

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Danilchanka** Jméno: **Aliaksandr** Osobní číslo: **469835**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Android aplikace pro správu úvěrů

Název bakalářské práce anglicky:

Android credit management application

Pokyny pro vypracování:

Navrhněte a vytvořte mobilní aplikaci pro správu úvěrů.

- 1) Vyhledejte a analyzujte existující systémy poskytující podobnou funkcionalitu.
- 2) Na základě vyhodnocení těchto poznatků, specifikujte požadavky na funkcionalitu navrhovaného systému.
- 3) Seznamte se s technologiemi potřebnými pro vytvoření mobilní aplikace.
- 4) Navrhněte vhodné grafické rozhraní aplikace.
- 5) Navrhněte a implementujte mobilní aplikaci.
- 6) Otestujte aplikaci včetně uživatelských testů a výsledky vyhodnoťte.
- 7) Práce vyhodnoťte a uveďte možná budoucí vylepšení aplikace.

Seznam doporučené literatury:

- [1] Pressmann R. S.: Software Engineering,
- [2] Vávru J., Ujbányai M.: Programujeme pro Android, Grada
- [3] <https://uvero.cz/aplikace-pro-spravu-financi-ktere-stoji-za-vyzkouseni/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D., kabinet výuky informatiky FEL

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

Datum zadání bakalářské práce: **14.02.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Android aplikace pro správu úvěrů

Aliaksandr Danilchanka

Vedoucí práce: Ing. Božena Mannová, Ph.D.

Studijní program: Softwarové inženýrství a technologie

květen 2020

Poděkování

Rád bych poděkoval vedoucí své práce, Ing. Boženě Mannové, za její podporu, kvalitní a cenné rady, trpělivost a velké zkušenosti se znalostí bakalářských projektů.

Prohlášení

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

V Praze dne 21.05.2020

.....

Abstract

My work deals with the design and implementation of Android applications for loan management. The basic requirements for this application are to provide secure access to an overview of the user's current loans. Using the application, users can find out the current status, an overview of all documents and announcements of important dates, which simplifies the process of establishing and managing loans.

The work describes the analysis of problems and the design of their solution on the Android platform.

Kotlin programming language and Android Studio environment were used in the development.

Abstrakt

Práce se zabývá návrhem a implementací Android aplikace pro správu úvěrů. Základní požadavky jsou dosáhnout zabezpečeného přístupu uživatelů k přehledu svých aktuálních úvěrů. Pomocí aplikace mohou uživatelé zjistit aktuální stav, přehled všech dokumentů i oznámení důležitých termínů, což zjednoduší proces založení a vedení úvěrů.

V práci je popsána analýza problémů a návrh jejich řešení na platformě operačního systému Android.

Při vývoji byl použit programovací jazyk Kotlin a vývojové prostředí Android Studio.

Obsah

1	Úvod	1
2	Rešerše	3
2.1	Motivace pro rešerši	3
2.2	Cíl rešerše	3
2.3	Podobné projekty a současný stav na trhu	3
2.4	Postup rešerše	4
2.5	Závěr rešeršní práce	4
3	Popis problému a specifikace cíle	5
3.1	Záměr projektu	5
3.2	Obsah celého projektu	6
3.2.1	Diagram nasazení	6
3.3	Cíl a obsah mé bakalářské práce	7
4	Analýza	9
4.1	Funkční požadavky	9
4.1.1	Přihlásit se.	9
4.1.2	Vytvořit PIN.	9
4.1.3	Alternativní přihlášení.	9
4.1.4	Zabezpečení vstupu do aplikace.	9
4.1.5	Zobrazení seznamu úvěrů.	10
4.1.6	Zobrazení nastavení.	10
4.1.7	Změna PINu.	10
4.1.8	Zapnutí/vypnutí oznámení.	10
4.1.9	Zapnutí/vypnutí alternativního přihlášení.	10
4.1.10	Navštívení webu společnosti.	10
4.1.11	Zavolat svému poradci.	10
4.1.12	Odhlásit se.	10
4.1.13	Otevření detailů úvěru.	10
4.2	Nefunkční požadavky	11
4.2.1	Vývojové prostředí.	11
4.2.2	Minimální verzi SDK (Software Developer's Kit).	11
4.2.3	Aplikační jazyk.	11
4.2.4	Programovací jazyk.	11

4.2.5	Orientace obrazovek.	11
4.2.6	Cílové zařízení.	11
4.3	Průchod aplikací	11
5	Návrh systému	13
5.1	Moduly	13
5.2	App modul	13
5.3	Balíky	14
5.3.1	loanapp	15
5.3.2	loanapp.feature.network.app	15
5.4	Složky	15
5.5	Modelové třídy	16
5.5.1	GenericServerException	16
5.5.2	Activation	16
5.5.3	Token	16
5.5.4	DeviceId	17
5.5.5	CreatePin	17
5.5.6	UserInfo	18
5.5.7	Pin	18
5.5.8	LoginCredentials	18
5.5.9	Loan	18
5.5.10	Doc	18
5.5.11	LoanDetail	18
5.6	API	18
5.6.1	Login	18
5.6.2	Create Pin	19
5.6.3	Loan List	19
5.6.4	Loan Detail	20
5.6.5	Init	20
5.6.6	Notification	21
5.6.7	Touch Id	21
5.6.8	Verify Pin	21
5.6.9	Change Pin	22
5.6.10	Errors	22
6	Realizace projektu	23
6.1	Použité nástroje	23
6.2	Použité knihovny	23
7	Realizace UI	25
7.1	UI prvky	25
7.1.1	EditText	25
7.1.2	Button	26
7.1.3	Toolbar	26
7.1.4	CardView	26
7.1.5	Switch	26

7.1.6	TabLayout	27
7.2	Obrazovky	27
7.2.1	Login Screen	27
7.2.2	Create Pin Screen	27
7.2.3	Login with Pin Screen	27
7.2.4	Loan List Screen	28
7.2.5	Loan Detail Screen	28
7.2.6	Settings Screen	29
7.2.7	WEB Screen	30
7.2.8	Change Screen	30
8	Testování	31
8.1	Uživatelské testování	31
8.1.1	Test č.1	31
8.1.2	Test č.2	32
8.1.3	Test č.3	32
8.1.4	Test č.4	32
8.1.5	Závěr testování	32
9	Zhodnocení provedené práce	35
10	Další návrh realizace	37
A	Seznam použitých zkratk	41
B	Obsah přiloženého CD	43

Seznam obrázků

3.1	Diagram nasazení	6
4.1	Navigace v aplikaci	12
5.1	Moduly aplikace	13
5.2	Obsah app modulu	14
5.3	Balíky aplikace	14
5.4	Složky aplikace	16
5.5	Diagram modelových tříd	17
7.1	Použité barvy	25
7.2	EditText	26
7.3	Tlačítka	26
7.4	Toolbar	26
7.5	CardViews	27
7.6	Switch	27
7.7	TabLayout	27
7.8	Login Screen	28
7.9	PIN Screen	28
7.10	Loan List Screen	29
7.11	Loan Detail Screen	29
7.12	Settings Screen	30
7.13	WEB Screen	30

Kapitola 1

Úvod

V současnosti jsou úvěrové produkty součástí našeho běžného života. Alespoň každý z nás slyšel nebo měl zkušenosti se splácením úvěrů. Ze zkušenosti mých rodičů a kamarádů vím, že to není jednoduchý proces. Sjednání úvěru je proces, který mimo jiné zahrnuje i nutnost klienta splnit celou řadu nejrůznějších předpokladů a doložit různé typy dokumentů. Ale tato skutečnost není až tak obtížná. Náročnější je čas, který musíme strávit komunikací s bankou či jiným poskytovatelem úvěru, nebo čekat ve frontách, abychom mohli získat aktuální informace o stavu svých úvěrů či žádostí.

Na základě svých softwarových znalostí a výsledků provedených pomoci dotazníků jsem navrhnul řešení pro tento problém. Řešení nejenom pomůže uživateli získat aktuální informace, ale navíc zjednoduší proces založení a splácení úvěrů. Uživateli aplikace poskytne možnost sledovat důležité termíny a poskytne potřebné informace o jejich plnění. Tím zajistí i včasné splátky úvěrů.

Kapitola 2

Rešerše

2.1 Motivace pro rešerši

V současnosti je mobilní zařízení nedílná součást našeho života. Většina z nás dnes častěji používá svůj mobilní telefon než klasický počítač. Vyhledáváme informace na internetu, objednávat si jídlo, kupujeme auta, letenky, hrajeme hry, sledujeme filmy a už jsme na to tak zvyklí, že si neumíme představit život bez toho, co máme ve svých rukách.

I když mnohdy myslíme, že máme ve svém mobilním zařízení vše co potřebujeme, stále existuje prostor, který není z pohledu aplikací plně pokrytý. Jeden z potenciálních prostorů na vylepšení je právě segment s úvěry. Moje aplikace má pomoci s řešením situací, které zákazník potřebuje vyřešit, ale nechce jimi ztrácet čas.

Mým cílem je, aby usnadnila především proces sjednání a vedení úvěru a aby měl uživatel pod kontrolou všechny své platby. Neklade si za cíl poskytovat všechny funkce.

2.2 Cíl rešerše

Tato rešerše má za cíl zjistit optimální funkční řešení, které by měla obsahovat mobilní aplikace, aby vyřešila kritické problémy, se kterými se setkávají lidé během zahájení a plnění úvěrů.

2.3 Podobné projekty a současný stav na trhu

V současné době na prostorech českého PlayMarketu¹ a AppStoru², není žádná mobilní aplikace, která by byla přímo profilovaná na práce s úvěry.

¹<https://play.google.com/store>

²<https://www.apple.com/ios/app-store/>

Existují aplikace, které umožňují výpočet procent a výšku splátky, takzvané kalkulačky úvěrů.

Na českém trhu je například bankovní aplikace Moneta banky³, která umožňuje sjednat půjčku prostřednictvím aplikace a peníze získat na svůj účet během 5 až 10 minut. Je to ale jediná mobilní aplikace na českém trhu, která umožňuje práci s úvěry, nicméně podmínkou je, že uživatel má založený účet v bance, a tato aplikace není určena pro neklienty dané banky.

2.4 Postup rešerše

Rešerše byla provedena v několika krocích.

V prvním kroku jsem připravil dotazník pro potenciální uživatele, kteří mají zkušenosti se zřízením úvěrů a dotazník pro ty, kteří pracují v bance či úvěrové společnosti, abych mohl na základě jejich odpovědí stanovit hlavní problémy, které mají zákazníci během zahájení a splácení úvěrů.

V druhém kroku jsem rozeslal dotazník.

V třetím kroku jsem zanalyzoval odpovědi a na základě toho jsem identifikoval hlavní problémy:

- Zákazníci mohou zapomenout vyplnit či doručit všechny potřebné dokumenty.
- Zákazníci musí splácet včas.
- Zákazníci mají problém pamatovat si důležité termíny.

2.5 Závěr rešeršní práce

Na základě rešerše bylo stanoveno navrhnout takovou aplikaci, která bude mít jednoduchý interface a ovládání, která umožní uživateli mít přehled o všech svých úvěrech a která by oznamovala uživateli důležité termíny, aby na to nezapomněl a neměl strach, že může zapomenout.

Aplikace bude informovat o částce a úrokové sazbě úvěru, tyto informace bude uchovávat a v případě ztráty mobilního zařízení zajistí jejich ochranu.

³<<https://play.google.com/store/apps/details?id=cz.moneta.smartbanka>>

Kapitola 3

Popis problému a specifikace cíle

3.1 Záměr projektu

Záměr celého projektu je navrhnout mobilní aplikaci, kterou budou používat zákazníci banky během celého procesu založení a splácení úvěrů.

Aplikace musí poskytovat jednoduchý přehled všech aktuálních úvěrů, aby uživatel neměl zbytečné informace o už splacených úvěrech a mohl se jednoduchým způsobem orientovat v seznamu a najít správný úvěr, o kterém chce zjistit jeho aktuální stav.

Aplikace musí poskytovat detailnější informace o úvěru:

- **Název úvěru** - aby uživatel přesně věděl o jakém úvěru jsou detailnější informace.
- **Zbývající částku** - aby uživatel měl informace, kolik mu ještě zbývá zaplatit pomocí posílání a přijímání zpráv.
- **Příští termín placení** - aby uživatel nezapomněl zaplatit včas.
- **Číslo svého kontraktu** - aby se v případě nejasnosti mohl uživatel obrátit na poradce přímo s konkrétním dotazem ohledně konkrétního úvěru.
- **Částku placení** - aby uživatel věděl kolik má zaplatit.
- **Sazba**
- **Datum začátku**
- **Datum konce**
- **Přehled všech důležitých termínů** - aby uživatel na nic nezapoměl.
- **Přehled zbývajících dokumentů** - aby uživatel věděl, jaké dokumenty ještě musí odevzdat.
- **Přehled zbývajících dokumentů**

Kromě uživatelských požadavků, se moje aplikace bude zaměřovat na poskytnutí spolehlivé komunikace se serverem, aby se uživatel nebál o ztrátu osobních údajů a aby byly zachovány práva zákazníka na ochranu osobních dat.

3.2 Obsah celého projektu

Cílem celého projektu bude realizace FE(Front-end) a BE(Back-end) služeb, které budou mezi sebou komunikovat pomocí REST API.

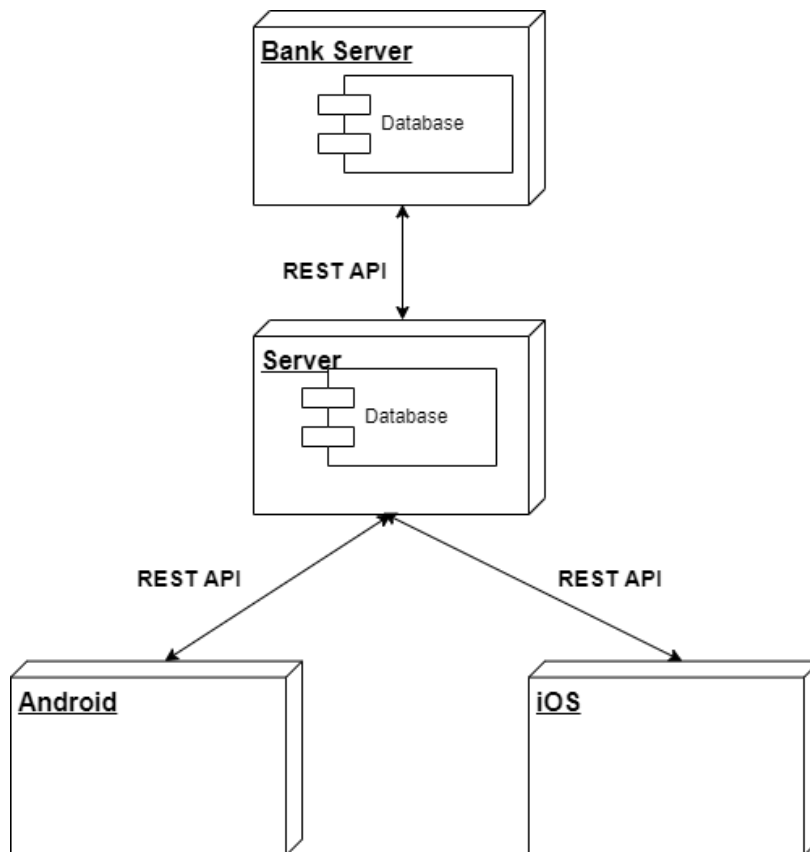
BE služba bude reprezentovaná Serverovou částí, která bude sloužit jako mezikrok v komunikaci FE se serverem zákazníka. Jejím hlavním úkolem bude generování a uložení přístupových tokenů, pomocí kterých zákazník bude získávat zabezpečený přístup k soukromým datům.

FE služba bude reprezentovaná mobilní aplikací, která bude realizována pro platformy iOS a Android. Hlavním úkolem aplikace bude získávání dat ze serveru zákazníka a umožnění jejich zobrazení uživatelům.

Po nasazení aplikace bude dostupná ke stažení ze servisu PlayMarket anebo AppStore.

3.2.1 Diagram nasazení

Závislosti v rámci celého systému jsou znázorněny na obrázku 3.1.



Obrázek 3.1: Diagram nasazení

3.3 Cíl a obsah mé bakalářské práce

Po dohodě s vedoucí práce jsem navrhl a implementoval mobilní aplikace pro Android OS (Operation System), která bude splňovat minimální požadavky zákazníka a grafické standardy pro vývoj od Googlu¹.

Obsah mé bakalářské práce:

- **Mobilní aplikace pro Android.**

Realizována aplikace bude obsahovat plně funkční kód, který bude získávat, zobrazovat a posílat data na server na základě navržené funkcionality.

- **Serverová část.**

Komunikace se serverem bude realizována pomocí **mockovacích** JSON(JavaScript Object Notation) dat přímo v aplikaci.

Výstupem práce bude funkční prototyp se zdrojovým kódem, který můžeme reprezentovat zákazníkům. Zdrojový kód bude flexibilně navržen pro další neomezený vývoj.

¹<<https://material.io/design/>>

Kapitola 4

Analýza

Tato kapitola bude obsahovat funkční a nefunkční požadavky aplikace, na základě kterých bude aplikace implementována.

4.1 Funkční požadavky

Funkční požadavky[12] objasňují, co se musí udělat a identifikují nutné úkony, aktivity a akce, které musí být vykonány. Analýza funkčních požadavků bude použita jako základ takzvané „top-level“ funkce systému pro funkční analýzu.

4.1.1 Přihlásit se.

Nepřihlášený uživatel bude mít možnost se přihlásit, přihlašovací údaje může získat u svého poradce.

4.1.2 Vytvořit PIN.

Po loginu se uživateli zobrazí obrazovka nastavení PIN kódu. Na ní si uživatel může vytvořit nový pin pro budoucí přihlášení do aplikace.

4.1.3 Alternativní přihlášení.

Na obrazovce nastavení PIN se uživateli automaticky zobrazí možnost alternativního přihlášení do aplikace pomocí otisku prstů za podmínky, že má na svém zařízení povolenou tuto alternativní přihlašovací metodu.

4.1.4 Zabezpečení vstupu do aplikace.

Aby aplikace byla zabezpečena i pro případ ztráty, bude podporovat přihlášení pomocí PINu nebo otisku prstů.

4.1.5 Zobrazení seznamu úvěrů.

Na hlavní stránce po přihlášení se uživatel načte a zobrazí seznam všech jeho úvěrů, kde bude zobrazen název úvěrů, příští důležitý termín splátky a zbývající částka, aby měl jasný přehled o aktuálním stavu tohoto úvěru.

4.1.6 Zobrazení nastavení.

Na hlavní stránce nahoře bude ikona "Settings". Jejím otevřením se uživatel dostane na obrazovku "Nastavení", kde bude mít možnost změnit PIN, zapnout/vypnout oznámení, zapnout/vypnout alternativní přihlášení pomocí otisků prstů, otevřít webovou stránku společnosti, zavolat svému poradci a odhlásit se.

4.1.7 Změna PINu.

Uživatel si může změnit PIN.

4.1.8 Zapnutí/vypnutí oznámení.

Uživatel si může zapnout nebo vypnout oznámení (tzv. „push notifikace“).

4.1.9 Zapnutí/vypnutí alternativního přihlášení.

Uživatel si může zapnout nebo vypnout alternativní způsob přihlášení.

4.1.10 Navštívení webu společnosti.

Uživatel může otevřít webovou stránku společnosti přímo v aplikaci.

4.1.11 Zavolat svému poradci.

Uživatel může zavolat svému poradci pomocí mobilní aplikace. Zmáčknutím tlačítka s číslem na obrazovce "Nastavení" se otevře výběrová obrazovka, kde může uživatel zvolit způsob volání. Po zvolení způsobu volání se mu otevře zvolená aplikace.

4.1.12 Odhlásit se.

Uživatel se může odhlásit pomocí tlačítka "Odhlásit" na obrazovce "Nastavení".

4.1.13 Otevření detailů úvěru.

Uživatel může otevřít detail svého úvěru, kde najde důležité informace o svém úvěru.

4.2 Nefunkční požadavky

Nefunkční požadavky[12] jsou požadavky, které kladou omezení na design a implementaci (například požadavky na výkonnost, standardy kvality, nebo designové omezení).

4.2.1 Vývojové prostředí.

Aplikace bude implementována ve vývojovém prostředí Android Studio.

4.2.2 Minimální verzi SDK (Software Developer's Kit).

Minimální verzi SDK jsem zvolil 21 (Android 5.0 Lollipop Direct).

4.2.3 Aplikační jazyk.

Aplikace bude realizována v anglickém jazyce.

4.2.4 Programovací jazyk.

Na základě informace z oficiální stránky Android Developers[5]:
"Write better Android apps faster with Kotlin. Kotlin is a modern statically typed programming language that will boost your productivity and increase your developer happiness."
a kvůli tomu, že od roku 2018 je Kotlin oficiální programovací jazyk pro Android, aplikace bude vyvíjena v programovacím jazyce Kotlin.

4.2.5 Orientace obrazovek.

Všechny obrazovky v aplikaci budou mít portrétovou orientaci.

4.2.6 Cílové zařízení.

Aplikace bude použitelná i na tabletech i na mobilních zařízeních.

4.3 Průchod aplikací

Klíčovým při analýze a návrhu aplikace je stanovit její průchod. Na základě funkčních a nefunkčních požadavků, byla navržena aplikační navigace 4.1. V souvislosti s tím, zda je uživatel přihlášen do aplikace nebo ne, budeme mít dva scénáře průchodu aplikací

- **Nepřihlášený uživatel.**

Po vstupu do aplikace se nepřihlášenému uživateli zobrazí obrazovka pro zadávání uživatelského jména a hesla pro vstup do aplikace.

Po úspěšném zadání vstupních údajů uživatele se přesměruje na obrazovku pro vytvoření PINu, kde musí vytvořit a potvrdit svůj přihlašovací PIN.

Dále již uživatel bude přihlášen a bude přesměrován na obrazovku "Seznam úvěrů".

- **Přihlášený uživatel.**

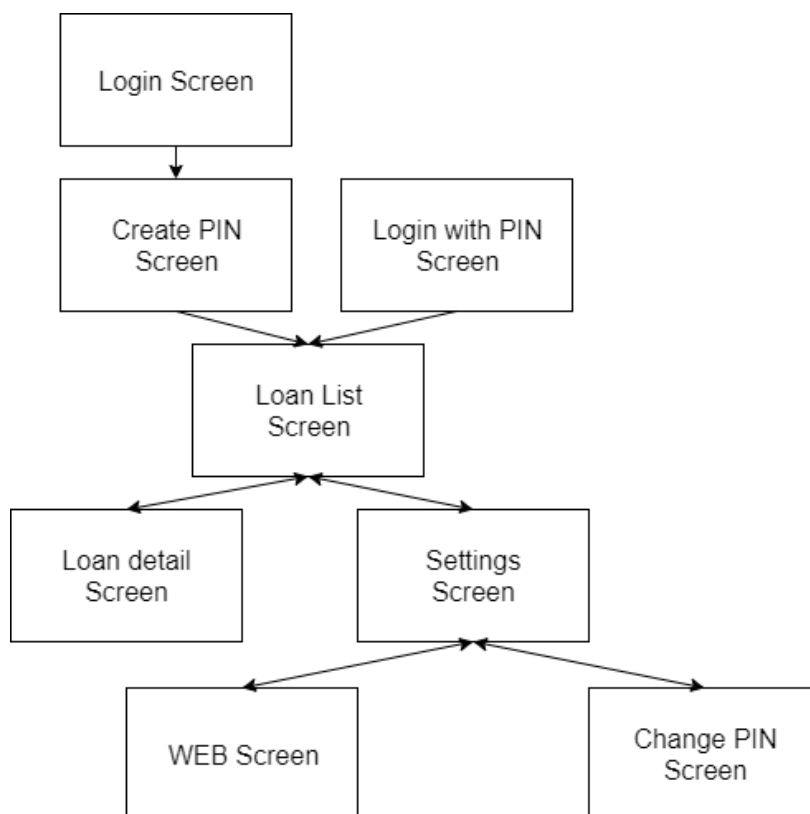
Po vstupu do aplikace se přihlášenému uživateli zobrazí obrazovka pro přihlášení pomocí PINu. Na této obrazovce se uživatel buď může přihlásit pomocí zadání svého uloženého PINu nebo, pokud má povoleno alternativní přihlášení, pomocí otisku prstu. Po úspěšném přihlášení se uživateli zobrazí obrazovka s přehledem všech jeho úvěrů, kde má možnost otevřít detail úvěru anebo zobrazit obrazovku "Nastavení".

Z obrazovky "Detail úvěru" se uživatel může vrátit pouze o krok zpět.

Z obrazovky "Nastavení" se uživatel může vrátit zpět na přehled všech úvěrů, otevřít webovou stránku společnosti anebo otevřít obrazovku "Změna PINu".

Z webu společnosti se uživatel může vrátit zpátky na obrazovku "Nastavení".

Na obrazovce pro změnu PINu, se může uživatel vrátit zpět na obrazovku "Nastavení" nebo může provést změnu PINu. Pro změnu PINu uživatel bude muset nejprve zadat svůj aktuální PIN, po úspěšném zadání aktuálního PINu bude muset vytvořit a potvrdit nový PIN. Po úspěšné změně PINu bude uživatel automaticky přesměrován na obrazovku "Nastavení".



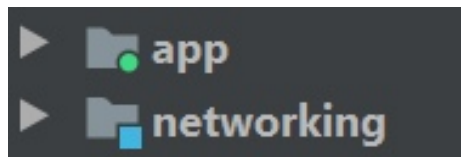
Obrázek 4.1: Navigace v aplikaci

Kapitola 5

Návrh systému

Tato kapitola popisuje navrženou strukturu modulů, balíčků, složek, modelových tříd a API aplikace.

5.1 Moduly



Obrázek 5.1: Moduly aplikace

app

Obsahuje aplikační prvky.

networking

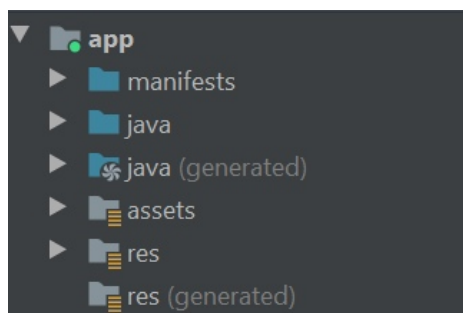
Obsahuje abstraktní třídy pro zpracování serverových dotazů a odpovědí.

5.2 App modul

manifests

Obsahuje AndroidManifest.xml soubor. Tento soubor slouží pro nastavení inicializace celého projektu. Jeho obsahem jsou:

- **uses-permission** popisuje povolení, které aplikace potřebuje, aby měla přístup k vnitřní funkcionalitě zařízení.
- **application** popisuje celou aplikaci. Její název, orientaci a všechny aktivity, které má.



Obrázek 5.2: Obsah app modulu

java

Obsahuje implementační kód celé aplikace.

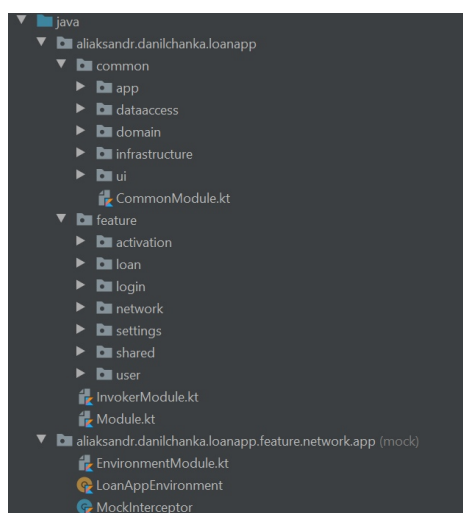
assets

Obsahuje mockovací API celé aplikace.

res

Obsahuje všechny zdrojové soubory celé aplikace.

5.3 Balíky



Obrázek 5.3: Balíky aplikace

5.3.1 loanapp

Obsahuje v sobě:

- **common balík.**
Balík, kde používané třídy napříč celou aplikaci.
- **feature balík.**
Balík, který v sobě obsahuje různé pojmenované složky. Složky jsou pojmenované podle části aplikace, za který odpovídají.
- **Třída InvokerModule.**
Je to DI(Dependency Injection) třída, která provádí injektování všech invoker tříd.
- **Třída Module.**
Je to DI třída, která provádí injektování všech DI modulů.

5.3.2 loanapp.feature.network.app

Obsah tohoto balíku se mění v souvislosti s aktuálním buildem. Balík obsahuje třídy, které popisují připojení, logování a podmínky komunikace se serverem.

5.4 Složky

app

Obsahuje v sobě controllery a resource třídy.

dataaccess

Obsahuje v sobě:

- **gateway**, kde jsou adaptéry a servisní třídy.
- **resource**, kde jsou DTO(Data Transfer Object) třídy.

domain

Obsahuje v sobě modelové třídy.

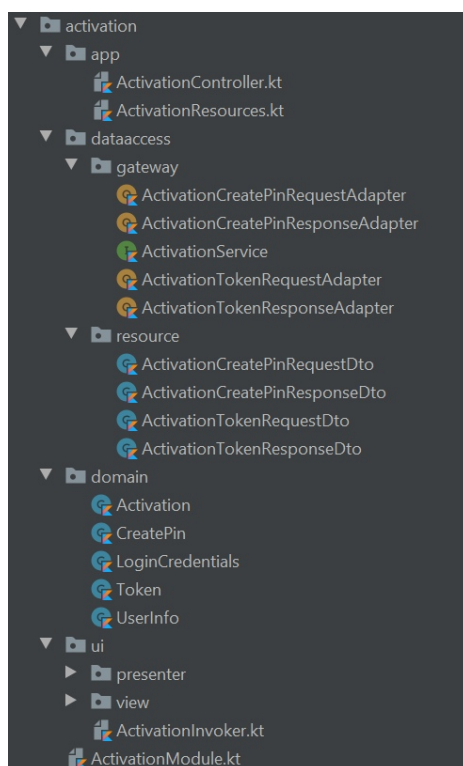
ui

Obsahuje v sobě:

- **presenter**, kde jsou presenery.
- **view**, kde jsou view třídy.
- **Třída Invoker**, kde jsou funkce pro zobrazení konkrétních obrazovek.

Třída Modul

Je to DI třída, která provádí injektování všech tříd této složky.



Obrázek 5.4: Složky aplikace

5.5 Modelové třídy

Na obrázku 5.5 je diagram všech navržených a použitých modelových tříd. Každá modelová třída má svůj účel a konkrétní místo použití.

5.5.1 GenericServerException

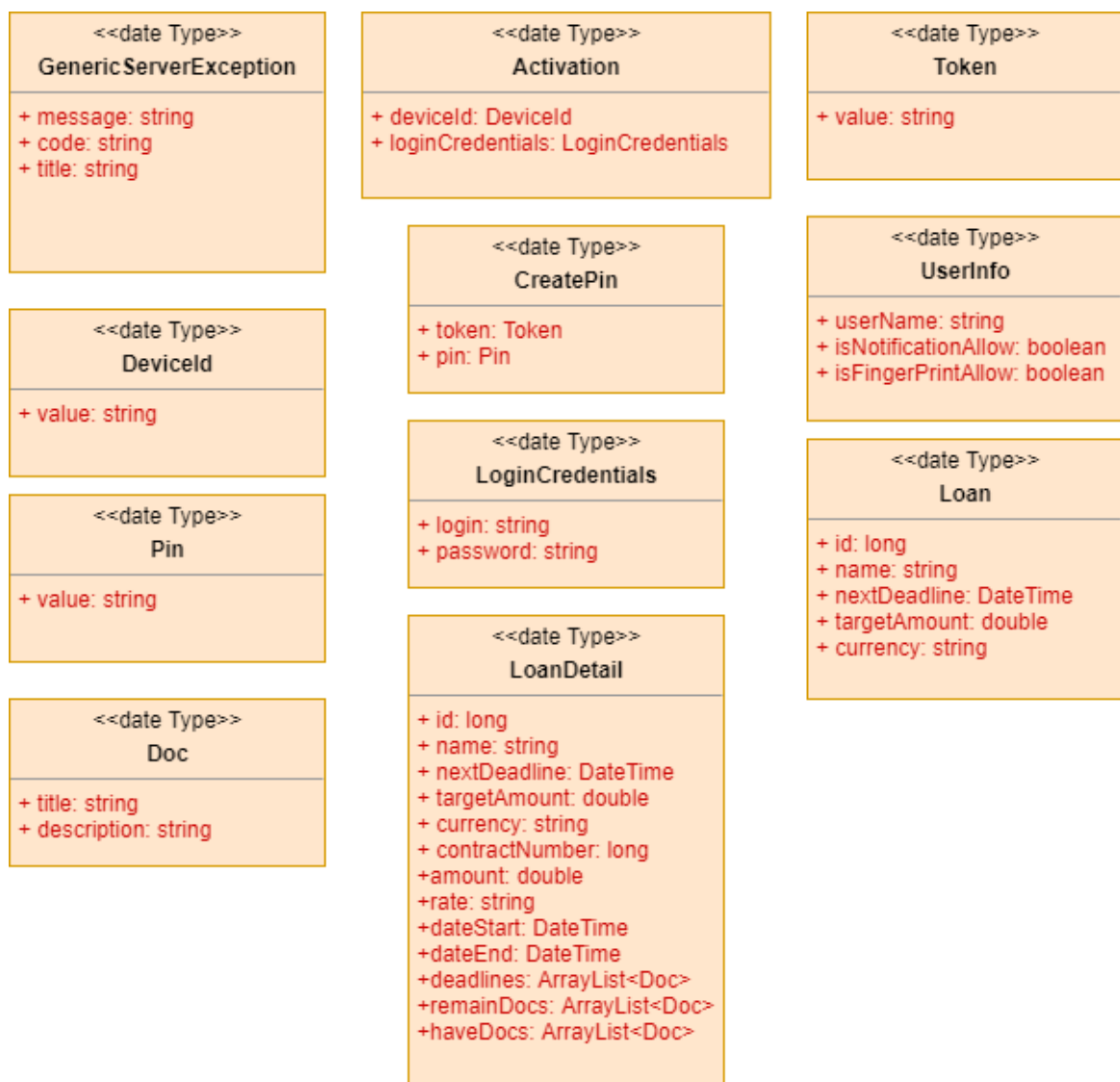
Třída určena pro práci s chybovými hlášeními, které vrací server.

5.5.2 Activation

Třída obsahující data, které potřebujeme posílat na server při loginu pomocí jména a hesla.

5.5.3 Token

Třída určena pro udržování hodnoty tokenu, kterou získáváme ze serveru.



Obrázek 5.5: Diagram modelových tříd

5.5.4 DeviceId

Třída určena pro udržování hodnoty deviceId, které jsme vygenerovali pro konkrétní zařízení.

5.5.5 CreatePin

Třída obsahující data, které posíláme na server při vytváření PINu.

5.5.6 UserInfo

Třída obsahující aktuální informace o uživateli, kterou získáváme ze serveru.

5.5.7 Pin

Třída určena pro udržování hodnoty PINu.

5.5.8 LoginCredentials

Třída určena pro udržování přihlašovacích údajů.

5.5.9 Loan

Třída obsahující krátkou informaci o úvěru, kterou získáváme ze serveru.

5.5.10 Doc

Třída obsahující informaci o konkrétním dokumentu.

5.5.11 LoanDetail

Třída obsahující detailnější informace o úvěru.

Každá modelová třída obsahuje buď primitivní atributy nebo modelové třídy. Jediný atribut, který není primitivní je `DateTime`. Tento atribut slouží pro udržování konkrétního data, který můžeme formátovat podle designových potřeb.

5.6 API

Na základě funkčních požadavků jsem navrhnul komunikaci se serverem. Komunikace bude probíhat pomocí REST (Representational State Transfer) API (Application Programming Interface) a JSON formátu.

5.6.1 Login

POST /login

Dotaz:

Body: { "login", "password", "deviceId" }

Odpověď:

```
200 (OK) { "token" }
```

Chybové odpovědi (viz [Errors](#))

5.6.2 Create Pin

POST /createPin

Dotaz:

```
Body: { "token", "pin" }
```

Odpověď:

```
200 (OK) { "userName", "isNotificationAllow", "isFingerPrintAllow" }
```

Chybové odpovědi (viz [Errors](#))

5.6.3 Loan List

GET /loans

Odpověď:

```
200 (OK)
```

```
[ {  
  "id",  
  "name",  
  "nextDeadline",  
  "targetAmount",  
  "currency"  
} ]
```

Chybové odpovědi (viz [Errors](#))

5.6.4 Loan Detail

GET /loan/{id}

Dotaz:

Path: **id** - Unikátní id úvěrů.

Odpověď:

200 (OK)

```
{
  "id",
  "name",
  "nextDeadline",
  "targetAmount",
  "currency",
  "contractNumber",
  "amount",
  "rate",
  "dateStart",
  "dateEnd",
  "deadlines": [{ "title", "description" }],
  "remainDocs": [{ "title", "description" }],
  "haveDocs": [{ "title", "description" }]
}
```

Chybové odpovědi (viz [Errors](#))

5.6.5 Init

POST /init

Dotaz:

Body: { "token", "pin" }

Odpověď:

200 (OK) { "userName", "isNotificationAllow", "isFingerPrintAllow" }

Chybové odpovědi (viz [Errors](#))

5.6.6 Notification

POST /notification

Dotaz:

Body: { "isAllow" }

Odpověď:

200 (OK) Empty Response

Chybové odpovědi (viz [Errors](#))

5.6.7 Touch Id

POST /touchId

Dotaz:

Body: { "isAllow" }

Odpověď:

200 (OK) Empty Response

Chybové odpovědi (viz [Errors](#))

5.6.8 Verify Pin

POST /verifyPin

Dotaz:

Body: { "pin" }

Odpověď:

200 (OK) Empty Response

Chybové odpovědi (viz [Errors](#))

5.6.9 Change Pin

POST /changePin

Dotaz:

Body: { "pin" }

Odpověď:

200 (OK) Empty Response

Chybové odpovědi (viz [Errors](#))

5.6.10 Errors

400 (Bad request) { "code", "message" }

401 (Unauthorized) { "code", "message" }

403 (Forbidden) { "code", "message" }

404 (Not Found) { "code", "message" }

500 (Server Error) { "code", "message" }

Kapitola 6

Realizace projektu

Projekt používá MVC (Model View Controller) architekturu a postaven na observable attributech, což usnadňuje reakce na změnu dat a jejich momentální zobrazení.

6.1 Použité nástroje

Na základě nefunkčních požadavků je aplikace realizována ve vývojovém prostředí Android Studio verze 3.6.1.

Pro konfiguraci a nastavení požadavků projektu je použit Gradle¹, který je realizován v programovacím jazyce Kotlin.

Protože Kotlin je oficiální programovací jazyk pro Android a mám s ním dva roky zkušenosti, pro realizaci projektu byl použit právě tento programovací jazyk, a to ve verzi 1.3.50.

Aby aplikace byla jednoduše realizována a zároveň spustitelná na 85% zařízeních, byla pro ni zvolena minimální verze SDK 21 (Android 5.0 Lollipop Direct).

6.2 Použité knihovny

Cleverlance Screens (*com.github.matko21:screens*)[1]
Knihovna od společnosti Cleverlance², která usnadňuje implementovat aplikaci postavenou na observable attributech.

RxBinding (*com.jakewharton.rxbinding3*)[9]
Knihovna od autora Jake Wharton³, která obsahuje v sobě observable reakce UI prvků na

¹<<https://developer.android.com/studio/build>>

²<<https://www.cleverlance.com>>

³<<https://jakewharton.com/>>

změnu jejich stavu.

Retrofit (*com.squareup.retrofit2*)[15]

Knihovna, která usnadňuje práci s REST API.

OkHttp3 (*com.squareup.okhttp3*)[14]

Knihovna, která spolu s Retrofitem slouží pro loggování údajů, které proběhají během komunikace se serverem.

RxJava2 (*io.reactivex.rxjava2*)[13]

Knihovna, která v sobě implementuje observable prvky a postup práce s nimi.

RxRelay2 (*com.jakewharton.rxrelay2*)[10]

Knihovna od autora Jake Wharton⁴, která v sobě obsahuje méně náročné prvky na použití observable prvků.

Primárně relay prvky se používají v prezentační vrstvě pro uchování stavu UI prvků.

Material (*com.google.android.material*)[7]

Knihovna, která povoluje použití Material Design⁵ prvků.

ConstraintLayout (*androidx.constraintlayout*)[4]

Knihovna, která povoluje snadnější navržení dobrého UI.

LifeCycle (*androidx.lifecycle*)[6]

Knihovna, která umožňuje vytváření komponentů, které reagují na životní cyklus jiných komponentů.

Biometric (*androidx.biometric*)[2]

Knihovna, která umožňuje práci s biometrickými údaji.

RecyclerView (*androidx.recyclerview*)[8]

Knihovna, která usnadňuje zobrazení seznamu UI prvků.

Browser (*androidx.browser*)[3]

Knihovna, která umožňuje použití webového prohlížeče uvnitř aplikace.

Koin (*org.koin*)[11]

Knihovna, která umožňuje vytváření DI objekty pomocí použití koin modulů.

⁴<https://jakewharton.com/>

⁵<https://material.io/>

Kapitola 7

Realizace UI

Stylování projektu realizováno pomocí barev. Všechny použité barvy, včetně jejich RGB (Red Green Blue) kódu, jsou na obrázku 7.1.



Obrázek 7.1: Použité barvy

7.1 UI prvky

Tato sekce popisuje použité UI prvky přes celou aplikaci a jejich účel.

7.1.1 EditText

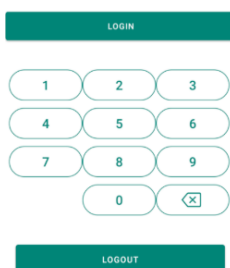
Standardní Android UI prvek, který je použitelný v případech, kdy potřebujeme vytvořit formulář na odesílání dat. Na obrázku 7.2 jsou EditTexty, které jsou použité v aplikaci.



Obrázek 7.2: EditText

7.1.2 Button

Standardní Android UI prvek, který je použitelný v případech, kdy potřebujeme vytvořit tlačítko a stylovat ho dle naší potřeby. Na obrázku 7.3 jsou tlačítka, která jsou použita v aplikaci.



Obrázek 7.3: Tlačítka

7.1.3 Toolbar

Standardní Android UI prvek, který používáme pro snadnější navigaci v aplikaci. Na obrázku 7.4 je Toolbar, který je použit na hlavní obrazovce "Seznam Úvěřů".



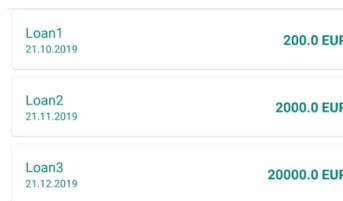
Obrázek 7.4: Toolbar

7.1.4 CardView

Android UI prvek, který používáme pro zvýraznění prvků na obrazovce pomocí stínu. Na obrázku 7.5 jsou CardView prvky, které jsou použité na obrazovce se seznamem úvěřů.

7.1.5 Switch

Android UI prvek, který používáme pro zapnutí/vypnutí funkcionality v aplikaci. Na obrázku 7.6 je Switch, který je použit na obrazovce "Nastavení".



Obrázek 7.5: CardViews



Obrázek 7.6: Switch

7.1.6 TabLayout

Android UI prvek, který dovoluje vytvořit několik záložek a přepínat mezi nimi. Na obrázku 7.7 jje TabLayout, který je použit na obrazovce "Detail úvěru".



Obrázek 7.7: TabLayout

7.2 Obrazovky

Tato sekce umožňuje přehled všech vytvořených obrazovek a popis akcí, které můžeme v rámci těchto obrazovek provádět.

7.2.1 Login Screen

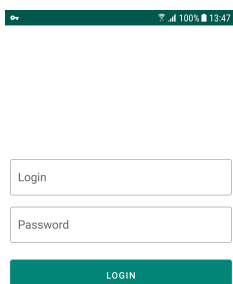
Obrazovka, 7.8 na které uživatel zadává své přihlašovací údaje a zmáčknutím tlačítka "Přihlásit se" odesílá údaje na server.

7.2.2 Create Pin Screen

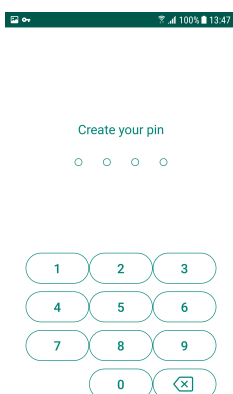
Obrazovka, 7.9 na které uživatel pomocí tlačítek 1 až 0 a tlačítka "smazat" vytvoří PIN a dále ho potvrdí.

7.2.3 Login with Pin Screen

Obrazovka, 7.9 na které uživatel pomocí tlačítek 1 až 0 a tlačítka "smazat" žádá svůj PIN a se přihlásí do aplikace.



Obrázek 7.8: Login Screen



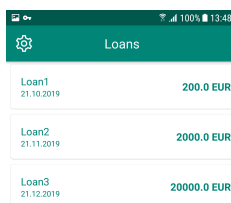
Obrázek 7.9: PIN Screen

7.2.4 Loan List Screen

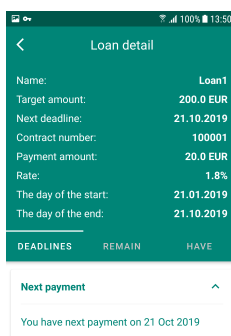
Obrazovka, [7.10](#) na které se uživateli zobrazí seznam všech jeho úvěrů. Zmáčknutím libovolného úvěru se uživateli zobrazí detail tohoto úvěru. Zmáčknutím tlačítka "nastavení" se uživateli zobrazí obrazovka "Nastavení".

7.2.5 Loan Detail Screen

Obrazovka, [7.11](#) na které se uživateli zobrazí aktuální informace o úvěru. Uživatel může přepínat mezi záložkami s dokumenty a budou mu zobrazené dokumenty, které patří do této záložky. Zmáčknutím libovolného dokumentu, v případě kdy tento dokument má popis, se uživateli zobrazí popis tohoto dokumentu. Zmáčknutím tlačítka "zpět" se uživatel vrátí na obrazovku "Seznam úvěrů".



Obrázek 7.10: Loan List Screen



Obrázek 7.11: Loan Detail Screen

7.2.6 Settings Screen

Obrazovka, [7.12](#) na které se uživatel zobrazí možnosti, které může v rámci nastavení provádět.

Zmáčknutím tlačítka "zpět" se uživatel vrátí na obrazovku "Seznam úvěrů".

Zmáčknutím UI prvku "Změna PINu" se uživatel zobrazí obrazovka "Změna PINu".

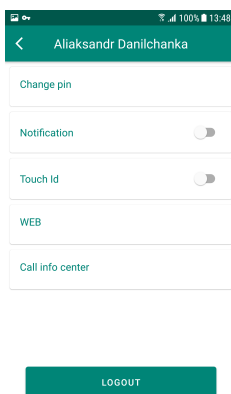
Zmáčknutím UI prvku "Oznámení" uživatel zapne/vypne oznámení v aplikaci.

Zmáčknutím UI prvku "Touch ID" uživatel zapne/vypne alternativní přihlášení pomocí otisku prstu.

Zmáčknutím UI prvku "WEB" se uživateli zobrazí obrazovka s webem společnosti.

Zmáčknutím UI prvku "Zavolat na call centrum" se uživateli otevře aplikace pro volání se zadaným číslem call centra společnosti.

Zmáčknutím UI tlačítka "Odhlásit se" se uživatel odhlásí z aplikace.



Obrázek 7.12: Settings Screen

7.2.7 WEB Screen

Obrazovka, 7.13 na které se uživateli zobrazí webová stránka společnosti. Zmáčknutím tlačítka "zavřít" se uživatel vrátí na obrazovku "Nastavení".



Obrázek 7.13: WEB Screen

7.2.8 Change Screen

Obrazovka, 7.9 na které uživatel pomocí tlačítek 1 až 0 a tlačítka "smazat" potvrdí svůj aktuální pin, vytvoří nový PIN a dále ho potvrdí. Zmáčknutím systémového tlačítka "zpět" se uživatel vrátí na obrazovku "Nastavení".

Kapitola 8

Testování

Tato kapitola je zaměřena na uživatelské testování aplikace.

Protože se má práce zabývat pouze implementací Android aplikace a serverová část je realizována pomocí mockovacích dat, provedl jsem pouze uživatelské testování.

8.1 Uživatelské testování

Během vývoje bylo prováděno testování všech funkcionalit a grafického zobrazení, ale ve skutečnosti programátor ne vždy dokáže zachytit všechny chyby a nedokonalosti svého produktu. Proto jsem navrhnul testovací scénář a zvolil jsem testovací skupinu uživatelů, které mají různé zkušenosti s používáním mobilních zařízení, abych mohl otestovat nejenom funkční část aplikace, ale i její GUI (Graphical User Interface).

Na začátku každého testu jsem seznámil testovaného uživatele se základní funkcionalitou aplikace. Dále jsem mu ukázal seznam úkolů, které potřebuje postupně splnit:

1. Úspěšně se přihlásit do aplikace.
2. Otevřít popis libovolného dokumentu, který mu ještě zbývá a který už má.
3. Zavolat do call centra.
4. Odhlásit se.

8.1.1 Test č.1

Testovaný: studentka ekonomického oboru univerzity VŠO. Věková skupina 20-35. (zařízení: LG Nexus 5, Android 5.1.0)

Všechny úkoly splnila rychle. Žádnou nápovědu nepotřebovala.

Zpětná vazba: "Úkoly byly jednoduché, kvůli každodennímu používání bankovních aplikací, které mají dost podobný design, ale během testování se objevila několik textových chyb."

8.1.2 Test č.2

Testovaný: student oboru FIT univerzity ČVUT. Věková skupina 20-35. (zařízení: Samsung Galaxy S6, Android 7.0)

Všechny úkoly splnil rychle. Žádnou nápovědu nepotřeboval.

Zpětná vazba: "Narazil jsem na to, že v aplikaci chybí přihlášení pomocí otisku prstu, ale během testování při pokusu zapnout Touch Id jsem se dozvěděl, že tato funkce bude implementována v příští verzi."

8.1.3 Test č.3

Testovaný: žena v domácnosti. Má zkušenosti s úvěrem. Věková skupina 35-50. (zařízení: Samsung Galaxy S6, Android 7.0)

Všechny úkoly splnila poměrně rychle. Žádnou nápovědu nepotřebovala.

Zpětná vazba: "Dosud nejsou implementované všechny funkce v aplikaci, ale až bude v provozu, byla by to aplikace, kterou bych ráda vyzkoušela v reálném životě."

8.1.4 Test č.4

Testovaný: UI tester. Má zkušenosti s testováním mobilních aplikací. Věková skupina 20-35. (zařízení: Google Pixel 4, Android 10.0)

Všechny úkoly splnil rychle. Žádnou nápovědu nepotřeboval.

Zpětná vazba: "Z grafického pohledu to není dobrá aplikace, ale až po testování jsem se dozvěděl, že při implementaci byly použité standardní barvy a UI prvky bez stylování. Narazil jsem na UI chybu, že při pokusu otevřít popis dokumentu otevření probíhá posouváním názvu nahoru, což není z hlediska UI dobře."

8.1.5 Závěr testování

Během testování se objevily textové a UI chyby, které jsem opravil.

Na základě zpětné vazby, jsem vytvořil seznam věci, které ještě chybí:

1. Aplikace je pouze prototyp a pro svůj provoz potřebuje mít spuštěný server, aby mohla v plně míře fungovat.
2. Aby aplikace byla konkurenceschopná na trhu, potřebuje mít dobrého designéra, který by navrhnul design, aby odpovídal standardům a současným trendům.

Na základě testování jsem provedl tento závěr:
Aby aplikace byla úspěšná na trhu v roce 2020 potřebuje tým odborníků, designerů a kreativců, kteří dokáží vytvořit kvalitní produkt, který očekává zákazník. Protože moje aplikace je zatím jen funkční prototyp, nemohla splnit všechny požadavky uživatelů, nicméně z pohledu funkcionalit potvrdili, že je aplikace funkční.

Kapitola 9

Zhodnocení provedené práce

Cílem mého projektu bylo navrhnout a implementovat Android aplikaci pro správu úvěrů.

Na základě požadavků, které jsem navrhl, jsem vytvořil plně funkční prototyp, který může být prezentovat zákazníkovi, jako 80% hotový produkt.

Podařilo se mi realizovat většinu požadavků. Z navržených požadavků chybí doplnit dva, tyto požadavky nemohou být realizované bez funkčního serveru.

1. Notifikace.
2. Přihlášení pomocí otisku prstu.

Tyto požadavky mohou být vyřešeny v další fázi vývoje za podmínky, že bude existovat plně funkční server, který umožní realizovat uložení otisku prstu a notifikační systém, který bude posílat oznámení na konkrétní zařízení.

Pro realizaci spojení se serverem a rozběhnutí projektu na reálných datech stačí přidat další typ Android buildu a změnit nastavení spojení ve třídách balíku [loanapp.feature.network.app](#)

Kapitola 10

Další návrh realizace

Aplikace, která byla realizována v rámci Bakalářské práce má, kromě toho co už bylo popsáno, další velký potenciál. Dalšími zajímavými funkcionalitami mohou být:

- **Zprávy** - aplikace může mít záložku "Zprávy", kam úvěrová společnost může přímo posílat dopisy se soukromou informací.
- **Dokumentace** - záložka, kam úvěrová společnost může ukládat důležité dokumenty k podepsání a odkud si uživatel může stahovat dokumenty uložené úvěrovou společností.
- **Chat s poradcem** - aplikace může obsahovat chat s poradcem, kde uživatel může přímo komunikovat se svým poradcem v případě dotazů a nejasností.
- **Možnost zaplacení** - uživatel bude mít možnost zaplatit svůj úvěr přímo v aplikaci.
- **Možnost vyhledávání a podepsání nového úvěru** - aplikace bude podporovat možnost uzavřít nový úvěr podle vlastních požadavků, a to zcela online a bez nutnosti osobního kontaktu s poradcem. Uživatel si bude moci nastavit filtr s preferencemi, které očekává od svého úvěru a aplikace mu nabídne možnosti.

Navrženou základní aplikaci, která zobrazuje přehled úvěrů a zasílá zprávy o důležitých termínech, můžeme rozšířit na zajímavý a užitečný produkt, který pomůže uživateli vyřešit problémy s úvěry: ztráta času cestami do úvěrové společnosti nebo strach, že něco zapomněl, a že nestihne včas odevzdat důležitou dokumentaci. Tuto správu úvěrů bude mít uživatel skutečně ve svých rukách.

Literatura

- [1] Cleverlance s.r.o. *Cleverlance Screen* [online]. stav z 10. 04. 2020. Dostupné z: <<https://github.com/Cleverlance/screens>>.
- [2] Google. *Biometric* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com/jetpack/androidx/releases/biometric>>.
- [3] Google. *Browser* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com/jetpack/androidx/releases/browser>>.
- [4] Google. *ConstraintLayout* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>>.
- [5] Google. *Android Developers* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com>>.
- [6] Google. *Lifecycle* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com/jetpack/androidx/releases/lifecycle>>.
- [7] Google. *Material Design Components* [online]. stav z 25. 3. 2020. Dostupné z: <<https://github.com/material-components/material-components-android/tree/master/lib/java/com/google/android/material>>.
- [8] Google. *Recyclerview* [online]. stav z 25. 3. 2020. Dostupné z: <<https://developer.android.com/jetpack/androidx/releases/recyclerview>>.
- [9] Jake Wharton. *RxBinding* [online]. stav z 10. 04. 2020. Dostupné z: <<https://github.com/JakeWharton/RxBinding>>.
- [10] Jake Wharton. *RxRelay* [online]. stav z 10. 04. 2020. Dostupné z: <<https://github.com/JakeWharton/RxRelay>>.
- [11] OpenCollective. *Koin* [online]. stav z 25. 3. 2020. Dostupné z: <<https://github.com/InsertKoinIO/koin>>.
- [12] Příspěvatelé Wikipedie. *Analýza požadavků* [online]. Jazyk Čeština, stav z 12. 1. 2020. Dostupné z: <https://en.wikipedia.org/wiki/Requirements_analysis>.
- [13] RxJava Contributors. *RxJava* [online]. stav z 10. 04. 2020. Dostupné z: <<https://github.com/ReactiveX/RxJava>>.

- [14] Square Inc. *OkHttp* [online]. stav z 10.04.2020. Dostupné z: <<https://github.com/square/okhttp>>.
- [15] Square Inc. *Retrofit* [online]. stav z 10.04.2020. Dostupné z: <<https://square.github.io/retrofit/>>.

Příloha A

Seznam použitých zkratk

FE Front-end

BE Back-end

REST Representational State Transfer

API Application Programming Interface

JSON JavaScript Object Notation

AS Android Studio

OS Operation System

SDK Software Developer's Kit

DI Dependency Injection

MVC Model View Controller

UI User Interface

DTO Data Transfer Object

RGB Red Green Blue

GUI Graphical User Interface

Příloha B

Obsah přiloženého CD

| source | LoanApp

| LoanApp.apk

| LoanApp.pdf

| readme.txt

Android Studio projekt se zdrojovým
kódem aplikace

Soubor pro instalaci aplikace

Bakalářská práce

Popis obsahu CD