

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Android aplikace pro podporu firemního účetnictví

Bc. Tomáš Kříčenský

Diplomová práce

2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Tomáš Kříčenský**
Osobní číslo: **I19283**
Studijní program: **N0613A140007 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Android aplikace pro podporu firemního účetnictví**
Zadávající katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Cílem diplomové práce je naprogramovat mobilní aplikaci pro operační systém Android, která umožní nahrávání a správu firemních účtenek a faktur. Účtenku bude možné nahrát z existujícího souboru nebo pořídít fotografii přímo v aplikaci. Nad vybraným obrázkem bude provedeno automatické vyhledání některých údajů pomocí OCR. Aplikace bude komunikovat se serverem pomocí technologie GraphQL.

Rozsah pracovní zprávy: **50**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

- *JEMEROV, Dmitry a Svetlana ISAKOVA. *Kotlin in action*. Shelter Island, NY: Manning Publications Co., 2017. ISBN 978-161-7293-290.
- *RAYWENDERLICH TUTORIAL TEAM, Filip BABIČ a Nishant SRIVASTAVA. *Kotlin Coroutines by Tutorials: Mastering coroutines in Kotlin and Android*. USA: Razeware, 2019. ISBN 978-1942878681.
- *PORCELLO, Eve a Alex BANKS. *Learning GraphQL: Declarative Data Fetching for Modern Web Apps*. USA: O'Reilly Media, 2018. ISBN 978-1492030713

Vedoucí diplomové práce: **Ing. Soňa Neradová, Ph.D.**
Katedra informačních technologií

Datum zadání diplomové práce: **6. listopadu 2020**
Termín odevzdání diplomové práce: **15. května 2021**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 30. listopadu 2020

Prohlašuji:

Práci s názvem *Android aplikace pro podporu firemního účetnictví* jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 8. 2021

Bc. Tomáš Kříčenský

Poděkování

Tímto bych chtěl poděkovat své vedoucí Ing. Soně Neradové, Ph.D. za odbornou pomoc a cenné rady, které mi pomohly při zpracování této práce. Děkuji i své rodině za veškerou podporu.

ANOTACE

Diplomová práce se zabývá tvorbou mobilní aplikace pro archivaci a správu účetních dokladů. Práce je implementována pro platformu Android s využitím návrhového vzoru Model-View-ViewModel. Před nahráním do systému jsou účetní doklady převedeny na text pomocí optického rozpoznávání znaků (OCR). Poté jsou v tomto textu automaticky nalezeny důležité údaje. V rámci práce jsou prozkoumány, realizovány a porovnány tři různé implementace OCR. Pro komunikaci se serverem, kde jsou data uložena, je použita technologie GraphQL. Aplikace funguje i v režimu bez připojení k internetu díky využití lokální databáze.

KLÍČOVÁ SLOVA

Android, mobilní aplikace, optické rozpoznávání znaků, účetní doklady, GraphQL.

TITLE

Android application to support corporate accounting

ANNOTATION

The thesis deals with the creation of a mobile application used to archive and manage business records. The application is implemented for the Android platform following the Model-View-ViewModel architectural pattern. Before the documents are uploaded to the system they are converted to text by optical character recognition (OCR) tools. This text is then scanned for important information. The thesis explores, realizes and compares three different OCR implementations. The application uses GraphQL technology to communicate with the server which stores all the data. The application works also without internet connection thanks to the use of a local database.

KEYWORDS

Android, mobile application, optical character recognition, business records, GraphQL.

OBSAH

| | |
|--|-----------|
| Seznam obrázků | 9 |
| Seznam tabulek | 10 |
| Seznam zkratk | 11 |
| Úvod | 12 |
| 1 Stávající řešení | 13 |
| 1.1 Popis systému | 14 |
| 1.2 Konkurenční aplikace | 16 |
| 1.2.1 Wflow | 16 |
| 1.2.2 INSIO DMS | 17 |
| 1.2.3 Porovnání | 18 |
| 2 Použité technologie | 20 |
| 2.1 Android SDK | 20 |
| 2.2 Kotlin | 21 |
| 2.2.1 Kotlin Coroutines | 22 |
| 2.3 Koin | 23 |
| 2.4 Model-View-ViewModel | 25 |
| 2.5 Material design a tmavý režim | 26 |
| 2.6 GraphQL | 27 |
| 2.7 CameraX | 28 |
| 3 Rozpoznání informací | 29 |
| 3.1 Rozpoznání textu na zařízení | 29 |
| 3.1.1 Tesseract | 30 |
| 3.1.2 ML Kit | 31 |
| 3.2 Rozpoznání textu v cloudu | 31 |
| 3.2.1 Microsoft Computer Vision | 32 |
| 3.3 Vyhledání potřebných dat v textu | 33 |

| | | |
|----------|---|-----------|
| 3.4 | Analýza výsledků | 35 |
| 4 | Popis aplikace | 41 |
| 4.1 | Spouštěcí obrazovka | 42 |
| 4.2 | Přihlášení | 43 |
| 4.3 | Seznam účtenek a měsíční souhrn | 44 |
| 4.3.1 | Seznam účtenek | 45 |
| 4.3.2 | Možnosti práce s účtenkou | 49 |
| 4.3.3 | Měsíční souhrn | 52 |
| 4.4 | Přidání nové účtenky | 54 |
| 4.4.1 | Fotoaparát v rámci aplikace | 55 |
| 4.4.2 | Formulář pro přidání účtenky | 57 |
| 4.5 | Detail účtenky | 62 |
| 4.5.1 | Detail dokumentu či fotografie | 65 |
| 4.6 | Úprava účtenky | 66 |
| 4.7 | Nastavení | 66 |
| 4.7.1 | Změna osobních údajů | 69 |
| 4.7.2 | Změna hesla | 70 |
| | Závěr | 71 |
| | Použitá literatura | 72 |

SEZNAM OBRÁZKŮ

| | | |
|------------|---|----|
| Obrázek 1 | Ukázka paralelního načítání zdrojů pomocí Kotlin Coroutines | 23 |
| Obrázek 2 | Ukázka definice pravidel pro vytváření závislostí pomocí Koin | 24 |
| Obrázek 3 | Interakce mezi modelem, view a ViewModelem v architektuře MVVM | 25 |
| Obrázek 4 | Ukázka GraphQL dotazu | 27 |
| Obrázek 5 | Počty správně předvyplněných hodnot nalezených pomocí OCR | 37 |
| Obrázek 6 | Počty správných hodnot nalezených pomocí OCR (ne vždy vybraných) | 39 |
| Obrázek 7 | Spouštěcí obrazovka ve světlém a tmavém režimu | 42 |
| Obrázek 8 | Přihlašovací obrazovka a chybová hláška | 43 |
| Obrázek 9 | Seznam účtenek a kalendář pro výběr měsíce | 46 |
| Obrázek 10 | Možnosti práce s účtenkou | 50 |
| Obrázek 11 | Měsíční souhrn ve světlém a tmavém režimu | 53 |
| Obrázek 12 | Fotoaparát uvnitř aplikace | 56 |
| Obrázek 13 | Přidání nové účtenky | 58 |
| Obrázek 14 | Detail účtenky | 63 |
| Obrázek 15 | Nastavení a změna hesla | 67 |

SEZNAM TABULEK

| | | |
|-----------|---|----|
| Tabulka 1 | Přihlašovací údaje pro testování odevzdané aplikace | 15 |
| Tabulka 2 | Porovnání výsledků jednotlivých implementací OCR | 36 |
| Tabulka 3 | Počty správně nalezených hodnot ve fotografiích | 38 |

SEZNAM ZKRATEK

| | |
|-------|--|
| API | Application Programming Interface (rozhraní pro programování aplikací) |
| DMS | Document Management System (systém pro správu dokumentů) |
| DPH | Daň z přidané hodnoty |
| DSL | Doménově specifický jazyk |
| HTTP | Hypertext Transfer Protocol |
| ISDOC | Information System Document (formát dokumentů pro elektronickou fakturaci) |
| ISO | International Organization for Standardization |
| JPEG | Komprimovaný formát pro fotografie |
| JSON | JavaScript Object Notation |
| MVVM | Model-View-ViewModel |
| NDK | Native Development Kit (softwarové nástroje pro vývoj v jazyce C nebo C++ pro operační systém Android) |
| OCR | Optical Character Recognition (optické rozpoznávání znaků) |
| OLED | Organic light-emitting diode (technologie displejů některých telefonů) |
| PDF | Portable Document Format (formát elektronických dokumentů) |
| REST | Representational state transfer (typ implementace serverového API) |
| SDK | Software Development Kit (softwarové nástroje pro vývojáře) |
| URL | Uniform Resource Locator (v případě této práce se jedná výhradně o webovou adresu) |

ÚVOD

Každá fyzická osoba nebo společnost, která má ze zákona povinnost vést účetnictví, musí archivovat účetní a mzdové doklady. Tyto doklady musí společnost uchovávat po dobu deseti let. Doklady je možné uchovávat v papírové nebo digitální formě. [1]

Cílem této diplomové práce je vytvoření mobilní aplikace, která dané společnosti usnadní evidenci těchto dokladů. Pomocí vytvořené mobilní aplikace mohou zaměstnanci dané společnosti evidovat účetní doklady, které vznikly v rámci jejich práce. Samotná mobilní aplikace doklady nearchivuje, ale nahrává je na firemní server, který data uchovává a poskytuje další funkce. Aplikaci je tudíž možné použít pouze v kombinaci se serverovým řešením, které je majetkem společnosti, pro kterou byla aplikace vytvořena. Tato aplikace je tedy určena pouze pro zaměstnance a vedení dané firmy.

Účetní doklady mohou uživatelé vyfotit pomocí fotoaparátu integrovaného přímo v mobilní aplikaci. Případně mohou vybrat již vyfocené fotografie účetních dokladů přímo z galerie mobilního telefonu. Také mohou vybrat PDF soubor daného dokladu z lokálního úložiště.

K takto pořízenému účetnímu dokladu uživatelé připojí informace, které se na něm nacházejí. Příkladem těchto informací mohou být celková cena, cena bez DPH, měna nebo datum nákupu. Tyto údaje uživatelé nemusejí vždy vyplňovat ručně, jelikož vypracovaná aplikace fotografie při zpracování automaticky převede na text pomocí optického rozpoznávání znaků (OCR) a následně v textu tyto údaje vyhledá.

V rámci této práce jsou prozkoumány tři různé implementace OCR a uživatel si v aplikaci může sám vybrat, kterou z implementací chce používat. Jednotlivé implementace OCR jsou v rámci této práce podrobeny testům přesnosti a porovnány. Přesnost však není jedinou porovnávanou metrikou. Dalšími metrikami jsou například rychlost zpracování obrázků nebo rychlost samotné implementace.

Vypracovaná aplikace také umožňuje správu nahraných účetních dokladů. Vedoucí pracovníci vidí všechny firemní doklady a mohou je jednotlivým uživatelům schvalovat nebo mohou celé měsíce uzavírat, aby do nich už žádný uživatel nemohl nahrát nové doklady. Aplikace obsahuje i jednoduchý měsíční přehled firemních výdajů, který ukazuje, v jaké kategorii společnost utrací nejvíce finančních prostředků.

1 STÁVAJÍCÍ ŘEŠENÍ

Aplikace vytvořená v rámci této práce byla vytvořena pro nejmenovanou společnost, která má vlastní serverové řešení pro evidenci a správu svých účetních dokladů. Toto serverové řešení mělo v době zadání této práce pouze jedno uživatelské rozhraní. Tím rozhraním byly webové stránky, které používali jak řadoví zaměstnanci společnosti, tak vedoucí pracovníci a účetní.

Výše zmíněné webové stránky poskytují přehled všech firemních účetních dokladů se skupených dle jednotlivých měsíců. Webové stránky poskytují možnost nahrání nových účetních dokladů i jejich správu. Kromě správy účetních dokladů umožňuje webová aplikace vedoucím pracovníkům spravovat seznam měn, kategorií a statusů účetních dokladů. Jelikož se jedná o uzavřený systém, do kterého se uživatel nemůže sám registrovat, vedoucí pracovník má možnost vytvořit novému zaměstnanci zaměstnanecký účet. Tento účet bude mít pouze oprávnění, která mu vedoucí nastaví. Tento uživatel poté například pouze uvidí účetní doklady, které do webové aplikace sám nahrál, a nevidí doklady ostatních uživatelů.

Motivací pro vytvoření mobilní aplikace bylo, že jediným existujícím uživatelským rozhraním pro dříve zmíněný systém bylo webové rozhraní. Přestože se jednalo o responzivní web, nebylo pro běžné uživatele příjemné ho využívat z mobilních zařízení. Načítání bylo pomalé a některé funkce webu byly vytvořeny na míru právě pro vedoucí pracovníky a jejich pokročilou správu.

Z toho důvodu vznikl nápad pro vytvoření mobilní aplikace, která by byla více zaměřena na běžné uživatele a jednoduchost nahrání nového účetního dokladu do systému. Tato aplikace by stále obsahovala ovládací prvky pro vedoucí pracovníky, ale také by zjednodušila a zpříjemnila proces pro přidání nových dokladů.

Mobilní aplikace zjednoduší přidávání nových účetních dokladů do systému možností vyfotit daný doklad přímo v mobilní aplikaci a také využitím optického rozpoznávání znaků (OCR). Aplikace při zpracování fotografie dokladu ve fotografii najde text a v tomto textu následně automaticky vyhledá důležité údaje, které by uživatel musel hledat ručně. Tyto údaje jsou následně předvyplněny do formuláře pro nahrání nového účetního dokladu.

1.1 Popis systému

Jak již bylo dříve zmíněno, české společnosti mají povinnost archivovat své účetní a mzdové doklady. Tyto doklady je nutné archivovat po dobu deseti let. Společnostem ale stačí tyto doklady uchovávat v digitální formě. K tomuto účelu si nejmenovaná společnost vytvořila vlastní serverové řešení, které poměrně jednoduchým způsobem řeší tento problém. [1]

Vytvoření vlastního řešení pro archivaci a správu účetních dokladů společnosti má několik výhod. První výhodou je, že se jedná o řešení na míru. Výsledný produkt pak neobsahuje zbytečné nebo složité funkce, které by nakonec nikdo nevyužíval. Naopak je možné do systému implementovat funkce, které by ostatní společnosti nevyužily, ale v této konkrétní společnosti by mohly zaměstnancům velmi usnadnit jejich práci.

Další výhodou pro mnohé společnosti je i to, že díky použití vlastního řešení se účetní doklady nikdy nedostanou mimo samotnou společnost. Doklady jsou uloženy na firemním serveru a spravuje je proprietární software dané společnosti. Tím je dosaženo zvýšeného zabezpečení, kterému společnost věří, jelikož ví, jak celý systém funguje.

Aby bylo dosaženo zabezpečení citlivých dat, které systém obsahuje, systém nedovoluje samostatnou registraci nových uživatelů. Každý uživatelský účet musí vytvořit stávající vedoucí pracovník s oprávněním administrátora. Nový uživatelský účet dostane přidělenou jednu ze tří předdefinovaných rolí v systému. Ke každé uživatelské roli se vážou jiná oprávnění pro to, co může uživatel v systému vidět a také co může měnit.

Základní úrovní oprávnění je běžný uživatel. Tuto uživatelskou roli má velká většina koncových uživatelů systému. Běžný uživatel vidí v přehledech pouze účetní doklady, které do systému sám nahrál nebo mu je přiřadil administrátor. Běžný uživatel tedy může do systému nahrávat nové doklady a může tyto doklady upravovat, dokud neprojdou schvalovacím procesem. Běžný uživatel si dále může měnit svoje osobní údaje včetně hesla a prohlížet si svoje dříve nahrané účetní doklady.

Prostřední úrovní uživatelských oprávnění je role účetní. Uživatel s tímto oprávněním je v systému většinou pouze jeden. Uživatel s touto rolí má všechna práva, která má běžný uživatel. Kromě těchto práv může uživatel s rolí účetní vidět všechny účetní doklady, které systém obsahuje. Účetní však nemá právo tyto doklady schvalovat či jakkoliv měnit.

Nejvyšší úroveň oprávnění má vedoucí pracovník neboli administrátor. Uživatelů s tímto oprávněním je ve společnosti jen několik a jedná se tedy o vysoce postavené pracovníky

nebo uživatele, kteří tuto úroveň přístupu potřebují k plnění své práce. Administrátor má samozřejmě všechna dříve zmíněná oprávnění, ale k tomu i mnohá další.

Jak bylo dříve zmíněno, administrátor má práva spravovat seznam uživatelů celého systému. To znamená, že může přidávat nové uživatele, může stávající uživatele mazat anebo jim změnit některé údaje, pokud je to potřeba. Kromě uživatelů může administrátor také spravovat seznam dostupných měn, kategorií účetních dokladů nebo jejich statusů. Všechny tyto úkony může provádět pomocí existující webové aplikace.

Kromě nastavování výše zmíněných možností má administrátor nadstandardní možnosti pro práci s účetními doklady. Administrátor může schvalovat nebo i zakazovat doklady nahrané ostatními uživateli. Na rozdíl od ostatních rolí má administrátor právo upravovat již schválené doklady. Nejčastěji je po schválení upravuje s účelem změnit jejich status, který říká, zda již byly proplaceny, nebo nikoliv.

Další velmi důležitou pravomocí administrátora je, že může uzavírat jednotlivé kalendářní měsíce. Ve chvíli, kdy administrátor měsíc uzavře, tak již není možné do něj nahrávat nové účetní doklady a již nahraným dokladům jsou vygenerovány unikátní identifikátory důležité pro účetní. V rámci uzavřeného měsíce již nesmí doklady upravovat ani administrátor a všechny účtenky tedy zůstanou ve stavu, ve kterém byly v době uzavření daného měsíce. Jedinou výjimkou pro možné úpravy po uzavření měsíce je úprava statusu účetního dokladu administrátorem.

Tabulka 1 obsahuje přihlašovací údaje, které je možné použít pro testování odevzdané aplikace. Jedná se o přístupy do testovacího prostředí a nemůže tedy dojít k úniku dat. Tabulka obsahuje údaje pro jednoho uživatele s každou podporovanou úrovní oprávnění. Je tedy možné v aplikaci pozorovat rozdíly mezi jednotlivými možnostmi, které uživatele mají.

Tabulka 1 – Přihlašovací údaje pro testování odevzdané aplikace.

| Přihlašovací jméno | Heslo | Úroveň oprávnění |
|---------------------------|--------------------|---------------------------------|
| administrator | AdminPa55word | Nejvyšší |
| ucetni | AccountantPa55word | Účetní (vidí vše, nemůže měnit) |
| uzivatel | UserPa55word | Běžný uživatel |

1.2 Konkurenční aplikace

V rámci této kapitoly jsou prozkoumána různá konkurenční řešení pro elektronickou archivaci dokumentů. Tato řešení budou porovnána mezi sebou a také s řešením využitým v aplikaci vypracované v rámci této práce.

1.2.1 Wflow

Prvním prozkoumaným konkurenčním řešením je wflow.com. Wflow si klade za cíl digitalizovat účetnictví společnosti, která tento systém používá. Primárním zdrojem účetních dokladů v systému je sběrný e-mail, na který je možné doklady posílat. Z e-mailové schránky jsou doklady automaticky vytěžovány a připraveny pro další zpracování a schválení v rámci systému. Kromě zasílání e-mailů je možné doklady do systému nahrávat ručně pomocí webového rozhraní nebo mobilní aplikace. [2]

Nahrané doklady jsou analyzovány pomocí umělé inteligence Rossum, která v dokladech najde požadované údaje. Nalezené údaje je nutné později ručně potvrdit při kontrole dokumentu. Pokud je nahraný dokument ve formátu ISDOC (Information System Document), není nutné v něm hledat údaje pomocí umělé inteligence, protože jsou údaje vytaženy přímo z dokumentu. [2]

Vytěžené dokumenty jsou v systému dostupné pro následné schvalování a fakturaci. V rámci schvalovacího procesu je možné nastavit, že dokument musí schválit několik různých osob. Po schválení dokumentu může dojít k automatickému zaúčtování dokladu. To je možné díky integrací s několika účetními systémy. Aktuálně podporované účetní systémy jsou Pohoda, HELIOS Orange, ABRA Flexi, 6K, Money S3, DUAL a Aconto. [2]

Jak bylo zmíněno výše, součástí produktu Wflow je i mobilní aplikace. Tato aplikace slouží pouze k nahrávání nových účetních dokladů. Doklady je možné fotit pomocí integrovaného fotoaparátu nebo je možné vybrat již vyfocené fotografie z galerie. Ve fotografii jsou automaticky identifikovány okraje účetního dokladu a do systému se nahraje pouze oříznutá část bez pozadí. V rámci jednoho dokumentu lze připojit do systému více fotografií. [2]

Wflow je možné zakoupit jako předplatné ve čtyřech různých konfiguracích. Nejlevnější verze stojí 109 Kč měsíčně a umožňuje systém používat pouze jednomu uživateli. Další uživatele je možné dokoupit za stejnou cenu jako celé předplatné. V této konfiguraci je

k dispozici také úložiště dokumentů o velikosti pouze 1 GB. Také je zde limit 15 dokumentů pro automatické vytěžování pomocí umělé inteligence Rossum. [3] Nejdražší verze předplatného Wflow nemá pevně stanovenou cenu a domlouvá se na míru pro danou organizaci. Tato verze obsahuje neomezené úložiště pro skladování dokumentů, neomezené množství dokumentů, systém pro objednávky a další. Nejdražší běžně dostupné předplatné je k dispozici za 499 Kč měsíčně a obsahuje stejné funkce, ale s omezením na 50 GB úložiště a stále je nutné přikupovat přístup pro další uživatele ve stejné výši, jako je cena celého předplatného. [3]

1.2.2 INSIO DMS

Druhým prozkoumaným konkurenčním řešením je INSIO DMS (Document management system) od společnosti INSIO software s.r.o. Jedná se o systém pro digitální správu dokumentů. [4]

INSIO DMS je možné ve firmě použít nejen pro správu faktur, ale i pro vytvoření objednávky. Po vytvoření musí objednávka projít schvalovacím procesem. Ve chvíli, kdy je objednávka schválena, je možné automaticky vygenerovat e-mail pro dodavatele spolu s dokumentem samotné objednávky. [4]

Samotná faktura bude do systému nahrána automaticky, pokud ji dodavatel pošle na sběrnou e-mailovou adresu. Údaje z této faktury může do systému uživatel přepsat ručně, napojit na dřívější objednávku nebo využije automatického vytěžování dat. INSIO podporuje dva systémy vytěžování, kterými jsou Rossum a DOCU-X. Pro využití vytěžených informací je uživatel musí nejdříve ručně zkontrolovat a potvrdit jejich správnost. [5]

Po vyplnění všech potřebných údajů do systému je faktura připravena ke schválení vedoucím pracovníkem. Pokud bylo využito systému INSIO již při vytvoření objednávky, je možné nastavit, aby nebylo nutné znovu schvalovat faktury, které byly již jednou schváleny při vytváření dané objednávky. [5]

Schválený účetní doklad může být na konci celého procesu automaticky zapsán do vybraného účetního systému. INSIO DMS podporuje hned několik účetních systémů. Příkladem těchto systémů může být účetní systém Pohoda, HELIOS, Money S3 a další. Doklad také samozřejmě zůstane uložen v systému INSIO DMS pro následnou archivaci. [5]

Produkt INSIO DMS je možné zakoupit pouze jako předplatné, a to ve třech různých konfiguracích. V nejnižší konfiguraci stojí 3 900 Kč měsíčně a nabídne pět uživatelských

úctů do webového systému. Za každého dalšího uživatele je poté nutno připlatit 400 Kč měsíčně. V této konfiguraci systém nepodporuje vytěžování faktur pomocí OCR, vlastního e-mailového klienta a ani neposkytuje mobilní aplikaci. [6]

Nejdražší konfigurace systému nemá pevně stanovenou cenu a nabídka je vytvořena dané firmě přímo na míru. Nejdražší běžně dostupné předplatné stojí 9 900 Kč měsíčně a nabízí až dvacet uživatelských účtů. Toto dražší předplatné přidává dříve chybějící funkce jako OCR nebo mobilní aplikaci. Dále přibývá možnost podpory více jazyků systému a napojení na běžné účetní systémy. [6]

1.2.3 Porovnání

Při srovnání těchto dvou systémů s tím použitým v rámci této práce je jasné, že se systémy v zásadě snaží plnit podobnou činnost. Tato činnost však není úplně stejná a jsou v ní větší či menší rozdíly.

V prozkoumaných konkurenčních řešeních byl dáván velký důraz na vytěžování dat z e-mailové schránky, do které chodí účetní doklady. Pokud by byl tento sběrný e-mail kompromitován, mohl by být útočníkem zahlcen nevyžádanou poštou. Faktury z těchto nevyžádaných e-mailů by poté mohly být zaneseny do systému a zbytečně analyzovány pomocí OCR. Jelikož systém použitý v této práci umožňuje nahrávání nových účetních dokladů pouze registrovaným uživatelům, toto riziko v něm odpadá.

Konkurenční systémy také obsahovaly možnost napojení na existující účetní systémy. To může být velmi užitečná funkce, která zatím bohužel v použitém systému chybí. Konkurenční systémy naopak neobsahovaly možnost prohlížet dokumenty po jednotlivých měsících a zamykat dané měsíce, aby do nich nebylo možné již žádné doklady přidat. V konkurenčních systémech tento zámeček většinou vznikl na úrovni jednotlivých schválených dokumentů.

Společným bodem všech tří systémů je využití OCR pro automatické nalezení dat v účetním dokladu. Je tedy vidět, že se v dnešní době jedná o velmi důležitou funkci, která může výrazně zkrátit potřebný čas pro zadávání nových dokumentů. V rámci této práce je optické rozpoznávání znaků využito přímo na mobilním zařízení a za kontrolu údajů odpovídá uživatel, který doklad do systému nahrává. Díky tomu se nemůže stát, že se v systému nachází doklady, které ještě nebyly zkontrolovány.

Obě konkurenční řešení také uživatelům poskytovala možnost využití mobilní aplikace. V případě systému Wflow se jednalo pouze pro aplikaci pro nahrávání nových účetních dokladů. Aplikace Wflow však v podstatě neobsahuje jiné funkce a nejspíš již není aktivně vyvíjena, jelikož poslední aktualizaci dostala v roce 2019. [7]

O mobilní aplikaci INSIO Software nebylo možné najít příliš mnoho informací, ale zdá se, že umožňuje procházení a správu jednotlivých dokumentů. Aplikace INSIO naopak neobsahuje možnost nahrát fotografii účetního dokladu do systému. Tato aplikace byla naposledy aktualizována tento kalendářní rok. [8]

Z porovnání vyplývá, že i když se jedná o podobné systémy, není možné jednoduše určit, že by byl některý z nich přímo horší nebo lepší než ten druhý. Každý systém řeší danou problematiku trochu jiným způsobem a může být vhodný pro různé společnosti. Jak již bylo řečeno, systém použitý v rámci této práce byl vytvořen na míru pro danou společnost a tudíž je pro ni tím nejvhodnějším. Na druhou stranu je vidět, že i běžně dostupné systémy jsou na vysoké úrovni. Tyto systémy mohou být dobrou volbou pro společnosti, které žádný systém ještě nemají, a vývoj vlastního systému by byl pro ně příliš nákladný.

2 POUŽITÉ TECHNOLOGIE

V této kapitole jsou popsány technologie použité při tvorbě této diplomové práce. Jsou zde popsány nejen obecné technologie jako samotné Android SDK nebo použitý programovací jazyk, ale dostane se i na konkrétní technologie pro implementaci vkládání závislostí nebo tmavého režimu vzhledu aplikace.

2.1 Android SDK

Nejdůležitější použitou technologií při tvorbě této práce bylo samotné Android SDK. Android SDK poskytuje čtyři hlavní pilíře pro psaní aplikací. Těmito pilíři jsou aktivity, služby (anglicky services), tzv. „broadcast receivers“ a poskytovatele obsahu (anglicky content providers). Další velmi důležité prvky poskytované v Android SDK jsou fragmenty a prvky uživatelského rozhraní. [9]

Celá aplikace popsaná v rámci této práce vznikla s využitím pouze jedné aktivity. Aktivita je brána pouze jako vstupní bod do aplikace. To znamená, že veškeré obrazovky, se kterými se uživatel v aplikaci setká, jsou implementovány v rámci fragmentů. Tyto fragmenty se střídají v rámci jednoho okna aktivity. [9]

Při vývoji Android aplikací je také velmi důležité správně vybrat minimální podporovanou verzi operačního systému. Minimální verze jasně označuje nejnižší možnou verzi operačního systému, na kterém půjde aplikace nainstalovat. [9] Pro větší rozsah podporovaných zařízení aplikace vypracovaná v rámci této práce podporuje zařízení s minimální verzí Android 5.0 (API verze 21).

Neméně důležitou volbou při vývoji Android aplikací je volba cílové verze Android SDK. Jedná se o verzi, proti které je aplikace primárně vyvíjena. Výběrem této verze se vývojář zavazuje k dodržování všech pravidel a omezení, která tato verze Androidu přináší. Například pokud aplikace cílí na Android 6.0 nebo novější, tak se na některá oprávnění musí ptát uživatele přímo při běhu aplikace. Pokud chce vývojář distribuovat aplikaci v obchodě Google Play, musí daná aplikace cílit na jednu z posledních dvou nejnovějších verzí Android SDK. Aplikace vypracovaná v rámci této práce cílí na Android 11 (API 30), který je v době psaní této práce nejnovější verzí operačního systému Android. [10]

2.2 Kotlin

Druhou nejdůležitější technologií pro vytvoření této práce je samotný programovací jazyk Kotlin. Kotlin je programovací jazyk, pomocí kterého lze psát jakékoliv aplikace, které mohou být napsané v Javě. Kotlin je tedy možné použít nejen pro vývoj Android aplikací, ale také například pro vývoj serverových aplikací. [11]

Kotlin byl roce 2017 uznán společností Google jako oficiální jazyk pro vývoj Android aplikací a zařadil se tím vedle dosud kralujícího jazyka Java. Podpora společnosti Google znamenala, že pro psaní Android aplikací v jazyce Kotlin nebylo potřeba do vývojového prostředí nic doinstalovat. Po oznámení oficiální podpory získal Kotlin v komunitě vývojářů aplikací velkou popularitu. [12]

Dva roky po začátku podpory Kotlinu jako oficiálního jazyka pro vývoj Android aplikací zašla společnost Google v podpoře ještě dál. Na konferenci Google I/O v roce 2019 byl oznámen takzvaný „Kotlin-first“ přístup. To znamenalo, že všechny knihovny pro vývoj aplikací od společnosti Google budou poskytovat rozhraní pro zjednodušení práce v Kotlinu, oficiální dokumentace vždy bude obsahovat ukázky kódu v Kotlinu a že bude Google dále podporovat vývoj samotného jazyka a nástrojů pro práci. V době psaní této práce již Google dokonce vydal nové vývojářské nástroje pro tvorbu uživatelského rozhraní. Tyto nástroje se jmenují „Jetpack Compose“ a je možné je použít pouze v kombinaci s programovacím jazykem Kotlin. [13]

Jedním z důvodů vysoké adopce Kotlinu mezi vývojáři je to, že v rámci jednoho projektu je možné libovolně kombinovat kód napsaný v Kotlinu a v Javě. To znamená, že Kotlin bylo možné použít i v existujících projektech. Také to znamená, že aplikace psané v Kotlinu mohou využít široké nabídky existujících knihoven, které jsou napsané v Javě. Příkladem, že tato kombinace může fungovat, je samotný Google, který tvrdí, že více než 60 jeho aplikací obsahuje kód psaný v Kotlinu. Těmito aplikacemi jsou například Google Maps, Google Drive a Google Play. [13]

Dalšími funkcemi, které Kotlin oproti Javě přináší, jsou datové třídy, bezpečnost null hodnot nebo třeba takzvané zapečetěné třídy (anglicky sealed class). Datové třídy jsou třídy, které obsahují datové položky. Kompilátor jazyka Kotlin automaticky těmto třídám vygeneruje metody „equals“, „hashCode“ a další. [11]

Bezpečnost null hodnot je zajištěna tím, že každá proměnná musí explicitně definovat, zda může obsahovat hodnotu null, nebo nemůže. Proměnná, která hodnotu null obsahovat může, nemůže být bez kontroly uložena do druhé proměnné stejného typu, která ji obsahovat nesmí. Kód, který nerespektuje bezpečnost null hodnot, ani nebude zkompileován a nedochází tedy k nechvalně známým „NullPointerException“. To, že proměnná může obsahovat hodnotu null, se označuje pomocí otazníku za samotným datovým typem. [11]

Posledním popsáním rozdílem oproti jazyku Java jsou například dříve zmíněné zapečetěné třídy. Kotlin se samozřejmě od Javy liší v mnohých dalších oblastech (například syntaxe), ale cílem této práce není pokrytí všech rozdílů. Zapečetěné třídy se v jazyce Kotlin označují pomocí klíčového slova „sealed“, které se píše před samotnou třídu. Když je třída zapečetěná, znamená to, že má omezený počet tříd, které z ní mohou dědit. Tyto třídy musí být definovány ve stejném souboru jako samotná zapečetěná třída. Zapečetěné třídy jsou užitečné ve chvíli, kdy bychom v Javě použili výčetový typ, ale zároveň bychom potřebovali, aby obsahoval nějaká specifická data. [11]

2.2.1 Kotlin Coroutines

Kotlin Coroutines je volitelná knihovna pro programovací jazyk Kotlin. Tato knihovna umožňuje psaní asynchronního kódu stejně, jako kdyby se jednalo o kód synchronní. To znamená, že pokud na jednom řádku provedeme vzdálené API, můžeme na následujícím řádku s výsledkem pracovat. Díky Kotlin Coroutines se tedy kód zbaví zpětných volání po dokončení asynchronní operace a výrazně se zvýší jeho přehlednost. [14]

Kotlin Coroutines jsou založeny na pozastavitelných funkcích. Pozastavitelné funkce se v kódu označují pomocí klíčového slova „suspend“. Jak již název napovídá, jedná se o funkce, které je možné pozastavit ve chvíli, kdy je nutné počkat na výsledek asynchronní operace. Ve chvíli, kdy se daná operace dokončí, se pozastavená funkce znovu rozběhne z místa, kde byla pozastavena. Pozastavitelné funkce je možné volat pouze z dalších pozastavitelných funkcí. [14]

Přestože Kotlin Coroutines využívají na pozadí vlákna, neznámá to, že každé spuštění nové Coroutine spustí nové vlákno. Při spuštění si programátor může vybrat, který „dispečer“ bude zodpovědný za plánování dané Coroutine. Za běhu Coroutine je možné změnit aktuálního dispečera (a tím i vlákno) pomocí funkce „withContext“. Nejen díky tomuto přístupu jsou Kotlin Coroutines mnohem méně náročné na výkon než přímá práce

s vlákny. Je tedy možné spustit tisíce Coroutines najednou bez velkého dopadu na běh aplikace. [14]

Obrázek 1 ukazuje pokročilý způsob volání několika pozastavitelných funkcí současně. Nejdříve je spuštěna samotná Coroutine pomocí „viewModelScope.launch()“ a následně je zavoláno několik pozastavitelných funkcí. Pozastavitelné funkce jsou zavolány pomocí pomocné funkce „async“. To znamená, že se na tomto řádku nebude čekat na výsledek funkce a umožňuje to současné čekání na všech pět potřebných zdrojů. Pro získání samotných dat pro další práci je třeba zavolat funkci „await“ nad dříve uloženými asynchronními voláními. Pokud by nebylo vyžadováno současné načítání všech zdrojů, tak by vůbec nebylo potřeba používat volání funkcí „async“ a „await“ a stačilo by funkce volat standardním způsobem.

```
viewModelScope.launch(exceptionHandler) { this: CoroutineScope
    try {
        // Paralelní získání 5 zdrojů z internetu či databáze.
        val currenciesDeferred = async { currencyRepository.getCurrencies() }
        val categoriesDeferred = async { categoryRepository.getCategories() }
        val statusesDeferred = async { statusRepository.getStatuses() }
        val usersDeferred = async { userRepository.getUsers() }
        val permissionsDeferred = async { userRepository.getCurrentPermissions() }

        // Přečtení výsledků a práce s nimi.
        val currencies = currenciesDeferred.await()
        val categories = categoriesDeferred.await()
        val statuses = statusesDeferred.await()
        val users = usersDeferred.await()
        val permissions = permissionsDeferred.await()
    }
}
```

Obrázek 1 – Ukázka paralelního načítání zdrojů pomocí Kotlin Coroutines.

2.3 Koin

Koin je populární framework pro vkládání závislostí (anglicky dependency injection). Je primárně určen pro použití v kombinaci s programovacím jazykem Kotlin, ale je možné pomocí něj vkládat závislosti i do tříd napsaných v jazyce Java. [15]

Koin poskytuje jednoduché rozhraní založené na DSL (doménově specifický jazyk). Samozřejmě se jedná o interní DSL. To znamená, že je založené na existujícím jazyce. V tomto případě je tím jazykem Kotlin. Veškerá konfigurace pravidel pro vytváření závis-

lostí se tedy popisuje pomocí DSL na rozdíl od konkurenční knihovny „Dagger 2“, která využívá anotace a generování kódu. [16]

```
val loginModule = module { this: Module

    factory<LoginViewDataMapper> { LoginViewDataMapperImpl(get()) }

    viewModel { LoginViewModel(get(), get(), get()) }

    factory<LoginUseCase> { LoginUseCaseImpl(get(), get(), get(), get(), get()) }
}
```

Obrázek 2 – Ukázka definice pravidel pro vytváření závislostí pomocí Koin.

Na obrázku 2 je ukázán příklad definice modulu pro definici komponent, do kterých budou při vytvoření vloženy komponenty pomocí Koin. Modul je definován jednoduše pomocí klíčového slova „module“ a složených závorek. Všechny další definice, které se nacházejí mezi těmito složenými závorkami, jsou součástí tohoto modulu. [15]

V rámci modulu lze použít funkci „factory“, ve které je popsáno, jak vytvořit jednu závislost. Slovo „factory“ znamená, že v případě potřeby bude tato závislost vytvořena pokaždé znovu a její instance se nebude nikde zachovávat. Opakem „factory“ je funkce „single“, pomocí které lze definovat závislosti, kterých bude vytvořena pouze jediná instance. Další možností omezení vytváření nových instancí je funkce „viewModel“, která vytváří novou instanci třídy dle pravidel životního cyklu ViewModelů v Android aplikacích. [15]

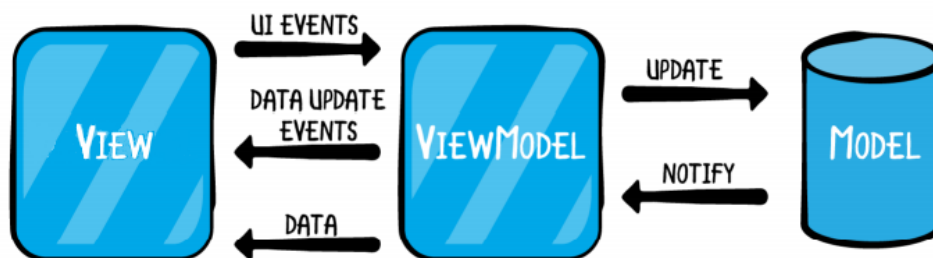
Může být žádoucí, aby byla závislost nebyla poskytována přímo, ale jako rozhraní, které implementuje. V tom případě stačí vyplnit datový typ za slovo „factory“ mezi špičaté závorky. Tento přístup je možné použít i v případě funkcí „single“ nebo „viewModel“. [15]

Poslední, co ukazuje obrázek 2, je získávání instancí tříd potřebných pro vytvoření právě popisované třídy. To se dělá pomocí funkce „get“, která automaticky zjistí datový typ, který je potřeba vložit. Také je možné datový typ specifikovat ručně mezi špičaté závorky anebo kvalifikátorem určit, která implementace dané třídy je požadována.

2.4 Model-View-ViewModel

Model-View-ViewModel, zkráceně MVVM, je architektonický návrhový vzor pro tvorbu aplikací. V rámci této aplikace byl tento návrhový vzor implementován pomocí knihovny Android Architecture Components od Google. Jedná se o knihovnu, která poskytne základní implementaci ViewModelu a stará se o jeho životní cyklus. [17]

Jak již název napovídá, návrhový vzor Model-View-ViewModel se skládá ze tří vrstev. Těmito vrstvami jsou model, view a ViewModel. Z pohledu interakce s aplikací je první vrstvou view. Jedná se o samotné uživatelské rozhraní, které má pouze tři hlavní cíle. Prvním cílem je zobrazovat data uživateli v pochopitelném formátu. Druhým cílem je čekat na uživatelské interakce a posílat informace o nich do ViewModelu. A posledním úkolem je sledovat samotný ViewModel a měnit svůj stav dle instrukcí. [17]



Obrázek 3 – Interakce mezi modelem, view a ViewModelem v architektuře MVVM (upraveno). [17]

Druhou vrstvou je ViewModel, který se chová jako prostředník mezi vrstvou view a modelem. ViewModel získává data z vrstvy modelu, spravuje svůj vlastní stav a předává instrukce do view vrstvy. Na rozdíl od jiných běžných architektur v sobě ViewModel nikdy nedrží žádnou referenci na view vrstvu. Tyto dvě vrstvy spolu vždy komunikují pomocí návrhového vzoru pozorovatel (observer). [17]

Poslední vrstvou této architektury je samotný model. Model zapouzdřuje datovou vrstvu aplikace do jednoho celku. Obsahuje tedy třídy, které načítají, ukládají či mění data v lokálním úložišti nebo na vzdáleném serveru. [17] V této aplikaci byla vrstva modelu implementována pomocí několika repozitářů.

2.5 Material design a tmavý režim

Design této aplikace byl implementován dle pravidel Material designu. Material design je soubor pravidel a pouček pro implementaci jednotného vzhledu všech aplikací pro platformu Android. [9]

Avšak všechny komponenty Material designu nejsou automaticky implementovány operačním systémem. Pro jejich použití je tedy nutné využít knihovny „Material Components“, která pochází přímo od společnosti Google. Tato knihovna implementuje velké množství komponent a zajišťuje jejich jednotné chování na všech podporovaných verzích operačního systému. To je velmi důležité pro zachování jednotného vzhledu aplikace pro všechny uživatele.

Při implementaci designu této aplikace bylo využito možnosti globálního stylování použitých komponent. To je výhodné, protože stačí požadovaný vzhled dané komponenty nastavit pouze jednou v globálním stylu aplikace a všechny komponenty daného typu budou automaticky dodržovat daný vzhled.

Material design nabízí i možnost implementovat do aplikace světlý a tmavý režim. Tato implementace je kompatibilní se systémovým nastavením tmavého režimu, které je dostupné od Androidu 10. [18] Tmavý režim byl implementován i v této aplikaci.

Při implementaci tmavého režimu je důležité si v globálním stylu správně nadefinovat barvy. Samozřejmě je potřeba změnit barvu textu a pozadí aplikace. Ale také je důležité změnit i další barvy aplikace, jelikož není dobré používat v tmavém režimu příliš satureované barvy. [19]

Jelikož Material design často pracuje se světlem a stíny, jsou stíny dalším problémem, který je potřeba nějakým způsobem vyřešit. Standardní stín by v tmavém režimu aplikace vůbec nebyl vidět. Z toho důvodu se v tmavém režimu aplikace nepoužívají stíny, ale takzvané překrytí. Vysvětlení je takové, že čím větší by byl stín ve světlém režimu, tím větší překrytí světlou barvou systém na daný prvek aplikuje. A to z toho důvodu, že čím blíže je prvek k pomyslnému zdroji světla, tím světlejší by měl být. [19]

2.6 GraphQL

Pro komunikaci se serverem bylo využito technologie GraphQL. GraphQL je v podstatě dotazovací jazyk pro API. Na rozdíl od REST rozhraní má GraphQL API pouze jeden endpoint a různé dotazy jsou prováděny právě za pomoci dotazovacího jazyka. [20]

Obrázek 4 ukazuje příklad jednoduchého GraphQL dotazu. Jedná se o dotaz na seznam kategorií v rámci této aplikace. Na prvním řádku je specifikováno, že se jedná o dotaz se jménem „GetCategories“. Dotaz je pojmenovaný, aby se dal provolat z Android aplikace. Dále je v dotazu specifikováno, že klient požaduje seznam kategorií v daném formátu.

Výhodou GraphQL je, že si klient určí, které části dat potřebuje, a server mu poté odešle data ve specifikovaném formátu. Díky tomu nedochází ke zbytečnému přenosu informací, které klient k ničemu nevyužije. [20]

Také se může stát, že klient potřebuje vnořené informace o některých objektech. V rámci této aplikace se jedná například o název kategorií, statusů a podobně v dotazu na seznam účtenek. V takovém případě klient v dotazu specifikuje, o jaké vnořené objekty (a v jaké podobě) má zájem, a dostane odpověď v daném tvaru. Díky tomu nemusí klient provádět dodatečné dotazy na tyto vnořené objekty, když mu stačilo pouze malé množství informací. [20]

| Dotaz | Odpověď |
|--|--|
| <pre>query GetCategories { categories { id, displayName, vat } }</pre> | <pre>{ "data": { "categories": [{ "__typename": "Category", "id": 1, "displayName": "General", "vat": "TAXABLE_CATEGORY" }, { "__typename": "Category", "id": 2, "displayName": "Traveling", "vat": "TAXABLE_CATEGORY" },], }, }</pre> |

Obrázek 4 – Ukázka GraphQL dotazu.

GraphQL je však pouze specifikace a existuje několik knihoven, které je možné využít pro jeho použití v aplikacích. V rámci této práce byl využit GraphQL klient „Apollo Android“. [20]

Jednou z možností této knihovny je využití normalizované cache paměti. Jedná se v podstatě o jednoduchou SQLite databázi, která udržuje všechna načtená data dle unikátního ID. Jelikož různé objekty v rámci této aplikace neměly jednoznačně unikátní identifikátory, bylo na začátek každého identifikátoru uloženo jméno typu daného objektu. Použití této lokální databáze umožnilo implementaci offline režimu celé aplikace. [21]

2.7 CameraX

Pro implementaci fotoaparátu uvnitř aplikace byla použita knihovna CameraX. Tato knihovna je součástí balíku knihoven „Android Jetpack“ od společnosti Google. [22]

CameraX na pozadí využívá existující rozhraní pro práci s fotoaparátem (Camera2 API), které zabaluje do jednoduššího a moderního rozhraní. Rozhraní CameraX zahrnuje automatické sledování životního cyklu aplikace pomocí přiděleného „LifecycleOwner“. LifecycleOwner může být například aktuální fragment nebo aktivita. [22]

Používání CameraX je zaměřeno na takzvané případy užití. Případy užití se vždy připojí k danému objektu fotoaparátu s daným životním cyklem. Poté je již možné tyto případy užití používat v kódu aplikace. [22]

V rámci této práce byly využity dva případy užití. Prvním případem je automaticky aktualizovaný náhled toho, co fotoaparát aktuálně snímá. Druhým použitým případem užití je samotné vyfocení fotografie a uložení do zařízení. Dalšími případy užití, které nebyly v rámci této práce použity, jsou například natáčení videa nebo analýza průběžných snímků z fotoaparátu. [22]

Implementace části uživatelského rozhraní, které je zodpovědné za práci s fotoaparátem, bylo v této práci implementováno jako samostatný Android modul. V tomto modulu je zapouzdřena celá funkcionality fotoaparátu včetně všech interakcí se samotnou knihovnou CameraX. Tento samostatný modul je tedy velmi jednoduché z projektu vyjmout a použít v projektu následujícím. Protože hlavní kód aplikace neví nic o přesné implementaci práce s fotoaparátem, pak by bylo velmi jednoduché v případě potřeby implementaci modulu vyměnit za jinou bez nutnosti změny kódu hlavní aplikace.

3 ROZPOZNÁNÍ INFORMACÍ

Součástí vypracované aplikace je automatické rozpoznání vybraných údajů na fotografiích účtenek či faktur pomocí OCR. Tato činnost se skládala ze dvou kroků. Prvním krokem bylo nalezení textu ve vybrané fotografii faktury. Druhým krokem bylo nalezení požadovaných informací v tomto textu.

Nalezení textu ve fotografiích bylo implementováno za pomoci tří různých knihoven, které budou popsány v následujících podkapitolách. Uživatel si za běhu aplikace může vybrat, která z těchto implementací mu nejvíce vyhovuje a produkuje nejlepší výsledky.

Druhým krokem cesty k nalezení údajů je již dříve zmíněné hledání požadovaných informací v textu, který byl extrahován z fotografie účtenky či faktury. Toto hledání informací bylo implementováno samostatně v rámci vypracované aplikace. Logika pro hledání informací v textu je sdílená pro všechny tři implementace a bude hlouběji popsána v kapitole 3.3.

3.1 Rozpoznání textu na zařízení

Knihovny využívající strojové učení pro rozpoznání textu ve fotografiích se dělí na dvě základní skupiny. První skupinou jsou knihovny, které provádějí samotné hledání přímo v mobilním zařízení. Knihovny z této skupiny budou popsány v rámci této kapitoly. Ve druhé skupině se naopak nachází knihovny, které tuto těžkou práci přenáší ze zařízení na server. Serverové implementace budou popsány v kapitole 3.2.

Modely strojového učení, které provádějí analýzu obrazu na mobilním zařízení, mají často k dispozici nižší výpočetní výkon než modely, které běží na serveru. Jsou tedy o něco jednodušší, aby byly pořád dostatečně rychlé a aby tolik nezatěžovaly samotné zařízení. [23]

Není ale možné jednotně říci, že modely vyhodnocované lokálně na zařízení jsou horší. Lokální vyhodnocení obrazu může mít své výhody. Výhodou je například rychlost a možnost použití bez internetového připojení. Pokud se obraz vyhodnocuje přímo na zařízení, tak celý proces nezpomaluje žádný přenos dat a záleží pouze na rychlosti daného modelu. Některé lokální modely lze i použít k analýze videa v reálném čase. [23]

3.1.1 Tesseract

První implementace OCR použitá v této aplikaci je Tesseract. Tesseract je multiplatformní OCR knihovna s otevřeným zdrojovým kódem. Je tedy možné ho používat zdarma dle podmínek licence Apache 2.0. [24]

Přestože je Tesseract multiplatformní nástroj, tak není jednoduše dostupný jako běžná knihovna Android aplikací. Tesseract je napsaný v jazyce C++ a pro použití v Android aplikaci by bylo nutné využítí Android NDK (native development kit). Z toho důvodu je výhodné využít knihovny třetích stran, které kód zkompilují a připraví rozhraní v jazyce Java. V rámci této práce bylo využito rozhraní připravené českou firmou Adaptech s.r.o., které se jmenuje „Tesseract4Android“.

Před tím, než je možné pomocí Tesseract hledat text ve fotografiích, je zapotřebí do úložiště telefonu nahrát takzvaná „Traineddata“. Jedná se o soubory s příponou „traineddata“, které obsahují nekomprimované komponenty potřebné pro OCR. [25]

V rámci této práce byla využita již naučená data poskytnutá společností Google. Tyto datové soubory jsou do vypracované aplikace staženy při prvním použití Tesseract OCR. A to z toho důvodu, že soubory pro podporu českého a anglického jazyka jsou dohromady až několik desítek megabytů velké. Bylo by zbytečné aplikaci o tolik zvětšovat pro všechny uživatele, když každý uživatel může využívat jinou implementaci OCR. Ve chvíli, kdy si uživatel vybere jinou implementaci než Tesseract, se tyto stažené soubory ze zařízení smažou, aby zbytečně nezabíraly místo.

Dokumentace pro Tesseract API je dostupná samozřejmě pro původní zdrojové kódy napsané v C++. Pro rozhraní v jazyce Java tedy dokumentace není, ale toto rozhraní obsahuje stejné metody jako původní rozhraní. Neobsahuje sice všechny metody, ale pro jednoduché použití to stačí a při implementaci tedy lze použít původní dokumentaci.

Samotné programové rozhraní je synchronní a může zpracovávat vždy pouze jeden obrázek najednou. V případě vypracované aplikace to znamená, že při zpracovávání PDF souborů bylo nutné vždy skenovat pouze jednu stránku a až po rozpoznání celého textu přejít na stránku další. V rámci Android aplikace by bylo nepřijatelné toto dlouhotrvající rozpoznávání textu provádět na hlavním vlákne. Protože Tesseract neposkytuje žádné asynchronní rozhraní, bylo nutné toto ošetřit přímo v aplikaci s pomocí Kotlin Coroutines.

3.1.2 ML Kit

Druhá implementace OCR použitá ve vypracované aplikaci je ML Kit. Jedná se o knihovnu umožňující využití strojového učení přímo na mobilním zařízení. Poskytuje rozhraní pro řešení mnoha běžných problémů jako například skenování čárových kódů, rozpoznání obličejů a také v této aplikaci využití rozpoznání textu. [26]

ML Kit je produkt společnosti Google a vznikl díky rozdělení původního produktu „ML Kit for Firebase“. Tento produkt byl rozdělen na dvě části. První částí je ML Kit, který umožňuje využívat možnosti strojového učení přímo na mobilním zařízení. Druhou částí je takzvaný „Firebase Machine Learning“, který se zaměřuje na cloudové řešení. [27]

ML Kit je možné v mobilních aplikacích používat zdarma. ML Kit byl přímo vytvořen pro použití na mobilních zařízeních a poskytuje tedy moderní rozhraní, které je dobře zdokumentováno. [26] Google dokonce poskytuje celé aplikace se vzorovou implementací a také takzvané „Codelabs“ (kódové laboratoře), ve kterých je možné se naučit toto SDK používat krok za krokem. [28]

Programové rozhraní ML Kit je asynchronní a lze jednoduše integrovat do aplikace za pomoci Kotlin Coroutines. ML Kit také umožňuje analyzovat několik obrázků současně a tím zrychlit analýzu několikastránkového PDF dokumentu. Použití v rámci mobilní aplikace je také velmi jednoduché, jelikož není potřeba trénovat vlastní neuronovou síť. Stejně jako Tesseract potřebuje ML Kit pro nalezení textu natrénovaný model. Tento model si však stáhne automaticky po instalaci aplikace po nastavení tohoto chování v souboru „AndroidManifest.xml“. [28]

3.2 Rozpoznání textu v cloudu

Opakem pro rozpoznávání textu přímo na mobilním zařízení je využití cloudového řešení. Vlastní cloudové řešení poskytují všichni tři dnešní technologičtí giganti: Google, Amazon a Microsoft. [29]

Cloudové služby poskytující rozpoznávání textu ve fotografiích často běží na specializovaném hardwaru s vysokým výkonem. To umožňuje využití větších, složitějších a potenciálně lepších modelů strojového učení. [29]

Výhodou cloudového řešení může být možnost nekupovat drahou licenci OCR produktu a pouze platit službu dle reálného využití. Další výhodou může být jednoduchá možnost

škálování výkonu a možností OCR pomocí zaplacení vyšší úrovně dané služby. Pro mobilní zařízení s menším úložištěm také může být výhodné, že není nutné, aby v něm byl uložen velký model strojového učení, a stačí pouze komunikovat s externím serverem. [29]

Naopak nevýhodou může být, že je potřeba mít aktivní připojení k internetu. Jelikož je pro analýzu potřeba odeslat na server fotografii, tak může být spotřeba dat pro mobilní připojení až příliš velká. Dále by mohlo být při zpracovávání citlivých dat nežádoucí, aby tato data opustila zařízení, na kterém se nachází.

3.2.1 Microsoft Computer Vision

Jediným reprezentantem cloudových technologií použitým ve vypracované aplikaci je Microsoft Computer Vision. Tato technologie je součástí skupiny Microsoft Cognitive Services. Jelikož se nejedná o lokální technologii, je možné ji použít z jakékoliv platformy. Komunikace mezi zařízením a serverem probíhá s využitím REST API přes protokol HTTP. Výsledek analýzy odešle API jako JSON objekt. [30]

Služba Microsoft Computer Vision lze použít zdarma s určitými omezeními. Prvním omezením je limit na 5 000 provedených transakcí za měsíc. A druhým limitem je četnost jednotlivých volání. Zdarma lze provést pouze 20 volání za minutu. Pro mnoho menších projektů nejsou tyto limity omezující a mohou tedy využít verzi zdarma. Ze stejného důvodu mohla být verze zdarma využita i v rámci této práce. [31]

Pokud by verze zdarma nebyla dostačující, samozřejmě existuje i placená varianta. V placené verzi neexistuje omezení na počet provedených transakcí měsíčně. Naopak by si provozovatel této služby určitě přál co největší využití, jelikož se platí za každých tisíc volání měsíčně. Cena za tisíc volání je jeden americký dolar, ale může být i nižší při využití milionů transakcí měsíčně. V placené verzi však stále existuje omezení na 10 volání za sekundu. Toto omezení je však nesrovnatelně menší než to ve verzi zdarma. [31]

V rámci vypracované aplikace byla pro komunikaci s OCR serverem využita knihovna „Cognitive Vision Android“. Jedná se o knihovnu přímo od společnosti Microsoft, která má ulehčit komunikaci s jejich API v rámci Android aplikací. Integrace této knihovny v rámci aplikace je poměrně jednoduchá, avšak celý proces komplikuje špatná až chybějící dokumentace.

Pro využití Microsoft Computer Vision je nutné upravit analyzované fotografie, aby splňovaly určitá kritéria. Obrázek může být pouze ve formátu JPEG, PNG nebo BMP. Maximální povolená velikost obrázku je 4 MB a zároveň může mít maximální rozlišení 4200x4200 pixelů. [32]

3.3 Vyhledání potřebných dat v textu

Předchozí kapitoly popisovaly použité způsoby pro získání textu z fotografie účtenky. Tato kapitola se zaměří na získání potřebných dat z takto nalezeného textu. Pro nalezení dat v textu nebyla použita žádná externí knihovna a hledání bylo vypracováno samostatně v rámci této práce. Data jsou v textu lokalizována díky pokročilému využití regulárních výrazů.

V získaném textu jsou vyhledávány následující údaje: celková cena, cena bez DPH, měna a datum splatnosti. Tyto údaje jsou vyhledávány nezávisle na sobě, jelikož neplatí žádná univerzální pravidla, která by umožnila spojení více hledání dohromady. Například měna se nenachází vždy vedle ceny a může být v rámci faktury na úplně jiném místě.

Při hledání data splatnosti faktury bylo pro úspěšné nalezení nutné vyřešit několik problémů. Datum lze psát mnoha různými způsoby a je nutné tedy kontrolovat co nejvíce různých formátů. V rámci této aplikace je podporováno 15 nejběžnějších zápisů dat, včetně několika specifických pro český a anglický jazyk.

Při hledání data však nestačí, že text odpovídá nějakému formátu, ale stále se musí jednat o validní datum. Řetězec „1. 0. 2021“ sice vypadá jako datum, ale validním datem rozhodně není – tudíž ho validátor nesmí přijmou jako datum, i když je ve správném formátu.

Dalším problémem, který bylo potřeba vyřešit, bylo, že v rámci jedné faktury může být uvedeno více dat (například datum vystavení, datum zdanitelného plnění a podobně). Z těchto dat je potřeba vybrat to správné pro tuto aplikaci. Datum by bylo možné vybírat například dle polohy v samotném dokumentu – zda se nachází nahoře nebo dole. Data lze také porovnávat chronologicky a vybrat to nejstarší či naopak to nejnovější. V rámci vypracované aplikace je jako výsledné datum vybráno to nejstarší, které se na faktuře nachází.

Ale ani to nejstarší datum by nemuselo být vždy správné, jelikož některé země data zadávají ve stejném formátu, ale v jiném pořadí významu (např. prohozen měsíc a den).

Tudíž stroj často nedokáže poznat, zda je datum ve formátu „DD/MM/YYYY“ nebo „MM/DD/YYYY“. Pokud je číslo dne vyšší než 12, tak je to jednoznačné, ale někdy tomu tak není a obě data by mohla být validní.

Také při hledání měny bylo nutné řešit hned několik „problémů“. Měnu je možné zapsat ve dvou různých reprezentacích. První možností je výpis kódu měny – například CZK nebo USD. Druhý zápis obsahuje samotný symbol, který tuto měnu reprezentuje – například znak amerického dolaru nebo znaky reprezentující českou korunu (Kč). V textu je tudíž nutné vyhledávat měnu v obou těchto formátech.

Jelikož je na světě velmi velké množství různých měn, je lepší nevyhledávat v textu úplně všechny. Prvním důvodem je, že by vyhledávání všech různých měn v textu trvalo několiknásobně déle než vyhledání několika měn, které zná systém, se kterým aplikace komunikuje. A druhým důvodem je, že je úplně zbytečné najít v textu měnu, kterou server nezná, jelikož stejně nebude možné ji k dané účtence později přiřadit. Z toho důvodu si aplikace před prvním pokusem o vyhledání měn v textu nejprve načte seznam měn, které server umí zpracovat.

Měna může být někdy zapsána před částkou nebo i za částkou. Není tedy možné měnu hledat pouze za samotnou cenou. Znak měny se také může nacházet přímo vedle částky – tudíž bez oddělení mezerou (například „\$100“). Z toho důvodu není možné spoléhat ani na to, že bude měna v textu figurovat jako samostatné slovo, jako je například datum.

Jelikož některé implementace OCR měly problém s diakritikou, byla při hledání měny implementována jednoduchá optimalizace. A to taková, že se při hledání měny kontrolovaly i varianty bez háčků a čárek. Tudíž text „Kc“ bude rozpoznán jako označení pro českou korunu.

Stejně jako dat, i měn může být na jedné účtence vypsáno hned několik. Například nejmenovaný obchodní řetězec na účtence vytiskne i kurz české koruny oproti euru. To znamená, že na takovéto účtence jsou při analýze nalezeny měny obě. Tudíž jako výsledná měna, která bude předvyplněna do formuláře pro přidání nové účtenky, je vybrána ta, která se na účtence nachází nejčastěji.

Před pokusem o nalezení ceny účtenky bylo potřeba definovat, jak vlastně taková cena vypadá, jelikož různých čísel se může na faktuře nacházet mnoho různých druhů. V rámci této práce byla tedy cena definována jako číslo, které má vždy dvě desetinná místa.

V této práci bylo potřeba na účtence najít dva různé údaje o ceně – celkovou cenu a cenu bez DPH. Jako tyto údaje byla vybrána nejvyšší a druhá nejvyšší nalezená cena na účtence. Samozřejmě se nemusí vždy jednat o správnou hodnotu, jelikož na účtence může být indikována i určitá sleva a nebo cena nejdražší položky může být vyšší než celková cena bez DPH. I přesto bylo toto řešení vybráno, jelikož ve většině případů určí aplikace cenu správně.

Jinou možností identifikace celkové ceny by mohlo být hledání ceny, která se na účtence nachází co nejnižší. To by ale mohly zhatit dříve zmíněné kurzy, které by mohly být vypsány pod celkovou částkou.

I u ceny existují různé možnosti zápisu. Některé země používají desetinnou čárku, jiné zase tečku. Liší se také oddělovače tisíců. V českém zápisu může být tímto oddělovačem mezera, ale v jiném to může být právě dříve zmíněná čárka.

Ve výsledné implementaci jsou při hledání ceny nejprve hledáni takzvaní kandidáti na to, aby se v nich nacházela cena. Kandidáti jsou hledáni s využitím regulárních výrazů. Po jejich nalezení se validuje, že nalezená cena neobsahuje různé oddělovače tisíců. Po úspěšné validaci se text složený z celé i desetinné části čísla převede na reprezentaci čísla v programovacím jazyce.

Aby bylo převedené číslo přijato, musí být při jeho načítání přečtený celý kandidátní text. Jelikož některé implementace OCR občas načtou tečku jako čárku a opačně, nevaliduje se, zda číslo používá jiný oddělovač tisíců než oddělovač desetinné části. Pokud by bylo důležitější zvýšení přesnosti na úkor četnosti nalezení správné hodnoty, bylo by velmi jednoduché do kódu takovou validaci přidat.

3.4 Analýza výsledků

Po implementaci tří výše zmíněných OCR řešení byla tato řešení podrobena analýze. Pro analýzu bylo využito 182 vstupních souborů účtenek a faktur. Přibližně polovina zdrojových souborů byla ve formátu PDF a druhá polovina byla ve standardním formátu fotografie JPEG. Přesně vyčísleno se jednalo o 95 PDF souborů a 87 fotografií.

U těchto souborů byly ručně nalezeny hledané údaje, kterými jsou cena bez DPH, celková cena, datum a měna. Tyto údaje byly poté přiřazeny k daným souborům a strojově porovnávány s automaticky nalezenými údaji.

Pro možnost analýzy toho, zda nedošlo při automatickém hledání hodnot k chybě v pouze posledním kroku, byly při analýze výsledků kontrolovány všechny hodnoty nalezené ve fotografii účtenky. To znamená, že například kromě jedné algoritmem vybrané hodnoty ceny bez DPH byly kontrolovány i všechny ostatní nalezené ceny. Díky tomuto postupu bylo možné zjistit, kolik hodnot bylo algoritmem nalezeno, ale nakonec nebylo vybráno.

Testování jednotlivých řešení OCR probíhalo na mobilním telefonu „OnePlus 8 Pro“ s operačním systémem Android 11. Přestože se jedná o velmi výkonný telefon, jedna z implementací OCR byla příliš pomalá pro možnost nasazení v praxi. Výkon telefonu by však neměl mít žádný vliv na přesnost výsledků – pouze na rychlost analýzy obrázků.

Tabulka 2 ukazuje naměřené hodnoty při testování jednotlivých implementací. V každém řádku tabulky se nacházejí naměřené výsledky každé implementace OCR (včetně následného nalezení dat v textu). Jednotlivé sloupce pak ukazují počet správně nalezených hodnot daného typu (například datum).

Tabulka 2 – Porovnání výsledků jednotlivých implementací OCR.
(číslo bez jednotky udává počet správně nalezených hodnot ze 182)

| | Cena bez DPH | Celková cena | Měna | Datum | Prům. doba zpracování | Celková doba zpracování |
|----------------------------|-------------------------|-------------------------|------------------|------------------|----------------------------------|------------------------------------|
| ML Kit | 88 (48,35 %) | 125 (68,68 %) | 156 (85,71 %) | 162 (89,01 %) | 0,39 s | 70,47 s (1,17 min) |
| Tesseract | 49 (26,92 %) | 65 (35,71 %) | 143 (78,57 %) | 142 (78,02 %) | 19,08 s | 3 472,23 s (57,87 min) |
| Computer Vision | 69 (37,91 %) | 109 (59,89 %) | 163 (89,56 %) | 117 (64,29 %) | 1,22 s | 222,69 s (3,71 min) |

V tabulce 2 je tedy vidět, že kromě správného nalezení měny byla hledaná data vždy nejlépe nalezena pomocí implementace ML Kit od Google. Tato implementace byla také bezkonkurenčně nejrychlejší a je schopná pracovat bez přístupu k internetu. Po vypočítání poměru všech nalezených a hledaných hodnot vychází, že tato implementace našla správné údaje v 72,94 % případů.

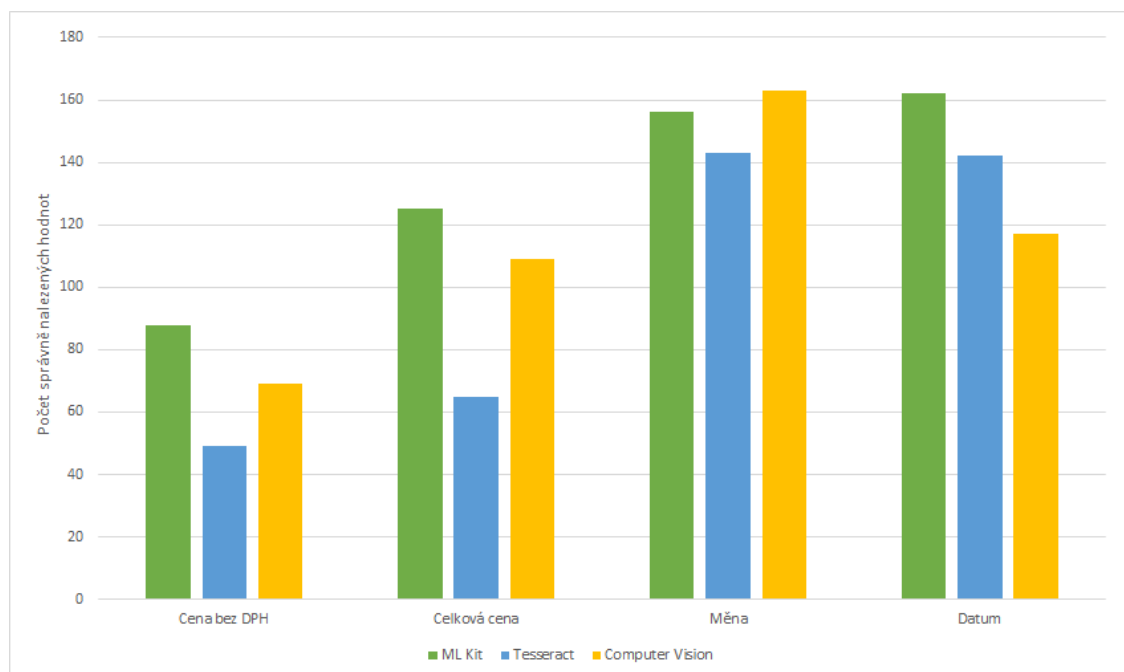
Druhou nejúspěšnější implementací OCR byla Microsoft Computer Vision. Naměřené hodnoty pro tuto implementaci se nacházejí na posledním řádku tabulky 2. Tabulka 2 ukazuje, že tato implementace byla nejlepší v nacházení měny ve fotografiích. Celková

přesnost nalezení hodnot u této implementace je 62,91 % a to ji staví na celkové druhé místo v přesnosti.

U Microsoft Computer Vision je doba zpracování jedné účtenky vyšší než u předchozí implementace, ale pořadí není příliš dlouhá. Jelikož Microsoft Computer Vision provádí samotné rozpoznání znaků v cloudu, tak rychlost zpracování závisí i na rychlosti internetu na daném zařízení. Při provádění těchto testů bylo využito internetové připojení s rychlostí kolem 60 Mb/s.

Poslední a dle testů nejhorší implementací je Tesseract. Dle dat z tabulky 2 má Tesseract nejhorší přesnost nalezení dat ve všech kategoriích kromě hledání data. Při hledání data se Tesseract umístil na druhém místě a byl tedy přesnější než Microsoft Computer Vision. Celková přesnost nalezení hodnot pomocí Tesseractu je 54,81 %.

Analýza fotografií pomocí Tesseractu byla při testování velmi pomalá. Všechny testy běžely u této implementace necelou hodinu, což je mnohem více než u zbývajících dvou implementací, kde testy běžely maximálně jednotky minut. Zpracování jedné fotografie trvalo u Tesseractu průměrně 19 sekund. Takto dlouhý čas čekání je pro použití v mobilní aplikaci nepřijatelný.



Obrázek 5 – Počty správně předvyplněných hodnot nalezených pomocí OCR.

Data z tabulky 2 přehledně vizualizuje obrázek 5. Na tomto grafu je vidět, že přesnost implementace ML Kit je téměř pokaždé nejvyšší. A pokud nejvyšší není, tak je pořád velmi vysoká.

Analýzou výsledků tedy bylo zjištěno, která implementace produkuje nejlepší výsledky. Druhým cílem této analýzy bylo zjistit, která implementace má nejvyšší možný potenciál.

Pro zjištění potenciálu nalezení správných hodnot byl upraven výstup z OCR takovým způsobem, že algoritmus vrátil všechny hodnoty daného typu v jednoduchém seznamu a nevybíral tedy jednu hodnotu, kterou uznal za správnou. Pro dosažení tohoto stavu byl jednoduše vynechán poslední krok hledání dat, který z nalezených údajů vybíral ten správný pomocí dříve popsaných pravidel.

Tabulka 3 tedy ukazuje počty hodnot, které byly nalezeny v textu rozpoznaném pomocí OCR. Oproti správně nalezeným a vybraným hodnotám tedy tato čísla obsahují i případy, kdy správná hodnota byla nalezena, ale z nějakého důvodu nebyla algoritmem vybrána jako ta pravá.

Tabulka 3 – Počty správně nalezených hodnot ve fotografiích, které však nemusely být vybrány. (číslo bez jednotky udává počet správně nalezených hodnot ze 182)

| | Cena bez DPH | Celková cena | Měna | Datum | Celková přesnost |
|----------------------------|-------------------------|-------------------------|------------------|------------------|-----------------------------|
| ML Kit | 152 (83,52 %) | 167 (91,76 %) | 156 (85,71 %) | 176 (96,70 %) | 89,42 % |
| Tesseract | 117 (64,29 %) | 145 (79,67 %) | 143 (78,57 %) | 157 (86,26 %) | 77,20 % |
| Computer Vision | 130 (71,43 %) | 138 (75,82 %) | 164 (90,11 %) | 132 (72,53 %) | 77,47 % |

Při porovnání těchto dat s daty z tabulky 2 je vidět, že téměř ve všech případech se alespoň jednou stalo, že správná hodnota sice byla nalezena, ale nebyla vybrána.

Tento případ nejčastěji nastával při hledání cen. Úplně nejčastěji k tomu docházelo u hledání ceny bez DPH. To znamená, že pro výrazné zlepšení počtu správně vybraných hodnot by stačilo vylepšit algoritmus výběru správné ceny ze seznamu nalezených hodnot.

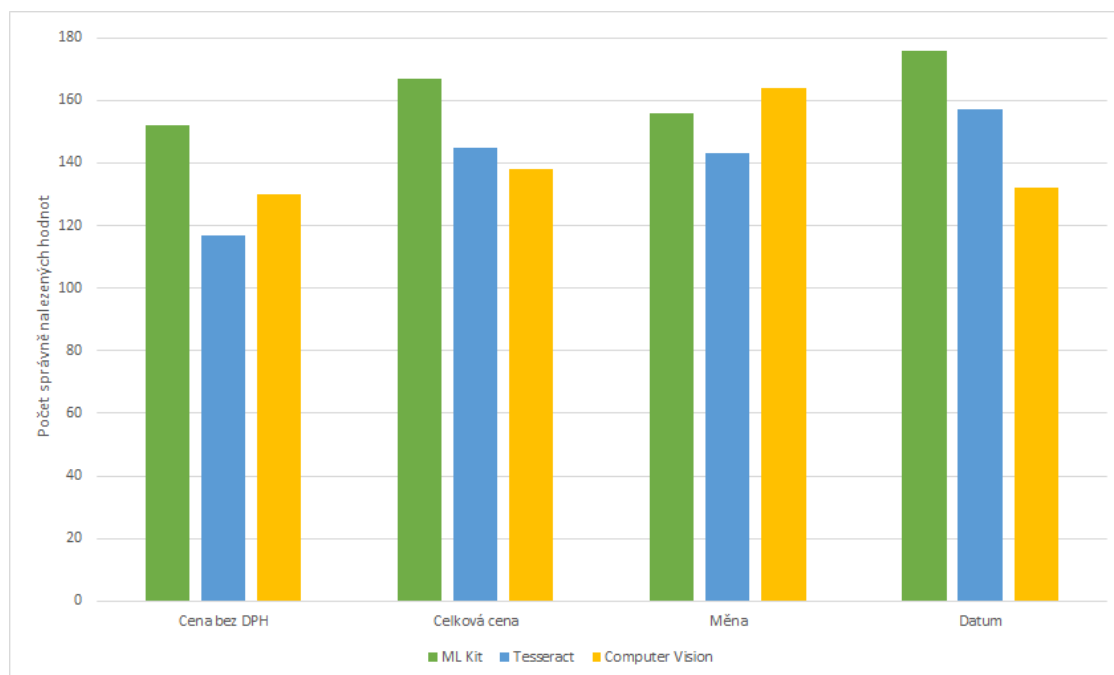
Automatické hledání ceny bez DPH však není příliš jednoduchý úkol, protože v množství cen je téměř nemožné bez kontextu určit, která z těchto cen je cenou bez DPH. Jelikož se na faktuře může nacházet více produktů, může se stát, že cena bez DPH není druhá

a třeba ani třetí nejvyšší cena na této faktuře. Existence několika sazeb DPH, které se mohou v rámci jedné faktury libovolně kombinovat, také nepomáhá schopnosti automaticky nalézt správný údaj. Na některých fakturách se samotná cena bez DPH dokonce ani neuvádí a člověk si ji musí vypočítat ručně z celkové ceny a výše DPH.

Nalezení správné hodnoty, která později není vybrána algoritmem jako ta pravá, se stává i při hledání měny či data. Četnost těchto případů je ale mnohem nižší a dle mého názoru je pro tyto položky přesnost výběru dostatečná. Pokud by však dostatečná nebyla, šla by o něco zvýšit propracovanějším algoritmem.

Dalším možným řešením toho, že je někdy vybrána špatná hodnota i ve chvíli, kdy byla nalezena ta správná, je, že by bylo možné neřešit tento problém algoritmem, ale přesunout ho zpět na samotného uživatele aplikace. Uživateli by se tedy mohl zobrazit seznam všech nalezených cen a on by z něj již mohl jednoduše vybrat celkovou cenu, cenu bez DPH a podobně. Toto řešení by však na uživatele přeneslo činnost, kterou by za něj měl mohla řešit aplikace automaticky.

Hodnoty v tabulce 3 opakují to, co bylo zjištěno původní analýzou. Nejčastěji se správné hodnoty daří nacházet implementaci využívající ML Kit. Tato implementace nachází správné hodnoty nejčastěji ve všech případech kromě hledání měny. Celkově bylo pomocí této implementace nalezeno až 89,42 % hodnot.



Obrázek 6 – Počty správných hodnot nalezených pomocí OCR (ne vždy vybraných).

Stejně jako v původní analýze se i tentokrát na pomyslném druhém místě nachází implementace využívající Microsoft Computer Vision s celkovou přesností 77,47 %. Tentokrát však není rozdíl mezi druhým a třetím místem příliš velký, jelikož se na třetím místě nachází Tesseract s celkovou přesností 77,20 %.

Data z tabulky 3 byla vizualizována do sloupcového grafu, který ukazuje obrázek 6. Na tomto obrázku je vidět, že počet správně nalezených hodnot je ve všech kategoriích hledaných hodnot velmi vysoký. Zároveň také ukazuje, že přesnost jednotlivých implementací je poměrně podobná.

Výsledkem analýzy přesnosti jednotlivých implementací tedy je to, že nejvyšší přesnost výsledků vykazuje implementace využívající ML Kit od společnosti Google. Tato implementace je velmi přesná a přitom je ze všech testovaných implementací nejrychlejší. Využití knihovny ML Kit bych tedy doporučil pro použití v mobilních aplikacích pro její přesnost a rychlost. Také bych ji však doporučil i jako vývojář pro její moderní API, dobrou dokumentaci a jednoduché použití.

Druhým výstupem této analýzy je zjištění, že algoritmus pro určení správné ceny bez DPH ze seznamu nalezených cen by bylo vhodné vylepšit. Tím by bylo možné výrazně zvýšit počet správně vyplněných hodnot, jelikož dané hodnoty jsou v textu nalezeny, ale poté nejsou vybrány jako správné. Hledání správné ceny bez DPH však není tak přímočaré jako hledání správné měny a možná by bylo vhodné tento úkol prozatím ponechat na uživateli a dát mu na výběr z nalezených hodnot.

4 POPIS APLIKACE

V této kapitole je popsána vytvořená aplikace. Jak již bylo řečeno, jedná se o mobilní aplikaci určenou pro platformu Android. Tato aplikace je určena primárně řadovým zaměstnancům firmy, ale obsahuje i skryté funkce. Tyto skryté funkce využívají vedoucí zaměstnanci firmy a účetní.

Řadoví zaměstnanci budou tuto aplikaci využívat pro ukládání firemních účtenek a faktur, které v průběhu své práce získali. Své uložené účtenky si mohou prohlédnout v měsíčních výpisech. Tyto účtenky mohou zaměstnanci fotit přímo v aplikaci a nemusejí používat systémový fotoaparát. Po vyfocení účtenky proběhne automatické nalezení textu a poté vyhledání důležitých údajů. Díky automatickému načtení údajů nebude muset zaměstnanec před nahráním účtenky vyplňovat například cenu nebo datum a celý proces se tím výrazně urychlí.

Běžní zaměstnanci si také mohou v aplikaci změnit své přihlašovací údaje a další nastavení aplikace. Těmito nastaveními jsou různé možnosti nastavení tmavého vzhledu aplikace a také různé implementace OCR. Uživatel si může nastavit, zda chce, aby tmavý režim dodržoval nastavení operačního systému, následoval spořič baterie nebo byl vždy zapnutý, nebo vypnutý. Všechna tato nastavení tmavého režimu jsou dostupná i ve chvíli, kdy operační systém daného zařízení nepodporuje tmavý režim celého systému. Všichni zaměstnanci si také mohou vybrat implementaci automatického rozpoznávání údajů v účtenkách nebo ho dokonce úplně vypnout.

Účetní může dělat vše, co má povolený řadový zaměstnanec, ale navíc vidí v seznamu účtenek všechny nahrané účtenky. Účetní nemůže upravovat cizí účtenky, pouze si je může zobrazit. Pro účetní je nejdůležitějším údajem unikátní číslo účtenky, které se účtenkám přidělí po uzavření daného měsíce.

Vedoucí zaměstnanec neboli administrátor také vidí všechny nahrané účtenky, ale oproti účetní může administrátor všechny účtenky také spravovat. To znamená, že schvaluje účtenky ostatních zaměstnanců, upravuje již schválené účtenky a účetně zavírá měsíce.

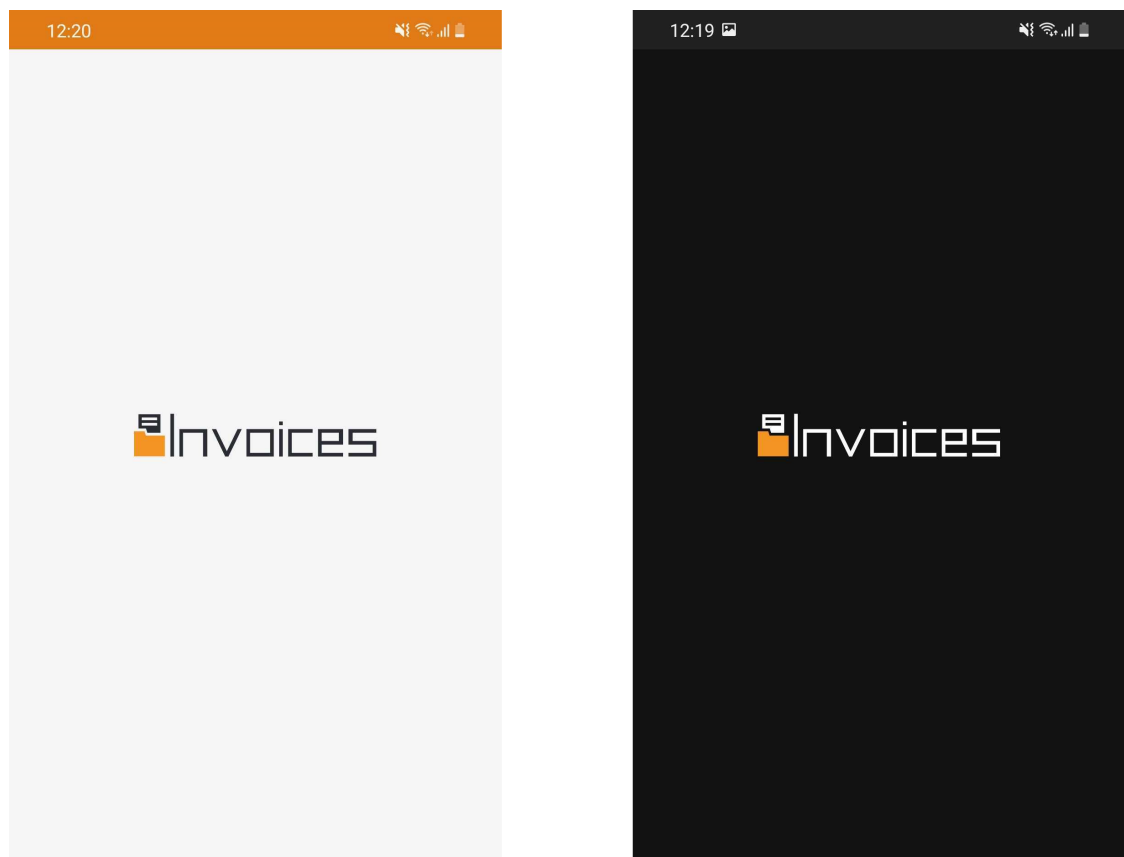
V následujících podkapitolách budou podrobně popsány jednotlivé obrazovky, které se nacházejí v této aplikaci.

4.1 Spouštěcí obrazovka

První, co uživatel po spuštění aplikace uvidí, je takzvaná spouštěcí obrazovka (anglicky splash screen). Tato obrazovka se v aplikacích ukazuje během načítání první reálné obrazovky a většinou neobsahuje žádné interaktivní prvky.

Při implementaci spouštěcí obrazovky pro Android aplikace je důležité, aby samotné vykreslování spouštěcí obrazovky nezpomalovalo start aplikace. Nejlepší je tedy nastavit spouštěcí obrazovku jako obrázek složený z několika vrstev. Tento obrázek je potřeba nastavit jako pozadí okna aplikace v globálním stylu. Po načtení aplikace je možné tento styl nahradit stylem, který obsahuje normální pozadí.

Pokud by byla spouštěcí obrazovka vytvořena pomocí standardních komponent uživatelského rozhraní, tak by před jejím vykreslením byla vidět obrazovka s výchozím bílým pozadím a až po načtení aplikace by se zobrazila falešná spouštěcí obrazovka. To je špatné řešení, protože aby uživatel vůbec stihl spatřit tuto falešnou spouštěcí obrazovku, tak by aplikace musela sama sebe na vteřinu pozastavit a tím zpomalit celé spuštění aplikace.

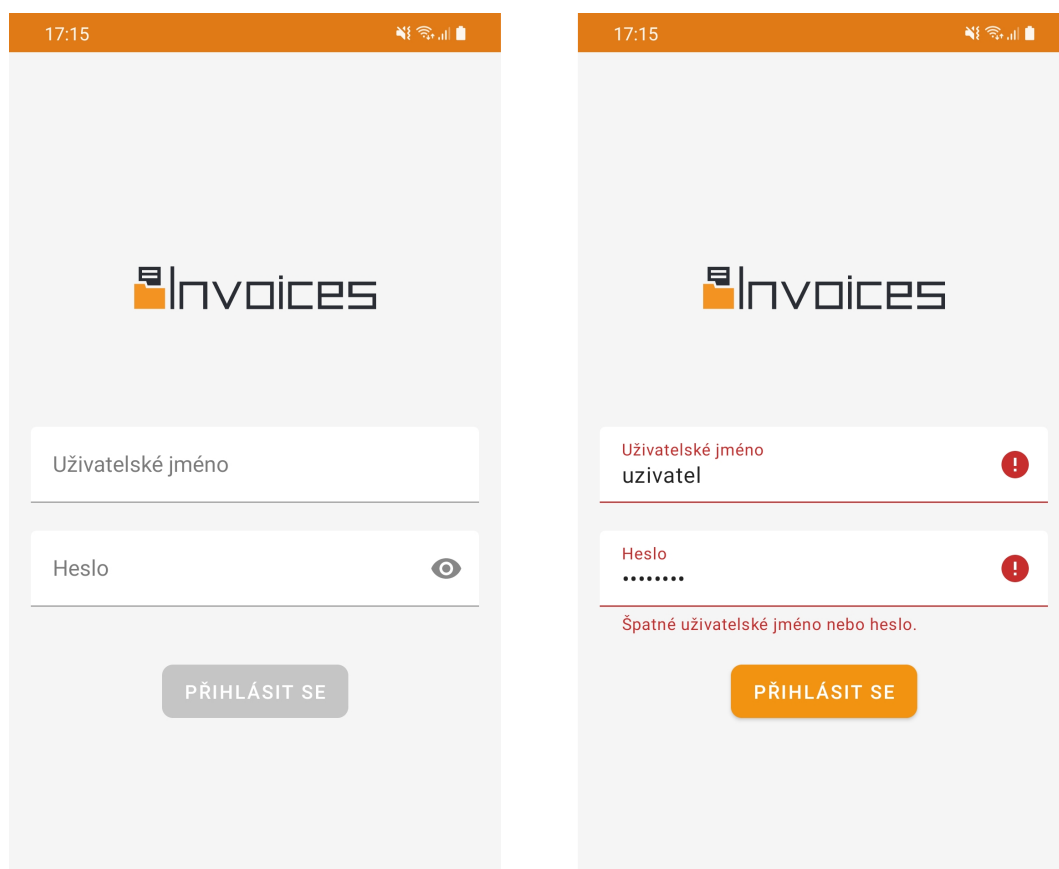


Obrázek 7 – Spouštěcí obrazovka ve světlém a tmavém režimu.

Jak ukazuje obrázek 7, v této aplikaci spouštěcí obrazovka obsahuje barevné pozadí a logo aplikace. Aby spouštěcí obrazovka neoslnila uživatele, kteří mají celý operační systém v tmavém režimu, je důležité, aby i spouštěcí obrazovka respektovala toto nastavení. Není tedy vhodné, aby byla spouštěcí obrazovka pouze tmavá nebo pouze světlá, protože by svým vzhledem nezapadala do zbytku aplikace. Z toho důvodu má v této aplikaci spouštěcí obrazovka bílé pozadí ve světlém režimu a černé pozadí v tmavém režimu. Pokud by ale byla upravena pouze barva pozadí, pak by se logo aplikace mohlo stát nečitelným. Z tohoto důvodu obsahuje aplikace logo ve verzi pro světlý i tmavý podklad.

4.2 Přihlášení

První obrazovka, kterou uživatel uvidí po zapnutí aplikace, je přihlašovací obrazovka. Pokud uživatel nebyl dříve přihlášen, tak se ukáže hned po skrytí spouštěcí obrazovky. Na přihlašovací obrazovce je znovu zobrazeno logo aplikace, aby uživatel věděl, do kterého systému se přihlašuje. I toto logo mění svou barvu, aby bylo viditelné jak na světlém, tak tmavém podkladu.



Obrázek 8 – Přihlašovací obrazovka a chybová hláška.

Pod logem se nachází samotný přihlašovací formulář. Přihlašovací formulář se skládá ze dvou textových polí a jednoho tlačítka. První textové pole vyžaduje zadání přihlašovacího jména. Přihlašovací jméno může být jakýkoliv text a nemusí se tedy jednat pouze o e-mail. Druhé textové pole je určené pro zadání hesla. Textové pole pro heslo samozřejmě nezobrazuje napsané heslo na obrazovce, ale nahradí jednotlivé znaky tečkami. Pokud si uživatel není jistý tím, co do textového pole vyplnil, nemusí celé heslo smazat a psát znovu. A to proto, že může kliknout na ikonu oka v pravé části textového pole a tím heslo zobrazí. Kliknutím na to samé tlačítko může po úpravě hesla napsaný text znovu skrýt.

Poslední viditelný prvek na přihlašovací obrazovce je tlačítko „Přihlásit se“. Toto tlačítko má při načtení této obrazovky zakázané veškeré interakce, jelikož nejsou zadány žádné údaje. Po vyplnění přihlašovacího jména a hesla se tlačítko zpřístupní a změní barvu na oranžovou. Po kliknutí toto tlačítko zmizí a místo něj se zobrazí kruhový načítací symbol.

Při případné chybě při přihlašování se textová pole zabarví do červena a pod textovým polem pro vyplnění hesla se zobrazí chybová hláška. Tato hláška skutečně odpovídá chybě, která nastala, a není pouze obecná. Většinou uživateli sdělí, že zadané přihlašovací údaje jsou špatné nebo že telefon není připojen k internetu. Tato chybová hláška zmizí ve chvíli, kdy uživatel začne znovu psát přihlašovací jméno nebo heslo.

Pokud uživatel nemá v systému vytvořený svůj účet, tak se na této obrazovce nemůže registrovat. A to z toho důvodu, že se jedná o uzavřený systém a firma si nepřeje, aby se mohl do tohoto systému přihlásit kdokoliv. Pokud do firmy přijde nový zaměstnanec a potřebuje účet, musí požádat svého vedoucího. Vedoucí zaměstnanec mu účet jednoduše vytvoří ve webovém rozhraní. Poté se uživatel může přihlásit v mobilní aplikaci a změnit si heslo.

4.3 Seznam účtenek a měsíční souhrn

Ihned po úspěšném přihlášení se uživateli otevře obrazovka, která obsahuje seznam účtenek a faktur. Tento stav je ale pouze jedním ze dvou režimů této obrazovky. Nahoře na této obrazovce se nachází oranžové pole (nebo šedé ve tmavém režimu), které obsahuje sdílené ovládací prvky pro oba režimy této obrazovky.

Prvním ovládacím prvkem v této oblasti je přepínač mezi režimy této obrazovky. Jedná se o přepínací tlačítko, které lze přepnout z režimu „Položky“ do režimu „Souhrn“ a zase zpátky. V režimu „Položky“ tato obrazovka ukazuje seznam účtenek, jak bude popsáno dále v kapitole 4.3.1. V režimu „Souhrn“ tato obrazovka ukazuje souhrnná data dle kategorií za vybraný měsíc. Režim „Souhrn“ bude podrobně popsán v kapitole 4.3.3.

V této sekci nahoře na obrazovce se také nachází aktuálně vybraný měsíc, pro který je zobrazen seznam účtenek nebo měsíční souhrn. Při spuštění aplikace je vybraný měsíc ten, ve kterém se aktuálně nachází uživatel, když aplikaci používá. Tento vybraný měsíc jde změnit pomocí ikon, které indikují posun o jeden měsíc do minulosti nebo do budoucnosti. Tyto ikony se nachází hned vedle výpisu aktuálně vybraného měsíce.

Druhou možností, jak změnit vybraný měsíc, je kliknout na text aktuálně vybraného měsíce. Po kliknutí se přes celou obrazovku zobrazí kalendář, ve kterém je vyznačen aktuálně zvolený měsíc jako vybraný rozsah. V tomto kalendáři si uživatel může vybrat jakýkoliv měsíc kliknutím na den v tomto měsíci. Po kliknutí na jeden den se v kalendáři automaticky vybere celý měsíc, protože server zatím nenabízí možnost filtrovat účtenky po jednotlivých dnech. V případě, že uživatel nechce měsíc hledat pomocí kalendáře, může v rámci kalendáře kliknout na ikonu tužky a tím otevřít možnost napsat datum do textového pole. Pro aplikování vybraného data je potřeba výběr potvrdit pomocí tlačítka „Uložit“ vpravo nahoře.

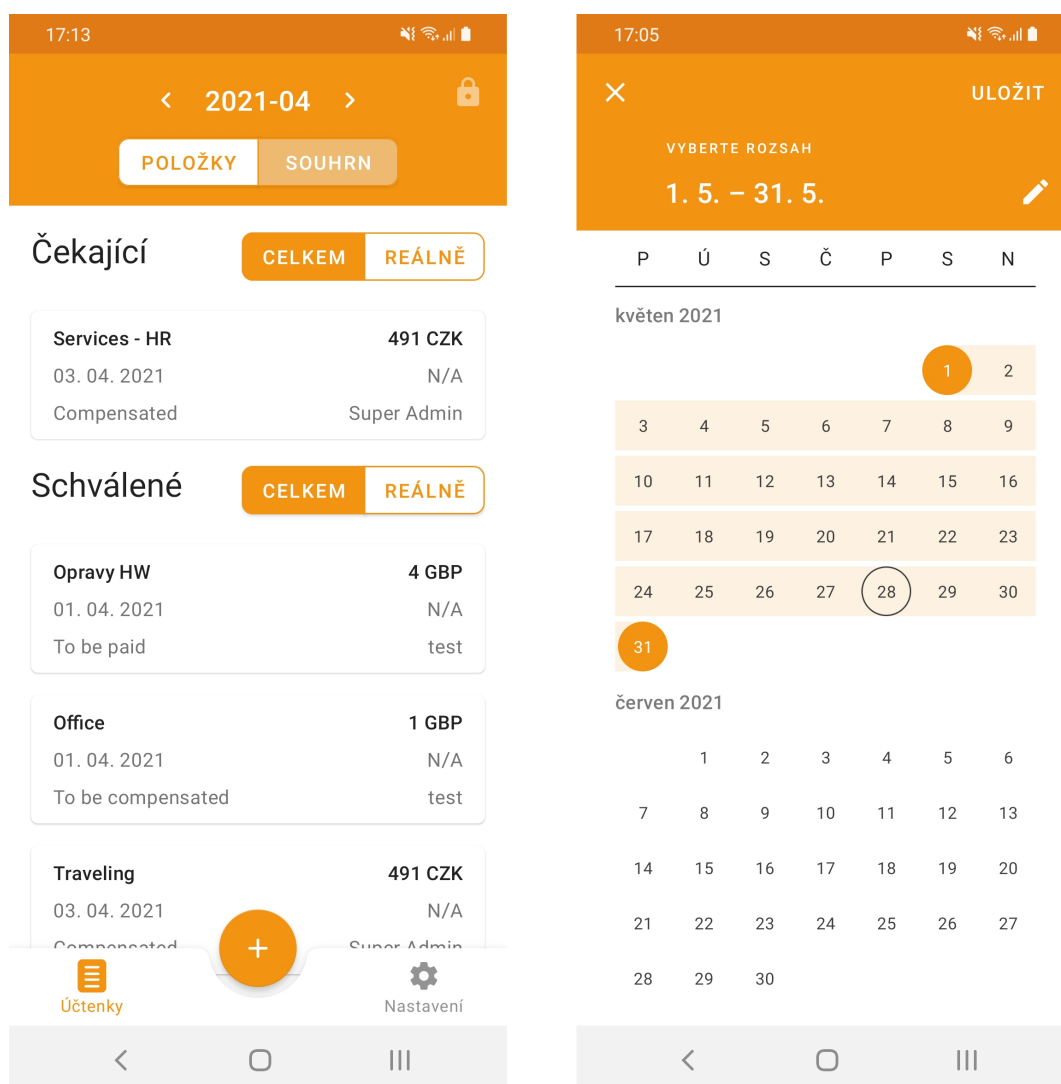
4.3.1 Seznam účtenek

Jak bylo zmíněno v předcházející kapitole, prvním režimem této obrazovky je režim pro seznam účtenek. V tomto režimu uživatel na obrazovce vidí seznam svých účtenek a faktur, které spadají pod vybraný měsíc. Účetní a vedoucí pracovníci zde vidí seznam všech účtenek ve firmě.

Pokud je v aplikaci přihlášen vedoucí pracovník, tak v tomto režimu vidí v ovládací oblasti nahoře nový ovládací prvek. Tím prvkem je ikona zámku. Pomocí tohoto zámku může vedoucí pracovník uzamknout daný měsíc, aby do něj nebylo možné nahrávat nové účtenky. Na obrázku 9 lze vidět, že zámek může být i poloprůhledný. Když je zámek průhledný, nelze na něj kliknout a uzamknout daný měsíc. A to proto, že tento měsíc pořád ještě obsahuje neschválené účtenky. Pro možnost uzavření měsíce je nutné jednotlivé účtenky buď schválit, nebo smazat.

Po úspěšném zamknutí měsíce se ikona zámku změní na ikonu klíče. Tato ikona indikuje, že vedoucí pracovník může tento měsíc znovu odemknout. Měsíc je možné odemknout pouze tehdy, když se jedná o poslední uzamčený měsíc. Naopak zamknout měsíc je možné pouze v případě, že se jedná o první využitý měsíc v historii aplikace anebo se jedná o měsíc, který přímo následuje po již uzamčeném měsíci.

Při uzamčení měsíce dostanou všechny účtenky v tomto měsíci unikátní identifikátor v rámci celého systému. Tento identifikátor následně používá účetní při práci s účtenkami. Pokud by byl měsíc odemčen a byla do něj nahrána nová faktura, je nutné, aby existující identifikátory pořád zůstaly stejné. Při znovu uzamčení tohoto měsíce bude vygenerován nový identifikátor pouze pro ty faktury, které před původním uzamčením měsíce v systému nebyly.



Obrázek 9 – Seznam účtenek a kalendář pro výběr měsíce.

Samotný seznam účtenek obsahuje dva typy položek. První z nich je hlavička sekce. Účtenky jsou v seznamu rozděleny na dvě sekce. První sekcí jsou čekající účtenky. Tato sekce obsahuje účtenky, které ještě čekají na schválení vedoucím pracovníkem. Poté, co uživatel nahraje novou účtenku, se tato účtenka objeví v této sekci. Dokud není účtenka schválená, může ji ten, kdo ji nahrál, pořád upravovat. Druhá sekce obsahuje již schválené účtenky. Schválené účtenky mohou upravovat pouze vedoucí pracovníci. Sekce neschválených účtenek je zobrazena pouze tehdy, pokud měsíc ještě není uzamčen. V uzamčeném měsíci sekce neschválených účtenek nedává smysl a je tedy skrytá.

Dříve zmíněná hlavička sekce vždy obsahuje také přepínací tlačítko. Toto tlačítko přepíná zobrazení ceny pro účtenky v celém seznamu. Při každém přepnutí tohoto tlačítka se jeho hodnota aktualizuje pro všechny sekce. První a zároveň výchozí možností je zobrazení celkové ceny, která byla na účtence uvedena. Druhou možností je zobrazení takzvané reálné ceny. Reálná cena je často stejná jako ta celková, ale nemusí to tak být vždy. Například pokud se jedná o daňový náklad firmy, je reálná cena pro firmu cena bez DPH. Taková účtenka se také v tomto systému musí nacházet v kategorii typu „daňová“.

Tyto hlavičky sekcí byly v aplikaci implementovány jako „sticky headers“ (volně přeloženo jako lepící hlavičky). To znamená, že při procházení dlouhého seznamu účtenek zůstane hlavička vždy „přilepena“ na horní okraj seznamu. Ve chvíli, kdy je jedna přilepená hlavička nahrazována další, vizuálně to vypadá, jako že je původní hlavička posouvána mimo seznam. Tuto funkci využijí hlavně vedoucí pracovníci, kteří vidí účtenky všech zaměstnanců.

Lepící hlavičky jsou v aplikacích pro operační systém Android implementovány pomocí takzvané dekorace seznamu (původní název je „RecyclerView decoration“). Jedná se o techniku, která využívá kreslení aktuální hlavičky na vrchol seznamu. Tato hlavička se skutečně kreslí na plátno a nevyužívá běžné prostředky pro tvorbu uživatelského rozhraní (takzvané „Views“). Aby uživatel mohl s těmito hlavičkami vykreslenými nad samotným seznamem interagovat, je potřeba, aby aplikace ručně poslouchala na doteky na samotné hlavičce a převáděla je na kliknutí na pomyslná tlačítka.

Druhým typem položky ve vykreslovaném seznamu je položka samotné účtenky. Tato položka vypadá jako karta a obsahuje všechny důležité informace o dané účtence. Aby byla každá karta účtenky dobře viditelná a oddělitelná od ostatních, tak ve světlém režimu vrhá jemný stín. V tmavém režimu by stín nebyl vidět, tak je karta zvýrazněna zesvětlením

jejího pozadí oproti barevnému podkladu celého seznamu. Karta, která by vrhala stín, je v tmavém režimu blíž pomyslnému zdroji světla. Z toho důvodu je barva karty světlejší než barva celého pozadí.

Na kartě účtenky v seznamu je zobrazeno šest nejdůležitějších informací. Tyto informace jsou zobrazeny ve dvou sloupcích a třech řádcích. Údaje na prvním řádku jsou zvýrazněny tloušťkou a barvou písma. Vlevo se na prvním řádku nachází název kategorie, do které byla účtenka uživatelem nahrána. Vpravo je na prvním řádku vypsána cena této účtenky. Cena je zobrazena v dobře čitelné podobě s oddělovači tisíců. Vedle samotné částky je také zobrazen kód měny, který je udána na účtence. Zobrazená cena se mění v závislosti na vybrané hodnotě přepínacího tlačítka v hlavičce sekce. Cena tedy může být celková cena nebo takzvaná reálná cena.

Na druhém řádku se v kartě účtenky nachází datum, kdy byla účtenka zaplacená. Také se může jednat o datum splatnosti faktury. Dle aktuální jazykové mutace aplikace je datum vypsáno ve standardním českém formátu nebo jako DD-MM-YYYY. Tudíž první část data je číslo dne v měsíci, které je vždy vypsáno jako dva znaky. To znamená, že číslo dne nižší než desátý den v měsíci budou začínat nulou. Druhá část data obsahuje číslo měsíce v roce. Pokud je číslo měsíce příliš nízké, tak také bude vypsáno s počáteční nulou. Poslední částí data je rok, který je vypsán v plné délce čtyř znaků.

Na pravé straně druhého řádku se nachází unikátní číslo účtenky, které je generováno po uzamčení měsíce. Před samotným číslem účtenky je vypsána mřížka (#), která indikuje, že se jedná o pořadové číslo účtenky. Pokud měsíc ještě nebyl uzamčen, toto číslo není ještě dostupné. V tom případě je místo čísla účtenky vypsán údaj „N/A“.

Na třetím řádku se v kartě účtenky nachází její status. Status účtenky může být například: „zaplacená“ nebo „nutno zaplatit“. Počet statusů není omezen a vedoucí pracovník může tvořit nové dle potřeby. Každá účtenka ale může mít vždy pouze jeden aktivní status. Poslední informací na kartě účtenky je informace o tom, kdo účtenku do systému nahrál. Tato informace není pro řadové zaměstnance nijak důležitá, protože řadoví zaměstnanci vidí pouze své vlastní účtenky. Je ale velmi důležitá pro vedoucí pracovníky, kteří hned vidí, kdo je zodpovědný za tuto účtenku nebo komu je potřeba proplatit výdaje.

Při načítání položek do seznamu se využívá principu, kdy se nejdříve aplikace pokusí stáhnout data z internetu, aby se vždy zajistila aktuálnost dat. Pokud uživatel není připojen k internetu, budou v seznamu místo chyby zobrazena data uložená v lokální databázi.

Uživatel si může zobrazená data aktualizovat pomocí gesta, kterému se anglicky říká „pull to refresh“. Při využití tohoto gesta se při načítání dat nepoužije lokální databáze, ale aplikace se pokusí data získat pouze ze serveru. Pokud uživatel stále není připojen k internetu zobrazí se místo seznamu účtenek chybový stav. Chybová obrazovka uživateli vysvětlí, že si má zapnout internetové připojení, a dá mu možnost pokusit se načíst data znovu. Při pokusu načíst data po chybě se znovu využije lokální databáze, pokud server stále neodpovídá. Lokální databáze se použije proto, aby uživatel nebyl zaseknutý v chybovém stavu bez jakékoliv možnosti procházet aplikací bez internetu.

Jak již bylo zmíněno dříve, aplikace otevře tuto obrazovku automaticky po přihlášení. Pokud je uživatel v jiné části aplikace, může tuto obrazovku otevřít manuálně pomocí první položky spodního navigačního menu. Toto navigační menu je viditelné na obrazovkách nejvyšší úrovně. Tedy obrazovkách, které nejsou hlouběji zanořené v aplikaci.

Další možností pro otevření této obrazovky je, že uživatel klikne na webovou adresu ukazující na stejnou stránku ve webovém rozhraní. Po kliknutí na tuto adresu se otevře aplikace a načte data pro měsíc, který byl určený v dané adrese. Poslední možností navigace je automatické otevření seznamu účtenek po nahrání účtenky do daného měsíce. Po nahrání účtenky se otevře seznam účtenek a načte data měsíce, do kterého byla účtenka nahrána.

4.3.2 Možnosti práce s účtenkou

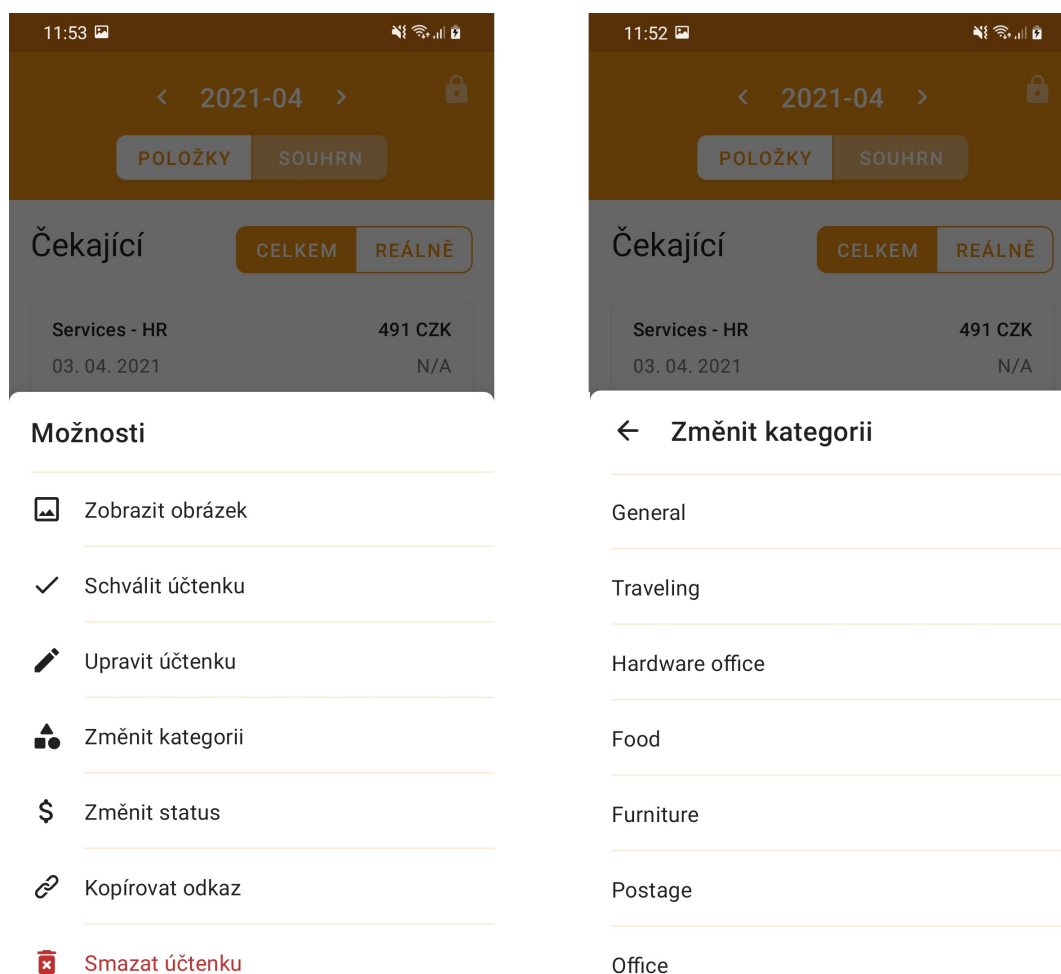
Další důležitou obrazovkou, která se objevuje přímo nad seznamem účtenek, je obrazovka pro práci s účtenkou. Tato obrazovka se otevře, pokud uživatel podrží prst na jednom záznamu účtenky v seznamu. Otevřená obrazovka vždy provádí své změny nad tou účtenkou, kterou uživatel vybral v předchozím kroku.

Tato obrazovka je prezentována jako „BottomSheetDialog“. To znamená, že je otevřená nad předchozí obrazovkou, která je zakrytá šedým průhledným pozadím. BottomSheetDialog se od normálního dialogu liší tím, že je ukotven ke spodní části obrazovky. Při otevření obrazovky tohoto typu se její obsah zobrazí pouze na 50 % obrazovky a při uživatelské interakci se rozbálí na 100 % své výšky. [33]

Přestože tato obrazovka vypadá poměrně jednoduše, téměř každá položka má své podmínky, které musejí být splněny, aby mohla být zobrazena. Obrazovka tedy byla imple-

mentována jako dynamický seznam, do kterého jsou přidávány možnosti při splnění všech jejich podmínek.

Nahoře na obrazovce se nachází její jméno, jak ukazuje obrázek 10. Pokud obrazovka obsahuje tolik položek, že se jejich seznamem musí posouvat, tak tento název obrazovky zůstává ukotven nahoře pro jednoduchou orientaci. Pokud otevřeme obrazovku pro změnu kategorie nebo statusu účtenky, tak se vedle názvu obrazovky animovaně objeví šipka zpět, která nám umožní se dostat zpět do výchozího stavu. Tyto obrazovky budou popsány o několik odstavců později.



Obrázek 10 – Možnosti práce s účtenkou.

První možností v seznamu je možnost prohlédnout si obrázek účtenky. Tato možnost je v seznamu dostupná vždy a neváží se na ni žádné podmínky. Po zvolení této možnosti se zavře aktuální obrazovka s možnostmi práce s účtenkou a otevře se nová obrazovka. Na otevřené obrazovce se nachází fotka nahrané účtenky či faktury. Tuto fotku je možné zvětšit a prohlížet pomocí gest.

Druhou položkou v seznamu je možnost schválit účtenku. Pokud byla účtenka už schválená, tato možnost je nahrazena možností vrátit účtenku do sekce neschválených. Účtenky samozřejmě nemůže schvalovat kdokoliv. Tato možnost je tedy skrytá, pokud přihlášený uživatel nedisponuje právem schvalování účtenek. Toto právo mají v současné době pouze vedoucí pracovníci. Druhým kritériem, které musí být splněno pro zobrazení této možnosti, je, že měsíc, ve kterém se účtenka nachází, nesmí být uzavřený.

Další možností v seznamu je možnost upravit účtenku. Po zvolení této možnosti se zavře aktuální obrazovka a otevře se nová obrazovka pro úpravu této účtenky. Zobrazení této položky je znovu podmíněno hned několika pravidly. Prvním pravidlem je, že měsíc nesmí být uzavřen. Druhé pravidlo je složeno ze dvou částí. Účtenka ještě nesmí být schválená nebo aktuálně přihlášený uživatel musí disponovat právem upravovat schválené účtenky. I toto právo mají pouze vedoucí pracovníci.

Třetí možností je změna kategorie účtenky. Po zvolení této možnosti se změní nadpis aktuální obrazovky na změnu kategorie a vedle něj se objeví šipka zpět. Po stisknutí šipky se uživatel může vrátit do seznamu s možnostmi účtenek. Seznam možností samozřejmě zmizí a místo něj se objeví seznam dostupných kategorií. Po kliknutí na jednu z kategorií se změní kategorie aktuálně upravované účtenky na tu zvolenou. Zároveň se po vybrání kategorie tato obrazovka uzavře a znovu se ukáže seznam možností práce s účtenkou. Aby zůstal seznam účtenek vykreslený pod obrazovkou s možnostmi aktuální, tak se po úspěšné změně kategorie aktualizuje. Pokud by se stalo, že uživatel kliká moc rychle a povede se mu kliknout na možnost „Změnit kategorii“ ještě před tím, než se načte seznam kategorií, zobrazí se uživateli hláška, která ho vyzve, aby chvíli počkal, než se kategorie načtou do paměti. Ve chvíli, kdy se kategorie skutečně načtou, je uživatel notifikován a ví, že může pokračovat v práci.

Po možnosti změnit kategorii následuje možnost změnit status účtenky. Zvolení této možnosti způsobí stejné důsledky jako změna kategorie s tím rozdílem, že se na účtence nakonec nezmění kategorie, ale status.

Dříve popsané možnosti změny kategorie a statusu se samozřejmě také neukážou za každých okolností a pro jejich použití musejí být splněny určité podmínky. Tyto podmínky jsou stejné pro obě možnosti. První podmínkou je, že účtenka nesmí být schválená nebo uživatel musí disponovat právem upravovat schválené účtenky. Druhou podmínkou je, že měsíc, ve kterém se upravovaná účtenka nachází, nesmí být uzavřen. Při splnění

obou podmínek je tedy možné měnit účtence kategorii a status pomocí této obrazovky a nemusí se zbytečně otevírat obrazovka pro plnou úpravu účtenky.

Další možností je možnost zkopírovat odkaz na tuto účtenku. Po zvolení této možnosti se uživateli do schránky uloží webový odkaz na tuto účtenku. Pokud na tento odkaz někdo klikne na zařízení, na kterém je tato aplikace nainstalovaná, tak se mu místo webu otevře tato aplikace na stránce detailu dané účtenky. Po zkopírování odkazu do schránky se uživateli zobrazí zpráva, že byl odkaz vložen do schránky. Stejně jako zobrazení obrázku účtenky ani na tuto položku se nevztahují žádná pravidla zobrazení a je viditelná vždy.

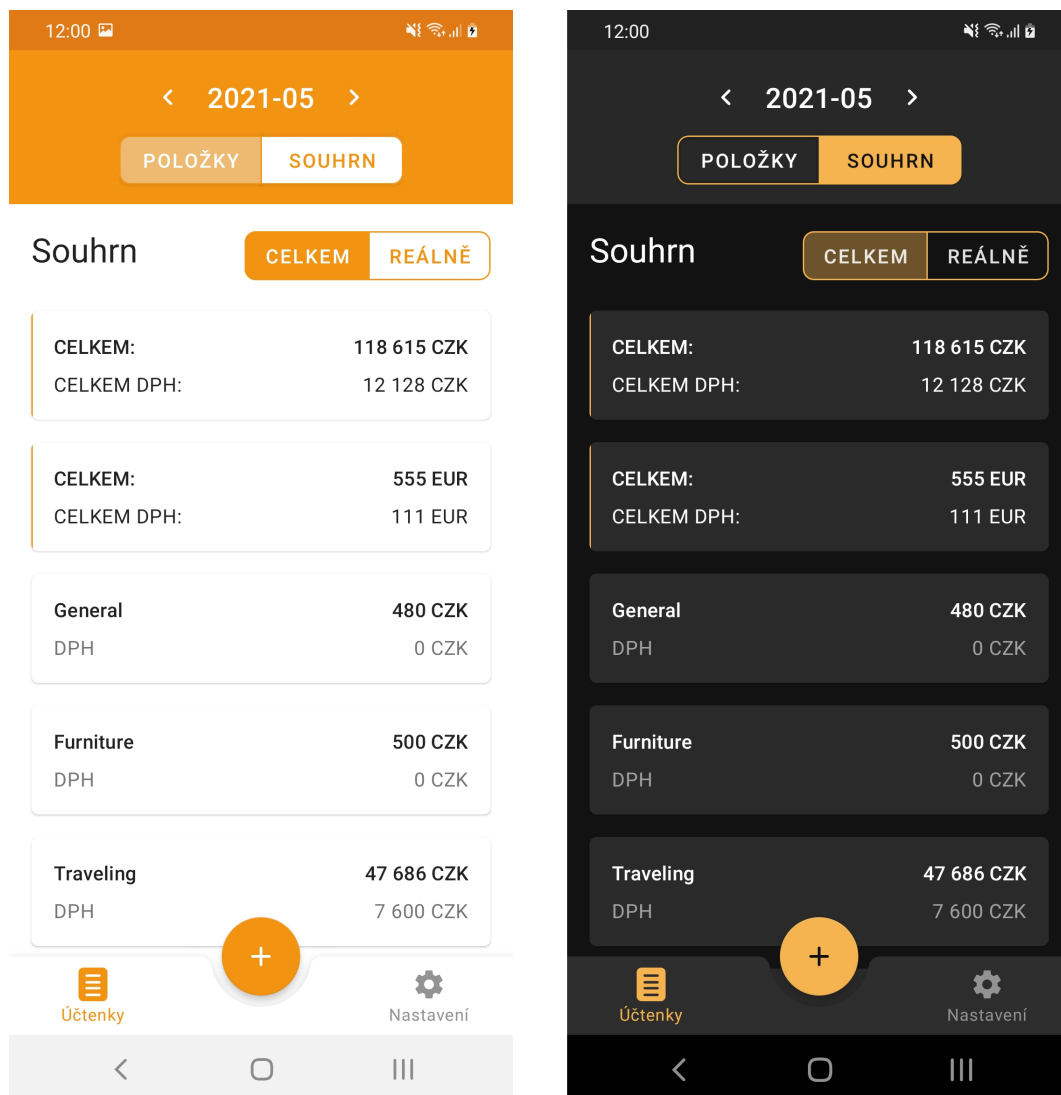
Poslední možností v tomto seznamu pro práci s účtenkou je možnost účtenku smazat. Tato možnost je v seznamu zvýrazněna červeně, aby bylo uživateli jasné, že je akce destruktivní. Po vybrání této možnosti se ještě zobrazí potvrzovací dialog, zda si je uživatel skutečně jistý, že chce tuto účtenku smazat. Po potvrzení se účtenka skutečně smaže. Po úspěšném smazání účtenky se zavře aktuální obrazovka a uživatel je vrácen na seznam účtenek. Tento seznam se uživatele automaticky aktualizuje, aby se v něm nezobrazovala právě smazaná účtenka.

4.3.3 Měsíční souhrn

Druhým stavem obrazovky, na které se běžně nachází seznam účtenek, je měsíční souhrn. Do tohoto režimu je možné obrazovku přepnout pomocí přepínacího tlačítka nahoře v ovládací oblasti. Při zvolení možnosti „Souhrn“ se tedy obrazovka přepne do tohoto režimu. Tato obrazovka obsahuje souhrnné záznamy o tom, kolik peněz bylo utraceno za určité kategorie účtenek.

Jak ukazuje obrázek 11, tato obrazovka vypadá podobně jako v režimu, kdy ukazuje seznam účtenek. Jedná se také o seznam, ale tento seznam obsahuje jiné položky. První položkou je znovu hlavička sekce. V měsíčním souhrnu je tentokrát sekce pouze jedna, a to „Souhrn“. Hlavička sekce také obsahuje přepínací tlačítko, pomocí kterého lze přepnout režim ceny zobrazovaný v dalších položkách. Při posouvání se seznamem dál zůstává hlavička připnutá k hornímu okraji seznamu. To, že hlavička zůstává ukotvena, umožňuje rychlé přepínání typu vypisované ceny i při prohlížení dlouhého seznamu.

Hned pod hlavičkou sekce se nachází první typ položky v měsíčním souhrnu. Tuto položku lze snadno poznat tak, že je zvýrazněna oranžovým pruhem v levé části své karty. Jedná se o souhrnnou položku, která obsahuje součet cen účtenek ze všech kategorií



Obrázek 11 – Měsíční souhrn ve světlém a tmavém režimu.

ve vybraném měsíci. Na prvním řádku této položky je tedy celková cena a na druhém řádku je celkové DPH. Těchto zvýrazněných položek se může v seznamu nacházet i více. A to z toho důvodu, že není možné numericky sčítat ceny z různých měn. Proto se na začátku seznamu nachází tolik zvýrazněných položek, kolik bylo v tomto měsíci různých měn.

Posledním typem položky seznamu je položka souhrnu za danou kategorii. Položek tohoto typu je v seznamu tolik, kolik bylo použito kategorií za vybraný měsíc. Na prvním řádku této položky se nachází celková cena všech účtenek s touto kategorií ve vybraném období. Na druhém řádku se nachází celkové DPH ze všech účtenek v této kategorii. Při přepínání režimu ceny z celkové na reálnou se mění pouze údaje v prvním řádku, jelikož DPH vždy zůstává stejné.

4.4 Přidání nové účtenky

Hlavním úkolem této aplikace je nahrávání nových účtenek a faktur do systému. Přidání nové účtenky se skládá hned z několika kroků, které budou popsány v této kapitole.

Pro započetí tohoto procesu stačí kliknout na oranžové kruhové tlačítko s ikonou plus. Toto tlačítko se nachází uprostřed spodní navigační lišty, která je viditelná na všech obrazovkách první úrovně. Toto tlačítko není přímo součástí navigační lišty, ale vykrajuje z ní svůj tvar a vznáší se nad obsahem právě zobrazené obrazovky. Dříve popsanou polohu tlačítka ukazuje obrázek číslo 9, který současně ukazuje obrazovku se seznamem účtenek.

Po kliknutí na dříve popsané tlačítko se otevře obrazovka, která umožňuje vyfotit účtenku přímo v aplikaci. Při otevírání této obrazovky se přehraje animace, které se říká kruhové odhalení (anglicky „circular reveal“). To znamená, že se nová obrazovka bude objevovat ve stále se zvětšujícím kruhu, dokud nezaplní celou obrazovku. Tato animace samozřejmě začíná v místě, odkud bylo kliknuto na tlačítko. Při návratu z této obrazovky zpět se tato animace přehraje přesně naopak. Obrazovka fotoaparátu bude mizet v neustále se zmenšujícím kruhu, který skončí přímo v kruhovém tlačítku pro otevření této obrazovky.

Předtím, než je možné ukázat obrazovku s fotoaparátem, je nutné získat od uživatele jednotlivá oprávnění. Pro správnou funkci této obrazovky potřebuje aplikace tato tři oprávnění: CAMERA, READ_EXTERNAL_STORAGE a WRITE_EXTERNAL_STORAGE. Od Androidu 6.0 se na tato oprávnění musí aplikace ptát za běhu a musí dostat výslovný souhlas uživatele. Pokud by aplikace běžela na Androidu nižší verze, tak by byla všechna oprávnění schválena hned instalací aplikace.

Název prvního oprávnění je samovysvětlující – jedná se o právo přístupu k fotoaparátu. Toto právo je potřeba k tomu, aby bylo možné na obrazovce vykreslit náhled z fotoaparátu, ovládat blesk a následně fotografii skutečně pořídit.

Druhé oprávnění umožňuje aplikaci přistupovat k souborům v telefonu. V tomto případě umožňuje přístup hlavně k médiím a je využito k zobrazení náhledu naposledy vyfocené fotografie. Poslední oprávnění naopak umožňuje aplikaci zapisovat data do úložiště telefonu. Toto oprávnění je využito pro ukládání vyfocených fotografií do telefonu. Fotografie nejsou aplikací automaticky mazány, aby k nim měl uživatel přístup i v galerii, kdyby je potřeboval.

V případě, že by uživatel tato oprávnění nepovolil, nemůže tuto obrazovku otevřít. Když uživatel oprávnění odmítne, zobrazí se mu informativní hláška, která vysvětlí, že jsou oprávnění potřeba a nemůže bez nich pokračovat dál. Při odmítání oprávnění může uživatel také vybrat, aby se ho systém na dané oprávnění už znovu neptal. Pokud uživatel tuto volbu vybere, tak se mu po kliknutí na tlačítko pro přidání účtenky zobrazí hláška, že pro pokračování je nutné přijmout tato oprávnění. V této hlášce bude navíc tlačítko, které po kliknutí vezme uživatele do systémového nastavení, kde může oprávnění povolit.

4.4.1 Fotoaparát v rámci aplikace

První obrazovkou, kterou uživatel navštíví, pokud chce nahrát novou účtenku, je obrazovka s fotoaparátem. Kromě přímočaré techniky otevírání této obrazovky přes plovoucí tlačítko v rámci spodní navigační lišty existují pro uživatele další dvě možnosti.

První možností je, když se uživatel pokusí otevřít odkaz, který by ve webové aplikaci označoval adresu formuláře pro nahrání nové účtenky. Pokud je uživatel přihlášen, tak se po kliknutí na tento odkaz aplikace otevře přímo na obrazovce s fotoaparátem. Pokud není přihlášen, tak se nejdříve ukáže přihlašovací formulář. Po úspěšném přihlášení se následně objeví přímo obrazovka s fotoaparátem.

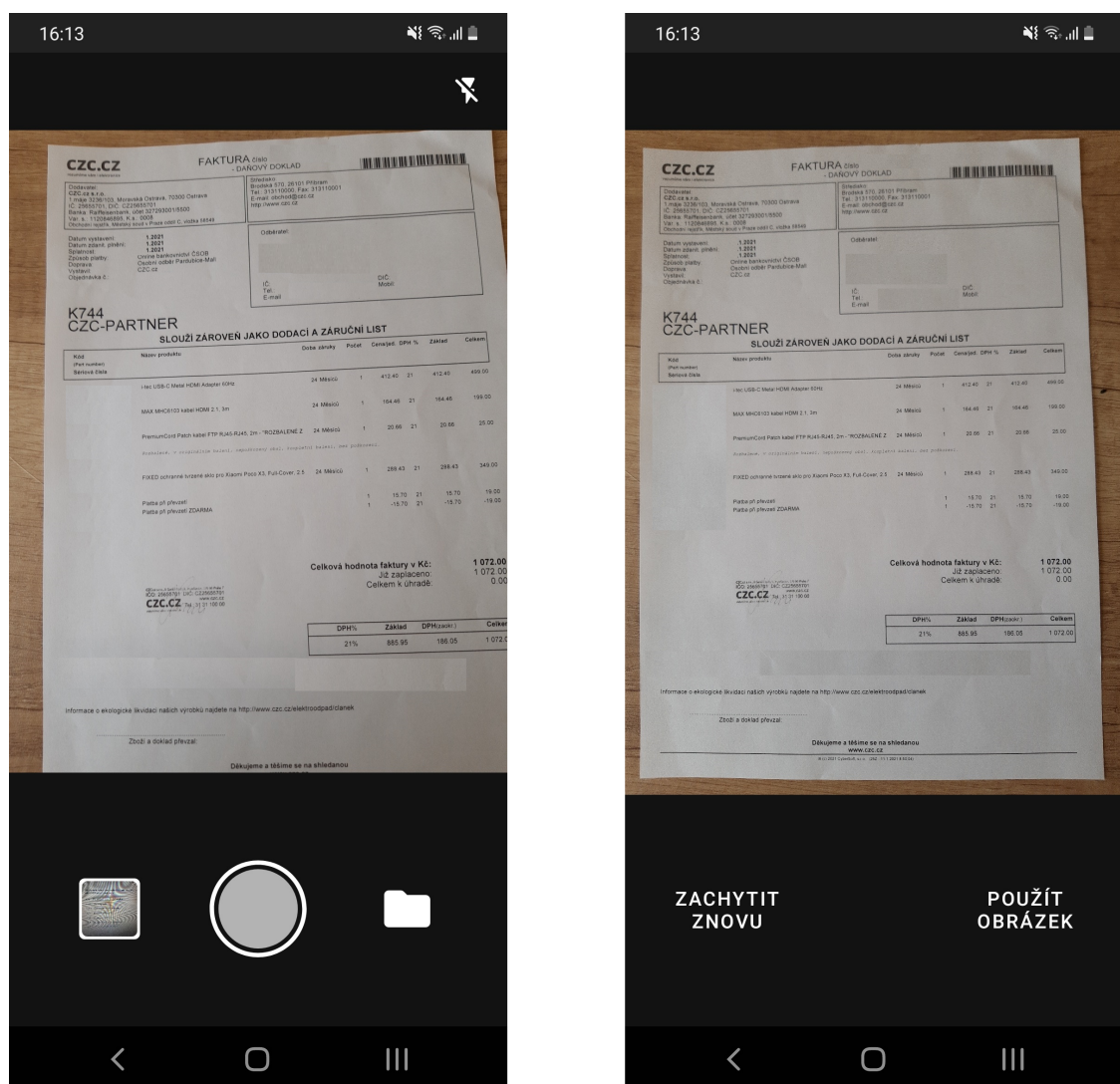
Druhou nekonvenční možností, jak otevřít tuto obrazovku, je využití statických aplikačních zkratk (anglicky „App shortcuts“). Pro využití těchto zkratk je potřeba telefon s verzí Androidu 7.1 a vyšší. Z pohledu vývojáře se jedná o verzi API číslo 25. [34]

Aplikační zkratku uživatel použije tak, že ve svém spouštěči aplikací dlouho podrží prst na této aplikaci. Poté se mu zobrazí kontextové menu, které obsahuje nejen registrované zkratky, ale například i možnost odinstalovat danou aplikaci. Jednotlivé zkratky je možné prstem přetáhnout a uložit si je na plochu telefonu. Po kliknutí na zkratku se aplikace chová stejně jako při kliknutí na webový odkaz a otevře se na obrazovce s fotoaparátem. Tato aplikace má pouze jednu takto registrovanou zkratku.

Obrazovka s fotoaparátem má černé pozadí v tmavém i světlém režimu, jelikož by nedávalo smysl mít kolem náhledu fotoaparátu bílé pozadí. Uprostřed obrazovky se nachází dříve zmíněný náhled přímo ze zadního fotoaparátu na zařízení. Tento náhled se neustále aktualizuje, aby uživatel věděl, co fotí. Pokud by byla faktura v náhledu málo ostrá, může do něj uživatel kliknout prstem. Na toto vybrané místo fotoaparát zaostří. Uživateli je

to zvýrazněno pomocí jednoduché animace. Samotný náhled z fotoaparátu a následně i vyfocené fotografie mají vždy poměr stran 3:4.

Vpravo nahoře nad náhledem fotoaparátu se nachází ikona, která označuje aktuální status blesku. Pokud telefon nemá u zadního fotoaparátu blesk, tato ikona je poloprůhledná a nelze s ní interagovat. Většina telefonů ale blesk má a po kliknutí na tuto ikonu se změní jeho režim při fotografování v této aplikaci. Výchozím stavem pro focení je, že je blesk vypnutý. Dalšími dostupnými režimy jsou automatický blesk a vždy zapnutý. Pokud by uživatel nevěděl, co daná ikona znamená, může na ikoně dlouho přidržit prst a zobrazí se kontextová informace vysvětlující aktuální stav blesku.



Obrázek 12 – Fotoaparát uvnitř aplikace.

Vlevo dole se pod náhledem fotoaparátu nachází náhled naposled vyfocené fotografie v bílém rámečku. Pokud na tomto telefonu nejsou uloženy žádné fotografie, tak se ukáže

pouze ikona galerie bez rámečku. Na tuto ikonu uživatel klikne, pokud nechce využít fotoaparát přímo v aplikaci, ale chce vybrat dříve vyfocenou fotografii z galerie. Po kliknutí na tuto ikonu se tedy otevře systémová galerie, ve které může uživatel vybrat jeden obrázek. Po potvrzení výběru je uživatel přesměrován na formulář pro nahrání nové účtenky, který je popsán v kapitole 4.4.2.

Vpravo dole se pod náhledem fotoaparátu nachází ikona složky. Tato ikona indikuje možnost vybrat PDF dokument přímo ze souborového systému telefonu. Po kliknutí na tuto ikonu se tedy otevře systémová aplikace pro výběr PDF dokumentu. Vybraný dokument může mít jednu nebo i více stran. Do systému bude nahrán jako soubor a ne jako obrázek, tudíž nedojde k žádné ztrátě dat. Po potvrzení výběru je uživatel přesměrován na formulář pro nahrání nové účtenky.

Poslední prvek, který se nachází pod náhledem z fotoaparátu, je samotná spoušť. Horizontálně se nachází uprostřed obrazovky a je tedy mezi dvěma naposledy popsanými ikonami. Spoušť vypadá jako světle šedý kruh, kolem kterého je bílá kružnice, jak ukazuje obrázek 12.

Při kliknutí na spoušť fotoaparátu se na chvíli zobrazí načítací symbol, který zmizí ve chvíli, kdy se skutečně uloží fotografie na disk. Zachycená fotografie se poté ukáže přímo místo náhledu. Uživatel si tedy může určit, zda mu zachycená fotografie přijde dostatečně kvalitní a chce s ní pokračovat.

Pokud fotografie není dostatečně ostrá nebo je nekompletní, může uživatel kliknout na tlačítko „Zachytit znovu“. Po tomto kliknutí se smaže aktuálně vyfocená fotografie a znovu se zobrazí režim fotoaparátu s aktualizujícím se náhledem. Stejného efektu může uživatel dosáhnout, i pokud zmáčkne systémové tlačítko zpět.

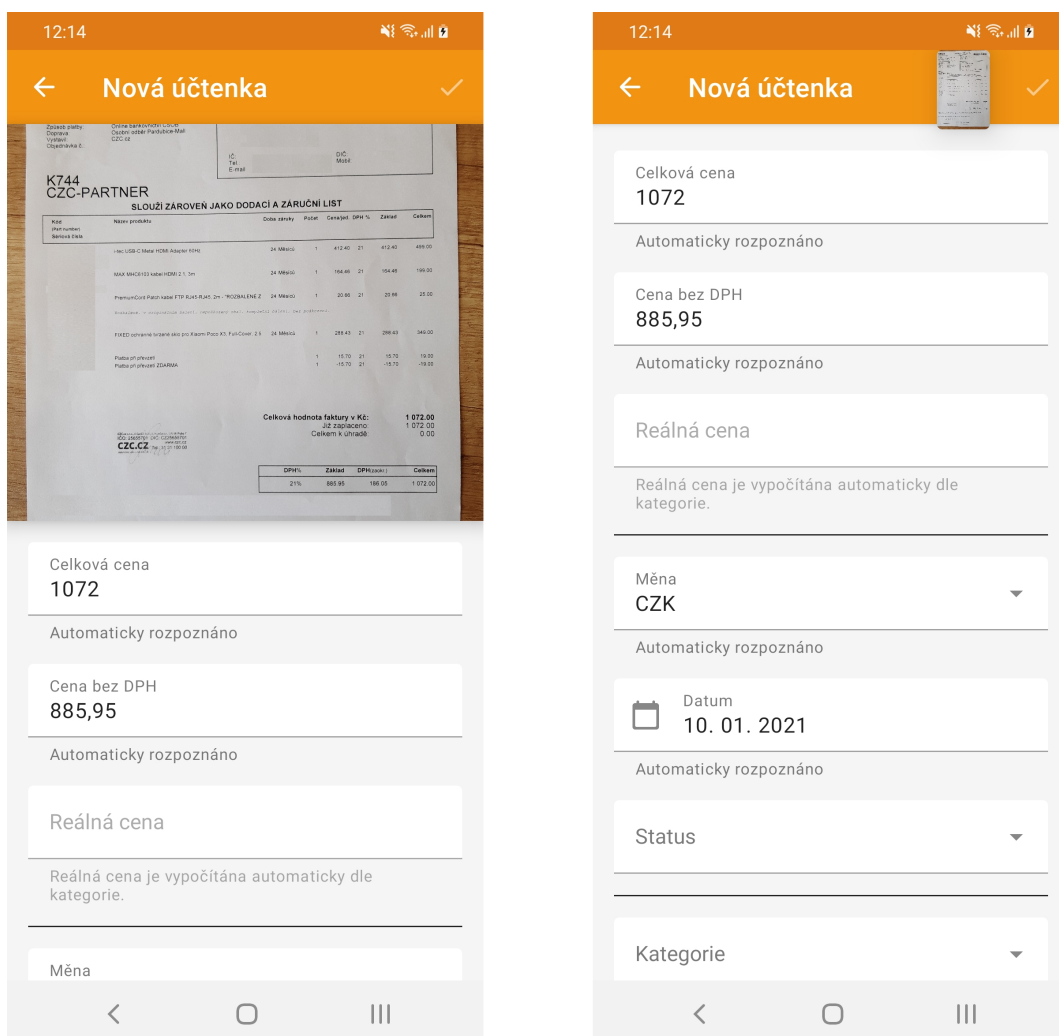
Pokud je fotografie kvalitní, tak uživatel klikne na tlačítko „Použít obrázek“. Fotka tedy nebude smazána a bude uložena do složky této aplikace v systémovém úložišti obrázků. Po kliknutí na toto tlačítko se tedy zavře obrazovka fotoaparátu a otevře se nová obrazovka s formulářem pro nahrání nové účtenky.

4.4.2 Formulář pro přidání účtenky

Po úspěšném výběru fotografie či dokumentu se uživateli zobrazí další obrazovka. Tato obrazovka již obsahuje samotný formulář, který je nutný vyplnit před nahráním účtenky. Část tohoto formuláře může být vyplněna automaticky, pokud zvolená implementace OCR

ve faktuře nalezne příslušné údaje. Obrázek 13 ukazuje stav, kdy byly správně nalezeny všechny automaticky vyhledávané údaje.

Pokud má uživatel povolené automatické rozpoznávání údajů ve fotkách účtenek a faktur, tak se ihned po otevření této obrazovky zobrazí načítací symbol. Pod načítacím symbolem bude tmavé poloprůhledné pozadí, které znázorňuje, že během načítání není možné vyplňovat údaje do formuláře. Během tohoto načítání se stahuje seznam dostupných kategorií, statusů a podobně. Zároveň probíhá automatické rozpoznání údajů ve fotce pomocí dříve zvolené metody. Všechny tyto procesy probíhají paralelně díky využití Kotlin Coroutines. Po dokončení všech dříve zmíněných úkonů načítání zmizí a pokud byly ve fotografii nalezeny nějaké údaje, tak se předvyplní do formuláře.



Obrázek 13 – Přidání nové účtenky.

Na obrázku vlevo se nachází počáteční stav, vpravo je stav po posunutí se v seznamu.

Tato obrazovka byla implementována s pomocí rozložení, které se jmenuje MotionLayout. MotionLayout je rozšířením o něco staršího rozložení s názvem ConstraintLayout. MotionLayout umožňuje automaticky animovat určité prvky při dané uživatelské interakci.

V případě této obrazovky je touto interakcí posouvání se ve formuláři pro přidání nové účtenky. V základním stavu této obrazovky se pod navigační lištou a zároveň nad formulářem nachází náhled fotografie účtenky. Poté, co začne uživatel posouvat formulářem, aby viděl další textová pole, se začne náhled fotografie zmenšovat a posouvat do pravého horního rohu obrazovky vedle tlačítka pro potvrzení vyplnění formuláře. Poté, co fotografie dojede na své nové místo, se při dalším posouváním ve formuláři nic zvláštního neděje. Pokud by se uživatel chtěl znovu podívat na náhled fotografie účtenky, tak se nemusí posunout ve formuláři úplně nahoru. Stačí, když stáhne prstem zmenšený náhled fotografie dolů do své původní pozice. Pokud se uživatel nechce trefovat přímo na zmenšený náhled fotografie, může stejné gesto udělat i na celé navigační liště na horním okraji obrazovky.

Dříve zmíněná navigační lišta se nachází přímo pod zmenšeným náhledem fotografie. Kromě fotografie se v této liště nachází ikona šipky doleva. Po kliknutí na tuto ikonu se zobrazí potvrzovací dialog. Po potvrzení, že uživatel chce skutečně opustit obrazovku pro přidání nové účtenky, se vrátí přímo do seznamu účtenek.

Vedle dříve popsané ikony pro opuštění stávající obrazovky se v navigační liště nachází název této obrazovky. Jedná se o standardní zobrazení v rámci Android aplikací. V poslední řadě je v navigační liště i tlačítko pro potvrzení zadaných údajů ve formuláři pro nahrání nové účtenky. Při úvodním otevření této obrazovky je toto tlačítko znepřístupněné a tudíž poloprůhledné. Po vyplnění všech potřebných údajů se tlačítko zpřístupní a je možné na něj kliknout.

Po kliknutí se znepřístupní formulář pro úpravy a vyplněné údaje spolu s fotkou účtenky se nahrají na server. Pokud by se při nahrávání účtenky vyskytla chyba, tak se dole na obrazovce zobrazí informativní chybová hláška a formulář se znovu zpřístupní pro úpravy. V případě, že žádná chyba nenastane, tak se aktuální obrazovka uzavře a otevře se obrazovka se seznamem účtenek v tom měsíci, do kterého byla tato účtenka nahrána.

Pod navigační lištou se nachází dříve zmíněný náhled fotky účtenky nebo faktury. Náhled neukazuje celou fotografii, protože by zabíral příliš velkou část obrazovky. Náhled je tedy zobrazen tak, aby se neměnil poměr stran a fotografie se v případě potřeby přiblí-

žila a vycentrovala. V případě, že uživatel místo fotografie nahrává PDF dokument, tak bude v tomto náhledu zobrazena jeho první stránka. V případě, že se uživatel chce podívat do tohoto dokumentu, aby zjistil potřebné údaje, tak stačí kliknout na tento náhled a otevře se mu nová obrazovka. Tato obrazovka umožňuje podrobné prohlédnutí fotografií či dokumentů a bude podrobně popsána v kapitole 4.5.1.

Pod náhledem fotografie se již nachází samotný formulář. Formulář je složen z několika sekcí, které jsou odděleny horizontální čarou. Tento styl byl využit z toho důvodu, že se ve formuláři nachází hodně textových polí a čáry měly vizuálně oddělit jednotlivé sekce pro větší přehlednost.

První položkou ve formuláři je textové pole pro zadání celkové ceny. Je do něj možné zadat cenu s přesností na dvě desetinná místa. Jako oddělovač desetinných míst je možné použít jak čárku, tak i tečku. Pro zpříjemnění psaní čísel je při zápisu ceny použita klávesnice, na které jsou pouze číslice. Pokud byla celková cena automaticky nalezena ve fotografii účtenky pomocí OCR, tak je pod textovým polem zobrazena poznámka „Automaticky rozpoznáno“. Pokud uživatel automaticky rozpoznanou cenu změní, tato poznámka zmizí.

Druhé textové pole ve formuláři se také týká ceny. Tentokrát jde o cenu bez DPH a při zápisu platí stejná pravidla, která platí pro zadávání celkové ceny. I do tohoto pole může být cena automaticky vyplněna, pokud aplikace rozezná ve fotografii cenu bez DPH.

Poslední textové pole v první sekci obsahuje takzvanou reálnou cenu. Tuto cenu nezadává uživatel, a proto je textové pole zašedlé a má zakázanou interakci. Pod textovým polem je zobrazen informativní text, který vysvětluje, že tato cena je vypočítána automaticky na základě vybrané kategorie. Toto pole ve formuláři tedy zůstává prázdné, dokud uživatel nevyplní všechny potřebné údaje pro výpočet. Uživatel tedy musí vyplnit celkovou cenu, cenu bez DPH a vybrat kategorii účtenky.

V druhé sekci se na prvním místě nenachází textové pole, ale rozbalovací nabídka s výběrem možností. V této nabídce může uživatel vybrat měnu, ve které byla účtenka placena. Měny jsou reprezentovány svým ISO kódem. I měna může být ve fotografii účtenky automaticky rozpoznána. Pokud byla měna automaticky nalezena v účtence, tak pod rozbalovací nabídkou bude již dříve zmíněná poznámka, která oznamuje tuto skutečnost.

Pod nabídkou pro výběr měny se nachází textové pole, které obsahuje datum splatnosti účtenky. Pokud bylo datum automaticky rozpoznáno, tak bude pod tímto textovým

polem již dříve zmíněná poznámka. Vedle samotného data se vlevo v textovém poli nachází jednoduchá ikona kalendáře, díky které je jednodušší toto pole identifikovat v celém formuláři.

Do tohoto textového pole uživatel ale nemůže psát přímo. Po kliknutí na toto textové pole se objeví dialog s možností výběru data. V tomto dialogu je při jeho otevření vybráno to datum, které bylo ve formuláři vypsáno. Uživatel zde může jednoduše vybrat datum. Pokud by uživatel nechtěl datum hledat ve zobrazeném kalendáři, tak může kliknout na ikonu tužky v pravém horním rohu dialogu. Po kliknutí zmizí kalendář a objeví se textové pole, do kterého může uživatel napsat datum přímo. Po potvrzení napsaného data se dialog uzavře a datum se vyplní přímo do formuláře.

Dalším prvkem ve formuláři je možnost vybrat status účtenky. Status může být například zaplacený nebo čekající na proplacení. Status uživatel vybírá z rozbalovací nabídky stejně jako měnu. Status není možné v účtence rozpoznat automaticky a musí být vždy vyplněn uživatelem. Rozbalovací nabídka pro výběr statusu je posledním prvkem ve této sekci formuláře.

První položkou následující sekce je rozbalovací nabídka pro výběr kategorie. Jelikož je kategorií v systému poměrně hodně, chová se tato rozbalovací nabídka jinak než jakákoliv dříve popsaná. Stejně jako v dříve popsáných případech si může uživatel po kliknutí na tuto nabídku vybrat položku ze seznamu. Navíc uživatel může dlouhý seznam vyfiltrovat tím, že začne psát název požadované kategorie. Při psaní budou ze seznamu automaticky mizet ty položky, které neobsahují napsaný text. Nakonec je nutné na vybranou položku kliknout pro potvrzení volby.

Pod nabídkou pro výběr kategorie se nachází přepínač. Tento přepínač určuje, zda je možné odečíst daň z celkové částky při výpočtu reálné ceny pro firmu. Tento přepínač je dostupný pouze ve chvíli, kdy byla v předchozím kroku vybrána takzvaná daňová kategorie. Pokud nebyla, tak je s tímto přepínačem zakázáno interagovat a jeho hodnota není využita.

Další položkou formuláře je rozbalovací menu pro výběr zaměstnance, kterému náleží tato účtenka. Tuto položku ale nevidí všichni uživatelé této aplikace. Položka se zobrazí pouze, když přihlášený uživatel disponuje právem přiřazovat k účtenkám různé uživatele. Tímto právem v současné době disponují pouze vedoucí pracovníci.

Položka pro výběr člověka je pro vedoucí pracovníky velmi důležitá, jelikož mohou do systému nahrát faktury, které mají být později proplaceny jiným uživatelům. Toto rozbalovací menu funguje stejně jako to pro výběr kategorie. To znamená, že dlouhý seznam možností je možné nejprve vyfiltrovat a následně vybrat daného člověka.

Jako poslední se ve formuláři nachází textové pole pro vyplnění popisu účtenky. Jako jediný údaj ve formuláři je tento popis nepovinný. Jako popis je možné vyplnit i delší text, jelikož lze do textového pole vložit i nové řádky. Při psaní delšího textu se textové pole přizpůsobuje svému obsahu a roztahuje se na výšku dle potřeby.

4.5 Detail účtenky

Další důležitou obrazovkou je obrazovka, která reprezentuje detail nahrané účtenky. Na detail účtenky se uživatel dostane velmi jednoduše, a to kliknutím na položku účtenky v seznamu.

Druhou možností, jak je možné se na tuto obrazovku dostat, je kliknutí na webový odkaz, který odkazuje na tuto účtenku. Tento odkaz je možné vygenerovat na stránce s možnostmi práce s účtenkou pomocí volby „Kopírovat odkaz“. Další variantou, jak odkaz získat, je pomocí tlačítka „Sdílet“ v rámci této obrazovky nebo prostřednictvím webové aplikace.

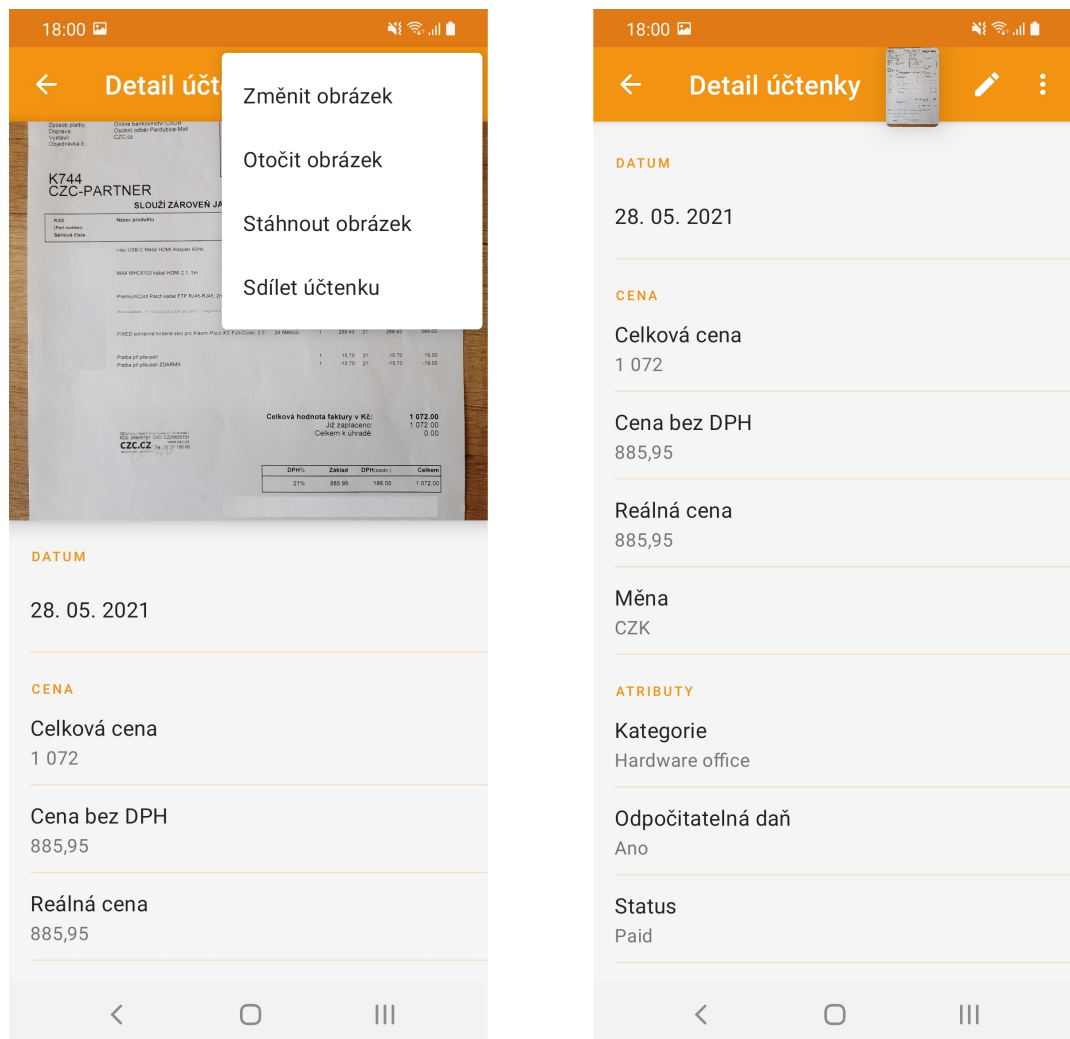
Stejně jako na obrazovce pro přidání nové účtenky se i na této obrazovce nahoře nachází navigační lišta. Tato lišta znovu obsahuje název této obrazovky a šipku zpět, pomocí které se může uživatel vrátit do seznamu účtenek. Kromě těchto běžných prvků může navigační lišta obsahovat i několik dalších položek specifických pro tuto obrazovku. Na zobrazení těchto položek se ale většinou vztahují různá pravidla.

První dříve zmíněnou ikonou, která není zobrazena vždy, je ikona tužky. Tato ikona značí možnost úpravy údajů účtenky. Po kliknutí na tuto ikonu se otevře obrazovka pro úpravu účtenky, která bude podrobně popsána v kapitole 4.6. Tato ikona je zobrazena pouze tehdy, když měsíc, ve kterém se účtenka nachází, není uzavřen. Druhou podmínkou je, že účtenka ještě nebyla schválena nebo má uživatel právo upravovat již schválené účtenky. Toto právo mají pouze vedoucí pracovníci.

Druhou položkou v navigační liště jsou tři tečky. Tyto tečky jsou sice zobrazeny vždy, ale možnosti, které obsahují, jsou vázány různými podmínkami. Po kliknutí na tyto tři

tečky se otevře kontextové menu, které obsahuje několik dalších položek. Toto kontextové menu ukazuje obrázek 14.

Jako první se v kontextovém menu nachází možnost „Změnit obrázek“. Podmínka pro zobrazení této možnosti je stejná jako podmínka pro zobrazení možnosti úpravy účtenky. Po výběru této možnosti se kontextové menu zavře a otevře se nový Bottom-Sheet dialog (dialog ukotvený ke spodní části obrazovky, vysvětleno v kapitole 4.3.2). Tento dialog obsahuje tři možnosti, jak změnit obrázek dané účtenky. Těmito možnostmi jsou vyfotit nový obrázek pomocí systémového fotoaparátu, vybrat nový obrázek z galerie nebo vybrat PDF soubor pomocí aplikace k tomu určené. Po vybrání nového obrázku se pod navigační lištou na chvíli zobrazí načítací pruh a po úspěšném nahrání se vybraný obrázek objeví v náhledu.



Obrázek 14 – Detail účtenky s otevřeným kontextovým menu.

Druhou položkou kontextového menu je možnost „Otočit obrázek“. Tato možnost je zobrazena pouze ve chvíli, kdy měsíc není uzavřen a účtenka není schválená nebo má uživatel oprávnění upravovat schválené účtenky. Poslední podmínkou pro zobrazení této možnosti je, že aktuální soubor účtenky musí být obrázek. Nesmí se tedy jednat o PDF soubor.

Třetí možnost kontextového menu není vázána žádnými podmínkami a je zobrazena vždy. Jedná se o možnost stáhnout si obrázek účtenky jako soubor do úložiště telefonu. Po vybrání této možnosti se dole na obrazovce objeví informativní zpráva, že začalo stahování obrázku účtenky. Průběh stahování může uživatel sledovat v systémové notifikaci. Soubor účtenky bude uložen do stažených souborů. Pokud účtenka neobsahuje obrázek, ale PDF soubor, tak bude do zařízení skutečně uložen samotný PDF soubor a ne jeho fotografie.

Poslední možností v kontextovém menu je možnost sdílet účtenku. Ani tato možnost není vázaná žádnými podmínkami a je zde dostupná vždy. Po vybrání této možnosti se otevře systémový dialog pro sdílení obsahu. Uživatel si tedy může vybrat, zda chce účtenku sdílet přímo skrz nějakou aplikaci nebo zda si chce odkaz pouze zkopírovat do schránky. Při výběru sdílení přímo přes vybranou aplikaci se do této aplikace pošle pouze odkaz na tuto účtenku, který jde otevřít přímo v této aplikaci.

Pod navigační lištou se nachází náhled obrázku této účtenky. Tento náhled se chová stejně jako na obrazovce pro přidání nové účtenky. To znamená, že se při posouvání v seznamu zmenšuje a posouvá se nad navigační lištu, jak ukazuje obrázek 14. I na této obrazovce může uživatel potáhnout po zmenšeném obrázku prstem a stáhnout ho tím do původní pozice pod navigační lištou. Na obrázek je možné kliknout a otevřít tím celoobrazovkový náhled popsany v kapitole 4.5.1.

Pod náhledem účtenky se nachází seznam jednotlivých údajů rozdělený do několika sekcí pro větší přehlednost. První sekcí je unikátní číslo dokumentu. Tato sekce je zobrazena, pouze pokud účtenka již má nějaké číslo dokumentu. Účtenka má vygenerováno unikátní číslo dokumentu, pokud se nachází v uzavřeném měsíci.

Druhou velmi jednoduchou sekcí s údaji je sekce obsahující datum splatnosti účtenky. V této sekci se nachází pouze jediný údaj. Formát zobrazení data záleží na používané jazykové mutaci aplikace. Datum účtenky je zobrazeno vždy bez jakýchkoliv podmínek.

Třetí sekci v seznamu je cena. Tato sekce již obsahuje čtyři položky, kterými jsou: celková cena, cena bez DPH, reálná cena a měna. Tyto údaje jsou vždy vypsány pod názvem hodnoty v každé buňce. Jako všude v rámci aplikace jsou ceny vypsány s přesností na dvě desetinná místa. Pokud cena neobsahuje desetinnou část, tak je vypsána bez ní.

Další sekci jsou jednotlivé atributy účtenky. Jedná se tedy o všechny zbývající údaje, které si ke každé účtence systém ukládá. Na prvním místě je zobrazena kategorie účtenky spolu s informací, zda se jedná o účtenku s odpočitatelnou daní. Jako další je zobrazena informace o statusu účtenky a o tom, kdo tuto účtenku nahrál do systému. Jako poslední atribut se pak v seznamu nachází popis. Pokud nebyl popis vyplněn, pak se místo prázdného popisu zobrazí hodnota „N/A“.

Poslední sekci jsou různé akce, které lze provádět s celou účtenkou. Tato sekce je zobrazena pouze v případě, že měsíc není uzavřen a uživatel má právo měnit danou účtenku. Pokud účtenka ještě není schválená, první akcí v této sekci je akce schválení účtenky. Pokud účtenka již schválená byla, objeví se opačná akce pro zamítnutí účtenky.

Jako poslední v dříve zmíněné sekci akcí se nachází možnost smazat účtenku. Po kliknutí na tuto možnost se nejdříve objeví potvrzovací dialog. Po potvrzení se účtenka smaže a tato obrazovka se zavře.

4.5.1 Detail dokumentu či fotografie

Jak bylo zmíněno v předchozí kapitole, uživatel si může zobrazit detail obrázku vybrané účtenky po kliknutí na její náhled. Na tuto obrazovku se uživatel může dostat ze všech obrazovek, na kterých je zobrazen náhled fotografie účtenky. Druhou možností, jak se může uživatel dostat na tuto obrazovku, je výběr možnosti „Zobrazit obrázek“ na obrazovce s možnostmi práce s účtenkou.

Pokud se na tuto obrazovku dostane uživatel poslední zmíněnou možností, je možné, že obrázek účtenky ještě nebyl načtený. V tom případě se mu na chvíli zobrazí načítání a poté se ukáže požadovaný obrázek. Pokud by se uživatel na tuto obrazovku dostal bez připojení k internetu, tak se ukáže dříve načtený obrázek. Pokud obrázek ještě načtený nebyl, tak se zobrazí chybová hláška přes celou obrazovku vysvětlující, co se stalo. V takovém případě má uživatel možnost si obnovit internetové připojení a zkusit obrázek načíst znovu.

Po úspěšném načtení se na této obrazovce ukáže co největší fotografie účtenky. Fotografie vždy zachovává původní poměr stran, tudíž je možné, že nad a pod fotografií bude

nevyužitý prostor. Pokud je písmo na fotografii stále příliš malé, může si uživatel fotografii přiblížit pomocí běžných gest pro přibližování fotografií. Při přiblížení fotografie je využit celý dříve nevyužitý prostor na obrazovce.

Pokud soubor nahraný jako dokument účtenky nebyla fotografie, ale PDF soubor, zobrazí se úplně stejně, jako bylo popsáno výše. Stejně jako v případě fotografie si může uživatel obsah PDF souboru přiblížit pomocí gest pro lepší čitelnost. Jelikož PDF soubory mohou mít více stránek, bylo nutné, aby bylo možné v aplikaci prohlédnout všechny stránky souboru. Pro prohlížení dalších stránek stačí přejet prstem po obrázku aktuální stránky zprava doleva. Pro návrat na předchozí stránku stačí gesto provést v opačném směru.

4.6 Úprava účtenky

Další důležitou obrazovkou je obrazovka pro změnu údajů nahrané účtenky. Na tuto obrazovku je možné se dostat přímo z obrazovky detailu účtenky. Další možností, jak tuto obrazovku otevřít uvnitř aplikace, je pomocí obrazovky s možnostmi práce s účtenkou. Poslední možností, jak tuto obrazovku otevřít, je kliknutím na webový odkaz získaný přímo z webové aplikace.

Obrazovka pro úpravu účtenky je vizuálně skoro stejná jako obrazovka pro přidání nové účtenky. Jediná vizuální změna na této obrazovce je jiný název obrazovky v navigační liště. Pod navigační lištou se tedy také nachází náhled obrázku účtenky a při posunu ve formuláři se zmenší a odsune na navigační lištu.

Na rozdíl od obrazovky pro přidání nové účtenky se na této obrazovce již nemusí analyzovat obrázek účtenky. Místo toho se na této obrazovce na začátku načítají data účtenky, její fotografie a potřebné údaje pro rozbalovací nabídky ve formuláři.

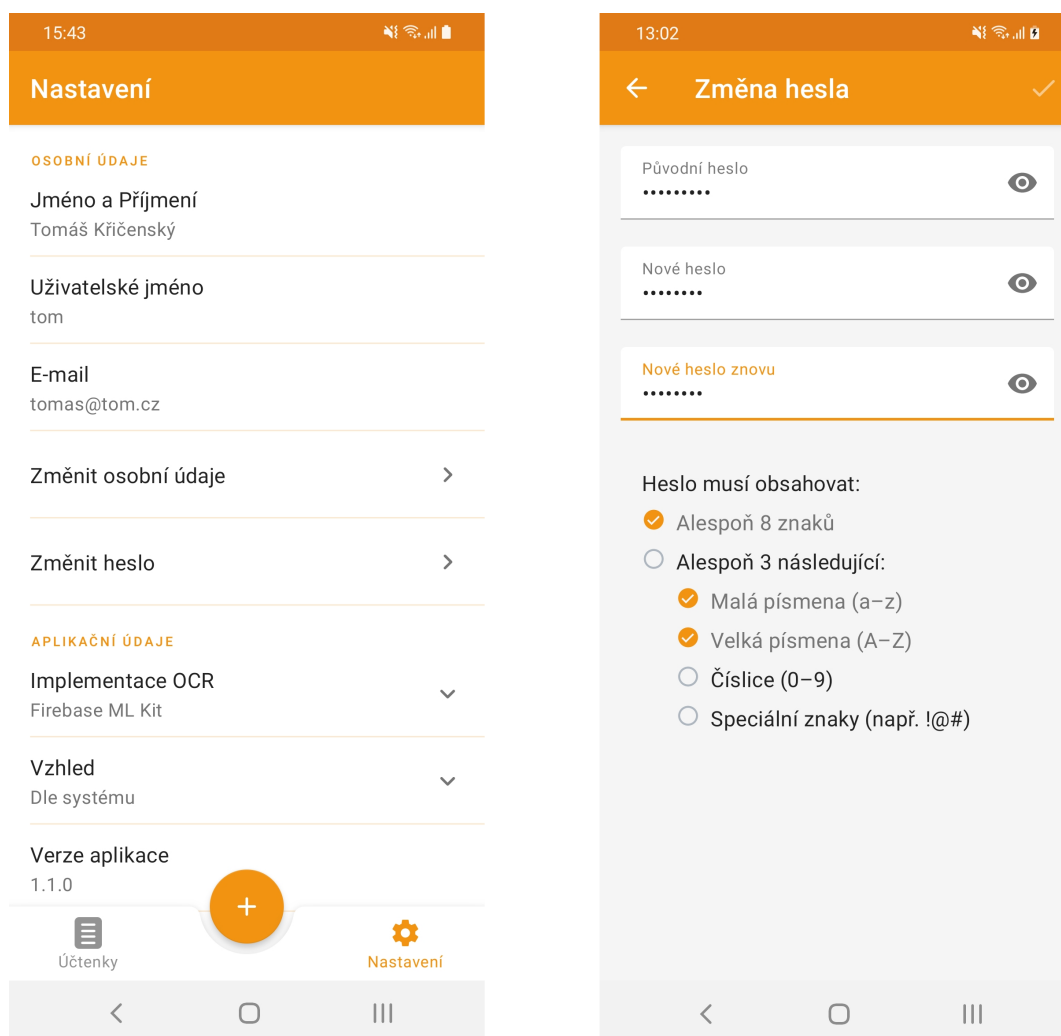
Pokud uživatel provede ve formuláři nějaké změny a poté se pokusí opustit obrazovku, zobrazí se potvrzovací dialog, aby nedošlo ke ztrátě dat.

4.7 Nastavení

Poslední sekci v této aplikaci je nastavení. Do nastavení je možné se dostat jednoduše pomocí tlačítka ve spodním navigačním menu. Sekce nastavení je složena ze tří různých obrazovek, které budou popsány v následujících podkapitolách.

První obrazovkou v sekci nastavení je obrazovka, na které je přehled všech údajů a vybraných hodnot nastavení. Tato obrazovka je rozdělena do tří sekcí dle typu obsahu. Většina této obrazovky je vidět vlevo na obrázku 15.

První sekce v seznamu se jmenuje „Osobní údaje“. Tato sekce, jak již název napovídá, obsahuje osobní údaje přihlášeného uživatele a možnost, jak je upravit. První tři položky této sekce jsou buňky, které vždy obsahují název dané položky a její hodnotu. Hodnota se vždy nachází pod názvem položky a je vypsána o něco menším písmem.



Obrázek 15 – Nastavení a změna hesla.

První sekce tedy obsahuje jméno a příjmení přihlášeného uživatele, jeho uživatelské jméno a také e-mail. Do této aplikace se uživatel přihlašuje pomocí uživatelského jména a hesla, tudíž je e-mail nepovinný údaj. Pokud uživatel e-mail nevyplnil, objeví se místo něj hodnota „N/A“. Tyto osobní údaje může uživatel kdykoliv změnit po kliknutí na čtvrtou

položku v sekci osobní údaje. Po výběru této možnosti se otevře nová obrazovka, která bude popsána v kapitole 4.7.1.

Poslední položka v sekci osobních údajů je možnost, která po kliknutí otevře novou obrazovku. Tentokrát se jedná o možnost změnit heslo právě přihlášeného uživatele. Tyto poslední dvě zmíněné položky, které otvírají novou obrazovku, vypadají jinak než běžné datové buňky. Tyto buňky obsahují pouze název akce a žádnou hodnotu. Název je vertikálně vycentrován. Pro zdůraznění, že se jedná o klikatelnou buňku, která otvírá novou obrazovku, je na pravé straně zobrazena ikona ve tvaru písmene „V“ otočená o 90 stupňů.

Druhá sekce nastavení se jmenuje „Aplikační údaje“. Tato sekce pro změnu obsahuje informace o verzi aplikace a také různé možnosti týkající se samotné aplikace. Buňky pro možnosti nastavení aplikace vypadají stejně jako standardní datové buňky, ale na pravé straně se nachází ikona ve tvaru písmene „V“. Tato ikona indikuje, že se jedná o klikatelnou buňku, která neotevřít novou obrazovku, ale zobrazuje další možnosti.

První takovou položkou v této sekci je položka pro změnu implementace OCR. Po kliknutí na buňku se otevře kontextové menu, ve kterém si uživatel může vybrat jednu ze čtyř možností. První možností je, že uživatel nechce OCR vůbec využívat. Tato možnost je zde pro ty, kteří nechtějí být zdržováni automatickým vyhledáváním údajů ve fotografiích nebo chtějí požadované údaje vždy vyplňovat sami. Druhou možností je „Firebase ML Kit“. Tato možnost je automaticky vybrána při instalaci aplikace. Třetí možností je „Tesseract“ a poslední je cloudová implementace Microsoft Cognitive Vision. Všechny výše zmíněné implementace jsou popsány v kapitole 3.

Druhou položkou v této sekci je možnost nastavení vzhledu aplikace. Vzhled aplikace je vždy světlý, nebo tmavý, ale uživatel má čtyři možnosti, dle kterých se určí, který vzhled má být použit. První a zároveň výchozí možností je, že se má vzhled aplikace automaticky přizpůsobit operačnímu systému Android. Tudíž pokud uživatel používá celý systém v tmavém režimu, tak by i tato aplikace byla tmavá. Jelikož možnost změny vzhledu celého operačního systému existuje až od Androidu verze 10, je vhodné dát uživateli možnost přizpůsobit vzhled aplikace, i když si nemůže upravit celý operační systém. Následující dvě možnosti tedy umožňují uživateli, aby si vynutil světlý či tmavý vzhled aplikace dle svého výběru. Poslední možností nastavení vzhledu aplikace je, aby se řídil režimem pro úsporu baterie. Jelikož OLED obrazovky spotřebovávají menší množství energie při zobrazování

tmavých povrchů, může být pro baterii přínosné, aby se aplikace přepnula do tmavého režimu, pokud uživatel chce ušetřit energii. [18]

Poslední položkou v sekci údajů o aplikaci je verze této aplikace. Jedná se o jednoduchou datovou buňku, na kterou se nedá kliknout. Ukazuje aktuální verzi nainstalované aplikace. Zobrazená verze využívá sémantické verzování. Jedná se o tři čísla, která jsou od sebe vždy oddělena tečkou. První číslo udává hlavní verzi a mění se, když nastane změna, která není zpětně kompatibilní. Druhé a třetí číslo je poté vždy méně významné. [35]

Poslední sekci a zároveň i klikatelnou položkou je možnost se z aplikace odhlásit. Po výběru této možnosti bude uživatel vrácen na přihlašovací obrazovku. Zároveň aplikace smaže všechna lokálně uložená data, aby jiný přihlášený uživatel neměl přístup k cizím citlivým údajům.

4.7.1 Změna osobních údajů

Druhou obrazovkou v sekci nastavení je obrazovka pro úpravu osobních údajů. Jak již bylo zmíněno dříve, na tuto obrazovku se uživatel dostane z hlavní obrazovky nastavení po kliknutí na buňku „Změnit osobní údaje“.

Na této obrazovce se nahoře nachází standardní navigační lišta, která obsahuje název obrazovky a možnost navigace zpět. Na pravé straně navigační lišty se také nachází ikona pro uložení změn osobních údajů.

Pod navigační lištou se nachází jednoduchý formulář, ve kterém se nachází tři textová pole pro úpravu osobních údajů. První textové pole obsahuje jméno a příjmení přihlášeného uživatele. Toto jméno se v aplikaci ukazuje u nahraných účtenek daným uživatelem.

Druhé textové pole obsahuje uživatelské jméno. Jedná se o jméno, které uživatel používá při přihlašování do aplikace. Ostatní uživatelé cizí přihlašovací jméno v aplikaci nevidí.

Poslední textové pole obsahuje e-mail uživatele. Tato položka je nepovinná a může tedy zůstat nevyplněná. Pokud ji ale uživatel začne vyplňovat, tak musí obsahovat validní e-mailovou adresu. Pokud zadaná adresa nebude validní, formulář nepůjde odeslat. E-mail není v této aplikaci použit pro přihlašování, ale jsou na něj zaměstnancům zasílány zprávy o nově nahraných účtenkách a podobně.

Po kliknutí na tlačítko v navigační liště pro uložení změn se pod touto lištou na chvíli zobrazí lineární ukazatel průběhu. Pokud uložení formuláře selhalo, tak se dole zobrazí informativní hláška, která vysvětlí, proč se tak stalo. Pokud bylo uložení úspěšné, tak se

aktuální obrazovka zavře a znovu se ukáže hlavní obrazovka s přehledem všech nastavení a na ní budou již aktualizované osobní údaje. Také se uživateli zobrazí informativní hláška o tom, že změny byly uloženy.

4.7.2 Změna hesla

Poslední samostatnou obrazovkou v sekci nastavení je obrazovka pro změnu hesla. Na tuto obrazovku se uživatel dostane po výběru možnosti „Změnit heslo“ na hlavní obrazovce nastavení. I na této obrazovce se nahoře nachází standardní navigační lišta. Vpravo se v této liště znovu nachází tlačítko pro uložení změn na této obrazovce.

Formulář pro změnu hesla také obsahuje tři textová pole. Všechna tato textová pole jsou upravena pro zadávání hesla. To znamená, že při psaní hesla se jednotlivé znaky mění na kolečko, aby nebylo možné heslo přečíst. Pokud by si uživatel chtěl napsané heslo zobrazit stačí kliknout na ikonu oka na pravé straně textového pole.

První textové pole je určené k zadávání původního hesla uživatele. Jedná se tedy o právě platné heslo. Aktuální heslo je vyžadováno, aby se nemohlo stát, že uživateli někdo jiný změnil heslo do aplikace z důvodu např. odcizení telefonu nebo nezamčené obrazovky. Pokud uživatel zadá původní heslo špatně, tak nebude možné heslo změnit a uživateli se zobrazí vysvětlující chybová hláška.

Druhé a třetí textové pole je určeno k vyplnění nového hesla do aplikace. Textové pole pro nové heslo je zde zobrazeno dvakrát, aby se nemohlo stát, že uživatel v novém hesle udělá chybu a uloží si nové heslo s chybou. Tudiž aby bylo možné odeslat tento formulář, obě textová pole musejí obsahovat stejné heslo.

Jelikož tato aplikace obsahuje citlivé údaje, hesla uživatelů musejí splňovat určité podmínky. Tyto podmínky jsou vypsány pod samotným formulářem, jak ukazuje obrázek 15. Při psaní hesla jsou podmínky kontrolovány a uživatel rovnou pod formulářem vidí, které podmínky již splnil. Pokud nejsou podmínky na bezpečnost hesla splněny, pak nemůže být nové heslo uloženo.

ZÁVĚR

Cílem diplomové práce bylo vytvoření mobilní aplikace, která dané společnosti usnadní evidenci účetních dokladů. Aplikace měla být určena především pro řadové zaměstnance společnosti a usnadnit jim proces nahrávání nových účetních dokladů do systému. Součástí vypracované aplikace mělo být automatické vyhledání některých dat ve fotografiích a PDF souborech účetních dokladů pro zrychlení přepisování údajů z dokladů do systému.

Všech výše zmíněných cílů bylo úspěšně dosaženo. Mobilní aplikace byla naprogramována a vydána v rámci společnosti. Aplikace tedy je již mezi reálnými uživateli a sbírá pozitivní ohlasy. Aplikace obsahuje všechny předem vytyčené funkce včetně správy účetních dokladů a automatického vyhledání dat v souborech účetních dokladů.

Automatického vyhledání dat v souborech účetních dokladů bylo dosaženo za pomoci optického rozpoznávání znaků (OCR). V rámci této práce byla implementována tři různá OCR řešení. Těmito řešeními byly ML Kit, Tesseract a Microsoft Computer Vision. Tyto implementace byly podrobeny testům přesnosti a rychlosti.

Testy přesnosti OCR ukázaly, že správné údaje byly nalezeny nejčastěji za použití OCR ML Kit. S pomocí této implementace bylo správně předvyplněno 72,94 % údajů. Zároveň bylo zjištěno, že pokud by byl v budoucnu vylepšen algoritmus pro výběr správné hodnoty ze seznamu nalezených, dosahovalo vyhledání hodnot pomocí ML Kit přesnosti až 89,42 %. ML Kit byl zároveň i nejrychlejší využitou implementací s průměrnou rychlostí zpracování jednoho účetního dokladu v plné velikosti 387 milisekund.

Do budoucna by bylo možné vypracovanou aplikaci rozšířit o možnosti pokročilé správy systému pro vedoucí pracovníky a vylepšený offline režim. Ten by uživatelům umožňoval přidání nových účetních dokladů bez přístupu k internetu. Další možné vylepšení by byla možnost nastavení výchozích hodnot pro uživatele, kteří stále nahrávají účetní doklady stejného typu.

POUŽITÁ LITERATURA

- [1] ČESKO. § 35 odst. 2 zákona č. 235/2004 Sb., o dani z přidané hodnoty. In: *Zákony pro lidi.cz* [online]. © AION CS 2010-2021 [cit. 19.7.2021]. Dostupné z: <https://www.zakonyprolidi.cz/cs/2004-235#p35-2>
- [2] wflow.com Czech Republic. Jak wflow.com funguje. In: *wflow.com* [online]. 2021 [cit. 19.7.2021] Dostupné z: <https://wflow.com/o-wflow/jak-to-funguje>
- [3] wflow.com Czech Republic. Cenové plány. In: *wflow.com* [online]. 2021 [cit. 19.7.2021] Dostupné z: <https://wflow.com/cena/>
- [4] INSIO Software. INSIO DMS. In: *INSIO Software* [online]. [cit. 21.7.2021] Dostupné z: <https://insio.cz/aplikace/dms/>
- [5] INSIO Software. INSIO Bezpapírová kancelář — Ukázka práce s objednávkami a fakturami. In: *YouTube* [online]. 12.11.2020 [cit. 21.7.2021] Dostupné z: <https://youtu.be/TUZ0NZBdnWY>
- [6] INSIO Software. Ceník. In: *INSIO Software* [online]. [cit. 21.7.2021] Dostupné z: <https://insio.cz/cenik/>
- [7] wflow.com Czech Republic. wflow.com. In: *Google Play* [online]. 15.7.2019 [cit. 21.7.2021] Dostupné z: <https://play.google.com/store/apps/details?id=com.wflow.app>
- [8] Intrasoft. INSIO software. In: *Google Play* [online]. 15.7.2019 [cit. 21.7.2021] Dostupné z: <https://play.google.com/store/apps/details?id=cz.intrasoft.mobileza>
- [9] LACKO, Luboslav. *Mistrovství — Android*. Brno: Computer Press, 2017. Mistrovství. ISBN 978-80-251-4875-4.
- [10] Google. Meet Google Play's target API level requirement. In: *Android Developers* [online]. 21.5.2021 [cit. 11.6.2021] Dostupné z: <https://developer.android.com/distribute/best-practices/develop/target-sdk>
- [11] JEMEROV, Dmitry a Svetlana ISAKOVA. *Kotlin in action*. Shelter Island, NY: Manning Publications Co., 2017. ISBN 978-161-7293-290.

- [12] SHAFIROV, Maxim. Kotlin on Android. Now official. In: *The Kotlin Blog* [online]. 17.5.2017 [cit. 11.6.2021] Dostupné z: <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/>
- [13] Google. Android's Kotlin-first approach. In: *Android Developers* [online]. 6.5.2021 [cit. 11.6.2021] Dostupné z: <https://developer.android.com/kotlin/first>
- [14] RAYWENDERLICH TUTORIAL TEAM, Filip BABIĆ a Nishant SRIVASTAVA. *Kotlin Coroutines by Tutorials: Mastering coroutines in Kotlin and Android*. Second Edition. USA: Razeware, 2019. ISBN 978-1942878681.
- [15] Koin. Koin DSL. In: *Koin* [online]. [cit. 11.6.2021] Dostupné z: <https://insert-koin.io/docs/reference/koin-core/dsl/>
- [16] SUBRAMANIAM, Venkat. *Programming DSLs in Kotlin: Design Expressive and Robust Special Purpose Code*. Raleigh: The Pragmatic Bookshelf, 2021. ISBN 978-1-68050-793-5.
- [17] RAYWENDERLICH TUTORIAL TEAM, Yun CHENG a Aldo OLIVARES DOMÍNGUEZ. *Advanced Android App Architecture: Real-world app architecture in Kotlin 1.3*. USA: Razeware, 2019. ISBN 978-1942878698.
- [18] Google. Dark theme. In: *Android Developers* [online]. 18.5.2021 [cit. 11.6.2021] Dostupné z: <https://developer.android.com/guide/topics/ui/look-and-feel/darktheme>
- [19] BANES, Chris. Building a Material Dark Theme on Android. In: *Material Design* [online]. 26.8.2020 [cit. 13.6.2021] Dostupné z: <https://material.io/blog/android-dark-theme-tutorial>
- [20] PORCELLO, Eve a Alex BANKS. *Learning GraphQL: Declarative Data Fetching for Modern Web Apps*. USA: O'Reilly Media, 2018. ISBN 978-1492030713.
- [21] Apollo Graph Inc. Normalized cache. In: *Apollo GraphQL Docs* [online]. [cit. 14.6.2021] Dostupné z: <https://www.apollographql.com/docs/android/essentials/normalized-cache/>

- [22] Google. CameraX architecture. In: *Android Developers* [online]. 8.6.2021 [cit. 15.6.2021] Dostupné z:
<https://developer.android.com/training/camerax/architecture>
- [23] Google. Why On-Device Machine Learning? In: *Google Developers* [online]. [cit. 18.6.2021] Dostupné z:
<https://developers.google.com/learn/topics/on-device-ml/learn-more>
- [24] Tesseract User Manual. In: *tessdoc* [online]. [cit. 17.6.2021] Dostupné z:
<https://tesseract-ocr.github.io/tessdoc/>
- [25] Traineddata Files for Version 4.00 +. In: *tessdoc* [online]. [cit. 18.6.2021] Dostupné z: <https://tesseract-ocr.github.io/tessdoc/Data-Files.html>
- [26] Google. ML Kit. In: *Google Developers* [online]. 2.6.2021 [cit. 18.6.2021] Dostupné z:
<https://developers.google.com/ml-kit/guides>
- [27] Google. Migration Guide. In: *Google Developers* [online]. 18.12.2020 [cit. 18.6.2021] Dostupné z: <https://developers.google.com/ml-kit/migration>
- [28] Google. Recognize text in images with ML Kit on Android. In: *Google Developers* [online]. 15.6.2021 [cit. 18.6.2021] Dostupné z:
<https://developers.google.com/ml-kit/vision/text-recognition/android>
- [29] HUMMEL, Guy a Jeremy COOK. What are the Benefits of Machine Learning in the Cloud? In: *Cloud Academy* [online]. 23.8.2018 [cit. 19.6.2021] Dostupné z:
<https://cloudacademy.com/blog/what-are-the-benefits-of-machine-learning-in-the-cloud/>
- [30] DEL SOLE, Alessandro. *Microsoft Computer Vision APIs Distilled: Getting Started with Cognitive Services*. Cremona: Apress, 2017. ISBN 978-1-4842-3341-2.
- [31] Microsoft. Cenová kalkulačka. In: *Microsoft Azure* [online]. 2021 [cit. 20.6.2021] Dostupné z:
<https://azure.microsoft.com/cs-cz/pricing/calculator/?service=cognitive-services>

- [32] Microsoft. Computer Vision API (v2.0). In: *Cognitive Services APIs Reference* [online]. 2017 [cit. 20.6.2021] Dostupné z: <https://westus.dev.cognitive.microsoft.com/docs/services/5adf991815e1060e6355ad44/operations/587f2c6a154055056008f200>
- [33] Google. Sheets: bottom. In: *Material Design* [online]. [cit. 11.6.2021] Dostupné z: <https://material.io/components/sheets-bottom>
- [34] Google. App shortcuts overview. In: *Android Developers* [online]. 18.5.2021 [cit. 11.6.2021] Dostupné z: <https://developer.android.com/guide/topics/ui/shortcuts>
- [35] PRESTON-WERNER, Tom. Semantic Versioning 2.0.0. In: *Semantic Versioning* [online]. [cit. 11.6.2021] Dostupné z: <https://semver.org/spec/v2.0.0.html>