

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Platforma pro poskytování služby praní prádla



2022

Vedoucí práce:
RNDr. Martin Trnečka, Ph.D.

Rozniuk Karyna

Studijní program: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Rozniuk Karyna
Název práce: Platforma pro poskytování služby praní prádla
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní program: Aplikovaná informatika, prezenční forma
Vedoucí práce: RNDr. Martin Trnečka, Ph.D.
Počet stran: 29
Přílohy: 1 CD
Jazyk práce: český

Bibliographic info

Author: Rozniuk Karyna
Title: Platform for providing laundry services
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study program: Applied Computer Science, full-time form
Supervisor: RNDr. Martin Trnečka, Ph.D.
Page count: 29
Supplements: 1 CD
Thesis language: Czech

Anotace

Cílem této práce je vytvořit multiplatformní aplikaci, která bude umožňovat zprostředkování služeb v oblastech jako je praní, žehlední a podobné.

Synopsis

The aim of this work is to create a multi-platform application that will enable the mediation of services in areas such as laundry, ironing and the such.

Klíčová slova: webová aplikace; mobilní aplikace; JavaScript; React Native

Keywords: web application; mobile application; JavaScript; React Native

Děkuji vedoucímu mé práce RNDr. Martinu Trnečkovi, Ph.D za vstřícnost, nápady a ochotu mi pomoci implementovat nápad. Také moc děkuji mé rodině za podporu během studia.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracovala samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	6
2	Technologie	7
2.1	React Native a Expo	7
2.2	Express.js	7
2.3	Databáze MySQL	7
2.4	JWT	8
3	Popis aplikace	9
3.1	Životní cyklus objednávky	9
3.2	Společná funkcionalita pro zákazníky a uživatele	10
3.2.1	Přihlášení	10
3.2.2	Registrace	11
3.2.3	Profil	12
3.2.3.1	Jak upravit profil	13
3.2.4	Objednávky	13
3.2.5	Notifikace	13
3.2.5.1	Příklad upozornění, které může uživatel vidět	14
3.3	Funkcionalita pro zákazníka	15
3.3.1	Domovská stránka a vytvoření objednávky	15
3.3.2	Přehled objednávek	17
3.3.3	Detail objednávky	17
3.3.4	Hodnocení	18
3.4	Funkcionalita pro vykonavatele	19
3.4.1	Domovská stránka vykonavatele	19
3.4.2	Objednávky	20
3.4.3	Detail objednávky	20
4	Schéma databáze	22
5	Budoucí vylepsení	25
5.1	Platba za služby kartou v aplikaci	25
5.2	Registrace / Přihlášení prostřednictvím jiné aplikace	25
5.3	Aktuální pozice vykonavatele	25
	Závěr	26
	Conclusions	27
A	Obsah příloženého CD	28
	Literatura	29

1 Úvod

Služby na vyžádání se staly nedílnou součástí našeho života. Služby, jako je rozvoz jídla, nákupy potravin, doprava zboží jsou nyní vzdáleny jen několik kliknutí mobilních aplikacích. Cílem této práce je vytvořit multiplatformní aplikaci, která bude umožňovat zprostředkování služeb jako je praní, žehlední a podobné. Pokud mluvíme o pracujících párech nebo studentech, nemají volný čas na praní špinavého prádla nebo návštěvu prádelny, hlavní myšlenkou aplikace je tedy usnadnit lidem řešení této problematiky. Aplikace se zabývá pouze zprostředkováním služeb mezi lidmi, existují v ní dva typy uživatelů. Prvním typ je zákazník, který poptává některou z definovaných služeb. Druhým typem je vykonavatel, který naopak může na základě poptávky v jeho oblasti, odpovědět na poptávky zákazníků a dát svou cenovou nabídku.

2 Technologie

V následující kapitole popíšeme technologie použité při vývoji aplikace, která je výstupem práce.

2.1 React Native a Expo

React native je postaven na platformě react, ale místo toho, aby se orientoval na prohlížeč, je zaměřený multiplatformě s důrazem na mobilní platformy [1].

React native je framework pro vytváření nativních aplikací pro iOS, Android a webové prohlížeče pomocí JavaScriptu. Tyto aplikace jsou skutečně nativní nejedná se o webové aplikace, které vypadají jako mobilní aplikace. Díky tomu v případě vývoje v react native není potřeba mít žádné znalosti programování pro iOS nebo Android, pokud samozřejmě nebude potřeba pracovat přímo s nativním rozhraním API dané platformy. Příkladem nativních API rozhraní můžou být například služby jako Google Pay nebo Apple Pay. Existují dva způsoby, jak vytvářet aplikace React native - běžný react native nebo Expo [2], což je sada nástrojů a framework, který je umístěn nad React native a skrývá velkou část složitosti, takže je to neuvěřitelně rychlé a snadné.

Při vytváření projektu pomocí nativního rozhraní React CLI, bude mít projekt složky Android a iOS. Jedná se o nativní projekty pro Android a iOS, jakož i o složku JavaScript, jejíž kód lze sdílet mezi oběma platformami. Při vytváření projektu pomocí aplikace Expo nebudou existovat žádné podprojekty pro Android a iOS, ale pouze kód JavaScriptu, což znamená, že nelze pracovat přímo s nativním rozhraním API těchto platform, budeme omezeni na to, co nám Expo nabízí z hlediska nativních funkcí.

2.2 Express.js

K vytvoření serveru je použitý webový Node.js framework Express.js [3]. Express.js je minimální a flexibilní framework pro open source webové aplikace. Je vytvořen pro tvorbu serverových aplikací a nabízí skvělou sadu funkcí pro tvorbu webových aplikací, můžeme v něm snadno vytvořit webovou aplikaci, která dokáže zpracovat více HTTP požadavků zároveň. I když je samotný Express poměrně minimalistický, bylo pro něj vytvořeno mnoho kompatibilních balíčků middleware, které řeší téměř jakýkoli problém při vývoji webu. Existují knihovny pro práci se soubory cookie, databázemi, přihlašování uživatelů, parametry URL, bezpečnostními hlavičkami a mnoha dalšími. K implementaci serverové logiky se používá JavaScript a díky tomu je v naší aplikaci také možné snadné sdílení kódu mezi serverovou a klientskou částí aplikace, například různé validační logiky.

2.3 Databáze MySQL

Databáze je souhrn strukturovaných dat, s nimiž lze pomocí nějakého databázového systému manipulovat (vyhledávat, porovnávat, editovat). Existují různé

systémy pro správu dat, které se dělí na dvě kategorie: relační a nerelační. V relačních databázích jsou data uložena v tabulkách vzájemně propojených vztahy. SQL je standardizovaný jazyk pro dotazování a aktualizaci dat. MySQL [4] je relační databázový systém založený na jazyce SQL a v našem případě je použit jako hlavní úložiště aplikačních dat. Obsahuje také uložené procedury, které zjednodušují serverovou logiku. Na serveru se proto nemusí zpracovávat tabulky přímo, ale mohou se použít uložené procedury.

2.4 JWT

JWT JSON Web token je považován za jeden z bezpečných způsobů ověření identity uživatelů v síti internet [5].

JWT je řetězec ve formátu `header.payload.signature`. Header udává jaký kryptografický algoritmus byl použit pro generování podpisu. Payload obsahuje užitečné informace o držiteli jako je ID uživatele, výčet jeho práv, platnost. Signature je elektronický podpis vygenerovaný pomocí daného kryptografického algoritmu a tajného klíče, není ho tedy možné jednoduše podvrhnout. Základní myšlenkou je, že pokaždé, když se uživatel přihlásí, chceme vytvořit jiný token spojený s ID uživatele. Token má také expirační dobu po kterou je validní. Tento token budeme používat přidávat ke každému požadavku na API rozhraní serveru. Pokud chce například uživatel změnit své uživatelské jméno nebo heslo, musíme při požadavku ověřit, zda na server uživatel zaslal platný JWT token.

3 Popis aplikace

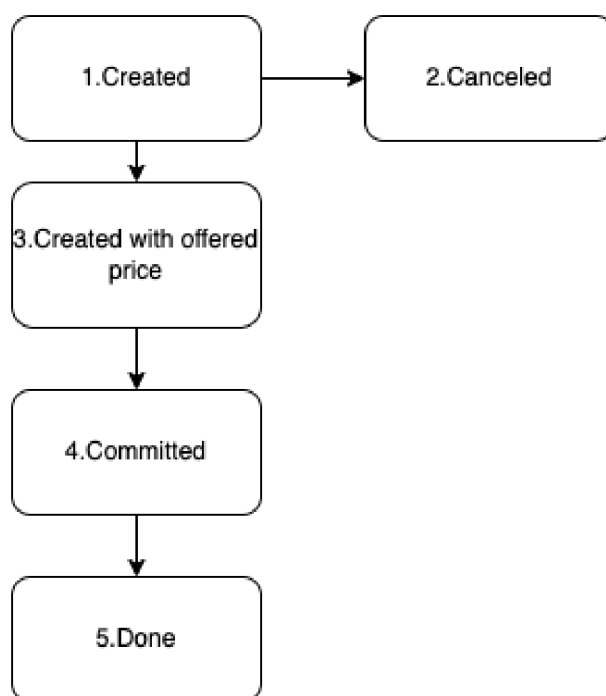
V této kapitole se budeme zabývat popisem vytvořené aplikace. Základem aplikace jsou uživatelé, zákazníci (customers) a vykonavatelé (executors). Ti mezi sebou interagují pomocí objednávek. Aplikace zprostředkovává pouze základní funkce pro řešení dané problematiky. Funkce jako je platba nebo vzájemné posílání zpráv a komunikaci si uživatelé musí řešit za pomoci jiného komunikačního kanálu, například telefonu který je u každé objednávky obou stranám kontraktu k dispozici.

3.1 Životní cyklus objednávky

Diagram na obrázku 1 znázorňuje životní cyklus objednávky. Stav objednávky se mění v závislosti na tom, zda s ní interaguje zákazník nebo vykonavatel. Zákazník může vytvořit objednávku, zrušit ji nebo přijmout nabídku od vykonavatele služeb. Vykonavatel může nabídnout u objednávek svou zamýšlenou cenu za požadované služby a následně potvrdit realizaci zakázky.

Pět kroků životního cyklu objednávky je:

1. `Created` - stav který objednávka nabývá hned po vytvoření.
2. `Canceled` - může se stát, že si zákazník zadání objednávky rozmyslí nebo službu již nepotřebuje. Dokud je objednávka v stavu `Created`, může ji klient zrušit kliknutím na tlačítko „Cancel“. Objednávka nebude pro vykonavatele viditelná a její stav se změní.
3. `Created with offered price` - po vytvoření bude objednávka viditelná pro vykonavatele v dané lokalitě. Pokud má vykonavatel o zakázku zájem, může nabídnout cenu za své služby. Objednávka zůstává ve stavu `Created`.
4. `Committed` - pokud má vykonavatel o zakázku zájem, bude o tom zákazník informován. Na stránce `Order Details` si zákazník může prohlédnout všechny nabídky a vybrat si tu, která mu nejvíce vyhovuje. Kliknutím na potvrzovací tlačítko se stav objednávky změní na `Committed`, vykonavatel bude informován a může přistoupit k řešení zakázky.
5. `Done` - po doručení oblečení zákazníkovi, musí dodavatel potvrdit dokončení obchodu kliknutím na tlačítko „Done“, které nachází v detailu objednávky. Poté přijde zákazníkovi upozornění a může objednávku ohodnotit.



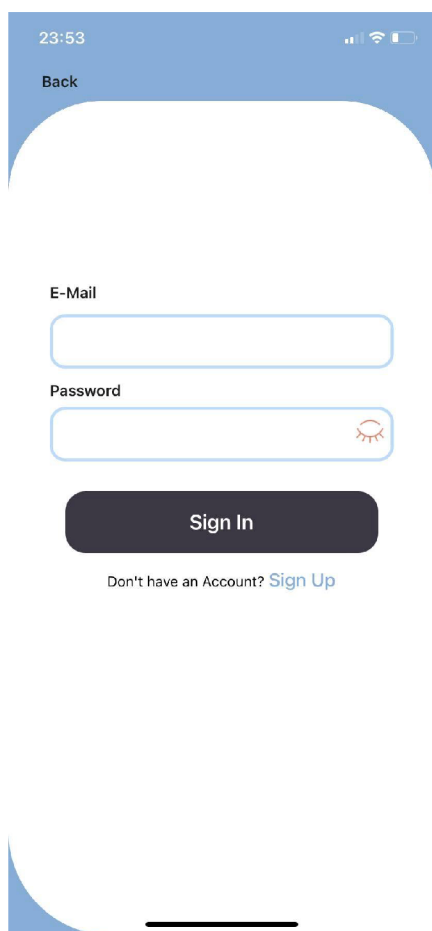
Obrázek 1: Životní cyklus objednávky

Aplikace se skládá ze dvou částí: části pro zadavatele a části pro zákazníka. Po registraci může uživatel podle toho, zda si zvolil roli zákazníka nebo dodavatele, vstoupit do svého účtu, který obsahuje potřebné funkce. Navigace v aplikaci je uskutečněna pomocí Tab Navigátor, který reprezentují záložky na spodku obrazovky.

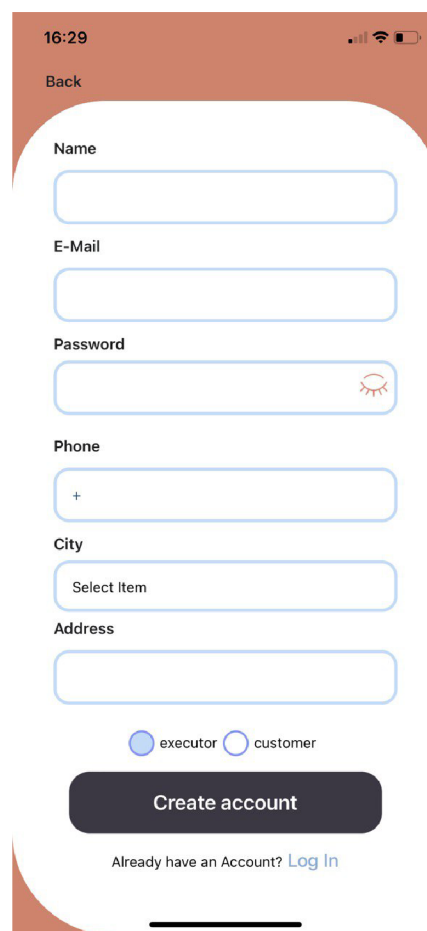
3.2 Společná funkcionalita pro zákazníky a uživatele

3.2.1 Přihlášení

První, s čím se uživatel setká, je výzva k přihlášení do aplikace (ukázka na obrázku 2). Pro přihlášení je třeba zadat e-mailovou adresu a heslo. Tyto údaje musí být stejné jako údaje, které uživatel použil při registraci. Pokud je e-mail nebo heslo zadáno nesprávně, aplikace informuje uživatele o chybě „Wrong username/password combination“. Pokud uživatel nechá pole prázdné, aplikace vás o tom bude informovat a požádá uživatele o zadání údajů. Pokud uživatel nemá v aplikaci účet, může použít tlačítko „Sign Up“, které ho přesměruje na registrační stránku.



Obrázek 2: Přihlášení



Obrázek 3: Registrace

3.2.2 Registrace

Registrace uživatele je vytvoření účtu, který bude uložen na serveru a jehož prostřednictvím se uživatel může přihlásit do aplikace a vytvářet nebo plnit objednávky. Po příchodu na registrační stránku bude uživatel vyzván k vyplnění registračního formuláře (ukázka na obrázku 3).

Při registraci jsou všechna pole povinná, pokud jsou prázdná nebo nesplňují validační požadavky, bude o tom uživatel informován.

Pole Name slouží uživateli jako informativní pole, aby mohl identifikovat, s kým bude spolupracovat na objednávce.

E-mail a password - tyto údaje budou použity k přihlášení do aplikace. Uživatelský účet je spojen s e-mailem, takže není možné e-mail změnit. Heslo musí obsahovat nejméně osm znaků, z toho alespoň jeden velký znak, jeden malý znak a jednu číslici.

Phone začíná znaménkem plus a musí obsahovat předvolbu země. Uživatel si může vybrat město (pole city), ve kterém chce zadávat nebo plnit objednávky. Po registraci se uživateli zobrazí pouze objednávky, které odpovídají zvolenému místu.

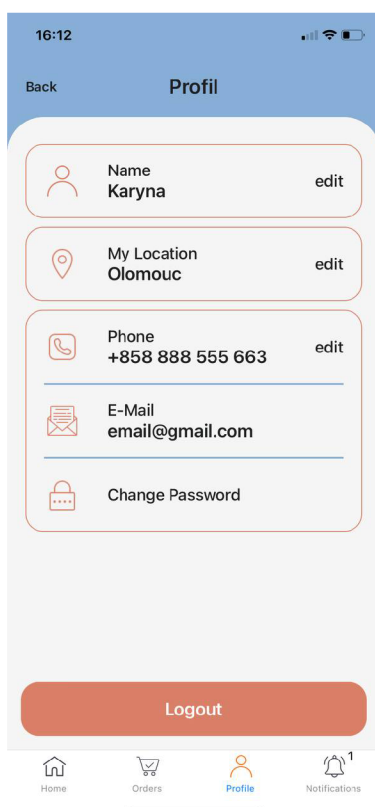
Pole `address` je nezbytné pro obě strany a především pro vykonavatele, aby věděl, zda je pro něj daná vzdálenost od zákazníka přijatelná.

Posledním bodem před ukončením registrace je výběr role, kterou chce uživatel vykonávat. Na výběr je role `customer` (pole `customer`), který je schopen vytvořit zakázku, a `executor` (pole `executor`), který je schopen vybrat ze seznamu zakázek a v budoucnu dokončit zakázku v nastaveném časovém rámci, který byl předem definován zákazníkem.

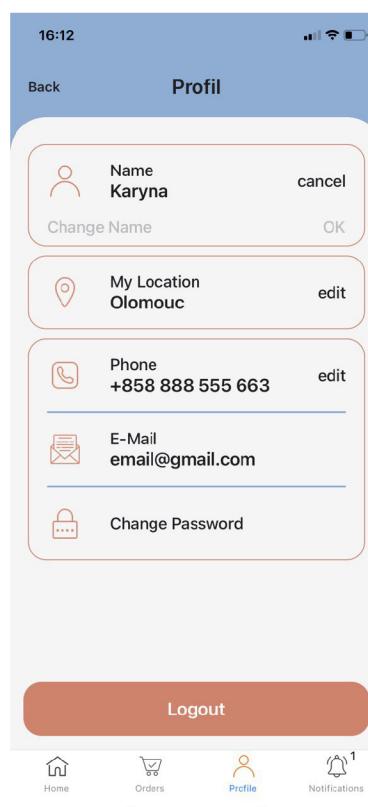
Jakmile uživatel stiskne tlačítko „Create account“, všechny zadané údaje se ověří, a pokud jsou některé z nich neplatné nebo neodpovídají podmínkám aplikace, bude o tom uživatel informován. Pokud je registrace úspěšná, bude uživatel přesměrován na přihlašovací stránku. Uživatel může kdykoli změnit své údaje s výjimkou e-mailu.

3.2.3 Profil

Uživatelský profil je charakteristika jednotlivého uživatele aplikace, která je dána souborem atributů. Chcete-li otevřít profil aktuálního uživatele, klikněte na tlačítko „Profile“ v pravém dolním rohu stránky. Otevře se stránka se zobrazením profilu uživatele (ukázka na obrázku 4). Profilová stránka obsahuje: osobní údaje, lokalitu, kontaktní informace (telefonní číslo, e-mail) a změnu hesla.



Obrázek 4: Profil



Obrázek 5: Editace jména na profilu.

3.2.3.1 Jak upravit profil

Chcete-li upravit atribut, klikněte na tlačítko „Edit“ a zobrazí se pole, ve kterém můžete provést změny. Po zadání textu nebo výběru položky stačí kliknout na tlačítko „OK“ (ukázka na obrázku 5). Poté se provede ověření, zda nová hodnota splňuje podmínky aplikace. Pokud si uživatel změny rozmyslí, stačí stisknout tlačítko „Cancel“.

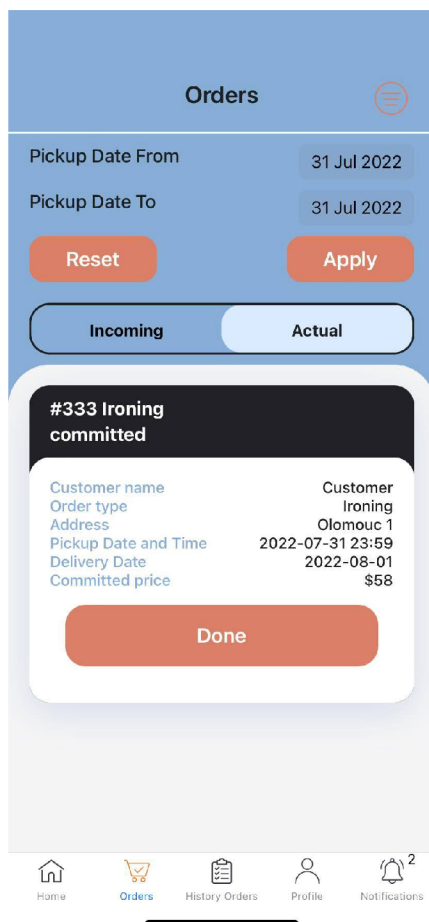
3.2.4 Objednávky

Objednávky mají několik stavů: `created`, `committed`, `done`, a `canceled`. V hlavičce objednávky vždy vidíme její číslo, typ a také aktuální stav. Komponenta ve skrolovacím seznamu obsahuje další důležité informace o objednávce: adresu na kterou zákazník očekává doručení oblečení, službu kterou vybral, stav objednávky, stanovený termín kdy zákazník získá oblečení zpět. Na stránkách které zobrazují objednávky uživatelů je přidán filter pro filtrování objednávek podle data vyzvednutí. Příklad vidíme na obrázku číslo 6.

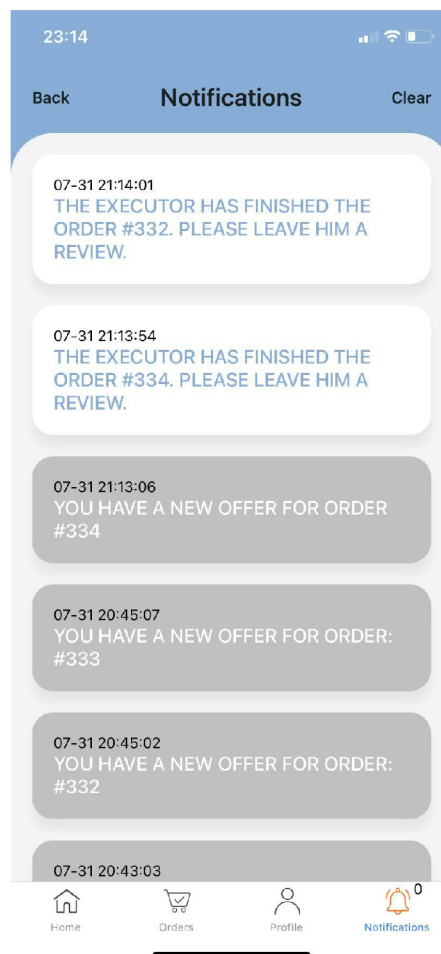
3.2.5 Notifikace

Notifikace nám umožňují informovat uživatele o změnách stavů objednávek. Chcete-li otevřít seznam notifikací, ve spodní liště klikneme na tlačítko „Notification“. Příklad notifikace vidíme na obrázku číslo 7.

Během přítomnosti uživatele v aplikaci se každých pět sekund aplikace dotazuje na server o nové notifikace, v případě že uživatel má nějaké nové notifikace, ve spodní liště uvidíme číslo reprezentující počet nových notifikací. Výstrahy, které byly již dříve zobrazeny, jsou vyzobrazeny šedě. Výstrahy, které jsou nové a uživatel je ještě nečetl, jsou zobrazeny bíle. Po otevření stránky s oznámeními se všechna oznámení automaticky přečtou. Kontejner s popiskem obsahuje zprávu a datum, kdy byla notifikace vytvořena. Po kliku na notifikaci, je uživatel přeměrován na detail objednávky. Pokud uživatel nemá žádná upozornění, zobrazí se kontejner s popiskem - „You don't have any notification yet“.



Obrázek 6: Filter objednávek



Obrázek 7: Notifikace

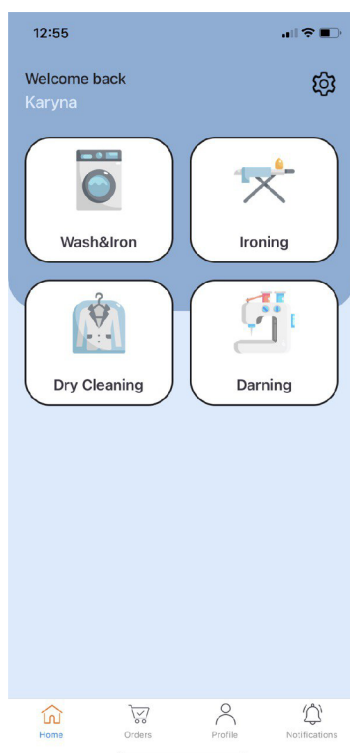
3.2.5.1 Příklad upozornění, které může uživatel vidět

- „Customer has accepted your offer for order #...“ - v případě, že byl zákazník spokojen s cenou, kterou dodavatel nabídl.
- „You have a new offer for order #...“ - pokud vykonavatel služeb nabídl cenu za objednávku, kterou zákazník dříve zadal.
- „The executor has finished the order #... Please leave him a review.“ - pokud objednávka je ve stavu Done, zákazník může ohodnotit služby vykonavatele.
- „The order: #... was rated“ - pokud zákazník ohodnotí již vykonanou objednávku.

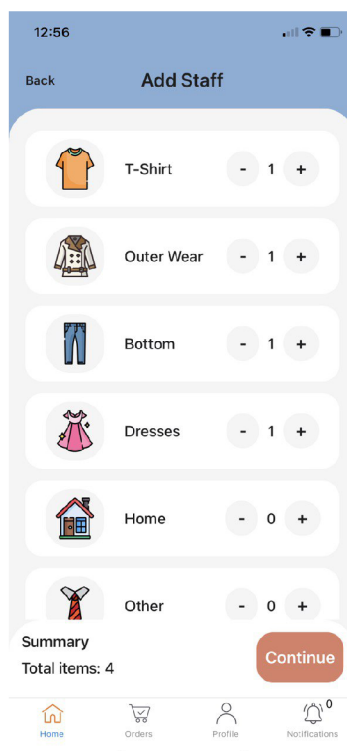
3.3 Funkcionalita pro zákazníka

3.3.1 Domovská stránka a vytvoření objednávky

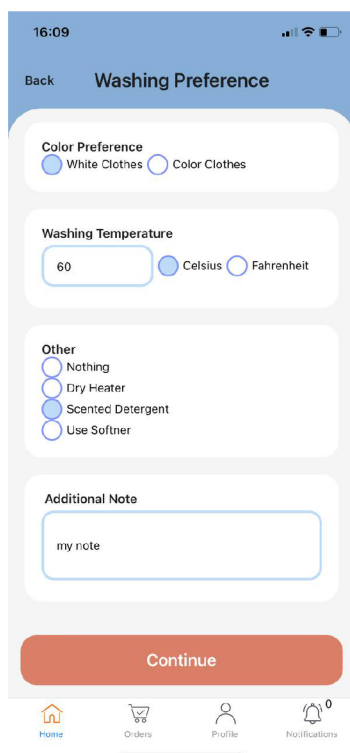
Po vstupu do aplikace se zákazníkovi zobrazí stránka, kde si může vybrat typ objednávky, např. praní, žehlení, šití nebo chemické čištění (obrázek 8). Po výběru jedné ze služeb bude uživatel přesměrován na stránku přidání oblečení `Add Staff` se seznamem položek které reprezentují druhy oblečení, například šaty, kalhoty, věci z domácnosti a tak dále (ukázka na obrázku 9). Dále, pokud zákazník vybral službu praní a žehlení, bude přesměrován na stránku `Washing Preferences` (obrázek 10). Na stránce s položkami si zákazník vybere barvu prádla podle toho, zda chce prát barevné nebo bílé věci, a preferovanou teplotu. Výchozí nastavení je bílá barva a teplota se udává ve stupních Celsia. Zákazník taky může zvolit i další možnosti, např. použít změkčovač tkanin nebo aviváž. Pole „Note“ je nepovinné - zákazník může dodavateli zanechat další informace. Po vyplnění všech polí může klient kliknutím na tlačítko „Continue“, tím bude přesměrován na další stránku - `PickUpDetails`, která je společná pro všechny typy objednávek (příklad na obrázku 11). Zákazník musí uvést adresu, na které si chce oblečení předat, datum a čas předání a preferovaný datum vracení oblečení. Při přechodu ze stránky na stránku aplikace kontroluje zadané údaje a informuje zákazníka, pokud jsou údaje zadány nesprávně nebo v nesprávném formátu. Objednávka bude vytvořena až po vyplnění všech povinných polí ve správném formátu. Po stisknutí tlačítka „Done“ se vytvoří objednávka a zobrazí se pro všechny vykonavatele ve vybrané lokalitě.



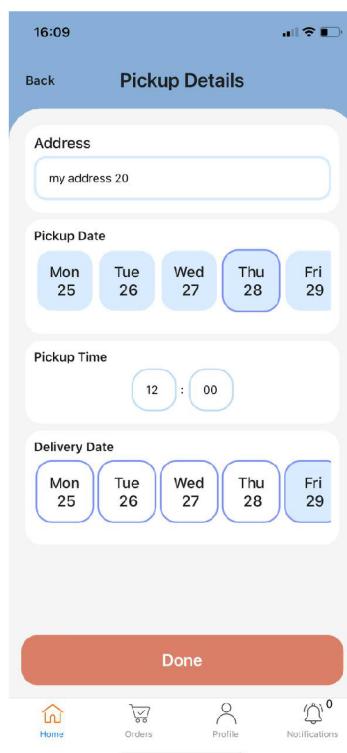
Obrázek 8: Domovská stránka zákazníka



Obrázek 9: Výběr oblečení



Obrázek 10: Práci preference

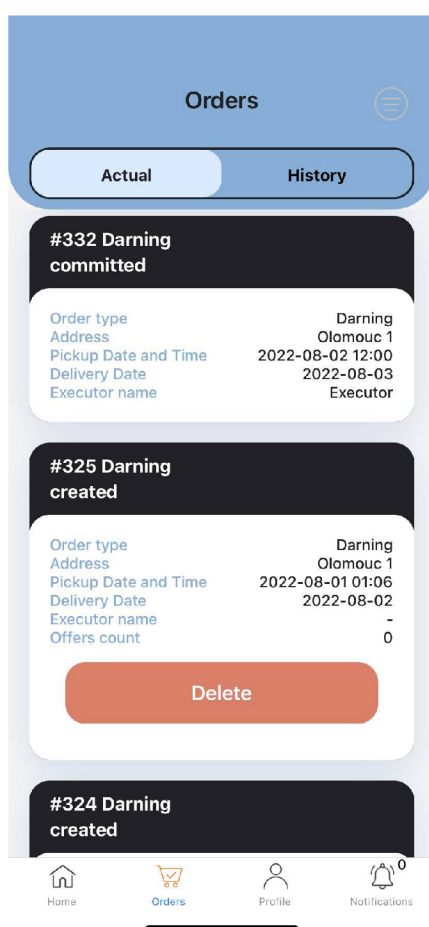


Obrázek 11: Detaily doručení

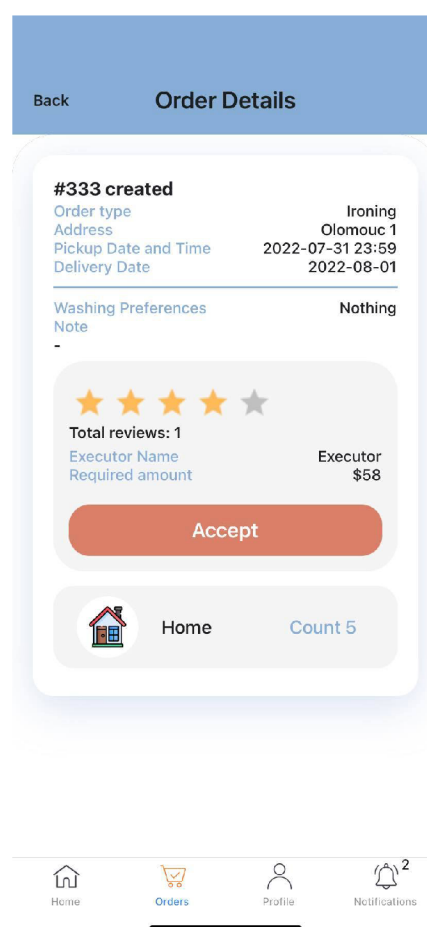
3.3.2 Přehled objednávek

Pokud zákazník chce mít přehled o objednávkách, stačí kliknout na tlačítko „Orders“ v dolní liště.

V záložce Actual jsou všechny objednávky, které byly dříve vytvořeny zákazníkem (příklad na obrázku 12). Objednávky mohou mít stav Created - dokud nebude přidělena vykonavateli nebo stav Committed. Dokud se stav objednávky nezmění na Committed, může zákazník objednávku zrušit kliknutím na tlačítko „Delete“. Poté se stav objednávky změní na Canceled a nebude se již poskytovatelům zobrazovat. V záložce History jsou objednávky které mají stav Done a Canceled. Stav Done objednávka nabývá v moment, kdy vykonavatel předá zpátky zákazníkovi věci a v aplikaci potvrdí doručení.



Obrázek 12: Aktuální objednávky - zákazník



Obrázek 13: Detail objednávky s nabídnou cenou

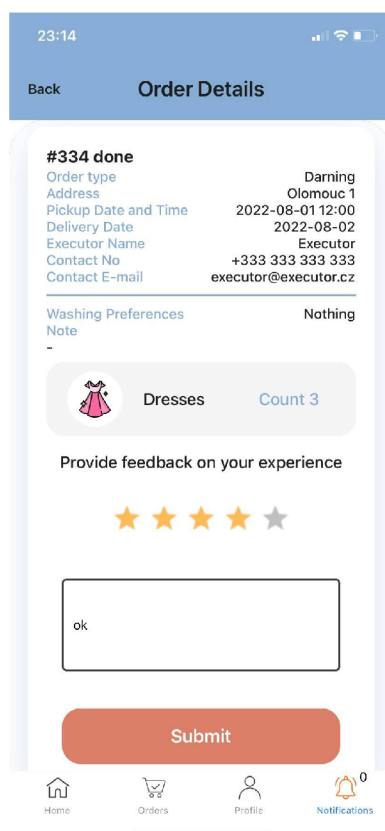
3.3.3 Detail objednávky

Pokud chce zákazník zobrazit podrobnější informace o objednávce, stačí kliknout na komponentu. Klient bude přesměrován na stránku s podrobnostmi o objed-

návce, kde budou uvedeny všechny informace, které klient uvedl při vytváření objednávky, například typ služby, název a množství položek atd. Pokud má objednávka na seznamu objednávek u hodnoty „Offers count“ hodnotu vyšší než 0, znamená to, že o zakázku má zájem nějaký vykonavatel a nabídl částku, za kterou je ochoten ji vykonat (obrázek 13). V podrobnostech objednávky se klientovi zobrazí seznam všech účinkujících, jejich hodnocení, stručné informace o nich a nabízená cena. V případě že objednávka je ve stavu Done, zákazník může ohodnotit služby vykonavatele a nechat recenzi.

3.3.4 Hodnocení

Pokud vykonavatel potvrdil vykonání objednávky, objednávka je ve stavu Done. Zákazník dostane notifikaci ze jeho objednávka je hotová a že ji může ohodnotit (obrázek 14). Po kliknutí na notifikaci, bude uživatel přesměrován na detail objednávky, kde se mu zobrazí hodnotící formulář. Zákazník pomocí hvězdičkového hodnocení může ohodnotit služby vykonavatele, případně zanechat zprávu. Recenzi od zákazníka si může vykonavatel přečíst na detailu již vykonané objednávky. Zákazník může ohodnotit objednávku pouze jednou. Hodnocení vykonavatele se sčítá, a počítá se průměr, který se později zobrazuje na domovské stránce vykonavatele a také při zobrazení nabídek v detailu objednávky u zákazníka.

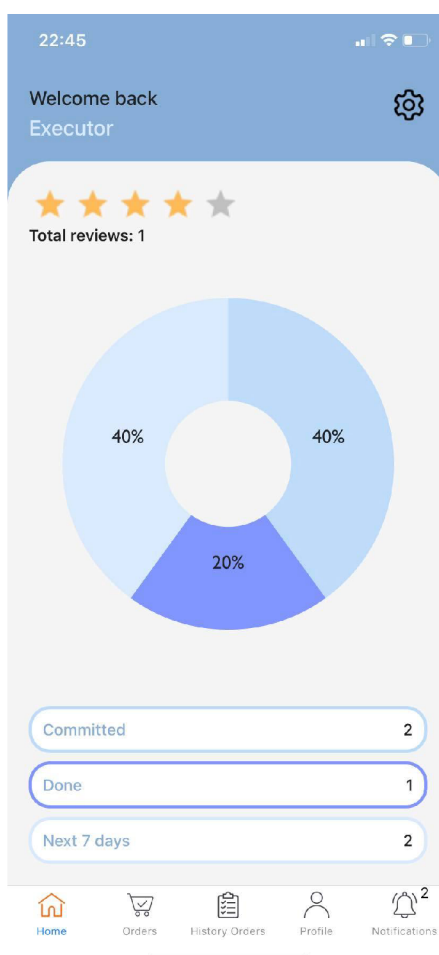


Obrázek 14: Hodnocení služeb vykonavatele

3.4 Funkcionalita pro vykonavatele

3.4.1 Domovská stránka vykonavatele

Domovská stránka obsahuje graf, který zobrazuje historickou a budoucí aktivitu vykonavatele (obrázek 15), aktuální hodnocení vykonavatele a množství hlasů. Graf je rozdělen na čtyři části odpovídající statusu objednávek: dokončené, objednávky přijaté zákazníkem, objednávky, které mají být provedeny do konce dne a objednávky na následující týden. Zpočátku, když je uživatelský účet prázdný a vykonavatel ještě nezačal zpracovávat objednávky, je graf zcela šedý a nemá žádnou vypovídací hodnotu. Jakmile dojde k nějaké aktivitě týkající se objednávky, graf se automaticky změní. Vnější část grafu zobrazuje procentuální poměr vybrané skupiny k ostatním skupinám. Po kliknutí na vnější stranu grafu, se ve vnitřní části grafu zobrazí počet objednávek a název dané kategorie.



Obrázek 15: Domovská stránka vykonavatele

3.4.2 Objednávky

Chcete-li uživatel zobrazit aktuální objednávky, musí kliknout na záložku `Orders` v dolním navigačním menu. V záložce `Incoming` jsou všechny objednávky, které odpovídají aktuální lokalitě a mají stav `Created`. Pokud vykonavatel nemá zájem o objednávku, může ji explicitně zamítnout kliknutím na tlačítko „Reject“ (obrázek 16). Objednávka bude zamítnuta a nebude se již vykonavateli zobrazovat.

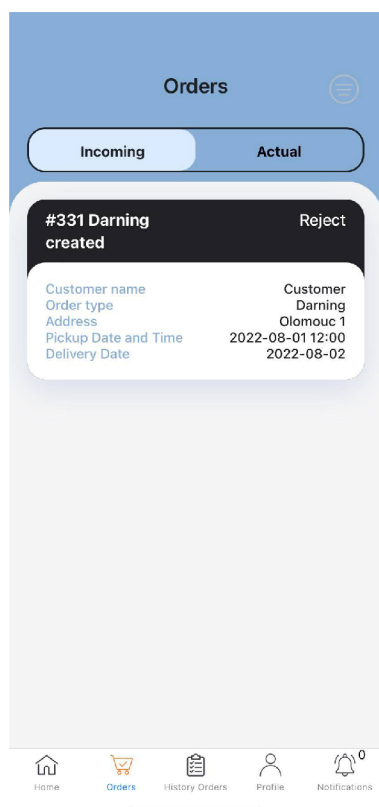
Záložka `Actual` obsahuje objednávky, které byly přiděleny vykonavateli. Aby si vykonavatel mohl prohlédnout objednávky, které již realizoval, stačí kliknout na tlačítko „History Orders“ v dolní navigační nabídce.

Objednávky, které stále mají stav `Created`, ale datum převzetí již uplynul, budou automaticky smazány. K mazání objednávek dochází každých pět minut a zajišťuje ho serverová část aplikace.

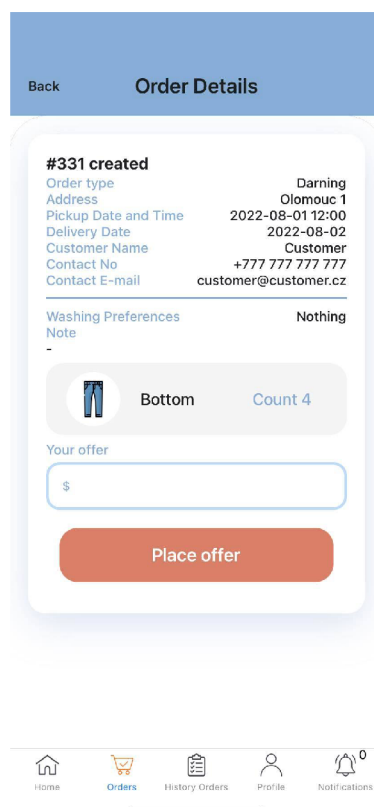
3.4.3 Detail objednávky

Po kliknutí na objednávku se uživateli zobrazí podrobnosti o objednávce. Zobrazí se seznam oblečení a množství, které zákazník předá, podrobnosti o objednávce (teplota, barva, případné další požadavky). Vykonavatel může nabídnout cenu pro zákazníka, za kterou je ochoten poskytnout svojí služby (obrázek 17). Pokud zákazník přijme vykonavatelem navrhnoutou cenu, změní se stav objednávky na `Committed`, což znamená, že se poskytovatel služeb zavazuje provést objednávku dle data určeného zákazníkem.

Objednávku s novým stavem vykonavatel najde v záložce `Actual`. Ke všem základním informacím se přidá cena, kterou vykonavatel za objednávku nabídl. Poskytovatel služeb může na této kartě také potvrdit, že objednávka byla provedena a doručena zákazníkovi. Obsahuje informaci o celkové objednávce: jméno zákazníka, adresu na kterou zákazník očekává doručení oblečení, číslo objednávky, službu kterou vybral, stav objednávky, stanovený termín kdy zákazník získá oblečení zpět.



Obrázek 16: Aktuální objednávky - vykonavatel



Obrázek 17: Detail objednávky pro vykonavatele

4 Schéma databáze

Databáze obsahuje údaje o objednávkách, včetně všekých údajů zákazníků, dataily objednavek apod. Databázové schéma je znázorněné na obrázku 19 a skládá se z osmi tabulek:

- `usersData`
- `orderDetails`
- `orders`
- `deliveryDetails`
- `orderStaff`
- `orderOffers`
- `userNotifications`
- `locations`

V tabulce `usersData` je pro každého klienta jednoznačně definováno primární klíč - `id`. Dále tabulka obsahuje jméno - `name`, `password`, `phone`, `address`, `role`, `email`, `locationId` (cizí klíč pro propojení s tabulkou `locations`), `executorRating`, `executorRatingsCount`. Heslo uživatele je uloženo v šifrované podobě. Před uložením hesla je k heslu přidán náhodny řetězec, takzvaná „sůl“, poté je heslo zašifrováno a uloženo spolu s nezašifrovanou solí. Při ověřování totožnosti uživatele je k příchozímu heslu přidána daná sůl, heslo je poté zašifrováno a výstup je porovnán s šifrovaným heslem uloženým v databázi.

Tabulka `orderDetails` obsahuje informaci o detailu objednávky. Obsahuje atributy `orderId` (cizí klíč pro propojení s tabulkou `orders`), `deliveryId` (cizí klíč pro propojení s tabulkou `deliveryDetails`), `temprature`, `color`, `price`, `note`, `tempratureType`, `otherWashingPreferences`.

Tabulka `orders` obsahuje údaje o provedení objednávky. Každá objednávka je jednoznačně definované primárním klíčem - `id`, je uveden kód zákazníka, který objednávku vytvořil - `customerId` (cizí klíč pro propojení s tabulkou `usersData`), kód vykonavatele, který objednávku vykonával - `executorId` (cizí klíč pro propojení s tabulkou `usersData`), `status`, `orderType`, a cizí klíč `locationId`, pro komunikaci s tabulkou `locations`.

Tabulka `deliveryDetails` obsahuje informaci o doručení objednávky. Obsahuje atributy: `dateTime`, `address`, `pickupDate`, `pickupTime`, `deliveryDate`.

Tabulka `orderStaff` obsahuje údaje o věcích které obsahuje objednávka, `orderId` (cizí klíč pro propojení s tabulkou `orders`), `staffType` který udává typ oblečení a `staffCount` udáva jeho počet.

Tabulka `orderOffers` obsahuje údaje o nabídky všech vykonavatelů ke konkrétním zakázkám. Obsahuje atributy: `orderId` (cizí klíč pro propojení s tabulkou `orders`), `executorId` (cizí klíč pro propojení s tabulkou `usersData`), `price`.

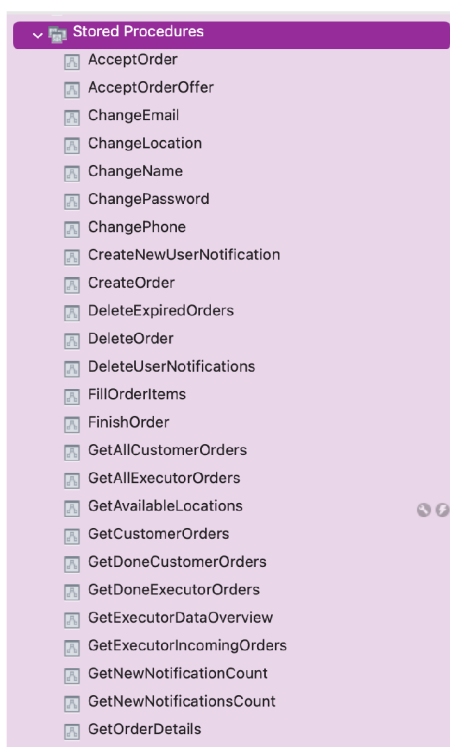
Tabulka `userNotifications` obsahuje údaje o notifikacích pro konkrétního uživatele. Obsahuje atributy `notificationText`, `userId` (externí klíč pro komunikaci s tabulkou `usersData`), `isRead` (hodnota ke zjištění jestli notifikace již byla přečtena), `dateTime` (datum a čas vytvoření notifikace).

Tabulka `locations` obsahuje seznam možných měst (pole `locationName`). Každé město má svůj primární klíč (`id`). Tabulka není přístupná pro změny uživatelů, ale pouze pro administrátora aplikace. Pokud by administrátor chtěl přidat další lokaci, například Přerov, musí na MySQL serveru spustit následující dotaz:

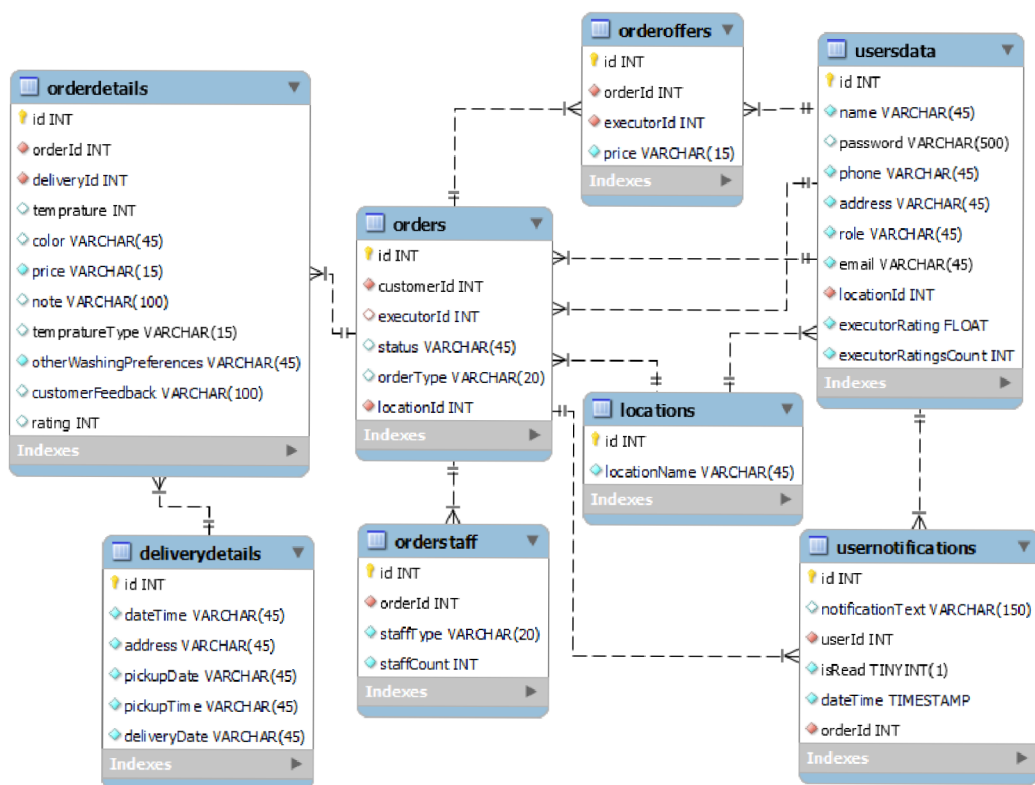
```
1 INSERT INTO locations (locationName) VALUES ("Přerov");
```

Zdrojový kód 1: SQL

Většina databázové logiky je implementována přímo pomocí databázových procedur (obrázek 18). Uložené procedury umožňují kombinovat sekvence dotazů a ukládat je na serveru.



Obrázek 18: Databázové procedury



Obrázek 19: Schéma databáze

5 Budoucí vylepšení

Nejdříve je třeba zjistit důvody, proč je třeba aplikaci modernizovat, zjednodušeně řečeno upravit ji podle potřeb uživatele. Každým rokem přibývá lidí, kteří používají mobilní telefony nebo se stávají uživateli Internetu. Většina z nich využívá nejen fotoaparáty nebo některé z klasických aplikací telefonu či počítače, ale také aplikace, které uživateli umožňují plnit jeho potřeby, jako je online bankovníctví, streamovací služby apod. Aplikace je třeba neustále zdokonalovat a vyvíjet. Nejde jen o přidávání nových funkcí, ale také o vylepšování těch, které již fungují. Použitelnost je schopnost produktu být pro uživatele v dané situaci srozumitelná a atraktivní. Pro zákazníka je důležité, aby vše bylo vyřešeno v mobilním telefonu. Existuje několik vylepšení, která by rozhodně usnadnili práci s aplikací a její použitelnost.

5.1 Platba za služby kartou v aplikaci

V současné době aplikace funguje tak, že služby vykonavatele budou zaplacené při dodání zboží od vykonavatele k zákazníkovi. K vyřešení tohoto problému je nutná implementace platební brány. Mezi oblíbené patří PayPal, RazorPay, Stripe případně také nativní možnosti mobilních telefonů jako je Google Pay nebo Apple Pay.

5.2 Registrace / Přihlášení prostřednictvím jiné aplikace

Každá aplikace má možnost přihlášení a registrace, ale pokud by byla možnost přihlášení prostřednictvím sociálních sítí, bylo by pro uživatele mnohem snazší a atraktivnější se zaregistrovat. Takovým zprostředkovatelem registrace může být např. Facebook nebo Google.

5.3 Aktuální pozice vykonavatele

Řekněme, že je den, kdy má dodavatel předat věci, ale zákazník je příliš zaneprázdněný a nemá čas ani chuť někomu volat a zjišťovat, kdy vykonavatel dorazí. Každý uživatel by ocenil možnost sledovat polohu vykonavatele, to by vyžadovalo přidání dalšího stavu objednávky mezi Committed a Done - například na cestě. Zákazník by viděl mapu, která by ukazovala polohu vykonavatele. Přesné informace o poloze vykonavatele by také umožnily způsob, jak zvýšit jeho efektivitu. Na základě těchto údajů by aplikace mohla zobrazovat objednávky, které se nacházejí blízko něj.

Závěr

V rámci bakalářské práce byla implementována multiplatformní aplikace Laundry App zabývající se zprostředkováním služeb v oblasti praní, žehlení a podobných. Aplikace podporuje dva typy uživatelů zákazník a vykonavatele, kteří pomocí ní mohou vzájemně interagovat. Hlavním prvkem aplikace jsou objednávky, které může zákazník zadávat. Vykonavatelé mohou u volných objednávek navrhnout svou odměnu za odvedené služby, pokud daný zákazník jejich nabídku přijme tak musí vykonavatel objednávku vyplnit.

Uživatelské rozhraní je implementováno pomocí technologie React native, serverová část pomocí Express.js a aplikační data se ukládají do databázového systému MySQL. Aplikace je do budoucna snadno rozšiřitelná. Aplikace lze pustit jak v internetovém prohlížeči, tak na mobilních telefonech.

Conclusions

Within the framework of the bachelor thesis, a multiplatform application Laundry App was implemented, which deals with the mediation of services in the field of laundry, ironing and similar. The app supports two types of users customers and executors who can interact with each other using the app. The main element of the app is the orders that can be placed by the customer. Executors can propose their remuneration for the services they provide for free orders, if the customer accepts their offer the executor has to fill the order.

The user interface is implemented using React native technology, the server side using Express.js and the application data is stored in MySQL database system. The application is easily extensible in the future. The application can be run both in a web browser and on mobile phones.

A Obsah příloženého CD

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod

src/

Kompletní zdrojové kódy aplikace LaundryApp se všemi soubory potřebnými pro bezproblémové vytvoření databáze, serveru a klientské aplikace.

readme.txt

Kompletní instrukce pro instalaci a spuštění aplikace LaundryApp, včetně odkazů pro bezproblémové nainstalování potřebných aplikací.

Literatura

- [1] *React Native*. Dostupný z: <https://reactnative.dev/>.
- [2] *Expo*. Dostupný z: <https://docs.expo.dev/>.
- [3] Brown, Ethan. *Web Development with Node and Express*. First. 2014. ISBN 978-1-491-94930-6.
- [4] Kofler, Michael. *The Definitive Guide to MySQL5*. Third. 2005. ISBN 1-59059-535-1.
- [5] Peyrott, Sebastián E. *The JWT Handbook: the next decade for digital libraries*. 2016-2018. Dostupný také z: <https://auth0.com/resources/ebooks/jwt-handbook>.