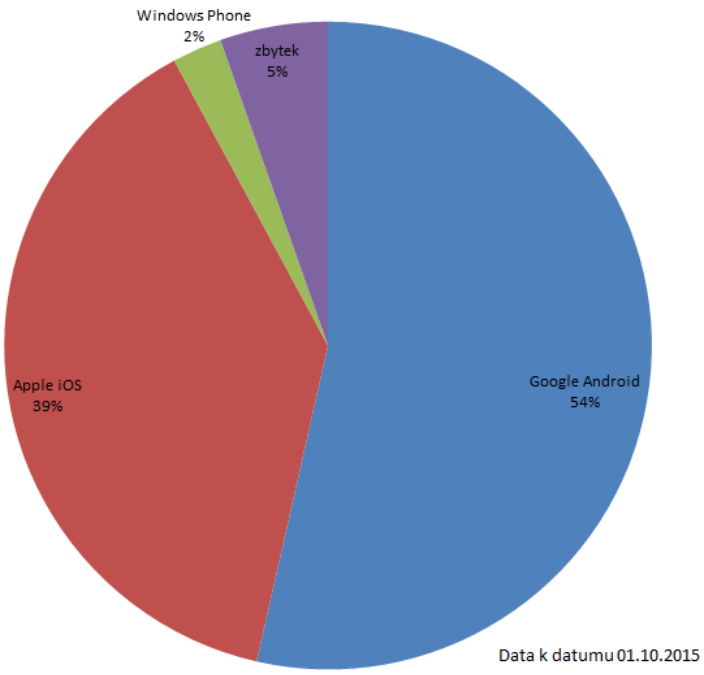
Na základě těchto zjištění jsem se rozhodl pro databázový systém PostgreSQL. Lze pro něj zajistit webový hosting na českém trhu ve spojení s frameworkem Nodejs, existují ovladače pro jejich vzájemnou komunikaci, z mého pohledu je uživatelsky přívětivý, nabízí více datových typů, systémových funkcí a „vychytávek“ než systém MySQL a také s ním již mám zkušenosti při realizaci bakalářské práce, při které jsem jej využíval.

5 Mobilní aplikace

V této kapitole jsou popsány základní informace o jednotlivých mobilních platformách, možnosti vývoje mobilních aplikací a také jejich základní komunikační mechanismy pro hybridní přístup vývoje aplikací.

5.1 Mobilní platformy

Na úvod této kapitoly jsem shrnul podíl jednotlivých mobilních platforem na trhu a popsal jejich základní informace. Mezi nejrozšířenější mobilní platformy v dnešní době patří Google Android, Apple iOS a Windows Phone.



Obrázek 5.1: Podíl mobilních platforem na trhu

Z grafu lze vyčíst, že největší podíl na trhu má platforma Android i přesto, že ke dni 25. 9. 2015 byl uveden na trh iPhone 6S a iPhone 6S Plus. Tuto studii provedla společnost NetMarketShare, která sbírala data celé září roku 2015 [36].

5.1.1 Google Android

V aktuální době je Android nejsilnějším otevřeným (open source) operačním systémem na trhu, který je licencovaný pod licencí Apache2.0/MIT a je vyvíjen společností Google. Díky ní se mohou aplikace psané pro tento systém volně rozšiřovat a systém může být použit na kterémkoliv hardwaru [37].

Architektura tohoto systému je rozdělena do několika vrstev. Nejnižší vrstvou je samotné jádro systému, které je postaveno na jádře Linuxu a tvoří abstraktní vrstvu mezi hardwarem zařízení a zbytkem softwaru ve vyšších vrstvách.

Vývoj pro tuto platformu je možný jak na systému Microsoft Windows, OS X, tak i na Linuxu. Aplikace se programují v programovacím jazyce Java s rozšířením o Android SDK[1], který je samozřejmě dostupný pro všechny platformy systémů. Toto rozšíření je primárně rozděleno do 3 sad:

- základní – sada nezbytná pro vývoj aplikací;
- doporučená – sada pro možnost nahrát aplikace do zařízení, různé ukázky kódů a dokumentace;
- plná – sada, která umožňuje přístup ke Google API a obsahuje knihovnu pro ověření licence aplikace, zda se nejedná o nelegální kopii.

Všechny výše zmíněné sady obsahují i emulátor pro testování aplikací.

Aplikace je možné distribuovat přes různé obchody, které jsou dostupné v zařízeních (telefony, tablety atd.), ale je zde také možnost vlastního stahování.

Nejrozšířenějším obchodem je Google Play. Jedná se o digitální obchod, ve kterém lze zakoupit nejen aplikace, ale také elektronické knihy, filmy, hudbu atd. Společnosti Samsung a Amazon mají taktéž svůj obchod.

Od vydání první verze v dubnu 2009 bylo již několik aktualizací, které přidávají nové funkce a opravují chyby. Poslední verze byla vydána v říjnu roku 2015 s názvem Android 6.0 Marshmallow. Jednotlivé verze jsou vždy pojmenovány podle sladkostí (Cupcake, Donut, Eclair, Froyo, Gingerbread,

[1] SDK Software Development Kit

Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop a nejnovější Marshmallow) [38].

5.1.2 Apple iOS

Tento operační systém je vyvíjen společností Apple a není licencován ostatním distributorům zařízení - to znamená, že se nachází pouze na zařízeních této značky [39].

Také architektura tohoto systému je rozdělena do vrstev. Jedná se o samotné jádro OS, vrstvu služeb pro podporu jádra, vrstvu podpory médií a vrstvu Cocoa Touch. iOS je „odlehčená“ verze operačního systému OS X, který je využíván v počítačích od společnosti Apple a jedná se tedy také o systém UNIXového typu. Jelikož je tento systém určen pro mobilní zařízení, neobsahuje veškerou funkcionalitu jako OS X, ale byla přidána podpora dotykového ovládání.

K vývoji aplikací pro tuto platformu je nutné mít OS X, který je potřebný pro vývoj aplikací na některém druhu z počítačů od firmy Apple. Aplikace se programují v programovacím jazyce Objective-C (nadstavba jazyka C) a v novém jazyce Swift. Dále je nutné mít nainstalovaný iOS Software Development Kit (iOS SDK). Ideální je využít IDE[2] Xcode od společnosti Apple, který v sobě obsahuje již vše potřebné pro vývoj aplikací a je zcela zdarma.

Aplikace je možné distribuovat pouze přes Apple App Store, který má navíc velmi specifická pravidla pro zveřejnění aplikace. V tomto digitálním obchodě je možné sehnat nejenom aplikace, ale také filmy, hudbu atd. [40].

První verze operačního systému iPhone OS 1.x je datována k březnu roku 2008 a od té doby již bylo vydáno mnoho dalších verzí. Nejnovější verze se datuje k březnu roku 2016 s označením iOS 9.3 [39].

5.1.3 Windows Phone

Tento operační systém je vyvíjen společností Microsoft a je vydáván s Proprietární licencí, typicky Microsoft EULA. Jedná se o software s uzavřeným

[2]IDE Integrated Development Environment

kódem (closed source), ve kterém nelze zpravidla dělat vlastní úpravy a je v něm definováno, co uživatel smí a nesmí.

Mezi Windows Phone 7 a Windows Phone 8 došlo ke změně jádra. Z monolitického jádra založeného na Windows CE se přešlo na hybridní jádro založené na Windows NT, což ale má za následek vzájemnou nekompatibilitu.

Pro vývoj aplikací na tuto platformu je nutné mít operační systém Windows. Aplikace jsou napsány v programovacím jazyce C# a dále je nutné rozšíření Windows Phone SDK. Ideálním nástrojem pro vývoj je IDE Microsoft Visual Studio, které ale není zdarma jako v případě IDE Xcode od společnosti Apple [41].

Platforma Windows Phone má také svůj obchod s názvem Store, který je velmi podobný již zmíněným obchodům pro platformu Android a iOS.

Tato platforma je velmi mladá a její počátky se datují k říjnu roku 2010 s názvem Windows Phone 7. Poslední oficiální vydanou verzí je systém Windows Phone 8 a v dnešní době dochází k testování Windows 10, který má být „univerzálním“ systémem jak pro mobilní zařízení, tak pro počítače.

5.2 Možnosti vývoje pro mobilní platformy

V předchozí kapitole jsem shrnul hlavní platformy pro mobilní zařízení, avšak teď nastává otázka, jaký zvolit přístup pro samotný vývoj aplikace. Máme základní 3 typy přístupů pro vývoj aplikací a to nativní, webový nebo hybridní. Všechny tyto přístupy mají své výhody i nevýhody, které se pokusím vysvětlit a shrnout v jednotlivých podkapitolách.

5.2.1 Nativní přístup

Jedná se o přístup, kdy je aplikace vždy implementována přímo pro konkrétní platformu. To znamená, že pokud je aplikace implementována pro platformu Android, tím pádem není přenositelná na jiné platformy, což se dá pokládat za její velkou nevýhodu. Pokud má být tato aplikace dostupná pro všechny platformy, programátor musí vyvíjet aplikaci pro každou platformu zvlášť, což je velice časově i finančně náročné a programátor musí umět konkrétní

programovací jazyk jednotlivých platforem.

Tyto aplikace nevyžadují připojení k internetu a mají „vždy“ lepší použitelnost, funkčnost a jistotu graficky stejného prostředí a umožňují plně využívat hardware mobilního zařízení jako např. kameru, akcelometr aj.

Další výhodou je, že při využití doporučených IDE vývojářských nástrojů (např. Android Studio) je možnost využít jejich ladící nástroje, at' už se jedná o debugging, testování, či jejich emulátor zařízení. Také díky tomuto přístupu je pohodlnější distribuce aplikace do konkrétního obchodu [42].

5.2.2 Webový přístup

Tento přístup je postaven na principu webových stránek, které jsou známy z klasických stolních počítačů či notebooků. Jedná se o stránky, které jsou nastylovány pomocí CSS[3] jazyka pro různá rozlišení displejů. To znamená, že pokud je stránka načtena v zařízení s malým rozlišením, vypadá jinak, než když je načtena v rozlišení velkém. Dochází zde k takové úpravě zobrazení, aby byl obsah čitelný jak na malém, tak i velkém rozlišení.

Hlavní výhoda tohoto přístupu je v tom, že tyto „webové aplikace“ nebo také „HTML5 mobilní aplikace“ jsou snadno dostupné na všech platformách a jejich aktualizace je velmi snadná oproti nativnímu vývoji, při kterém by bylo nutné upravit aplikaci pro všechny platformy. V tomto případě stačí upravit pouze jednu webovou stránku a tím pádem se uživatelé těchto „webových aplikací“ nemusí o nic starat. Další výhodou je, že vývojáři mohou naimplementovat pouze jedny webové stránky se styly pro různá rozlišení, což umožní zajistit dostupnost „webové aplikace“ pro všechny platformy.

Hlavní nevýhodou tohoto přístupu je závislost na webovém prohlížeči a internetu, bez kterého se systém nenačte. Většinou jsou tyto aplikace nekonzistentní z hlediska uživatelského rozhraní a nepodporují aplikační rozhraní jednotlivých zařízení, jako je např. práce s kamerou, akcelerometrem atd.

Otázkou však zůstává, zda je výhodou či nevýhodou, že se tyto aplikace neinstalují a tím pádem nejsou umístěny v jednotlivých obchodech pro každou platformu [43].

[3] CSS Cascading Style Sheets

5.2.3 Hybridní přístup

Tento přístup je spojením nativního a webového přístupu. Aplikace se implementuje pomocí webových technologií (HTML5[4], CSS3 a Javascript) a je plně funkční pro všechny mobilní platformy. Zároveň se ale chová jako nativní mobilní aplikace a to díky tomu, že dokáže pracovat s nativním API jednotlivých typů zařízení. Prakticky se jedná o webovou aplikaci, která je obalena nativní „slupkou“, přes kterou lze přistupovat k nativnímu hardwaru zařízení [42].

Jak již bylo zmíněno, hlavní výhodou tohoto přístupu je možnost naimplementování a otestování pouze jedné aplikace vývojářem, která je pak dostupná pro všechny platformy, je schopna využívat nativního API a lze ji umístit do jednotlivých digitálních obchodů. I přesto, že se jedná o webovou aplikaci, není nutné připojení k internetu.

Nevýhodou tohoto přístupu oproti nativnímu přístupu je odezva aplikace (rychlost), proto není úplně vhodný např. pro hry či aplikace, které jsou závislé na rychlé odezvě.

V dnešní době je však tento typ přístupu velmi populární a je vyvíjeno mnoho frameworků jak pro uživatelské rozhraní (např. Ionic), tak také pro komunikaci s nativním aplikačním rozhraním (např. Cordova) [43].

[4]HTML Hypertext Transfer Protocol

5.2.4 Shrnutí

Tato tabulka ukazuje moje vlastní srovnání jednotlivých přístupů, které byly popsány v předchozích kapitolách.

Přístupy	**Nativní**	**Webový**	**Hybridní**
Kvalita uživatelského rozhraní	10/10	5/10	9/10
Kvalita aplikací	10/10	5/10	8/10
Odezva aplikací	10/10	5/10	7/10
Programátorské dovednosti	Objective-C/Swif, Java, C#	HTML, CSS, JavaScript	HTML, CSS, JavaScript
Aktualizace aplikace	Nová verze v digitálním obchodě	Stačí aktualizovat aplikaci na serveru	Nová verze v digitálním obchodě
Cena vývoje aplikace	10/10	3/10	6/10
Závislost na internetu	NE	ANO	NE
Nativní API	ANO	NE	ANO

Tabulka 5.1: Tabulka srovnání přístupů vývoje mobilní aplikace

Každý ze zmíněných přístupů má své výhody a nevýhody, avšak za nejslabší bych považoval Webový přístup, protože mobilní rozvoj je neustále v pohybu a přibližně každých šest měsíců je nová verze mobilního operačního systému s unikátními vlastnostmi, které jsou přístupné pouze s nativními API a „webové aplikace“ tak dělají pokroky pouze jednou za několik let. Tento přístup ale může být vhodný a velmi rychlý pro jednoduché aplikace, které zobrazují pouze textový obsah [43].

Co se týče zbylých dvou přístupů, je hodnocení velice vyrovnané. Zkrátka jde o to si odpovědět na otázky, zda chci vyvíjet aplikace na všechny platformy, zda umím nebo mám čas se naučit každý programovací jazyk pro každou platformu a jestli kladu 100% důraz na uživatelskou platformu.

Z mého pohledu se jeví jako nejlepší hybridní přístup, jelikož dnešní frameworky umí velice dobře pracovat v nativním API mobilních zařízení a také mají velmi podobné, skoro až k nerozeznání uživatelské rozhraní od nativních aplikací a stačí naimplementovat a otestovat pouze jednu aplikaci. Na základě

toho jsem se rozhodl tento přístup použít pro implementaci mobilních aplikací pro službu Partyboard.

5.3 Mechanismy komunikace mobilních aplikací

Mezi základní mechanismy komunikace mobilních aplikací se serverem patří jak trvalé, tak dočasné spojení klient – server.

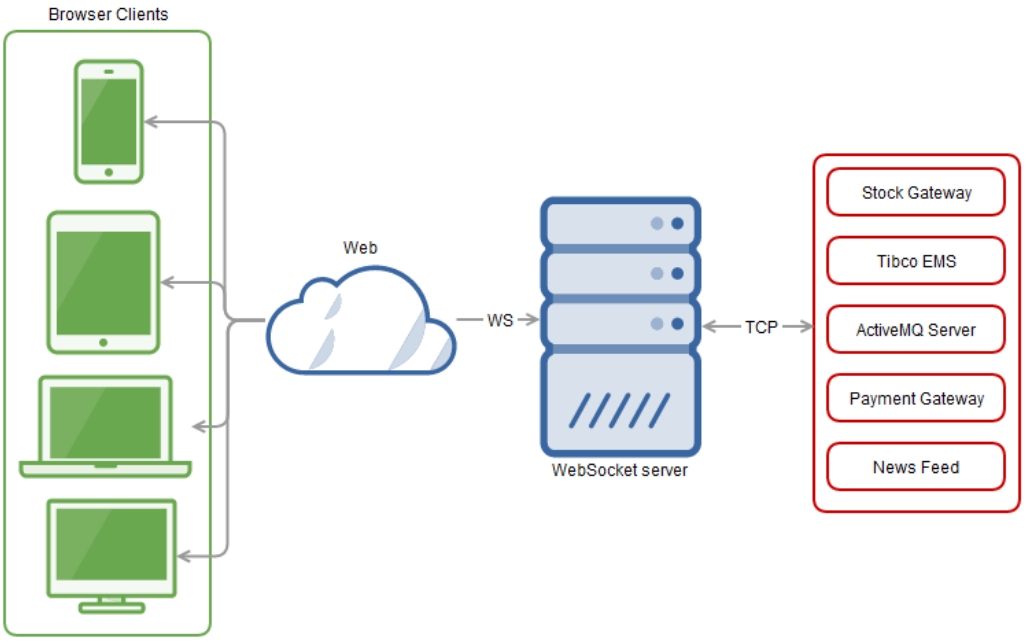
Při výběru komunikačního mechanismu mezi serverem a mobilní aplikací (klientem) je důležité si nejprve rozmyslet, o jakou výměnu informací půjde - zda je potřeba udržovat trvalé spojení mezi aplikací a serverem nebo se stačí dotazovat na server pro nové informace (data) v určitém intervalu a při určité události. Dalším faktorem při rozhodování by měl být samotný přístup vývoje aplikace, zda se jedná o nativní, webový či hybridní přístup.

Vzhledem k tomu, že v mém případě se jedná o hybridní přístup vývoje aplikace (prakticky webová aplikace s nativním obalem), tak bych dále rád popsal základní možnosti tohoto přístupu pro trvalé i dočasné spojení mobilní aplikace se serverem.

5.3.1 Web Sockets

Web Sockets je definován jako RFC 6455 a vychází ze starších technologií, jako je AJAX[5] či Comet (dlouhodobě vyhrazené HTTP[6] spojení) a nabízí programátorům to, co tyto technologie nedokázaly - jednoduché rozhraní pro navázání spojení a vzájemnou výměnu zpráv mezi klientem a serverem [44].

[5] AJAX Asynchronous JavaScript and XML
[6] HTTP Hypertext Transfer Protocol



Obrázek 5.2: Struktura spojení klient - server

V tomto případě se tedy jedná o trvalé spojení mezi klientem a serverem, kde „socket“ znamená propojení serveru s klientem prostřednictvím dvojice IP[7] adresy a portu. Díky tomu si mohou oba koncové body v reálném čase zasílat data simultánně přes jeden socket. Na bázi aplikační vrstvy je však nezbytné udělat změnu a místo tradičního protokolu HTTP použít protokol WS[8], pro zabezpečené spojení pak WSS[9] místo HTTPS[10] [45].

Na straně klienta jsou WebSockets implementovány v JavaScriptu jako třída WebSocket s tím, že je potřeba více než běžný HTTP server, který podporuje také technologii WebSockets, jako např. lokální WebSockets node [44].

U WebSocketu nejsou vyměňované zprávy žádné sofistikované objekty, jde pouze o prosté řetězce.

[7]IP Internet Protocol
[8]WS WebSocket
[9]WSS WebSocket with SSL
[10]HTTPS Hypertext Transfer Protocol over SSL

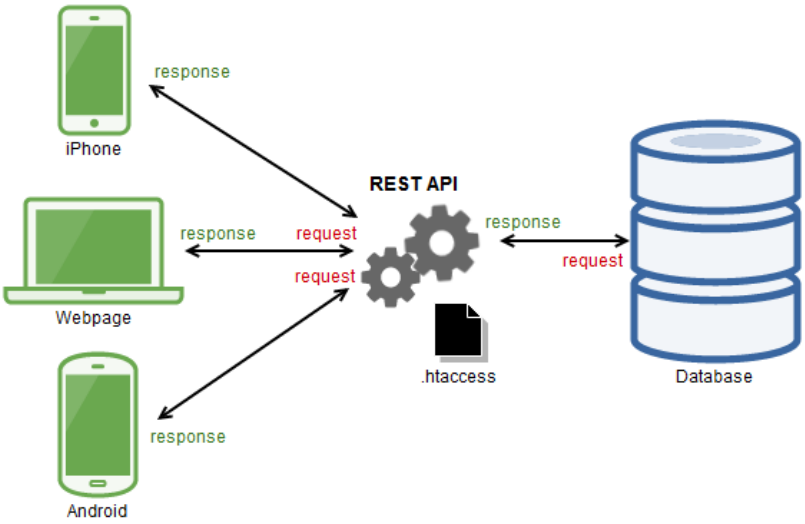
Třída WebSocket nabízí tyto funkce:

- onopen – slouží pro otevření spojení mezi klientem a serverem;
- onmessage – slouží pro výměnu dat mezi klientem a serverem, je zavolána ve chvíli, kdy ze serveru přijde zpráva;
- onclose – slouží pro oznámení uzavření spojení mezi klientem a serverem.

Pro posílání zpráv slouží metoda SEND, jejímž parametrem je řetězec, který má být poslán [44].

5.3.2 REST API

REST[11] je architektura určená pro komunikaci v distribuovaném prostředí, která umožňuje CRUD[12] operace pomocí standardních HTTP dotazů [46]. Klient zašle požadavek na server, ten ho zpracuje a zašle odpověď zpět přímo klientovi.



Obrázek 5.3: Struktura komunikace REST API

[11] REST Representational State Transfer
[12] CRUD Create, Read/Retrieve, Update, Delete

Toto rozhraní je orientováno datově, nikoli procedurálně jako protokol XML-RPC a SOAP[13]. Webové služby definují protokol pro volání vzdálených procedur a REST určuje, jak se má přistupovat k datům.

REST nabývá v dnešní době na významu a stává se spolu s JSON[14] standardem pro API webových služeb. K jeho rozšíření napomáhá jednak technika AJAX, které REST vychází vstříc, tak i to, že se nijak zásadně neliší od standardního volání a získávání dat pomocí HTTP, pouze jej zobecňuje.

Moderní frameworky pro vývoj „server-side“ aplikací pomáhají vytváření REST rozhraní tím, že dokáží nadefinovat patřičné procedury pro všechny potřebné metody, takže vytvoření vlastního REST API je opravdu snadné. REST svou bezestavovostí vychází vstříc moderním metodám vývoje webových aplikací, které jsou založené na paralelním zpracování distribuovaného obsahu [46].

REST API je použitelné pro jednotný a snadný přístup ke zdrojům. Zdrojem mohou být jak data, tak také stavy aplikace za předpokladu, že je lze popsat konkrétními daty. Všechny zdroje mají vlastní identifikátor URI[15] a REST definuje čtyři základní metody pro přístup k nim, které jsou označeny zkratkou CRUD. Tyto metody jsou implementovány pomocí odpovídajících metod HTTP protokolu [47].

Metoda GET (Retrieve)

Jedná se o základní metodu, jejímž účelem je získat požadovaný zdroj ze serveru. Tato metoda by neměla mít žádný postranní efekt jako je mazání či tvorba zdrojů. Každý uživatel webu se s ní setkává dnes a denně, jedná se o klasický požadavek na stránku. Například tento požadavek by vrátil informace o uživatelích na serveru:

[GET /USERS/{ID_USER}]

Parametry dotazu se využívají pro filtraci dat:

[GET /USERS/{?LIMIT,OFSET,DELETE}]
Host: www.partyboard.com

[13]SOAP Simple Object Access Protocol
[14]JSON JavaScript Object Nonation
[15]URI Uniform Resource Identifier

Metoda POST (Create)

Tato metoda slouží pro odesílání dat na server ke zpracování a tato data jsou vložena do těla požadavku. U této metody není ve chvíli volání známý přesný identifikátor zdroje, jelikož zdroj ještě neexistuje a proto se pro POST používá domluvený společný identifikátor („endpoint“). Tato metoda se může použít jak pro úpravu zdrojů, tak také pro jejich tvorbu. Po odeslání požadavku a jeho zpracování by měl server vrátit patřičný návratový kód, který vyrozumí klienta, zda došlo ke správnému zpracování požadavku nebo k chybě.

[POST /USERS]
Host: www.partyboard.com

Skupina kódů	**Popis**
1xx	Informační kódy.
2xx	Kódy označující úspěšné vykonání požadavku.
3xx	Přesměrování - označuje odpověď, která obsahuje adresu, na které se nachází požadovaný zdroj.
4xx	Chybové kódy – označují problém při zpracování požadavku způsobený klientem, např. odesílání nesprávných dat.
5xx	Chybové kódy – označují problém při zpracování požadavku, který vznikl na straně serveru.

Tabulka 5.2: Tabulka se skupinami HTTP kódů

Metoda DELETE

Metoda DELETE je určena pro smazání zdroje. Volání je obdobné jako u metody GET:

[DELETE /USERS/{ID_USER}]
Host: www.partyboard.com

V praxi bývá někdy problematické vyvolat HTTP metodu DELETE – mnoho nástrojů určených k volání HTTP požadavků či HTML formuláře jsou omezeny pouze na metody POST a GET. V praxi se proto u REST rozhraní používají náhradní způsoby, např. volání pomocí metody POST s parametrem, který sděluje, že má být ve skutečnosti použita metoda DELETE nebo speciální URI, což se ale neslučuje s definicí REST.

Metoda PUT (Update)

Tato metoda (operace změny) je velice podobná metodě POST (operaci vytvoření), jen s tím rozdílem, že je volána konkrétní URI na konkrétní zdroj, který chceme změnit a v těle požadavku jsou předány nové hodnoty. Na rozdíl od metody POST je u úpravy zdroje URI vždy známá už na počátku, takže ji lze zadat. U metody PUT platí totéž co u metody DELETE. Ne každý nástroj ji podporuje a proto některá REST API používají různé náhradní metody, jak již bylo zmíněno výše.

[PUT /USERS/{ID_USER}]
Host: www.partyboard.com

6 Analýza a implementace mobilní aplikace

Na základě získaných informací z kapitoly **Možnosti vývoje pro mobilní aplikace** jsem se rozhodl pro hybridní přístup vývoje mobilní části systému Partyboard, protože v mém případě se jedná o aplikaci, která primárně slouží pro zobrazování a odesílání dat. Dalším důvodem výběru tohoto přístupu je jeho síla v implementaci a testování jedné aplikace, kterou lze pak sestavit a vydat pro různé mobilní platformy.

6.1 Architektura aplikace

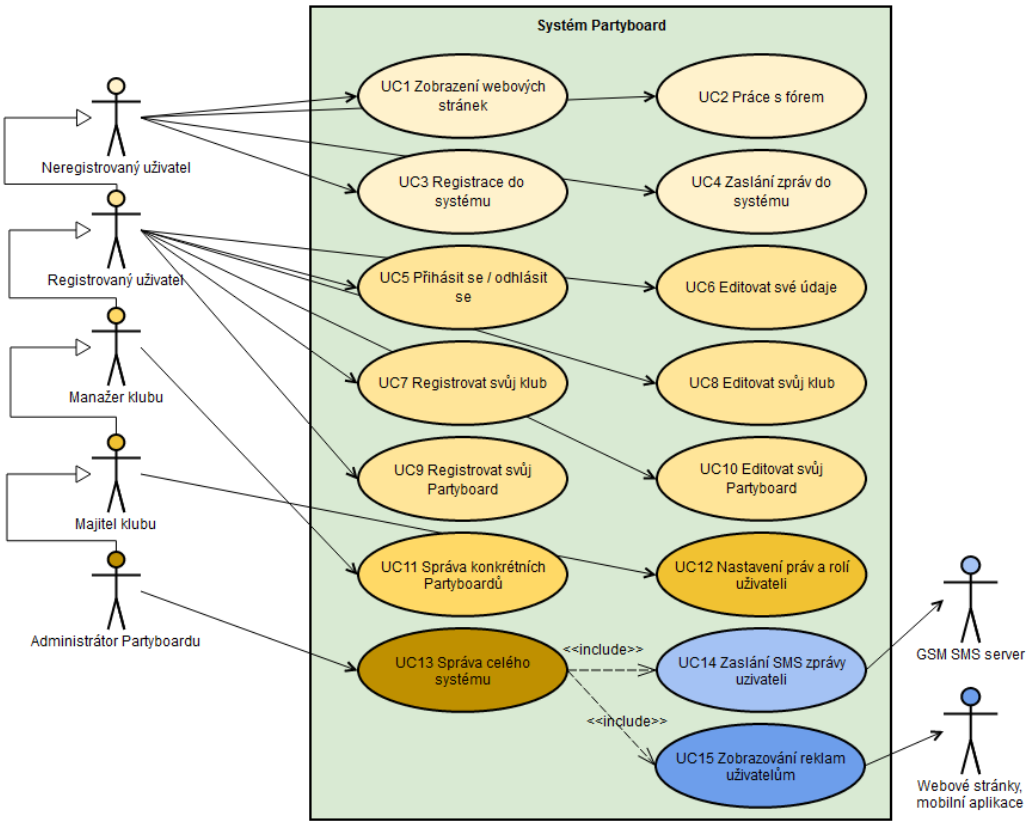
Pro popis architektury jsem použil systém 4+1 pohledů, který se skládá z fyzického, procesního, logického, vývojového pohledu a z popisu funkčnosti systému. Převážně jsem popisoval svou část projektu, avšak v některých pohledech popisuji celý projekt, aby si čtenář mohl dát vše snáze do souvislostí.

Přehled požadavků na systém

Pro popis funkčnosti jsem použil diagram případů užití neboli „User Case Diagram“. Tento diagram popisuje systém s ohledem na funkčnost, jak jej vidí sám uživatel. Jedná se o základní UML[1] diagram, který zachycuje jednotlivé typy uživatelů, kteří se systémem mohou pracovat a jaké činnosti mohou v rámci systému vykonávat. Popisuje vztah mezi systémem a uživatelem a na základě toho nám pomáhá zachytit funkční požadavky na systém [48].

Následující obrázek znázorňuje diagram případu užití mezi uživateli služby Partyboard a webovým serverem.

[1] UML Unified Modeling Language



Obrázek 6.1: Diagram případu užití

Vysvětlení vybraných činností:

- UC4 – jedná se o SMS a internetové zprávy;
- UC11 – jedná se o vytvoření nastavení pro konkrétní Partyboard (barva pozadí a textu, nastavení loga, soutěží, zablokování (ban) uživatele pro konkrétní Partyboard atd.);
- UC12 – majitel klubu má právo nastavit konkrétnímu uživateli příslušná práva pro správu konkrétního Partyboardu (UC11);
- UC13 – jedná se o možnost nastavení číselníku (předdefinovaných hodnot), správu všech Partyboardů, uživatelů, soutěží, nevhodných slov atd.;
- UC14 – systém umožňuje zaslání SMS zpráv uživatelům, např. výherní kódy;

- UC15 – systém umožňuje zobrazování reklam uživatelům.

Systém Partyboard musí umožňovat veškerou funkcionalitu vyplývající z diagramu a z popisu vybraných činností. Tato funkčnost musí být zajištěna za předpokladu splnění mimofunkčních požadavků na základě odhadu počtu klubů využívajících službu Partyboard.

- Předpokládaný počet klubů: 30;
- Průměrný počet zobrazovacích zařízení v klubu: 2;
- Průměrný počet návštěvníků v klubu: 200;
- Průměrné načtení zpráv na webových stránkách: 1x/sekundu;
- Průměrný počet načítání zpráv v mobilní aplikaci: 0,2x/sekundu.

Na základě těchto odhadů by měl být průměrný počet dotazů na server a do databáze 1 260 za sekundu. Systém si klade za cíl zajistit plnou funkčnost při **1 900** dotazech za sekundu, tedy přibližně 1,5x více než je odhadované množství dotazů. Vůči této hodnotě bude výsledný systém testován.

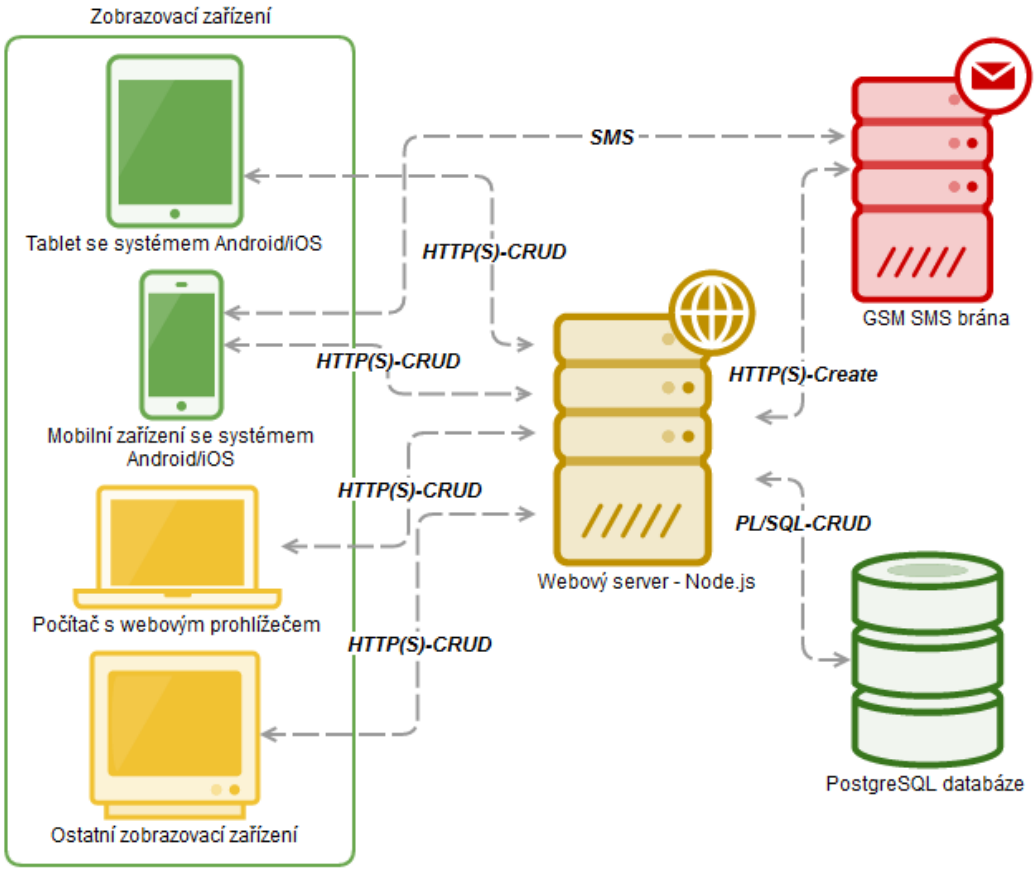
Přihlášení do systému a veškerá komunikace s webovým serverem musí být zajištěna pomocí implementace JSON Web Token (viz podkapitola **Použité technologie**) a veškerá hesla musí být zašifrována.

Fyzický pohled (pohled nasazení)

Tento pohled se zabývá navázáním systému na topologii hardwarových a dalších softwarových komponent (jakým způsobem jsou spouštěče a ostatní běhové komponenty mapovány do základních platforem), fyzické rozložení komponent, nasazení, instalace a ladění výkonu [49].

V tomto pohledu jsou zobrazeny veškeré komponenty celého projektu, avšak já jsem pracoval pouze na databázovém serveru a na mobilní aplikaci. Webovým serverem a webovými stránkami pro zobrazování dat (zpráv) se zabývá ve své diplomové práci již dříve zmíněný Antonín Neumann.

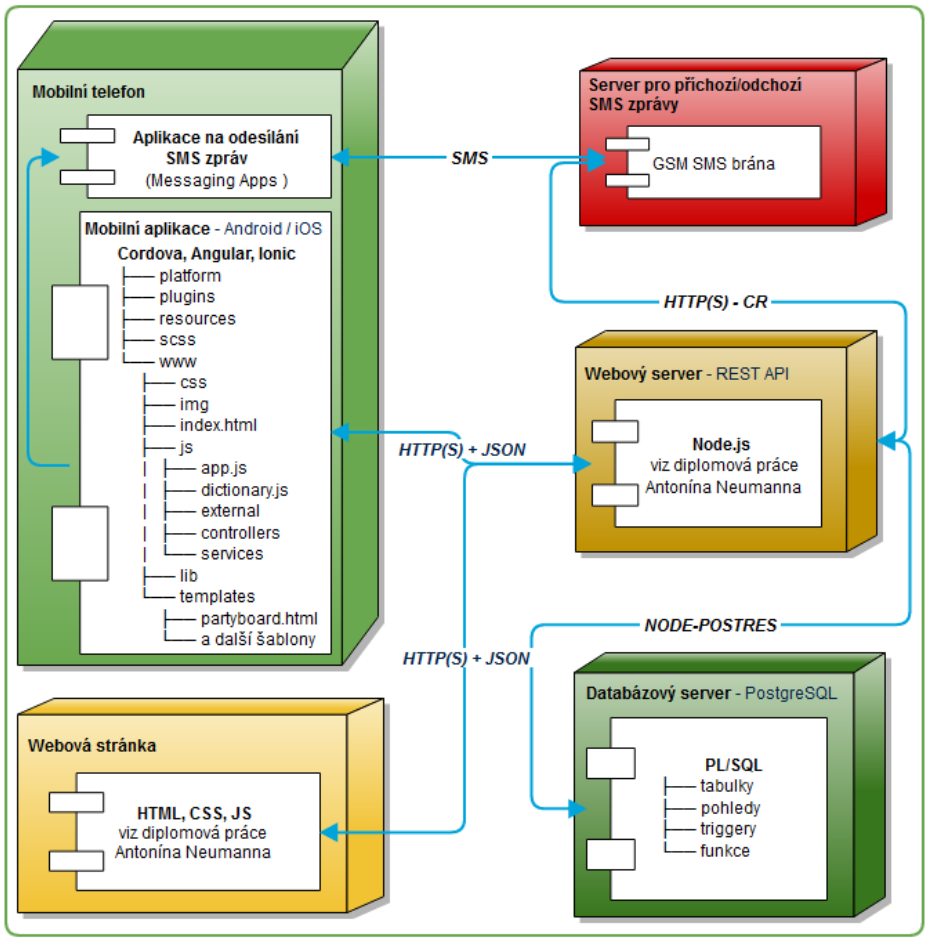
Viz následující diagram:



Obrázek 6.2: Fyzický pohled

Mobilní aplikace fyzicky běží na mobilních telefonech či tabletech a vzdáleně komunikuje s webovým serverem přes HTTP protokol, který umožňuje REST (GET, POST, PUT, DELETE). Komunikace mezi databázovým serverem a webovým serverem je zajištěna pomocí modulu NODE-POSTGRES [50]. Dále je však také možnost zasílat SMS zprávy, které jsou přijímány GSM SMS bránou, která také komunikuje s webovým serverem přes HTTP protokol. Tato brána je zajištěna externí firmou.

Rozdělení komponent systému Partyboard:



Obrázek 6.3: Komponenty systému Partyboard

Procesní pohled

Tento pohled se zabývá chováním systému, paralelismem, propustností, automatizovanými úlohami, tolerancí chyb, zotavením se z běhových chyb, odezvou systému na vnější podněty, rozšiřitelností systému, výkonností a přípustností [49].

Mobilní aplikace je podporována všemi moderními smartphony či tablety s operačním systémem iOS nebo Android. Jedná se o jednovláknovou aplikaci vytvořenou pomocí frameworku AngularJS, která komunikuje s webovým serverem pomocí HTTP protokolu. Jde o asynchronní komunikaci za

pomoci služby „$HTTP“, kterou poskytuje AngularJS pro čtení či zasílání dat na vzdálený server. Služba „$HTTP“ AngularJS vytvoří a zašle požadavek na server, který požadavek zpracuje a zašle odpověď zpět odesílateli. K výměně dat je použit datový formát JSON (JavaScript Object Notation), který je popsán v podkapitole Použité technologie.

Webový server s mnou navrženou PostgreSQL databází komunikuje asynchronně za pomoci knihovny NODE-POSTGRES. Více o tom v diplomové práci A. Neumanna v kapitole **Návrh a implementace systému**.

Aplikace je škálovatelná jak z administrativního, geografického, zátěžového, tak i z funkčního pohledu.

- Administrativní škálovatelnost – schopnost sdílet distribuovaný systém pro větší počet uživatelů. Tato schopnost je zajištěna pomocí digitálních obchodů (App Store, Google Play).
- Geografická škálovatelnost – schopnost udržovat výkon a efektivitu při fyzickému rozšíření systému do větší oblasti. Při velkém geografickém rozšíření by mělo dojít k navýšení počtu uživatelů aplikace. Vzhledem k tomu, že mobilní aplikace běží na zařízeních uživatelů, tak v tomto směru problém není. Avšak díky tomu dojde k většímu vytížení jak webového serveru (viz diplomová práce A. Neumanna v kapitole **Testování a pilotní nasazení**), tak také SŘBD.
- Zátěžová škálovatelnost – schopnost efektivně využít prostředky při větším zatížení (např. v již zmíněném přenosu dat). Při velkém vytížení databázového serveru z důvodu velkého množství příchozích požadavků by došlo k „partitioningu“, tedy k fyzickému rozdělení dat na více serverů.
- Funkční škálovatelnost – schopnost vylepšit systém přidáváním nových funkcí s minimální režií. Databázový model, webový server a mobilní aplikace jsou navrženy tak, aby se daly snadno rozšířit o novou funkcionalitu. Z pohledu datového modelu je např. navržená tabulka COMPETITION_PARTYBOARDS uchovávající si strukturu a pravidla pro možnosti nadefinování vlastních soutěží z pohledu nájemce služby Partyboard.

Logický pohled

Tento pohled se zabývá logickou strukturou systému z hlediska výsledné funkčnosti (co by systém měl vykonávat), identifikuje hlavní (věcné) balíky, subsystémy, třídy a vazby mezi nimi a model perzistentních informací (data a funkce) [49].

Pro vývoj celého projektu jsme zvolili architekturu SOA (Service-oriented architecture). Jedná se o obecný architektonický vzor založený na spolupráci navzájem nezávislých služeb. Služba je určitá část funkčnosti aplikace, která je zpřístupněna pomocí definovaného rozhraní. V našem případě se jedná o REST API. Termín REST API již byl vysvětlen již dříve v podkapitole **REST API**.

Základní principy architektury SOA [51]:

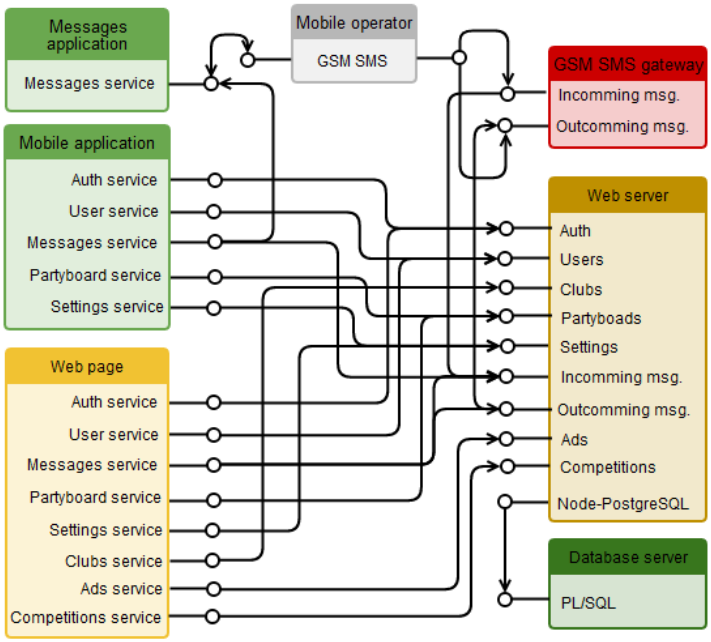
- Standardizovaný kontrakt služby – musí být přesně definované technické rozhraní a formát dat, který bude služba přijímat či odesílat.
- Slabé vazby mezi komponentami - vazby mezi jednotlivými komponentami mají být co nejtenčí.
- Princip abstrakce - snaha co nejvíce skrýt implementační detaily služby či komponenty. Při správném využití tohoto principu dojde ke zlepšení granularity systému a také velmi usnadňuje správu, popřípadě pozdější úpravy systému.
- Znovupoužití - při návrhu komponenty či služby by se mělo počítat s jejím využitím i jinde, než pouze pro jeden konkrétní projekt. Na základě toho by se měla maximalizovat použitelnost vyvíjené části.
- Nezávislost – aby služba mohla dodržet definovaný kontrakt (zajistit správné fungování), musí v sobě obsahovat vnitřní nezávislou logiku pro správu zdrojů, ze kterých čerpá.
- Bezstavovost - v tomto případě bezstavovost přímo umožňuje znovupoužití, protože komponenty, které si nepamatují stav, lze bez jakýchkoliv inicializačních požadavků okamžitě využít jinde. Tento princip zajišťuje také větší možnosti škálovatelnosti systému v budoucnu.
- Princip identifikovatelnosti – pokud je služba lehce identifikovatelná (lze lehce zjistit způsob jejího použití), má velkou výhodu na trhu

např. oproti mnohem kvalitnější službě, ale málo využívané díky její nepřístupnosti široké veřejnosti. Identifikovatelnost v SOA je zajištěna WSDL jazykem.

- Princip skládání – jde o rozdělení celé aplikace na malé podproblémy („kostičky“), které lze pak zpravidla snadněji řešit. Jednotlivá řešení jsou potom seskládána do výsledného systému tak, aby dokázala zvládnout komplexní problém.

Systém Partyboard je strukturován do 4 hlavních oblastí, které poskytují služby a zdroje:

- Správa uživatelů – autorizace, registrace, editace;
- Správa klubů – registrace, editace;
- Správa Partyboardů – registrace, editace, nastavení, soutěže, reklamy;
- Správa přijatých/odeslaných zpráv – příjem/ odesílání SMS a internetových zpráv.



Obrázek 6.4: Logický pohled

Vývojový pohled

Jedná se o pohled, který popisuje organizaci statických softwarových komponent, moduly ve vývojovém prostředí, zdrojové kódy a datové soubory. Je také zaměřen na rozdělení systému do menších, samostatně realizovatelných komponent [49].

Celý vývoj systému je rozdělen do 4 základních částí: návrh a realizace struktury databáze, návrh REST API a implementace serveru, návrh a implementace mobilní aplikace a webových stránek pro systém Partyboard.

Tato diplomová práce obsahuje: **návrh a realizaci struktury databáze** a **návrh a implementaci mobilní aplikace**. Antonín Neumann ve své diplomové práci s názvem **Jádro aplikace, rozhraní webových služeb a prezentační vrstva pro systém PartyBoard** měl na starosti: **návrh REST API, implementaci serveru** a **návrh a implementaci webových stránek**.

Vývojové prostředí si mohl zvolit každý vývojář sám dle svého uvážení. Na nástrojové sadě, úložišti, serveru a výběru SŘBD se vývojáři domluvili po analýze svých komponent.

Komentáře kódu jsou psané v českém jazyce s tím, že nebylo nutné komentovat zřejmé konstrukce, např. HTML elementy. Naopak jsou okomentovány klíčové části kódu z důvodu udržitelnosti aplikace a možnosti snadnějšího rozšíření.

Popis organizace souborů mobilní aplikace

Jedná se o jednu mobilní aplikaci pro platformu Android a iOS, která je postavena na technologiích Cordova, AngularJS a Ionic a je členěna do následující struktury:

- Složka **platform** obsahuje zdrojové kódy pro buildování (sestavení) aplikace pro jednotlivé platformy.
- Složka **plugins** obsahuje Cordova pluginy, např. plugin pro využívání hardwarových tlačítek.
- Složka **resources** obsahuje úvodní obrázek při spuštění aplikace a její ikonu.

- Složka **scss** uschovává zdrojové SCSS, ze kterých se generuje CSS.
- Složka **www** je hlavním adresářem webové aplikace (zobrazené ve WebView).
- Složka **img** obsahuje obrázky k aplikaci.
- Soubor **index.html** je hlavní šablonou aplikace, do které se vkládají ostatní šablony.
- Složka **js** obsahuje celou logiku aplikace.
- Soubor **app.js** obsahuje hlavní konfiguraci a routování.
- Soubor **dictionary.js** obsahuje veškeré texty použité v aplikaci. Slouží jako česko-anglický slovník.
- Složka **external** obsahuje veškeré mnou přidané externí knihovny.
- Složka **controllers** obsahuje kontroléry pro šablony aplikace.
- Složka **services** obsahuje „service“ a „factory“ aplikace.
- Složka **lib** obsahuje knihovnu Ionic.
- Složka **templates** obsahuje šablony aplikace.

Bližší popis provázanosti technologií a architektury mobilní aplikace je popsáno v podkapitole **Návrh a implementace mobilní aplikace**. Model perzistentních informací neboli datový model je vysvětlen v kapitole **Návrh a realizace struktury databáze**.

6.2 Použité technologie

Technologie použité pro vývoj mobilní aplikace:

- **Ionic** je HTML5 SDK framework, který umožňuje vytvářet nativní mobilní aplikace za pomoci webových technologií HTML, CSS a JavaScriptu. Ionic je zaměřen především na vzhled a interakci s uživatelským rozhraním aplikace. Aby měl framework Ionic plnou podporu, je nutné využít AngularJS. I přesto je stále možné využít klasické kaskádové styly, jenže kvůli tomu budou možná odebrána různá gesta, animace aj. [52].

- **AngularJS** je MVC[2] framework, obsahující řadu zajímavých myšlenek, přičemž mezi nejužitečnější koncepty frameworku patří Two Way Data-Binding („dvoucestná synchronizace dat“ – automatická synchronizace stavů a view), implementace Dependency Injection (návrhový vzor, který řeší závislost mezi jednotlivými komponentami programu), testovatelnost, direktivy („způsob jak naučit HTML novým trikům“) a znovu-použitelnost komponent.

- **Node.js** je serverový framework. Jedná se o vysoce výkonné, událostmi řízené prostředí pro JavaScript. Základem Node.js je javascriptový interpret V8 z Google Chrome a nad ním je tenká vrstva kódu v programovacím jazyce C++, který poskytuje minimální nutné zázemí [53].

- **Cordova** je nástroj pro vývoj mobilních aplikací, které využívají HTML, JavaScript a CSS pro jejich implementaci. Největší síla tohoto nástroje je v tom, že dokáže zabalit aplikaci do nativního kontejneru, který má přístup k funkcím přístroje jako např. fotoaparát. Tyto funkce jsou vystaveny prostřednictvím sjednoceného JavaScript API, což umožňuje snadno psát jednu sadu kódu (jednu aplikaci), kterou lze pak sestavit téměř pro jakýkoli telefon nebo tablet dostupný na dnešním trhu a publikovat ji do digitálních obchodů (např. App Store) [54].

- **REST API** je obecný architektonický vzor založený na spolupráci navzájem nezávislých služeb. Tato technologie byla popsána již v kapitole REST API.

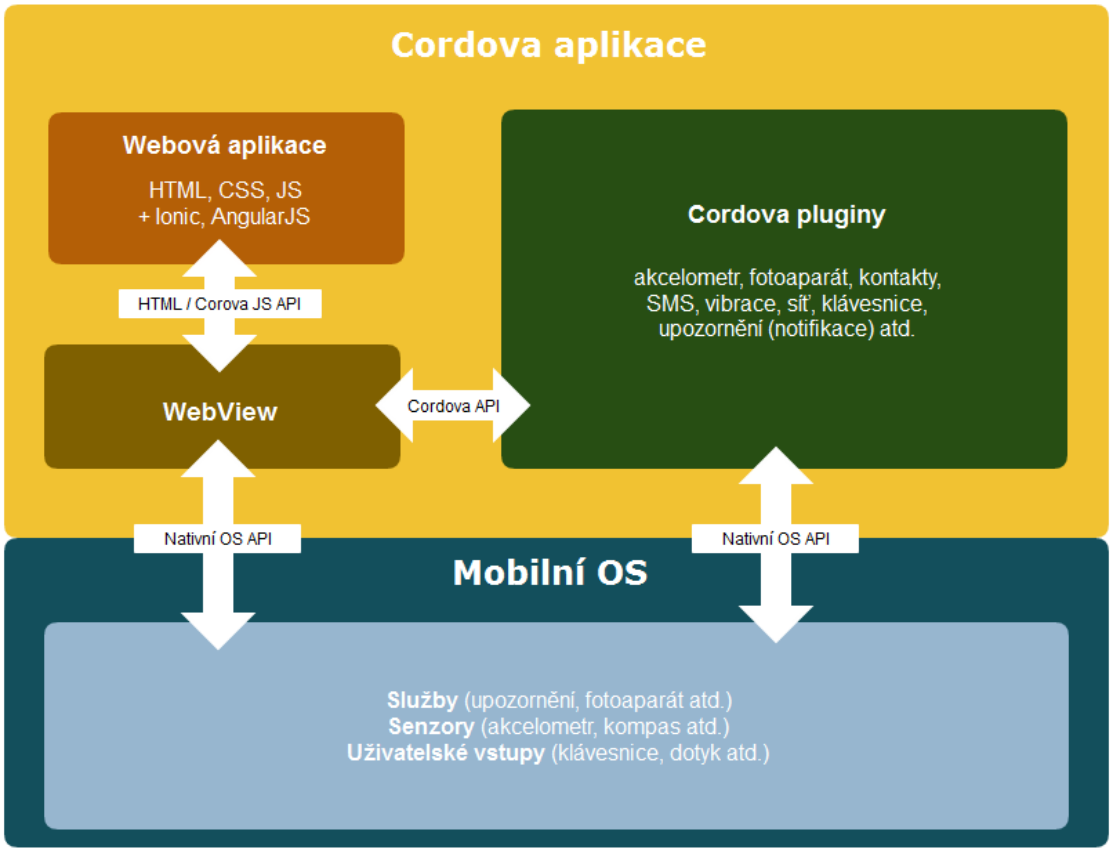
- **JSON** je typ způsobu zápisu dat (datový formát), který je nezávislý na počítačové platformě a je primárně určený pro přenos dat, která mohou být agregována v objektech nebo organizována v polích. Vstupem je libovolná datová struktura (číslo, řetězec, boolean, objekt nebo z nich složené pole) a výstupem je vždy řetězec. Složitost hierarchie vstupní proměnné není teoreticky nijak omezena [55].

- **JSON Web Token** je otevřený standard (RFC 7519), který definuje kompaktní a soběstačnou URL[3]-safe (bezpečná URL) pro bezpečný přenos informací mezi dvěma stranami pomocí JSON objektu, který je kryptograficky podepsán. Tyto informace mohou být ověřené a důvěryhodné pomocí digitálního podpisu. [56].

[2]MVC Model-view-controller
[3]URL Uniform Resource Locator

6.3 Návrh a implementace mobilní aplikace

Jak jsem se již dříve zmínil, pro vývoj mobilní aplikace byl použit hybridní přístup za pomoci technologie Cordova, AngularJS a Ionic. Rád bych dále vysvětlil samotné propojení zmíněných technologií.



Obrázek 6.5: Architektura mobilní aplikace

Cordova při vývoji hybridních aplikací slouží jako jakýsi „nativní obal“, který zobrazuje webovou aplikaci v komponentě WebView a umožňuje jí spojení s nativními službami v telefonu jako např. fotoaparát, klávesnice, upozornění (notifikace) atd. Přístup k těmto službám je zajištěn pomocí Cordova pluginů a fungování těchto služeb se pak nijak neliší od použití služeb v nativních aplikacích.

Část „webová aplikace“ je realizována pomocí Ionic a AngularJS frameworku.

Toto spojení je postaveno na MVC architektuře. Modelem jsou získaná data z databáze za pomoci dotazů odesílaných na server, který poskytuje REST API. Pro implementaci View jsou použity komponenty již zmíněného Ionic frameworku. Komunikace mezi Modelem a View je zajištěna pomocí kontrolérů, které jsou realizovány AngularuJS frameworkem, kde je ve velké míře využíváno „Two Way Data-Binding“ vlastnosti.

Struktura samotné aplikace již byla také vysvětlena dříve v bodě **Logický pohled**, který je součástí podkapitoly **Architektura aplikace**. Vlastní implementace je umístěna na CD, které je přílohou této diplomové práce.

Mimo základních pluginů (jako je např. plugin pro využívání klávesnice), které jsou automaticky přidány při vytvoření projektu, je také používán plugin pro práci s SMS zprávami a plugin pro detekci, zda je mobilní zařízení připojeno k internetu. Dále je využita knihovna „angular-translate“, která slouží pro automatickou změnu jazyka v celé aplikaci na základě nadefinovaného slovníku. View je tvořeno z několika HTML šablon, pro které je vždy naimplementován příslušný kontrolér. Pro vytvoření vlastních služeb je využito „service“ a „factory“, přičemž rozdíl mezi nimi je v tom, že „service“ si vždy vytváří a vrací nový objekt, kdežto „factory“ jeden a ten samý. Pro komunikaci se serverem využívám „service“ $HTTP, která je součástí AngularJS a umožňuje metody GET, POST, PUT a DELETE.

7 Návrh a realizace struktury databáze

V této kapitole jsou popsány jednotlivé tabulky, funkce, triggery a pohledy, které uchovávají veškerá data v systému Partyboard a usnadňují s nimi práci. Celkový návrh databáze vznikl na základě analýzy požadavků funkčnosti od potenciálních zákazníků. Na konci podkapitoly **Vysvětlení tabulek realizovaného datového modelu** je umístěn zjednodušený datový model s relačními vazbami a v přílohách této diplomové práce lze nalézt kompletní relační datový model, včetně jeho detailního technického popisu.

7.1 Vysvětlení tabulek realizovaného datového modelu

Tabulka USERS
Tato tabulka uchovává informace o všech registrovaných osobách (od nájemců až po běžné uživatele) v systému Partyboard.

Tabulka ROLES_ACCOUNT
Jedná se o číselník, který uchovává jednotlivé typy rolí. Tyto role jsou přiřazovány k registrovaným osobám pro systém Partyboard a jsou přiřazeny administrátorem systému Partyboard.

TABULKA PERMISSIONS_ACCOUNT
Jedná se o číselník, který uchovává jednotlivé typy práv, které jsou přidělovány ke konkrétním rolím. Tato práva jsou předdefinována administrátorem systému Partyboard.

Tabulka PAYMENTS_INFO
Tato tabulka uchovává informace o platebních údajích registrovaných osob v systému Partyboard.

Tabulka BANKS_CODE
Jedná se o číselník, který uchovává kódy jednotlivých bank.

TABULKA TOWNS
Jedná se o číselník, který uchovává názvy měst České republiky.

Tabulka ROLES_USERS_PARTYBOARDS
Tato tabulka uchovává informace o přiřazených rolích registrovaným osobám vůči konkrétním Partyboardům.

Tabulka ROLES_PARTYBOARD
Jedná se o číselník, který uchovává jednotlivé typy rolí, které mohou být registrovaným osobám přiřazeny pro správu jednotlivých Partyboardů. Tyto role jsou předdefinovány administrátorem systému Partyboard a jsou přiřazovány majitelem klubu.

TABULKA PERMISSIONS_PARTYBOARD
Jedná se o číselník, který uchovává jednotlivé typy práv, které je možné přidělit konkrétním rolím pro správu jednotlivých Partyboardů. Tato práva jsou předdefinována administrátorem systému Partyboard.

Tabulka BAN_USER_PARTYBOARD
Tato tabulka uchovává informace o osobách, které mají/měly ban vůči konkrétním Partyboardům. Nastavit ban osobě na Partyboard může vždy pouze manažer konkrétního Partyboardu.

Tabulka CLUBS
Tato tabulka uchovává informace o registrovaných klubech (včetně virtuálních klubů pro přenosné Partyboardy) v systému Partyboard.

Tabulka DIRTY_WORDS
Tato tabulka uchovává nevhodná (sprostá) slova, která slouží pro filtraci slov v příchozích zprávách.

TABULKA PARTYBOARDS
Tato tabulka uchovává vytvořené Partyboardy jednotlivých klubů a přenosné Partyboardy.

Tabulka GROUP_SETTINGS
Tato tabulka uchovává skupiny nastavení pro jednotlivé Partyboardy. Tyto skupiny nastavení vytváří samotní manažeři Partyboardů.

Tabulka SETTINGS
Tato tabulka uchovává konkrétní nastavení, ze kterých se skládají jednotlivé

skupiny nastavení. Toto nastavení provádí samotní manažeři Partyboardů na základě nadefinovaných struktur administrátorem.

Tabulka TYPE_SETTINGS
Tato tabulka uchovává možnosti (typy) nastavení pro jednotlivé Partyboardy. Tyto možnosti jsou definovány pro všechny Partyboardy administrátorem.

Tabulka DATA_TYPES
Jedná se o číselník, který uchovává jednotlivé datové typy, které je možné přidělit konkrétním typům nastavení. Tyto možnosti jsou definovány pro všechny typy nastavení administrátorem.

TABULKA OUTCOMMING_MESSAGES
Tato tabulka uchovává veškeré odeslané zprávy na mobilní zařízení ve formě SMS zprávy. Primárně slouží pouze jako log odeslaných zpráv.

TABULKA COMPETITION_PARTYBOARDS
Tato tabulka uchovává jednotlivé nastavení předdefinovaných soutěží pro jednotlivé Partyboardy. Tato nastavení jsou provedena manažery konkrétních Partyboardů.

TABULKA TYPE_COMPETITIONS
Tato tabulka uchovává struktury nastavení soutěží, které poskytuje systém Partyboard. Tyto struktury jsou definovány pro všechny Partyboardy administrátorem.

TABULKA SETTING_MESSAGES
Tato tabulka uchovává jednotlivé předdefinované texty výherních zpráv. Tento text definují manažeři konkrétních Partyboardů.

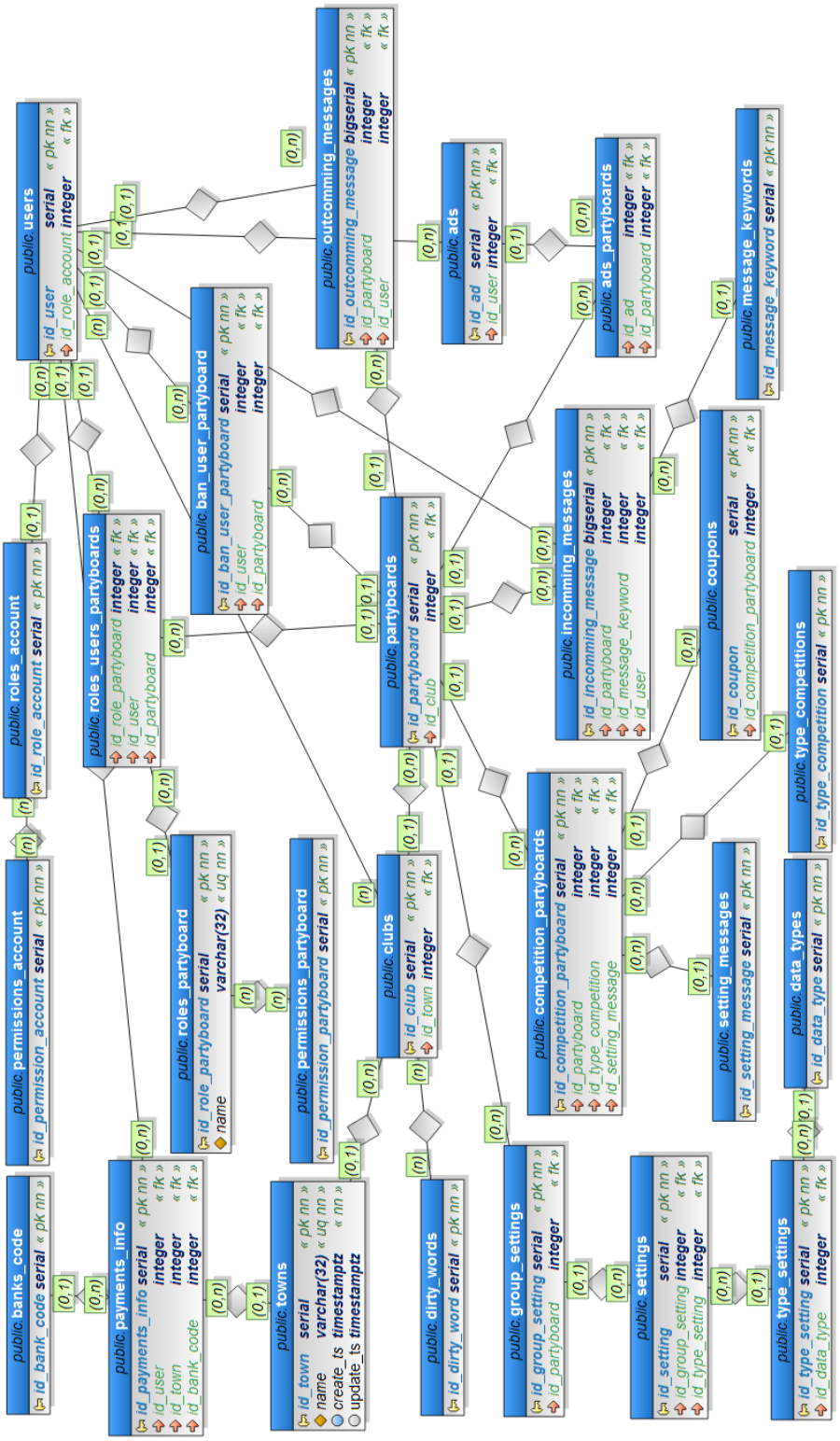
TABULKA COUPONS
Tato tabulka uchovává veškeré výherní kódy pro jednotlivé soutěže. Slouží jako přehled výherních kupónů pro manažery Partyboardů.

Tabulka INCOMMING_MESSAGES
Tato tabulka uchovává veškeré přijaté zprávy z internetu i z SMS zpráv.

Tabulka MESSAGE_KEYWORDS
Jedná se o číselník, který uchovává jednotlivé typy možných zpráv, které lze odesílat do systému Partyboard. Tyto struktury jsou definovány pro všechny Partyboardy administrátorem.

TABULKA ADS
Tato tabulka uchovává veškeré reklamy, které poskytuje systém Partyboard. Reklamy může přidávat jak administrátor, tak manažeři Partyboardů.

Tabulka ADS_PARTYBOARDS
Tato tabulka uchovává nastavení reklam pro jednotlivé Partyboardy. Toto nastavení provádí samotní manažeři Partyboardů nebo administrátor systému.



Obrázek 7.1: Zjednodušený relační databázový model

7.2 Sekvence v realizované databázi

Sekvence jsou speciální jednořádkové tabulky vytvořené příkazem `CREATE SEQUENCE název_sekvence`, které jsou běžně používány pro generování unikátních identifikátorů pro řádky tabulky.Sekvenční funkce poskytují jednoduché, víceuživatelské bezpečné metody pro získávání postupných hodnot ze sekvenční tabulky [59].

Funkce	**Návratová hodnota**	**Popis**
currval(regclass)	bigint	Vrátí poslední vrácenou hodnotu pro konkrétní sekvenci.
lastval()	bigint	Vrátí poslední vrácenou hodnotu poslední sekvence v dané session.
nextval(regclass)	bigint	Vrátí novou hodnotu.
setval(regclass, bigint)	bigint	Nastaví konkrétní sekvenci na novou hodnotu.
setval(regclass, bigint, boolean)	bigint	Nastaví konkrétní sekvenci na novou hodnotu s tím, že třetí parametr určuje, zda již byla sekvence volána. Pokud true (ano) a druhý parametr je například hodnota 10, tak při zavolání nextval(regclass) nám sekvence vrátí hodnotu 11. V opačném případě (false) vrátí 10.

Tabulka 7.1: Tabulka s popisem funkcí, které poskytuje sekvence

Sekvence využívám pro zajištění auto inkrementace všech tabulek v databázi při přidání nového záznamu do tabulky.

7.3 Pohledy v realizované databázi

Jak již bylo zmíněno v kapitole **Analýza jednotlivých SŘBD systému**, jedná se o databázové objekty, které poskytují data ve stejné podobě jako tabulky. K jejich vytvoření slouží příkaz `CREATE VIEW název_pohledu AS sql_dotaz`.

Tyto pohledy využívám pro usnadnění práce s daty uložené v databázi.

Pohled ROLE_PERMISSIONS_ACCOUNT
Tento pohled slouží pro výpis všech rolí s přiřazenými právy vůči celému systému Partyboard. Při přidání klauzule WHERE je pak možné rychle zjistit příslušnou roli (kterou má přiřazená každá registrovaná osoba) a k ní příslušná práva.

Pohled ROLE_PERMISSIONS_PARTYBOARD
Tento pohled slouží pro výpis všech rolí s přiřazenými právy vůči konkrétním registrovaným Partyboardům. Při přidání klauzule WHERE je pak možné rychle zjistit příslušnou roli (kterou má přiřazená každá registrovaná osoba) a k ní příslušná práva.

Pohled SETTINGS_DATA_TYPE
Tento pohled slouží pro výpis všech předdefinovaných možností nastavení s konkrétními datovými typy pro jednotlivé Partyboardy. Při přidání klauzule WHERE je pak možné rychle zjistit příslušný datový typ k jednomu nastavení.

Pohled SETTINGS_TYPE_SETTINGS
Tento pohled slouží pro výpis všech nadefinovaných nastavení manažery Partyboardů, které mají k sobě přiřazené hodnoty z pohledu SETTINGS_DATA _TYPE. Při přidání klauzule WHERE je pak možné rychle zjistit informace o konkrétním nastavení.

Pohled SETTINGS_GROUP_SETTINGS
Tento pohled slouží pro výpis všech nadefinovaných nastavení z pohledu SETTINGS_TYPE_SETTINGS s přiřazenými skupinami a jejich popisem. Při přidání klauzule WHERE je pak možné rychle zjistit příslušná nastavení jedné skupiny, kterou má přiřazenou konkrétní Partyboard.

Pohled BAN_USERS_PARTYBOARDS
Tento pohled slouží pro výpis všech uživatelů, kteří mají ban na některý z registrovaných Partyboardů. Při přidání klauzule WHERE je pak možné rychle zjistit všechny uživatele, kteří mají ban ke konkrétnímu Partyboardu nebo zjistit kolik banů má jeden uživatel vůči všem Partyboardům.

Pohled CLUB_TOWN
Tento pohled slouží pro výpis všech registrovaných klubů s příslušným městem v systému Partyboard. Při přidání klauzule WHERE je pak možné rychle zjistit kompletní informace o konkrétním klubu.

Pohled PARTYBOARD_DIRTY_WORDS
Tento pohled slouží pro výpis všech registrovaných nevhodných slov s příslušným Partyboardem. Při přidání klauzule WHERE je pak možné rychle zjistit nevhodná slova ke konkrétnímu Partyboardu.

Pohled ROLES_USERS_PARTYBOARDS
Tento pohled slouží pro výpis všech nastavených práv z pohledu ROLE_PERMISSIONS_PARTYBOARD k uživatelům a Partyboardům. Při přidání klauzule WHERE je pak možné rychle zjistit příslušné informace ke konkrétnímu uživateli a Partyboardu.

7.4 Funkce v realizované databázi

Jak již bylo zmíněno v kapitole **Analýza jednotlivých SŘBD systému**, jedná se o bloky programu, které při jejich zavolání vykonají nadefinovanou činnost.

Např. kód pro vytvoření funkce vypadá následovně:

```
CREATE FUNCTION create_current_date() RETURNS "trigger" AS
 $BODY$
 BEGIN
   New.create_ts := now();
   Return NEW;
 END;
 $BODY$
LANGUAGE 'plpgsql' VOLATILE;
```

Funkce CREATE_CURRENT_DATE
Tato funkce vrací aktuální čas a je volána ze všech triggerů s příkazem/podmínkou BEFORE INSERT. Slouží pro automatické vložení aktuálního času do sloupce CREATE_TS před vytvořením záznamu pro jednotlivé tabulky.

Funkce UPDATE_CURRENT_DATE
Tato funkce vrací aktuální čas a je volána ze všech triggerů s příkazem/podmínkou BEFORE UPDATE. Slouží pro automatické vložení aktuálního času do sloupce UPDATE_TS před upravením záznamu pro jednotlivé tabulky.

Funkce CREATE_DIRTY_WORDS
Tato funkce je volána z triggeru CREATE_DIRTY_WORDS_CLUBS s příkazem/podmínkou AFTER INSERT. Slouží pro automatické vytvoření záznamů s povinným seznamem nevhodných slov pro vytvořený klub.

7.5 Triggery v realizované databázi

Jak již bylo zmíněno v podkapitole **Analýza jednotlivých SŘBD systému**, jedná se o nadefinované činnosti, které se provedou při vyvolání události nad databázovou tabulkou či jejím sloupci.

Např. trigger pro tabulku ROLES_ACCOUNT vypadá následovně:

```
CREATE TRIGGER update_role_account
  BEFORE INSERT/UPDATE
  ON roles_account
  FOR EACH ROW
EXECUTE PROCEDURE create/update_current_date();
```

Triggery jsou využívány pro automatické vložení aktuálního času při vytvoření nebo upravení záznamu tabulky v databázi. Triggery jsou vytvořeny nad všemi tabulkami.

Trigger CREATE_DIRTY_WORDS_CLUBS
Tento trigger je spuštěn po vytvoření záznamu v tabulce CLUBS, který volá funkci CREATE_DIRTY_WORDS pro automatické vytvoření záznamů s povinným seznamem nevhodných slov pro vytvořený klub.

8 Nasazení a testování systému Partyboard

Tato kapitola pojednává o testování systému Partybord z hlediska zátěžových a funkčních testů přes mobilní aplikaci, které vycházely z navrženého diagramu případu užití.

Abych mohl pokračovat ve své práci na vývoji mobilní aplikace, než Antonín Neumann navrhne a naimplementuje výsledný REST API server pro mnou navržený databázový model systému Partyboard, naimplementoval jsem si pro účely testování funkčnosti mobilní aplikace vlastní REST API server v technologii PHP. Tento server mi umožňoval komunikaci mezi mobilním zařízením a databázovým serverem pomocí HTTP metod GET, POST, PUT a DELETE. Pro jeho implementaci byl využit Slim framework [60]. Server obsahuje 4 zmíněné základní metody, které jsou schopny pracovat se všemi tabulkami v databázi. Na základě URL adresy je vždy sestaven konkrétní SQL příkaz, který provede příslušnou operaci v databázi a vrátí výsledek.

Veškeré dále popsané testování již bylo provedeno na serveru od A. Neumanna.

8.1 Testování systému Partyboard přes mobilní aplikaci

V prvotní fázi došlo k naplnění databázového zdroje zkušebními daty, na kterých byly provedeny jednotlivé testy.

Zátěžový test vycházel z mimofunkčních požadavků, u kterých byl stanoven průměrný počet dotazů (1 900) za sekundu na webový server a do relační databáze. Tento počet dotazů je zaměřen hlavně na tabulku INCOMMING_MESSAGES, která je jednou z nejvytěžovanějších tabulek celé databáze, jelikož uchovává veškeré přijaté zprávy všech Partyboardů.

K tomuto testu byl využit nástroj „ApacheBench“ a webový server s databází byl nainstalován na počítač s parametry:

- Operační systém: MS Windows 10 64-bit;
- CPU: Intel Core i7 6700K (Skylake) @ 4.0 GHz, turbo 4.2 GHz;
- RAM: 32GB DDR4 @ 2666 MHz;
- Disk: Intel SSD 320 Series, 120 GB.

SQL dotazy na tento server byly zasílány z počítače s parametry:

- Operační systém: MS Windows 7 64-bit;
- CPU: Intel Core i7-2670QM CPU @ 2.2 GHz, turbo 3.1 GHz;
- RAM: 8GB DDR3 @ 1333 MHz;
- Disk: HDD 1000 GB, 5400 RPM.

Tyto počítače byly mezi sebou propojeny 1Gb/s ethernetovým (síťovým) kabelem.

Parametry testování:

- Počet záznamů v tabulce INCOMMING_MESSAGES: 90 000;
- SQL dotaz: SELECT * FROM INCOMMING_MESSAGES LIMIT 30;
- Počet současných požadavku: 100-1000;
- Celkový počet požadavků: 10 000 – 60 000.

Testování bylo provedeno s různými hodnotami počtu současných požadavků a celkovým počtem dotazů. Hodnota současných požadavků se inkrementovala po 100 a celkový počet dotazů po 10 000. Pro každou konfiguraci bylo měření provedeno 3x.

Výsledkem tohoto testování bylo, že webový server se SŘBD je schopný obsloužit průměrně 1450 dotazů s maximem **1986** dotazů za sekundu.

Funkčnost systému a mobilní aplikace jsem ověřil otestováním následujících položek:

- Registrace uživatele;
- Přihlášení/odhlášení uživatele;
- Výběr jazyka a úprava vzhledu aplikace;
- Zvolení Partyboardu, na který se mají posílat zprávy;
- Zobrazování a zasílání internetových a SMS zpráv;
- Manažer – nastavení konkrétního Partyboardu;
- Manažer – mazání přijatých zpráv na konkrétním Partyboardu;
- Manažer – zablokování uživatele pro konkrétní Partyboard.

8.2 Zhodnocení testování

I přesto, že provedení zátěžových testů neodpovídá přesné simulaci reálného provozu systému (nemožnos nasimulovat odhadovaný počet dotazujících se zařízení, veškeré možnosti požadavků na webový a databázový server, včetně neznalosti výsledného hardwaru pro reálný provoz systému Partyboard), tak z výsledů testů vyplynulo, že systém splňuje mimofunkční požadavky z hlediska zátěžových i funkčních testů pro mobilní aplikaci. Proto je možné tento systém nasadit do provozu pro jeden klub, který ho otestuje v rámci několika akcí.

9 Závěr

Cílem tohoto projektu bylo vytvoření komplexního systému s názvem Partyboard, na kterém jsme pracovali společně s Antonínem Neumannem. Mojí úlohou na tomto projektu bylo prostudování a návrh business a marketingového plánu na základě analýzy požadavků potenciálních zákazníků, návrh a realizace relačního datového modelu pro uchování dat ve zvoleném systému řízení báze dat a dále implementace mobilní aplikace s funkčností pro uživatele a administrátora. Antonín Neumann měl za úkol vytvoření korporátních stránek pro tento projekt, návrh a realizaci REST API, které využívá mnou vytvořený datový model a samotné stránky pro zobrazování zpráv.

Pro správné navržení business plánu a relačního modelu bylo velice důležité nastudovat různé možnosti tvorby business plánu. Tento plán byl následně sepsán na základě teoretických poznatků a na základě pochopení požadavků na systém od potenciálních uživatelů. Dále nedílnou součásti této práce bylo vybrat vhodný SŘBD, který měl za úkol pokrýt veškeré požadavky na správu dat. Z analýzy jednotlivých systémů vyplynulo, že všechny tyto systémy by plně postačily pro tuto aplikaci. Rozhodl jsem se pro databázový systém PostgreSQL na základě různých „vylepšení“ oproti ostatním analyzovaným systémům a také na základě osobní zkušenosti s tímto systémem.

Tato práce pro mě byla velice přínosná a to nejen z hlediska nových zkušeností s vývojem aplikací pro mobilní platformy, ale i z hlediska získání nových poznatků o méně známých open source databázových systémech. Také jsem se seznámil s možnostmi tvorby business plánu i jeho samotnou realizací, díky které jsem pochopil, jak je důležitá analýza celého projektu ze všech úhlů pohledu (uživatele, architekta, programátora, manažera atd.).

Pro tento projekt je aktuálně navržena podrobná analýza celého systému s kompletním relačním databázovým modelem a serverem, který poskytuje RESP API. Z důvodu velkého množství stráveného času nad podrobnou analýzou (ze které vyplynul business plán, relační model, REST API, webové stránky a mobilní aplikace), nedošlo ke kompletní implementaci funkčnosti celého systému, avšak hlavní funkčnost systému byla naimplementována a otestována. Tento systém aktuálně umožňuje uživatelskou registraci do systému, registraci klubů a k nim vytvoření webových stránek pro zobrazování přijatých internetových a SMS zpráv, jejich základní nastavení, nastavení rolí vůči celému systému Partyboard a vůči konkrétním Partyboardům registro-

vaných k jednotlivým klubům. Nastavení Partyboardu, zobrazování a zasílání zpráv je možno provádět přes mobilní aplikaci.

Systém se bude dále rozšiřovat o již zmíněné soutěže, ankety, reklamy atd., které jsou zahrnuty v již navržené architektuře systému. Relační model a REST API je pro toto rozšíření již plně připraveno.

Z výsledů testování vyplývá, že systém splňuje mimofunkční požadavky jak z hlediska zátěžových testů, tak také funkčních testů, které byly provedeny na základě diagramu případu užití.

Na základě popsání všech informací o výběru technologií, postupu při realizaci a výsledků testování jsem přesvědčen, že práci jsem splnil v plném rozsahu.

Seznam použitých zkratek

ACID *(Atomicity, Consistency, Isolation, Durability)*
Akronym pro vlastnosti databázové transakce.

AJAX *(Asynchronous JavaScript and XML)*
Jedná se o technologii vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich kompletního znovunačtení za pomoci asynchronního zpracování webových stránek.

API *(Application Programming Interface)*
Tato zkratka označuje programové rozhraní aplikací.

CRUD *(Create, Read/Retrieve, Update, Delete)*
Jedná se o zkratku, která shrnuje čtyři základní operace (vytvořit, číst / získat, upravit, odstranit), které lze vykonat nad záznamem v trvalém úložišti.

CSS *(Cascading Style Sheets)*
Jedná se jazyk, který slouží ke stylování webových stránek (HTML elementů).

HTML *(HyperText Markup Language)*
Jde o značkovací jazyk primárně určen pro webové stránky.

HTTP *(Hypertext Transfer Protocol)*
HTTP je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML.

HTTPS *(Hypertext Transfer Protocol over SSL)*
Jedná se o internetový protokol HTTP se zabezpečením SSL.

IDE *(Integrated Development Environment)*
Jedná se o zkratku, která označuje obecně jakékoli vývojové prostředí.

IP *(Internet Protocol)*
IP je v informatice základním protokolem pracujícím na síťové vrstvě používaným v počítačových sítích a internetu.

JSON *(JavaScriot Object Nonation)*
Jde o způsob zápisu dat (datový formát) určený pro jejich přenos.

MGA *(Multi-generation architecture)*
SŘBD vlastnost, která umožňuje vývoj a podporu hybridního OLTP a OLAP aplikací.

MVC *(Model-view-controller)*
Obecný architektonický vzor, který rozděluje aplikaci do tří částí.

MVCC *(Multi-Version Concurrency Control)*
SŘBD vlastnost, která řeší přístup více uživatelů ke stejným datům.

OLAP *(On-line Analytical Processing)*
Druh technologie uložení dat v databázi.

OLTP *(On-line Transaction Processing)*
Druh technologie uložení dat v databázi.

OS X *(Operating Systém X)*
OS X je označení operačního systému, který vznikl ze systému Mac OS (Macintosh Operating System), kde X značí verzi pořadí systému.

QUEL *(Relational database query language)*
Předchozí verze dotazovacího jazyka SQL.

REST *(Representational State Transfer)*
REST je architektura rozhraní navržená pro distribuované prostředí.

SCSS
Jedná se o modifikované (zjednodušené) kaskádové styly, ze kterých se generují CSS soubory.

SDK *(Software Development Kit)*
SDK označuje rozšíření pro vývoj aplikací pro daný systém (Android/iOS).

SQL *(Structured Query Language)*
Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.

SMART *(Specific, Measurable, Achievable, Realistic, Timed]*
SMART je zkratka složená z prvních písmen anglických slov pro mnemotechnickou pomůcku používanou v projektovém řízení ve fázích stanovení cílů.

SOA *(Service-oriented architecture)*
Jedná se o architekturu orientovanou na služby.

SOAP *(Simple Object Access Protocol)*
SOAP označuje protokol pro výměnu zpráv založených na XML přes síť, hlavně pomocí HTTP.

SŘBD *(Systém řízení báze dat)*
Softwarové vybavení, které zajišťuje práci s databází.

SSL *(Secure Sockets Layer)*
SSL je nekomerční otevřený protokol pro zabezpečení datových přenosů v rámci internetu mezi serverem s webovou prezentací a prohlížečem (uživatelem).

SWOT *(Strengths , Weaknesses, Opportunities, Threats)*
SWOT je zkratka odvozena z prvních písmen anglických názvů a jedná se o metodu, díky které je možno identifikovat silné a slabé stránky, příležitosti a hrozby, které jsou spojeny s určitým projektem, typem podnikání, podnikatelským záměrem apod.

TCP *(Transmission Control Protocol)*
TCP je základní internetový protokol pro komunikaci po internetu.

UML *(Unified Modeling Language)*
Grafický jazyk sloužící pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů.

URI *(Uniform Resource Identifier)*
URI označuje „jednotný identifikátor zdroje“.

URL *(Uniform Resource Locator)*
URL označuje „jednotnou adresu zdroje“.

WS *(Web Socket)*

WS je protokol, který poskytuje simultánní komunikační kanál mezi dvěma body přes jedno připojení TCP.

WSDL *(Web Services Description Language)*

WSDL je jazyk pro popis funkcí, jež nabízí webová služba. Dále slouží pro popis vstupů a výstupů těchto funkcí.

WSS *(Web Socket with SSL)*

WSS je stejný protokol jako WS, ale se zabezpečením SSL.

XML-RPC

XML-RPC je protokol, s jehož pomocí lze velice jednoduše provádět vzdálené volání procedur.

Literatura

[1] BLAŽKOVÁ, Martina. *Marketingové řízení a plánování pro malé a střední firmy*. 1. vyd. Praha: Grada, 2007. Manažer. ISBN 978-80-247-1535-3.

[2] PAYNE, Adrian. *Marketing služeb*. Vyd. 1. Praha: Grada, 1996. ISBN 80-716-9276-X.

[3] Podnikatelský plán a strategie *BusinessInfo.cz* [online]. [cit. 2016-03-18]. Dostupné z: `http://www.businessinfo.cz/cs/clanky/podnikatelsky-plan-a-strategie-23349.html#!&chapter=1`

[4] Jak napsat dobrý business plán *Inovace.cz* [online]. [cit. 2016-03-18]. Dostupné z: `http://www.inovace.cz/inovujte-efektivne/radce-pro-podnikatele/jak-napsat-dobr%C3%BD-business-pl%C3%A1n`

[5] Jak se může dobrá firma stát skvělou *Penize.cz* [online]. [cit. 2016-03-30]. Dostupné z: `http://www.penize.cz/podnikani/48191-jak-se-muze-dobra-firma-stat-skvelou`

[6] Smart Goals: 5 Steps To Smart Goal Setting (With Free Goal Planner Template) *Getorganizedwizard.com* [online]. [cit. 2016-04-02]. Dostupné z: `http://www.getorganizedwizard.com/blog/2009/02/smart-goals-5-steps-to-smart-goal-setting-with-free-goal-planner-template`

[7] HESKETT, James L. *Služby - cesta k úspěchu*. 1. vyd. Praha: Victoria Publishing, 1993, s. 164-165. ISBN 80-85605-36-8.

[8] STOLIČNÝ, Peter. *Marketingové komunikace v oboru služeb II: souvislosti marketingu a mediálních forem komunikace*. Vyd. 1. Praha: Vysoká škola hotelová v Praze 8, 2006. ISBN 80-86578-58-5.

[9] Jak si udržet zákazníka *Freshmarketing.cz* [online]. [cit. 2016-03-30]. Dostupné z: http://www.freshmarketing.cz/clanky/jak-si-udrzet-zakaznika

[10] 11 tipů, které udrží firmu na špici *Archiv.ihned.cz* [online]. [cit. 2016-03-30]. Dostupné z: http://archiv.ihned.cz/c1-41569650-11-tipu-ktere-udrzi-firmu-na-spici

[11] 12 nejčastějších chyb, kterými nevědomky posíláte vaše zákazníky nakupovat ke konkurenci... a jak se jich vyvarovat *Jakudrzetzakazniky.cz* [online]. [cit. 2016-03-30]. Dostupné z: http://www.jakudrzetzakazniky.cz

[12] 5 pravidel, jak získat a udržet věrného zákazníka *Podnikatel.cz* [online]. [cit. 2016-03-30]. Dostupné z: http://www.podnikatel.cz/clanky/5-pravidel-jak-ziskat-a-udrzet-verneho-zakaznika

[13] The History of MySQL. *Docstore.mik.ua* [online]. [cit. 2016-04-02]. Dostupné z: http://docstore.mik.ua/orelly/weblinux2/mysql/ch01_02.htm

[14] MATĚJKA, Martin. *Implementace testu k porovnání výkonnosti databázových systémů*. Praha, 2012. Bakalářská práce. Vysoká škola ekonomická v Praze. Vedoucí práce Ing. Dušan Chlapek, Ph.D.

[15] Oracle's ambitious plans for integrating Sun's technology *Infoworld.com* [online]. [cit. 2016-04-03]. Dostupné z: http://www.infoworld.com/article/2627785/m-a/oracle-s-ambitious-plans-for-integrating-sun-s-technology.html

[16] UPDATE Syntax. *Dev.mysql.com* [online]. [cit. 2016-04-03]. Dostupné z: http://dev.mysql.com/doc/refman/5.7/en/

[17] About MySQL. *Mysql.com* [online]. [cit. 2016-04-02]. Dostupné z: http://www.mysql.com/about/

[18] What Is New in MySQL 5.5. *Dev.mysql.com* [online]. [cit. 2016-04-03]. Dostupné z: http://dev.mysql.com/doc/refman/5.5/en/mysql-nutshell.html

[19] MySQL 5.7 Reference Manual. *Dev.mysql.com* [online]. [cit. 2016-04-02]. Dostupné z: http://dev.mysql.com/doc/refman/5.7/en/index.html

[20] SHOW CREATE VIEW Syntax. *Dev.mysql.com* [online]. [cit. 2016-04-04]. Dostupné z: http://dev.mysql.com/doc/refman/5.0/en/show-create-view.html

[21] KRÁL, Jakub. *Porovnání open source databázových systémů s využitím TPC-C testu*. Praha, 2013. Bakalářská práce. Vysoká škola ekonomická v Praze. Vedoucí práce Ing. Dušan Chlapek, Ph.D.

[22] MySQL. *Dev.mysql.com* [online]. [cit. 2016-04-04]. Dostupné z: http://dev.mysql.com/

[23] History PostgreSQL. *Postgresql.org* [online]. [cit. 2016-04-04]. Dostupné z: http://www.postgresql.org/about/history/

[24] PostgreSQL. *Postgresql.org* [online]. [cit. 2016-04-04]. Dostupné z: http://www.postgresql.org/

[25] KOŠÁREK, Lukáš. *Výkonnostní srovnání relačních databází*. Brno, 2010. Bakalářská práce. Masarykova univerzita. Vedoucí práce RNDr. Vlastislav Dohnal, Ph.D.

[26] About PostgreSQL. *Postgresql.org* [online]. [cit. 2016-04-05]. Dostupné z: http://www.postgresql.org/about/

[27] Functions and Operators. *Postgresql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.postgresql.org/docs/9.2/static/functions.html

[28] Introduction. *Postgresql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.postgresql.org/docs/9.2/static/mvcc-intro.html

[29] Inheritance. *Postgresql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.postgresql.org/docs/9.2/static/tutorial-inheritance.html

[30] Caveats. *Postgresql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.postgresql.org/docs/8.3/static/ddl-inherit.html\#DDL-INHERIT-CAVEATS

[31] ČINČURA, Jiří. *Systémové tabulky Firebirdu a využití jejich obsahu pro zjišťování struktury databázových objektů*. Brno, 2012. Bakalářská práce. Masarykova univerzita. Vedoucí práce RNDr. Jaroslav Pelikán, Ph.D.

[32] Firebird 2.5.2. *Firebirdsql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.firebirdsql.org/en/firebird-2-5-5/

[33] Features Firebird. *Firebirdsql.org* [online]. [cit. 2016-04-06]. Dostupné z: http://www.firebirdsql.org/en/features/

[34] Installing the JavaScript Connector *Dev.mysql.com* [online]. [cit. 2016-04-04]. Dostupné z: https://dev.mysql.com/doc/ndbapi/en/ndb-nodejs-setup.html

[35] Node.js hosting *Rosti.cz* [online]. [cit. 2016-04-06]. Dostupné z: https://rosti.cz/nodejs-hosting/

[36] Android zvyšuje podíl na celosvětovém trhu, iOS ztrácí*svetmobilne.cz* [online]. [cit. 2016-04-08]. Dostupné z: http://www.svetmobilne.cz/android-zvysuje-podil-na-celosvetovem-trhu-ios-ztraci/3588

[37] Android - licenses *Source.android.com* [online]. [cit. 2016-04-08]. Dostupné z: https://source.android.com/source/licenses.html

[38] Historie Androidu v kostce aneb Od verze 1.0 až po Android M *Svetandroida.cz* [online]. [cit. 2016-04-08]. Dostupné z: http://www.svetandroida.cz/historie-androidu-201506

[39] iOS: A visual history *Theverge.com* [online]. [cit. 2016-04-08]. Dostupné z: http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad

[40] Chci začít s programováním iOS aplikací. Kde mám začít? *Objevit.cz* [online]. [cit. 2016-04-08]. Dostupné z: http://objevit.cz/chci-zacit-s-programovanim-ios-aplikaci-kde-mam-zacit-t133970

[41] Getting started with developing for Windows Phone 8 *Msdn.microsoft.com* [online]. [cit. 2016-04-08]. Dostupné z: https://msdn.microsoft.com/library/windows/apps/ff402529

[42] Hybridní aplikace *Vyvojmobilniaplikace.cz* [online]. [cit. 2016-04-10]. Dostupné z: http://www.vyvojmobilniaplikace.cz/hybridni-aplikace/

[43] Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options *Developer.salesforce.com* [online]. [cit. 2016-04-10]. Dostupné z: https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options

[44] Web Sockets *Zdrojak.cz* [online]. [cit. 2016-04-10]. Dostupné z: https://www.zdrojak.cz/clanky/web-sockets/

[45] Komunikace v reálném čase díky Server-Sent Events a Web Sockets *Interval.cz* [online]. [cit. 2016-04-12]. Dostupné z: https://www.interval.cz/clanky/komunikace-v-realnem-case-diky-server-sent-events-a-web-sockets

[46] REST: architektura pro webové API *Zdrojak.cz* [online]. [cit. 2016-04-12]. Dostupné z: https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/

[47] KOUDELKA, Jakub. *Metoda návrhu REST API.* Praha, 2012. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Ing. Lukáš Burkoň

[48] Diagram případů užití. *Interval.cz* [online]. [cit. 2016-04-18]. Dostupné z: http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-slozitejsi-diagram-pripadu-uziti/

[49] Architektura 4+1 pohledů podle Kruchtena *Is.mendelu.cz* [online]. [cit. 2016-04-18]. Dostupné z: http://is.mendelu.cz/eknihovna/opory/index.pl?cast=4758

[50] PostgreSQL client for node.js *Npmjs.com* [online]. [cit. 2016-04-18]. Dostupné z: https://www.npmjs.com/package/pg

[51] SOA - Overview *Serviceorientation.com* [online]. [cit. 2016-04-24]. Dostupné z: http://serviceorientation.com/index.php/serviceorientation/index

[52] Ionic Documentation Overview *Ionicframework.com* [online]. [cit. 2016-04-24]. Dostupné z: http://ionicframework.com/docs/overview/#download

[53] JavaScript na serveru: Začínáme s Node.js *Zdrojak.cz* [online]. [cit. 2016-04-24]. Dostupné z: https://www.zdrojak.cz/clanky/javascript-na-serveru-zaciname-s-node-js/

[54] Apache Cordova *Cordova.apache.org* [online]. [cit. 2016-04-24]. Dostupné z: http://cordova.apache.org/

[55] Úvod do JSON *Json.org* [online]. [cit. 2016-04-24]. Dostupné z: http://json.org/json-cz.html

[56] JSON Web Token *Jwt.io* [online]. [cit. 2016-04-24]. Dostupné z: https://jwt.io/introduction/

[57] Vyvíjíme hybridní aplikace v Ionicu: Úvod a instalace*Zdrojak.cz* [online]. [cit. 2016-04-24]. Dostupné z: `https://www.zdrojak.cz/clanky/vyvijime-hybridni-aplikace-v-ionicu/`

[58] Angular translate *Github.com* [online]. [cit. 2016-04-24]. Dostupné z: `https://github.com/angular-translate/angular-translate`

[59] PostgreSQL 9.1.21 Documentation *Postgresql.org* [online]. [cit. 2016-04-24]. Dostupné z: `http://www.postgresql.org/docs/9.1/static/functions-sequence.html`

[60] Slim a micro framework for PHP *Slimframework.com* [online]. [cit. 2016-04-24]. Dostupné z: `http://www.slimframework.com/`

A Technický popis relační databáze

TABULKA USERS
Tato tabulka uchovává informace o všech registrovaných osobách (od nájemců až po běžné uživatele) v systému Partyboard.

`Id_user` - primární klíč tabulky.
`Id_role_account` - jedná se o cizí klíč dosazený z tabulky ROLES_ACCOUNT, která je ve vazbě 1:N s tabulkou USERS a určuje jednotlivé role osob vůči celému systému Partyboard. Vazba 1:N znamená, že každá osoba může mít pouze jednu roli a zároveň stejná role může být přiřazena více osobám.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_user`	1 (novak.jan@gmail.com)	`serial`
`id_role_account`	1 (owner)	`integer`
`e-mail`	novak.jan@gmail.com	`varchar(64)`
`password`	dsfsd48sdf. . . .dsdgg7gcqjuzk	`varchar(128)`
`nick`	Nowitz	`varchar(16)`
`phone`	00 420 728 635 942	`integer`
`firstname`	Jan	`varchar(32)`
`lastname`	Novák	`varchar(64)`
`birthdate`	1991-05-17	`date`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.1: Tabulka USER

Tabulka ROLES_ACCOUNT
Jedná se o číselník, který uchovává jednotlivé typy rolí. Tyto role jsou přiřazovány k registrovaným osobám pro systém Partyboard a jsou přiřazeny administrátorem systému Partyboard.

`Id_role_account` - primární klíč tabulky.
`name` - administrator, owner, user.

Název sloupce	Ukázková hodnota	Datový typ
id_role_account	1	serial
name	owner	varchar(32)
description	Majitel clubu.	text
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.2: Tabulka ROLES_ACCOUNT

Tabulka PERMISSIONS_ACCOUNT
Jedná se o číselník, který uchovává jednotlivé typy práv, které jsou přidělovány ke konkrétním rolím. Tato práva jsou předdefinována administrátorem systému Partyboard.

`Id_permission_account` - primární klíč tabulky.
`name` - create_partyboard, edit_partyboard, delete_partyboard atd.

Název sloupce	Ukázková hodnota	Datový typ
id_permission_account	1	serial
name	create_partyboard	varchar(32)
description	Založení Partyboardu.	text
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.3: Tabulka PERMISSIONS_ACCOUNT

Tabulka PERMISSIONS_ACCOUNT má vazbu M:N s tabulkou ROLES_ACCOUNT, protože každé právo může být přiřazeno k více rolím a zároveň každá role se může skládat z více práv.

Tabulka BANKS_CODE
Jedná se o číselník, který uchovává kód jednotlivých bank.

`Id_bank_code` - primární klíč tabulky.
`name` - FIO banka, Česká spořitelna atd.
`code` - 2010, 0800, atd.

Název sloupce	Ukázková hodnota	Datový typ
id_bank_code	1	serial
name	ČSOB	varchar(64)
code	0300	integer
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.4: Tabulka BANKS_CODE

Tabulka TOWNS
Jedná se o číselník, který uchovává názvy měst České republiky.

`Id_town` - primární klíč tabulky.
`name` - České Budějovice, Praha atd.

Název sloupce	Ukázková hodnota	Datový typ
id_town	1	serial
name	Plzeň	varchar(32)
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.5: Tabulka TOWNS

Tabulka PAYMENTS_INFO
Tato tabulka uchovává informace o platebních údajích registrovaných osob v systému Partyboard.

`Id_paments_info` - primární klíč tabulky.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou PAYMENTS_INFO a přiřazuje platební informace k osobám. Vazba 1:N znamená, že každá osoba může mít více platebních údajů a zároveň platební údaje mohou mít přiřazenou pouze jednu osobu.
`Id_bank_code` - jedná se o cizí klíč dosazený z tabulky BANKS_CODE, která je ve vazbě 1:N s tabulkou PAYMENTS_INFO a přiřazuje jednotlivé banky k platebním informacím. Vazba 1:N znamená, že každá banka může být přiřazena k více platebním údajům a zároveň platební údaje mohou mít přiřazenou pouze jednu banku.
`Id_town` - jedná se o cizí klíč dosazený z tabulky TOWNS, která je ve vazbě 1:N s tabulkou PAYMENTS_INFO a přiřazuje jednotlivá města k platebním informacím. Vazba 1:N znamená, že každé město může být přiřazeno k více

platebním údajům a zároveň platební údaje mohou mít přiřazené pouze jedno město.

Název sloupce	Ukázková hodnota	Datový typ
id_payments_info	1	serial
id_user	1 (novak.jan@gmail.com)	integer
id_town	1 (Plzeň)	integer
id_bank_code	1 (ČSOB)	integer
street	Dr. Stejskala	varchar(128)
street_number	23	integer
post_code	37001	integer
bank_account	215368951	integer
variable_symbol	9245321685	integer
delete	false	boolean
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.6: Tabulka PAYMENTS_INFO

Tabulka ROLES_USERS_PARTYBOARDS
Tato tabulka uchovává informace o přiřazených rolích registrovaným osobám vůči konkrétním Partyboardům.

`Id_role_partyboard` - jedná se o cizí klíč dosazený z tabulky ROLES_PARTYBOARD, která je ve vazbě 1:N s tabulkou ROLES_USERS_PARTYBOARDS a přiřazuje jednotlivé role vůči konkrétním uživatelům a Partyboardům. Vazba 1:N znamená, že každý Partyboard může mít více záznamů v této tabulce a zároveň každý záznam může mít přiřazenou pouze jednu roli.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou ROLES_USERS_PARTYBOARDS a určuje konkrétní uživatele, kterým je přiřazena konkrétní role vůči jednotlivým Partyboardům. Vazba 1:N znamená, že každá osoba může mít více záznamů v této tabulce a zároveň každý záznam může mít přiřazenou pouze jednu osobu.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou ROLES_USERS_PARTYBOARDS a určuje konkrétní Partybordy k jednotlivým osobám s určitou rolí. Vazba 1:N znamená, že každý Partyboard může mít více záznamů v této tabulce a zároveň každý záznam může mít přiřazen pouze jeden Partyboard.

Název sloupce	Ukázková hodnota	Datový typ
id_role_partyboard	1 (manager)	integer
id_user	1 (novak.jan@gmail.com)	integer
id_partyboard	1 (Mexx_valentýn)	integer
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.7: Tabulka ROLES_USERS_PARTYBOARDS

Tabulka ROLES_PARTYBOARD
Jedná se o číselník, který uchovává jednotlivé typy rolí, které mohou být registrovaným osobám přiřazeny pro správu jednotlivých Partyboardů. Tyto role jsou předdefinovány administrátorem systému Partyboard a jsou přiřazovány majitelem klubu.

`Id_role_partyboard` - primární klíč tabulky.
`name` - manager, dj.

Název sloupce	Ukázková hodnota	Datový typ
id_role_partyboard	1	serial
name	Manager	varchar(32)
description	Správce Partyboardu.	text
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.8: Tabulka ROLES_PARTYBOARD

Tabulka PERMISSIONS_PARTYBOARD
Jedná se o číselník, který uchovává jednotlivé typy práv, které je možné přidělit konkrétním rolím pro správu jednotlivých Partyboardů. Tato práva jsou předdefinována administrátorem systému Partyboard.

`Id_permission_partyboard` - primární klíč tabulky.
`name` - create_pb_ban, update_pb_ban, delete_pb_ban atd.

Název sloupce	Ukázková hodnota	Datový typ
id_permission_partyboard	1	serial
name	create_pb_ban	varchar(32)
description	Právo pro přidání reklamy.	text
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.9: Tabulka PERMISSIONS_PARTYBOARD

Tabulka PERMISSIONS_PARTYBOARD má vazbu M:N s tabulkou ROLES_PARTYBOARD, protože každé právo může být přiřazeno k více rolím a zároveň se každá role může skládat z více práv.

Tabulka CLUBS
Tato tabulka uchovává informace o registrovaných klubech (včetně virtuálních klubů pro přenosné Partyboardy) v systému Partyboard.

`Id_club` - primární klíč tabulky.
`Id_town` - jedná se o cizí klíč dosazený z tabulky TOWNS, která je ve vazbě 1:N s tabulkou CLUBS a přiřazuje jednotlivá města ke klubům. Vazba 1:N znamená, že každé město může být přiřazeno k více klubům a zároveň každý klub může mít přiřazené pouze jedno město.

Název sloupce	Ukázková hodnota	Datový typ
id_club	1	serial
id_town	1 (Plzeň)	integer
name	Mexx	varchar(64)
street	Pohůrecká	varchar(128)
street_number	45	integer
post_code	256 24	integer
delete	false	boolean
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.10: Tabulka CLUBS

Tabulka DIRTY_WORDS
Tato tabulka uchovává nevhodná (sprostá) slova, která slouží pro filtraci slov v příchozích zprávách.

`Id_dirty_word` - primární klíč tabulky.
`word` - kráva, debil atd.

Název sloupce	Ukázková hodnota	Datový typ
`id_dirty_word`	1	`serial`
`word`	kráva	`varchar(16)`
`main`	true (platný pro všechny kluby)	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.11: Tabulka DIRTY_WORDS

Tabulka DIRTY_WORDS má vazbu M:N s tabulkou CLUBS a to kvůli tomu, že každé sprosté slovo může být přiřazeno k více klubům a zároveň každý klub může mít přiřazených více sprostých slov.

Tabulka PARTYBOARDS
Tato tabulka uchovává vytvořené Partyboardy jednotlivých klubů a přenosné Partyboardy.

`Id_partyboard` - primární klíč tabulky.
`Id_club` - jedná se o cizí klíč dosazený z tabulky CLUBS, která má vazbu 1:N s tabulkou CLUBS a přiřazuje jednotlivé Partyboardy ke klubům. Vazba 1:N znamená, že každému klubu může být přiřazeno více Partyboardů a zároveň každý Partyboard může být přiřazen pouze k jednomu klubu.

Název sloupce	Ukázková hodnota	Datový typ
`id_partyboard`	1	`serial`
`id_club`	1 (Mexx)	`integer`
`name`	Mexx_valentýn	`varchar(32)`
`sms_key`	key_mexx (unikátní sms klíč)	`varchar(32)`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.12: Tabulka PARTYBOARDS

Tabulka BAN_USER_PARTYBOARD
Tato tabulka uchovává informace o osobách, které mají/měly ban vůči konkrétním Partyboardům. Nastavit ban osobě na Pártyboard může vždy pouze manažer konkrétního Partyboardu.

`Id_ban_user_partyboard` - primární klíč tabulky.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou BAN_USER_PARTYBOARD a určuje konkrétní uživatele, kteří dostali ban na konkrétní Partyboardy. Vazba 1:N znamená, že každá osoba může mít více záznamů v této tabulce a zároveň každý záznam může mít přiřazenou pouze jednu osobu.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou BAN_USER_PARTYBOARD a určuje konkrétní Partyboardy, na které uživatelé dostali ban. Vazba 1:N znamená, že každý Partyboard může mít více záznamů v této tabulce a zároveň každý záznam může mít přiřazený pouze jeden Partyboard.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_ban_user_partyboard`	1	`serial`
`id_user`	1 (novak.jan@gmail.com)	`integer`
`id_partyboard`	1 (Mexx_valentýn)	`integer`
`phone`	null	`varchar(14)`
`length_hour`	64 (hodin)	`integer`
`ongoing`	true (probíhající ban)	`boolean`
`description`	Zpráva jako reklama.	`text`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.13: Tabulka BAN_USER_PARTYBOARD

Tabulka GROUP_SETTINGS
Tato tabulka uchovává skupiny nastavení pro jednotlivé Partyboardy. Tyto skupiny nastavení vytváří samotní manažeři Partyboardů.

`Id_group_setting` - primární klíč tabulky.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou GROUP_SETTINGS a přiřazuje konkrétní skupiny nastavení k Partyboardům. Vazba 1:N znamená, že každý Partyboard může mít více skupin nastavení a zároveň každý Partyboard může mít přiřazenou pouze jednu skupinu nastavení.

Název sloupce	Ukázková hodnota	Datový typ
id_group_setting	1	serial
id_partyboard	1 (Mexx_valentýn)	integer
name	Valentýn	varchar(32)
active	true	boolean
description	Valentýnský nastavení.	text
delete	false	boolean
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.14: Tabulka GROUP_SETTINGS

Tabulka SETTINGS
Tato tabulka uchovává konkrétní nastavení, ze kterých se skládají jednotlivé skupiny nastavení. Toto nastavení provádí samotní manažeři Partyboardů na základě nadefinovaných struktur administrátorem.

`Id_setting` - primární klíč tabulky.
`Id_group_setting` - jedná se o cizí klíč dosazený z tabulky GROUP_SETTINGS, která je ve vazbě 1:N s tabulkou SETTINGS a slouží pro přiřazení konkrétních nastavení ke skupinám nastavení. Vazba 1:N znamená, že každá skupina nastavení může mít více konkrétních nastavení a zároveň každé konkrétní nastavení může být přiřazeno pouze k jedné skupině nastavení.
`Id_type_setting` - jedná se o cizí klíč dosazený z tabulky TYPE_SETTINGS, která je ve vazbě 1:N s tabulkou SETTINGS a slouží k výběru předdefinovaných konkrétních typů nastavení. Vazba 1:N znamená, že každý předdefinovaný typ nastavení může být přiřazen k více nastavením a zároveň každé konkrétní nastavení může mít přiřazený pouze jeden předdefinovaný typ nastavení.

Název sloupce	Ukázková hodnota	Datový typ
id_setting	1	serial
id_group_setting	1 (Valentýn)	integer
id_type_setting	1 (text_color)	integer
value	#FFFFFF	varchar(1024)
create_ts	2016-04-15 10:23:54+01	timestamptz
update_ts	2016-05-03 21:43:12+01	timestamptz

Tabulka A.15: Tabulka SETTINGS

Tabulka TYPE_SETTINGS
Tato tabulka uchovává možnosti (typy) nastavení pro jednotlivé Partyboardy. Tyto možnosti jsou definovány pro všechny Partyboardy administrátorem.

`Id_type_setting` - primární klíč tabulky.
`Id_data_type` - jedná se o cizí klíč dosazený z tabulky DATA_TYPE, která je ve vazbě 1:N s tabulkou DATA_TYPES a určuje datové typy hodnot nastavení. Vazba 1:N znamená, že každý datový typ může být přiřazen k více možnostem nastavení a zároveň každá možnost nastavení může mít přiřazený pouze jeden datový typ.
`name` - color, ads, logo atd.

Název sloupce	Ukázková hodnota	Datový typ
`id_type_setting`	1	`serial`
`id_data_type`	1 (String)	`integer`
`name`	text_color	`varchar(16)`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.16: Tabulka TYPE_SETTINGS

Tabulka DATA_TYPES
Jedná se o číselník, který uchovává jednotlivé datové typy, které je možné přidělit konkrétním typům nastavení. Tyto možnosti jsou definovány pro všechny typy nastavení administrátorem.

`Id_data_type` - primární klíč tabulky.
`name` - string, url, number atd.
`regex` - .+, atd.

Název sloupce	Ukázková hodnota	Datový typ
`id_data_type`	1	`serial`
`name`	string	`varchar(16)`
`regex`	.+	`varchar(1024)`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.17: Tabulka DATA_TYPES

Tabulka OUTCOMMING_MESSAGES
Tato tabulka uchovává veškeré odeslané zprávy na mobilní zařízení ve formě SMS zprávy. Primárně slouží pouze jako log odeslaných zpráv.

`Id_outcomming_message` - primární klíč tabulky.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou OUTCOMMING_MESSAGES a přiřazuje odeslané zprávy k osobám. Vazba 1:N nám říká, že každé osobě může být posláno více zpráv a zároveň každá zpráva může být odeslána pouze jedné osobě.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou OUTCOMMING_MESSAGES a přiřazuje konkrétní Partyboardy k odeslaným zprávám. Vazba 1:N znamená, že každý Partyboard může odeslat více zpráv a zároveň každá zpráva může být odeslána pouze z jednoho Partyboardu.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_outcomming_message`	1	`bigserial`
`id_partyboard`	1 (Mexx_valentýn)	`integer`
`id_user`	1 (novak.jan@gmail.com)	`integer`
`phone`	00 000 728 635 942	`varchar(14)`
`text`	Váš výherní kód je: 356acs :-)	`text`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`

Tabulka A.18: Tabulka OUTCOMMING_MESSAGES

Tabulka COMPETITION_PARTYBOARDS
Tato tabulka uchovává jednotlivé nastavení předdefinovaných soutěží pro jednotlivé Partyboardy. Tato nastavení jsou provedena manažery konkrétních Partyboardů.

`Id_competition_partyboard` - primární klíč tabulky.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou COMPETITION_PARTYBOARDS a přiřazuje konkrétní Partyboardy k jednotlivým nastavením soutěží. Vazba 1:N znamená, že každý Partyboard může mít přiřazených více soutěží a každá soutěž může být přiřazena pouze k jednomu Partyboardu.
`Id_type_competition` - jedná se o cizí klíč dosazený z tabulky TYPE_COMPETITIONS, která je ve vazbě 1:N s tabulkou COMPETITION_PARTYBOARDS a určuje typ předdefinovaných soutěží.

Vazba 1:N znamená, že každý předdefinovaný typ soutěže může být použit pro více nadefinovaných soutěží a zároveň každá nadefinovaná soutěž může mít přirazený pouze jeden předdefinovaný typ soutěže.
`Id_setting_message` - jedná se o cizí klíč dosazený z tabulky SETTINGS_MESSAGES, která je ve vazbě 1:N s tabulkou COMPETITION_PARTYBOARDS a určuje konkrétní typ výherních zpráv. Vazba 1:N znamená, že každý typ výherní zprávy může být použit pro více nadefinovaných soutěží a zároveň každá nadefinovaná soutěž může mít přirazený pouze jeden typ výherní zprávy.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_competition_partyboard`	1	`serial`
`id_partyboard`	1 (Mexx_valentýn)	`integer`
`id_type_competition`	1 (x_sms_win)	`integer`
`id_setting_message`	1 (Váš výherní kód je:)	`integer`
`name`	X-tá SMS zpráva vyhrává.	`varchar(32)`
`input_vars`	each_sms:5	`text`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.19: Tabulka COMPETITION_PARTYBOARDS

Tabulka SETTING_MESSAGES
Tato tabulka uchovává jednotlivé předdefinované texty výherních zpráv. Tento text definují manažeři konkrétních Partyboardů.

`Id_settings_message` - primární klíč tabulky.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_setting_message`	1	`serial`
`text`	Váš výherní kód je:	`text`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.20: Tabulka SETTING_MESSAGES

Tabulka TYPE_COMPETITIONS
Tato tabulka uchovává struktury nastavení soutěží, které poskytuje systém Partyboard. Tyto struktury jsou definovány pro všechny Partyboardy administrátorem.

`Id_type_competition` - primární klíč tabulky.

Název sloupce	Ukázková hodnota	Datový typ
`id_type_competition`	1	`serial`
`name`	x_sms_win	`varchar(32)`
`input_vars`	each_sms:X	`text`
`output_vars`	356acs (vygenerovaný kód)	`text`
`input_params`	key_mexx; Mexx_valentýn;	`text`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.21: Tabulka TYPE_COMPETITIONS

Tabulka INCOMMING_MESSAGES
Tato tabulka uchovává veškeré přijaté zprávy z internetu i SMS zpráv.

`Id_incomming_message` - primární klíč tabulky.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou INCOMMING_MESSAGES a přiřazuje přijaté zprávy k osobám. Vazba 1:N znamená, že každá osoba může poslat více zpráv a zároveň každá zpráva může mít pouze jednoho odesílatele.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou INCOMMING_MESSAGES a určuje, na které Partyboardy byly příchozí zprávy zaslány. Vazba 1:N znamená, že na každý Partyboard může být odesláno více zpráv a zároveň každá zpráva může být odeslána pouze na jeden Partyboard.
`Id_message_keyword` - jedná se o cizí klíč dosazený z tabulky MESSAGE_KEYWORDS, která je ve vazbě 1:N s tabulkou INCOMMING_MESSAGES a určuje jednotlivé typy zpráv. Vazba 1:N znamená, že každý typ zprávy může být použit pro více příchozích zpráv a zároveň každá zpráva může mít přiřazený pouze jeden typ zprávy.

Název sloupce	Ukázková hodnota	Datový typ
id_incomming_message	1	bigserial
id_partyboard	1 (Mexx_valentýn)	integer
id_message_keyword	1 (SMS)	integer
id_user	null	integer
nick	Nowitz	varchar(16)
text	Ahoj lidi, jak se dneska bavíte?	text
phone	00 000 728 635 942	varchar(32)
delete	false	boolean
create_ts	2016-04-15 10:23:54+01	timestamptz

Tabulka A.22: Tabulka INCOMMING_MESSAGES

Tabulka COUPONS
Tato tabulka uchovává veškeré výherní kódy pro jednotlivé soutěže. Slouží jako přehled výherních kupónů pro manažery Partyboardů.

`Id_coupons` - primární klíč tabulky.
`Id_competition_partyboard` - jedná se o cizí klíč dosazený z tabulky COMPETITION_PARTYBOARDS, která je ve vazbě 1:N s tabulkou COUPONS a přiřazuje konkrétní nastavení soutěží k výherním kupónům. Vazba 1:N znamená, že každá nadefinovaná soutěž může mít více výherních kupónů a zároveň každý výherní kupón může být přirazen pouze k jedné nadefinované soutěži.

Název sloupce	Ukázková hodnota	Datový typ
id_coupon	1	serial
id_competition_partyboard	1	integer
code	356acs	varchar(32)
valid_date	2016-04-22 13:59:59+01	timestamptz
used	false (zda byl kód využit)	boolean
delete	false	boolean
create_ts	2016-04-15 10:23:54+01	timestamptz

Tabulka A.23: Tabulka COUPONS

Tabulka MESSAGE_KEYWORDS
Jedná se o číselník, který uchovává jednotlivé typy možných zpráv, které je lze odesílat do systému Partyboard. Tyto struktury jsou definovány pro

všechny Partyboardy administrátorem.

`Id_message_keyword` - primární klíč tabulky.
`name` - network, SMS atd.

Název sloupce	Ukázková hodnota	Datový typ
`id_message_keyword`	1	`serial`
`name`	SMS	`varchar(16)`
`description`	SMS zpráva	`text`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.24: Tabulka MESSAGE_KEYWORDS

Tabulka ADS
Tato tabulka uchovává veškeré reklamy, které poskytuje systém Partyboard. Reklamy může přidávat jak administrátor, tak i manažeři Partyboardů.

`Id_ad` - primární klíč tabulky.
`Id_user` - jedná se o cizí klíč dosazený z tabulky USERS, která je ve vazbě 1:N s tabulkou ADS a určuje uživatele, kteří přidali reklamy. Vazba 1:N znamená, že každá pověřená osoba může přidat více reklam a zároveň každá reklama může být přidána pouze jednou osobou.

Název sloupce	Ukázková hodnota	Datový typ
`id_ad`	1	`serial`
`id_user`	1 (novak.jan@gmail.com)	`integer`
`name`	Amundsen vodka	`varchar(16)`
`accepted`	true (zda byla schválena)	`boolean`
`url_img`	./img/ads/amn01.png	`varchar(128)`
`price`	100 (Kč/zobrazení)	`integer`
`description`	Reklama na novou vodku.	`text`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.25: Tabulka ADS

Tabulka ADS_PARTYBOARDS
Tato tabulka uchovává nastavení reklam pro jednotlivé Partyboardy. Toto nastavení provádí samotní manažeři Partyboardů nebo administrátor systému.

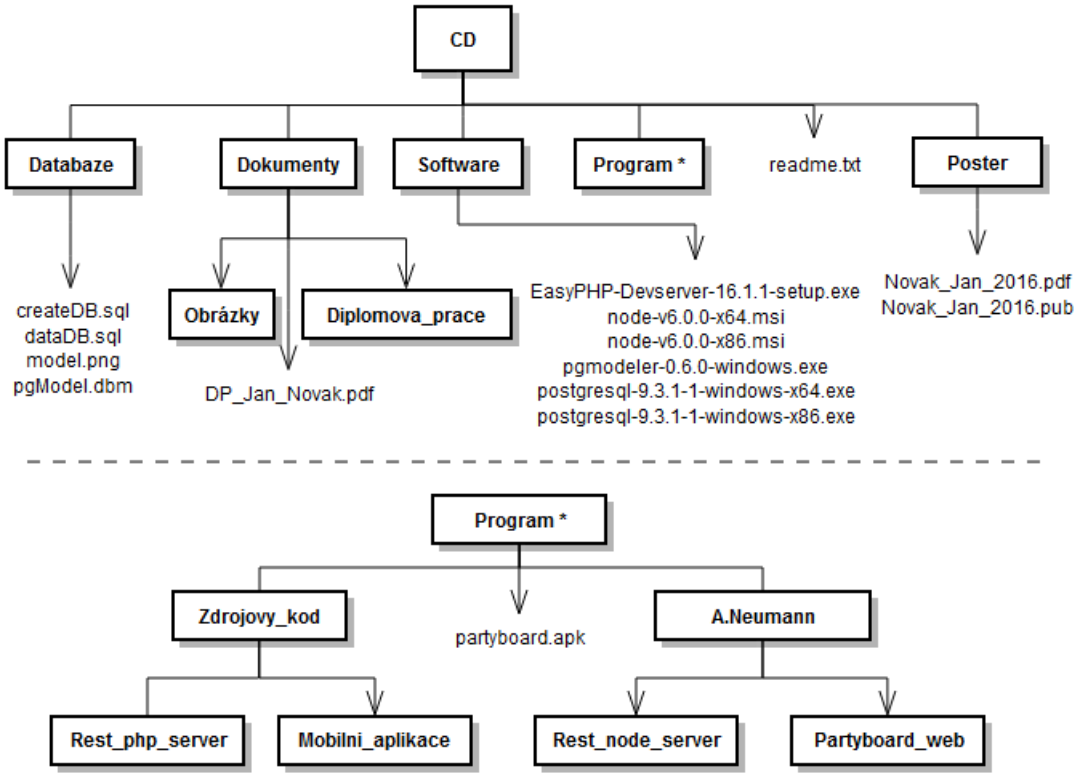
`Id_ad` - jedná se o cizí klíč dosazený z tabulky ADS, která je ve vazbě 1:N s tabulkou ADS_PARTYBOARDS a určuje konkrétní reklamy pro Partyboardy.
Vazba 1:N znamená, že každá reklama může mít více nastavení a zároveň každé nastavení reklamy může být přiřazeno pouze jednomu Partyboardu.
`Id_partyboard` - jedná se o cizí klíč dosazený z tabulky PARTYBOARDS, která je ve vazbě 1:N s tabulkou ADS_PARTYBOARDS a určuje Partyboardy, kterým jsou reklamy přiřazené. Vazba 1:N znamená, že každý Partyboard může mít přiřazených více nastavených reklam a zároveň každá nastavená reklama může být přidána pouze k jednomu Partyboardu.

Název sloupce	**Ukázková hodnota**	**Datový typ**
`id_ad`	1 (Amundsen vodka)	`integer`
`id_partyboard`	1 (Mexx_valentýn)	`integer`
`views_number`	0 (počet zobrazení)	`integer`
`timeout`	3 (zobrazeno vždy na 3 sek.)	`integer`
`repeat`	1000 (počet opakování)	`integer`
`hour_from`	23:00:00+01	`timetz`
`hour_to`	02:00:00+01	`timetz`
`active_from`	2016-04-15	`date`
`active_to`	2016-05-06	`date`
`active`	true	`boolean`
`delete`	false	`boolean`
`create_ts`	2016-04-15 10:23:54+01	`timestamptz`
`update_ts`	2016-05-03 21:43:12+01	`timestamptz`

Tabulka A.26: Tabulka ADS_PARTYBOARDS

Výsledný relační databázový model je přiložen v deskách této diplomové práce.

B Struktura obsahu CD



Obrázek B.1: Struktura obsahu CD

- **Databaze**
  - createDB.sql – SQL soubor pro vytvoření databáze bez zkušebních dat.
  - dataDB.sql - Testovací data, funkce, triggery atd.
  - model.png – Obrázek relačního datového modelu.
  - pdModel.dbm – Soubor pro otevření projektu databázového modelu.
- **Dokumenty**
  - Diplomova_prace – Složka se zdrojovými soubory (TEX) pro vytvoření dokumentu.

  - Obrazky – Složka s originálními obrátky použitých v diplomové práci.
  - DP_Jan_Novak.pdf – Výsledný dokument diplomové práce.

- **Program**
  - partyboard.apk - Mobilní aplikace, napojená na testovací prostředí.
    (Přihlašovací údaje - admin:root@root.cz/root, user: user@user.cz/user)
  - Zdrojovy_kod
    * Mobilni_aplikace - Složka s projektem mobilní aplikace.
    * Rest_php_server - Složka s projektem REST API PHP serveru.
  - A.Neumann - Složka s komponentami A. Neumanna.
    * Partyboard_web - Složka s projektem boardu, který zobrazuje přijaté zprávy konkrétního Partyboardu.
    * Rest_node_server - Složka s projektem REST API Node.js serveru.

- **Poster**
  - Novak_Jan_2016.pub - Poster diplomové práce ve formátu PUB.
  - Novak_Jan_2016.pdf - Poster diplomové práce ve formátu PDF.

- **Software**
  - pgmodeler-0.6.0-windows.exe – PostgreSQL Database Modeler.
  - postgresql-9.3.1-1-windows-x64.exe – PostgreSQL, MS Windows, 64bit.
  - postgresql-9.3.4-3-windows-x86.exe – PostgreSQL, MS Windows, 32bit.
  - EasyPHP-Devserver-16.1.1-setup.exe – PHP server. MS Windows.
  - node-v6.0.0-x64.msi - Node.js, MS Windows, 64bit.
  - node-v6.0.0-x86.msi - Node.js, MS Windows, 32bit.