



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

APLIKACE PRO SPRÁVU ZAKÁZEK

CONTRACT MANAGEMENT APPLICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Roman Filipovič

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2020

Zadání bakalářské práce

Ústav:	Ústav informatiky
Student:	Roman Filipovič
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Manažerská informatika
Vedoucí práce:	Ing. Petr Dydowicz, Ph.D.
Akademický rok:	2019/20

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

Aplikace pro správu zakázek

Charakteristika problematiky úkolu:

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

Cíle, kterých má být dosaženo:

Návrh aplikace pro mobilní zařízení, jejíž účelem bude správa zakázek. Aplikace bude vyvíjena pro společnosti, které platí své zaměstnance podle výkonu. Tato aplikace bude obsahovat pro každého uživatele seznam prací, které zadává jejich nadřízený s možností je přijmout nebo odmítnout. Zároveň se bude dít kontrolovat seznam již vykonaných zakázek pro vybrané období. Aplikace bude upozorňovat na nové pracovní příležitosti a konečné termíny již přijatých zakázek.

Základní literární prameny:

GARGENTA, M. Learning Android. Sebastopol, Calif.: O'Reilly, 2011. 245 p. ISBN 14-493-9050-1.
LEE, W.,M. Beginning Android application development. Indianapolis, IN: Wiley Pub., 2011. 428 s. ISBN 978-111-8087-800.
MARTIŠEK, D. Algoritmizace a programování v Delphi. Brno: Littera, 2007. 230 s. ISBN 978-8-85763-37-9.

UJBÁNYAI, M. Programujeme pro Android. Praha: Grada, 2012. 187 s. ISBN 978-80-247-3995-3.

VELTE, A., T. VELTE a R. ELSENPETER. Cloud Computing: praktický průvodce. Brno: Computer Press, 2011. 344 s. ISBN 978-80-251-3333-0.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

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

Abstrakt

Bakalářská práce zabývající se návrhem mobilní aplikace. Ta je navržena univerzálně pro všechny operační systémy. Naprogramovat se dá v jakémkoliv jazyce pro mobilní aplikace. Návrh je přizpůsoben pro jazyky Java a Swift.

Abstract

Bachelor's thesis focused on design of a mobile application. The application itself is designed universally for all operating systems. It can be programmed in any programming language for mobile applications. The design itself is based on Java and Swift.

Klíčová slova

mobilní aplikace, iOS, Android, nákupní objednávka

Key words

mobile application, iOS, Android, purchase order

Bibliografická citace

FILIPOVIČ, Roman. *Aplikace pro správu zakázek* [online]. Brno, 2020 [cit. 2020-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/127501>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Petr Dydowicz.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V, dne

.....

Podpis studenta

Poděkování

Touto cestou bych chtěl poděkovat vedoucímu práce Ing. Petru Dydowiczovi, Ph.D. za pomoc a cenné rady při zpracování bakalářské práce.

OBSAH

ÚVOD	10
1 CÍL A METODIKA PRÁCE	11
2 TEORETICKÁ VÝCHODISKA PRÁCE.....	12
2.1 Mobilní aplikace.....	12
2.2 Typy aplikací.....	12
2.2.1 Webová aplikace	13
2.2.2 Nativní aplikace.....	13
2.2.3 Hybridní aplikace	14
2.2.4 Srovnání typů aplikace	14
2.3 Operační systémy	15
2.3.1 Android.....	15
2.3.2 iOS.....	17
2.3.3 Windows Phone.....	18
2.4 Programování	19
2.4.1 Programovací jazyk.....	19
2.4.2 Java.....	20
2.4.3 Swift	21
3 ANALÝZA SOUČASNÉHO STAVU	23
3.1 Prostředí Google Play.....	23
3.2 Prostředí App Store	23
3.3 Prostředí Microsoft Store	24
3.4 Výběr prostředí pro aplikaci.....	24
3.5 Monetizace aplikace	25
3.5.1 Placená aplikace	26
3.5.2 Mikrotransakce	26

3.5.3 Reklama.....	26
3.6 Konkurenční aplikace.....	26
3.6.1 SAP.....	27
3.6.2 Kissflow.....	27
4 VLASTNÍ NÁVRH ŘEŠENÍ.....	28
4.1 Přihlašovací obrazovka.....	29
4.2 Registrace	30
4.3 Hlavní obrazovka	32
4.4 Aktivní zakázky.....	34
4.5 Dokončené zakázky.....	36
4.5.1 Nákupní objednávka.....	37
4.6 Historie zakázek	39
4.7 Volné zakázky	40
4.8 Nová zakázka	43
4.9 Formulář pro VP.....	44
4.10 Zprávy	46
4.11 Databáze	49
4.12 Přihlašování a registrace.....	51
ZÁVĚR.....	54
SEZNAM DOSTUPNÝCH ZDROJŮ	55
SEZNAM OBRÁZKŮ	58
SEZNAM TABULEK.....	59
SEZNAM GRAFŮ	60

ÚVOD

Menší firmy nebo živnostníci v současné době nemají široké spektrum mobilních aplikací vypomáhajících jim z účetnického hlediska. Většina těchto aplikací je navrhována pro velké korporáty a podle nich jsou také vyvíjeny a nastaveny ceny. Tato aplikace by měla pomoci menším subjektům a ulehčit jim jejich denní fungování. Navržená aplikace bude obsahovat vše potřebné ke správě a přijímání zakázek, včetně vystavování nákupních objednávek zaměstnancům nebo klientům.

1 CÍL A METODIKA PRÁCE

Aplikace, které mají za úkol správu zakázek jsou velmi obsáhlé a kvůli tomu i drahé. Žádné z nich také nejsou optimalizovány pro mobilní zařízení. Proto hlavním cílem této práce bude návrh aplikace takové, která by mohla sloužit i pro menší podniky s nižšími obraty. Aplikace bude jednoúčelová a bude sloužit ke správě zakázek uvnitř firmy. Při plné implementaci ji bude využívat každá vrstva firemní struktury.

Cílem aplikace samotné je vytvoření možností správy zakázek pro takové firmy, které pomocí zakázek vyplácejí zaměstnance. Součástí aplikace bude uživatelské rozhraní s přehledem všech aktivních zakázek pro nejnižše postavené zaměstnance ve firmě. Při splnění podmínek dané zakázky ji potvrdí a hotový produkt nebo službu odevzdá nadřízenému. Nadřízený produkt zkontroluje a potvrdí v aplikaci hotovou zakázku. Součástí jednotlivých zakázek bude také místo pro komunikaci. Aplikace bude obsahovat měsíční nebo týdenní přehled za účelem pohodlného vyplnění výkazu práce zaměstnancem. Další součástí bude nabídka zakázek, které vydávají nadřízení a zaměstnanci si zde vybírají zakázky, které jim nejvíce vyhovují.

V práci navrhnu celé uživatelské rozhraní aplikace pro jednotlivé obrazovky a tyto obrazovky popíšu. Zároveň se zaměřím na navržení vývojového diagramu, popisujícího veškeré náležitosti potřebné k vytvoření aplikace. Nakonec zhodnotím návrh a jeho implementaci ekonomicky.

2 TEORETICKÁ VÝCHODISKA PRÁCE

V teoretických východiscích této práce vyjádřím důležité informace k samotnému zpracování aplikace. Popíšu jednotlivé operační systémy, na které je možnost aplikaci vyvíjet a zároveň programovací jazyky k nim potřebné. Vysvětlím jednotlivé jazyky a druhy programování, které k nim náleží.

2.1 Mobilní aplikace

Mobilní aplikací se nazývá softwarový program, vyvíjený na jakémkoliv mobilní zařízení (nejčastěji chytré telefony a tablety). Tyto aplikace využívají pro své fungování hardwaru samotného zařízení, proto je potřeba zajistit jejich vzájemnou kompatibilitu. Navíc se ještě přidává skutečnost, že kompatibilní aplikace a zařízení jedné značky nezaručuje kompatibilitu značek jiných. K zaručení správné funkčnosti je třeba již od začátku programování počítat a průběžně opravovat chyby pro všechna zařízení, pro která je aplikace určena. Je možné také využívat některé softwarové služby mobilního zařízení, zde je ovšem potřeba brát v potaz, že každý uživatel musí souhlasit s využíváním dat jeho zařízení ke správnému fungování navržené aplikace. [1]

Funkcí mobilních aplikací je nespočetně. Většinou se ale jednotlivé aplikace navrhuje tak, aby fungovaly za jednotným cílem. Pro příklad mohou být aplikace využívány za účelem navigace, zábavy, sportu, zvýšení produktivity a spousty dalších.

Od normálních aplikací se mobilní liší hlavně přizpůsobením a funkcionalitou. Ovšem většina desktopových aplikací se nedá využívat na mobilních zařízeních bez nějaké úpravy. Zároveň je u těchto zařízení potřeba využívat funkce pro dotek, vzhledem ke skutečnosti, že v současnosti se především využívají zařízení s dotykovými obrazovkami.

Jednotlivé aplikace musí mít již od návrhu zvoleno na jaké typy zařízení se bude zaměřovat. Podle toho je potřeba vybrat typ aplikace a jejího vývoje.

2.2 Typy aplikací

Pro vývoj aplikace je potřeba nejdříve projít jednotlivé typy aplikací, k čemuž se pojí výhody a nevýhody jejich využití. Aplikace se dělí podle toho, na jaké platformy

(resp. operační systémy) jsou vyvíjeny. V souladu s tím se také vybírá samotný programovací jazyk využitý na implementaci návrhu.

2.2.1 Webová aplikace

Tyto aplikace jsou svojí funkčností podobné hybridním aplikacím. Webové aplikace využívají prohlížeč k jejich fungování. Většinou jsou psány v kombinaci programovacích jazyků HTML5, JavaScript a CSS.

Webové aplikace využívají minimum paměti zařízení. Kvůli tomu, že všechny databáze jsou uloženy na serveru, uživatelé k nim mají přístup z téměř jakéhokoliv zařízení s možností připojení k internetu. Vzhledem k této skutečnosti, pokud má zařízení špatné připojení k internetu, tak i samotná uživatelská zkušenost klesá. [2]

Obsah těchto aplikací je tedy na uživatelském zařízení považován za tzv. „obal“. Většina načítaných dat se nachází na serveru a uživateli se pouze zobrazují. Díky tomu rychlost této aplikace nezáleží na výkonu samotného zařízení, ale pouze na již zmíněné kvalitě připojení k internetu.

2.2.2 Nativní aplikace

Nativní aplikace jsou vyvíjeny pouze pro zařízení jednoho operačního systému, takže jsou nativní právě pro jednu platformu nebo zařízení. Aplikace tohoto typu, které jsou navrženy na jednu platformu, nemohou být použity na žádnou jinou platformu. Mají za cíl být navrženy přímo pro svoji cílovou skupinu. Vyskytují se pouze na obchodu s aplikacemi platformy, na kterou jsou navrženy. [2]

Hlavní výhodou tohoto typu je vysoký výkon. K tomu pojící se dobrá uživatelská zkušenost díky tomu, že vývojář využívá k vývoji stejné uživatelské rozhraní. Dále mají tyto aplikace dostupné mnohem větší výběr API (rozhraní pro programování aplikací). [2]

Nevýhodami těchto aplikací je převážně potřeba vývoje další oddělené aplikace, pokud se vývojář rozhodne rozšíření aplikace na další platformu. K tomu se pojí také oddělená podpora a údržba. Z těchto důvodů bývá většinou cena nativních aplikací větší. [2]

Programovacích jazyků, které se dají využít na vývoj nativních aplikací je spousta. Za účelem této práce budu brát v úvahu nejvyužívanější jazyky pro jednotlivé platformy:

- **Swift** pro iOS aplikace
- **Java** pro Android aplikace
- **Net** pro Windows Phone aplikace

2.2.3 Hybridní aplikace

Hybridní aplikace jsou vyvíjeny s využitím multiplatformních webových technologií (např. HTML5, CSS, JavaScript). Stejně jako u webových aplikací, většina procesů je vykonávána na straně serveru a uživatel opět má na svém zařízení „obal“.

Spolu s webovými aplikacemi také sdílí výhody a nevýhody. Navíc ale hybridní aplikace mají nedostatek ve výkonu a rychlosti. Tím, že jsou vyvíjeny na všechny platformy také mají horší optimalizaci. Některé aplikace mohou mít též problém jak ve vizuálu, tak ve funkcích, vzhledem k tomu, že každá platforma funguje na jiném základě. Z tohoto důvodu může mít perfektně fungující aplikace na jedné platformě problémy a nekonzistence na platformě jiné. Ovšem mají jednotný kód a díky tomu mohou mít jednotný vývoj, podporu a údržbu. [2]

2.2.4 Srovnání typů aplikace

Pro srovnání jednotlivých typů aplikací je třeba vzít do úvahy několik metrik.

Co se ceny týče, nejdražší na vývoj a chod aplikace bude bez pochyby nativní aplikace. Tento typ bude nejspolehlivější, nejrychlejší a nepříjemnější pro samotného uživatele a jeho zkušenost s aplikací. Hybridní a webové aplikace se cenově pohybují v podobné kategorii, závisí tedy na konkrétním návrhu a jeho ceně. Je potřeba brát do úvahy, že podle ceny vývoje aplikace se často odvíjí její cena pro koncové uživatele. Proto je při výběru typu potřeba určení cílové skupiny a analýzy finančních možností jejich uživatelů.

Nativní aplikace zároveň nabízí skvělou optimalizaci a podporu pro uživatele. Co se týče řešení chyb nebo podpory při dotazech uživatelů, nativní aplikace obvykle zastiňují zbylé typy.

Tab. 1: Srovnání typů aplikací

	Webová aplikace	Hybridní aplikace	Nativní aplikace
Náklady	Velmi nízké náklady z důvodu jednotného kódu	Nižší náklady	Vysoké náklady při vývoji pro více než jednu platformu
Výkon	Střední výkon, který závisí na serveru a jeho vytíženosti		Vysoký výkon z důvodu ukládání přímo do paměti zařízení
Monetizace	Monetizace tvořena především reklamou na webu nebo předplatným	Nákupy přímo v aplikaci, reklamy a nákup samotné aplikace	
Údržba/Aktualizace	Jednotné pro všechny platformy. Jednodušší, s možností výskytu chyb mezi platformami		Probíhá pro každou platformu zvlášť. Více časově náročné, vyšší kvalita

2.3 Operační systémy

Operační systémy pro mobilní zařízení jsou softwary, které umožňují mobilním zařízením spuštění a chod veškerých aplikací a programů. Mobilní operační systémy jsou téměř totožné k operačním systémům desktopových zařízení. Operační systém se načítá ihned po startu zařízení. Většina operačních systémů má již do svého základu integrovány funkce, jako je kalkulačka, poznámkový blok a další. Rozsáhlost těchto funkcí se liší u každého operačního systému. [3]

Mobilní operační systémy také nabízejí přístup na internet skrze zabudovaný modem v zařízení. Využívá připojení k internetu podle mobilní sítě, kterou využívá skrze SIM kartu. Součástí operačního systému také bývá prohlížeč, který se také liší podle typu systému. [3]

2.3.1 Android

Operační systém Android je nejvyužívanějším systémem na světě. Má bohatou historii vývoje a s ním i zkušeností pro své uživatele. Tento operační systém je vyvíjen

na bázi Linux, primárně pro zařízení s dotykovými displeji. Jedná se o open source systém, každý má tedy možnost měnit jeho kód podle své potřeby. Tohoto využívají značky mobilních telefonů k vývoji jejich vlastního operačního systému, který je založen na Androidu, kompletně ale mění uživatelské rozhraní a další funkce. [4]

Android je silný operační systém, který podporuje velké množství aplikací převážně pro chytré telefony a tablety. Je vyvíjen pro zařízení, která využívají hardware architektury ARM, jež je využívána pro svůj výkon a nízkou spotřebu. [4]

V roce 2005 byl Android koupen společností Google. Cíle vývoje ovšem zůstaly stejné a nadále fungují jako dvě oddělené společnosti, ovšem již pod jednou značkou. [5]

Co se zabezpečení týče, vývojáři neustále pracují na zlepšování. Současně začíná zabezpečení Androidu na aplikační vrstvě v podobě zabudovaného štítu proti malware. Tento štít se neustále zlepšuje, a to kvůli partnerství se společností Google a využívání jejich databází. Díky tomu aplikace Google běží na systémech Android lépe a zabezpečení je o to lepší. Zařízení s tímto operačním systémem jsou každý den skenované na přítomnost nežádoucích nebo škodlivých aplikací. Zároveň s tím se skenuje i celý obchod s aplikacemi a škodlivé aplikace jsou ihned smazány. I přes to, že Android a s ním i Google Play obchod nabízí volnost, co se výběru aplikací týče, neustále se snaží zvyšovat zabezpečení. [6]

Android se snaží získat dominantní postavení oproti konkurenci tím, že nabízí operační systém s transparentním designem. Kdykoliv aplikace vyžaduje citlivá data, systém okamžitě žádá potvrzení uživatele, že tato data může aplikaci nabídnout. Bez toho se aplikace nebo její část ani nespustí. [6]

Partnerství se společností Google má kromě zabezpečení spoustu dalších pozitivních aspektů. Pevně se jedná o navržení mobilních Google aplikací tak, aby byly optimalizovány pro tento operační systém. To zaručuje jak rychlost aplikací, tak kvalitní podporu a aktualizace.

Vytvoření aplikace pro využívání veřejnosti je u Androidu jednoduché, jak to jen jde. Stačí zapnout Google Play aplikaci a přejít do sekce Vytvořit aplikaci. Následně si stačí přečíst podmínky, zvolit distribuci, cenu aplikace a potvrdit. [7]

2.3.2 iOS

Operační systém iOS pro mobilní zařízení byl vytvořen a je nadále vyvíjen společností Apple Inc., která zároveň vytváří hardware pro tento systém. Jedná se o druhý nejvyžívanější systém na světě jednotně vyvíjen pro mobilní zařízení iPhone, iPad a iPod. Aktualizace probíhají pravidelně, jednou ročně, pro všechny podporované zařízení. Každý rok také firma vydává seznam starých zařízení, která odcházejí z jejich podpory, a tedy nové aktualizace nebudou s těmito zařízeními kompatibilní. [8]

Na rozdíl od systému Android není iOS open source. Právě naopak, společnost Apple Inc. si vývoj systému nechává pouze pro sebe a nedává jiné společnosti práva na jeho využívání. Sama společnost ovšem vyvíjí spoustu aplikací, jako je hudba, tvůrce dokumentů a další.

Se systémem iOS získávají uživatelé také možnost využívání systému iCloud. Jedná se o úložiště dat, které se dá využívat na všech zařízeních používajících iOS, tedy i na desktopových zařízeních. Každý uživatel má 5 GB bezplatného úložiště s možností dokoupení rozšíření. Apple zaručuje maximální bezpečnost uložených dat, ale také se zaměřuje na jejich přehlednost a jednoduchost uživatelského rozhraní. [9]

Zabezpečení zařízení je jedním z hlavních taháků tohoto systému. Vývojáři systému iOS neustále zlepšují a vyvíjí lepší zabezpečení systému a samotných zařízení. Bezpečnost systému je navržena tak, aby útoky byly zastaveny dříve, než dojde k ohrožení dat nebo zařízení. U nejnovějších zařízení dokáže systém předejít i útokům, které degradují verzi systému. [10]

Co se týče aplikací, ty berou vývojáři jako nejvíce kritické elementy pro bezpečnost. Kvůli tomu Apple neustále vyvíjí vrstvy protekce, aby byla všechna zařízení zabezpečena. Na svém obchodě – AppStore mají pouze zkontrolované aplikace, kterým mohou uživatelé věřit a nemusí se obávat útoků. Co se týče jiných aplikací, u zařízení využívající iOS nelze stahovat jakékoliv aplikace mimo oficiální obchod. Apple tedy tímto krokem prioritizuje bezpečnost před širšími možnostmi aplikací pro uživatele. [11]

Systém iOS je plně uzavřený a nedovoluje uživatelům žádné změny v systému. To je hlavním rozdílem od systému Android, který má politiku kompletně opačnou. Vývoj

aplikací na tento systém může být složitější a firma Apple může požadovat v kódu aplikace změny směrem k větší transparentnosti, aby si vývojáři byli jisti její bezpečností.

2.3.3 Windows Phone

Co se Windows Phone týče, tento operační systém byl veřejnosti spuštěn roku 2010. Hlavním účelem bylo napodobení funkčnosti iOS, ovšem za levnější cenu. Od začátku se Microsoft, který tento systém vyvíjel, snažil o podobné cíle, jako iOS. Systém se stavěl na základu transparentnosti mezi zařízeními tak, aby bylo zajištěno maximální pohodlí pro uživatele. [12]

Ovšem Windows Phone se nikdy ve své historii ani nepřiblížil tržnímu podílu podobnému k iOS. Microsoft se snažil vyvíjet operační systém na svůj vlastní hardware, a tedy mobilní zařízení své značky. Většina z těchto zařízení ale nebyla veřejností přijata velmi dobře a v prosinci roku 2019 Windows oznámil konec vývoje a podpory pro zařízení Windows Phone. [13]

I přes to, že byl ukončen vývoj operačního systému, aplikace na něj nijak zasaženy nejsou a je možnost stále vytvářet nové. Je potřeba brát do úvahy, jak rychle se bude snižovat marketingový poměr mobilních zařízení Windows Phone po oznámení této skutečnosti.

Operační systém Windows Phone jako takový se dá považovat středem mezi optimalizací systému iOS a otevřeností Androidu. Jedná se o systém „closed-source“, tudíž jeho kód je součástí vlastnického práva a jakékoliv jeho úpravy jsou nelegální. Transparentnost mezi zařízeními je možná pro všechna zařízení se stejným typem operačního systému. Tedy například zařízení s operačním systémem Windows Phone 8.1 a stolní počítač se systémem Windows 8.1 mohou sdílet téměř všechny fungující aplikace. [14]

Nahrávání aplikací na obchod platformy Microsoft Store je podobné systémům iOS s rozdílem v zabezpečení. Microsoft nemá potřebu tak velkého zabezpečení, jako uzavřené systémy iOS, proto vyvíjení aplikací na tuto platformu není tak složité, co se transparentnosti týče.

2.4 Programování

Programování je proces vytváření algoritmu a jeho zápisu do programovacího jazyka tak, aby počítač mohl tento algoritmus provést. Samotná definice toho, co algoritmus je, může být složitá. Dalo by se říct, že se jedná o postup provedení nějakého procesu vyjádřeného tak, aby tomu počítač rozuměl. Jako příklad algoritmu se dá použít například binární vyhledávání. To se často hodí v různých programech. Jedná se o proces, při kterém se půlí celkový rozsah vyhledávání a určuje se postupně v jaké půlce se objekt vyhledávání nachází. Jako příklad může sloužit vyhledávání v encyklopedii, kterou bychom otevřeli v půlce a zjistili, jestli se slovo, které hledáme, nachází v první nebo druhé půlce (podle toho, na jaké písmeno začíná). Jakmile zjistíme, v jaké půlce se nachází, opět ji půlíme, a to až dokud nenalezneme výsledek. [15] Existuje velké množství programovacích jazyků, vždy je ale potřeba mít nejdříve cíl a cestu k jeho provedení. Algoritmy popisují řešení problému v podobě takového zápisu, aby ho mohl počítač přečíst a aplikovat. Počítač ale umí vykonávat aktivity pouze v binární soustavě a psát kód v této soustavě by bylo pro člověka příliš složité, proto byly vyvinuty programovací jazyky. [16]

Vzhledem k tomu, že počítač pracuje v binární soustavě, jeho výkon je lehce optimalizován. Jediné chyby, které mohou v programu nebo aplikaci vzniknout jsou chyby v samotném kódu tohoto programu. Další výhodou je rychlost počítačů. Jejich kapacita je téměř neomezená a stále se vyvíjí lepší hardware. Průměrný počítač v dnešní době sečte dvě čísla za nanosekundu. [17]

2.4.1 Programovací jazyk

Jak již bylo zmíněno, počítač umí operovat pouze v binární soustavě a kvůli tomu byly vytvořeny programovací jazyky. Úkolem programovacích jazyků je převod matematických operací a jejich výsledků do binární soustavy. Každý programovací jazyk má svá jasně daná pravidla a syntaxe. Jazyky jsou velmi podobné jejich funkcionalitou, protože myšlenka jednotlivých algoritmů musí být totožná v každém jazyce. Programovací jazyky se dělí na vyšší a nižší. [18]

Nižší programovací jazyky jsou takové, které dávají své příkazy přímo procesoru, který tyto příkazy vykonává tak, jak jsou napsány. Tím, že se příkazy dávají přímo

procesoru se stává kód velmi obsáhlý a složitý. Těchto druhů jazyků se ovšem používá převážně kvůli tomu, aby programátor mohl využívat všechny funkce procesoru a systému. Dokonce i takové, které by s vyšším programovacím jazykem využívat nemohl. Hlavním reprezentantem tohoto druhu je programovací jazyk **Assembler**. [19]

Vyšší programovací jazyky jsou problémově orientované. Kód je lehčí a tzv. čistější. Jejich struktura se skládá z logických operací a algoritmů, které kompilátor dále převádí do zdrojového kódu, aby instrukce mohl počítač vykonat. Do této skupiny patří i jazyk **C**, který sice pracuje přímo s pamětí a tedy hardwarem, pořád se ale řadí do jazyků problémově orientovaných vzhledem ke struktuře kódu, ve kterém je psán. [19]

Jednotlivé programovací jazyky jsou také hodnoceny pro jejich kvalitu a vývoj. Hodnocení se skládá z mnoha faktorů a pochází od programátorů z celého světa. Popularitu a oblíbenost jazyků hodnotí tzv. **TIOBE Index**. [20]

Tab. 2: Srovnání programovacích jazyků

Pořadí	Programovací jazyk	Hodnocení
1	Java	17,78 %
2	C	16,33 %
3	Python	10,11 %
4	C++	6,79 %
5	C#	5,32 %
6	Visual Basic .NET	5,26 %
7	JavaScript	2,05 %

2.4.2 Java

Java je všeobecně účelový programovací jazyk, který umožňuje chod více algoritmů nebo funkcí zároveň. Jedná se o objektově orientovaný programovací jazyk. Kvůli skvělému využití operační paměti dosahuje velkého výkonu. Program nebo aplikace napsány v tomto programovacím jazyce mohou být kompilovány přes všechny platformy, které Javu podporují, bez potřeby jakýchkoliv dalších úprav. Například je možné napsat Java program na operačním systému UNIX a spustit ho na systému Microsoft Windows bez potřeby úpravy zdrojového kódu. Tohoto Java dosahuje z důvodu kompilace celého programu do dalšího jazyka jménem **bytecode**. Formát tohoto jazyka je navržen tak, aby nebyl závislý na žádné platformě, ale každá ho mohla číst. [21]

Výhody Javy:

- **Objevení chyb** – Kvůli okamžité kompilaci kódu se dá objevit chyby zdrojového kódu již v počátku;
- **Stabilita** – Interpret nemá možnost kód upravovat ani spouštět, to vykonává virtuální stroj, proto má jazyk velkou stabilitu;
- **Rychlost** – Startování aplikace může být pomalejší, ale přenos dat a samotná kompilace je velmi rychlá;
- **Minimální zranitelnost kódu** – Vzhledem k tomu, že se kód kompiluje, možnost jeho napadení při přesunu mezi virtuálními stroji je minimální. [22]

Při programování v Javě je doporučeno používat prostředí pro vývoj **Eclipse**, a to z důvodu jeho flexibility a možnosti rozšíření o doplňky z jiných programovacích jazyků, jako PHP, Python a další. Pro využívání Eclipse je třeba nainstalovat i prostředí JRE (Java Runtime Environment). [23]

Další důležitou věcí při samotném programování je vytvoření emulátoru. Není doporučeno testovat funkčnost aplikace na svém běžícím zařízení, kvůli různým chybám, které mohou nastat. Zároveň s tím mají zařízení Android často jiné verze operačního systému a k tomu ještě nadstavby systémů od značky, která telefon prodává. V emulátoru lze testovat spousty těchto zařízení včetně tabletů a hodinek. Co se týče navrhované aplikace, je potřeba ji testovat minimálně na dvou virtuálních zařízeních – nejnovější verzi Androidu pro mobilní telefony a pro tablety. Před započítím programování samotné aplikace je třeba vyzkoušet na emulátoru nějaký program se základními příkazy a cykly, pro ověření správné funkčnosti. [23]

Vytvoření samotného projektu se provádí ve výše zmíněném prostředí Eclipse, které obsahuje vše potřebné, včetně průvodce pro založení projektu. Prostředí obsahuje také „debug“ možnost pro průběžnou kontrolu, jestli se v aplikaci nevyskytuje kritická chyba, a to ještě před samotným spuštěním aplikace v emulátoru. [23]

2.4.3 Swift

Programovací jazyk Swift je jazykem pro telefony, servery, desktopy a další zařízení, které umožňují spouštění kódu. Jedná se o rychlý a bezpečný interaktivní jazyk,

který aplikuje dlouholeté zkušenosti firmy Apple. Právě z tohoto důvodu je to ideální jazyk na programování aplikací, určených pro produkty Apple. [24]

Je navržen pro co největší přehlednost a čistotu kódu. Zároveň je uživatelské rozhraní tohoto jazyka navrženo tak, aby se jazyk co nejpohodlněji četl a opravoval. Je navržen, stejně jako všechny produkty Apple, pro maximální zabezpečení. Proměnné jsou v kódu inicializovány před jejich použitím a paměť je řízena automaticky. To sice omezuje možnosti pro vývojáře, přináší to ovšem jiné kvality. Tento jazyk také nabízí velké množství knihoven, se kterými mohou pracovat všichni vývojáři. [25]

I tento programovací jazyk má prostředí pro zjednodušení, ukládání knihoven a dalších funkcí. Toto prostředí se nazývá Xcode. Funguje podobně jako Eclipse pro Android zařízení. Uživatel programující v prostředí Xcode si může propojit účet s „Apple Developer Program“, díky kterému získá různé benefity a také možnost kontaktování podpory při vytváření aplikace. [26]

3 ANALÝZA SOUČASNÉHO STAVU

V této části práce se budu zabývat analýzou trhu a konkurenčních aplikací. Správná analýza je pro takový projekt nezbytná, a to z důvodu, že je v současné době složitější prosadit se na trhu mobilních aplikací, než tomu bylo dříve. Zanalyzuji tři primární prostředí pro mobilní aplikace – Google Play, App Store, Microsoft Store. Zvážím také faktory výběru správného prostředí a strategie monetizace pro danou aplikaci.

3.1 Prostředí Google Play

Google Play je v současné době dominantní prostředí z pohledu počtu aplikací. V prvním kvartálu roku 2020 na této platformě bylo 2 570 000 aplikací. [28] Dominance Google Play je primárně způsobena tím, že je vyvíjena výhradně na mobilní zařízení s operačním systémem Android, který je open-source. Vývojáři tedy preferují volnost a větší možnosti systému Android. Zároveň má operační systém Android podle statistik z roku 2020 dominantní podíl na trhu – 73,52 %, což se také musí brát do úvahy při výběru platformy. [27]

3.2 Prostředí App Store

App Store je obchod s aplikacemi vytvořen výhradně pro zařízení s operačním systémem iOS a macOS, tedy produkty společnosti Apple Inc., vyrábějící široké množství elektroniky. V této práci se ovšem zaměřím pouze na mobilní zařízení, a tedy operační systém iOS. Jejich podíl na trhu, který činí 25,33 % není nijak závratný, ovšem Apple cílí své produkty na určitou cílovou skupinu. Pro zařízení iPhone se našlo velké využití ve vyšší střední vrstvě obyvatelstva a ve větších firmách a nadnárodních korporátech. [27]

Systém iOS je poměrně uzavřený a nedovoluje uživatelům instalovat programy, které nebyly získány z obchodu App Store. Zároveň každá aplikace musí projít schvalovacími procesy před publikací na App Store. Tímto zamezuje přístup infikovaných aplikací k uživatelům, zároveň ale ztěžuje proces jejich implementace do obchodu App Store.

3.3 Prostředí Microsoft Store

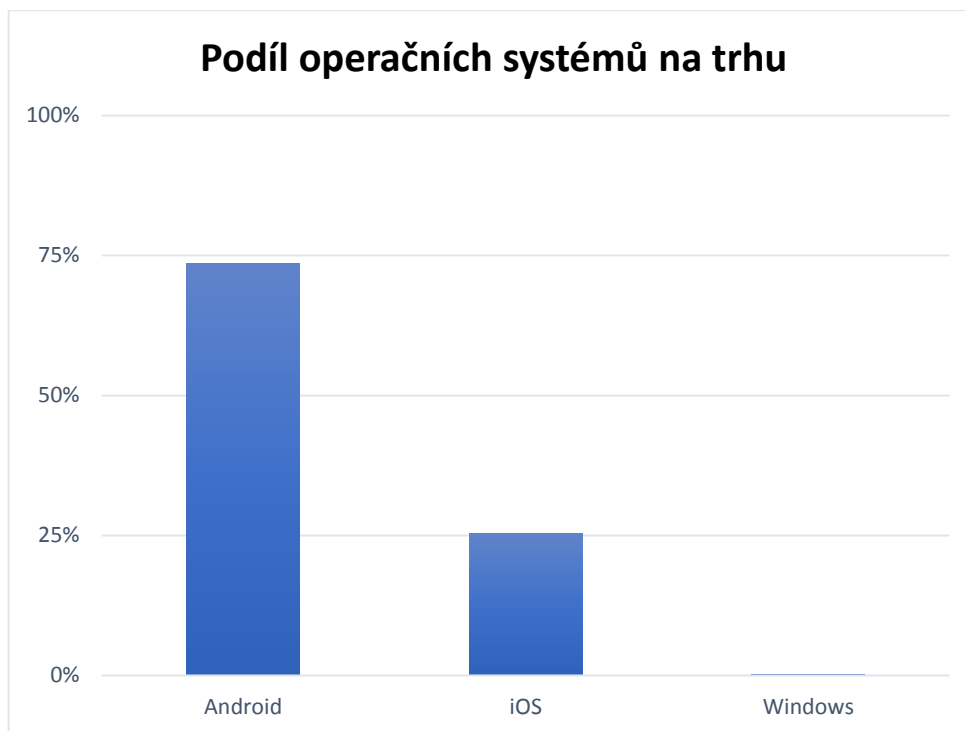
Microsoft Store je obchodní síť vyvíjena primárně pro operační systém Windows. Zabývá se produkty jako jsou telefony, tablety a PC. Stejně jako Android nejsou cíleny výhradně na jednu značku, ovšem mají partnerství s třetími stranami, jako jsou Acer, Dell, HP a další. Podíl Microsoft Store na trhu je minimální a aplikací ve čtvrtém kvartálu roku 2019 zde bylo 669 000, což je téměř čtvrtina oproti Google Play. [28]

Microsoft Store také prošel velkými změnami. Dříve se na mobilní telefony používal pouze obchod Windows Store, který ale vzhledem k neoblíbenosti operačního systému Windows upadal. Proto firma přišla s univerzálním řešením Microsoft Store, fungujícího na všech zařízeních využívajících operační systém Windows.

V mobilním průmyslu má Microsoft Store mizivých 0,14 % podílu na trhu, ovšem díky orientaci na více zařízení by tento obchod mohl být využitelný na trhu pro firmy. [27]

3.4 Výběr prostředí pro aplikaci

Pro výběr správného prostředí je potřeba zvážit několik primárních faktorů. Prvním z nich je podíl jednotlivých prostředí na trhu. Zde jednoznačně vyhrává Google Play se svým 73,52% podílem. Nejde ovšem brát v potaz pouze tento faktor a spousta uživatelů této platformy ani vyvíjenou aplikaci nevyužije. Na rozdíl od Google Play je 0,14% podíl Microsoft Store na trhu alarmující a je potřeba určit, zda by na tomto trhu vůbec byla dostatečná poptávka po tomto produktu. [27]



Graf 1: Podíl operačních systémů na trhu

Dalším faktorem při výběru prostředí bude cílová skupina pro aplikaci. Je potřeba si z předchozí analýzy určit počet uživatelů na cílovém trhu a zkombinovat tyto informace se skupinami, na které chceme cílit. Dominantní postavení Google Play nám ukazuje na velkou všeobecnou poptávku po produktu, ovšem většina uživatelů operačního systému Android je nižší sociální vrstva, která z většiny nebude mít zájem o aplikaci vyvíjenou pro firmy. Zde se naskýtá výhoda pro App Store, jehož uživatelé by mohli mít větší zájem o nabízený produkt, vzhledem k tomu, že primárním odběratelem mobilních telefonů řady iPhone je vyšší střední vrstva nebo korporáty, které tyto telefony zapůjčují svým zaměstnancům.

3.5 Monetizace aplikace

Pro všechna řešená prostředí se nabízí stejné typy a možnosti monetizace. Cílovému zákazníkovi je potřeba nabízet i nějaké služby navíc, jako je online podpora, aktualizace aplikace a její zabezpečení. Z těchto důvodů je potřeba aplikaci zpeněžit.

3.5.1 Placená aplikace

Monetizace způsobem jednorázové platby při pořízení aplikace je hlavním trendem u firemních aplikací tohoto typu. Pro zvýšení zisku firem s větším množstvím zaměstnanců, kteří benefitují z této aplikace, je možné zavést platbu podle velikosti firmy nebo počtu instalací aplikace. Například základní balíček pro 50 instalací, po jehož naplnění nabízet rozšíření o další počet instalací za příplatek.

3.5.2 Mikrotransakce

Mikrotransakce se především používají u her na těchto obchodních platformách, ovšem může se zvážit i implementace pro tento typ aplikací. Zpravidla se mikrotransakce vyskytují u aplikací zdarma ke stažení a postupně se snaží uživatele přimět ke koupi předmětů nebo prémiových funkcí pomocí menších poplatků. Není doporučeno skrýt za poplatek některou z primárních funkcí aplikace. Tím by mohla klesnout její úspěšnost a uživatelé by se mohli rozhodnout pro konkurenční produkt.

3.5.3 Reklama

Reklama se pro zisk finančních prostředků z aplikace může využít v podstatě jakkoliv. Je možné ji zobrazit na úvodní obrazovce při načítání aplikace, na přihlašovacím okně nebo kdekoli v aplikaci. Neměla by ovšem nijak narušovat funkčnost aplikace pro uživatele (např. vyskakující okna s reklamou).

Tato možnost monetizace se dá také zkombinovat s mikrotransakcí. Vývojáři aplikací využívají reklam pro zavedení určitého prémiového účtu, jehož hlavním účelem je skrytí všech reklam po zaplacení menšího poplatku. Zisk z tohoto poplatku ale musí být vyšší, než kolik průměrně vydělá aplikace na reklamě pro jednoho uživatele. Při balíčku pro firmy se také dá využít příplatek pro odstranění reklam pro všechny uživatele.

3.6 Konkurenční aplikace

V oboru podnikových aplikací pro správu zakázek není velká konkurence. Na trhu se nenachází velké množství aplikací se stejným využitím, jako má tento návrh. Ty, které na trhu najdeme, jsou převážně komplexnějšími aplikacemi ke správě středních nebo velkých firem pro větší množství zaměstnanců, dodavatelů a distribučních cest. Tyto

konkurenční aplikace jsou také řešeny pro vnější využití, tedy pro správu zakázek prodeje zboží. Aplikace navržená v této práci bude řešena i pro interní účely a správu zakázek mezi jednotlivými články výrobního procesu nebo procesu služeb.

3.6.1 SAP

SAP je společností, která v oboru obchodních procesů svým podílem pokrývá většinu trhu. Jedná se o obsáhlou aplikaci s využitím ve spoustě sférách pro výrobu produktů i poskytovatele služeb. Tato společnost postupně také svůj produkt zlepšuje a vydává nové a lepší verze. Nejnovější verzí je produkt jménem **mySAP**. Tento produkt je rozdělen do několika balíčků pro různé odvětví a využití, např. marketing, analýza, management, správa skladu a další. [29]

Produkty společnosti SAP jsou kvalitní, ale tomu je také úměrná cena. Průměrná cena za balíček služeb se pohybuje okolo 80 000 Kč. Tak velké výdaje si malé nebo některé střední podniky nemohou dovolit obětovat pouze pro jednu službu nebo produkt a zde by měla mít výhodu aplikace navržená v této práci. Vzhledem k přesnému cílení na menší nebo střední podniky není potřeba nadsadit cenu tak vysoko, jako je tomu u společnosti SAP. [30]

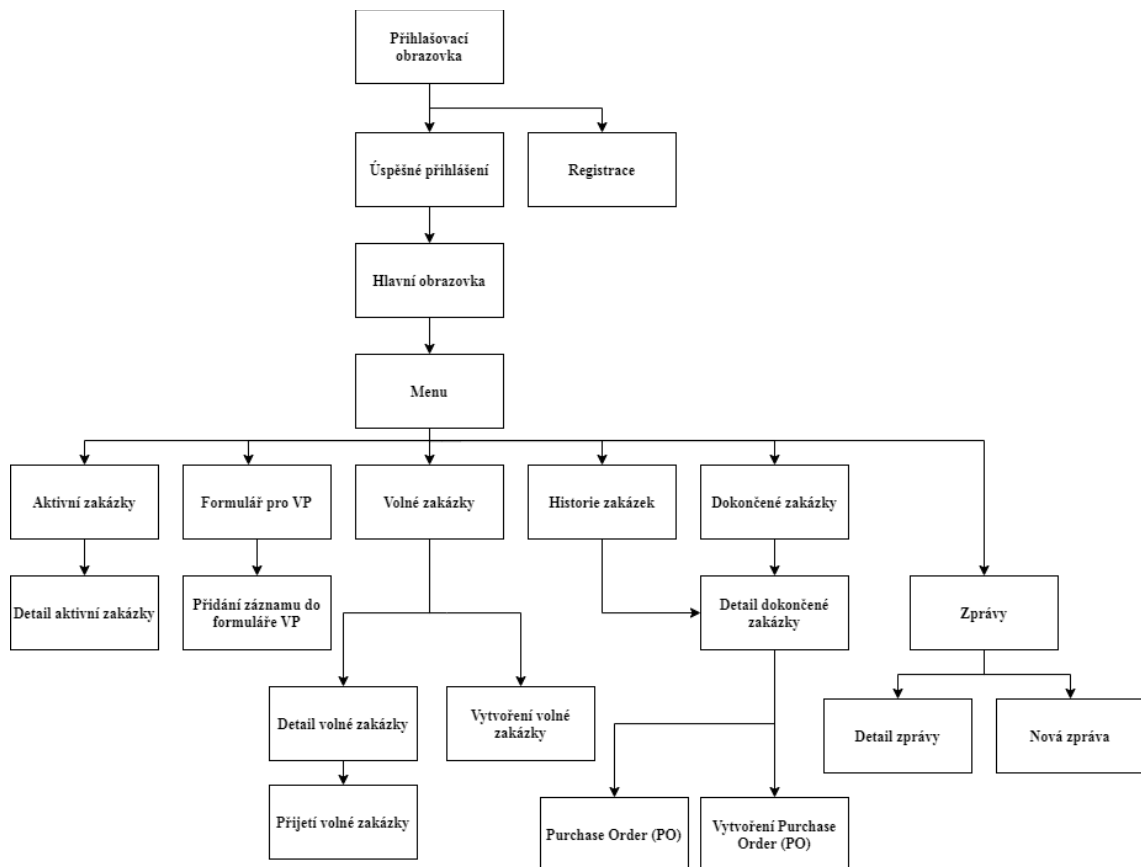
3.6.2 Kissflow

Kissflow je produkt zaměřený spíše na uživatelské rozhraní. Příjemně vypadá a jednoduše se užívá, ovšem nejde v něm provádět komplexnější operace. V ohledu správy zakázek vypadá vše uspořádaně, zakázky jsou v aplikaci seřazeny do časové řady. Chybí zde ovšem možnosti, jako je převedení zakázky na jiného člověka, komentáře nebo jakýkoliv způsob komunikace uvnitř zakázky. Kissflow má stejnou cílovou skupinu jako aplikace v této práci, ovšem cena je stále příliš vysoká. Základní a zároveň nejlevnější plán tohoto produktu stojí \$8 280 ročně. [31] Za tuto cenu je nabízeno více služeb, které nemusí být menší firmou využity. V navržené aplikaci budou řešeny zmíněné nedostatky aplikace Kissflow se zaměřením na konkrétní zakázky, jejich strukturu a správu. [32]

4 VLASTNÍ NÁVRH ŘEŠENÍ

Návrh aplikace je tvořen pro obecné využití na kteroukoliv ze zmíněných platform. Není tedy vázána na jeden operační systém. Přestože programovací jazyky pro každý systém mají určité rozdíly, návrh je tvořen tak, aby měla aplikace široké využití napříč platform. Avšak primární systém pro tento návrh bude Android, vzhledem k jeho velkému podílu na trhu, a tedy největší pravděpodobnosti využití aplikace právě na této platformě. K tomu vázající se programovací jazyk se použije Java, vzhledem ke skvělé kompatibilitě k Androidu.

Před započítím programovacích prací je potřeba udělat si obecný obrázek ohledně toho, jak bude aplikace fungovat, jaké obrazovky na sebe navazují a s jakými informacemi budou pracovat. K tomu nám nejlépe poslouží diagram.



Obr. 1: Diagram obrazovek

Většina z obrazovek bude zobrazovat jiné výsledky, a to podle pracovní pozice, kterou přihlášený zaměstnanec ve firmě zastává. Je tedy potřeba aktivní propojení a

komunikace s databází, ve které budou přístupy pro každého zaměstnance popsány a uloženy.

4.1 Přihlašovací obrazovka

Při spuštění aplikace se uživatelům ihned objeví přihlašovací obrazovka, která se bude skládat z polí pro vyplnění. Jedná se o pole ID firmy, přihlašovací jméno a heslo. ID firmy si může každá firma zvolit jaké bude chtít, bude se jednat o 5 čísel, které firma zadá při zakoupení softwaru. Součástí této koupě tedy bude komunikace s poskytovatelem a zadání tohoto ID do databáze. Jedná se o triviální činnost, kterou může provést jakýkoliv zaměstnanec, například člen zákaznické podpory. Přihlašovací jména si také firma může volit sama, ovšem musí počítat s tím, že jména musí být unikátní. Pokud se stane, že některý ze zadávaných parametrů nebude odpovídat záznamu v databázi, uživateli se ukáže červená chybová hláška na spodu obrazovky. Tato hláška bude upozorňovat na chybné ID firmy nebo špatné přihlašovací údaje.

Tlačítko přihlásit naváže komunikaci s databází a ověří správnost zadaných informací. Text registrace se také chová jako tlačítko, které ovšem nemá žádné ohraničení. Celkový dojem z přihlašovací obrazovky má být decentní, žádné zbytečné dekorace a podobné neúčinné věci.

The image shows a login form on a grey background. At the top, the word "Zakázkovna" is written in blue. Below it are three input fields: "ID podniku", "Přihlašovací jméno", and "Heslo". At the bottom left is the text "Registrace" and at the bottom right is a green button with the text "Přihlásit".

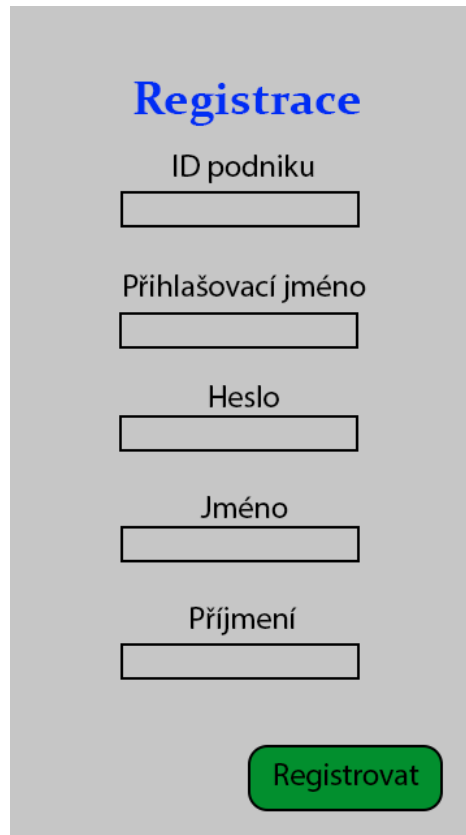
Obr. 2: Přihlašovací obrazovka

4.2 Registrace

Po stisknutí textu registrace na přihlašovací obrazovce se zobrazí formulář pro registraci. Při registraci samotné se zadává ID podniku, přihlašovací jméno, heslo, jméno, příjmení a email, ke kterému se bude potřeba posouvat dolů, vzhledem k rozložení a omezení velikosti obrazovky. Všechny tyto parametry se také zapisují do databáze uživatelů, ovšem až poté, co je registrace schválena nějakým nadřízeným. Toto je vytvořeno za účelem bezpečnosti proti útočníkům na firmu, kteří by se snažili zjistit ID podniku a mohli by tak vytvářet nové profily zaměstnanců, kteří neexistují. Kvůli tomu se některému z nadřízených na hlavní obrazovce ukáže tato nová registrace a ten potvrzuje, jestli se opravdu jedná o nového zaměstnance.

Při zadání špatných údajů se postupuje stejně jako u přihlašovací obrazovky, kdy se v její spodní části červeně zobrazí chyba. Chybová hláška bude obsahovat informaci o

duplikátu některého z parametrů a bude odkazovat na technickou podporu, pokud uživatel zapomněl heslo.

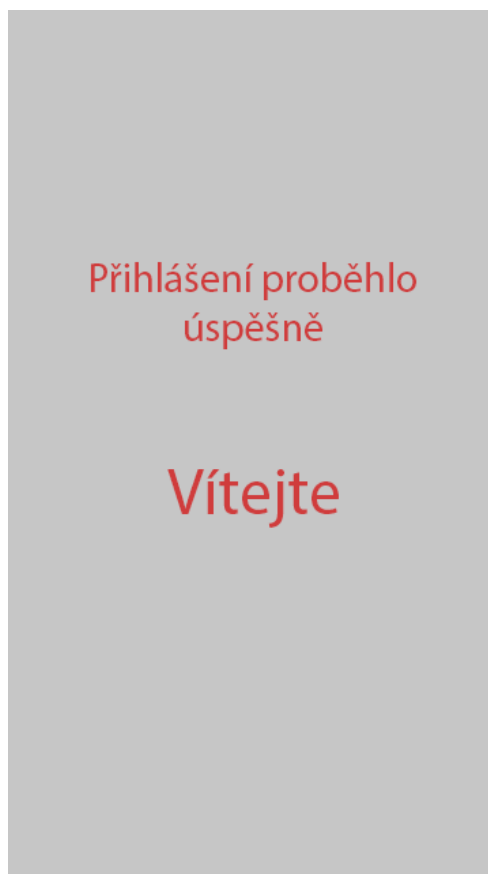


The image shows a registration form titled "Registrace" in blue text. Below the title are five input fields, each with a label above it: "ID podniku", "Přihlašovací jméno", "Heslo", "Jméno", and "Příjmení". At the bottom right of the form is a green button with the text "Registrovat".

Obr. 3: Registrace

Uvítání

Součástí této obrazovky bude pouze přivítání uživatele a potvrzení úspěšného přihlášení. K tomu se využije efektu přechodu slábnutí tak, aby obrazovky na sebe plynule navázaly. Přivítání se zobrazí na 4 sekundy, poté přejde na hlavní obrazovku.



Obr. 4: Uvítání

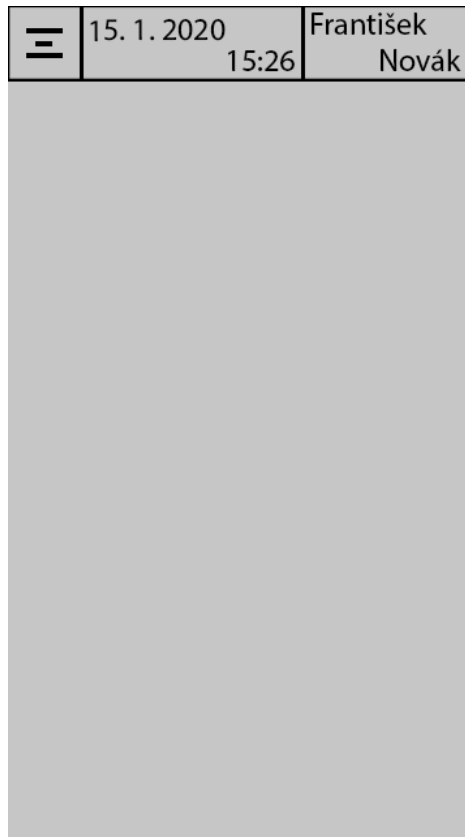
4.3 Hlavní obrazovka

Od hlavní obrazovky a jejího uživatelského rozhraní se bude odvíjet celá aplikace. Tato obrazovka již musí komunikovat s databází a přenést do záhlaví jméno a příjmení uživatele, který je zrovna přihlášen. Zároveň se ze serveru do záhlaví bude načítat čas. Je důležité, aby čas byl serverový, ne uživatelský. Tato aplikace je vyvíjena tak, aby brala co nejméně informací ze zařízení uživatele. Zároveň serverový čas bude jednotný a přesnější než čas zařízení jednotlivých uživatelů aplikace.

Ikona tří čar pod sebou bude v celé aplikaci vyjadřovat ikonu hlavního menu. Tato ikona bude interaktivní, stává se z ní tedy tlačítko, které otevírá menu. Nenačítá ovšem novou obrazovku, pouze se rozbálí na obrazovce stávající. Rozbaluje se postupně přechodem, neobjevuje se ihned. To by působilo špatným dojmem pro uživatele.

Pod záhlavím se budou na hlavní obrazovce načítat aktivity daného uživatele. Nebude se jednat přímo o pracovní aktivity spojené se zakázkami. Jsou to organizační

aktivity pro výše postavené zaměstnance firmy. Převážně je toto místo vyhrazeno pro potvrzování nových registrací, které se zde budou zobrazovat.



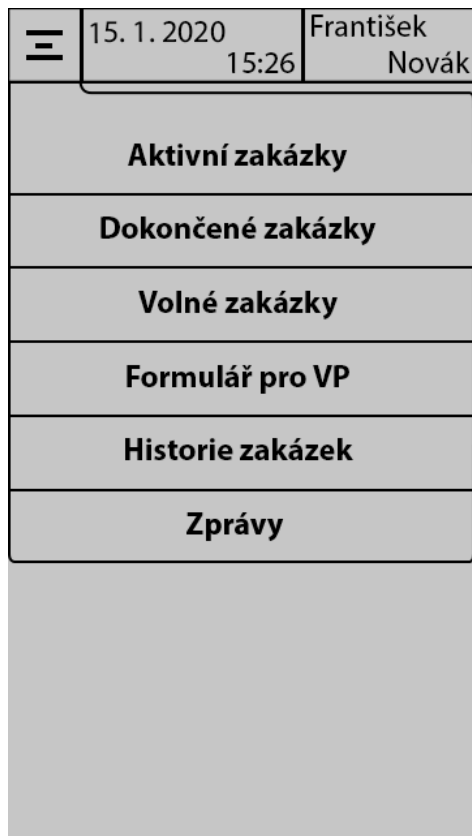
Obr. 5: Hlavní obrazovka

Menu

Další důležitou částí aplikace bude samotné menu, které se otevírá po stisknutí ikony se třemi vodorovnými čarami. Menu se rozbaluje postupně přechodem a překrývá pouze hlavní část obrazovky. Záhlaví s datem a jménem zůstává viditelné i po rozbalení.

Samotné položky v menu jsou navrženy pro základní potřebu využívání aplikace pro správu zakázek. Ovšem samotné položky v menu by neměly být naprogramovány fixně. Uživatelské rozhraní by mělo být dynamické a je navrženo tak, aby byla možnost přidat další položky do menu. Při přidávání musí být zachováno stejné rozhraní i formátování jako při již existujících položkách, kvůli konzistenci.

Jednotlivé objekty v menu jsou aktivní zakázky, dokončené zakázky, volné zakázky, formulář pro VP, historie zakázek a zprávy. Tyto objekty jsou tlačítka, při jejichž stisknutí se načte nová obrazovka s návazností podle diagramu.



Obr. 6: Menu

4.4 Aktivní zakázky

Při výběru možnosti Aktivní zakázky z menu se načte nová obrazovka, u které zůstává záhlaví, do kterého se přidává nadpis aktivních zakázek. Pod tímto záhlavím se budou zobrazovat veškeré přijaté zakázky, které ještě nebyly odevzdány. Tato obrazovka slouží k přehledu, je tedy u každé aktivní zakázky napsáno datum přijetí v levém a datum odevzdání v pravém horním rohu. Následně se ve spodních rozích nachází ID zakázky nebo objednávky a vyjádření stavu zakázky. Tento stav může být zobrazen v procentech nebo jiných číselných jednotkách. Toto číslo se nenačítá samo, ale zadává ho uživatel v detailu aktivní zakázky. Tato funkce tedy není nutná, ale může se využívat převážně při větších zakázkách z důvodu přehlednosti. Uprostřed je nadpis aktivity samotné zakázky.

Jednotlivé zakázky se budou dát posunovat pohybem prstu po displeji nahoru a dolů. Záhlaví se neposunuje a stále zůstává nahoře obrazovky. Zakázky jsou seřazeny podle nejbližšího data odevzdání, tudíž je opět potřeba komunikace se serverem. Toto seřazení se zajistí programovacím cyklem.

Na každou zakázku se dá klepnout prstem pro přenesení do detailu aktivní zakázky. Poté se uživateli zobrazí detail dané zakázky společně s poznámkami a místem pro zadání jejího stavu.

☰	15. 1. 2020 15:26	František Novák
Aktivní zakázky		
14. 1. 2020 01-158654	Překlad	20. 1. 2020 68%
15. 1. 2020 01-158655	Překlad	22. 1. 2020 6/18

Obr. 7: Aktivní zakázky

Detail aktivní zakázky

Na obrazovku Detail aktivní zakázky se uživatel dostane při klepnutí na libovolnou aktivní zakázku. Obsahuje stejné záhlaví, co se týče data, času a jména, ovšem ikona menu se změní na ikonu šipky zpět. Touto šipkou se uživatel dostane zpět na aktivní zakázky. Zároveň se uživatel přesune zpět po kliknutí na tlačítko Zpět na svém zařízení.

Součástí záhlaví se stane název aktivity, kterou si uživatel rozklikl, a její ID v levém dolním rohu. Dále datum přijetí a datum (případně datum s časem) odevzdání. Tato doplňující data jsou také jména zaměstnanců, od kterých byla zakázka přijata a komu se odevzdává nebo kdo kontroluje její správnost. Uprostřed obrazovky se nachází rámeček, kam uživatel vypisuje kolik z dané zakázky má hotovo. Tato informace může být procentuální nebo číselná s jednotkou. Ve spodní půlce obrazovky je možnost poznámek nebo nařízení při vykonávání zakázky. Většina zakázek je individuálních, proto by zde

mělo být upřesněno jaká práce se vykonává, bezpečnostní pokyny nebo jiné nařízení. Tyto poznámky jsou vyplňovány nadřízeným, který zakázku zadával do systému. Uživatel sledující detail aktivní zakázky nemůže tyto poznámky upravovat. Pokud se naskytou nějaké dotazy nebo nepřesnosti, může kontaktovat zaměstnance, který mu zadal zakázku.

Při zadání kompletního zpracování v okně uprostřed se na spodní části obrazovky objeví zelené tlačítko dokončit. Toto tlačítko se zde nenachází, dokud uživatel nevyplní kompletní dokončení zakázky. Jestliže je objem práce na zakázce měřen v časových jednotkách, uživatel vyplní dobu zpracování a stiskne tlačítko dokončit. Tato akce přenese uživatele do dokončených zakázek, kam se přesune také tato zakázka. V databázi se změní v tabulce zakázka pole „Aktivní“ na 0.

←	15. 1. 2020 15:26	František Novák
Překlad		
01-158654		
14. 1. 2020		Do: 20. 1. 2020
Přijato od: Jan Horký		Odevzdání: Jan Horký
100 %		Doba zpracování:
Poznámky:		2,5 h
Dokončit		

Obr. 8: Detail Aktivní zakázky

4.5 Dokončené zakázky

Obrazovka dokončených zakázek se bude lišit podle toho, kterému uživateli bude zobrazena. Zaměstnanec firmy, který zakázky pouze vykonává, uvidí na této obrazovce

seznam dokončených zakázek, které ještě nebyly zaneseny do výkazu práce. Jednotlivé zakázky budou obsahovat datum přijetí a odevzdání, ID zakázky a její aktivitu. Následně v pravém dolním rohu bude tlačítko „PO“ (Purchase order) pro nákupní objednávku. Text tohoto tlačítka je vyznačen zeleně, pokud již byla nákupní objednávka vydána uživatelem, který zakázku zadával. Naopak červená barva textu se vyskytne tehdy, pokud nákupní objednávka ještě vydána nebyla.

Pro nadřízené ve firmě nebo uživatele zadávající zakázky do systému bude uživatelské rozhraní vypadat podobně, uvidí navíc jméno osoby, která zakázku vykonala. Toto jméno se vyskytne nad provedenou aktivitou. Dále bude mít tlačítko „PO“ jinou funkčnost pro zmíněné uživatele.

☰	15. 1. 2020 15:26	František Novák
Dokončené zakázky		
9. 1. 2020 01-158653	Překlad	12. 1. 2020 PO
9. 1. 2020 01-158652	Kontrola	11. 1. 2020 PO

Obr. 9: Dokončené zakázky

4.5.1 Nákupní objednávka

Při prokliku na tlačítko „PO“ se načte obrazovka nákupní objednávky pro zakázku, u které se tlačítko nacházelo. Záhloví obrazovky bude obsahovat tlačítko zpět, které

odkazuje na dokončené zakázky, nadpis „Nákupní objednávka – PO“ s číslem dané zakázky v levém dolním rohu záhlaví.

Jestliže se uživateli, který zakázky vykonával, zobrazuje tlačítko „PO“ zeleně, po klepnutí se mu objeví vyplněné boxy objednatel a dodavatel. Box dodavatel se vyplňuje automaticky, pomocí dotazů z databáze o tomto uživateli. Objednatel bude firma nebo osoba, která zakázku zadávala a tato osoba zároveň vyplňuje tento box. Stejně tak vyplňuje další box, ve kterém se nachází ID-PO, datum, aktivita, množství, jednotka, cena za jednotku a celková cena. V zápatí stránky se nachází tlačítko přeposlat na e-mail, jež automaticky pošle uživateli nákupní objednávku na e-mail, který zadal při registraci. Pokud má tlačítko červené, nákupní objednávka ještě nebyla vytvořena a tlačítko je neaktivní.

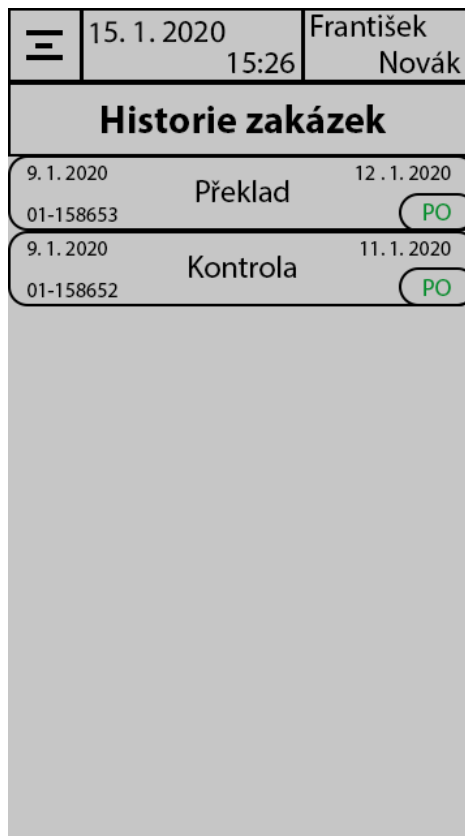
Pro zadavatele zakázek jsou funkční obě tlačítka. Po dokončení prací na zakázce se zakázka objeví na této obrazovce s červeným tlačítkem „PO“. Při kliknutí na něj se zobrazí prázdná obrazovka pro nákupní objednávku a uživatel ji může vyplnit. V tomto případě se vedle tlačítka Přeposlat na e-mail v levé části zápatí objeví tlačítko uložit, které ukládá změny vytvořené na nákupní objednávce. Obrazovka obsahuje kontroly, aby žádné pole nezůstalo prázdné, v takové situaci nepůjde objednávka uložit. Po uložení objednávky tlačítko „PO“ zezelená pro obě strany, které se účastnily zakázky. Zadavatel může objednávku stále měnit i po jejím vyplnění a může si ji také přeposlat na e-mail. Boxy pro vyplnění objednatele a dodavatele jsou fixní velikosti, ovšem obsah může přesahovat jejich vymezení. V tomto případě se bude dát posunovat pohybem po boxu. Box s informacemi o objednávce má fixní velikost a nijak se neposunuje. Texty datum, aktivita, množství, jednotka, cena za jednotku a cena celkem jsou součástí šablony, vytvořené při programování, zobrazují se tedy i při vyplňování nové objednávky.

←	15. 1. 2020 15:26	František Novák
Nákupní objednávka - PO <small>01-158653</small>		
Objednatel:	Dodavatel:	
[]	[]	
ID-PO: 01-158653		
Datum:	12. 1. 2020	
Aktivita:	Překlad	
Množství:	1200	
Jednotka:	slovo	
Cena za jednotku:	0,25 Kč	
Cena celkem: 300 Kč		
<input type="button" value="Přeposlat na e-mail"/>		

Obr. 10: Nákupní objednávka

4.6 Historie zakázek

Historie zakázek bude rozhraním velice podobná dokončeným zakázkám. Obsahuje stejný seznam zakázek, ovšem do této obrazovky se zakázky budou přidávat samy, po zanesení do výkazu práce. Při uzávěrce výkazu práce se automaticky přesunou zakázky z dokončených do historie. Vzhledem k tomu, že se do výkazu práce dává ID z nákupních objednávek, všechny tlačítka „PO“ budou zelená a budou aktivní k prokliku na detail nákupní objednávky.

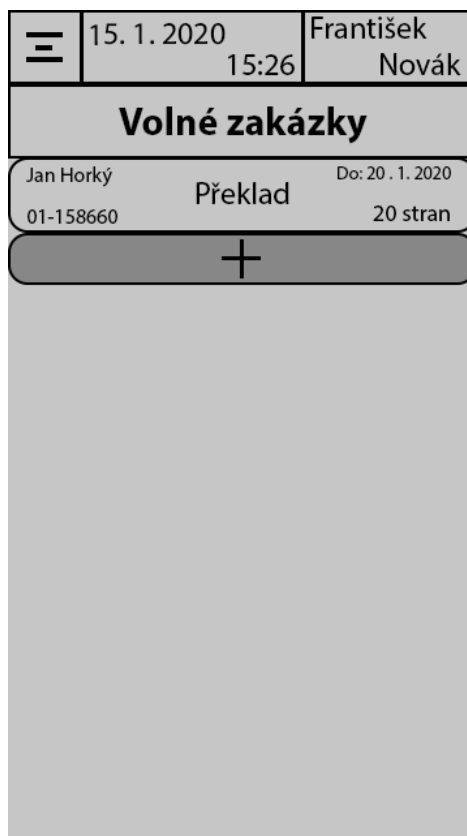


Obr. 11: Historie zakázek

4.7 Volné zakázky

Obrazovka volných zakázek bude nejvyužívanější obrazovkou. Dá se chápat jako takový „trh“ zakázek, kde se budou zakázky nabízet a ostatní uživatelé je budou moci přijmout. Uživatelské rozhraní bude konzistentní s předchozími. Znovu se zde nachází seznam zakázek, u kterých jsou základní informace s možností poklepnání pro detail. Informace na této stránce se snaží popsat zakázku co nejlépe, aby uživatel hledající zakázku mohl již při náhledu projevit zájem nebo naopak a nemusel přecházet na detail každé zakázky. Náhled na zakázku bude obsahovat jméno a příjmení zadávajícího, datum dokončení, rozsah zakázky, její aktivitu a ID.

Opět se bude rozhraní měnit podle přístupů uživatelů. Zaměstnanci zadávající zakázky budou mít na této obrazovce z důvodu přehlednosti pouze jejich volné zakázky. Ty budou seřazovány podle nejbližšího data dokončení sestupně. Navíc tito uživatelé budou mít na konci jejich objednávek (jako poslední položku) tlačítko s ikonou pro přidání. Toto tlačítko je přeneseno na obrazovku vytvoření nové zakázky.



Obr. 12: Volné zakázky

Detail volné zakázky

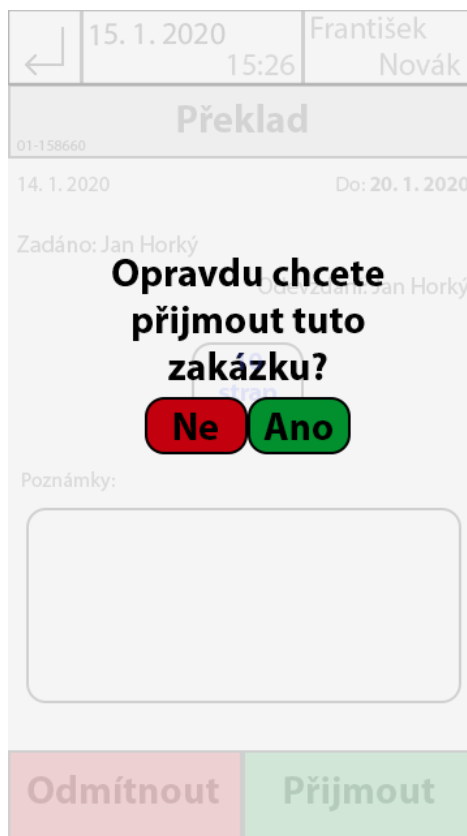
Při klepnutí na volnou zakázku je uživatel přesměrován na obrazovku Detailu volné zakázky, obsahující ID a její aktivitu v záhlaví. Rozhraní je podobné jako u Detailu dokončené zakázky, aby celá aplikace působila konzistentně a byla jednoduchá na orientaci. Z většiny není pro uživatele interaktivní, spíše informativní. Jsou zde veškeré informace vyplněny ze záložky nová zakázka. V zápatí se nacházejí tlačítka na přijetí nebo odmítnutí zakázky.

Pokud se uživatel rozhodne zakázku přijmout, zobrazí se potvrzení, jestli zakázku opravdu chce nebo klepl na přijetí omylem. Při odmítnutí zakázky se nic dále nezobrazuje, uživatel je přemístěn zpět na seznam volných zakázek. Odmítnutá zakázka se již uživateli nezobrazuje. Uživatel si tedy může sám vybrat, jestli zakázky, o které nemá zájem, si v aplikaci nechá a bude je ignorovat nebo je bude postupně odmítat pro větší přehlednost.

←	15. 1. 2020 15:26	František Novák
Překlad		
01-158660		
14. 1. 2020		Do: 20. 1. 2020
Zadáno: Jan Horký		Odevzdání: Jan Horký
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;"> 10 stran </div>		
Poznámky:		
Odmítnout	Přijmout	

Obr. 13: Detail volné zakázky

Pakliže uživatel chce zakázku přijmout, klikne na tlačítko pro přijetí zakázky. Následně je dotázán, jestli opravdu chce zakázku přijmout. Po potvrzení pomocí poklepání na tlačítko „Ano“ je uživatel přenesen na obrazovku aktivních zakázek, ve které se nově objeví i právě potvrzená zakázka.



Obr. 14: Potvrzení zakázky

4.8 Nová zakázka

Nové zakázky budou k dispozici pouze těm, jejichž účty mají v databázi příslušnou hodnotu a tím povolení k tomuto úkonu. Nové zakázky jsou navrženy tak, aby jejich vytvoření bylo co nejjednodušší a nejrychlejší. V záhlaví uživatel vyplní aktivitu, při poklepnutí na záhlaví zmizí text „Aktivita“ a uživatel vyplní pro kterou aktivitu zakázku vytváří, stejně tak vyplní ID v levém dolním rohu záhlaví. Datum zadání a jméno zadávajícího se vyplní automaticky při komunikaci se serverem a databází. Uživatel dále vyplní do kdy musí být zakázka hotová a komu se bude odevzdávat. V tomto případě se mu při psaní jména budou automaticky objevovat kontakty, které sedí k vyplněným znakům. Z nabídky uživatel vybere jméno a tím se také zajistí propojení k dané osobě. Následně už vyplní pouze objem práce a vyplní poznámky, ve kterých se práce upřesňuje a zadávají se doplňující informace, kterými se pracovník bude držet při práci na zakázce. Po stisknutí tlačítka sdílet se ověří, jestli žádné vyplněné pole nezůstalo prázdné, v opačném případě se červeně zvýrazní prázdné pole s upozorněním pro jeho vyplnění.

Úspěšné sdílení přenese uživatele na obrazovku volných zakázek, která již bude doplněna o zakázku právě zadanou.

←	15. 1. 2020 15:26	František Novák
Aktivita: 01-158658		
14. 1. 2020		Do: 20. 1. 2020
Jan Horký		Odevzdání: Jan Horký
16 stran		
Poznámky:		
Sdílet		

Obr. 15: Nová zakázka

4.9 Formulář pro VP

Obrazovka sloužící pro vyplňování formuláře hotových objednávek do výkazu práce. Uživateli se automaticky, voláním z databáze, vyplní hlavička se jménem, bydlištěm a datem narození. Dále sám už vyplní, za jaké období výkaz práce vyplňuje a datum zvolí až po přidání posledního záznamu, před odevzdáním hotového výkazu.

V hlavní části se nachází samotné zakázky, které za dané období uživatel zpracovával. V hlavičce jsou nejnütnější informace o zakázkách, tedy ID nákupní objednávky, datum odevzdání objednávky, aktivita, doba zpracování a celková cena zakázky. Až na dobu zpracování musí být všechny položky vyplněny. Doba zpracování může být prázdná, pokud je zaměstnanec placen jinou sazbou než hodinovou (například u sazby podle zhotovené práce se vyplní pouze celková cena). Toto záhlaví se neposunuje a zůstává stále na stejném místě obrazovky. Samotné záznamy se přidávají pomocí

tlačítka s ikonou „plus“. Po stisknutí tohoto tlačítka se objeví seznam zakázek, nacházejících se v dokončených zakázkách, ovšem pouze pokud již byla vydána nákupní objednávka. Pro přidání stačí pouze klepnout na danou zakázku. Díky tomu se vyplní všechny potřebné náležitosti a uživatel nemusí vyplňovat výkaz práce celý ručně. V samotných záznamech jde listovat a posunovat je. Tlačítko pro přidání záznamu se vždy nachází pod místem posledního přidaného záznamu.

	15. 1. 2020 15:26	František Novák		
Formulář pro VP				
Příjmení: Novák		Dat. narození: 12. 2. 1995		
Jméno: František		Adresa: Česká 1, Brno		
Výkaz za období:		Prosinec 2019		
Datum vyplnění:		30. 12. 2019		
ID-PO	Datum	Aktivita	Doba zpracování	Cena
12-554896	15. 12. 2019	Překlad	2,5 h	400 Kč
+				
Doba trvání práce: 2,5 h				
<input type="button" value="Odeslat VP e-mailem"/>		Cena celkem: 400 Kč		

Obr. 16: Formulář pro VP

Po rozbalení menu při stisku tlačítka pro přidání záznamu přejde pozadí do 8% transparentnosti a objeví se zmíněné dokončené zakázky, které již mají vydanou nákupní objednávku. Důležité je, aby se v záhlaví v tomto kroku změnila ikona menu na tlačítko zpět, bez něj by mohlo toto prostředí působit zavádějícím dojmem, při kterém není způsob se z tohoto kroku dostat zpět. Klepnutím na zakázku se přejde zpět do formuláře, zakázka se přidá pod poslední záznam v seznamu a tlačítko pro přidání nového záznamu se taktéž posune o jeden řádek. Pro přidání záznamu do výkazu práce se využívá dalších příkazů pro databázi. Při klepnutí na dokončenou zakázku se do databáze odešle příkaz pro načtení záznamů ID-PO, data dokončení zakázky, aktivity, doby zpracování a ceny.

Vyhledávání se vykoná podle parametru ID zakázky, díky čemuž databáze nalezne potřebné informace.

ID-PO	Datum	Aktivita	Doba zpracování	Cena
12-554896	15. 12. 2019	Překlad	2,5 h	400 Kč

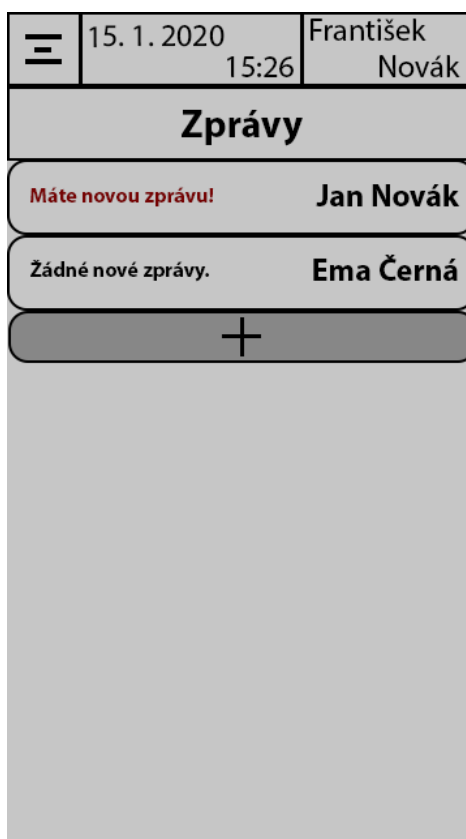
Obr. 17: Přidání záznamu do VP

V zápatí formuláře se nachází doba trvání práce a celková cena za výkaz práce. Tyto informace jsou automaticky sečteny pomocí jednoduchého volání z databáze, při kterém se tyto parametry sečtou. Dále se v zápatí nachází tlačítko pro odeslání výkazu práce na e-mail uživatele ve formátu PDF. Celé zápatí zůstává stále na stejném místě a je ukotveno ve spodní části obrazovky a není ovlivněno posunem v jednotlivých záznamech. Formulář pro výkaz práce není potřeba ukládat – ten se ukládá automaticky po přidání záznamu.

4.10 Zprávy

Poslední možností na výběr v menu jsou zprávy. Tyto zprávy se dají využívat jak k primární komunikaci, tak jako sekundární. Obrazovka zpráv bude jedinou obrazovkou dávající notifikace uživateli. Přesněji při obdržení nové zprávy se na mobilním zařízení objeví notifikace, ve které bude zmíněno jméno odesílajícího zprávy. Stejně tak jako

v samotném seznamu zpráv se nachází u konverzací jména koncového uživatele. Zprávy jsou těsně propojeny s databází, proto si tato jména i jednotlivé zprávy aplikace načítá z databáze.

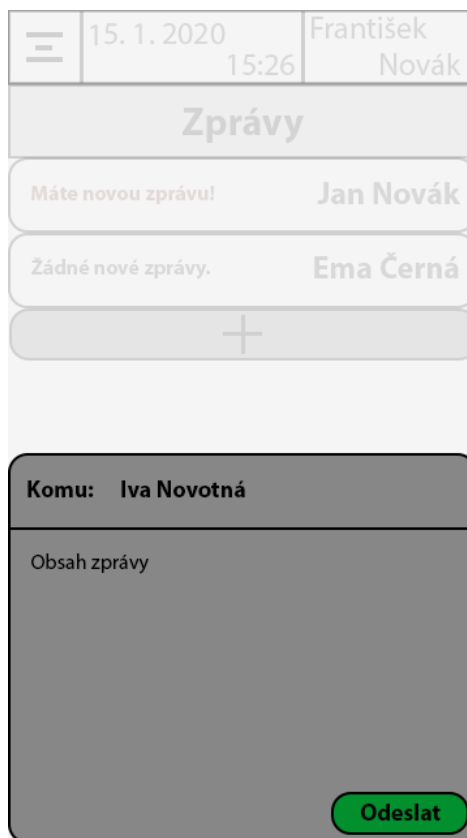


Obr. 18: Zprávy

Samotné oznamování zpráv je odlišeno barvami. Při nové zprávě je text červený pro lepší viditelnost od uživatele. Zároveň je konverzace posunuta nad všechny ostatní. Tlačítko s ikonou „plus“ slouží pro přidání nové konverzace do seznamu. Jakmile je konverzace otevřena, již není potřeba psát novou zprávu přes tlačítko, nýbrž stačí otevřít danou konverzaci poklepáním na ni.

Nová zpráva

Po stisknutí tlačítka na napsání nové zprávy současná obrazovka přejde do 15% transparentnosti a ve spodní části se objeví okno s adresátem a obsahem zprávy. V pravém dolním rohu zelené tlačítko odeslat ověřuje s databází, jestli adresát existuje v dané firmě. Při vyplňování adresáta se automaticky ukazují jména, jež se shodují se znaky, které uživatel do pole napsal.

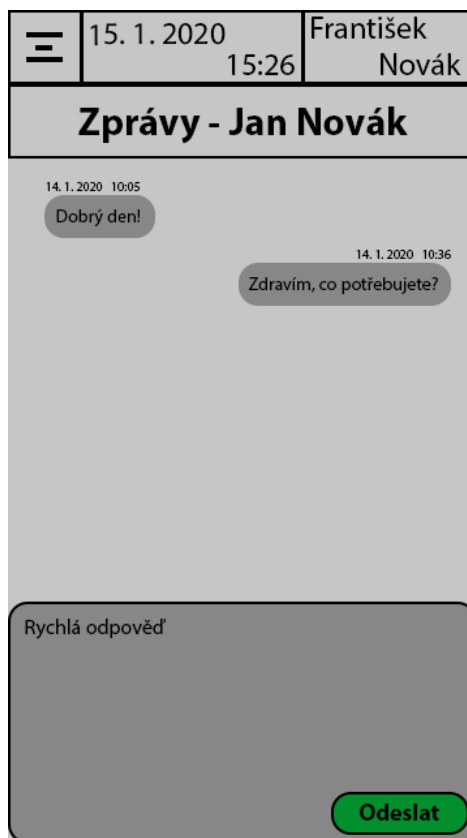


Obr. 19: Nová zpráva

Každé jméno, které se nachází v detailech jakékoliv zakázky, se stane interaktivním. Jakmile uživatel na jméno v aplikaci klepne, je přesunut právě na obrazovku nové zprávy s adresátem předvyplněným podle jména, na které poklepal. Tato interakce je vytvořena pro rychlé zprávy, kterou může být zpoždění zakázky.

Detail zprávy

Poklepáním na jakoukoliv aktivní konverzaci se uživatel přenesení na rozhraní pro danou konverzaci. Doručené zprávy se nachází nalevo, odeslané napravo. Nad každou zprávou je datum a čas odeslání. Zprávy i tato data jsou brány z tabulky zpráv v databázi. Text v objektu pro rychlou odpověď zmizí při poklepu na toto okno, slouží tak pouze k pojmenování okna pro uživatele. V záhlaví zprávy se ukazuje také jméno uživatele, se kterým je konverzace vedena. Je důležité, aby se v rozhraní toto jméno vyskytovalo.



Obr. 20: Detail zprávy

4.11 Databáze

Databáze je navržena pro účely registrací, přihlašování a ověřování pravomocí uživatelů v aplikaci. Během těchto aktivit spolupracuje s aplikací a jsou z ní volána data, stejně tak jsou díky databázi ověřována data vstupní. Databáze je navržena stálým způsobem, ovšem její data jsou spravována interním zaměstnancem firmy, který je touto aktivitou povolán.

Tabulka uživatelů je hlavní tabulka celé databáze. Obsahuje informace o samotných uživatelích. Ti nemohou po registraci informace v této tabulce měnit. Pro jejich změnu se ovšem mohou obrátit na zaměstnance firmy, který byl povolán správou databáze.

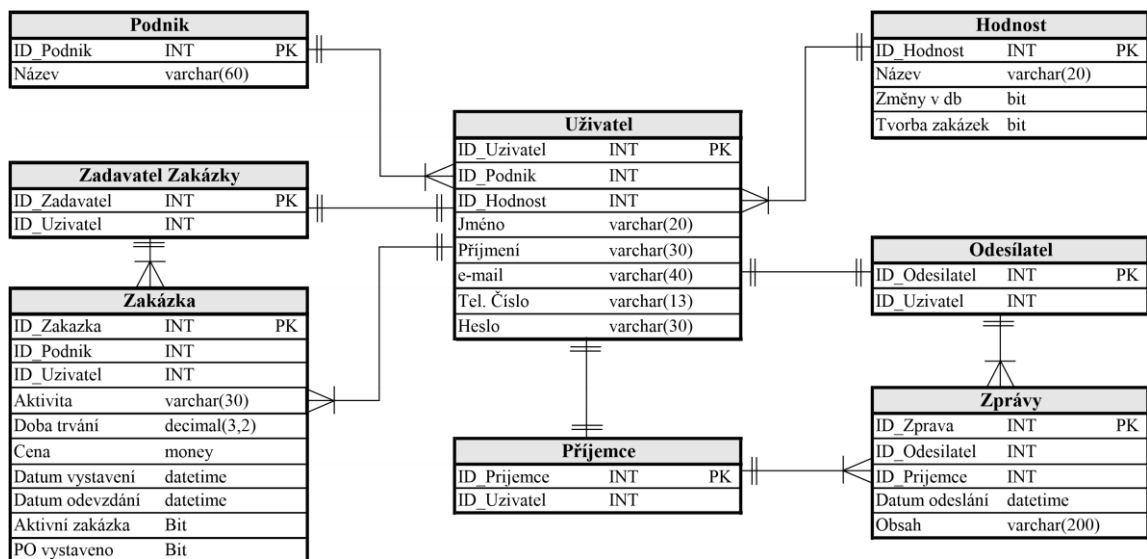
Ke každému uživateli náleží určitá hodnota, jejíž pojmenování si určuje sama firma. Hodnosti se dělí podle pravomocí zaměstnanců provádět změny v databázi a možnosti vytvářet (a tím i spravovat) jejich vlastní zakázky. Tabulka hodností je připojena pouze k tabulce uživatelů. Stejně tak jako tabulka podniku, která slouží jako číselník. Vyplní se zde pouze název podniku nebo jeho části a určí se ID, které je pouze celočíselné. Databáze

má definovány délky všech atributů, ale neověřuje jejich dodržení. To má na starosti samotná aplikace, která dává v chybových hláškách uživateli vědět, že některý z vyplněných vstupů přesáhl požadovanou délku.

Zakázky mají vlastní tabulku, která bude v databázi největší. Obsahuje všechny informace o zakázkách, včetně zadavatele a zpracovatele. Tyto informace se do tabulky zapisují postupně tím, jak zakázka prochází svým cyklem. Informaci, jestli je zakázka aktivní nebo byla dokončena zajišťuje položka „Aktivní zakázka“, která je datovým typem bit. Konkrétně tedy pokud je zakázka aktivní, je její hodnota 1. Pokud aktivní není, hodnota je 0. Obdobně s položkou „PO vystaveno“, kde tato hodnota určuje, jestli na obrazovce hotových zakázek tlačítko „PO“ svítí červeně nebo zeleně. Položka aktivita je v této tabulce omezená pouze na 30 znaků. Je to tak navrženo proto, že by měl být popis aktivity co nejkratší a delší názvy by měly problém s velikostí na zařízeních s menšími displeji. Doba trvání je kvůli definici „decimal(3,2)“ omezena na trojčíferné číslo s maximálně dvěma desetinnými místy.

Co se zpráv týče, informaci o odesílateli a příjemci si aplikace bere přes relační propojení na tyto tabulky a jejich spojení s tabulkou uživatelů. Samotná tabulka pak má pouze auto inkrementační primární klíč, datum odeslání zprávy a její obsah. Text zprávy je omezen na 200 znaků, opět z důvodu velikosti obsahu na menších mobilních zařízeních. Pokud bude chtít uživatel poslat delší zprávu, může ji rozdělit na několik menších.

Samotné relační schéma je navrženo pro maximální rychlost při vyhledávání v databázi. Vzhledem k tomu, že aplikace bude pracovat s databází poněkud často, je potřeba zajistit co největší efektivitu a rychlost těchto procesů. Je úspěšně zajištěno, aby jakýkoliv proces v aplikaci nemusel volat data přes více než 2 tabulky.



Obr. 21: Relační model databáze

4.12 Přihlašování a registrace

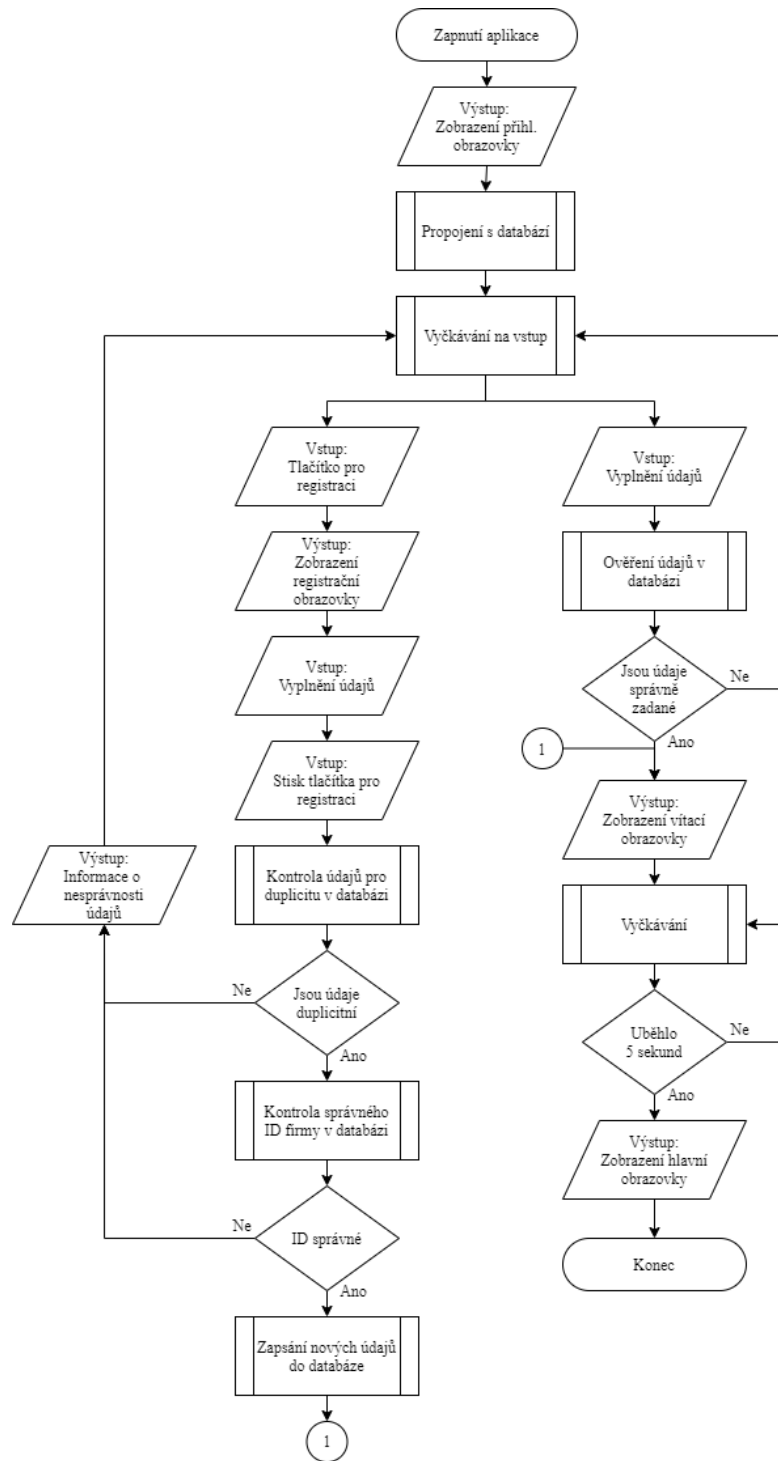
Kapitola slouží k upřesnění procesů přihlašování a registrace. Ty budou vyvolány ihned po zapnutí aplikace a zobrazení přihlašovací obrazovky, což je zároveň prvním procesem, který bude aplikace provádět. Ihned po zobrazení prostředí pro přihlášení započne pokus o propojení s databází. Proces může selhat z důvodů problému v databázi nebo internetovému připojení uživatele. V obou případech se při selhání vypíše červená zpráva „Připojení selhalo. Zkontrolujte internetové připojení nebo kontaktujte správce sítě“. Jestliže připojení k databázi proběhne úspěšně, aplikace přejde do stavu vyčkávání na vstup od uživatele. V tomto kroku se budou aktivity aplikace dělit podle těchto vstupů.

Jakmile uživatel vyplní přihlašovací údaje potřebné k přihlášení, odešle se databázi příkaz zvaný „SELECT“, do kterého se vyplní uživatelem vložená data. Jestliže databáze nalezne spojitost mezi údaji, zobrazí se přivítací obrazovka. Po pěti sekundách obrazovka mizí a objeví se hlavní obrazovka se jménem přihlášeného uživatele. Tímto procesem přihlašování končí a je také ukončeno aktivní spojení s databází. Ta ale stále čeká na další příkazy, které může aplikace vyžádat.

Při stisknutí tlačítka pro registraci, namísto vyplnění přihlašovacích údajů, aplikace načte registrační obrazovku, obsahující důležitá data o uživateli. Dále očekává vstup od uživatele v podobě vyplnění těchto dat a následného stisknutí tlačítka pro registraci. V tu chvíli vysílá aplikace podobný příkaz do databáze jako při přihlašování. V tomto případě

ověřuje duplicitu dat. Jestliže dojde k duplicitě dat v tomto kroku, na obrazovce se zobrazí upozornění o tom, že se v databázi již vyskytuje uživatel s těmito údaji. Na duplicitu se kontrolují všechny vyplněné údaje, vyjma hesla a ID firmy. Je-li výsledek tohoto hledání nulový, a tedy data neobsahují žádné duplicity, je databázi vyslán požadavek na ověření, jestli je ID firmy ve správném formátu a zda opravdu patří některé z firem. Při úspěšném ověření je do tabulky uživatelů v databázi vložen záznam o nově registrovaném uživateli pomocí klauzule „INSERT INTO“. Současně je uživatel automaticky přihlášen a přesunut na uvítací obrazovku. Dále se postupuje, jako při úspěšném přihlášení.

Podle tohoto postupu je navržen následující vývojový diagram, zachycující procesy z více technického hlediska. V tomto diagramu jsou vytvořeny procesy vyčkávání pro návraty z podmíněných výrazů. Spojovací značka je využita pro zjednodušení a zpřehlednění samotného diagramu. Vývojový diagram má nevýhodu jednotné značky pro vstup i výstup. Vzhledem k tomu je každá z těchto značek popsána, aby byl jejich význam přesně definovaný.



Obr. 22: Vývojový diagram přihlašovací obrazovky

ZÁVĚR

Cíl práce byl určen jako navržení mobilní aplikace, která bude umožňovat správu zakázek uvnitř menších ekonomických subjektů.

Provedená analýza ukázala na podíly na trhu jednotlivých prostředí, společně s výhodami a nevýhodami jednotlivých operačních systémů. Ukázalo se, že nejefektivnější by v současné době bylo aplikaci vytvářet na operační systémy Android pro prostředí Google Play, vzhledem k jejich dominantnímu podílu na trhu mobilních zařízení. Dále se pokračovalo v analýze konkurenčních aplikací, které byly popsány. Zároveň se kladlo důraz na vystižení výhod a nevýhod těchto aplikací.

V samotném zpracování bylo nejdříve vysvětlen celkový koncept aplikace, společně s diagramem jednotlivých obrazovek, které byly popsány následně. Ke konci byla navržena a vysvětlena databáze, která je pro aplikaci velmi důležitou, společně s vývojovým diagramem pro přihlašování a registrace.

Aplikace dosáhla obsáhlosti, na kterou byla původně zamýšlena. Během zpracování dokonce bylo i několik prvků přidáno navíc pro pohodlnost nebo efektivitu uživatelů, kteří ji budou používat.

SEZNAM DOSTUPNÝCH ZDROJŮ

- [1] A Beginner's Guide to Mobile Apps. Lifewire [online]. [cit. 2020-01-16]. Dostupné z: <https://www.lifewire.com/what-is-a-mobile-application-2373354>
- [2] Types of apps, different categories of mobile applications - 2020. ThinkMobiles [online]. [cit. 2020-01-16]. Dostupné z: <https://thinkmobiles.com/blog/popular-types-of-apps/>
- [3] What is a Mobile Operating System (Mobile OS). SearchMobileComputing [online]. [cit. 2020-01-16]. Dostupné z: <https://searchmobilecomputing.techtarget.com/definition/mobile-operating-system>
- [4] What is Android? Introduction of Android OS & it's Applications. ElProCus [online]. [cit. 2020-01-16]. Dostupné z: <https://www.elprocus.com/what-is-android-introduction-features-applications/>
- [5] Google bought Android 13 years ago. It was a massive bargain. AndroidAuthority [online]. [cit. 2020-01-16]. Dostupné z: <https://www.androidauthority.com/google-android-acquisition-884194/>
- [6] Android Security Center. Android [online]. [cit. 2020-01-16]. Dostupné z: <https://www.android.com/security-center/>
- [7] Upload an app – Play Console Help. Google support [online]. [cit. 2020-01-16]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/113469?hl=en>
- [8] What is iOS? All About iOS Data Recovery. Cleverfiles [online]. [cit. 2020-01-17]. Dostupné z: <https://www.cleverfiles.com/howto/what-is-ios.html>
- [9] iCloud – Apple (CZ). Apple [online]. [cit. 2020-01-17]. Dostupné z: <https://www.apple.com/cz/icloud/>
- [10] System security overview – Podpora Apple. Apple Support [online]. [cit. 2020-01-17]. Dostupné z: <https://support.apple.com/cs-cz/guide/security/sec114e4db04/1/web/1>

- [11] App security overview – Podpora Apple. Apple Support [online]. [cit. 2020-01-17]. Dostupné z: <https://support.apple.com/cs-cz/guide/security/sec35dd877d0/1/web/1>
- [12] Windows Phone – Wikipedia. Wikipedia [online]. [cit. 2020-01-20]. Dostupné z: https://en.wikipedia.org/wiki/Windows_Phone#Windows_10_Mobile
- [13] Microsoft is killing Windows Phone support on 10 December. Pocket-lint [online]. [cit. 2020-01-20]. Dostupné z: <https://www.pocket-lint.com/phones/news/microsoft/146804-microsoft-windows-phone-support>
- [14] Windows Phone 8.1 includes universal apps and lots of feature updates – The Verge. The Verge [online]. [cit. 2020-01-20]. Dostupné z: <https://www.theverge.com/2014/2/11/5400660/windows-phone-8-1-features-leaked>
- [15] MAREŠ M. a VALLA T. *Průvodce labyrintem algoritmů*. Praha: CZ.NIC, 2017, s. 23-27 [cit. 2020-01-21], ISBN 978-80-88168-22-5
- [16] 1.4. What Is Programming? – Problem Solving with Algorithms and Data Structures. Runestone Academy [online]. [cit. 2020-02-05]. Dostupné z: <https://runestone.academy/runestone/books/published/pythonds/Introduction/WhatIsProgramming.html>
- [17] What is programming. hackr.io [online]. [cit. 2020-02-05]. Dostupné z: <https://hackr.io/blog/what-is-programming>
- [18] Programming Language Definition. Webopedia [online]. [cit. 2020-02-05]. Dostupné z: https://www.webopedia.com/TERM/P/programming_language.html
- [19] 14. Programovací jazyky – Stránky k výuce informatiky. Stránky k výuce informatiky [online]. [cit. 2020-02-05]. Dostupné z: <http://www.ivt.mzf.cz/seminar/14-programovaci-jazyky/>
- [20] index | TIOBE – The Software Quality Company. Tiobe [online]. [cit. 2020-02-05]. Dostupné z: <https://www.tiobe.com/tiobe-index/>

- [21] What is Java Programming Language? - HowToDoInJava. HowToDoInJava [online]. [cit. 2020-02-05]. Dostupné z: <https://howtodoinjava.com/java/basics/what-is-java-programming-language/>
- [22] Lekce 1 – Úvod do jazyka Java. ITnetwork [online]. [cit. 2020-02-05]. Dostupné z: <https://www.itnetwork.cz/java/zaklady/java-tutorial-uvod-do-jazyka-java>
- [23] LACKO L. *Vývoj aplikací pro Android*. Brno: Computer Press, 2015, s. 17-29 [cit. 2020-02-12], ISBN 978-80-251-4347-6
- [24] About Swift – The Swift Programming Language (Swift 5.2). Swift [online]. [cit. 2020-02-16]. Dostupné z: <https://docs.swift.org/swift-book/>
- [25] Swift – Apple Developer. Apple Developer [online]. [cit. 2020-02-16]. Dostupné z: <https://developer.apple.com/swift/>
- [26] BAKIR A. *Program the Internet of Things with Swift for iOS*. Apress, 2018, s. 3-4 [cit. 2020-02-16]
- [27] Mobile Operating System Market Share Worldwide | StatCounter Global Stats. Statcounter [online]. [cit. 2020-03-07]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [28] App stores: number of apps in leading app stores 2020. Statista [online]. [cit. 2020-05-06]. Dostupné z: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>
- [29] SAP Business Software. Sap [online]. [cit. 2020-03-20]. Dostupné z: <https://www.sap.com/why-sap.html>
- [30] SAP Store. Sap [online]. [cit. 2020-03-20]. Dostupné z: <https://www.sapstore.com/search>
- [31] Procurement Pricing Plans. Kissflow [online]. [cit. 2020-03-20]. Dostupné z: <https://kissflow.com/procurement/pricing/>
- [32] Procurement Cloud Software. Kissflow [online]. [cit. 2020-03-20]. Dostupné z: <https://kissflow.com/procurement/>

SEZNAM OBRÁZKŮ

Obr. 1: Diagram obrazovek	28
Obr. 2: Přihlašovací obrazovka.....	30
Obr. 3: Registrace	31
Obr. 4: Uvítání	32
Obr. 5: Hlavní obrazovka.....	33
Obr. 6: Menu.....	34
Obr. 7: Aktivní zakázky.....	35
Obr. 8: Detail Aktivní zakázky	36
Obr. 9: Dokončené zakázky.....	37
Obr. 10: Nákupní objednávka.....	39
Obr. 11: Historie zakázek	40
Obr. 12: Volné zakázky	41
Obr. 13: Detail volné zakázky	42
Obr. 14: Potvrzení zakázky.....	43
Obr. 15: Nová zakázka.....	44
Obr. 16: Formulář pro VP.....	45
Obr. 17: Přidání záznamu do VP	46
Obr. 18: Zprávy.....	47
Obr. 19: Nová zpráva.....	48
Obr. 20: Detail zprávy	49
Obr. 21: Relační model databáze.....	51
Obr. 22: Vývojový diagram přihlašovací obrazovky.....	53

SEZNAM TABULEK

Tab. 1: Srovnání typů aplikací [2]	15
Tab. 2: Srovnání programovacích jazyků [20]	20

SEZNAM GRAFŮ

Graf 1: Podíl operačních systémů na trhu [28]	25
---	----