

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hanč** Jméno: **Jan** Osobní číslo: **457484**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Studijní obor: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Návrh a implementace nativní mobilní aplikace pro obchodní zástupce nebankovní společnosti

Název diplomové práce anglicky:

Design and implementation of a native mobile application for non-bank business representatives

Pokyny pro vypracování:

- Proveďte analýzu stávající mobilní aplikace pro podporu obchodních zástupců. Zaměřte se na nové požadavky vycházející z business analýzy a analýzy požadavků uživatelů. Vyhodnoťte, které z požadavků je vhodné implementovat. Pro vybrané požadavky a funkce vytvořte případy užití, scénáře a návrhy uživatelských rozhraní vycházející z aktuálních postupů práce. Implementujte prototypy aplikace na platformě Android. Výsledné návrhy otestujte v testech použitelnosti s alespoň 5 uživateli.
- Implementujte podle návrhu aplikaci, která bude odpovídat business požadavkům. Aplikaci otestujte pomocí metod testování software.

Seznam doporučené literatury:

- Material design. Material Design [online]. Mountain View, CA: Google, 2017 [cit. 2018-01-07]. Dostupné z: <https://material.io/>
- EISENMAN, Bonnie. Learning React Native: building mobile applications with JavaScript. Beijing: O'Reilly, 2015. ISBN 1491929006.
- NUDELMAN, Greg. Android design patterns: interaction design solutions for developers. Indianapolis, In.: Wiley, c2013. ISBN 11-183-9415-1.
- MALÝ, Ivo. Analysis of Usability Tests with Context Model. Praha, 2012. PhD. Thesis. České vysoké učení technické v Praze, Fakulta elektrotechnická.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Pavel Malina, Provident Financial s.r.o.

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **19.02.2018**

Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **30.09.2019**

Ing. Pavel Malina
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Diplomová práce

**Návrh a implementace nativní mobilní aplikace pro obchodní
zástupce nebankovní společnosti**

Bc. Jan Hanč

Vedoucí práce: Ing. Pavel Malina

Studijní program: Otevřená informatika, Magisterský

Obor: Softwarové inženýrství

24. května 2018

Poděkování

Děkuji svému vedoucímu práce panu Ing. Pavlu Malinovi, za konzultace a odborné vedení. Dále bych chtěl poděkovat své přítelkyni a rodině za jejich podporu při studiích. Děkuji také společnosti Provident Financial s.r.o. za poskytnutí nezbytných informací a zdrojů potřebných pro napsání této práce.

Prohlášení

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

V Praze dne 25. 5. 2018

.....

Abstract

This thesis deals with design and implementation of a mobile application for the non-banking company. The app is intended for business representatives who visit clients and write contracts, offer additional products and choose installments. A sales representative needs to know the information about the client, so they need to keep them together, so this application helps them. Offers basic statistics, client lists, and forms to fill in. It also provides him with the necessary tools to make it easier to work with the customer.

Key words: React native, native application, Application Analysis and Design, Application Development, Marketing

Abstrakt

Tato práce se zabývá návrhem a implementací mobilní aplikace pro nebankovní společnost. Aplikace je určena obchodním zástupcům, kteří navštěvují klienty a sepisují s nimi smlouvy, nabízejí další produkty a vybírají splátky. Obchodní zástupce potřebuje znát informace o klientovi, takže je potřebuje mít stále při sobě, tedy tato aplikace mu k tomu dopomáhá. Nabízí základní statistiky, seznamy klientů a formuláře k vyplnění. Také mu poskytuje potřebné pomůcky k jednodušší práci se zákazníkem.

Klíčová slova: React native, nativní aplikace, analýza a návrh aplikace, vývoj aplikace, marketing aplikace

Obsah

1	Úvod	1
1.1	O společnosti	1
1.2	Popis práce obchodního zástupce	2
2	Analýza	3
2.1	Aktuální řešení	3
2.1.1	Popis	3
2.1.2	Technologie	4
2.1.2.1	Mobilní aplikace	4
2.1.2.2	Aplikační server	5
2.1.2.3	Databáze	5
2.1.3	Výhody a nevýhody řešení	6
2.2	Potřeby obchodních zástupců	6
2.2.1	Výsledky průzkumu	6
2.2.2	Vyhodnocení průzkumu	8
2.3	Business požadavky	8
2.3.1	Nefunkční požadavky	8
2.3.1.1	Offline	8
2.3.1.2	Synchronizace	9
2.3.1.3	Logování	9
2.3.1.4	Zabezpečení	9
2.3.1.5	Provázanost na jiné aplikace	9
2.3.2	Funkční požadavky	10
2.3.2.1	Po spuštění	10
2.3.2.2	Osobní informace	11
2.3.2.3	Skóring	11
2.3.2.4	Produktová tabulka	11
2.3.2.5	Přehled zákazníků a jejich detail	12
2.3.2.6	Seznam kontaktů	12

2.3.2.7	Reportování	12
3	Návrh řešení	13
3.1	Návrh nefunkčních požadavků	13
3.1.1	Ukládání dat do mezipaměti	13
3.1.2	Synchronizace	13
3.1.2.1	Bez ohledu na připojení	14
3.1.2.2	S ohledem na připojení	14
3.1.2.3	Na požádání	14
3.1.3	Logování	15
3.1.4	Zabezpečení	15
3.1.5	Spolupráce s jinými aplikacemi	15
3.1.5.1	Odesílání textových zpráv	15
3.1.5.2	Přímé volání	16
3.1.5.3	Mapy	16
3.2	Návrh funkčních požadavků	17
3.2.1	Přihlášení	17
3.2.2	Úvodní obrazovka	17
3.2.3	Rozbalovací menu	18
3.2.4	Skóringová zpráva	19
3.2.5	Reportování	20
3.2.6	Produktová tabulka	20
3.2.7	Přehled zákazníků	21
3.2.8	Detail zákazníka	22
3.2.9	Kontakty	22
3.2.10	Nastavení	23
3.3	Testování s uživateli	24
3.3.1	Vyplnění skóringové informace a odešlete zprávu	24
3.3.2	Otevření navigace k zákazníkovi	24
3.3.3	Zavolání na helpdesk	25
3.3.4	Nastavení automatického stahování základních informací o zákaznících přes datové připojení	25
3.3.5	Odhlášení	25
3.3.6	Shrnutí	25
4	Implementace	27
4.1	Technologie pro vývoj	27
4.1.1	Nativní aplikace	27
4.1.2	Webová aplikace	27

4.1.3	Hybridní aplikace	28
4.1.3.1	Xamarin	28
4.1.3.2	Ionic	28
4.1.3.3	React Native	29
4.2	Prostředí pro vývoj	31
4.2.1	Node.js	31
4.3	Android Studio	31
4.3.1	Xcode	32
4.3.2	Visual Studio Code	32
4.4	Server	32
4.4.1	Autorizační aplikace	32
4.4.2	Enterprise aplikace	33
4.4.2.1	API Základních informací	33
4.4.2.2	API obchodních výsledků	34
4.4.2.3	Skóringové API	34
4.4.2.4	Zákaznické API	36
4.4.2.5	Reportovací API	37
4.4.2.6	Produktové API	38
4.4.2.7	API kontaktů	39
4.5	Řešení nefunkčních požadavků	39
4.5.1	Jazykové překlady	40
4.5.2	Ukládání dat do mezipaměti	40
4.5.3	Synchronizace	40
4.5.3.1	Běh aplikace	40
4.5.3.2	Zobrazení základních dat	40
4.5.3.3	Zobrazení detailních dat	40
4.5.4	Logování	41
4.5.5	Zabezpečení	41
4.5.6	Odesílání textových zpráv a přímé volání	41
4.5.7	Provázanost na jiné aplikace	42
4.6	Řešení funkčních požadavků	42
4.6.1	Přihlášení	42
4.6.2	Úvodní obrazovka	43
4.6.3	Rozbalovací menu	44
4.6.4	Skóringová zpráva	45
4.6.5	Produktová tabulka	46
4.6.6	Přehled zákazníků	47
4.6.7	Detail zákazníka	47

4.6.8	Kontakty	48
4.6.9	Reportování	49
4.6.10	Nastavení	50
5	Testování	51
5.1	Testovací scénáře	51
5.1.1	Zobrazení detailu zákazníka z Aše	51
5.1.2	Zobrazení titulní stránky s obchodními výsledky	52
5.1.3	Vyplnění a odeslání denních výsledků	52
5.1.4	Zavolání na helpdesk	53
5.1.5	Odhlášení z aplikace	53
5.2	Hodnocení uživatelů	53
5.3	Vyhodnocení testování aplikace	54
6	Závěr	55
A	Obsah příloženého CD	59
B	Tok obrazovek	61

Seznam obrázků

2.1	Vylepšená aplikace	4
2.2	Architektura současného řešení	5
2.3	Případy užití aplikace	10
3.1	Oauth2 autentizace	16
3.2	Okna přihlašování do aplikace	17
3.3	Úvodní obrazovka po přihlášení do aplikace	18
3.4	Rozbalovací menu	19
3.5	Obrazovky pro odesílání skóringu	19
3.6	Obrazovky pro vyplnění a odeslání výsledků	20
3.7	Produktová tabulka	21
3.8	Přehled zákazníků	21
3.9	Detail zákazníka	22
3.10	Obrazovka kontaktních informací společnosti	23
3.11	Obrazovka nastavení	23
4.1	Obrazovky pro přihlášení do aplikace	43
4.2	Úvodní obrazovky po přihlášení	44
4.3	Rozbalovací menu	44
4.4	Obrazovky pro vyplnění a odeslání skóringu	45
4.5	Produktová tabulka	46
4.6	základní přehled zákazníků	47
4.7	detailní informace o zákazníkovi	48
4.8	Obrazovky zobrazující seznamy kontaktů	49
4.9	Obrazovky pro vyplnění a odeslání reportingu	50
4.10	Obrazovka nastavení	50

Kapitola 1

Úvod

Tato diplomová práce se zaměřuje na obchodní zástupce (nebo také agenty) společnosti Provident Financial s.r.o. (dále společnost), kteří mají za úkol kontaktovat stávající i potenciální zákazníky, sepisovat s nimi smlouvy, poskytovat půjčky a vybírat splátky.

Aplikace, kterou máme za cíl vyvinout, má obchodním zástupcům zjednodušit práci, zefektivnit ji a dosáhnout tak lepších zisků pro společnost. Hlavním cílem aplikace je, aby agent nedělal práci navíc a mohl se dále věnovat svým úkolům. Do aplikace by měl vyplnit údaje jen jednou, a ty se propíší nadále do ostatních systémů, procesů a aktivit, aby mohl stihnout za stejnou dobu, více zákazníků a tím zvýšil nejen své zisky, ale i zisky pro společnost. Zároveň se aplikace bude snažit splnit požadavky obchodu na to, aby agent byl informován o svých výsledcích.

Nejprve zanalyzujeme stávající aplikaci, poté navrheme řešení pro novou aplikaci a na závěr ji vyvineme. Aplikaci necháme průběžně testovat uživateli, kteří ji budou v budoucnu používat v provozu.

1.1 O společnosti

Společnost Provident Financial s.r.o. působí na českém trhu již od roku 1997 a od té doby její služby využilo více než jeden milion zákazníků. Transparentně nabízí bezúčelové hotovostní i bezhotovostní půjčky, bez ručitele a bez skrytých poplatků. Mezinárodní skupina International Personal Finance (IPF), do níž patří i česká pobočka Provident Financial s.r.o., působí ve Velké Británii již od roku 1880. IPF v současnosti poskytuje služby ve dvanácti zemích světa: v České republice, Polsku, Maďarsku, Rumunsku, Bulharsku, Finsku, Estonsku, Lotyšsku, Litvě, Španělsku, Austrálii a v Mexiku. Jako jedni z prvních poskytovatelů půjček, získali licenci od ČNB, která je zárukou důvěryhodnosti, transparentního financování i odborné způsobilosti a potvrzuje silné postavení na českém finančním trhu. [10]

1.2 Popis práce obchodního zástupce

Obchodní zástupce je osoba, kterou si určila společnost, jako zástupce pro kontakt se zákazníkem. Tato osoba má v náplni práce aktivně vyhledávat potencionální zákazníky se zájmem o půjčku, informovat zájemce o všech podmínkách, které musí splnit, pro získání půjčky a následně jej provést celým procesem žádosti a sjednání půjčky.

Po vyplacení (podepsání smlouvy a předání sjednaného obnosu), se nový zákazník stává členem portfolia obchodního zástupce. Zástupce pro své portfolio klientů zajišťuje servis v podobě pravidelných návštěv, kde od zákazníků vybírá splátky na jejich hotovostní zápůjčky a informuje zákazníky o novinkách ve společnosti a možnostech dalšího financování, nebo řeší jakým způsobem srovná zákazník případné nedoplatky, pokud se dostane do prodlení.

Nedílnou součástí práce obchodního zástupce jsou i retenční nabídky, a to jak před doplacením, tak u zákazníků, kteří již zaplatili veškeré závazky vůči společnosti.

Každý obchodní zástupce má přehled o svých klientech. Seznam těchto klientů má ve svých seznamech, které dostává v tištěné podobě do svých rukou od svého nadřízeného. Nadřízenému se říká manažer a je zodpovědný za více agentů. Skupina, do které patří všichni jeho agenti se nazývá agentura a zároveň je to oblast, kterou má tento manažer na starosti. Manažer si se zástupci dává pravidelná setkání, kde si domlouvají další postupy a manažer pro ně připravuje další materiály pro práci.

Kapitola 2

Analýza

V této části se pokusíme shrnout aktuální řešení, které ve společnosti existuje, dále se budeme věnovat práci Obchodního zástupce a provedeme mezi nimi průzkum. Průzkumem zjistíme hodnocení aktuální aplikace, poté sestavíme seznam funkčních a nefunkčních požadavků k požadované aplikaci.

2.1 Aktuální řešení

Tato část se zaměřuje na popis stávající aplikace, použité technologie a funkcionality, které jsou v aktuální aplikaci použity.

2.1.1 Popis

Současná aplikace „OZ Cockpit“ (OZ = Obchodní Zástupce), která slouží obchodním zástupcům ve věci kontaktu se zákazníkem, byla vyvinuta na počátku roku 2016 jedním z kolegů a stala se tak první a jedinou aplikací ve společnosti napříč všemi trhy. Prvotní verze aplikace uměla pouze přihlášení do aplikace pomocí uživatelského jména a hesla. Jméno bylo osobní číslo a heslo telefonní číslo. Aplikace uměla zobrazit pouze vstupní pole pro skóringovou zprávu. Přihlášení probíhalo tak, že uživatel byl po zadání těchto informací ověřen v databázi přes Simple Object Access Protocol¹ (SOAP) službu, a pokud se informace shodovaly, byla odpověď o přihlášení kladná, byl vpuštěn. Omezením bylo to, že při přihlášení se odesílalo i unikátní číslo mobilního telefonu (IMEI), které se uložilo v databázi a žádný jiný telefon se poté nemohl přihlásit pod tímto uživatelem.

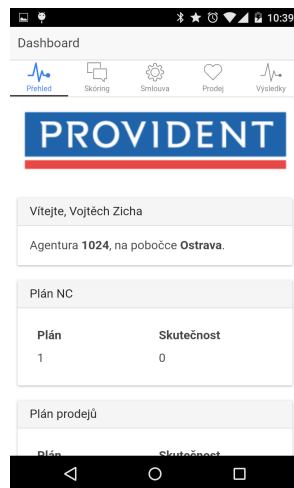
První aplikace uměla také, jak již bylo řečeno, odesílat tzv. skóringovou zprávu. Ta slouží k ověření zákazníka v Registru dlužníků a v dalších systémech, které určují, jestli má zákazník

¹protokol pro výměnu zpráv ve formátu XML přes síť

nárok na půjčku. Tato zpráva se skládá z parametrů o zákazníkovi, jako je jméno, rodné číslo, měsíční příjmy a výdaje, požadovaná výše půjčky a další nezbytné údaje. Tyto údaje jsou také, v případě kladné odpovědi ze systémů ověřující bezdlužnost (že zákazník má nárok na půjčku), odeslány do interního systému, který vypočítá maximální možnou výši a typ půjčky, kterou mu agent může nabídnout. Tyto informace se opět odesílají zpět na telefon agenta formou textové zprávy.

Po několika vylepšeních aplikace, ke stavu dne 1.1.2018, umožňuje aplikace přihlášení agenta stejným způsobem jako první verze, vyplnění skóringového, výsledkového formuláře a formuláře pro dohodu o snížených splátkách. Výsledkový formulář je soubor polí, které agent denně vyplňuje a odesílá svému nadřízenému. Pole obsahují informace o počtech uzavřených smluv, vybraných částkách atp. Dohoda o snížených splátkách informuje nadřízeného o tom, že zákazník v tuto chvíli nemá na zaplacení splátky a požaduje odložení splátek. Agent se se zákazníkem dohodne a informuje tímto nadřízeného. Ve zprávě se odesílá zákaznické číslo, dohodnutá splátka a den splátky. Tyto informace z formulářů se po stisknutí tlačítka „Odeslat“ odesílou v daném formátu svému nadřízenému či do integračních systémů.

Další funkcionalitou je zobrazování aktuálních nabízených produktů. Nevýhodou tohoto řešení je to, že aplikace produktovou tabulku obsahuje v kódu. Aplikace ještě ve stavu implementace této funkcionality neměla žádný backend, mimo samotného přihlašování. Stejně tak předpis pro formuláře, pro skóring a výsledky, je v mobilní aplikaci implementován, a při každé změně je nutná přeinstalace aplikace. Poslední přidanou funkcionalitou je zobrazení obchodních výsledků, které jako první začaly komunikovat s backendovou službou na firemním serveru. Obchodní výsledky jsou informace o tom, jak si agent vede v účtovacím období, jestli plní cíle a kolik mu ještě zbývá do splnění cíle uzavřít smluv, vybrat na splátkách a další informace potřebné v rychlém přehledu.



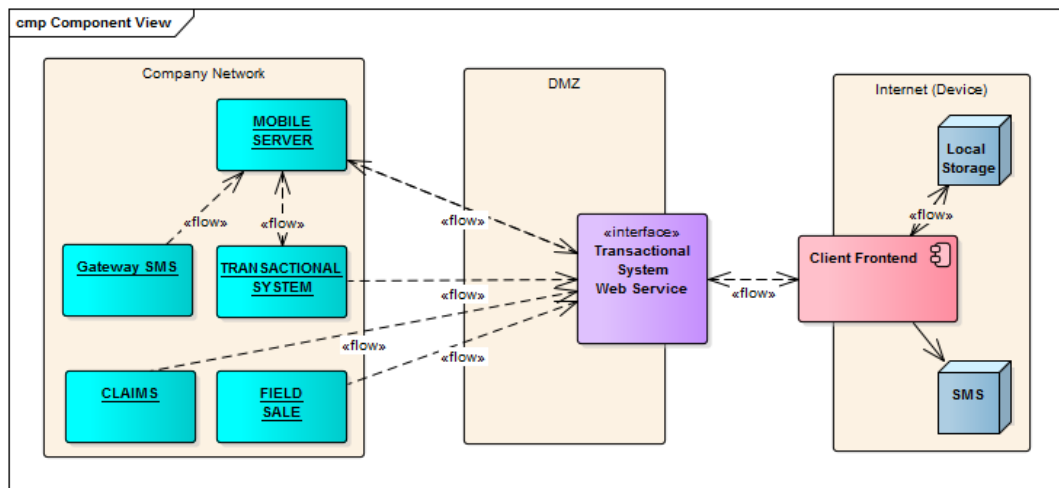
Obrázek 2.1: Vylepšená aplikace

2.1.2 Technologie

Jedná se o třívrstvou architekturu, která je rozdělena na prezentační vrstvu (Mobilní aplikaci), obchodní vrstvu (Aplikační server) a datovou vrstvu (Databáze). Na obrázku 2.2 je vidět, jak aplikace komunikuje. Komunikace probíhá pomocí SOAP, nebo REST služeb.

2.1.2.1 Mobilní aplikace

K vývoji první verze byl použit Ionic framework v.1, který byl schopen pokrýt požadavky na vyplnění polí, komunikaci se službou na serveru a také byl schopen vytvořit a odeslat tex-



Obrázek 2.2: Architektura současného řešení

ovou zprávu. Aplikace, pro které byl použit tento framework, jsou vyvinuty primárně přes Ionic command line utility (neboli „CLI“) a používají Cordovu na sestavní (build) a na nasazení (deploy) jako nativní aplikaci. Po naprogramování bylo možno tuto aplikaci sestavit do podoby pro tři základní podporované mobilní systémy (Android, iOS a Windows). Aplikace se dále rozšiřovala a v současné době pro další rozšíření přestal tento framework stačit. Nebyl schopen pokrýt další požadavky, a proto byl vývoj v tomto frameworku zastaven. Zároveň již byl vyvinut Ionic framework 2.0, který výrazně změnil a zjednodušil původní verzi, která se dále nerozšiřovala. Lze tedy Ionic framework v.1 považovat za zastaralou technologii. [9]

2.1.2.2 Aplikační server

K vývoji serverové aplikace, která je umístěna v demilitarizované zóně (DMZ), což je server, který má vystaveny endpointy do internetu, je použit jazyk C# s využitím ASP.NET Web Aplikace (.NET Framework 4.5.1), která je nasazena na serveru Internet Information Services (IIS). Jedná se o dvě serverové aplikace, první je autorizační, která má jeden endpoint, druhá je služba, ke které lze přistoupit po úspěšném ověření, která poskytuje informace pro agenta. Tento server lze dále rozšiřovat o nové endpointy.

2.1.2.3 Databáze

Pro ukládání dat je využit SQL Server 2008 R2. Na serveru je databáze MS SQL, která se serverem komunikuje pomocí Entity Frameworku. Entity framework je objektově-relační zobrazovač, který umožňuje .NET vývojářům jednodušší práci s relačními databázemi, ome-

zuje velikost napsaného kódu a umožňuje práci s databázovými tabulkami, jako s objekty. [2]

2.1.3 Výhody a nevýhody řešení

Díky této aplikaci si agent ušetřil mnoho času, vzhledem k tomu, že při vyplnění formuláře je potřeba dbát na přesné pořadí a syntaxi. Aplikace mu ji ověřila a díky tomu se vyvaruje několikanásobné úpravě zprávy a odesílání. Nyní tyto pole stačí vyplnit pouze jednou, vše je zkontrolováno a odesláno již validní. Agent má také přehled o svém aktuálním stavu, který je pro něj motivující, pokud se mu daří plán plnit. S tím souvisí i odměny pro agenta za každou uzavřenou smlouvu a při splnění plánu dostává další bonus.

Aktuální řešení pro potřeby agentů je nedostatečné z důvodu omezené funkčnosti aplikace, která pouze nabízí přihlášení a vyplnění několika z potřebných formulářů. Formuláře nejsou dynamicky rozšiřující, a tedy pro každou úpravu detailu nějakého formuláře, je třeba aktualizovat aplikaci. Další nevýhodou aplikace je produktová tabulka, která je také uložena přímo v aplikaci a pro každou změnu produktů, je třeba aplikaci změnit. Pokud se aplikace změní, je nasazena aktualizace do Google Play a Play Store, kde se uživatelům aktualizuje i po několika dnech. To má za následek, že při odesílání zprávy z neaktualizované aplikace, kde chybí několik polí, vrátí odpověď s tím, že je chybně vyplněn formulář, nebo že agent nabízí zákazníkovi produkt, který již není v nabídce.

2.2 Potřeby obchodních zástupců

Byl proveden průzkum mezi jednotlivými agenty o tom, co vykonávají, a jaké vlastnosti či pomůcky, kdyby měli s sebou, by jim ulehčily práci. Otázky, které byly položeny, byly směřovány na aktuální funkčnost aplikace i na její možné rozšíření. Každá otázka byla ohodnocena body, které stanovilo obchodní oddělení. Body byly převedeny na procentuální ohodnocení, ze kterého byl vypočten průměr.

2.2.1 Výsledky průzkumu

Průzkum byl proveden mezi dvaasedmdesáti obchodními zástupci, kteří vyplnili dotazník a výsledky jsou následující.²

- Jak často používáte aplikaci OZ Cockpit? - Možnosti “Několikrát denně” odpovědělo 75 % uživatelů, 13 % “Jednou denně” a 10 % “několikrát týdně”. Z toho vyplývá, že agenti jsou seznámeni s aplikací a dle výsledku se dá usuzovat, že je využívána ve velké míře.

²Pozn. Výsledky pod 5 % nejsou zobrazeny ve vyhodnocení.

- Jak jste celkově spojeni s aplikací? - Možnosti “Velmi spokojeni” odpovědělo 13 %. Nejvíce, 56 % agentů odpovědělo že jsou “převážně spokojeni” a 18 % jsou zástupci, kteří nejsou ani spokojeni, ani nespokojeni. Důvody s nespokojeností jsou v největší míře v rychlosti aplikace. Agenti si také stěžují na funkcionality, které jim chybí. Některé z funkcionalit v aplikaci jsou, ale vzhledem k tomu, že o nich agent neví, lze usuzovat, že agenti nejsou informováni o nových funkcionalitách, nebo je aplikace příliš složitá na používání.
- Jak hodnotíte spolehlivost? - To, že výsledky nejsou nijak příjemné, je patrné již z první odpovědi. Odpověď “Velmi vysoká spolehlivost” odpovědělo 10 % uživatelů a druhou možnost “Vysoká spolehlivost” odpovědělo 38 % uživatelů. Nejvyšší skóre dostalo hodnocení “Ani vysoká, ani nízká spolehlivost”, které obdrželo 49 %. Toto hodnocení je kvůli tomu, že aplikace ne vždy dokáže správně ověřit vstupní data, která jsou vyplněna v aplikaci, tedy se stane, že při odeslání skóringu se vrátí odpověď o chybně vyplněných údajích.
- Jak hodnotíte jednoduchost používání aplikace? - Odpověď “Velmi jednoduché používání” označilo 22 % dotazovaných, 41 % dotazovaných označilo odpověď “Jednoduché používání” a odpověď “Normální používání” označilo 34 % dotazovaných. Aplikace je podle výsledků jednoduchá na používání, ale vzhledem k tomu, kolik funkcí uživatelé používají, lze usuzovat, že se v aplikaci ne všichni zcela vyznají.
- Jak často využíváte jednotlivé funkcionality? - V této otázce byly vyjmenovány jednotlivé funkcionality, které aktuálně aplikace nabízí. Možnosti byly “Velmi často”, “Často”, “Ne moc často” a “Vůbec ne”. Průměrně se funkcionality používají na 57 % což udává to, že ne všechny funkcionality jsou v aplikaci potřebné, nebo je uživatelé neumí používat. Dle odpovědí na jednotlivé otázky se nejméně používá funkcionality “Dohoda o snížených splátkách”, na kterou odpovědělo “Vůbec ne” nejvíce uživatelů a celkové hodnocení má pouhých 21 %. Nejlépe dopadly funkcionality “Skóring” a “Výsledkové zprávy”. Ty byly implementovány první.
- Uvítali byste některou z následujících funkcionalit? - Byl poslední z dotazů, kde byl dán seznam šesti funkcionalit, které se plánovaly vyvinout a které byly i součástí obchodních požadavků. Nejvíce dotazovaných by uvítalo funkcionality seznamu kontaktů na jejich nadřízené, helpdesk a další. Naopak nejméně by využili funkcionality “Kalendáře pro plánování návštěv”. Díky tomuto průzkumu a dohodě s oddělením obchodu, se rozhodlo, jaké funkcionality se budou vyvíjet dále.
- Obchodní zástupci byli také dotazováni otevřenými otázkami na to, co jim v aplikaci chybí, co by doplnili, či pokud jsou s nějakou funkcionalitou nespokojeni. Měli napsat svůj komentář. Objevovali se zde komentáře k jednotlivým funkčnostem, ale také k

rychlejšímu načítání aplikace. Pro další návrh by tyto připomínky měly být brány v potaz a měly by se aplikovat, či by se jim mělo vyhnout v dalším návrhu.

2.2.2 Vyhodnocení průzkumu

Většinu funkcionalit agenti používají, ale ne vždy je hodnotí pozitivně. Spolehlivost není na nejlepší úrovni, což může být použitou technologií, či vysokou náročností aplikace na telefon. Vzhledem k tomu, že v aplikaci uživatelům chybí nějaké doplňky, bylo rozhodnuto, že aplikace nedostačuje aktuálním potřebám agentů a je potřeba ji změnit. Je potřeba zjednodušit celou aplikaci, protože ne vždy uživatelé vědí, co se k čemu používá a potřebuje.

2.3 Business požadavky

Požadavky na aplikaci byly stanoveny jednotlivými obchodními manažery, kteří se vyjádřili k tomu, které funkce jsou v aplikaci navíc, a které jim tam chybí. Odpovědi byly rozděleny do dvou kategorií. První kategorie se zabývá technickými funkcionalitami, kterým by měla aplikace odpovídat bez zobrazení jednotlivých oken. Druhá kategorie je požadovaný seznam oken a jednotlivých obchodních požadavků, které informace, nebo pole by měla aplikace poskytovat.

Aplikace bude určena agentům, kteří se budou setkávat se zákazníky. Musí být bezpečná a reakce na požadavek musí být rychlá. Také musí být přehledná, aby uživatel měl vždy přehled o jednotlivých možnostech aplikace.

2.3.1 Nefunkční požadavky

Seznam nefunkčních požadavků je soupis vlastností aplikace, které bude splňovat bez ohledu na počet zobrazených oken.

2.3.1.1 Offline

Pokud uživatel bude v oblasti, kde není dostatečný přístup k internetu (např. slabý signál), nebo nebude mít v telefonu připojení přes datové připojení, tak se stále musí dostat k informacím o zákazníkovi, které si předem vybere a mít možnost odeslat vyplněné formuláře jinou cestou než přes webové služby, například pomocí textových zpráv. Zobrazení obchodních výsledků, tvaru zpráv, nebo produktů, které jsou aktualizované při dotazu na server, musí být zobrazena poslední známá verze. Tyto informace se aktualizují při dostupnosti datového připojení.

2.3.1.2 Synchronizace

Částečná synchronizace dat se provede za jakéhokoli internetového připojení na žádost uživatele, nebo pokud nějaká data potřebuje načíst aktuálně. Jsou to převážně informace nutné pro prvotní načtení aplikace, nebo informace, které nejsou náročné na spotřebu dat.

Úplná synchronizace dat ze zařízení se provede automaticky při připojení k Wi-Fi síti, kvůli šetření mobilních dat. Aktualizují se informace o zákaznících a produkty. Uživatel by měl mít možnost nastavit si sám, jestli dovolí aplikaci automaticky aktualizovat tyto informace při datovém připojení.

Agenti mají firemní tarif, který obsahuje 1,5GB mobilních dat měsíčně, tedy velikost synchronizace pro jeden dotaz přes mobilní data, nesmí přesáhnout 1MB.

2.3.1.3 Logování

Logování je nutné k případnému zjištění chyb, či běhu aplikace. Logování je prováděno automaticky a v případě výskytu chyby se automaticky odešle zpráva na podporu, která zaeviduje incident a postoupí jej dále k řešení. V logu musí být patrné, jaké kroky uživatel provedl, než se stala chyba, tedy log musí mít historii.

2.3.1.4 Zabezpečení

Požadavek na bezpečnost je nutnost, neboť se pracuje s citlivými daty o zákaznících. Je tedy potřeba zajistit, aby osoba, která není oprávněná tyto data zobrazit, se k nim nedostala. Dále je potřeba mít možnost data na dálku odstranit v případě, že by došlo ke ztrátě nebo odcizení mobilního telefonu agenta.

Přenos na server musí být šifrovaný přes OpenSSL a odesílaný protokolem HTTPS, protože komunikace mezi aplikací a serverem může být odposlouchávána a informace zneužity.

2.3.1.5 Provázanost na jiné aplikace

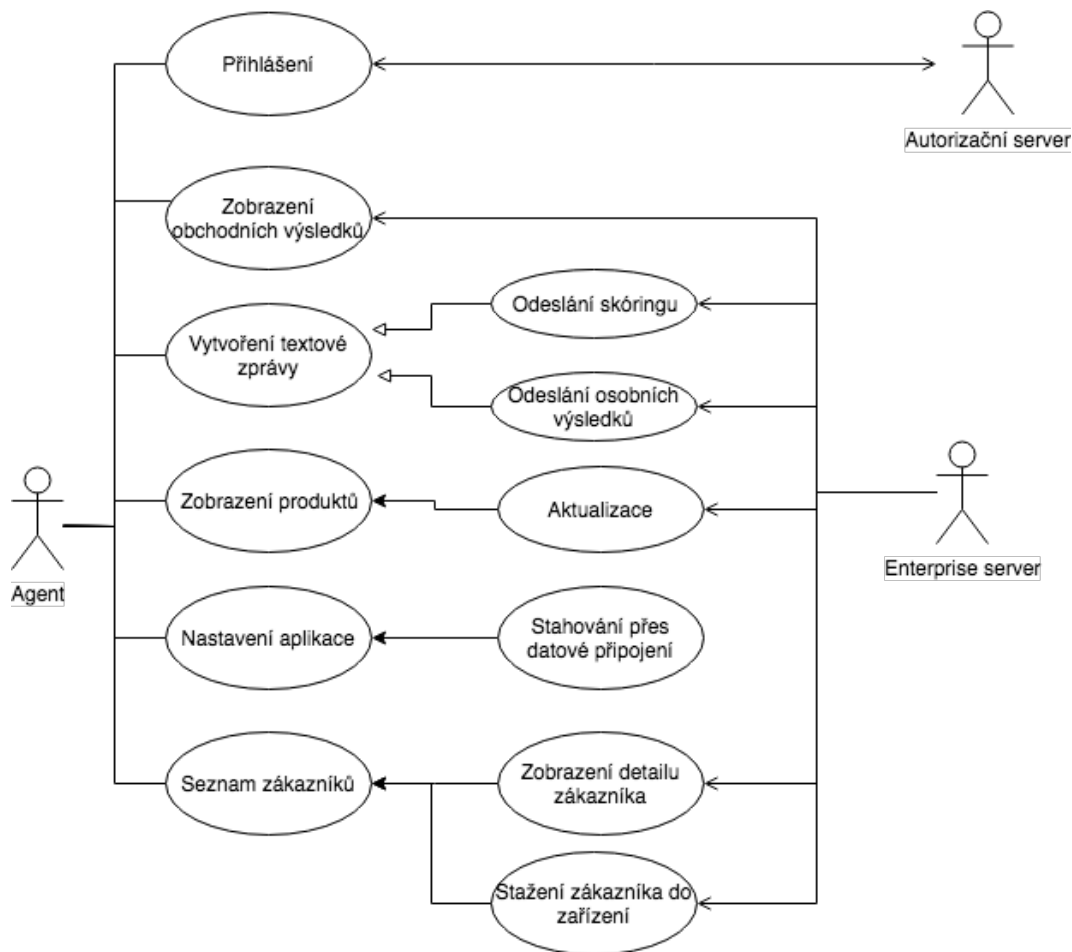
Aplikace musí podporovat funkci mapy, která po požadavku na zobrazení adresy bydliště na mapě, zobrazí pozici uvedené adresy.

Odesílání textových zpráv z telefonu je nutností z důvodu skóringu nebo reportingu, při kterém je potřeba, aby se zpráva sestavila do přesně stanoveného formátu a byla odeslána na určité telefonní číslo. Tyto informace jsou za normálních okolností odeslány přes webovou službu, ale v případě nedostupnosti služby, je nutné je odeslat jiným způsobem.

Další funkcí, kterou musí aplikace podporovat je přímé vytáčení. Tedy volání po vybraní telefonního čísla, a požadavku na vytáčení. Je potřeba zajistit vyplnění telefonního čísla pro přímé volání.

2.3.2 Funkční požadavky

Zde si popíšeme chování a jednotlivé požadavky na okna aplikace od obchodního oddělení. Tyto okna se mohou zobrazovat, nebo nemusí, v závislosti na nastavení v databázi pro konkrétní uživatele nebo skupiny. Toto nastavení se odesílá při prvním otevření aplikace, nebo při aktualizaci. Následující usecase 2.3, popisuje interakci uživatele (agenta) s aplikací a interakci aplikace se servery.



Obrázek 2.3: Případy užití aplikace

2.3.2.1 Po spuštění

Při prvním otevření aplikace, nebo při odhlášení z aplikace, se uživateli zobrazí okno pro přihlášení. Přihlášení bude prováděno pomocí čísla agenta a telefonního čísla uživatele. Ověřovací zprávou na uvedené telefonní číslo se odešle potvrzovací kód ze serveru, který po zadání do aplikace povolí vstup agentovi k poskytovaným informacím.

V případě špatného zadání, aplikace umožní uživateli změnit vstupní data a zkusit se znovu přihlásit.

Aby se agent přihlásil do aplikace, musí být připojen k internetu, buď na Wi-Fi nebo datovým připojením. Platnost přihlášení bude nastavena na 28 dní a poté dojde k automatickému odhlášení.

2.3.2.2 Osobní informace

Na úvodní obrazovce se zobrazí informace o agentovi (Jméno, příjmení, sekce a pobočka). Agent zde bude mít možnost vidět také Obchodní výsledky, kde bude zobrazen plán a aktuální výsledek pro každou kategorii. Každý agent může mít své kategorie výsledků, které chce jeho nadřízený zobrazovat. Je potřeba zajistit, aby se tyto kategorie zobrazovaly dynamicky. Každá kategorie bude obsahovat název, plán a realitu.

2.3.2.3 Skóring

Aplikace zobrazí obrazovku, ve které budou možnosti vkládání textu, přepínače a jiné podporující funkce, které dopomohou agentovi k jednoduššímu zadávání a automatickému ověřování. Pole pro vyplnění dostane aplikace ze serveru na žádost. Informace ze serveru obsahují popis pole, typ pole (přepínač ANO/NE, checkbox atd.), index pořadí v případě odeslání přes textovou zprávu a ID pole, které označuje klíč v JSON požadavku na server.

Skóring je po stisku tlačítka „Odeslat skóring“ odeslán přes internetové připojení na server ke zpracování. Toto se provede pouze pokud je agent připojen k internetu. Pokud agent nemá aktuálně přístup k internetu, je odeslána textová zpráva ve správném formátu.

Formát textových zpráv je posloupnost jednotlivých polí dle svého indexu, oddělena mřížkou. Na začátek je vždy připsán text „#SCOR#“, aby bylo poznat, že se jedná o skóringový typ zprávy.³

2.3.2.4 Produktová tabulka

Produktová tabulka slouží agentovi jako přehled možných nabízených produktů. Vzhledem k tomu, že se produkty velmi často mění, je potřeba zajistit dynamické vyplnění těchto údajů v aplikaci. Data do aplikace se dostanou z připravené webové služby, která poskytne název tohoto produktu a případně jeho detail.

³Příklad skóringové zprávy: #SKOR#Jan#Hanč#9302221212#20000#3000#45000#A

2.3.2.5 Přehled zákazníků a jejich detail

V tomto okně se zobrazí seznam zákazníků spadajících pod tohoto agenta. Na první pohled je potřeba rozeznat klienta, kterého si potřebuji zvolit a případně zobrazit jeho detail. K tomu je potřeba v seznamu (pro každého zákazníka) vidět jeho jméno a příjmení, zákaznické číslo a kontaktní adresa. Každý zákazník má své unikátní zákaznické číslo, které se používá pro další zobrazení detailů zákazníka.

Detail zákazníka je obrazovka, kde bude vidět jeho trvalá a kontaktní adresa, telefon, datum narození a celková dlužná částka. Tento detail se musí zobrazit po klepnutí na jednotlivého zákazníka. Je potřeba, aby detaily o zákaznících nevyužívaly velké množství mobilních dat, pokud to nebude nutné, nebo vyžádané uživatelem, neboť agenti mají k dispozici v terénu omezený objem mobilních dat.

2.3.2.6 Seznam kontaktů

Na základě pozice přihlášeného uživatele, tento seznam zobrazuje nadřízeného a ostatní agenty ve skupině. Zároveň zde budou zobrazeny kontakty na Helpdesk a oddělení pro jinou podporu. Společný telefonní seznam je potřeba dynamicky měnit. Každý kontakt bude obsahovat jméno a telefonní číslo.

Tento seznam agent využije například v případě, že nastanou nějaké problémy, nebude si vědět rady, nebo bude potřebovat kontaktovat svého nadřízeného.

2.3.2.7 Reportování

Každý den agent odesílá informace svému nadřízenému o svých výsledcích z aktuálního dne. Opět je nutno, jako ve skóringové zprávě, aby agent mohl jednotlivé parametry vyplnit jednoduše a některé (pokud jsou dostupné), mohly být předvyplněny místo něj a on by pouze zkontroloval, jestli je vše v pořádku a odeslal report.

Parametry, které agent musí vyplnit jsou parametry, které nastaví jeho nadřízený. Ze serveru se posílají informace, které obsahují seznam parametrů nastavených nadřízeným. Položka seznamu obsahuje název parametru, typ vyplňovaného pole (Text, číslo, ANO/NE, a další), index v případě odeslání textové zprávy a ID pole, které označuje klíč v JSON požadavku na server.

Report se odešle požadavkem na server, pokud je agent připojen k internetu. Pokud agent nemá aktuálně přístup k internetu, je odeslána textová zpráva v předem daném formátu. Formát textových zpráv je posloupnost jednotlivých polí dle svého indexu, oddělena mřížkou. Na začátek je vždy připsán text „#RES#“, aby bylo poznat, že se jedná o výsledkovou zprávu.

4

⁴Příklad výsledkové zprávy: #RES#3#2#22#20000#21400#45000

Kapitola 3

Návrh řešení

V této kapitole se budeme zabývat návrhem řešení aplikace. Design aplikace zpracujeme pomocí programu “Just In Mind (Free)”, který má omezené vlastnosti a funkce oproti plné verzi, kterou jsme si nevybrali, protože je placená a pro studenty není dostupná zdarma. Také si zde zobrazíme jednotlivé obrazovky a popíšeme jejich funkčnosti. Návrh nefunkčních požadavků zpracujeme a popíšeme navrhované řešení.

3.1 Návrh nefunkčních požadavků

Provedeme návrh řešení nefunkčních požadavků, aby splňovaly nefunkční požadavky z analýzy.

3.1.1 Ukládání dat do mezipaměti

Ukládání do mobilního zařízení, je řešeno pomocí interní paměti. Ukládání je v podobě klíč-hodnota do paměti telefonu. Veškeré nastavení nesmí být ukládáno do externí paměti z důvodu snadného přenosu informací do jiných zařízení. Tato paměť bude fungovat jako zdroj informací a nastavení, pokud server nebude k dispozici. Informace poskytující jsou například poslední známé obchodní výsledky, seznam oken a další.

3.1.2 Synchronizace

Pro synchronizaci dat používáme dotazování přes Web API využívající rozhraní REST¹ které je založeno na jednoduchých HTTP volání. K synchronizaci použijeme metody HTTP pro webové služby, které spadají do kategorie „RESTful“, tedy GET, POST, PUT a DELETE. Výhody REST komunikace oproti SOAP komunikace spočívají hlavně v objemu dat. K

¹Representational State Transfer

omezení objemu dat také přispěje zapnutí GZIP komprese přenášených zpráv ze strany IIS. [5]

Synchronizaci rozdělíme do tří typů. Tyto typy od sebe oddělují druh připojení k internetu a závažnost obsahových informací. Tyto typy se snaží brát v potaz množství dat, které může agent měsíčně použít a velikost úložného místa v telefonu, které omezuje část 3.1.2.3.

3.1.2.1 Bez ohledu na připojení

Pro správný běh aplikace je potřeba mít přístup k internetu při přihlašování a poté pro prvotní běh aplikace. Po přihlášení je potřeba stáhnout nastavení aplikace, které zaručuje správný běh této aplikace a to, že uživatel opravdu uvidí pouze obrazovky a nastavení, které je mu dovoleno. Tyto informace jsou stahovány bez ohledu na to, jestli je uživatel připojen k síti Wi-Fi, nebo pomocí datového připojení. Tyto informace jsou ukládány do lokální paměti a načteny v případě nedostupnosti internetového připojení. V případě dostupnosti, jsou data aktualizována při zapnutí aplikace.

V tomto režimu nastavení se počítá, že se jedná o požadavky na nastavení, oken aplikace, zobrazování obchodních výsledků a tvar skóringových a výsledkových polí. Počítá se, že tyto informace jsou buď velmi důležité, nebo velmi malé na objem přenesených dat.

3.1.2.2 S ohledem na připojení

Při ohledu na připojení je potřeba vzít v úvahu velikost přenášených dat. Data, která jsou potřeba pro zobrazení základních informací o klientech, nebo kontakty, jsou náročná na objem přenesených dat. Vzhledem k velikosti a počtu, jsou stažena nebo aktualizována okamžitě při připojení k Wi-Fi, protože pro agenta jsou důležitá. Při datovém připojení je na uživateli, zda povolí stahování těchto informací. Data jsou uložena v lokální paměti a načtena při otevření příslušného okna. Pokud je při otevření příslušného okna dostupné připojení k Wi-Fi, je automaticky stažena aktualizace.

3.1.2.3 Na požádání

Data, která uživatel nepotřebuje do té doby, dokud se neplánuje setkat s klientem, může si později požádat o stažení do paměti. Data nejsou automaticky stahována ani aktualizována, ale pokud uživatel ví, že za klientem půjde a bude potřebovat jeho detail, může si data o zákazníkovi stáhnout. Pokud uživatel zadá požadavek na stažení detailu klienta při připojení k Wi-Fi, tyto informace se stáhnou bez dotazu. Pokud uživatel bude připojen pouze k datovému připojení bude dotázán, jestli si přeje tyto informace přes toto připojení stáhnout. Možnost nastavení stahování bez dotázání je umožněna v nastavení aplikace. Data v

zařízení jsou uložena pouze dočasně po dobu 72 hodin kvůli šetření místa v paměti telefonu a bezpečnosti v případě ztráty zařízení.

3.1.3 Logování

Pro uchovávání logování se bude využívat logování do konzole, které pomocí funkcionalit bude zachytávat doplněk Firebase, který po vyhodnocení chyby, připraví report pro odeslání na administrátory či helpdesk.

Druhou možností je implementace vlastního loggeru, který bude fungovat na podobné bázi jako Firebase. Tato funkcionalita musí splňovat ukládání chybových hlášení a následně odeslání na server, při dostupnosti internetového připojení.

3.1.4 Zabezpečení

Pro přihlašování do aplikace, se používá dvoufázová autentizace OAuth2 (obr. 3.1). Nejprve probíhá ověření, jestli uživatel má právo přistoupit k citlivým datům. Uživatel bude nejprve ověřen odesláním autentizačních informací na server, na základě kterých se jako odpověď vrátí autorizační token. V případě správně vyplněných údajů, se ze serveru na telefonní číslo agenta odešle textová zpráva s vygenerovaným klíčem. Tento klíč uživatel zadá do aplikace a společně s autorizačním tokenem se odešle na autorizační server. Server ověří, jestli token a klíč souhlasí. V případě shody, vrátí uživateli přístupový token (bearer), který umožní uživateli přístup k chráněným informacím. V případě jakékoli neshody, uživatel musí začít znovu.

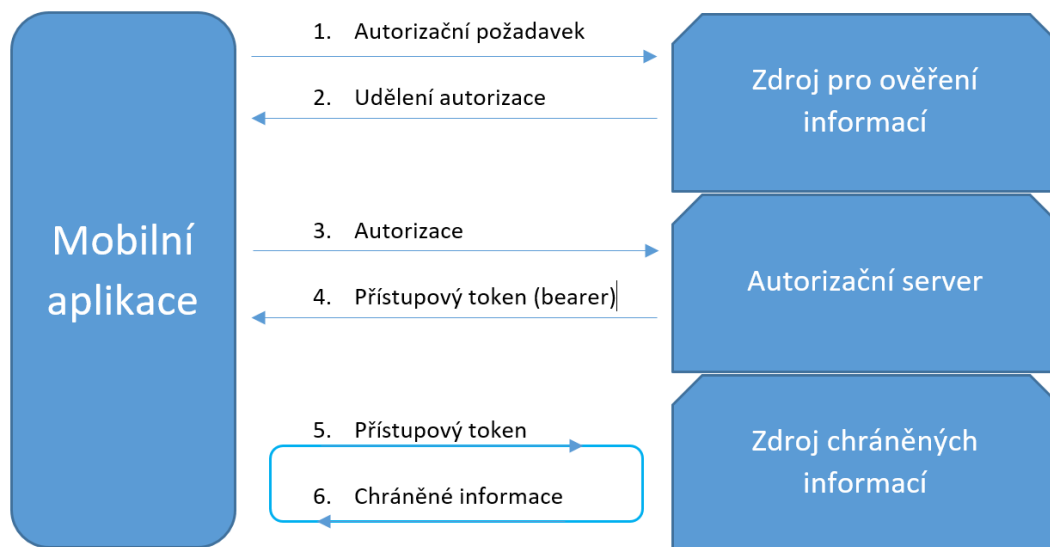
Informace uložené v interní paměti musí být šifrované šifrou AES-256 (Advanced Encryption Standard), která poskytuje dostatečné zabezpečení pro ukládání dat. Zároveň se při odhlášení z aplikace, musí lokální paměť smazat, aby se k informacím nedostal žádný jiný uživatel.

3.1.5 Spolupráce s jinými aplikacemi

V případě kontaktu se zákazníkem musí tato aplikace umět využívat i jiné aplikace, které jsou již vyvinuty a na telefonu standardně nainstalovány. Každý operační systém má své předdefinované aplikace, které umí využít. Těchto aplikací se budeme snažit využít.

3.1.5.1 Odesílání textových zpráv

V případě nedostupnosti internetového připojení, je potřeba skóring a reporty odeslat pomocí textové zprávy. Odesílání skóringu a reportu bude prováděno automaticky pokud to systém bude umožňovat. Operační systémy iOS mají omezení v odesílání zpráv z cizích



Obrázek 3.1: OAuth2 autentizace

zdrojů. V tomto případě zpráva bude sestavena a připravena v poli pro odeslání. Odeslat ji musí uživatel manuálně sám.

Agent má v systému registrované pouze jedno telefonní číslo, ze kterého mu je dovoleno odesílat tyto zprávy. Pokud aplikaci používá na jiném mobilním čísle a odešle skóring, nebude platný a automaticky obdrží odpověď, že jeho telefonní číslo není v systému nalezeno.

3.1.5.2 Přímé volání

Seznam kontaktů, nebo informace o zákazníkovi, obsahují informaci o telefonním čísle. Na toto telefonní číslo lze odesílat zprávy a lze na ně také volat. Aplikace musí dovolit přesměrování na výchozí aplikaci pro volání a předat jí toto telefonní číslo, které uživatel chce vytočit.

3.1.5.3 Mapy

Pokud se agent rozhodne kontaktovat zákazníka osobně, uvítá možnost „Zobrazení na mapě“, která zobrazí adresu zákazníka ve výchozí aplikaci na mapě. Pokud taková aplikace v telefonu neexistuje, zobrazí se adresa na mapě Google na webové stránce v okně prohlížeče.

3.2 Návrh funkčních požadavků

Zde popíšeme a ukážeme jednotlivé zpracování funkčních požadavků aplikace. Ke každé funkcionalitě je přiřazena obrazovka, která popisuje požadovanou funkci. Návrh obrazovek je tvořený pro mobilní telefon Nexus 5S.

3.2.1 Přihlášení

Obrazovka 3.2a se objeví jako první po spuštění aplikace. Tato obrazovka bude obsahovat dvě textová pole pro zadání přihlašovacích údajů. První na osobní číslo, druhé na telefonní číslo. Po vyplnění požadovaných údajů na úvodní stránce, dojde k odeslání požadavku na server a v případě kladné odpovědi, je obrazovka vyměněna za druhou obrazovku 3.2b, kam uživatel zadá ověřovací kód, který mu byl zaslán na jeho telefonní číslo. Při vyplnění správného čísla, přejde aplikace na úvodní stránku 3.3a.



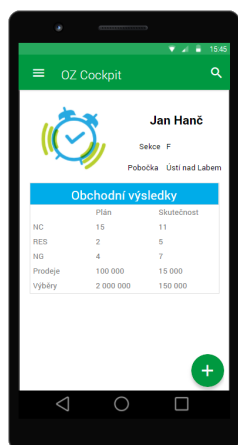
Obrázek 3.2: Okna přihlašování do aplikace

Pokud na první obrazovce přihlašování nastane chyba, uživateli se objeví obrazovka s chybou 3.2c, která oznamuje uživateli, že zadal nesprávné údaje a celé přihlašování musí začít znovu. Pokud nastane chyba v opisování zprávy, uživatel má možnost 2 oprav před tím, než aplikace zobrazí chybové hlášení a uživatel je přesměrován na první stránku přihlášení.

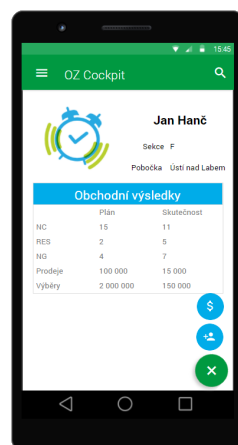
3.2.2 Úvodní obrazovka

Na úvodní obrazovce, v horní části, je zobrazen název aplikace. Pro rozbalení jednotlivých možností v levém horním rohu se zobrazí překrývající menu, kde je možno zvolit z jednotlivých obrazovek popsaných v kapitole 3.2.3.

Obsah první stránky (obr. 3.3a) tvoří jméno, příjmení, sekce kterou agent zastupuje a agenturu, ve které působí. Při prvním pohledu do aplikace, uživatel dostane přehled o svých obchodních výsledcích. Tabulka obsahuje výsledky jednotlivých kategorií. V prvním sloupci je název hodnotící kategorie, ve druhém sloupci je plán, který agentovi naplánoval nadřízený a ve třetím sloupci jsou reálné výsledky, kterých dosáhl za poslední zúčtovací období.



(a) Úvodní obrazovka



(b) Výběr nové zprávy

Obrázek 3.3: Úvodní obrazovka po přihlášení do aplikace

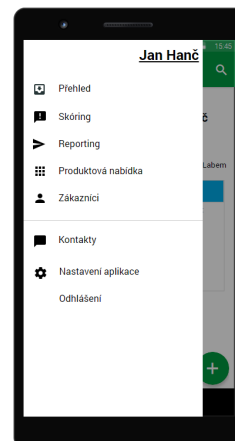
V pravém spodním rohu je kruhové zelené tlačítko se znakem plus. Pokud uživatel klepne na toto tlačítko, rozbálí se menu nad tímto tlačítkem pro vytvoření nové zprávy jako je možné vidět na obrázku 3.3b. Ikona plusu se změní na ikonu křížku, která po klepnutí opět schová toto menu. K dispozici jsou dva druhy zpráv. První druh zprávy je zpráva skóringová se značkou osoby s plusem. Po klepnutí na tuto ikonu, uživatel bude přesměrován na obrazovku skóringu popsanou v kapitole 3.2.4. Druhou ikonou je ikona dolaru, která představuje ikonu výsledkové zprávy, popsanou v kapitole 3.2.5, na kterou se po klepnutí toto okno přesměruje.

3.2.3 Rozbalovací menu

Rozbalovací menu je typem responsivní navigace a je určeno pro rychlou navigaci mezi požadovanými okny. Rozbalovací proto, protože se rozvine (rozbalí) z levé strany obrazovky a překryje obsah.

Hlavní obrazovky (obrazovky vyjmenované v seznamu) mají v levém horním rohu aplikace zobrazen symbol menu neboli tři vodorovné čárky nad sebou, též označován jako “hamburger menu”. Po kliknutí na ikonu menu, se z levé strany zobrazí menu, které zobrazuje přehled hlavních obrazovek. Toto menu lze také vyvolat přejetím z levé strany obrazovky doprava. Po klepnutí na název požadované funkce se zobrazí její obrazovka.

V horní části menu, je vidět jméno přihlášeného uživatele následované přidělenými obrazovkami, a dále nastavení aplikace a odhlášení. Tlačítko odhlášení, uživatele odhlásí z aplikace, tedy odstraní token z paměti a vymaže veškerá uložená data o aplikaci.



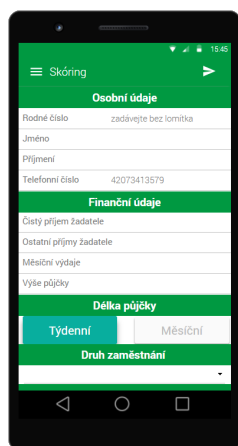
3.2.4 Skóringová zpráva

Po klepnutí na stránku skóringu je uživatel požádán o vyplnění veškerých údajů, které jsou potřeba pro správné oskórování zákazníka. To, že vyplněná pole jsou podle požadavků se ověřuje pomocí regulárního výrazu, který každé z těchto polí obsahuje. Pole jsou dynamická, takže jsou stahována přes webovou službu z databáze. Jedná se o pole, která budou textová, numerická, ANO/NE tlačítka s vlastním nastavením textu a výběrové menu z více možností. Pole jsou rozdělena do kategorií pro lepší přehlednost. Všechna jsou povinná.

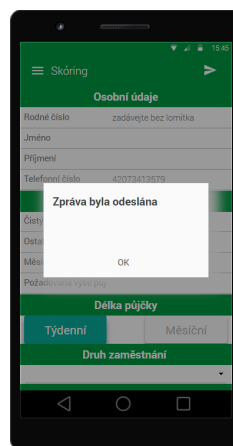
Obrázek 3.4: Rozbalovací menu

Po vyplnění veškerých polí (pokud je vše správně vyplněno), po stisknutí tlačítka “Skórovat”, nebo tlačítka složené papírové vložky v pravém horním rohu, se odešle zpráva sestavená dle požadavků a uživateli se zobrazí potvrzení o úspěšném odeslání zprávy. Pokud je uživatel připojen k internetu jakýmkoli způsobem, odešle se zpráva přes webovou službu, pokud ne, sestaví se do textové zprávy a odešle se.

V horním panelu uprostřed, je text s názvem okna, v levé části panelu je odkaz na menu 3.2.3.



(a) Vyplnění údajů

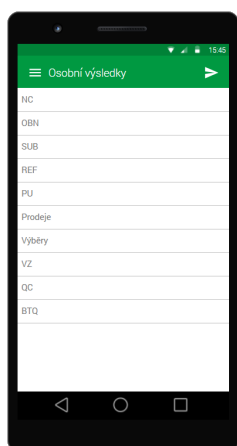


(b) Informace o odeslání

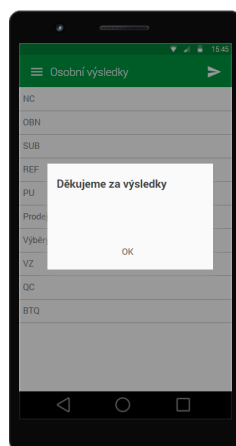
Obrázek 3.5: Obrazovky pro odesílání skóringu

3.2.5 Reportování

Reporting slouží k odesílání informací o denních výsledcích. Informace, která pole je povinen uživatel vyplnit jsou uložena v databázi a zobrazována jako řádky pod sebou. Každý řádek je jeden parametr pro vyplnění. Parametry pro vyplnění mohou být numerické, textové, nebo pole volby ze dvou, či více možností.



(a) Pole pro vyplnění



(b) Oznámení o odeslání

Obrázek 3.6: Obrazovky pro vyplnění a odeslání výsledků

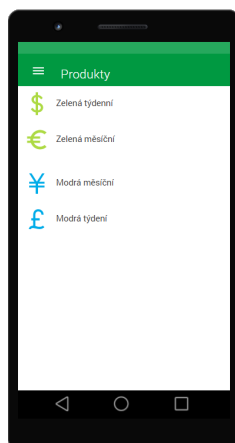
Na konci řádků pro vyplnění, bude tlačítko pro odeslání výsledků. Toto tlačítko má stejnou funkci jako tlačítko odeslat v pravém horním rohu (znak složené papírové vložky). Po stisku tohoto tlačítka, aplikace zajistí, že se odešle výsledková zpráva, v případě připojení k internetu přes webovou službu, v případě offline provozu se odešle textová zpráva v daném formátu. V případě správného odeslání zprávy, se zobrazí informace o odeslání výsledků. V případě neúspěchu se zobrazí informace, o tom, kde nastala chyba.

V horním panelu uprostřed, je text s názvem okna, v levé části panelu je odkaz na menu [3.2.3](#).

3.2.6 Produktová tabulka

Produktová tabulka obsahuje seznam půjček, které společnost aktuálně nabízí a agent je může nabídnout zákazníkovi. Tyto data se stahují přes webovou službu. Agent vidí jména půjček, jejich znak a barvu. Stránka nemá žádné přímé návaznosti na jiné stránky, je pouze zobrazovací.

V horním panelu uprostřed, je text s názvem okna, v levé části panelu je odkaz na menu [3.2.3](#).



Obrázek 3.7: Produktová tabulka

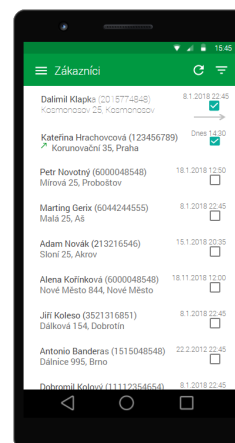
3.2.7 Přehled zákazníků

Jako další obrazovkou je zobrazení základních klientských informací. Klienti jsou zobrazeni v jednotlivých řádcích pod sebou. V řádku je zobrazeno jméno klienta a jeho osobní číslo v závorce. Pod jménem klienta je uvedena jeho adresa pro lepší orientaci. Vedle jména zákazníka je checkbox, který označuje, že detail zákazníka je v mobilním zařízení stažen pro použití offline. Zároveň je tento zákazník zachycen na vršek stránky. Datum nad checkboxem označuje datum posledního kontaktu zákazníka.

V horní části obrazovky jsou zobrazeny dvě tlačítka. Jedno aktualizací (zobrazeno jako stočená šipka sama do sebe), které aktualizuje základní informace o zákaznících. Aktualizace je na požádání při datovém připojení i při Wi-Fi připojení. Dále zde je zobrazeno tlačítko filtru (zobrazeno jako tři zmenšující se čárky pod sebou), které po stisku zobrazí vyhledávání zákazníka podle jména, osobního čísla a adresy. Toto vyhledávání není potřeba napsat přesně a celé, stačí napsat jen část a automaticky se vyhledají zákazníci, kteří tento text obsahují.

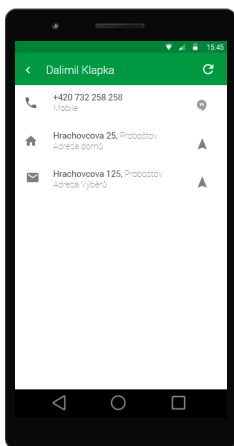
V horním panelu uprostřed, je text s názvem okna, v levé části panelu je odkaz na menu [3.2.3](#).

Každý zákazník obsahuje detail, kde se zobrazují další informace, potřebné pro agenta. K těmto informacím se uživatel dostane tak, že klepne na nějakého zákazníka a automaticky bude přesměrován na jeho detail popsaný v kapitole [3.2.8](#)



Obrázek 3.8: Přehled zákazníků

3.2.8 Detail zákazníka



Obrázek 3.9: Detail zákazníka

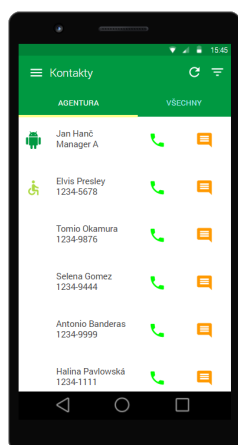
Do detailu zákazníka se dostane uživatel tak, že na kartě Zákazníci 3.2.7, klepne na jméno v řádku zákazníka, kterého detail chce zobrazit. Nová stránka s detaily zákazníka, se překryje přes tuto kartu. Zobrazené detaily na této stránce jsou: telefonní číslo, adresa domů a adresa kontaktní, což je adresa, kde se klient vyskytuje, kde ho má agent navštívit. Po stisku tlačítka vedle telefonního čísla vpravo, začne na toto číslo uživatel psát textovou zprávu. Po stisku tlačítka vpravo vedle adresy, se zobrazí tato adresa na mapě. Po klepnutí na telefonní číslo, aplikace toto telefonní číslo vytočí.

Tlačítkem v pravém horním rohu se aktualizují detailní informace o zákazníkovi, neboli se požádá o nové informace. Tyto informace mohou být zastaralé, pokud jsou v zařízení uloženy delší dobu. Tlačítko v levém horním rohu (šipka směrem doleva), se vrátí zpět na kartu Zákazníci, do hlavního postavení, na vrchol stránky.

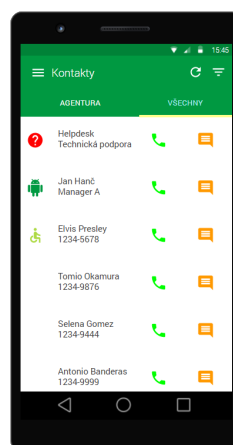
3.2.9 Kontakty

Na záložce Kontakty jsou vidět kontakty dvou druhů. Kontakty agentury (na obr. 3.10a) a všechny kontakty (na obr. 3.10b), které jsou agentovi k dispozici. V první záložce "Agentura", jsou kontakty z jeho agentury, do které spadá. Jako první se zobrazí jméno nadřízeného s jeho pozicí, druhý je zde uveden senior agent a další v pořadí jsou zobrazeny ostatní agenti, spadající do stejné agentury, jako přihlášený agent. V druhé záložce "Všechny kontakty" kde jsou navíc kontakty, které mu dopomohou v případě nějakých potíží, jako třeba přímé volání na helpdesk a další. Zároveň jsou zde zobrazeny kontakty z agentury, zobrazené pod těmito kontakty.

Ikona telefonu vedle každého kontaktu vytvoří okamžité volání na přidělené telefonní číslo. Ikona zprávy přesměruje uživatele do tvorby textové zprávy, kde předvyplní telefonní číslo kontaktu.



(a) Kontakty agentury

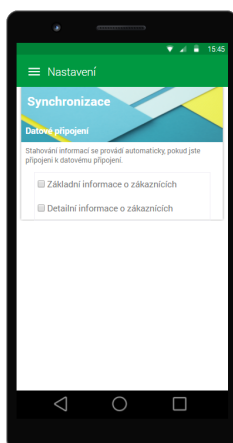


(b) Všechny kontakty

Obrázek 3.10: Obrazovka kontaktních informací společnosti

3.2.10 Nastavení

Na obrazovce nastavení si může uživatel zvolit nastavení aplikace. Jednotlivá nastavení zde mohou být vyjmenována, popsána a případně doplňována. Pro tuto verzi, budeme požadovat nastavení stahování informace a to konkrétně, jestli se budou stahovat automaticky informace o zákaznících, v případě připojení přes data. Pokud je zaškrtnuto první políčko „Základní informace o zákaznících“, automaticky se budou načítat informace o zákaznících bez upozornění, že uživatel je připojen k datovému připojení. To stejné se bude dít v případě detailních informací a zaškrtnutí druhého políčka „Detailní informace o zákaznících“. Uživatel tedy dále nebude upozorňován, jestli aplikace může tyto data synchronizovat.



Obrázek 3.11: Obrazovka nastavení

3.3 Testování s uživateli

Návrh aplikace byl vytvořen v programu „Justinmind Prototyper (FREE)“, který umožňuje návrh designu aplikace a přechody mezi jednotlivými obrazovkami. Toto vše mohlo dát dohromady alespoň hrubý návrh aplikace, která přechází mezi jednotlivými obrazovkami a mohli jsme dát vybraným budoucím uživatelům otestovat intuitivnost aplikace na jednotlivých, předem připravených, testovacích scénářích.

Na testování se podílelo pět budoucích uživatelů. Při testování se měřila rychlost nalezení správné cesty, jestli našli všechny cesty a jestli si všimli všech informací a tlačítek na stránce. Jako počáteční stránka byla vždy dána stránka po přihlášení, tedy "Úvodní obrazovka".

3.3.1 Vyplňte skóringové informace a odešlete zprávu

K našemu překvapení, většina uživatelů (4 z 5ti) neprocházela přes menu, které skrývá většinu z oken, ale přes tlačítko, které se nachází v pravém dolním rohu a přes něj prvním tlačítkem na skóring. Uživatelé dokázali najít tlačítko k vytvoření zprávy okamžitě, bez rozmýšlení.

Všichni uživatelé vyplnili zprávu správně. Všichni uživatelé, mimo jednoho, odeslali zprávu na konci stránky, pouze jeden uživatel si všiml a odeslal ji přes tlačítko v pravém horním rohu. Uživatelé, kteří odeslali zprávu přes tlačítko na konci stránky si nevšimli tlačítka v pravém horním rohu, nebo je nenapadlo, že se jedná o stejné tlačítko.

Lidé, kteří otevřeli zprávu přes tlačítko v pravém dolním rohu, pak také dokázali otevřít toto okno přes menu, ale uživatel, který otevřel prvně menu, nebyl schopen najít druhou cestu.

3.3.2 Otevření navigace k zákazníkovi

Uživatelé byli požádáni, aby zobrazili na mapě místo bydliště klienta, který bydlí v Aši. Uživatelé se přes tlačítko menu bez problémů dostali na stránku zákaznického přehledu. Všichni se na první pohled zorientovali na stránce. Pouze tři z nich, po klepnutí na zákazníka, se dostali na detail tohoto zákazníka. Zde se v první chvíli snažili klepnout na text s adresou, po chvíli si všimli znaku navigace a přes toto tlačítko, si její adresu zobrazili. Ostatní se snažili adresu přepsat do aplikace map. Všichni nakonec dokázali poklepat na tlačítko navigace, která by je (v případě funkčnosti) navedla na mapu.

Funkce zobrazení na mapě, by se měla přesunout na celé tlačítko, nejen pod ikonu navigace.

3.3.3 Zavolání na helpdesk

Uživatelé byli požádáni, aby s použitím aplikace našli kontakt na helpdesk a zavolali tam. Všichni neměli problém nalézt kartu s kontakty, ale problém nastal při hledání na kartě, kde kontakt na helpdesk naleznou. Pouze jeden uživatel bez problémů klepl na kartu se všemi kontakty a poté na tlačítko sluchátka u řádku "Helpdesk". Po zaváhání ostatních se podařilo i jim otevřít druhou záložku a také zavolat přes tlačítko sluchátka.

Tlačítko sluchátka je natolik intuitivní, že uživatelé neklikali na uživatele, ale přímo na tuto ikonu.

3.3.4 Nastavení automatického stahování základních informací o zákaznících přes datové připojení

V tomto případě byl jednoduchý úkol, který ve výsledku měl za cíl pouze nalézt stránku nastavení a zaškrtnout jedno políčko. Toto se ukázalo jako velký problém, neboť většina uživatelů neporozuměla otázce, ani tomu, proč by něco takového měli chtít nastavovat. Po detailním vysvětlení byli schopni všichni nalézt požadovanou stránku a zaškrtnout správné pole. S tímto úkolem měli uživatelé velký problém. Ukázal se jako nejméně intuitivní.

3.3.5 Odhlášení

Naopak nejvíce intuitivní pro uživatele bylo, odhlášení z aplikace. Všichni okamžitě otevřeli menu a stiskli tlačítko „Odhlásit se“.

3.3.6 Shrnutí

Při testování jsme zjistili, že aplikace potřebuje drobné úpravy, které nejsou vážné. Uživatelé při celkovém hodnocení hodnotili aplikaci jako přehlednou, chválili postranní menu, které uživatelům dopomáhá k lepší orientaci a ocenili tlačítko pro rychlé vytvoření zpráv.

Když si všimli kategorie nastavení, v první chvíli očekávali, že si zde mohou nastavit jazyk, barvy, zvuky, typy písma a další obecná nastavení. Nastavení, které ale aplikace umožňuje, jim nepřipadalo důležité i po vysvětlení.

Po vysvětlení další funkcionality, a to dynamického zobrazování polí ve skóringové zprávě, reportingu a v produktech, byli uživatelé nadšeni, protože se aplikace nadále nemusí aktualizovat při každé změně. Aktualizace aplikace vždy provázely nějaké problémy, tento přístup jim tedy přišel o hodně snesitelnější.

Uživatelé ocenili obrazovku kontakty, kterou do této chvíle neměli.

Připomínku měli k obrazovkám zákazníků, kde si stěžovali na částečnou nepřehlednost a nedostatek informací o zákazníkovi. Tyto informace nadále rozšíříme a upravíme řádkování pro větší přehlednost. Doplňme také popisky některých ikon, se kterými měli uživatelé problém. Některé ikony vyměníme za text.

Kapitola 4

Implementace

V této kapitole si popíšeme technologie, které můžeme použít a zvolíme si jednu z nich. Aplikaci vyvineme a poté popíšeme jednotlivé implementační kroky, které při vývoji aplikace nastaly. Popíšeme také jednotlivé obrazovky, jaké funkcionality se podařilo implementovat a jaké ne.

4.1 Technologie pro vývoj

Vzhledem k požadavkům na aplikaci, jakožto vývoj multiplatformní aplikace, jsme si mohli zvolit z nepřeberného množství hybridních technologií. Po výběru nejpoužívanějších s nejlepším hodnocením jsme vybrali tři technologie, které by byly vhodné pro implementaci.[4]

4.1.1 Nativní aplikace

Nativní aplikace je aplikace, která je vytvořena pouze pro jednu platformu (např. pouze pro iOS nebo pro Android). V případě, že ji chceme mít na jiné platformě, je nutné vytvořit jinou aplikaci, i když může mít shodnou funkcionalitu.

Výhodou je, že spolupracuje s hardware mobilních zařízení, může pracovat offline a umí lépe využít všech vlastností operačního systému - jsou tedy přirozeně rychlejší a mohou využívat dalších služeb nebo vlastností systému, jako jsou například pohybová gesta, a další.

4.1.2 Webová aplikace

Webové aplikace fungují obvykle na všech platformách jako přizpůsobené www stránky, tedy k provozu je potřeba pouze webový prohlížeč. Data se uchovávají na serveru, a tedy jsou přístupná odkudkoliv. Omezením bývá nižší možnost spolupráce s hardware mobilních zařízení a často je nelze používat offline. Jsou zde i možná bezpečnostní rizika úniku dat v případě nekvalitního poskytovatele.

4.1.3 Hybridní aplikace

Hybridní aplikace je kombinací webových a nativních technologií. Specializované nástroje umožňují vývoj pomocí k tomu určených webových technologií a následné konverze na nativní aplikace pro jednotlivé platformy.

Tyto aplikace jsou stále populárnější pro svou pohodlnost a rychlost vývoje a pro svou rychlost aplikace jako takové. Dají se vhodně použít pro práci s hardware telefonu, seznamy, propojení se sociálními sítěmi a další. Musí se počítat s pomalejšími časy pro složitější dotazy. Práce s audiem a videem také nepatří k silným stránkám technologie.

Vzhledem k potřebám, které vyžaduje tato aplikace, jsme si zvolili hybridní technologii k návrhu aplikace. Uvedeme si příklady frameworků a porovnáme, který je pro náš vývoj nejlepší.

4.1.3.1 Xamarin

Xamarin je platforma pro vývoj nativních mobilních aplikací pro platformy iOS, Android, Windows. Aplikace jsou programovány v jazyku C#/.NET, nebo jazyku F#. Xamarin umožňuje psaní kódu v jednom jazyce, ale také lze využít Xamarin.Forms, které umožní navrhnout UI pro více platform najednou.

Pro kompilaci využívá Just-in-time (JIT) nebo Ahead-of-time (AOT) kompilaci. JIT nelze použít pro iOS, tedy se využívá AOT. Na ostatních platformách může používat JIT nebo AOT dle nastavení. [7]

Tato platforma je rozšíření do Visual Studia, které v omezené funkcionalitě je Open-Source. Vývoj je rychlý a kompilace také, avšak vzhledem k C# kódu je často problém převést do platformy pro Android nebo iOS.

4.1.3.2 Ionic

Ionic je framework pro vývoj hybridních mobilních aplikací. Kód je psaný v Javascriptu/Typescriptu a také využívá Angular framework a HTML5. Pro zobrazování využívá web view, tedy si s sebou ponese vždy prohlížeč, ve kterém bude zobrazovat uživatelské rozhraní (UI) v HTML. Pro kompilaci využívá JIT. [7]

Bohužel aplikace napsané v Ionic frameworku se dlouho kompilují a v případě náročnějších (komplexnějších) aplikací a velkým nárokům na start aplikace, je běh samotný také pomalý.

4.1.3.3 React Native

React Native je framework pro vývoj nativních mobilních aplikací, který vyvinul Facebook. Hlavní výhodou technologie React Native je možnost vývoje pouze jednoho sdíleného kódu pro všechny platformy (iOS, Android, web). Kód je psaný v Javascriptu/Typescriptu a pro UI využívá speciální templovací jazyk, kterého je výsledkem vytvoření nativních widgetů. React dokáže detekovat změny a poté překreslit pouze je. Toto může čas kompilace urychlit. [8]

Nespornou výhodou pro vývoj v React Native, je použití velkého množství knihoven, které jsou zdarma a open source. Přidáním knihovny si můžeme nadále rozšířit možnosti s aplikací. Jako typickým příkladem knihovny, která nabízí spoustu komponent, které jsme dále použili v aplikaci, je knihovna Native Base. Tato knihovna nabízí spoustu komponent, které se v aplikaci dají použít a jsou pohodlné na použití, funkční a designově hezké. Příkladem takového kódu je 4.1.

Listing 4.1: Základní syntaxe NativeBase komponenty

```
import React, { Component } from 'react';
import { Container, Button, Text } from 'native-base';

export default class GeneralExample extends Component {
  render() {
    return (
      <Container>
        <Button>
          <Text>
            ButtonText
          </Text>
        </Button>
      </Container>
    );
  }
}
```

První dva řádky určují, jaké knihovny se použijí (react a native-base) a jaké se použijí komponenty z knihovny (Container z native-base). Tím se definují a předpokládá se jejich použití v kódu. Třída musí obsahovat "render" metodu. Tato metoda určuje, co se bude zobrazit. Co se bude zobrazovat určí v této metodě return, který v závorkách obsahuje výsledek. V returnu jdou vidět použité komponenty, tedy Container, který obsahuje tlačítko (Button), ve kterém je text (Text), ve kterém se text "ButtonText" zobrazí na tlačítku.

Určitě stojí za zmínku i dynamické vykreslování, které nám dále pomůže dynamicky zobrazovat to, co nám přijde v požadavku na vykreslení. Toto vykreslení by mělo být dynamicky rozšiřitelné, tedy je potřeba nějaká funkce, která bude bez ohledu na velikosti, zobrazovat jeho obsah. React Native má tu výhodu, že je to javascriptová aplikace, která umí využívat tuto syntaxi k dalšímu vykreslování. Pokud máme dohodnutý formát v JSON, který nám přijde v odpovědi na požadavek, a máme strukturu do které se obsah bude vykreslovat, lze použít pro dynamické vykreslování funkci *map*. Tato funkce dokáže při zavolání nad objektem vrátit přesně definovanou strukturu pro každý záznam.

Příklad kódu 4.2 je dynamické vykreslování řádků do mřížky. V prvním řádku jde vidět prvotní načtení .json souboru, ve kterém je uložen tento záznam, obsahující tři textové řetězce. JSON struktura se převede do objektu, a pak se s ním dále pracuje jako s dynamickým objektem a lze na něj odkazovat. Na takto načtený soubor se zavolá funkce *map*, a jako dva parametry jsou objekt, který to iterovalo (*item*) a index té iterace (*i*). Tyto proměnné se poté použijí ve funkci, která vytvoří nové objekty do tagů a do nich vloží tuto hodnotu. Aby bylo patrné, že se jedná o proměnné, musí se vložit do složených závorek. Tato syntaxe vykreslí jednotlivé řádky mřížky.

Listing 4.2: Dynamické vykreslování obsahu funkcí *map*

```
const example = require("example.json");
...
render() {
  return (
    <Content>
      <Grid>
        {
          example.map((item, i) => {
            return (
              <Row key={i} >
                <Col><Text style={styles.firstColumn}>{item.text1}</Text></Col>
                <Col style={styles.column}><Text>{item.text2}</Text></Col>
                <Col style={styles.column}><Text>{item.text3}</Text></Col>
              </Row>
            )
          })
        }
      </Grid>
    </Content>
  );
}
```

Rozhodně stojí za to si povšimnout stylů. Tyto styly jsou definované mimo tento kód, jako je tomu například u kaskádových stylů. React používá objektového přístupu, tedy ke stylu lze přistoupit jako k objektu. Tyto styly jsou definované v typescriptovém souboru, který je poté využíván. Příkladem takových stylů je 4.3, který definuje styl sloupce.

Listing 4.3: Příklad nastavení stylů

```
export default StyleSheet.create({
  column: {
    height: 24,
    alignItems: 'center'
  },
  firstColumn: {
    fontWeight: 'bold',
  }
});
```

Ikdyž framework je poměrně nový, má před sebou slibnou budoucnost. Časem své kompilace vysoce převažuje ostatní frameworky a rychlost jeho vývoje je poměrně vysoká, protože jsou potřeba pouze základní znalosti o tom, jak celý framework funguje, a tedy jsme si jej zvolili jako framework pro naši aplikaci.

4.2 Prostředí pro vývoj

K vývoji aplikace v React Native, je potřeba si zvolit vhodné vývojové prostředí a nainstalovat potřebné komponenty, které jsou nutností pro vývoj. Díky dobré dokumentaci na webových stránkách React Native, je instalace snadná. Je zde dobře popsána krok po kroku, někdy se zde zmiňují i chybová hlášení, která mohou nastat, jak je opravit.

4.2.1 Node.js

Pomocí programu Node.js jsme schopni spouštět příkazy v příkazovém řádku, které nám zjednoduší instalaci spousty doplňků. Pokud chceme výsledný kód spustit, je k tomu vyloženě nezbytný. Tato služba je nejobsáhlejší package službou, tedy není divu, že ji při kompilaci využívá.

4.3 Android Studio

Pokud vyvíjíme pro Android, Java a Android Studio je nutností, neboť překompilovaný kód je přeložen do jazyka Java a poté spuštěn na simulátoru Android Studia. Tyto dvě

aplikace jsou tedy nutné, pokud chceme vytvořit .apk soubor, který nadále nahráváme do mobilních zařízení.

4.3.1 Xcode

Pokud chceme, aby nám aplikace fungovala i na zařízení iOS, potřebujeme nejen správné softwarové nástroje, ale také hardwarové. Bohužel kód, který naprogramujeme na zařízení s Windows nebo Linux, nelze testovat, ani kompilovat pro zařízení užívající operační systém iOS. K tomu je potřeba zařízení od firmy Apple, do kterého lze nainstalovat program Xcode, díky kterému dostaneme simulátor iPhone, na kterém můžeme provádět testování. Na těchto zařízeních můžeme kód kompilovat do zařízení jak iOS, tak Android.

4.3.2 Visual Studio Code

Program Visual Studio Code od firmy Microsoft, se pro vývoj aplikace jeví jako vynikající volba, kde se dá psát nezbytný kód. Tento program má možnost nainstalování doplňků, které práci s kódem ulehčí, a ještě před samotnou kompilací upozorní na chyby, které zde programátor zanechal. Výhodou aplikace je GUI, které zvýrazňuje a barví části kódu, tedy je poté přehlednější.

4.4 Server

Vzhledem k tomu, že v této diplomové práci je řeč o mobilní aplikaci, která je napojena na backend, je potřeba popsat i tuto část, která bude s aplikací komunikovat. Na backendu se nachází IIS, běžící na Windows server 2008 R2, na které jsou nasazeny dvě aplikace v DMZ zóně, které jsou vystaveny pro uživatele. Aplikace jsou naprogramovány v jazyku C# ASP.NET a vystavují REST služby, které zpracovávají JSON požadavky a vrací také JSON odpovědi. Tyto aplikace jsou rozděleny do dvou částí, dle možných přístupů a dle toho, co dělají. Ty si detailněji popíšeme.

4.4.1 Autorizační aplikace

Tato aplikace má na starosti ověření uživatele, zda je autorizovaný používat firemní aplikace. Postup ověření je popsán detailně v kapitole [4.5.5](#). Aplikace vystavuje pouze jeden endpoint.


```
POST /oauth2/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
BODY
  username: ID uživatele
  password: Telefonní číslo uživatele
  grant_type: Typ přístupových údajů do aplikace
  client_id: ID aplikací a registrační název tohoto endpointu
  type: no2f/2f
  validationText: Text z SMS zprávy
  validationToken: Token, který se vrátí na první zavolání
```

Toto volání je provedeno dvakrát. Při prvním volání, jako nový uživatel, se odesílají pouze parametry `username`, `password`, `grant_type`, `client_id` a `type`, kde v `type` je `no2f`. Ve druhém volání, jsou navíc parametry `validationText` a `validationToken` a `type` se změní na `2f`.

První odovědí na první požadavek je validační token, ve druhé odpovědi je bearer JSON Web Token (JWT), který zaručuje ověření uživatele.

4.4.2 Enterprise aplikace

Hlavní aplikace složená z mikroservis, kde každá obsahuje jednu podmnožinu endpointů na získání informací z databáze a ze serveru. Každá mikroservisa poskytuje informace, nebo zpracovává odeslané informace do databáze, či pro jiné systémy. Tyto mikroservisy si popíšeme.

Předpokládejme pro všechny dotazy, které si tu budeme popisovat, že každý z nich obsahuje v Headers položku „Authorization: bearer eyJfakf..ss01“. Tato položka obsahuje za slovem bearer token, který obdrží uživatel po ověření přes autorizační aplikaci. Tento token je tzv. JWT token, který zabezpečuje komunikaci mezi dvěma stranami a obsahuje číslo agenta, které je použito na získávání specifických informací pro přihlášeného agenta.

4.4.2.1 API Základních informací

Mikroservisa, která poskytuje pouze jeden endpoint, který poskytuje základní informace o uživateli.

```
GET /basic HTTP/1.1
```

Listing 4.4: Příklad odpovědi na požadavek

```
{
  "firstName": "Jan",
  "lastName": "Hanč",
  "menuitems": ["scoring", "dailyMessages", "customers", "
    toc", "settings", "contacts"]
}
```

4.4.2.2 API obchodních výsledků

Mikroservisa na předávání obchodních výsledků získává data z databáze agenta a zobrazuje jeho průběžné výsledky.

```
GET /results HTTP/1.1
```

Listing 4.5: Příklad odpovědi na požadavek obchodních výsledků

```
[
  {
    "name": "NC",
    "plan": "24",
    "real": "2"
  },
  ...
]
```

4.4.2.3 Skóringové API

Tato služba vrací popis skóringové textové zprávy, kterou musí uživatel vyplnit do aplikace. Pole pro vyplnění zprávy jsou připravována ze strany obchodu a vkládána do databáze, kde se je aplikace bere.

GET /scoring HTTP/1.1

Listing 4.6: Příklad odpovědi na vzor skórovací zprávy

```
{
  "Scoring": {
    "formName": "Scoring",
    "fields": [
      {
        "type": "inputGroup",
        "name": "PERSONAL_DATA",
        "label": "Osobní informace",
        "fields": [
          {
            "name": "phoneNumber",
            "label": "Telefonní číslo",
            "type": "text",
            "keyboardType": "phone-pad",
            "placeholder": "+420734123456",
            "validators": [
              "required",
              {
                "regexp":
                  "^\\+420[0-9]{9,9}$",
                "errorMessage": "Zadané telefonní číslo není
                  platné (regex validace)"
              }
            ],
            "normalizer": {
              "name": "phoneNumberNormalizer"
            }
          },
          ...
        ]
      }
    ]
  }
}
```

Druhým dotazem na tuto službu je odeslání vyplněných polí na server. Toto je odesíláno jednoduše jako klíč-hodnota, odpověď je HTTP kód 200 bez textu, nebo 400 s prostým textem chyby v těle odpovědi.

Listing 4.7: Příklad požadavku odeslání skóringu

```
POST /scoring HTTP/1.1
{
    "firstName": "Karel",
    "lastName": "Hromada",
    "dob": "15.5.2000",
    ...
}
```

4.4.2.4 Zákaznické API

Tato část je důležitá pro ty agenty, kteří mají povoleno vidět seznam svých klientů. Prvním dotazem je zobrazení části klientů a jejich základní informace. Parametr si je start index neboli kolik zákazníků je již načteno. Nula je tedy index prvního zákazníka. Parametr count je počet zákazníků, v odpovědi.

```
GET /customers/basic?si=0&count=20 HTTP/1.1
```

Listing 4.8: Příklad odpovědi na získání základních informací o zákaznících

```
[
  {
    "fullName": "Jan Haramia",
    "section": "62739273",
    "address": "Kosmonosy 234, Kosmonosy",
    "lastVisit": "15.2.2018 11:30"
  },
  ...
]
```

Druhý dotaz je na detailní informace o zákazníkovi. Jako jediný parametr je id zákazníka, pod kterým je uložen v databázi

```
GET /customers/detail?id=64565421 HTTP/1.1
```

Listing 4.9: Příklad odpovědi na získání základních informací o zákaznících

```
{
  "section": "62739273",
  "address": "Doubravská 23, Rahlovice",
  "agreementNumber": "1515151351",
  "phone": "+420734839007",
  "dateOfAgreement": "15.05.2017"
}
```

4.4.2.5 Reportovací API

Někteří agenti také musí reportovat svým nadřízeným své výsledky. Tato služba jim vrátí strukturu zprávy, kterou je potřeba vyplnit.

```
GET /report HTTP/1.1
```

Listing 4.10: Příklad odpovědi na vzor reportovací zprávy

```
{
  "DailyMessage": {
    "formName": "DailyMessage",
    "fields": [
      {
        "name": "nc",
        "type": "number",
        "label": "NC",
        "validators": [
          "required",
          "numeric"
        ]
      },
      ...
    ]
  }
}
```

Další dotaz je odeslání vyplněných polí zpět serveru. Toto je podobné skóringové zprávě. Jako odpověď, pokud je vše v pořádku, je opět použit HTTP kód 200 bez textu, jinak kód

400 s chybovým hlášením.

Listing 4.11: Příklad požadavku odeslání reportingu

```
POST /report HTTP/1.1
{
    "nc": 15,
    "res": 21,
    "pu": "11",
    ...
}
```

4.4.2.6 Produktové API

Toto api je rozděleno na část zjišťovací a na část datovou, kde část zjišťovací vrátí pouze informaci o tom, jestli uživatel má nové produkty (odpověď true), nebo ne (odpověď false). K tomu slouží následující požadavek, ve kterém jediný parametr je hash, který je vypočítaný pro tuto produktovou tabulku.

```
GET /products?hash=e5sf4a5f4as5f4as4a5s4f5 HTTP/1.1
```

Druhý dotaz je na poslání produktové tabulky.

```
GET /products/new HTTP/1.1
```

Listing 4.12: Příklad odpovědi na produktové tabulky

```
{
  "Zelená": [
    {
      "key": 11,
      "iconName": "md-dollar",
      "name": "Zelená týdenní"
    }
  ],
  "Modrá": [
    ...
  ]
}
```

4.4.2.7 API kontaktů

Kontakty jsou přijímány ve dvou skupinách, první specifické pro každého agenta, druhá pro všechny agenty společná. Každý z těchto seznamů obsahuje jméno a telefonní číslo.

```
GET /contacts HTTP/1.1
```

Listing 4.13: Příklad odpovědi s kontakty pro agenta

```
{
  "agency": [
    {
      "key": 1,
      "iconName": "logo-android",
      "fullName": "Jan Hanč",
      "position": "Manager",
      "phoneNumber": "123456789"
    },
    ...
  ],
  "other": [
    {
      "key": 1001,
      "iconName": "md-help",
      "fullName": "Helpdesk",
      "position": "Technická podpora",
      "phoneNumber": "+420123456789"
    },
    ...
  ]
}
```

4.5 Řešení nefunkčních požadavků

Popíšeme si nefunkční požadavky, které aplikace musí splňovat, bez interakce s uživatelem.

4.5.1 Jazykové překlady

Vzhledem k tomu, že se jedná o mezinárodní aplikaci, tedy tvořenou pro český, maďarský, rumunský a polský Provident, musely být zajištěny překlady do těchto jazyků. Vzhledem k tomu, že funkcionality byla stejná, díky komponentě `react-i18next` bylo možné jednoduchým trikem provést všechny překlady. Jazyk překladu závisí na jazykovém nastavení telefonu. Pokud překlad pro tuto zemi neexistuje, je základním jazykem angličtina.

Vytvořily se tedy tři `.json` soubory, které obsahují totožnou strukturu, na kterou v textu stačí pouze poukázat a zobrazí se její obsah.

4.5.2 Ukládání dat do mezipaměti

Díky tomu, že React Native už sám od sebe umí ukládat data do mezipaměti, bylo výhodné toto využít a ukládat data to tzv. `AsyncStorage`, která na základě klíče, umí uložit, nebo také přečíst zadanou hodnotu. [1]

4.5.3 Synchronizace

Veškeré dotazy, které jsou směrem na server, probíhají pomocí REST komunikace přes HTTP protokol, popsané v kapitole 4.4.

4.5.3.1 Běh aplikace

Tyto informace jsou staženy při prvním přihlášení do aplikace. Jsou to informace o uživateli, na jaké je pobočce a jaké má uživatel dovoleno používat obrazovky.

4.5.3.2 Zobrazení základních dat

Základní data nejsou nutná pro běh aplikace. Zde se dostávají informace jako obchodní výsledky agenta, struktura skóringové zprávy, struktura reportovací zprávy, seznam zákazníků a kontakty. Tyto informace jsou základní, ale budou vyřízeny hned po informacích důležitých pro běh aplikace.

4.5.3.3 Zobrazení detailních dat

Detailní informace jsou informace, bez kterých aplikace poběží, avšak se nestane to, že by uživatel neměl cokoli k dispozici. To jsou například detailní informace o každém ze zákazníků, nebo produktová tabulka. Tyto informace jsou stahovány až v poslední fázi, nebo na požádání.

4.5.4 Logování

Dle návrhu řešení pomocí Firebase a detailnější analýze, bylo rozhodnuto, že se tato funkcionálníta nebude používat, vzhledem k tomu, že se jedná o placenou funkcionálnítu. Nahrádou za tuto funkcionálnítu, jsme naprogramovali logger pomocí mezipaměti a REST požadavků.

V aplikaci jsme přetížily metody pro logování do konzole, pro které jsme implementovali funkcionálnítu ukládání do mezipaměti ve formátu JSON. Jakmile logger shromáždí potřebná data, odešle je na server bez interakce s uživatelem, jakmile bude server přístupný.

4.5.5 Zabezpečení

Pokud si aplikaci nainstaluje třetí osoba, která například našla aplikaci v obchodu, je zajištěno, aby se do aplikace nedostala bez řádného prověření. Pro vstup do aplikace je potřeba znát dva druhy údajů, a to osobní číslo agenta a jeho registrované telefonní číslo. Pokud se toto potvrdí, následuje druhé ověření, a to ověření na základě odeslání textové zprávy na telefonní číslo agenta, který jej musí zadat do aplikace.

Dvoufázové zabezpečení je proces, který ověřuje totožnost uživatele právě dvěma nezávislými způsoby. V našem případě se jedná o ověření pomocí HTTP požadavku a o ověření pomocí textové zprávy. V moderní době se může využít hardwarových možností telefonu a ověření může probíhat pomocí otisku prstu, nebo testu oční duhovky. Toto ale v naší aplikaci nepoužíváme. [6]

Pro ukládání dat v telefonu se využívá AsyncStorage, kde se ukládají jednotlivé parametry jako klíč-hodnota. Tyto informace by byly jednoduše zneužity, proto jsme použili zabezpečení typu AES-256, kterým tyto záznamy zašifrujeme a uložíme pod jeden velký objekt. Tyto data nesmí být zneužita, pokud se útočník pokusí dostat do mobilního zařízení a zobrazit si tyto informace. Pokud se uživatel odhlásí, tyto informace jsou smazány z paměti.

4.5.6 Odesílání textových zpráv a přímé volání

K odesílání textových zpráv a k přímému vytočení hovoru, se s výhodou může využít komponenta v balíčku „react-native-communications“, která se nainstaluje pomocí Node.js. Komponenta se dá dále použít k odeslání emailů a otevření webové stránky. V budoucnu by se tyto funkcionality mohly hodit, tedy tuto komponentu považujeme za nejvhodnější volbu.

V porovnání jsme také měli komponenty z balíčku „react-native-sms-x“, ale balíček umí pouze odesílat textové zprávy a neumí vytáčet telefonní čísla, tedy se jeví jako horší varianta.

4.5.7 Provázanost na jiné aplikace

Hlavním požadavkem na aplikaci byla provázanost na aplikaci map. Jako solidní doplněk se jeví použití balíčku „react-native-maps“, která umí zobrazovat místa na mapě, zobrazit větší množství bodů, označení regionů a další. Bohužel jednou z nevýhod tohoto rozšíření, je zobrazení na mapě podle adresy. Toto rozšíření umí pouze zobrazovat podle GPS souřadnic. Zeměpisná šířka a zeměpisná délka se zadává v decimálním formátu (například 40.91599, 14.151644). Tedy pokud jsou tyto souřadnice dostupné ze serveru, požadavek na informace o zákazníkovi je odešle. Pokud na serveru nejsou k dispozici, aplikace si je musí zjistit sama. K tomu využije Google API.¹

Toto api slouží k zjištění GPS souřadnic, které vystavuje Google. Na GET požadavek s parametrem „address“ v URL, vrátí JSON, který obsahuje objekt „geometry“, ve kterém jsou uloženy souřadnice v objektu „location“. Toto API má také omezení. Tím je počet požadavků. Toto api umožňuje pouze 10 požadavků na server za minutu a 2500 požadavků za den. [3]

Požadavek na zobrazení adresy na mapě, po klepnutí na pole s adresou zákazníka, jsme bohužel nedokázali vyhovět. Navrhované rozšíření bylo instalováno, ale pokus o zobrazení na mapě se nezdařil. Aplikace neodpovídala a způsobovala další potíže. Po dohodě s obchodem byla tato funkcionality upravena na zobrazení na mapě ve webovém prohlížeči pomocí map Google.

4.6 Řešení funkčních požadavků

V této části si popíšeme aplikaci a splnění jejich požadavků. Také si popíšeme jednotlivé vlastnosti obrazovek a popíšeme si požadavky, jaké jsou na backend.

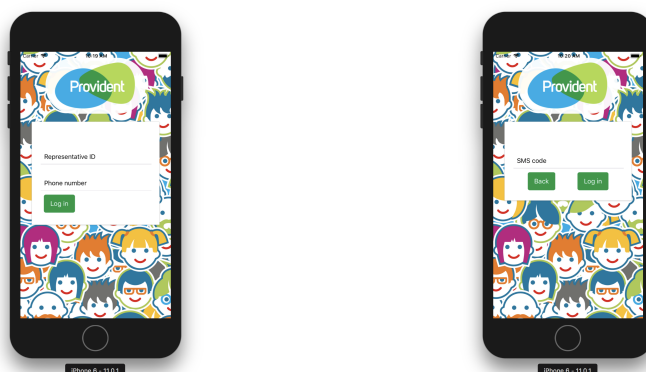
4.6.1 Přihlášení

Dle požadavků aplikace při prvotním zobrazení, nebo při odhlášení, zobrazí dvě pole. Viz obr. 4.1a. První je ID uživatele a druhé telefonní číslo, které uživatel musí vyplnit. Ověření probíhá na serveru, kam se tyto dva údaje odesílají. Požadavek se odešle na autorizační server po stisknutí tlačítka „Přihlásit se“, popsané v 4.4.1.

Uživatel se správnými údaji je ověřen a jako odpověď na požadavek je autorizační klíč, který se musí právě pro druhou autentizaci použít jako jeden z dalších parametrů. Druhou cestou úspěšného ověření je textová zpráva, kterou uživatel obdrží na své telefonní číslo, pokud bylo dobře vyplněno. Textová zpráva obsahuje šestimístný kód.

Druhá obrazovka 4.1b, obsahuje pole pro zadání kódu, který uživatel obdržel v textové zprávě. Pokud uživatel stiskne tlačítko „Zpět“, vrátí se na první přihlašovací obrazovku.

¹<<https://maps.googleapis.com/maps/api/geocode/json?address=Zikova4,Praha>>



(a) Základní přihlášení

(b) Ověřovací číslo

Obrázek 4.1: Obrazovky pro přihlášení do aplikace

Po stisku tlačítka „Přihlásit se“, se opět odešle stejný požadavek na server. Do požadavku se přidaly dva údaje. První údaj uživatel obdržel ve zprávě, druhý aplikace obdržela jako odpověď na první požadavek. Úspěšná odpověď na tento požadavek vrátí tzv. „bearer“ JWT token. Tento kód se používá při dalších dotazech na backend, tedy všechny dotazy na backend, které si budeme dále popisovat, jej obsahují jako Autorizační token.

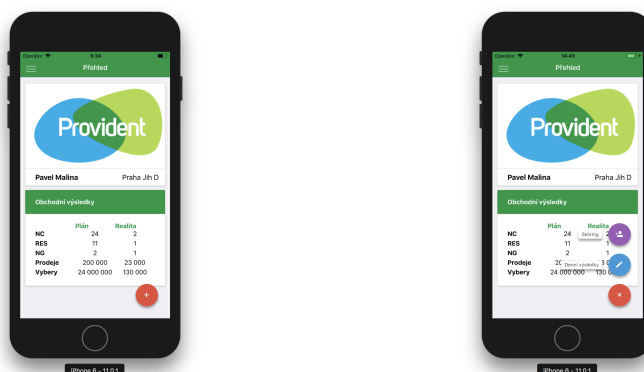
Pokud se jedná o neúspěšné přihlášení, tedy v prvním kroku uživatel zadal chybné údaje, nad textem se zobrazí červená chybová hláška ze serveru, který údaj byl chybně vyplněn, případně jiná chybová zpráva, kterou server odpověděl. Pokud odpověď z nějakého důvodu nepřišla, jako základní chybovou hláškou je žádost o kontaktování helpdesku. Stejně tak tomu je i ve druhém kroku po zadání chybného SMS kódu. Chybová zpráva se zobrazí nad polem k zadání kódu.

4.6.2 Úvodní obrazovka

Po úspěšném přihlášení se zobrazí úvodní obrazovka, na které jsou vidět dvě hlavní části. Logo s informacemi o přihlášeném uživateli a obchodní výsledky uživatele, jak si stál v účtovacím období. V levém horním rohu je tlačítko, tzv. „hamburger“, který rozbaluje menu popsané v kapitole 4.6.3.

K zobrazení základních informací je potřeba požadavek na server 4.4.2.1. Tento požadavek vrátí informace o uživateli, jako je jeho jméno, pozice a nastavení pro aplikaci. Jméno a sekce se vykresluje v horní části pod logem společnosti.

V případě požadavku na vyplnění skóringu, nebo vyplnění denních výsledků, uživatel může využít rozbalovacího tlačítka v pravém spodním rohu s ikonou plusu. Toto tlačítko po stisknutí rozbílí dvě možnosti. Po klepnutí na tyto tlačítka, uživatel se dostane přímo k vyplnění těchto zpráv.



(a) Úvodní obrazovka

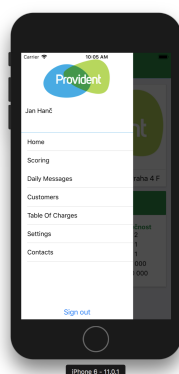
(b) Výběr nové zprávy

Obrázek 4.2: Úvodní obrazovky po přihlášení

Druhý požadavek, který se odesílá, je požadavek na obchodní výsledky (4.4.2.2). Úspěšná odpověď obsahuje pole obsahující objekty se třemi hodnotami, které jsou vykreslované do tabulky. Vždy se jedná o jméno, plán a realitu. Díky dynamickému vykreslování nezáleží na počtu objektů v poli, tedy lze jednoduše na backendu přidat objekt a na frontendu se vykreslí. Není třeba přenasazovat mobilní aplikaci.

V případě špatného dotazu, nebo nedostupnosti výsledků se ze serveru vrátí jiná odpověď, než 200 OK, se obsah nebude zobrazovat. Pokud server nerozpozná uživatele, server vrátí prázdnou hodnotu (null) a obchodní výsledky se nebudou zobrazovat.

4.6.3 Rozbalovací menu



Obrázek 4.3: Rozbalovací menu

Toto menu na obrázku 4.3 je takovým srdcem aplikace. Z něj se lze dostat prakticky

do všech částí aplikace pouze jedním kliknutím. Zde se zobrazuje seznam všech přístupných obrazovek, ke kterým má uživatel přístup.

Do menu se lze dostat ze všech obrazovek přejetím přes levý okraj obrazovky zařízení směrem vpravo. Jinak lze zobrazit menu z obrazovek (pokud ta možnost na obrazovce existuje), v levém horním rohu klepnutím na znak menu.

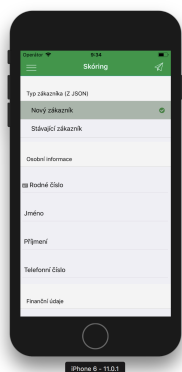
V horní části obrazovky pod logem je jméno uživatele, který je přihlášený, a poté následuje seznam obrazovek. Ve spodní části menu je tlačítko na odhlášení z aplikace.

V prvotním požadavku po přihlášení, ve 4.4.2.1, je v odpovědi seznam obrazovek (funkcionalit), které jsou přihlášenému uživateli přístupné. Seznam obrazovek je přidělen podle druhu přihlášeného uživatele a podle nastavení společnosti na backendu.

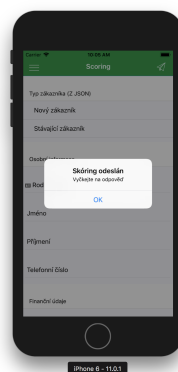
4.6.4 Skóringová zpráva

Skóring je základní funkcionalitou nezbytnou pro každého uživatele. Touto zprávou si agent ověří, jestli zákazník má nárok na půjčku. Zpráva je rozdělena do několika kategorií a možnosti pro vyplnění je několik. Proto byl vytvořen vzor pro pole, které se zobrazují. Vzorek obsahuje několik druhů polí, které jsou možné nastavit, jako textové pole, číselné pole, ANO/NE tlačítko a další.

Po přihlášení do aplikace, v požadavku na server (4.4.2.3), je doručeno nastavení pro skóringovou zprávu. Toto nastavení obsahuje informace ve strukturované podobě, o jaký druh pole se jedná, jaká klávesnice má být zobrazena a také regulární výraz, dle kterého se vložené pole ověřuje. Výsledkem toho jsou vložená pole pod sebe do formuláře pro skóringovou zprávu. Příklad takto připravených polí je vidět na obrázku 4.4a.



(a) Vyplnění polí



(b) Odeslaná zpráva

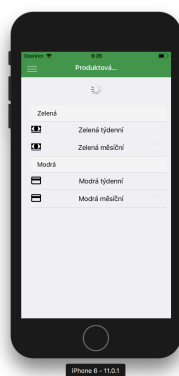
Obrázek 4.4: Obrazovky pro vyplnění a odeslání skóringu

Na konci stránky je tlačítko pro odeslání vyplněných údajů. V pravém horním rohu je toto tlačítko také. Tyto údaje se odesílají pomocí POST požadavku na server (4.4.2.3), který

údaje zpracuje a odešle k ověření. Po odeslání skóringu se zobrazí na obrazovce oznámení o úspěšném odeslání k ověření, jako je vidět na obrázku 4.4b. Odpověď s nabídkou přijde později pomocí textové zprávy na telefonní číslo agenta. V případě momentální nedostupnosti internetového připojení, je zpráva sestavena do textové podoby a odeslána textovou zprávou, jako to bylo v první verzi aplikace.

4.6.5 Produktová tabulka

Produktovou tabulkou se rozumí výčet produktů, které jsou aktuálně k dispozici. Vzhledem k tomu, že se produkty mění v nepravidelných intervalech, bylo zapotřebí tyto informace sdílet také z backendu. K tomu slouží požadavek na získání těchto informací. Tyto data jsou pro všechny uživatele stejná.

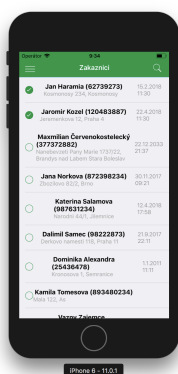


Obrázek 4.5: Produktová tabulka

Vzhledem k tomu, že někdy se stane, že produktů a informací k nim je více, je tedy přenášeno velké množství dat, bylo zapotřebí tyto data omezit. Řešením byl hash produktů, který se v pravidelných intervalech vypočítává na backendu a tento hash je ověřován při dotazu o nové produkty. Aplikace v sobě uchovává poslední známý hash produktů, které se zobrazují. Tento hash je vždy doručen s novými produkty a vše je uloženo v lokální paměti. Při inicializaci je tento hash prázdný, tedy aplikace při prvním spuštění neobsahuje žádné produkty. Teprve po prvním dotazu s prázdným hash se stáhne aktuální s aktuální produktovou tabulkou. Někdy uživatel nechce také plýtvat data na stažení těchto produktů, tedy po ověření je upozorněn na novou verzi a dotázán, zda chce tyto data stáhnout. Příklady požadavků jsou v kapitole 4.4.2.6.

4.6.6 Přehled zákazníků

Tento seznam zobrazuje zákazníky, kteří spadají pod přihlášeného uživatele. Každý zákazník má na levé straně checkbox, který značí to, jestli zákazník je k dispozici offline. Uprostřed je tučně napsané jeho jméno a v závorce zákaznické číslo a pod ním adresa jeho trvalého bydliště pro rychlejší orientaci. V pravé části je datum poslední ko kontaktu, neboli datum, kdy byl zákazník naposledy navštíven, nebo telefonicky kontaktován. Toto datum může být jakákoli kontrola, či výběr splátky. Při klepnutí na zákazníka v seznamu, se zobrazí jeho detail popsany v části 4.6.7.



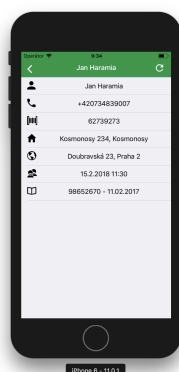
Obrázek 4.6: základní přehled zákazníků

Obsah je stahovaný dynamicky, tedy při každém odrolování na konec zákazníků se načtou se ze serveru další. Požadavek je popsany v kapitole 4.4.2.4. Na počátku seznamu jsou ti klienti, kteří jsou v stažení offline a není třeba je stahovat ze serveru. V pravém horním rohu je zobrazena lupa, která umožňuje vyhledávání v seznamu zákazníků, který je offline, tak online. Slouží to hlavně k rychlému nalezení potřebných klientů a případnému uložení detailů. Bohužel se nám tato funkce nepodařila implementovat a po dohodě s obchodním oddělením, kvůli splnění časového plánu a nevelkému počtu zákazníků v seznamu (jednotky až desítky), byla tato funkcionalita posunuta do příští verze.

4.6.7 Detail zákazníka

Pokud se v seznamu zákazníků klepne na zákazníka, zobrazí se jeho detail. V detailu jsou informace, které jsou potřebné ke kontaktování tohoto zákazníka, tedy jeho telefonní číslo, adresa a zároveň datum posledního kontaktu. Tato část se může rozšířit dle požadavků z obchodu. Je potřeba pouze doplnit pole, které se mají zobrazovat.

Pokud je v přehledu zaškrtnut checkbox, který označuje, že je zákazník offline, je i detail zákazníka načten offline, není třeba požadavku na server. Pouze v případě, že zákazník není



Obrázek 4.7: detailní informace o zákazníkovi

offline, nebo uživatel si přeje aktuální data a klepne v pravém horním rohu na aktualizací šipku, vyšle se požadavek na server. Požadavek je v sekci 4.4.2.4, který v sobě obsahuje číslo zákazníka. Jako odpověď se vrátí informace, které se mají o zákazníkovi zobrazit.

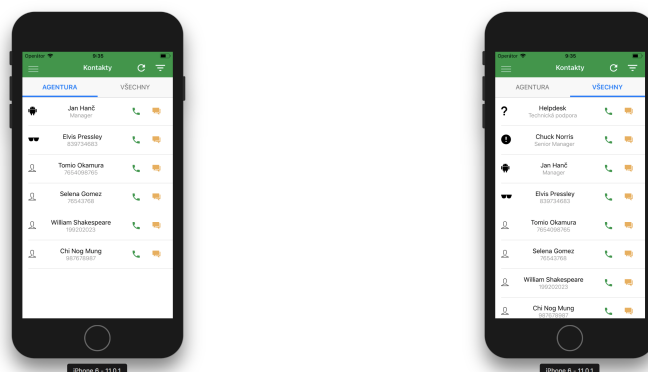
Pokud se jedná o telefonní číslo, je možné na toto číslo klepnout a nastane přímé vytáčení tohoto čísla. V případě položky adresy, je po klepnutí provedeno přímé přesměrování na mapy a tato adresa na ni zobrazena.

Po klepnutí na šipku (<) v levém horním rohu se uživatel dostane zpět na přehled zákazníků. Toto je jedna z obrazovek, na které se uživatel nedostane přímo z menu.

4.6.8 Kontakty

Seznam kontaktů je rozdělen na dvě části. V první části (zobrazena na obrázku 4.8a), se zobrazují informace z jeho vlastní agentury, tedy jeho kolegové a jeho přímý nadřízený. V druhé části, na obrázku 4.8b je zobrazen seznam, který je firemní, a zobrazují se zde kontakty na helpdesk a na senior managera, který je nadřízený jeho přímého nadřízeného v případě problémů, nebo nedostupnosti.

Kontakt má vlevo obrázek, uprostřed jméno a pod jménem pozici kterou vykonává a v pravé části dvě ikony. První ikona telefonu slouží po klepnutí k přímému volání tohoto kontaktu, druhá ikona textové zprávy k přesměrování na psaní zprávy pro tento kontakt.



(a) Kontakty agentury

(b) Všechny kontakty

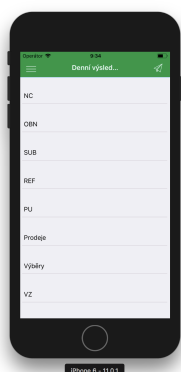
Obrázek 4.8: Obrazovky zobrazující seznamy kontaktů

Seznam kontaktů se dostane jako dotaz na backend na kontakty 4.4.2.7. V odpovědi jsou dva seznamy. První obsahující na prvním řádku přímého nadřízeného, na ostatních řádcích seznam kolegů seřazených od služebně nejstarších. Druhý obsahuje firemní seznam, kde na prvním řádku je kontakt na helpdesk a na dalších řádcích jsou další potřebné kontakty. Každý kontakt je objekt v seznamu, který obsahuje pole jméno, pozice a telefonní číslo, které má uvedeno jako kontaktní číslo v systému.

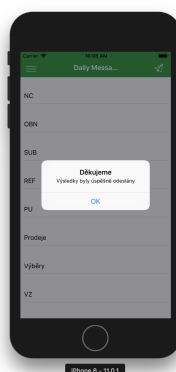
4.6.9 Reportování

Tato kapitola se zabývá pravidelným reportováním agenta svému manažerovi. Seznam polí, které manažer chce, aby mu agent reportoval nastavuje v externím systému a do aplikace pak přichází pouze seznam vybraných polí. Příklad takového seznamu, který se zobrazuje je na obrázku 4.9a.

Zobrazování těchto polí funguje na stejném principu jako skóring popsáný v části 4.6.4. Využívá se přitom stejného vzoru. Při požadavku na server 4.4.2.5, se vrátí informace ve stejné struktuře jako pro skóring, avšak obsahuje jiná pole. Tyto informace je uživatel povinen vyplnit a odeslat. K tomu slouží tlačítko na konci formuláře, ve spodní části obrazovky, nebo v pravém horním rohu ikona složené papírové vložky. Po úspěšném odeslání na server, se zobrazí děkovaná zpráva o odeslání těchto výsledků zobrazena na obrázku 4.9b.



(a) Vyplnění polí



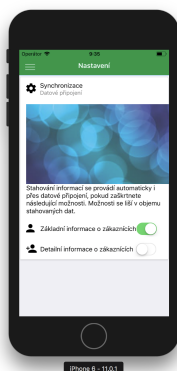
(b) Odeslaná zpráva

Obrázek 4.9: Obrazovky pro vyplnění a odeslání reportingu

4.6.10 Nastavení

Na obrazovce nastavení si uživatel (podle specifikace), může nastavit množství stahovaných dat o klientech. Jsou zde dvě zaškrťávací pole, první na automatické stahování základních informací o klientech a druhý pro automatické stahování detailních informace o klientech.

Toto nastavení se ukládá v lokální paměti a tyto informace nejsou synchronizovány se serverem.



Obrázek 4.10: Obrazovka nastavení

Kapitola 5

Testování

Pro testování s uživateli jsme si vybrali tři uživatele různých věkových kategorií, kteří jsou agenti společnosti a požádali jsme je o několik průchodů aplikací a na konci jsme je požádali o výsledné hodnocení aplikace.

Požádali jsme nejmladší z kolegyň, které je 23 let, o test aplikace. Uživatelka má kladný vztah k chytrým telefonům, ale nepoužívá je příliš často.

Ochotný pro test byl i uživatel z Prahy, kterému je 42 let. Tento uživatel má dobré technické znalosti, vzhledem ke stálému zájmu o techniku již od malička. Mobilní telefon používá velmi často.

Obchodní zástupkyně z Ústí nad Labem nám také byla k dispozici na testování aplikace z uživatelské přívětivosti. Této uživatelce je 65 let a příliš s mobilními telefony, jak říká, „není kamarádka“.

5.1 Testovací scénáře

Nejprve jsme budoucí uživatele požádali o přihlášení do aplikace a prvotní prozkoumání, na což jsme jim dali čas, a poté je požádali o provedení několika testovacích scénářů. Scénáře jsme si připravili předem a po splnění jsme se zeptali na připomínky.

5.1.1 Zobrazení detailu zákazníka z Aše

Nejmladší uživatelka neměla problém nalézt menu, otevřít kartu zákazníků a přes tlačítko vyhledávání, v pravém horním rohu se značkou lupy, vyhledat zákazníka, který v poli adresa má 'Aš'. Toto tlačítko ještě nebylo implementováno, proto musela zákazníka vyhledat manuálně posouváním zákazníků a vyhledat jej v seznamu. Poté klepnutím na jeho jméno zobrazila jeho detaily.

Druhý uživatel neměl jakékoli problémy nalézt tohoto zákazníka, ihned se orientoval, také na detail zákazníka klepl správně. K tomuto postupu nebyly žádné připomínky.

Pro první scénář měla třetí uživatelka dotazy k aplikaci, ale vše se rychle vysvětlilo a uživatelka rychle našla menu a vyhledala záložku se zákazníky. Pro vyhledávání použila tlačítko vyhledat v pravém horním rohu, které, jak bylo již řečeno, není zatím funkční, ale zákazníka našla v seznamu. Při pokusu přechodu na detail zákazníka, uživatelka zaškrtnula zaškrťovací políčko, kterým se domnívala, že zobrazí detail. Uživatelka ihned poznala, že toto není to, co potřebovala, toto políčko odškrtnula a klepla na jméno zákazníka a dostala se na jeho detail.

5.1.2 Zobrazení titulní stránky s obchodními výsledky

Zobrazení této stránky následovalo při přechodu z detailu zákazníka, jestli uživatel nalezne cestu, jak se má dostat zpět na hlavní stránku. Nejmladší uživatelka v detailu zákazníka bez problémů našla tlačítko zpět, pak také opět po stisku menu tlačítka se dostala na domovskou stránku s obchodními výsledky.

Tento přechod byl pro testovacího uživatele také bez jediného problému a přechod našel okamžitě.

Při odchodu ze stránky s detailem zákazníka, uživatelka nejprve klepla na tlačítko, které aktualizuje zákazníka. Po chvilkové orientaci, se uživatelka odkázala na správné tlačítko a přes menu tedy nebyl problém se dostat do části přehledu a poukázání na obchodní výsledky.

5.1.3 Vyplnění a odeslání denních výsledků

V testování Návrhu řešení, byli uživatelé požádáni o vyplnění skóringové zprávy. Nyní jsme je požádali, aby vyplnili výsledkovou zprávu. Postup je stejný, jako při vyplnění skóringu a odeslání.

Nejmladší uživatelka přes tlačítko menu, zobrazila reportovací okno, kde vyplnila své výsledky a na konci klepla na tlačítko odeslat výsledky. Uživatelka byla dotázána, jestli existuje další způsob odeslání výsledků, ale ikonu v pravém horním rohu nenalezla. Zároveň byla uživatelka dotázána, jestli existuje druhý způsob, jak se dostat na odeslání této zprávy z hlavní stránky, tlačítko plusu v pravém dolním rohu našla bez problémů a dostala se tak na požadovanou stránku.

První přechod automaticky uživatel klepl na ikonu plusu v pravém dolním rohu, poté na ikonu denních výsledků, vyplnil potřebné údaje a klepl na tlačítko v dolní straně obrazovky. Po dotazu, jak se může dostat na tuto stránku jiným způsobem, okamžitě přešel přes menu. Po dotazu na jinou možnost odeslání výsledků, ihned našel v pravém horním rohu ikonu odeslání.

Pro vyplnění denních výsledků, nejstarší uživatelka využila tlačítko plus v pravém dolním rohu a dostala se do části pro odeslání těchto výsledků. Po vyplnění bez problémů odeslala výsledky, k našemu překvapení, použila k tomu ikonu vlaštovky v pravém horním rohu. Sama zmínila, že by se dostala na toto okno i přes menu a výsledky odeslat tlačítkem na konci obrazovky. Tímto vyčerpala veškeré dotazy.

5.1.4 Zavolání na helpdesk

Nejmladší uživatelka opět bezchybně našla záložku kontakty. Poté se snažila kontakt Helpdesku najít v první kategorii Agentura. Po chvilkovém přemýšlení si zobrazila kartu "Všechny" a našla požadovaný kontakt. Při pokusu o zavolání, uživatelka se snažila nejprve klepnout na jméno kontaktu, poté teprve na ikonu sluchátka, které zajišťuje volání.

Nalezení kontaktů a přechod do další záložky, pro druhého uživatele, bylo velmi rychlé. Bez přemýšlení klepl na ikonu sluchátka u položky helpdesku.

Nejstarší uživatelka si počínala stejně zkušeně jako nejmladší, našla v menu kontakty a po chvilkovém hledání našla druhou záložku, kde oproti nejmladší kolegyni, klepla bez problémů na tlačítko sluchátka. Po dotazu, jestli dokáže odeslat zprávu kontaktu, klepla na ikonu zprávy, tedy navigace proběhla bez potíží.

5.1.5 Odhlášení z aplikace

Asi nejjednodušší úkol pro nejmladší uživatelku bylo odhlášení z aplikace. Bez problémů, po zobrazení menu klepla na text ve spodní části menu "Odhlásit se".

Uživatel hned našel skrze menu požadované tlačítko a odhlásil se z aplikace.

Odhlášení dalo uživatelce zabrat, protože nenašla v první chvíli tlačítko odeslání a toto tlačítko hledala na úvodní stránce. Po chvilkovém přemýšlení, uživatelka v menu našla požadované tlačítko a z aplikace se odhlásila.

5.2 Hodnocení uživatelů

Požádali jsme testovací uživatele o finální hodnocení aplikace, aby shrnuli výhody a nevýhody, jestli by řekli náměty pro vylepšení, nebo naopak řekli, co se jim na aplikaci líbilo.

Nejmladší uživatelka aplikaci hodnotí pozitivně, aplikace je pro ni přehledná, rychlá navigace mezi okny. Aplikace není složitá a je intuitivní. Pro další vývoj by doporučila vyměnit ikonu v odeslání skóringu a reportingu, místo ikony složené papírové vlaštovky, za text. Zároveň si nemyslí, že v dlouhých formulářích je toto tlačítko potřeba. Hlavní výhodou oproti

původní verzi je navigační menu, které je v levé části a schované a ze kterého se dá jednoduše dostat do dalších oken.

Uživatel by doporučil zprovoznit tlačítko pro vyhledávání zákazníka. Doporučil by vyhledávání podle adresy a podle jména. Aplikace se mu zdála intuitivní a přehledná, nejvíce se mu líbilo rozbalení menu z levé části a umístění tohoto menu. Aplikaci hodnotí velmi pozitivně.

Nejstarší uživatelka hodnotí aplikaci jako přehlednou. Vyhovuje jí, že stačí pár kliků na přechod. Pozitivně hodnotí menu, díky kterému se stala orientace jednodušší. Občas měla problém rozpoznat, co ikona v pravém horním rohu znamená, ale po vyzkoušení prý není problém. „V první chvíli člověk neví, jak to používat, ale po nějakém čase se to naučí a není problém.“ Co uživatelka rozhodně ocenila, byl seznam kontaktů, které se dají využít, protože čísla se často mění, nebo se mění kolegové, nebo vedoucí. Proto je vhodné mít toto v seznamu. Jako doporučení pro zlepšení je, aby po prvním spuštění běžel nějaký návod, který by ukázal, k čemu jaká ikona slouží a celkově by představil aplikaci, kde se co najde.

5.3 Vyhodnocení testování aplikace

Aplikace, po hodnocení uživatelů se jeví jako velmi úspěšná. Tato verze aplikace se jim lépe používá a hodnotí pozitivně přidané funkcionality. Při testování jsme narazili na několik problémů, které je potřeba do aplikace ještě zapracovat. Náměty a doporučení od uživatelů vezmeme v potaz a jsou dobrým doporučením pro další vývoj.

Kapitola 6

Závěr

V této diplomové práci jsme měli za cíl navrhnout a implementovat mobilní aplikaci pro obchodní zástupce. Tento návrh zahrnuje analýzu stávajícího stavu, kde jsme si popsali stávající aplikaci. Dále jsme provedli průzkum mezi obchodními zástupci, jak se jim aplikace používá, jaké má výhody, nevýhody a jaké funkcionality jim v aplikaci chybí. Dle těchto připomínek a požadavků obchodního oddělení jsme provedli návrh řešení aplikace.

Návrh jsme vymodelovali v programu "Just in Mind Prototyper (Free)". Tento návrh je simulací skutečné aplikace, a proto jsme jej nechali otestovat uživateli, jak jim takovéto rozvržení vyhovuje. Toto se ukázalo jako dobrá cesta.

Z připomínek od uživatelů a od obchodního oddělení, po zvolení správných nástrojů pro vývoj a programovacího jazyka, jsme naimplementovali aplikaci, která odpovídá business požadavkům. Aplikaci jsme nechali otestovat obchodními zástupci, kteří nám dali opět několik připomínek.

Aplikace funguje jak by si obchod představoval, ale rád by ještě do budoucna zprovoznil některé funkcionality, které by obchodním zástupcům ulehčily práci. Možnosti na rozšíření práce jsou:

- Propojení na mapy - V aktuální verzi se nám nepodařilo propojení na instalovanou aplikaci map realizovat z důvodu komplexnosti řešení, které by vyžadovalo detailnější pochopení rozšíření, případně vývoj nového rozšíření.
- Tlačítko vyhledat zákazníka - Díky malému množství zákazníků každého agenta, bylo odsunuta implementace tlačítka pro vyhledání zákazníka v seznamu, protože uživatel si jej může vyhledat sám bez dlouhého hledání. Uživatel rozhodně tuto ikonu ocení funkční, neboť při testování aplikace se na ni uživatelé již odkazovali.
- Automatické čtení textových zpráv - Uživatelům by také urychlilo přihlašování do aplikace, kdyby se jim text z sms zprávy, který jim přijde v ověřovací zprávě, automaticky

vyplnil do pole ve druhém kroku přihlášení. Je to pouze rozšíření, které je uživatelsky přívětivější.

- Bezpečnostní kontejner - Ve společnosti, vzhledem k citlivým informacím, se řeší zabezpečení veškerých aplikací a přístupů. Tyto přístupy jsou často řešeny svými způsoby, které vývojář uměl a naprogramoval. Jedním z rozšíření je použití bezpečnostního kontejneru, který funguje jako VPN (Virtuální privátní síť), který zajistí, že uživatel v kontejneru bude pracovat ve vnitřní síti společnosti. Tento kontejner zabezpečuje šifrování obsahu a šifrovanou komunikaci.
- Poznámky a upozornění - Někteří uživatelé by ocenili mít seznam upozornění o tom, kdy si domluvili schůzku s některým z klientů, nebo mít možnost si u něj zapsat poznámku. Tato obrazovka by mohla být dobrou součástí aplikace.

Aplikace byla implementována v React Native. Tento framework byl zvolen na základě porovnání s Ionic Frameworkem a Xamarin. Tento framework byl jednoduchý na programování a komponenty se jednoduše přidávaly a upravovaly. Při implementaci se projevilo, že tento framework ještě není zcela stabilní a že má ještě chyby, které výrazně omezovaly práci. Chyby, které tento framework vytvářel se musely složitě hledat přes jednotlivá fóra a ne vždy fungovaly, proto se muselo někdy začít od poslední funkční verze. Největší problém byl zprovoznit jej na reálném zařízení bez simulátoru.

Aplikace v React Native je funkční a k používání přívětivá. Po správném nastavení kroků pro její sestavení se rychle kompilovala a také její používání je velmi příjemné i pro uživatele, který není zdatný v používání aplikací na mobilním zařízení.

Literatura

- [1] AsyncStorage · React Native. Dostupné z: <<https://facebook.github.io/react-native/docs/asyncstorage.html>>.
- [2] Introduction to Entity Framework. Dostupné z: <[https://msdn.microsoft.com/en-us/library/aa937723\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx)>.
- [3] Geocoding API Usage Limits. Dostupné z: <<https://developers.google.com/maps/documentation/geocoding/usage-limits>>.
- [4] Hybridní, nativní nebo webové aplikace? Dostupné z: <<http://www.hybridniaplikace.cz/srovnani.html>>.
- [5] REST: architektura pro webové API, Aug 2009. Dostupné z: <<https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>>.
- [6] Two-Step Authentication. Dostupné z: <<https://uit.stanford.edu/service/authentication/twostep>>.
- [7] CRUXLAB, I. Xamarin vs Ionic vs React Native: differences under the hood, Aug 2017. Dostupné z: <<https://cruxlab.com/blog/reactnative-vs-xamarin/>>.
- [8] EISENMAN, B. *Learning React Native: building mobile applications with JavaScript*. Sebastopol, CA : O'Reilly, 2 edition, 2016. ISBN 9781491929001.
- [9] IONICFRAMEWORK. Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular. Dostupné z: <<https://ionicframework.com/>>.
- [10] S.R.O, P. F. Web společnosťi Provident Financial s.r.o. Dostupné z: <<https://www.provident.cz/>>.

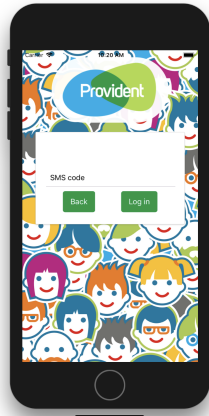
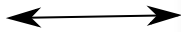
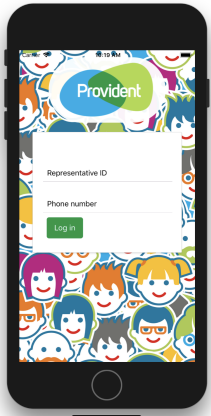
Příloha A

Obsah přiloženého CD

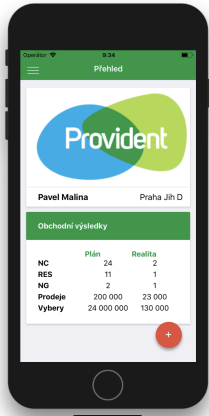
- code - složka obsahuje kód aplikace v React-Native.
- přílohy - složka obsahuje všechny přílohy v PDF.
- text - složka obsahuje text DP ve formátu PDF a v originálním formátu pro LaTeX.
- Hancjan1.vp - soubor, který lze otevřít v programu Just-In-Mind Prototyper, obsahující prototyp aplikace.
- README.txt - soubor, obsahující informace, jak spustit aplikaci

Příloha B

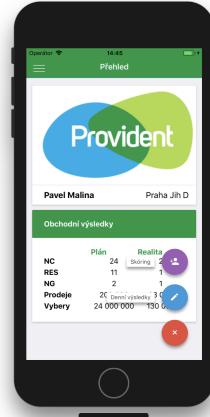
Tok obrazovek



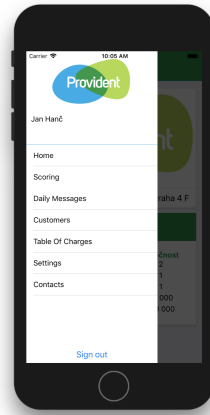
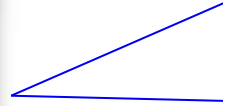
iPhone 6 - 11.0.1



iPhone 6 - 11.0.1



iPhone 6 - 11.0.1



iPhone 6 - 11.0.1

