

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Pokladní aplikace pro kavárnu na platformě Android



2023

Vedoucí práce:
Mgr. Radek Janoščík, Ph.D.

Matěj Dostál

Studijní program: Informatika,
Specializace: Programování a vývoj
software

Bibliografické údaje

Autor: Matěj Dostál
Název práce: Pokladní aplikace pro kavárnu na platformě Android
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2023
Studijní program: Informatika, Specializace: Programování a vývoj software
Vedoucí práce: Mgr. Radek Janošík, Ph.D.
Počet stran: 53
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Matěj Dostál
Title: Cash register application for a cafe on Android platform
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2023
Study program: Computer Science, Specialization: Programming and Software Development
Supervisor: Mgr. Radek Janošík, Ph.D.
Page count: 53
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

Intuitivní a přehledný pokladní systém je pro bezproblémový provoz kaváren a dalších restauračních zařízení stěžejní. V rámci této bakalářské práce byla navržena a vyvinuta pokladní aplikace pro mobilní telefony a tablety s operačním systémem Android. Tato aplikace umožňuje jednoduše vytvářet objednávky, slučovat je či přesouvat, platit zvlášť, vytvářet účtenky a tisknout je. Poskytuje také přehled statistik za vybrané časové období. Vše je implementováno s důrazem na jednoduchost zadávání a uživatelskou přívětivost.

Synopsis

An intuitive and clear cash register system is crucial for smooth operation cafés and other restaurant establishments. In this bachelor thesis, cash register application was designed and developed for Android mobile phones and tablets. This app allows to easily create orders, merge or move them, pay separately, create receipts and print them. It also provides an overview of statistics for a selected time period. Everything is implemented with an emphasis on ease of entry and user-friendliness.

Klíčová slova: Android; Kotlin; kavárna; kasa

Keywords: Android; Kotlin; cafe; cash register

Děkuji Mgr. Radku Janoščíkovi, PhD. za velmi užitečné rady, nápady a zkušenosti při vytváření bakalářské práce. Děkuji také Jakubu Jurkovi za navržení a vytvoření loga aplikace. Dále bych chtěl poděkovat rodině, přítelkyni a kamarádům za jejich nápady, rady a podporu. V neposlední řadě děkuji Lucii Uhrové za její zpětnou vazbu k celé mé práci.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	7
1.1	Analýza provozu	7
1.2	Cíle	8
2	Existující řešení	10
2.1	Dotykačka	10
2.2	Storyous	10
2.3	Septim Air	11
2.4	KASAmx	11
2.5	Smable	12
3	Programátorská dokumentace	13
3.1	Použité technologie	13
3.1.1	Kotlin	13
3.1.2	XML	14
3.1.3	SQLite	14
3.2	Použité knihovny	15
3.2.1	MPAndroidChart	15
3.2.2	EasyPermissions-ktx	15
3.2.3	JBCrypt	15
3.2.4	Printooth	15
3.2.5	Kotlinx coroutines	16
3.2.6	Android Jetpack	16
3.2.7	Material Design	17
3.3	Architektura aplikace	18
3.3.1	Aktivity	18
3.3.2	Fragmenty	19
3.3.3	Single activity application	20
3.3.4	Model-View-ViewModel	21
3.4	Zajímavé části projektu	22
3.4.1	Tisk	22
3.4.2	Adaptivní layout	23
3.4.3	Vytváření tabulek na účtence	25
4	Uživatelská dokumentace	26
4.1	Spuštění	26
4.2	Kavárna	26
4.2.1	Detail stolu	27
4.2.2	Placení	29
4.2.3	Přesun	29
4.2.4	Sleva	31
4.3	Tržba směny	32
4.3.1	Pohyb hotovosti	33

4.3.2	Uzavření směny	33
4.4	Přehled	34
4.4.1	Statistiky	34
4.4.2	Historie	36
4.4.3	Směny	37
4.5	Nastavení	38
4.5.1	Stoly	39
4.5.2	Produkty	40
4.5.3	Účtenka	42
4.5.4	Denní nabídka	44
4.5.5	Objednávkové bony	44
4.5.6	Uživatelé	44
4.5.7	Otevírací doba	45
4.5.8	Jednorázový prodej	45
4.5.9	Plátcí DPH	46
	Závěr	47
	Conclusions	48
	A Zpětná vazba majitelky kavárny	49
	B Obsah elektronických dat	50
	Literatura	51

1 Úvod

V kavárnách se během dne vystřídají desítky lidí – každý si něco objedná, poté si někdo přisedne, přiojedná si nebo naopak po chvíli objednávku zruší. Pokud kavárna nemá dobrý, intuitivní systém, pomocí kterého může snadno evidovat objednávky na jednotlivých stolech, tisknout objednávkové bony a například jednoduše zjistit tržbu směny nebo statistiky dne a veškeré tyto věci si eviduje pomocí ručně psaných papírků, snadno se dostane do problémů.

V průběhu čtyř let jsem v takových kavárnách pracoval. V žádné jsme nepoužívali systém, který by umožňoval výše zmíněné, nebo to nebylo uživatelsky natolik přívětivé a intuitivní, aby to skutečně ulehčilo práci.

1.1 Analýza provozu

V první kavárně, kde jsem pracoval, nebyl systém žádný. Běžně jsme fungovali s papírky a objednávky psali ručně. Tyto papírky pak obsluha položila na bar baristům, kteří podle nich chystali objednávky. Často byl problém s nečitelností rukopisu jednotlivých lidí – speciálně v případě, že k zaběhnuté zkratce produktu byla dopsána nějaká poznámka¹. Tyto papírky byly jedinou evidencí toho, co si zákazník skutečně objednal, a zároveň na konci směny to byl jediný zdroj informací pro počítání tržby. Stačilo tedy, aby se někomu něco vylilo, polil tím již zmíněné papírky a problém byl na světě. Poměrně často se také dělo, že obsluha při objednávání omylem napsala chybné číslo stolu, nikdo si toho nevšiml, protože objednávku odnášel stejný člověk, který ji psal, a v momentě, kdy chtěl zákazník platit, jsme zjistili, že máme dvě objednávky na jednom stole a museli jsme se zákazníkem postupně procházet co měl, abychom určili, co má vlastně platit. Sčítání částek k zaplacení navíc obsluha počítala z hlavy, často se dívala do nápojového lístku, kolik co vlastně stojí, a přitom to sčítala. Což v situaci, kdy je plná kavárna a na baru se hromadí objednávky, není příjemné a v důsledku to vede k dalším chybám.

Dalším problémem, který jsme kvůli tomuto způsobu fungování museli řešit, bylo již zmíněné počítání tržby na konci dne. Vzali jsme si kalkulačku, všechny papírky s objednávkami za daný den, kterých byly desítky, a sčítali jsme částky. Toto sčítání se muselo provést několikrát, aby se co nejvíce zamezilo chybě, takže když po dvou sčítáních vyšly různé částky, museli jsme počítat znovu.

Ve druhé kavárně jsme již měli tablet s pokladním systémem Storyous². Měli jsme tedy sice pokladní systém, ale práci neulehčoval tak, jak bych si představoval. Možná to bylo tím, že tento systém byl v té době necelé 3 roky starý a ještě ne natolik vyvinutý a spolehlivý. Měli jsme nadefinovanou nabídku produktů, stoly. Markovat produkty na jednotlivé stoly bylo tedy možné, což byla

¹např. že zákazník nechce posyp muškátovým oříškem

²viz sekce 2.2

pomoc, ale s tím spojený problém byl, že nešlo například označit, které dorty z celé nabídky jsou daný den k dispozici a musel jsem je hledat mezi všemi.

Hlavním problémem, který práci brzdil dle mého nejvíce, byla nespolehlivost tisku objednávkových bonů. Při zadání objednávky v tabletu se sice spustil tisk, ale ten často neproběhl. Musel jsem si tedy objednávky opět psát ručně a poté je zadat do tabletu na příslušný stůl. Další, pro mě jako tehdejšího uživatele, nepohodlnou věcí byl fakt, že stoly nešly rozdělit do jednotlivých místností. Všechny byly zobrazeny v seznamu pod sebou, takže mi chvilku trvalo, než jsem se v tom spolehlivě zorientoval. Co se počítání tržby na konci dne týče, byl sice k dispozici přehled vytvořených účtenek, ale bez zaškolení jsem ho nedokázal najít.

V obou zmíněných kavárnách se navíc na konci směny vybíraly peníze z pokladny nebo na začátku směny se naopak peníze do pokladny vložily, aby obsluha měla na začátku dne z čeho vracet. Nikde nebyl přehled kdo, kdy a kolik peněz do pokladny vložil či vybral. Snadno tak mohlo dojít k nesrovnalostem.

Dalším způsobem, kterým mohlo k nesrovnalostem dojít, bylo, že papírek s objednávkou mohla obsluha místo zaplacení vyhodit (omylem nebo úmyslně, například v případě návštěvy kamarádů dotyčného), a tím způsobit ztrátu peněz podniku.

1.2 Cíle

Na základě výše popsaných osobních zkušeností s prací v kavárně a (ne)přítomností pokladního systému bych chtěl vytvořit takový systém, který by poskytoval veškerou základní funkcionalitu potřebnou pro bezproblémový chod kavárny, ulehčil co nejvíce práci obsluze a zároveň vyvažoval nedostatky, které jsem vyzoroval při své práci v kavárně.

Jako hlavní priority, které by měl systém splňovat, aby pokryl veškeré problémy popsané v sekci 1.1, jsem si stanovil několik bodů:

- **Intuitivní ovládání** – co nejvíce klást důraz na intuitivnost rozložení ovládacích prvků a vzhledu aplikace, aby uživateli stačilo krátké zaškolení nebo jedno přečtení dokumentace pro schopnost základního ovládání aplikace. Tím je myšleno například markování produktů na stoly, přesouvání objednávek mezi stoly nebo vytváření účtenek.
- **Konfigurovatelnost** – udělat nastavení aplikace co nejvíce obecné, aby bylo schopné pojmut co největší množství podniků. Aplikace by měla poskytovat možnost například přidávat místnosti a do nich stoly. Stoly libovolně rozmístit po místnosti tak, aby jejich rozložení co nejvíce odpovídalo skutečnému rozložení kavárny. Toto například umožňuje v praxi často používané řešení účtů zaměstnanců. Stačí přidat místnost „Zaměstnanci“, ve které má každý zaměstnanec svůj virtuální stůl, na kterém má namarkovanou svou útratu.

- **Objednávkové bony** – umožnit spolehlivě tisknout objednávkové bony. Jakmile obsluha udělá změnu na nějakém stole, při uložení se vytiskne (je-li to v nastavení povoleno) objednávkový bon, na kterém jsou všechny potřebné informace pro obsluhu.
- **Poznámky k objednávce** – možnost zadání libovolné poznámky k produktům, která se vytiskne obsluze na objednávkovém bonu. Může to být poznámka typu „udělat kávu z alternativního mléka“³ apod.
- **Stornování produktů** – stornování již namarkovaných produktů by mělo být možné pouze po zadání poznámky, proč ke stornování dochází, a stornované položky zůstanou viditelné na daném účtě, i v historii (ne na vytisknuté účtence).
- **Přehled směny** – přehledně zobrazovat tržbu a počet účtenek za danou směnu k okamžiku zobrazení tohoto přehledu. Zobrazovat také všechny příslušné účtenky s možností k libovolné účtence zobrazit veškeré podrobnosti a znovu ji vytisknout. Uzavřením směny se vynuluje zobrazení těchto informací a tím začne nová směna.
- **Stav pokladny** – umožnit evidovat pohyby hotovosti ve fyzické pokladně a celkovou částku, která by měla v pokladně v daný okamžik být. Umožnit zadání vkladu/výběru hotovosti s poznámkami v přehledu směny. Obsluha směny na začátku, i kdykoli v průběhu, vidí, kolik peněz má v pokladně k dispozici.
- **Statistiky** – zobrazit statistiky prodeje a hodinového vytížení kavárny v podobě grafů, spolu s přehledy jednotlivých dní za libovolně vybrané časové období. Umožní tak i zpětnou kontrolu tržeb a tím větší kontrolu nad chodem podniku.
- **Správa uživatelů a oprávnění** – přidávat uživatele v nastavení s příslušnou úrovní oprávnění. Na základě tohoto oprávnění poté bude určeno, co v aplikaci mohou nebo nemohou dělat.

³alternativním mlékem se rozumí jiné než kravské

2 Existující řešení

Po zanalyzování základních potřeb pro usnadnění práce obsluhy a celkově pohodlný běh kavárny (viz sekce 1.1) jsem prozkoumal několik aplikací/systémů, které jsou zaměřené na stejnou problematiku. Vybral jsem pouze systémy, které jsou dostupné i na operačním systému Android, jelikož mé řešení je určeno pro zmíněný operační systém.

2.1 Dotykačka

Dotykačka [1] je asi největším, dokonce mezinárodním, systémem řešícím danou problematiku. Tento systém je velmi komplexní a nabízí řešení jak provozu kavárny či restaurace, tak možnost evidence skladů, mobilního číšníka, rezervací a online objednávek. Je propojena s mnoha službami třetích stran⁴. Dotykačka také nabízí řešení pro menší hotely nebo penziony, malé prodejny či služby. Obrovskému nárůstu se systém těšil s příchodem EET⁵, kdy velmi rychle tuto funkcionalitu implementoval a nabízel kompaktní řešení i pro ty nejmenší prodejce či podniky.

Systém je řešen cloudově, data lze tedy zobrazovat a spravovat i na dálku, což je nesporná výhoda. Co se funkcionalit týče, nabízí řešení v podstatě všech bodů zmíněných v sekci 1.2. Nicméně jako vše, i cloudové řešení má své nevýhody. První potenciální nevýhodou je, že pro provoz je potřeba Internet⁶, nicméně toto systém řeší zpětným nahráním dat, až je internet opět dostupný. Další nevýhodou je bezpečnost. Data v cloudu mohou být ohrožena hackerskými útoky a dalšími bezpečnostními hrozbami. Mezi lidmi často zmiňovanou nevýhodou je způsob platby. Dotykačka má platby postavené na bázi měsíčního předplatného, což není u takovýchto systémů příliš populární. Nabízí tři druhy předplatného: Neomezeně, Naplno a Snadno. Předplatné „Neomezeně“ stojí cca. 1 200 Kč/měsíc, „Snadno“ vyjde na cca 420 Kč/měsíc.

2.2 Storyous

Systém Storyous [3], nyní již spadající pod společnost SaltPat [4], je dalším velmi rozšířeným cloudově řešeným pokladním systémem. Stejně jako výše zmíněná Dotykačka poskytuje online přehled tržeb či obsazenosti. Nabízí také množství rozšiřujících funkcionalit, jako je evidence skladů, mobilní číšník, rozdělení uživatelů podle rolí či napojení na služby třetích stran⁴.

S tímto systémem jsem měl možnost pracovat v jedné z kaváren zmíněných v sekci 1.1. Práce s ním měla své výhody, ale i nevýhody. Ty jsou popsány v již zmíněné sekci 1.1.

⁴Jako např. Qerko [2] – aplikace umožňující zákazníkům platit, případně i objednávat, přímo od stolu za pomoci QR kódů.

⁵Dnes již zrušená elektronická evidence tržeb

⁶Byť samozřejmě v dnešní době jen v málo které kavárně či restauraci není dostupný internet skrze Wi-Fi.

Nevýhodou je opět licence postavená na měsíčním předplatném, jejíž cena se pohybuje okolo 700 Kč/měsíc⁷. Navíc, podle rozhovoru s některými majiteli kaváren, jsou tyto měsíční platby docela vysoké na jejich potřeby a vzhledem k jejich výdělkům. Ve srovnání s ostatními prozkoumanými systémy se jeví Storyous jako nejdražší, nicméně zákazníci si jej velmi chválí.

2.3 Septim Air

Septim Air [5] je cloudová verze systému Septim od společnosti ASW Systems [6]. Jedná se opět o velmi rozšířený pokladní systém. Nabízí také širokou škálu funkcionalit pro gastro podniky. Tento systém je dodáván zároveň s příslušným hardwarem (dotyková pokladna, tiskárny, platební terminál), který má zákazník pouze v pronájmu. Tento fakt se ovšem projevuje také na ceně – počáteční instalace, úvodní školení a základní nastavení vyjde zákazníka na 8 900 Kč. Následně je provoz, stejně jako tomu bylo u Storyous a Dotykačky, postaven na měsíčním tarifu, který u tohoto systému činí 3 900 Kč⁸. Tento systém je tedy dobrou volbou pro toho, kdo s instalací a provozem chce mít co nejméně práce a zároveň si to může finančně dovolit.

2.4 KASAmox

Tato společnost [7] vznikla původně jako e-shop pokladen, tiskáren a ostatního příslušenství pro pokladny. Postupem času ovšem začala vyvíjet svůj pokladní systém, který dodávají samostatně nebo spolu s hardwarem. Systém umožňuje prodávat jak v režimu pouze s číselníkem – tedy pomocí numerické klávesnice, kterou se zadá požadovaná částka, v dalším kroku se zadá sazba DPH a následně dojde k přičtení částky k celkové ceně účtu; tak v režimu s položkami, kdy uživatel kliká přímo na jednotlivé položky a tím je přidává na účet.

Pokladní systém KASAmox lze používat i v bezplatné verzi, která je však značně omezena – umožňuje například vytvořit pouze 30 účtenek během 30 kalendářních dnů. Samozřejmě lze zakoupit licenci, která je nabízena ve třech variantách: půlroční, roční a neomezená. Jako první prozkoumané existující řešení není provoz tohoto systému postaven na měsíčních platbách, ale licence se kupuje jednorázově⁹. Cena licencí se odvíjí od vybraného druhu, nejlevnější je 1 990 Kč za půlroční a nejdražší 9 990 Kč za neomezenou. Nákup tedy může být pro podnik jednorázovou investicí, což je výhoda.

⁷V ceně není zahrnut poplatek za pronájem hardwaru jako je např. platební terminál.

⁸V této ceně je zahrnutý také pronájem hardwaru a telefonická podpora

⁹Samozřejmě, pokud si zákazník koupí půlroční licenci a po půl roce ji kupuje znovu, už to jednorázové není, ale pořád i s touto licencí ho měsíc vyjde na přibližně 330 Kč, což je v porovnání s ostatními zmíněnými existujícími řešeními levnější.

2.5 Smable

Systém Smable [8] je nejmenším z prozkoumaných řešení z hlediska počtu zákazníků. Co se funkcionality týče, je také značně pokročilý – řeší správu skladů, mobilního číšníka a online přehledy. Systém je opět řešen cloudově a prodej je také možné provozovat pouze s číselníkem nebo pomocí markování položek, jako tomu bylo u předchozího zmíněného systému KASAmx.

Jako jediný systém ze zmíněných existujících řešení jsem si tento systém v rámci průzkumu vyzkoušel na jejich bezplatné demo verzi. Za první nevýhodu považuji to, že není možné k jednotlivým stolům nadefinovat jejich rozložení v místnosti. Lze pouze vytvářet seznam objednávek, které lze přiřadit na stůl. Tato chybějící funkcionality není omezením demo verze, není ani v placené verzi. Co považuji za nevýhodu je nastavení, které je řešeno pomocí tzv. „Backoffice“, což je webová aplikace umožňující nastavovat veškeré údaje a položky, zobrazovat přehledy a podobně. Pro nastavení v aplikaci se Backoffice zobrazuje v rámci aplikace pomocí `WebView`¹⁰ a je to velmi pomalé – po napsání znaku do textového vstupu někdy trvá i řádově pár vteřin, než se propíše, a v nastavení celkově se mi nepovedlo příliš zorientovat. Jinak je ovládání pokladny docela příjemné a intuitivní.

Co se platby za systém týče, opět to postaveno na měsíčním předplatném, nicméně ceny jsou mnohem přívětivější. Což je samozřejmě dané i tím, že je to menší systém než ostatní zmíněné a umí toho méně.

¹⁰Komponenta pro zobrazení webové stránky přímo v aplikaci bez nutnosti přepínat obrazovku na prohlížeč.

3 Programátorská dokumentace

Mnou navržená a vyvíjená aplikace Brewify je určena pro mobilní telefony a tablety s operačním systémem Android verze 12 a vyšší. Při používání na zařízeních s nižší verzí Androidu nemusí vše fungovat tak, jak by mělo. Je doporučováno aplikaci používat na tabletu pro co nejlepší ovládání, nicméně funkcionality je plně dostupná i na telefonech. Od zvoleného operačního systému se odvíjí jak použité technologie a knihovny, tak zvolená architektura aplikace. Vše zmíněné je popsáno v této kapitole.

3.1 Použité technologie

Pro vývoj aplikace jsem použil následující technologie. Jako vývojové prostředí jsem zvolil Android Studio, což je prostředí pro vývoj mobilních aplikací pro operační systém Android. Obsahuje mnoho nástrojů pro vývoj mobilních aplikací, včetně editoru kódu, nástrojů pro návrh uživatelského rozhraní, emulátoru pro testování aplikací a mnoho dalších, které vývojářům pomáhají vytvářet kvalitní a funkční aplikace.

3.1.1 Kotlin

Veškerá logika aplikace je psána v jazyce Kotlin [9]. Kotlin je moderní jazyk, který byl vyvinut společností JetBrains a byl navržen jako alternativa k jazyku Java¹¹ s cílem zjednodušit vývoj aplikací pro platformu Java a Android. V komunitě vývojářů aplikací pro Android se Kotlin stal tak populárním, že jej Google v roce 2017 přijal jako oficiální jazyk pro nativní vývoj mobilních aplikací pro Android a od té doby jeho popularita stále roste.

Nyní je tento jazyk open source, což znamená, že i vývojáři mimo JetBrains a Google do něj mohou přispívat a zlepšovat jej. Nicméně je s Javou velmi úzce spjat, například třídy v Kotlinu jsou interně namapovány na třídy z Java Class Library¹². Kotlin je možné kompilovat a pouštět v JVM¹³. [10]

Oproti Javě má však mnoho výhod. První z nich je bezpochyby jednodušší a čitelnější syntaxe. V Kotlinu například není třeba psát středníky na konci řádku nebo lze definovat jednořádkové funkce (viz kód 1). Ideologie Kotlinu je zbavit programování „zbytečného psaní“ a dát programátorovi to, co chce, s minimem napsaného kódu, což se podle mě velmi dobře daří. Další výhodou je, že novinky se do tohoto jazyka dostávají mnohem rychleji než do Javy. Je to hlavně díky tomu, že Kotlin je podstatně mladší jazyk a má tedy méně závislostí na stávajících funkcích.

¹¹do té doby byla oficiálním a jediným, společností Google přijatým, jazykem pro nativní vývoj mobilních aplikací právě Java

¹²balík dynamicky načítaných knihoven, které mohou jazyky podporované JVM za běhu využívat

¹³Java Virtual Machine

```

1 // definice funkce v Jave
2 private int addFive(int arg) {
3     return arg + 5;
4 }
5
6 // definice v Kotlinu
7 private fun addFive(arg: Int): Int = arg + 5

```

Zdrojový kód 1: Ukázka definování funkce přičítající hodnotu 5 k zadanému argumentu v Javě a Kotlinu.

3.1.2 XML

Uživatelské rozhraní lze v Android Studiu definovat pomocí XML [11]. XML je jazyk, který pomocí značek a atributů reprezentuje data, která jsou díky tomu snadno čitelná jak pro člověka, tak pro počítač. Celkové uživatelské rozhraní (UI) aplikace se skládá ze dvou základních prvků: `View` a `ViewGroup`. `View` je typický prvek, který je vykreslen na obrazovku a uživatel jej vidí a může s ním případně interagovat. `ViewGroup` je naopak neviditelný kontejner, který definuje strukturu rozložení pro další prvky `View` a `ViewGroup`.

Objekty `View` se nazývají „widgety“ a mohou být podtřídami jako například `Button` nebo `TextView`. Objekty `ViewGroup` se nazývají „layouts“ a definují různé struktury pro rozložení UI; například `LinearLayout` nebo `ConstraintLayout`.

3.1.3 SQLite

SQLite [12] je open source databáze, která umožňuje běžnou práci s relační databází a persistentní uložení dat. Tato databáze je zabudovaná v Androidu, její využívání je tedy velmi jednoduché.

API¹⁴, které Android poskytuje v základu, pro práci s touto databází je plně dostačující, avšak jeho přímé použití se již moc nedoporučuje. Je to kvůli tomu, že toto API neposkytuje kontrolu SQL dotazů v době kompilace a jednotlivé dotazy je při změně struktury tabulky potřeba ručně upravit, což může být časově náročné a také v tom lze snadno udělat chyby. Další nevýhodou je, že pro převedení dat získaných z databáze na nějakou kolekci objektů je potřeba napsat hodně nadbytečného kódu.

Pro práci s SQLite databází se tedy aktuálně doporučuje používat knihovnu `Room persistence library`, která je součástí balíčku `Android Jetpack`. Tato knihovna poskytuje vrstvu abstrakce pro práci s databází a poskytuje mnoho moderních mechanismů pro práci. Více o tomto balíčku i knihovně `ROOM` v sekci 3.2.6.

¹⁴Application program interface

3.2 Použité knihovny

V této podkapitole krátce představím knihovny použité napříč mou aplikací. Více detailů o každé z knihoven lze najít v příslušném zdroji.

3.2.1 MPAndroidChart

Díky této knihovně [13] lze jednoduše vytvářet grafy mnoha druhů a pracovat s nimi. Mezi druhy grafů, které lze díky této knihovně vytvořit, patří například: čárový graf, sloupcový (vertikální, horizontální, skupinový či skládaný), koláčový nebo tzv. radarový graf používaný pro zobrazení vícerozměrných dat ve dvou-rozměrném prostoru. Grafy lze dále libovolně zvětšovat/zmenšovat či animovat. Tato knihovna tedy umožňuje pokročilou práci s grafy, přičemž zachovává uživatelskou přívětivost. Jednoduchost práce s touto knihovnou navíc podporuje přehledná dokumentace s velkým množstvím příkladů i videoukázkou použití.

3.2.2 EasyPermissions-ktx

Knihovna EasyPermission-ktx [14] je odnoží knihovny EasyPermissions [15], které umožňuje tuto knihovnu používat i v Kotlinu. Knihovna řeší vyžadování oprávnění, které aplikace potřebuje povolit od uživatele za běhu. Na rozdíl od originální implementace řešení tohoto problému v Androidu je řešení touto knihovnou pro programátora elegantní a přehledné. Velmi jednoduše tak lze pracovat jak s přidělováním různých oprávnění potřebných pro běh aplikace, tak i definovat reakce na situace, kdy uživatel některé oprávnění odmítne udělit a podobně.

3.2.3 JBCrypt

Běžně dostupná hashovací funkce v Javě je SHA-256. Tento způsob hashování není však pro ukládání hesel optimální z několika důvodů. Historicky tento algoritmus nebyl vyvinut pro hashování hesel a je náchylný na běžné útoky, jako jsou útok hrubou silou či *Rainbow table* útok. Jeho pozitivní stránkou ovšem je, že je rychlý a vypočítání příslušného hashe je levnější z pohledu výpočetního výkonu.

Oproti tomu BCrypt byl vyvinut za účelem bezpečného hashování hesel, takže při vytváření hashe vyžaduje ještě tzv. sůl, která se přidá k heslu (dojde k tzv. „osolení“ hesla) a tím je výsledný hash těžší uhádnout/najít. Knihovna JBCrypt [16] je implementací tohoto způsobu hashování pro programovací jazyk Java (tudíž i pro Kotlin) a pomocí dvou funkcí lze jednoduše získat hash hesla a další funkcí ověřit, zda se uložené a uživatelem zadané heslo shodují.

3.2.4 Printooth

Cílem této knihovny [17] je poskytnout uživateli jednoduchou abstrakci pro používání Bluetooth tiskáren bez ohledu na jejich značku, model či jiný technický parametr.

Práce s ní je velmi jednoduchá – stačí pouze získat od uživatele příslušná oprávnění (toho jsem docílil použitím knihovny `EasyPermissions`, viz sekce 3.2.2), poté na jednom místě v aplikaci `Printooth` inicializovat a spárovat se zařízením a pak už stačí pouze posílat požadovaná data k vytištění na spárovanou tiskárnu. Pro párování s tiskárnou stačí spustit tzv. skenovací aktivitu, která zobrazí dostupná Bluetooth zařízení v okolí a uživatel jen vybere svou tiskárnu. Vše je přehledně popsáno v dokumentaci této knihovny na příkladech.

3.2.5 Kotlinx coroutines

Coroutines je mechanismus umožňující asynchronní vykonávání programu za pomoci běžných výrazů používaných v sekvenčním programování. Navrženy jsou tak, aby umožnily jednoduchou a efektivní asynchronní komunikaci mezi vlákny, přičemž minimalizují režii vláken a snižují nároky na systémové prostředky.

Coroutines ale nejsou doopravdy samostatná vlákna. Idea je stejná, ale coroutines se nevážou na žádné konkrétní vlákno. Vykonávání může být v jednom vlákně pozastaveno a v jiném opět spuštěno. Dalo by se jim říkat „light-weight“ vlákna, ale mají mnoho rozdílů. [18]

Asynchronní programování je ve vývoji mobilních aplikací důležité pro zachování co nejlepšího UX, neboli user experience. Například tedy práce s databází se provádí v separátním vlákně, rozumějte coroutině, a uživatelské vlákno, ve kterém běží uživatelské rozhraní, zůstane neblokované a obrazovka tak „nezamrzne“. Podpora coroutines je pro Kotlin poskytována v knihovně `kotlinx.coroutines`. [19]

3.2.6 Android Jetpack

Jelikož vývoj mobilních aplikací se v posledních letech stal velmi populárním a důležitým, vznikají balíčky a knihovny pro vývojáře, které mají vývoj aplikací pro Android co nejvíce usnadnit.

Hlavním takovým balíčkem pro vývoj je `Android Jetpack`, což je balíček skládající se z množství knihoven, díky kterým mohou vývojáři snadno dodržovat ověřené praktiky, zbavovat se redundance kódu a psát kód funkční napříč verzemi Androidu. [20]

V souvislosti s architekturou aplikace (viz sekce 3.3) poskytuje tento balíček velmi užitečné komponenty a mechanismy. Nyní uvedu pár příkladů, které využívám ve své aplikaci:

- **Navigation component** – [21] tato komponenta usnadňuje navigaci mezi jednotlivými obrazovkami – fragmenty aplikace. Hlavní částí této komponenty je navigační graf, ve kterém jsou zobrazeny všechny obrazovky aplikace, a my můžeme jednoduše definovat navigaci mezi nimi. Můžeme také definovat tzv. kořenový fragment, tedy fragment, který se zobrazuje jako

první při spuštění. Tato komponenta pak také zajišťuje několik funkcionalit, jako například transakce mezi jednotlivými fragmenty nebo používání navigačních UI komponent¹⁵ s minimem práce navíc.

- **ViewModel** – [22] tato třída poskytuje uživatelskému rozhraní data a zapouzdřuje příslušnou logiku. Její hlavní výhodou je, že data přežijí změny konfigurace¹⁶, tedy přežívají životní cyklus aktivity/fragmentu. Díky tomu nemusí docházet k opětovnému načítání dat – UI se pouze dotáže svého ViewModelu a ten mu je poskytne ze své paměti.
- **Room** – [23] celým názvem Room persistence library, je knihovna poskytující abstrakci nad lokální SQLite databází, přičemž zachovává plnou sílu SQLite. Hlavními výhodami, které tato knihovna poskytuje, jsou:
 - kontrola SQL dotazu v době kompilace
 - využívání anotací pro jednoduchou a pohodlnou práci minimalizující opakující se kód
 - spolupráce s dalšími mechanismy, resp. komponentami (např. LiveData)

Room jako celek se skládá ze tří hlavních částí (viz [Obrázek 1](#)):

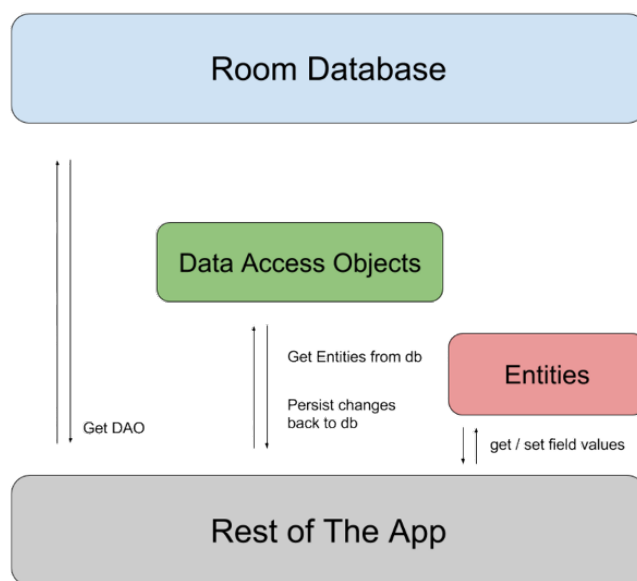
1. Databáze – tato třída obsahuje databázi a slouží jako hlavní přístupový bod do databáze pro zbytek aplikace.
2. Entity – třídy reprezentující jednotlivé tabulky v databázi. Opět pomocí anotací lze definovat primární, případně cizí klíč, jména jednotlivých sloupců a podobně.
3. Data access objects (DAOs) – třída poskytující metody pro práci s databází: získávání, vkládání, mazání nebo aktualizování záznamů. Tohoto lze v Roomu dosáhnout využitím anotací. Například pomocí anotací: `@Insert`, `@Update` nebo `@Delete` lze jednoduše vkládat, aktualizovat a mazat záznamy z příslušné tabulky bez psaní SQL dotazu. Chceme-li získat data vlastním dotazem, stačí použít anotaci `@Query` (...) a do závorek vložit požadovaný dotaz.

3.2.7 Material Design

Material Design [24] je designová filozofie vytvořená společností Google v roce 2014. Cílem této filozofie je vytvořit jednotný a konzistentní design aplikací a webových stránek napříč platformami. Material Design je navržený tak, aby

¹⁵Navigačními komponentami rozumíme například Bottom navigation view, navigation drawer a podobně.

¹⁶například otočení obrazovky



Obrázek 1: Tři základní části Room knihovny. Převzato z [23]

byl snadno přizpůsobitelný různým zařízením. K dispozici je mnoho předpřipravených komponent a stylů, aby bylo možné snadno vytvořit aplikace s konzistentním a moderním vzhledem.

Pro vývoj aplikací na Android je tedy tento design standardem a knihovny potřebné pro jeho použití jsou často automaticky součástí založeného projektu.

3.3 Architektura aplikace

Jak již bylo řečeno, aplikace je navržena a vyvíjena pro operační systém Android. Od tohoto faktu se odvíjí také použitá architektura a její návrhový vzor.

Abych mohl představit zvolenou architekturu a s ní související věci, je nejprve potřeba popsat základní prvky vývoje aplikací pro Android.

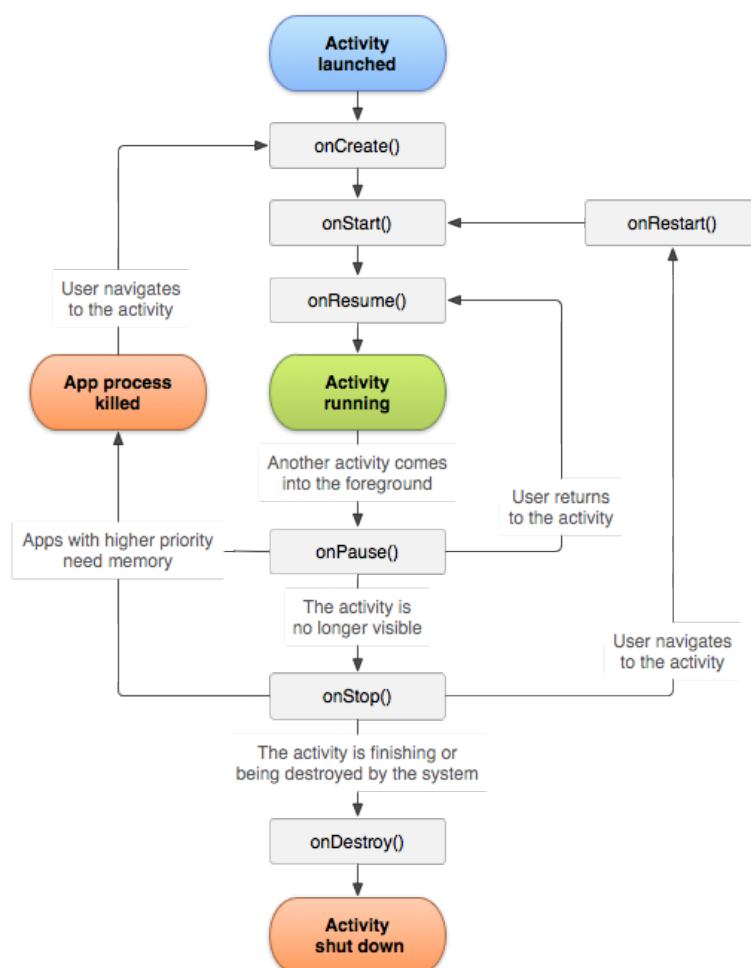
3.3.1 Aktivity

Aktivita je základním stavebním prvkem aplikací pro Android. Slouží v podstatě jako vstupní bod operačního systému do aplikace, takže místo spuštění celé aplikace jako celku dojde ke spuštění pouze jedné příslušné aktivity. Aktivitu si můžeme představit jako jednu obrazovku, která interaguje s uživatelem a obsahuje tedy prvky uživatelského rozhraní. Aktivity jsou navzájem nezávislé a každou lze spustit a ukončit samostatně. Jedna aktivita je vždy označena jako tzv. „MainActivity“. Tato aktivita se zobrazí jako první při spuštění aplikace. [25]

Aktivity umožňují aplikaci přepínat mezi různými obrazovkami a poskytují způsob, jak přenášet data mezi různými částmi aplikace. Například aktivita seznamu položek může spustit aktivitu detailu položky, která zobrazuje více infor-

mací o vybrané položce. Aktivity také umožňují interakci s dalšími aplikacemi a službami v operačním systému Android. [26]

Každá aktivita má svůj životní cyklus (viz [Obrázek 2](#)), který je řízen operačním systémem. Životní cyklus zahrnuje různé stavy, jako je spuštění, pozastavení, obnovení či zničení aktivity a umožňuje aplikaci spravovat svůj stav a reagovat na události jako například zničení aktivity operačním systémem.



Obrázek 2: Životní cyklus aplikace – převzato z [27]

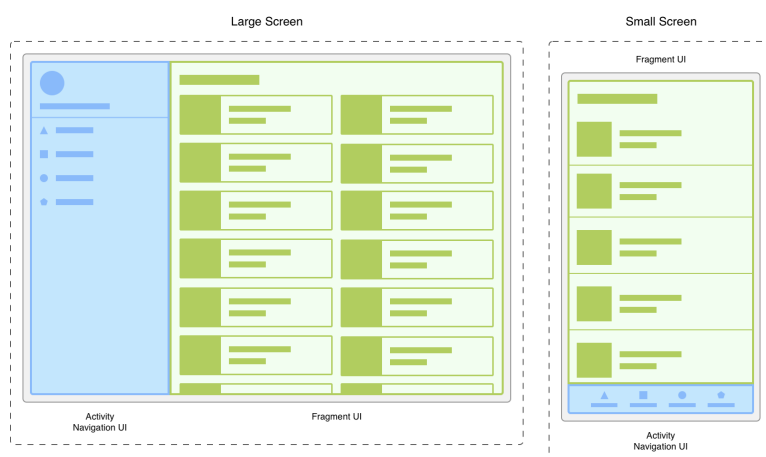
3.3.2 Fragmenty

Fragmentem můžeme rozumět jakousi „podaktivitu“ nebo „*light* aktivitu“. Fragmenty představují možnost rozdělit uživatelské rozhraní (UI) na menší znovupoužitelné části. Na rozdíl od aktivity, která představuje celé okno uživatelského rozhraní, fragment může obsahovat pouze část tohoto UI. Každý fragment má tedy své rozložení, ovládací prvky, svůj životní cyklus a může sám reagovat např. na vstupy od uživatele. Fragmenty nemohou existovat samy o sobě – musí být součástí jiné aktivity nebo fragmentu.

Stejně jako aktivity, poskytují i fragmenty možnost sdílení dat. Tato data mohou mezi sebou sdílet v rámci jedné aktivity, což velmi usnadňuje přenos dat v aplikaci a jejich správu a zobrazení v uživatelském rozhraní.

Fragmenty mohou být dynamicky libovolně vkládány či odstraňovány z aktivity. Přehled těchto změn je dostupný v tzv. `back-stacku`, neboli zásobníku, který spravuje aktivita a umožňuje obnovit předchozí stavy a tím reagovat například na situaci, kdy uživatel stiskne systémové tlačítko „Zpět“. Tato vlastnost dynamického přidávání a odstraňování umožňuje aplikaci reagovat na různé velikosti obrazovek a orientace zařízení. [28]

Díky tomu je možné vytvoření tzv. responzivního¹⁷ layoutu. Příklad takového layoutu za použití fragmentů lze vidět na [Obrázku 3](#).



Obrázek 3: Přizpůsobení velikosti obrazovky – na velké obrazovce je navigace zobrazena pomocí navigation drawer a napravo od ní je seznam položek s rozložením grid, oproti tomu na malé obrazovce je navigace zobrazena pomocí bottom-navigation a seznam položek je lineární. Převzato z [28]

3.3.3 Single activity application

Nyní již mohou popsat samotnou architekturu aplikace. Single activity application (SAA) je architektura, která se v roce 2018 stala standardem pro vývoj mobilních aplikací na Android. Ideou této architektury je nepoužívat pro každou obrazovku aplikace aktivitu, ale využívat pouze jednu hlavní aktivitu, která slouží jako kontejner pro fragmenty reprezentující jednotlivé obrazovky a udržovat je pomocí `back-stacku`. Hlavními výhodami tohoto přístupu je: [29]

- **Sdílení dat** – mezi fragmenty se data sdílejí mnohem jednodušeji než mezi aktivitami. Fragmenty mohou využívat aktivitu, rozumějte svůj kontejner,

¹⁷Responzivní znamená, že se rozložení UI automaticky přizpůsobí velikosti obrazovky tak, aby bylo místo efektivně využito a neobjevovaly se velké volné plochy nebo aby jednotlivé prvky nebyly moc široké nebo vysoké.

pro sdílení dat za pomoci ViewModelu navázaného na životní cyklus aktivity, který přežije výměnu fragmentů. Více o ViewModelech v sekci 3.2.6.

- **Testování uživatelského rozhraní** – jednoduše testovatelné izolované uživatelské prostředí.
- **Nezapořádání operačního systému pro změnu obrazovky** – každá aktivita při svém vytvoření či zničení musí informovat operační systém, což přidává aplikaci zbytečnou režii navíc. Použití fragmentů tedy méně zatěžuje OS.

3.3.4 Model-View-ViewModel

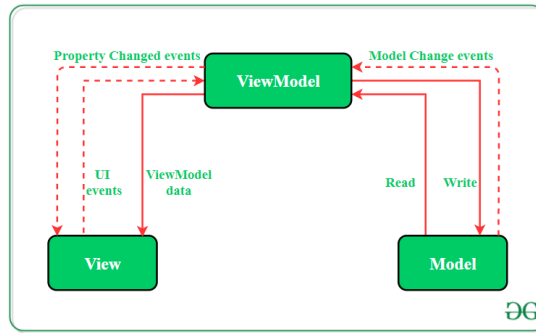
Model-View-ViewModel, neboli MVVM, [30] je návrhový vzor, který je balíčkem Android Jetpack (sekce 3.2.6) velmi dobře podporován. Ten totiž implementuje mnoho komponent pro práci s MVVM a dohromady vytváří velmi dobře funkční celek.

Základní myšlenkou tohoto návrhového vzoru je vytvořit třídu, která drží stav aplikace. Této třídy se pak dotazuje UI a vykresluje podle ní ovládací prvky. Naopak, pokud uživatel do UI zadá nějaké údaje, jsou automaticky propagovány do ViewModelu pomocí bindingu.

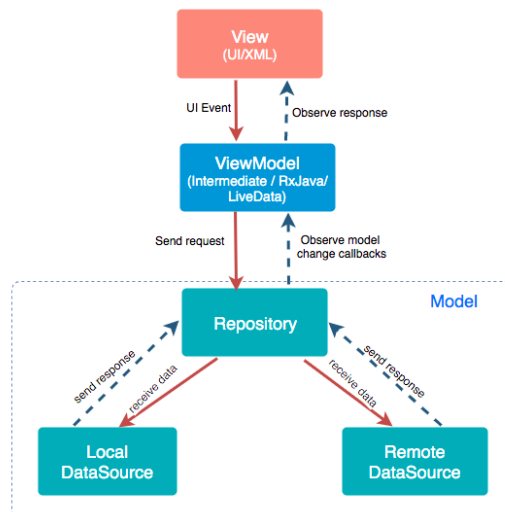
MVVM rozděluje kód do tří částí a tím umožňuje psát čistý, přehledný a snadno udržitelný a rozšiřitelný kód. Tyto tři části jsou:

- **Model** – popisuje data, se kterými aplikace pracuje. Model neví nic o stavu ovládacích prvků ani stavu aplikace jako celku. Model také spolupracuje s ViewModelem na načítání a ukládání dat.
- **View** – účelem této třídy je zobrazovat data z ViewModelu a informovat jej o uživatelských akcích. View pouze odebírá z ViewModelu a neobsahuje žádnou aplikační logiku.
- **ViewModel** – tato třída slouží k propojení předchozích dvou tříd a drží stav aplikace a její logiku. Ovládací prvky vykreslené ve View jsou propojeny pomocí bindingu s ViewModelem a čerpají z něj svůj obsah.

Jak již bylo řečeno, data v návrhovém vzoru MVVM poskytuje tzv. Model. Běžně se ve spojení s modelem využívá ještě tzv. repozitářů. Ty slouží k přidání vrstvy abstrakce nad samotným zdrojem dat. ViewModel potom komunikuje pouze s příslušným repozitářem a neví nic o zdroji dat, které obdržel. Díky tomu lze velmi jednoduše změnit například získávání dat z lokální databáze na jejich získávání posíláním požadavků serveru. Schéma této myšlenky je vidět na [Obrázku 5](#).



Obrázek 4: Schéma MVVM architektury. Převzato z [30]



Obrázek 5: Schéma MVVM s použitím repozitářů. Převzato z [31]

3.4 Zajímavé části projektu

V následující podkapitole stručně popíšu pár problémů řešených v aplikaci.

3.4.1 Tisk

Aktuální verze aplikace podporuje tisk objednávkových bonů a účtenek pouze přes Bluetooth tiskárny. Aby však nebyl problém přidat podporu dalších tiskáren, např. Wi-Fi či připojených přes kabel, je kód navržen tak, aby to bylo co nejjednodušší.

Třídy, které mají tisknout (v mé aplikaci tedy například detail stolu, který tiskne objednávkové bony, či detail účtenky, který tiskne samotné účtenky), musí implementovat rozhraní `Printable`. Třídy tak obsahují funkci `initPrinter` a objekt třídy `Printing`. Ve funkci `initPrinter` daná třída inicializuje objekt třídy `Printing` a v konstruktoru mu předá:

- Odkaz na hlavní aktivitu aplikace – pro přístup k jejímu `ViewModelu`¹⁸

¹⁸více o ViewModelech v sekci 3.2.6

a tím nastavení, které třída pro tisk potřebuje.

- Kontext – pro jednotlivé implementace komunikace s tiskárnou.

Dále je, v závislosti na použité technologii tisku, potřeba zavolat funkci `init`, která je dostupná přes objekt třídy `Printing`. Jako argument je předán callback pro zobrazování průběhu tisku, jelikož tuto informaci každá třída může chtít zobrazovat jinak. Poté již stačí zavolat funkci `printOrderBon` nebo `printReceipt`, které poskytuje objekt třídy `Printing`, a tím tisk spustit.

Třída `Printing` poskytuje vrstvu abstrakce nad zvolenou technologií tisku tak, aby se konkrétní technologie řešila v kódu tříd, které tisknou, co nejméně. Při inicializaci tato třída získá informaci, která technologie tisku se má použít, zda podporuje diakritiku a také maximální délku řádku. Podle technologie tisku pak volá metody příslušné třídy implementující tisk pomocí dané technologie. Pro přidání nové technologie tisku je potřeba přidat funkci `init` se specifickými argumenty přidávané technologie, které jsou potřeba. Dále je pak nutné rozšířit funkce `hasPairedDevice` a `removePrinter`. První z těchto funkcí vrací pravdivostní hodnotu, na základě které mohou třídy, které mají tisknout, zkontrolovat, zda je dostupná tiskárna či nikoli. Druhá funkce slouží k odebrání již přidané tiskárny z aplikace v nastavení.

O samotný tisk a komunikaci s příslušnou tiskárnou pomocí dané technologie slouží třídy implementující rozhraní `Printer`. Tyto třídy musí implementovat, kromě samotné komunikace s tiskárnou, funkce `printOrderBon` a `printReceipt`.

Pro přidání dalších technologií tisku stačí tedy implementovat příslušné callbacky a inicializaci ve třídách, které tisknou a zobrazují informace o průběhu a dále třídu, která komunikuje s daným hardwarem. Ve třídě `Printing` je potřeba přidat pro tuto technologii funkci `init` s příslušnými argumenty a rozšířit zmíněné funkce `hasPairedDevice` a `removePrinter`. Je také potřeba přidat tuto možnost technologie tisku do nastavení.

3.4.2 Adaptivní layout

Díky využití komponent, které to umožňují automaticky, dobrému rozdělení kódu do fragmentů¹⁹ a také díky definici některých layoutů pro různé velikosti obrazovky lze aplikaci používat jak na mobilních telefonech, tak na tabletech.

Jako komponentu/layout, který umožňuje vytvoření adaptivního layoutu automaticky, jsem použil `SlidingPaneLayout`²⁰. Tento layout je rozdělen na dva hlavní panely a hodí se tedy například na vytvoření layoutu nastavení, což jsem udělal.

Levý panel, který je zároveň startovní, slouží jako seznam obsahu. V layoutu nastavení tedy obsahuje seznam „kategorií“, které uživatel může rozkliknout a nastavovat.

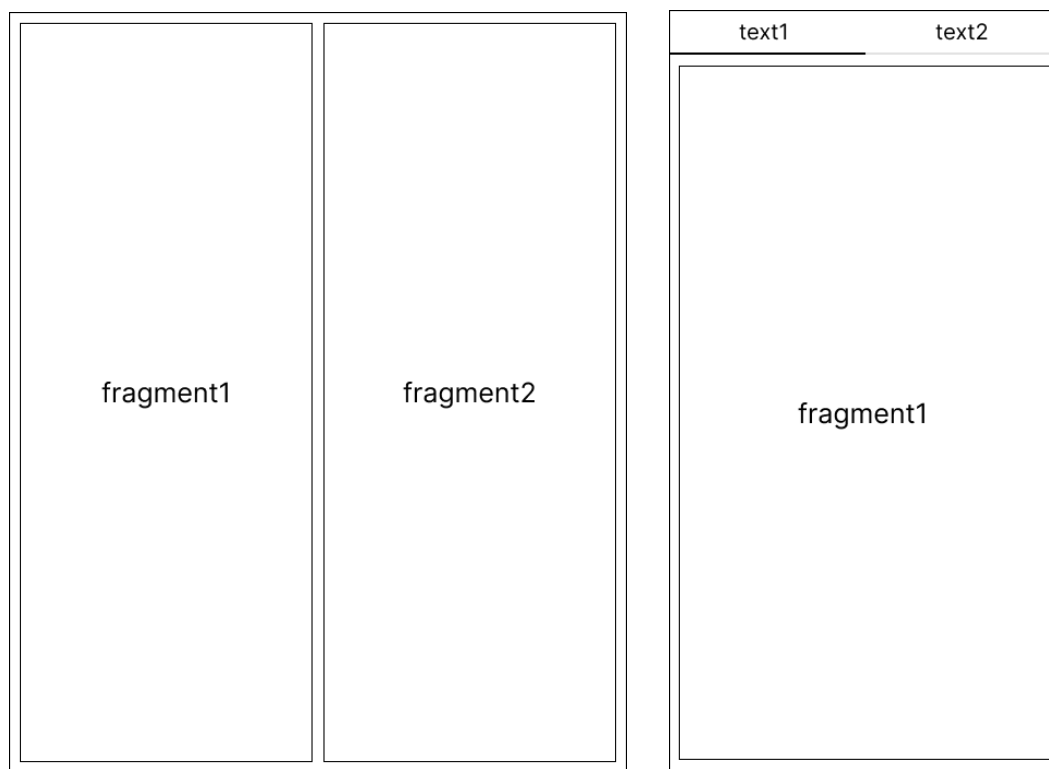
¹⁹více o fragmentech v sekci 3.3.2

²⁰viz <https://developer.android.com/reference/androidx/slidingpanelayout/widget/SlidingPaneLayout>

Kliknutím na některou z položek v levém panelu se zobrazí příslušný fragment v pravém panelu, který obsahuje podrobnější obsah ke zvolené položce.

Oběma těmito panely se nastaví požadovaná šířka a obsahy obou se natáhnou tak, aby vyplnily veškerý dostupný prostor. V případě, že obrazovka zařízení není dostatečně velká a nevejdou se oba panely vedle sebe, automaticky při kliknutí na položku v levém panelu dojde k překrytí pravým panelem, který je následně možné zavřít.

Prizpůsobení obrazovky detailu stolu (viz sekce 4.2.1) jsem dosáhl díky rozdělení celého layoutu do fragmentů. Tato obrazovka je rozdělena na dvě poloviny: levá obsahuje nabídku produktů, pravá účet na daném stole. Na tabletu, kde je obrazovka dostatečně široká, se oba tyto fragmenty zobrazují vedle sebe a každý zabírá polovinu obrazovky, viz [Obrázek 6a](#). Na telefonu by se vedle sebe ovšem nevešly – zobrazí se tedy pomocí záložek mezi kterými lze jednoduše přepínat, viz [Obrázek 6b](#).



(a) Zobrazení na tabletu

(b) Zobrazení na telefonu

Obrázek 6: Wireframe adaptivního layoutu

Ve zbytku uživatelského rozhraní stačilo upravit odsazení okrajů, velikosti jednotlivých prvků nebo velikost textu. Pro tento účel jsem tam, kde to bylo potřeba, nadefinoval jeden soubor se stylováním pro tablety a jeden pro telefony. Takto jsem nadefinoval hlavně soubor `styles.xml`, který obsahuje definice vzhledu hlavních prvků uživatelského rozhraní.

3.4.3 Vytváření tabulek na účtence

Jelikož každá tiskárna účtenek je jinak široká, má každá jiný počet znaků, které se vejdou na jeden řádek. Běžně tiskárny zvládnou 32-64 znaků na jednom řádku. Tento počet znaků tedy určuje dostupné místo při vytváření účtenky a je nastavitelný v nastavení aplikace.

Pro jednoduché vytváření tabulek, které se následně zobrazují v detailu účtenky (viz např. [Obrázek 9](#)) nebo jsou vtištěny přímo na účtenku (viz [Obrázek 19b](#)), jsem vytvořil třídu `TextTable`, která to velmi jednoduše umožňuje.

Pro vytvoření textové tabulky stačí třídě předat názvy sloupců, pole dat, kterými se má tabulky naplnit a pole anonymních funkcí, pomocí kterých se získá hodnota příslušného sloupce. Sloupců a anonymních funkcí musí být stejný počet. Lze samozřejmě nastavit maximální délku řádku. Třída také umožňuje nastavit:

- Zarovnání hodnot v jednotlivých sloupcích spolu s výchozím zarovnáním, které se použije, není-li zadáno jiné (např. na [Obrázku 9](#) mají první dva sloupce nastaveno zarovnání doleva a pro zbylé je nastaveno výchozí zarovnání doprava). Příklad nastavení zarovnání lze vidět ve zdrojovém kódu [2](#).
- Zalomení zobrazování hodnot po určitém sloupci (například na [Obrázku 9](#) je nastaveno zalomení po druhém sloupci). Pokud je zalomení nastaveno, názvy sloupců se rovnoměrně rozmístí na řádku v dostupné šířce, pokud není, všechny hodnoty se dají do jednoho řádku a šířka jednotlivých sloupců je daná délkou nejdelší hodnoty, kterou daný sloupec obsahuje.
- Zobrazování poznámek v tabulce. Pokud je tato možnost povolena, do tabulky se přidávají také případné poznámky, které byly k produktům při objednávce napsané. Tyto poznámky se zobrazují na samostatném řádku hned pod řádkem s informacemi o příslušném produktu.

```
1  val columns = mutableListOf("Ks", "Položka", "Kč/ks", "Kč")
2  val columnsAlignment = listOf(
3    Pair(0, TextTable.Alignment.LEFT), //první sloupec doleva
4    Pair(1, TextTable.Alignment.LEFT) //druhý sloupec doleva
5  )
6  val getters = mutableListOf(
7    { arg: Any -> "${(arg as OrderItem).amount}x" },
8    { arg: Any -> (arg as OrderItem).name },
9    { arg: Any -> (arg as OrderItem).price },
10   { arg: Any -> (arg as OrderItem).price * arg.amount }
11  )
```

Zdrojový kód 2: Ukázka definice sloupců a příslušných funkcí pro získání hodnot daných sloupců.

4 Uživatelská dokumentace

Pokladní systém Brewify lze nainstalovat pomocí přiloženého souboru (viz příloha B) a následováním instrukcí popsanych v souboru README.txt.

4.1 Spuštění

Při spuštění aplikace se zobrazí přihlašovací obrazovka. Pokud se při předešlém ukončení aplikace uživatel neodhlásil, je přihlášen automaticky. V případě, že je aplikace spuštěna poprvé po instalaci, přihlásit se lze výchozím účtem s oprávněním Admin, jehož uživatelské jméno je Admin a heslo kava. Heslo lze poté změnit v nastavení uživatelů (sekce 4.5.6).

4.2 Kavárna

Kavárna je výchozí obrazovkou, která se zobrazí po spuštění aplikace. Jsou zde vykresleny stoly a jednotlivé místnosti, které byly přidány v nastavení aplikace – sekce 4.5.1. Následující sekce se bude odkazovat na [Obrázek 7](#).

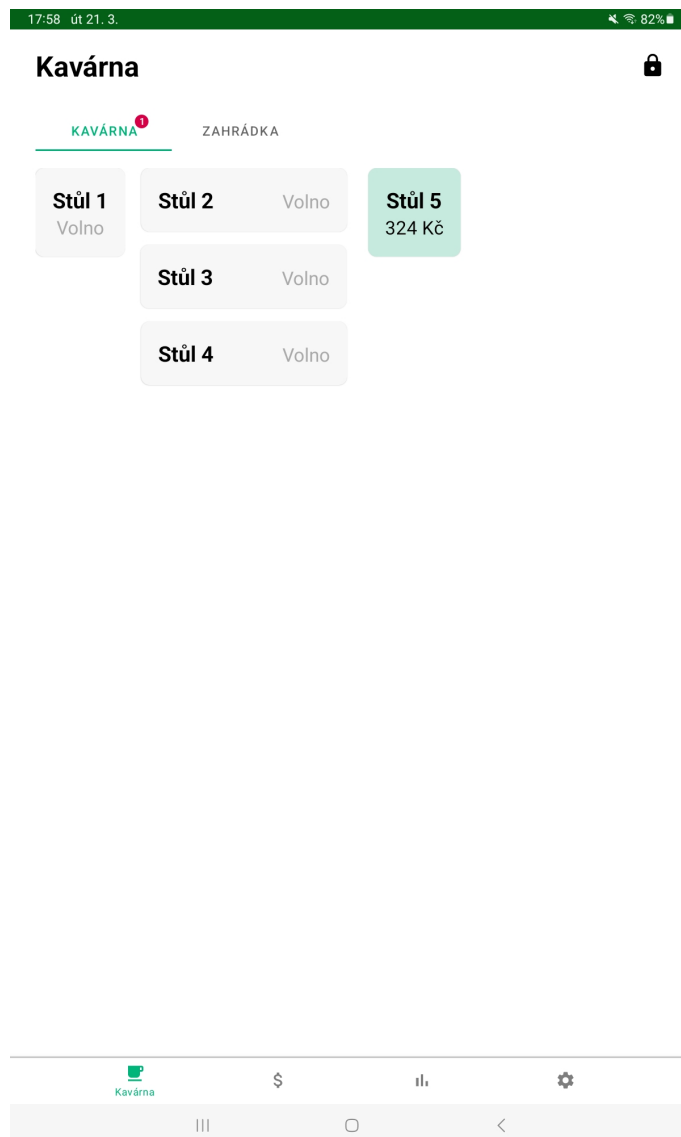
V pravém horním rohu obrazovky se nachází ikonka zámku. Ta slouží k zapnutí/vypnutí režimu pro přemísťování stolů v rámci aktivní místnosti. Pokud je zobrazena ikonka odemknutého zámku, lze stoly přemísťovat, ale nelze žádný otevřít. Stůl stačí pouze prstem přetáhnout na libovolné místo na obrazovce a pustit jej. Po dokončení přesouvání je zase třeba ukončit režim pro přemísťování stolů – zobrazena bude tedy ikonka zamknutého zámku.

Jednotlivé místnosti jsou reprezentovány záložkami, kdy každá z nich obsahuje název dané místnosti. Zobrazují se pouze stoly patřící do aktivní místnosti a ty lze přepínat klikáním na jednotlivé záložky. Nachází-li se v místnostech stoly, na kterých je otevřený účet, je počet takových stolů zobrazen v `badgi`²¹ u příslušné záložky.

Stoly jsou zobrazeny ve zbylém prostoru na obrazovce pod sekci se záložkami místností. U každého stolu je vidět jeho název a celková cena účtu na daném stole, pokud na něm nějaký existuje. Stoly, na nichž existuje účet, jsou navíc obarveny rozdílnou barvou a je na nich zobrazena celková částka objednávky, aby byly jednoduše vizuálně rozlišitelné. Plocha pro vykreslování stolů zabírá většinu obrazovky zařízení, čímž je zajištěn dostatek prostoru pro rozmístění stolů po obrazovce tak, aby náhled co nejvíce odpovídal reálnému rozložení kavárny. Díky tomu se uživatel, obsluha v kavárně, může jednodušeji zorientovat při zakládání účtu na libovolném stole. Kliknutím na stůl se zobrazí detail daného stolu, respektive účtu patřícího na daný stůl, kde jej lze vytvořit či upravovat.

Je-li aplikace spuštěna na telefonu, stoly jsou zobrazeny v lineárním seznamu pod sebou a možnost libovolného umístování není povolena.

²¹<https://m3.material.io/components/badges/overview>



Obrázek 7: Obrazovka Kavárna

4.2.1 Detail stolu

Tato obrazovka slouží k práci s účtem na příslušném stole. Obrazovka je rozdělena na dvě části, viz [Obrázek 8](#).

V horní části levé poloviny je pole pro vyhledávání mezi produkty podle jejich názvu. Pod ním jsou zobrazeny `chipy`²² reprezentující jednotlivé skupiny produktů²³. Kliknutím na některý z `chipů` lze zobrazit pouze produkty patřící

²²<https://m3.material.io/components/chips/overview>

²³Jednotlivé produkty jsou přidávány do logických skupin, jako jsou např. Kávy, Dorty, apod. Viz [Nastavení produktů](#).

do dané skupiny. Opětovným kliknutím se tento filtr zruší a zobrazí se opět všechny produkty. Ve zbylém prostoru, pod sekci se seznamem skupin produktů, jsou již zobrazeny jednotlivé produkty spolu s nadpisem své skupiny. Pokud je v nastavení aplikace navíc povolena možnost [Zobrazit denní nabídku](#), jsou produkty do ní patřící zobrazeny jako první v samostatné skupině. Navíc jsou označeny zeleným pruhem na levé straně pro lepší vizuální rozlišení.

Produkt lze na účet přidat kliknutím na něj. Pokud je potřeba k produktu připsat poznámku pro obsluhu²⁴, stačí ho přidržet a po zobrazení dialogového okna zadat poznámku a potvrdit. Tato poznámka se následně vytiskne na objednávkový bon, je-li jejich tisk povolen v nastavení.

Pravá polovina obrazovky obsahuje seznam produktů přidanych na účet a ve spodní části celkovou cenu účtu a tři tlačítka, kterými s ním lze provést akce. U přidanych produktů je kromě názvu zobrazen také počet, kolikrát byly objednány. V pravém spodním rohu karty přidaneho produktu je zobrazena celková cena za daný produkt v daném množství.

Množství lze zvýšit kliknutím jak na kartu přidaneho produktu, tak na kartu stejneho produktu v nabídce nalevo. Snížit ho lze pouze kliknutím na tlačítko „Odebrat“. Pokud navíc byl účet již uložen, a tím případně vytištěn objednávkový bon, tak je při snížení množství pod původní hranici vyžadována poznámka, proč k odebrání produktu dochází a poté se stornuje. Stornovaný produkt je přidán na konec objednávky, je obarven červeně a nelze jej upravovat. Jakmile je produkt stornován, nelze tuto akci vrátit. Zabráni se tak případnému záměrnému odstraňování produktů před placením a ztrátě peněz. Poznámku k produktu lze přidat nebo upravit přidržením dané karty produktu.

Pod seznamem přidanych produktů a celkovou cenou účtu se nachází tři tlačítka:

- Zaplatit
- Přesunout
- Zavřít/Uložit - text tlačítka se měni podle toho, jestli byla provedena nějaká změna nebo ne

Tlačítka Zaplatit a Přesunout po stisknutí aktivují režim výběru produktů, kde se vybírá, se kterými produkty (a jakým množstvím) se má zvolená akce provést. Tento režim je znázorněn na počátku tak, že všechny přidane produkty jsou obarveny šedě, zobrazuje se navíc celková cena výběru, tlačítko zvolené akce se zobrazuje jako první a prostředním tlačítkem je možné zvolenou akci zrušit, viz [Obrázek 11a](#).

Pokud nebyl přidán žádný produkt do výběru, provede se akce s celým účtem, tedy se všemi produkty. Text tlačítka zvolené akce se měni podle toho, má-li se akce provést se všemi produkty či pouze s jejich výběrem.

²⁴Např. že cappuccino má být z alternativního mléka

Kliknutím na produkt v tomto režimu se produkt přidá do výběru. Kliknutím na tlačítko „Odebrat“ se z výběru odebere. Produkty přidané do výběru již nejsou obarveny šedě a v levém dolním rohu karty produktu je zobrazeno množství, které bylo přidáno do výběru, viz [Obrázek 11b](#).

Stisknutím tlačítka Zavřít/Uložit dojde k zavření náhledu detailu stolu. V případě, že na účtu byly provedeny změny, dojde k uložení nového stavu. Pokud je navíc v nastavení povolen tisk objednávkových bonů, je spuštěn tisk. O průběhu tisku je uživatel informován zobrazením stavu na tlačítku. Po vytisknutí objednávkového bonu se informace o úspěšnosti tisku zobrazí na tlačítku a náhled stolu se automaticky zavře.

4.2.2 Placení

Tlačítko Zaplatit po stisknutí aktivuje režim výběru produktů popsany v předchozí kapitole.

Po vybrání produktů, které se mají zaplatit, se potvrzujícím stisknutím tlačítka zobrazí okno s údaji pro dokončení platby, viz [Obrázek 9](#). V okně je zobrazený název stolu, datum a čas provedení platby, souhrn vybraných produktů k placení a celková cena. Pod celkovou cenou lze zvolit, jestli se má účtenka vytisknout, či nikoli. Ve spodní části okna se pak nachází dvě tlačítka, kterými se vybírá druh placení – hotově či kartou.

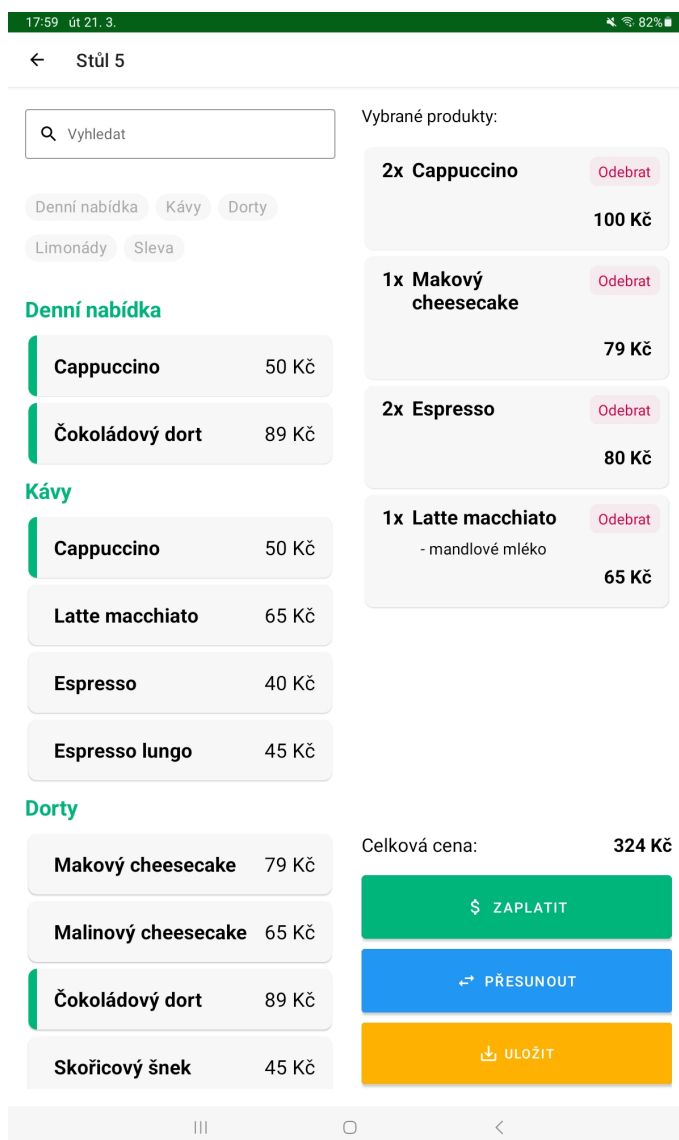
Po vybrání druhu platby a stisku příslušného tlačítka se buď účtenka pouze uloží a okno se zavře, nebo se spustí tisk a uživatel je o průběhu tisku informován zobrazením stavu na tlačítku zvoleného druhu platby. Jakmile byla účtenka úspěšně vytisknuta, tlačítko se obarví zelenou barvou, v opačném případě červenou. Opětovným kliknutím na tlačítko se okno zavře a pokud nebyly zaplacený všechny produkty, zobrazí se zpět náhled stolu. V opačném případě se zavře i náhled stolu a zobrazí se tak obrazovka Kavárna. Zobrazení stavu průběhu tisku a indikaci neúspěšného tisku účtenky lze vidět na [Obrázku 10](#).



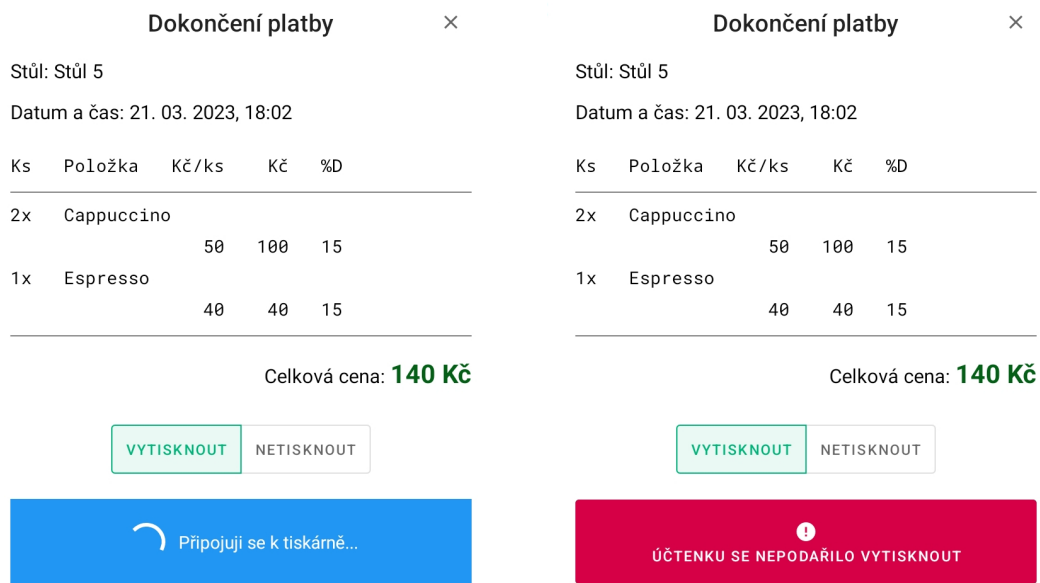
Obrázek 9: Okno s údaji pro dokončení platby

4.2.3 Přesun

Tlačítko Přesunout taktéž aktivuje režim výběru produktů popsany v kapitole [4.2.1](#). Při potvrzujícím kliknutí se zavře náhled detailu stolu a zobrazí se stránka [Kavárna](#) s tím rozdílem, že nadpis se změnil na „Kam přesunout?“ a stůl, ze kterého se mají produkty přesunout, je obarven modře. Produkty lze přesunout



Obrázek 8: Detail účtu na stole



(a) Zobrazení průběhu tisku účtenky

(b) Neúspěšný tisk účtenky

Obrázek 10: Zobrazení informování uživatele o průběhu tisku účtenky

na libovolný stůl v libovolné místnosti. Pokud na cílovém stole již existuje účet, objeví se potvrzovací okno, zda-li se mají účty sloučit na cílový stůl. Přesun je možné zrušit červeným tlačítkem v pravém spodním rohu obrazovky.

4.2.4 Sleva

Zadat slevu na celý účet je možné kliknutím na chip „Slevy“ a následně kliknutím na kartu „Sleva“. Tím se zobrazí dialogové okno.

V horní části tohoto okna jsou pod nadpisem „Předvolby“ tři tlačítka: 5 %, 10 % a 15 %. Tato tlačítka slouží jako rychlá možnost pro přidání nejpoužívanějších slev. Sleva se tedy přidá pouhým kliknutím na jedno z těchto tlačítek.

Pokud je potřeba zadat jinou slevu, při zobrazení tohoto okna se automaticky otevře klávesnice a označí se vstup pro hodnotu slevy. Stačí tedy na klávesnici napsat hodnotu slevy a poté jedním z tlačítek, které se nachází ve spodní části okna zvolit, jestli se jedná o slevu v korunách či procentech.

Je-li přidána sleva procentuální, při změně obsahu objednávky se sleva automaticky přepočítá. Počítá se tedy vždy z celkové ceny objednávky. Slevu lze kdykoli z objednávky odebrat a opět přidat. Při odebrání slevy není vyžadována poznámka, jako tomu je při odebrání produktů.

Vybrané produkty:

2x Cappuccino	Odebrat	100 Kč
1x Makový cheesecake	Odebrat	79 Kč
2x Espresso	Odebrat	80 Kč
1x Latte macchiato - mandlové mléko	Odebrat	65 Kč

Cena výběru: **0 Kč**

Celková cena: **324 Kč**

\$ ZAPLATIT VŠE

✕ ZRUŠIT PLATBU

↓ ULOŽIT

Vybrané produkty:

2x Cappuccino	Odebrat	100 Kč
2x (100 Kč)		
1x Makový cheesecake	Odebrat	79 Kč
2x Espresso	Odebrat	80 Kč
1x (40 Kč)		
1x Latte macchiato - mandlové mléko	Odebrat	65 Kč

Cena výběru: **140 Kč**

Celková cena: **324 Kč**

\$ ZAPLATIT VÝBĚR

✕ ZRUŠIT PLATBU

↓ ULOŽIT

(a) Začátek výběru produktů k placení

(b) Vybrané produkty k placení

Obrázek 11: Detail stolu – výběr produktů

4.3 Tržba směny

Tato stránka (viz [Obrázek 13](#)) slouží k rychlému přehledu o aktuální směně. Kromě seznamu jednotlivých účtenek, a volitelně pohybů hotovosti, je zde vidět tržba směny, počet účtenek a stav pokladny.

V pravém horním rohu se nachází dvě ikony, jejichž význam a funkcionality jsou popsány v sekci [4.3.1](#).

Pod těmito ikonkami jsou zobrazeny 3 zelené karty - přehled tržby aktuální směny. Význam jednotlivých karet postupně zleva (viz [Obrázek 13](#)):

- Tržba – jedná se o součet částek všech účtenek vytvořených v aktuální směně.
- Počet účtenek – celkový počet účtenek vytvořených za aktuální směnu k okamžiku zobrazení této obrazovky.
- Hotovost – jedná se o součet všech zadaných pohybů hotovosti (viz sekce 4.3.1) a účtenek, které byly zaplacený hotovostí. Částka zobrazená v této kartě by měla vždy odpovídat součtu peněz ve fyzické pokladně.

Pod přehledem tržby je zobrazen seznam jednotlivých účtenek. Na kartě účtenky jsou zobrazeny pouze nejdůležitější údaje jako je číslo účtenky, čas zaplacení, stůl, na kterém byla účtenka vytvořena nebo celková částka. Na každou účtenku lze kliknout a tím zobrazit okno se všemi údaji. V tomto okně jsou zobrazeny veškeré produkty, jejich sazba DPH včetně vysvětlující tabulky – je-li nastaveno, že je provozovatel daného podniku plátcem DPH. Ve spodní části tohoto okna se nachází tlačítko, kterým lze účtenku znovu vytisknout.

Mezi účtenkami se mohou zobrazovat pohyby hotovosti, které lze také rozkliknout, a tím zobrazit detail, ale nelze jej nijak upravovat ani odstranit.

4.3.1 Pohyb hotovosti

Tato funkcionalita umožňuje sledovat stav pokladny – lze tedy jednoduše odhalit jakékoli nesrovnalosti, co se hotovosti v pokladně týče.

V pravém horním rohu obrazovky se nachází dvě ikonky. Ikonka oka slouží k zobrazení/skrytí karet pohybů hotovosti v seznamu účtenek směny. Druhá ikonka, na které je zobrazen znak +/-, slouží k přidání pohybu hotovosti.

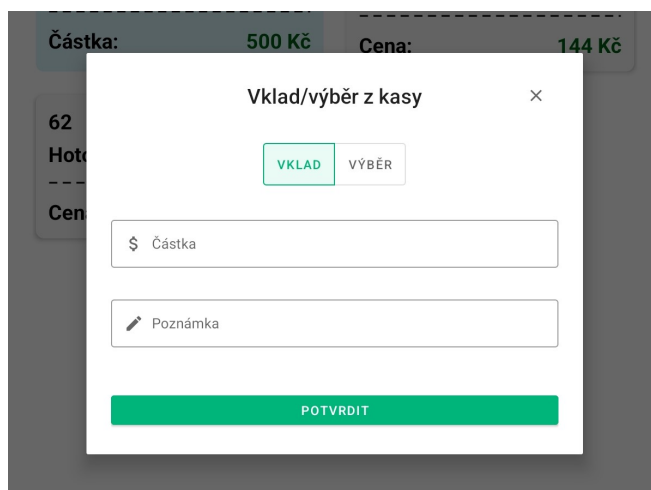
Po kliknutí na ikonku pro přidání pohybu hotovosti se zobrazí dialogové okno, viz [Obrázek 12](#). V tomto okně lze zvolit, zda se jedná o vklad či výběr, zadat požadovanou částku a volitelně poznámku, proč k dané akci došlo²⁵. Pro přidání pohybu hotovosti je vyžadováno oprávnění Vedoucí směny nebo vyšší.

Pohyby hotovosti jsou v seznamu účtenek zobrazeny, stejně jako účtenky, v podobě karty. Karta je nadepsána textem „Pohyb hotovosti“, pod tímto nadpisem je zobrazena případná poznámka a pod poznámkou zadaná částka. Aby bylo na první pohled jasné, jestli se jedná o vklad či výběr, jsou karty vkladu obarveny modře a karty výběru červeně.

4.3.2 Uzavření směny

Směnu lze uzavřít tlačítkem ve spodní části obrazovky. Pro uzavření směny je vyžadováno oprávnění Vedoucí směny nebo vyšší. Pokud aktuálně přihlášený uživatel nemá dostatečnou úroveň oprávnění, je po kliknutí na tlačítko nabídnuta možnost pro jednorázové přihlášení, které nastaví aktuální úroveň oprávnění

²⁵Např. na začátku směny vloží vedoucí směny 1 000 Kč v drobných, aby mohla obsluha vracet



Obrázek 12: Okno pro zadání vkladu/výběru hotovosti z kasy

pouze pro tuto jednu akci. Uživatelé se tak nemusí přehlašovat, aby bylo směnu možné zavřít.

Uzavřením směny dojde k vytvoření záznamu o dané směně, vynuluje se zobrazení tržby směny, počtu účtenek i jejich seznamu. Stav kasy zůstane stejný. Jinak řečeno, následující směna začíná se stavem kasy, se kterým předchozí směna skončila. Záznamy o směnách se všemi informacemi je možné zobrazit v přehledu směn (sekce 4.4.3).

4.4 Přehled

Sekce Přehled se skládá ze tří obrazovek. Poskytuje uživateli možnost zobrazit si statistiky za vybrané časové období, vyhledávat v celkové historii účtenek a uzavřených směn.

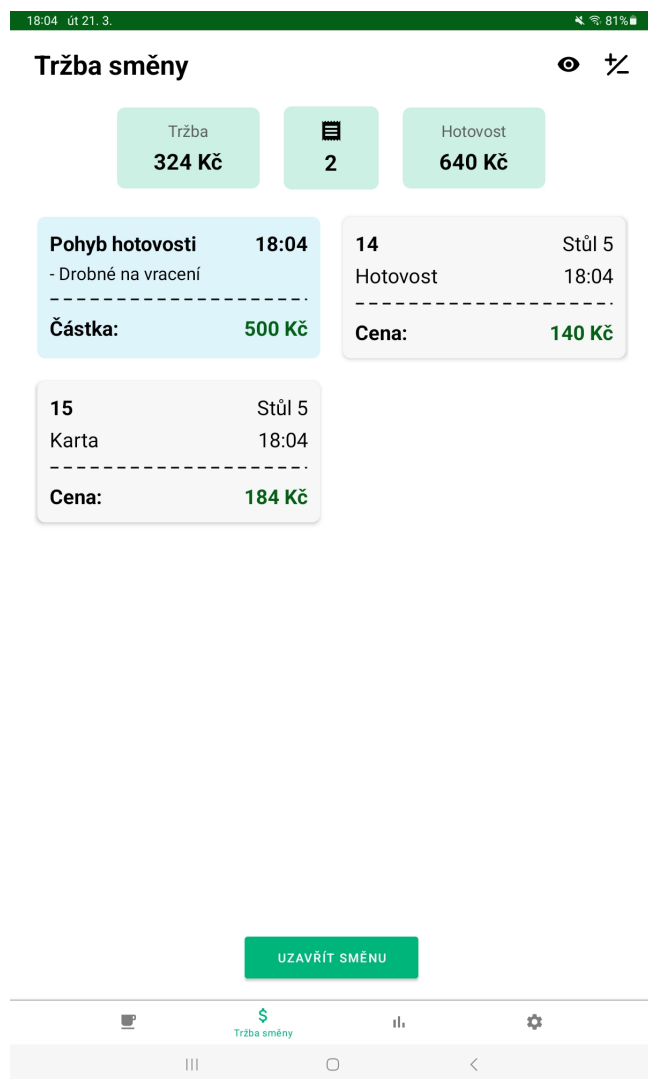
4.4.1 Statistiky

Tato obrazovka slouží pro zobrazení statistik prodeje za vybrané časové období. Statistiky jsou zobrazeny pomocí grafů a souhrnů jednotlivých dní – viz [Obrázek 14](#).

V horní části této obrazovky se nachází pole pro výběr časového období. Při spuštění je automaticky jako výchozí období nastaveno aktuální datum, zobrazují se tedy statistiky pouze za právě probíhající den. Období lze změnit tlačítkem „Vybrat období“. Kliknutím na něj se zobrazí dialogové okno pro výběr požadovaného rozmezí. Po potvrzení se aktualizují zobrazené statistiky.

Pod polem pro výběr časového rozmezí jsou zobrazeny dva grafy:

- První (koláčový) graf – zobrazuje statistiky prodeje nejprodávanějších produktů. Nabídkou pod legendou grafu lze změnit, kolik nejprodávanějších produktů se má zobrazovat. Výchozí, a zároveň minimální, hodnota je 5, maximální 15.



Obrázek 13: Obrazovka Tržba směny

- Druhý (sloupcový) graf – vykresluje hodinové vytížení kavárny na základě počtu vytvořených účtenek. Hodinové rozmezí grafu je dáno nastavením otevírací doby kavárny, viz nastavení sekce 4.5.7.

Pod sekci s grafy se nachází seznam jednotlivých dní za vybrané časové období. Zobrazují se souhrny pouze těch dní, ve kterých byla vytvořena nějaká účtenka. Na kartě souhrnu dne je zobrazeno datum, celková tržba (součet částek účtenek) a počet účtenek. Kliknutím na libovolný souhrn dne dojde k nastavení filtru v přehledu Historie na datum daného dne a přepnutí obrazovky na Historii. Zobrazeny jsou tedy pouze účtenky vytvořené v daný den.



Obrázek 14: Přehled – statistiky za vybrané období

4.4.2 Historie

Na této obrazovce lze procházet kompletní historii účtenek. Při prvním zobrazení se účtenky zobrazují postupně od nejstarší. Pro pohodlnější hledání mezi účtenkami je k dispozici několik filtrů:

- Datum – umožňuje filtrovat účtenky na základě data jejich vytvoření; možnosti filtru jsou:
 - Poslední měsíc
 - Aktuální měsíc
 - Posledních 12 měsíců

- Jiné – libovolné rozmezí, které si uživatel nastaví
- Čas – uživatel volí počáteční a/nebo koncový čas, podle kterého se mají účtenky filtrovat
- Cena – cenové rozmezí účtenek, případně účtenky s konkrétní částkou
- Číslo účtenky – rozmezí nebo jedno číslo účtenky
- Hotovost – zobrazí pouze účtenky, které byly placeny v hotovosti
- Karta – zobrazí pouze účtenky, které byly placeny kartou
- Pohyby hotovosti – umožňuje spolu s účtenkami zobrazit pohyby hotovosti²⁶

Veškeré nastavené filtry lze zrušit tlačítkem, které se nachází za posledním filtrem – Pohyby hotovosti. Automaticky se tak opět zobrazí celková historie účtenek od nejstarší. Filtry lze navíc libovolně kombinovat a jejich aktuální nastavení je zobrazeno v příslušném chipu²⁷ daného filtru.

Jakoukoli účtenku lze opět rozkliknout a zobrazit tím její detail, který vypadá téměř totožně jako [Obrázek 9](#), pouze s tím rozdílem, že navíc zobrazuje pořadové číslo účtenky, typ platby a tabulku s DPH, je-li provozovatel kavárny plátcem DPH, viz nastavení sekce [4.5.9](#). Účtenku je také možné vytisknout. Na takto vytištěné účtence je navíc informace, že se jedná o kopii dokladu.

4.4.3 Směny

Zde je přehled veškerých uzavřených směn. Zobrazovány jsou opět od nejstarší a lze mezi nimi filtrovat pomocí filtrů. Oproti historii účtenek (sekce [4.4.2](#)) jsou zde k dispozici pouze tři filtry, a to:

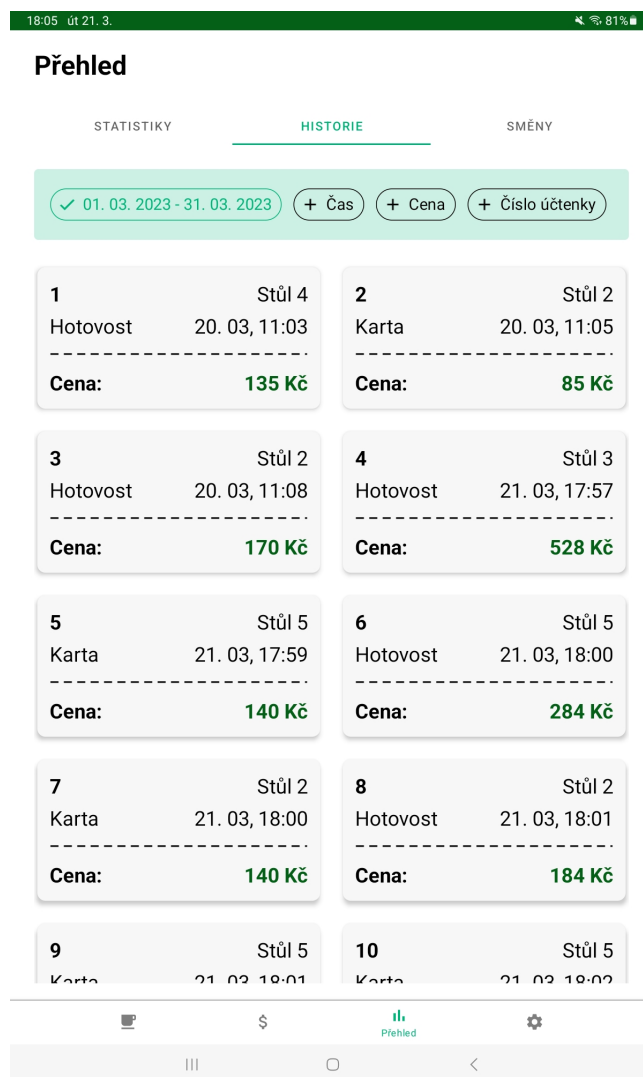
- Datum
- Čas
- Částka

Jejich použití i funkcionalita jsou totožné s filtry v historii účtenek.

Informace o směnách jsou zobrazeny v kartách, kde každá z nich obsahuje datum a čas uzavření dané směny, částku, se kterou daná směna skončila a jméno uživatele, který směnu uzavřel. Účtenky, které byly vytvořeny v rámci dané směny, lze zobrazit kliknutím na libovolnou kartu směny – dojde tak k nastavení filtru na čísla účtenek v historii a přepnutí na obrazovku historie.

²⁶viz sekce [4.3.1](#)

²⁷Na [obrázku 15](#) je nastavený filtr Datum na „Aktuální měsíc“.



Obrázek 15: Přehled – Historie s nastaveným filtrem Datum na „Aktuální měsíc“

4.5 Nastavení

Tato obrazovka je rozdělena na dvě části, viz [Obrázek 17](#). V horní části levé třetiny obrazovky je zobrazeno jméno aktuálně přihlášeného uživatele. Pod tímto údajem se nachází seznam jednotlivých sekcí nastavení. Tento seznam je rozdělen na dvě logické části:

1. Provoz – obsahuje nastavení potřebné pro chod kavárny. Na nastavování v této sekci stačí oprávnění 'Vedoucí směny'.
2. Ostatní – toto nastavení se týká podniku jako celku a nemění se v běžném provozu. Potřeba je tedy oprávnění 'Admin'.

Ve zbylých dvou třetinách obrazovky se zobrazuje aktuálně zvolená sekce nastavení. Při prvním spuštění je zobrazena sekce nastavení stůlů. Pokud nemá

Přehled

STATISTIKY	HISTORIE	SMĚNY
<div style="display: flex; justify-content: space-between; align-items: center;"> + Datum + Čas + Částka zrušit filtry </div>		
<div style="display: flex; justify-content: space-between; align-items: center; padding: 10px;"> <div style="display: flex; align-items: center;"> 📅 <div style="margin-left: 5px;">24. 02. 2023, 11:14</div> </div> <div style="margin-left: 20px;">\$ 1257 Kč</div> <div style="margin-left: 20px;"> 👤 <div style="margin-left: 5px;">Admin</div> </div> <div style="margin-left: 20px;"> <small>DETAILY ></small> </div> </div>		
<div style="display: flex; justify-content: space-between; align-items: center; padding: 10px;"> <div style="display: flex; align-items: center;"> 📅 <div style="margin-left: 5px;">24. 02. 2023, 17:57</div> </div> <div style="margin-left: 20px;">\$ 1387 Kč</div> <div style="margin-left: 20px;"> 👤 <div style="margin-left: 5px;">Admin</div> </div> <div style="margin-left: 20px;"> <small>DETAILY ></small> </div> </div>		
<div style="display: flex; justify-content: space-between; align-items: center; padding: 10px;"> <div style="display: flex; align-items: center;"> 📅 <div style="margin-left: 5px;">02. 03. 2023, 13:03</div> </div> <div style="margin-left: 20px;">\$ 500 Kč</div> <div style="margin-left: 20px;"> 👤 <div style="margin-left: 5px;">Admin</div> </div> <div style="margin-left: 20px;"> <small>DETAILY ></small> </div> </div>		

Obrázek 16: Přehled – Směny

přihlášený uživatel dostatečné oprávnění pro provedení požadované změny, je o tom informován a provedení změny mu není umožněno.

4.5.1 Stoly

Následující sekce se bude odkazovat na pravou část [Obrázku 17](#). V této sekci se nastavují jak místnosti, tak stoly do nich patřící. Jednotlivé místnosti jsou zobrazeny v horní části obrazovky v podobě chipů. Kliknutím na chip místnosti se místnost označí a zobrazí se příslušné stoly. Při spuštění je automaticky označena první přidaná místnost.

Stoly jsou zobrazeny ve vertikálním seznamu pod seznamem místností. Na kartě každého stolu je zobrazený jeho název a cena objednávky na daném stole. Pokud na něm objednávka neexistuje, karta stolu je obarvena šedě, v opačném případě je podbarvena zeleně.

Přidat novou místnost lze kliknutím na chip „Přidat místnost“, stůl kliknutím na tlačítko „Přidat stůl“. V obou případech se po kliknutí zobrazí dialogové okno, do kterého uživatel zadá název místnosti/stolu. Pokud je přidáván stůl, je k dispozici navíc sekce „Zobrazení“ stolu. Možnosti zobrazení jsou:

- Horizontální – informace na kartě stolu na obrazovce Kavárna (viz sekce [4.2](#)) budou na jednom řádku²⁸. Šířka této karty je určena dodatečným nastavením stolů, které je popsáno dále.

²⁸např. „Stůl 2“ nebo „Stůl 3“ na [Obrázku 7](#)

- Vertikální – informace na kartě stolu jsou zobrazeny ve sloupci²⁹. Šířka takto zobrazených stolů je určena délkou názvu stolu.

Kliknutím na kartu stolu se zobrazí dialogové okno pro upravení stolu. Při držení karty se zobrazí možnost stůl odstranit nebo upravit. Pokud na stole existuje objednávka, možnost odstranění stolu je vypnutá.

Místnost je taktéž možné odstranit či upravit přidržetím `chipu` dané místnosti. Zobrazí se stejná nabídka jako při přidržení karty stolu. Odstraněním místnosti dojde i k odstranění všech stolů, které do ní patří.

Pod seznamem stolů je sekce „Další nastavení“. Po rozkliknutí této sekce se zobrazí nastavení šířky stolů s horizontálním zobrazením. Toto nastavení se vztahuje ke všem stolům s horizontálním zobrazením ve všech místnostech. Na výběr jsou dvě možnosti, tedy šířka karty stolu bude určena:

1. Podle šířky textu
2. Procenta obrazovky – při zvolení této možnosti je nutné ještě zadat, kolik procent šířky obrazovky mají stoly s horizontálním zobrazením zabírat. Výchozí zvolená hodnota je 33 %.

4.5.2 Produkty

V této části nastavení se definují skupiny produktů a jednotlivé produkty do nich patřící. Stejně jako jsou v nastavení stolů zobrazeny jednotlivé místnosti (sekce 4.5.1), tak i zde jsou skupiny produktů zobrazeny v horní části v podobě `chipů`, viz [Obrázek 18](#). Jejich chování a ovládání je také stejné jako u nastavení místností – kliknutím na `chip` skupiny produktů se skupina označí a zobrazí se produkty patřící do dané skupiny.

Při spuštění se automaticky označí první přidaná skupina. Pokud je v nastavení Denní nabídky (sekce 4.5.4) povoleno zobrazovat denní nabídku a zároveň ji zobrazovat jako samostatnou skupinu, je tato skupina první. Pokud není povoleno zobrazovat ji jako samostatnou skupinu, jsou produkty patřící do denní nabídky označeny jednotlivě zeleným pruhem v levé části karty produktu,³⁰

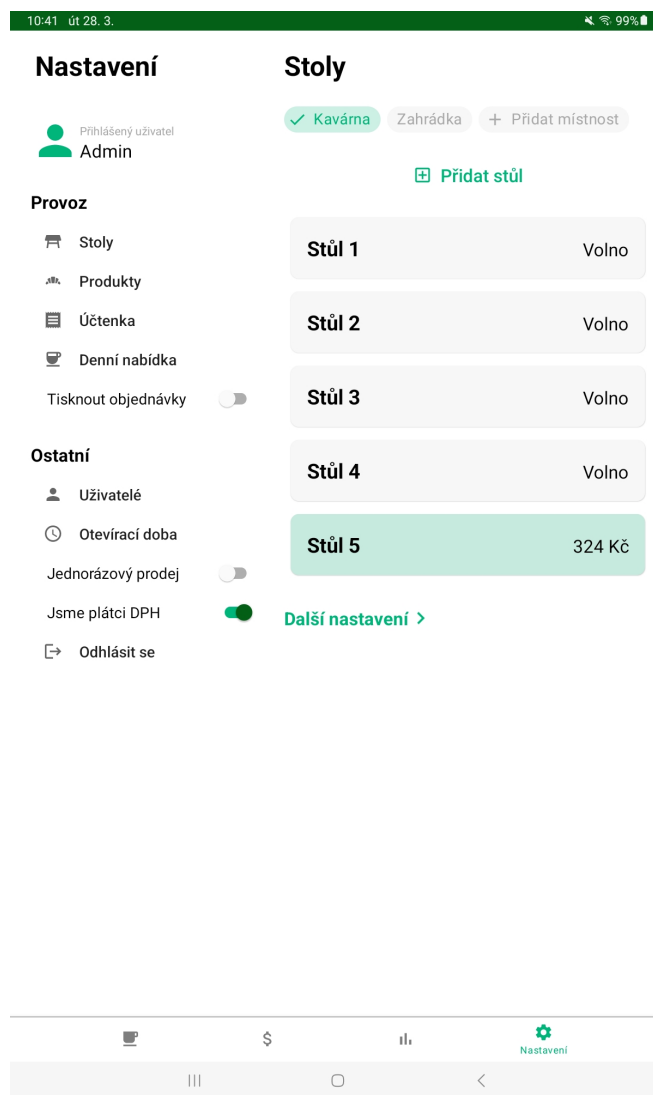
Pod seznamem skupin produktů je tlačítko „Přidat produkt“, kterým lze nadefinovat nový produkt patřící do aktuálně zvolené skupiny. Jestliže je zvolena skupina „Denní nabídka“, toto tlačítko není viditelné, protože do denní nabídky se produkty vybírají jednotlivě už z existujících produktů, viz sekce 4.5.4.

Následuje seznam produktů. Jednotlivé produkty jsou zobrazeny v podobě karet. Na každé kartě je zobrazen název produktu, jeho cena, kód a sazba DPH³¹, jak je vidět na [Obrázku 18](#).

²⁹např. „Stůl 1“ nebo „Stůl 5“ na [Obrázku 7](#)

³⁰viz například [Obrázek 18](#), produkt Cappuccino. Detailnější popis najdete v sekci 4.5.4

³¹Pokud je nastaveno, že provozovatel podniku je plátcem DPH, viz 4.5.9. Jinak se DPH nezobrazuje.



Obrázek 17: Nastavení

Každá skupina produktů má svůj jedinečný kód, kterým je vždy číslo v celých stovkách. Kód je automaticky generován při vytváření, např. tedy první přidaná skupina bude mít kód 200, druhá 300, apod.³². Každý produkt má také svůj kód, jehož generování je založeno na stejném principu. Řád stovek u kódu produktu je dán řádem stovek kódu skupiny, do které patří. To znamená, že produkty, které patří do skupiny s kódem např. 200, budou mít postupně kódy 201, 202, atd. Po vytvoření tyto kódy nelze upravovat, lze je ale změnit při definování nové skupiny/produktu. Je to proto, že skupiny i produkty se poté zobrazují v pořadí, které je dané právě kódem skupiny/produktu. Kdyby tedy došlo k odstranění produktu s kódem např. 202 a produkty s kódy 201 a 203 by existovaly, další kód, který by byl automaticky vygenerován při definování nového produktu, by

³²kód 100 je vyhrazen pro skupinu „Denní nabídka“

byl 204. Tento kód lze tedy změnit na 202, aby se nový produkt zobrazoval mezi již existujícími produkty, nikoli až za nimi. Stejně tak se dá ovlivňovat pořadí skupin produktů.

Upravování skupin a produktů je opět stejné jako tomu bylo s místnostmi a stoly (sekce 4.5.1). Přidržením skupiny produktů nebo samostatného produktu se zobrazí nabídka s možnostmi upravit nebo odstranit danou skupinu či produkt. Produkt lze navíc upravit kliknutím na kartu produktu.

Produkty

Denní nabídka **✓ Kávy** Dorty Limonády

+ Přidat skupinu

+ Přidat produkt

Cappuccino		
Cena	Kód	DPH
52 Kč	201	15 %

Latte macchiato		
Cena	Kód	DPH
65 Kč	202	15 %

Espresso		
Cena	Kód	DPH
40 Kč	203	15 %

Espresso lungo		
Cena	Kód	DPH
45 Kč	204	15 %

Obrázek 18: Nastavení – produkty

4.5.3 Účtenka

Aplikace umožňuje nastavení záhlaví a zápatí účtenky. Tím je konkrétně myšleno nastavení úvodního textu na účtence, viz část „Záhlaví účtenky“ na [Obrázku 19a](#) a úvodní řádky na [Obrázku 19b](#), a závěrečného textu, viz část „Zápatí účtenky“ a poslední řádek na zmíněných obrázcích. Vzhled zbývajících obsahu účtenky nelze nastavit ani ovlivnit.

Do záhlaví i zápatí se text přidává po řádcích. U každého řádku lze nastavit

zarovnání textu, a to: doleva, na střed nebo doprava. Vybrané zarovnání je znázorněno ikonou na levé straně každého řádku, jak je vidět na [Obrázku 19a](#). Na zmíněném obrázku je zarovnání u všech řádků nastaveno na zarovnání na střed. Dále lze u řádku nastavit, jestli se má vytisknout větším fontem. Příklad takto nastaveného a vytisknutého řádku je první řádek záhlaví na [Obrázku 19](#).

Pořadí jednotlivých řádků lze změnit chycením řádku, který má být přesunut, za ikonu šipek na pravé straně daného řádku, a přetažením na požadovanou pozici. Po puštění na cílovém místě se nové pořadí automaticky uloží.

Aktuální nastavení je možné zkušebně vytisknout tlačítkem „Zkušební tisk“. Na takto vytisknuté účtence jsou testovací data pro účely kontroly tisku a také informace, že se jedná pouze o zkušební tisk, nikoli o skutečný doklad.

Délka textu na řádku je omezena dodatečným nastavením, ve kterém se nastavuje maximální šířka řádku. Tato šířka je daná použitou tiskárnou. Dále se v „Dalším nastavení“ dá nastavit, má-li se účtenka tisknout s diakritikou nebo bez, což také záleží na konkrétní tiskárně. V příloženém obrázku je nastaven tisk bez diakritiky.

Účtenka

Záhlaví účtenky

[+](#) Přidat řádek

☰	KAVÁRNA	↑↓
☰	Josef Novák	↑↓
☰	Za branami, 357	↑↓
☰	Zderaz, 687 24	↑↓
☰	IČ: 1262717, DIČ: CZ16271717172	↑↓

Zápatí účtenky

[+](#) Přidat řádek

☰	Děkujeme za návštěvu!	↑↓
---	-----------------------	----

 ZKUŠEBNÍ TISK

[Další nastavení >](#)

KAVARNA				
Josef Novak				
Za branami, 357				
Zderaz, 687 24				
IC: 1262717, DIC: CZ16271717172				
Cislo uctenky: 70				
Stul: Stul 5				
Datum: 11. 03. 2023, 19:20				
Platba: Hotovost				
Ks	Polozka	Kc/ks	Kc	%D
2x	Cappuccino	52	104	15
1x	Malinovy cheesecake	65	65	21
Cena: 169 Kc				
%D	Zaklad	Dan	Celkem	
15	90.42	13.58	104.0	
21	53.71	11.29	65.0	
Dekujeme za navstevu!				

(a) Nastavení – text účtenky

(b) Příklad vytištěné účtenky

Obrázek 19: Nastavení – Účtenka a ukázka vytištěné účtenky

4.5.4 Denní nabídka

Funkcionalita „Denní nabídka“ je navržena z toho důvodu, že kavárny v nabídce střídají např. 30 druhů zákusků, ale každý den mají pouze určité množství z nich³³. Díky této funkcionalitě je možné – přidáním či odebráním jednotlivých produktů z denní nabídky, a tím jejich vizuálním odlišením od ostatních – zrychlit proces vyhledávání mezi produkty při přidávání produktu na stůl. Obsluha na první pohled vidí, které ze všech dortů jsou v daný okamžik k dispozici.

V nastavení je možné nastavit:

- Zobrazit – je-li tato funkcionalita povolena a bude tedy možné produkty přidat do denní nabídky, či nikoli.
- Zobrazit jako samostatnou skupinu produktů – toto nastavení má smysl, pouze pokud je povolena předchozí možnost. Pokud ano, a tato možnost je povolena, zobrazí se produkty vybrané do denní nabídky v samostatné skupině produktů a budou navíc označeny zeleným pruhem na levé straně karty produktu. Pokud je tato možnost vypnutá, skupina se nezobrazí a produkty budou pouze označeny zeleným pruhem.

Příklad denní nabídky, i s povolenou možností zobrazení v samostatné skupině produktů, lze vidět na [Obrázku 8](#).

4.5.5 Objednávkové bony

Je-li povolena možnost „Tisknout objednávky“, vytiskne se při zavření stolu po provedení změn objednávkový bon. Je na něm vytištěno jeho pořadové číslo, čas objednávky, stůl a jednotlivé produkty.

Pořadové číslo bonů se s každým dalším vytisknutým bonem zvyšuje o 1. Toto číslo se vynuluje uzavřením směny – první vytisknutý bon po uzavření směny bude mít tedy číslo 1.

4.5.6 Uživatelé

Tato část slouží ke správě uživatelů. Uživatelé jsou rozděleni do tří skupin, podle jejich oprávnění:

- Admin – uživatel s nejvyšším oprávněním, typicky tedy majitel/ka kavárny; může nastavovat a provádět vše v rámci celé aplikace.
- Vedoucí směny – uživatel s tímto oprávněním může nastavovat vše v části „Provoz“, přidávat pohyby hotovosti (viz sekce [4.3.1](#)) a uzavírat směny (viz sekce [4.3.2](#)).
- Brigádník – toto oprávnění stačí pouze na změny produktů patřících do denní nabídky.

³³Zákusky jsou pouze vybranou skupinou, může se to vztahovat na libovolnou skupinu produktů z nabídky kavárny.

Jakýkoli uživatel může vytvářet a platit účty na stolech a zobrazit si přehled tržby směny.

Přidání uživatelé jsou zobrazení v seznamu rozděleném podle oprávnění, viz [Obrázek 20a](#).

Při první instalaci je automaticky přidán uživatel s oprávněním „Admin“, jehož přihlašovací jméno je: Admin a heslo: kava. Heslo lze změnit kliknutím na kartu libovolného uživatele, rozkliknutím sekce „Změnit heslo“ a zadáním nového hesla. Heslo je, i při přidávání nového uživatele, nutné zadat dvakrát pro kontrolu správnosti hesla. Okno pro přidání uživatele je na [Obrázku 20b](#).

Uživatelé

+ Přidat uživatele

Admin

Jméno
Admin

Vedoucí směny

Jméno
Martin

Brigádník

Jméno
Jakub

Nový uživatel ×

Jméno

Heslo

Potvrďte heslo

Oprávnění:

Admin Vedoucí směny Brigádník

PŘIDAT

(a) Seznam uživatelů

(b) Okno pro přidání uživatele

Obrázek 20: Nastavení – uživatelé

4.5.7 Otevírací doba

Zde může majitel/ka nastavit otevírací dobu kavárny . V dialogovém okně vybere čas od a čas do. Tato otevírací doba ovlivňuje graf hodinového vytížení kavárny ve statistikách, viz sekce [4.4.1](#)).

4.5.8 Jednorázový prodej

Tento režim prodeje slouží k tzv. jednorázovému prodeji, tedy například k prodeji ve stánku či prodejně³⁴. Rozdíl oproti běžnému režimu je v tom, že v jednorázovém prodeji není k dispozici mapa stolů a místností – místo obrazovky [4.2](#) se zobrazuje vzhledově upravený detail stolu (viz obrazovka [8](#)). Tato obrazovka se jmenuje Jednorázový prodej a detail stolu, který se v ní zobrazuje, umožňuje pouze přidat produkty na účet a následně je zaplatit; tlačítka pro Přesun a Uložení tedy neobsahuje. Po zaplacení všech přidávaných produktů nedojde k zavření náhledu stolu jako tomu je v základní variantě se stoly, takže je možné rovnou markovat další produkty a vytvořit nový účet.

³⁴Prodejnou je myšlena běžná prodejna nějakého zboží bez míst k sezení.

4.5.9 Plátcí DPH

Jestliže je provozovatel kavárny plátcem DPH, musí tato možnost být povolena. Povolením dojde k zapnutí možnosti zadat k jednotlivým produktům sazbu DPH, a ta se bude následně propisovat na účtenky spolu s tabulkou obsahující výpočet základu a daně jednotlivých skupin DPH obsažených na účtence. Příklad je vidět na [Obrázku 19b](#).

Pokud je tato možnost vypnuta, na účtence je místo toho informace, že provozovatel podniku není plátcem DPH.

Závěr

V rámci této bakalářské práce jsem na základě vlastních zkušeností a analýzy již existujících řešení navrhl a vyvinul pokladní systém pro kavárny a další podobné podniky pro platformu Android. Aplikace je primárně vyvinuta pro tablety, lze ji ale používat i na mobilních telefonech bez omezení na funkčnosti. Celý pokladní systém byl původně mířen na kavárny, ale díky konfigurovatelnosti jej lze využít i v řadě dalších podniků, jako například ve stáncích, malých prodejnách nebo barech.

Aplikace byla konzultována s majitelkou kavárny Lucií Uhrovou, jejíž zpětná vazba je přiložena v příloze [A](#), a otestována zaměstnanci kavárny.

V blízké budoucnosti plánuji přidat možnost vytváření více oddělených účtů na jednom stole, export a import dat, export dat pro účetnictví, komunikaci s platebními terminály pro platbu kartou, prodej produktů na váhu, podporu dalších technologií tisku – například Wi-Fi tiskáren. V dlouhodobějším horizontu bych chtěl přidat správu skladů a možnost přesunout lokální databázi z tabletu na lokální webový server a následně dodělat mobilního číšníka.

Conclusions

In this bachelor thesis I have, based on my personal experience and analysis of existing solutions, designed and implemented cash register application for cafes and other similar businesses on Android platform. The application is primarily made for tablets, however it can be used on mobile phones as well without any limitations. Originally, the cash register application was supposed to be made only for cafes, but thanks to its adjustability, it can be used in number of other businesses too, such as kiosks or small bars or stores.

The application was consulted with Lucie Uhrová, owner of a cafe, whos feedback is in attachment [A](#) and tested by stuff of the cafe.

In the near future I am planning on adding possibility of creating multiple separated orders on one table, export and import of data, export of data for accountancy, communication with terminals for credit card payment, selling products by weight and support of other printing technologies such as Wi-Fi printers. In more distant future I would like to add storage management and an option of transferring the local database from tablet to local web server, thanks to which I would be able to develop “mobilního číšníka”.

A Zpětná vazba majitelky kavárny

V této příloze je zpětná vazba Lucie Uhrové, dlouholeté majitelky kavárny, které jsem aplikaci představil, a která měla možnost si ji vyzkoušet a otestovat.

„Prezentace pokladního systému byla skvělá. Aplikace je přehledná, tvárná, lehce pochopitelná, má pěkné uživatelské prostředí a barevné provedení. Ze základních bodů splňuje vše, co je třeba. Pokud by se dodělalo pár detailů, jako například vzdálený přístup, skladové hospodářství a také měsíční přehled tržeb s rozepsaným DPH, s klidným svědomím bych aplikaci využila jako pokladní systém do svého podniku.“

B Obsah elektronických dat

Součástí práce jsou elektronická data v systému katedry informatiky s touto strukturou:

bin/

Adresář obsahující soubor .apk pro instalaci aplikace.

src/

Adresář obsahující veškeré soubory a adresáře (v ZIP archivu) nutné pro vytvoření spustitelného .apk souboru pomocí Android Studia. Jde tedy o celý projekt Android Studia.

text/

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

README.txt

Textový soubor s detailně popsány jednotlivými instrukcemi pro instalaci a spuštění aplikace.

U veškerých cizích obsažených materiálů jejich zahrnutí dovoluují podmínky pro jejich veřejné šíření nebo přiložený souhlas držitele práv k užití. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy, je uveden jejich zdroj, např. webová adresa, v bibliografii nebo textu práce nebo souboru README.txt.

Literatura

- [1] *Dotykačka - pokladní systém pro restaurace, obchody a služby / Dotykačka*. [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://dotykacka.cz/>.
- [2] *Qerko / Nejvhodnější způsob, jak platit v restauracích*. [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://www.qerko.com/cs>.
- [3] *Storyous – moderní pokladní systém / Stvořeno pro gastro*. [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://storyous.com/cz/>.
- [4] *SaltPay - Přinášíme svět moderních platebních řešení*. [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://salt-pay.com/>.
- [5] <https://www.solidpixels.com> solidpixels. <https://www.solidpixels.com>. *Septim Air - pokladní systém pro restaurace* [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://www.septim-air.cz/>.
- [6] *ASW Systems - Pokladní systém Septim, hotelový systém Protel / ASW Systems a.s.* [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://www.septim.cz/>.
- [7] *Pokladní systémy, pokladny a aplikace / KASAmox*. [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://www.kasamax.cz/>.
- [8] Smable. *Pokladna Smable - Android pokladna - Vyzkoušejte zdarma* [online]. 2023 [cit. 2023-3-27]. Dostupný z: <https://www.smable.cz/>.
- [9] *Kotlin overview / Android Developers*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://developer.android.com/kotlin/overview>.
- [10] *Kotlin Programming Language*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://kotlinlang.org/>.
- [11] *Layouts / Android Developers*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://developer.android.com/develop/ui/views/layout/declaring-layout>.
- [12] *Android Sqlite Tutorial - Javatpoint*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://www.javatpoint.com/android-sqlite-tutorial>.
- [13] Jahoda, Phillip. *GitHub - PhilJay/MPAndroidChart: A powerful Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, panning and animations*. — [github.com](https://github.com/PhilJay/MPAndroidChart) [online]. [cit. 2023-3-22]. Dostupný z: <https://github.com/PhilJay/MPAndroidChart>.
- [14] *GitHub - vmadalin/easypermissions-ktx: Kotlin version of the popular google/easypermissions wrapper library to simplify basic system permissions logic on Android M or higher*. — [github.com](https://github.com/vmadalin/easypermissions-ktx). [online]. [cit. 2023-3-22]. Dostupný z: <https://github.com/vmadalin/easypermissions-ktx>.
- [15] *GitHub - googlesamples/easypermissions: Simplify Android M system permissions* — [github.com](https://github.com/googlesamples/easypermissions). [online]. [cit. 2023-3-22]. Dostupný z: <https://github.com/googlesamples/easypermissions>.

- [16] *Maven Repository: org.mindrot » jbcrypt*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://mvnrepository.com/artifact/org.mindrot/jbcrypt>.
- [17] Rashed, Mazen. *mazenrashed/Printooth: A well documented, high-level Android interface that makes printing via bluetooth printers easier* [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://github.com/mazenrashed/Printooth>.
- [18] *Coroutines basics / Kotlin Documentation*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://kotlinlang.org/docs/coroutines-basics.html#your-first-coroutine>.
- [19] *Kotlin/kotlinx.coroutines: Library support for Kotlin coroutines*. [online]. 2023 [cit. 2023-3-23]. Dostupný z: <https://github.com/Kotlin/kotlinx.coroutines>.
- [20] *Android Jetpack Dev Resources - Android Developers — developer.android.com*. [online]. [cit. 2023-3-21]. Dostupný z: <https://developer.android.com/jetpack>.
- [21] *Navigation / Android Developers — developer.android.com*. [online]. [cit. 2023-3-21]. Dostupný z: <https://developer.android.com/guide/navigation>.
- [22] *ViewModel overview / Android Developers — developer.android.com*. [online]. [cit. 2023-3-21]. Dostupný z: https://developer.android.com/topic/libraries/architecture/viewmodel?gclid=CjwKCAjwiOCgBhAgEiwAjbv5whBIQRKhfrhBLmYSsmQfJ-q4hO87lrcgT_8F5EweI-pN5ZvTayLbfzBoCbrsQAvD_BwE&gclidsrc=aw.ds.
- [23] *Save data in a local database using Room / Android Developers — developer.android.com*. Dostupný z: <https://developer.android.com/training/data-storage/room>.
- [24] *Material Design for Android / Android Developers*. [online]. 2022 [cit. 2023-3-23]. Dostupný z: <https://developer.android.com/develop/ui/views/theming/look-and-feel>.
- [25] *Introduction to activities / Android Developers*. [online]. [cit. 2023-3-21]. Dostupný z: <https://developer.android.com/guide/components/activities/intro-activities>.
- [26] Landa, Jaromír. *Aktivita—prvopočátek všeho — medium.com* [online]. 2021 [cit. 2023-3-21]. Dostupný z: <https://medium.com/android-na-pef/aktivita-prvopo%C4%8D%C3%A1tek-v%C5%A1eho-902aced25e1e>.
- [27] *Activity / Android Developers — developer.android.com*. [online]. [cit. 2023-3-21]. Dostupný z: <https://developer.android.com/reference/android/app/Activity>.

- [28] *Fragments / Android Developers* — *developer.android.com*. [online]. [cit. 2023-3-21]. Dostupný z: <https://developer.android.com/guide/fragments>.
- [29] Miguel. *Modernize Your Android App With The Single Activity Pattern* — *levelup.gitconnected.com* [online]. 2022 [cit. 2023-3-21]. Dostupný z: <https://levelup.gitconnected.com/modernize-your-android-app-with-the-single-activity-pattern-1526407a7d4>.
- [30] *MVVM (Model View ViewModel) Architecture Pattern in Android - GeeksforGeeks* — *geeksforgeeks.org*. [online]. [cit. 2023-3-21]. Dostupný z: <https://www.geeksforgeeks.org/mvvm-model-view-viewmodel-architecture-pattern-in-android/>.
- [31] *GitHub - amitshekhariitbhu/MVVM-Architecture-Android: MVVM architecture using Kotlin, Dagger, Retrofit, Coroutines, Flow, StateFlow, and etc.* — *github.com*. [online]. [cit. 2023-3-21]. Dostupný z: <https://github.com/amitshekhariitbhu/MVVM-Architecture-Android>.