



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Analyzátor finančních transakcí
Student:	Michal Brka
Vedoucí:	Ing. Jan Dufek
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Cílem práce je vytvoření standalone aplikace, která umožní nahrát výpisy z bankovního útu různých institucí a na jejich základech zobrazí zajímavé finanční statistiky i grafy. Uživatel bude mít možnost filtrovat vstupní data, nastavit časové období nebo granularitu. Aplikace bude vyvíjena jako open source v programovacím jazyce Java.

Součástí práce je také analýza typů statistik a grafů, které by mohly být pro uživatele zajímavé.

Metodika:

- 1) Proveďte analýzu typů statistik a grafů a vyhodnoďte, které by měly být použity.
- 2) Zhodnoďte a porovnejte možná existující řešení.
- 3) Proveďte analýzu možných technologií a vyberte nejvhodnější.
- 4) Udelejte návrh vzhledu a rozmístění ovládacích prvků.
- 5) Proveďte návrh implementace.
- 6) Implementujte návrh řešení.
- 7) Řešení otestujte na reálných uživateli.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 12. prosince 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVA



Bakalárska práca

Analyzátor finančných transakcií

Michal Brka

Vedúci práce: Ing. Jan Dufek

14. mája 2017

Pod'akovanie

Chcel by som poďakovať všetkým priateľom, známym a blízkej rodine, ktorá stála pri mne a podporovala ma pri tvorení bakalárskej práce. Pod'akovanie patrí hlavne vedúcemu práce Ing. Jánovi Dufekovi za vedenie, trpezlivosť a cenné rady.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 14. mája 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Michal Brka. Všetky práva vyhrazené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Brka, Michal. *Analýzátor finančných transakcií*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Práca je zameraná na návrh a implementáciu aplikácie, ktorá pomôže užívateľom získať rýchly prehľad o finančných transakciách na bankových účtoch prostredníctvom grafových a mapových vizualizátorov. Čitateľ sa postupne oboznámi s aktuálnymi formami riešenia od viacerých subjektov, analýzou moderných frameworkov, návrhom modernejšej, užívateľsky prívetivejšej aplikácie a v neposlednom rade aj konečnou implementáciou zvoleného riešenia.

Kľúčová slova finančné transakcie, analyzátor, mapy, grafy, java, javaFX, open source, frameworky, csv, xls

Abstract

Main goal of this bachelor thesis is to design and implement application, which will help users gain quick view on their's financial transactions by means of charts and maps. Reader is gradually acquainted with the actual forms of solutions from various subjects, analysis of modern frameworks, design of modern, user friendly application and last but not least with final implementation of selected solution.

Keywords financial transactions, analyzer, maps, charts, java, javaFX, open source, frameworks, csv, xls

Obsah

Úvod	1
1 Cieľ práce	3
2 Analýza	5
2.1 Standalone vs client-server aplikácie	5
2.2 Open source	6
2.3 Využitie analyzátorov	6
2.4 Aktuálne formy riešenia	7
2.5 Výber technológií	9
2.6 Grafy - vizualizácia dát	12
2.7 Dátové formáty	13
2.8 Podporované bankové subjekty	15
2.9 Štatistiky z transakcií	15
3 Návrh	19
3.1 Analýza požiadaviek	19
3.2 Návrh databázy	21
3.3 Diagram balíčkov	22
4 Implementácia	25
4.1 Dátové typy	25
4.2 Backendové časti aplikácie	26
4.3 Frontendové časti aplikácie	29
5 Testovanie	35
5.1 Testovanie na OS	35
5.2 Testovanie SQL dotazov	35
5.3 Testovanie na užívateľoch	35
5.4 Testovanie funkčných a nefunkčných požiadaviek	35

6	Perspektíva	37
6.1	Perspektíva XML formátu – jednotný štandard pre elektronické výpisy	37
6.2	Perspektíva AFT	38
	Záver	39
	Literatúra	41
A	Zoznam použitých skratiek	45
B	Obsah priloženého CD	47

Zoznam obrázkov

2.1	Prima banka - Peňaženka	8
2.2	HomeBank	9
2.3	Koláčový graf	17
2.4	Stĺpcový graf	17
3.1	Use case diagram	20
3.2	Logický model databázy	21
3.3	Relačný model databázy	21
3.4	Diagram balíčkov	23
4.1	Factory pattern - bankový model	28
4.2	Diagram tried - Filtre	29
4.3	Hlavné menu	31
4.4	Súborové menu	32
4.5	Grafové menu s chybami	33
4.6	Grafové menu bez chýb	33
4.7	Vlastný filter	34
4.8	Obrazovka s grafom	34

Úvod

Dnešok si stále viac vyžaduje rýchle, transparentné a aktuálne informácie, ktoré tvoria základnú podmienku pre správne rozhodnutia. Informačné technológie sa stali najrýchlejšie sa rozvíjajúcim priemyselným odvetvím, stali sa bežnou súčasťou každodenného života. Stretávame sa s nimi takmer všade. Špecifické požiadavky na informačné systémy majú v dnešnej dobe aj banky. Neodmysliteľnou súčasťou nášho života sú peniaze, banky, ich produkty a služby. Pretože bežný život každého z nás je poznačený digitalizáciou, v aktuálnom trende kráčajú aj banky. Len v nedávnej minulosti bankovní klienti dostávali výpisy z účtov v papierovej forme. Dnes, v dobe internetu a mobilov, sú tieto služby silno previazané s digitálnou platformou. Banky ponúkajú svojim klientom elektronické bankovníctvo, mobilné aplikácie s cieľom zlepšenia užívateľského komfortu a rýchlosti v poskytovaných bankových službách. Klienti tak majú svoju banku k dispozícii 7 dní v týždni a 24 hodín denne.

Cieľom bakalárskej práce je navrhnúť a vytvoriť softvérovú aplikáciu na analýzu finančných transakcií z pohybov na bankových účtoch. Dôraz je kladený predovšetkým na vytvorenie systému s jednoduchým ovládaním, ktorý bude open-source zameraný a bude umožňovať analyzovať finančné dáta z rôznych bankových výpisov súčasne. Postup riešenia je zhrnutý v niekoľkých krokoch. Najprv sa bakalárska práca venuje základným pojmom, ktoré sa používajú pri návrhu databáz a programátorských technológií. Po analýze súčasného stavu sú vyšpecifikované požiadavky na softvérovú aplikáciu. V ďalšom kroku je vytvorený model Analyzátoru finančných transakcií, návrh databázy, diagram balíčkov a jednoduché menu. Po vykonaní spomenutých bodov nasleduje implementácia systému v prostredí JavaFX, v závere prebehne testovanie na vybratých operačných platformách.

Cieľ práce

V súčasnosti existuje niekoľko aplikácií, ktoré slúžia na zobrazovanie a analýzu dát z bankových výpisov. Hlavným problémom týchto aplikácií je zameranie na jednu banku a uzavretosť projektu voči zmenám. Cieľom tejto bakalárskej práce je vytvoriť stand-alone počítačovú aplikáciu, ktorá bude open-source a aplikáciou bude možné analyzovať dáta z rôznych bankových výpisov súčasne. Medzi hlavné priority a zmeny voči existujúcim projektom zameraných na analýzu finančných transakcií budú patriť:

- Vedenie projektu ako open-source s možnosťou pridávania nových filtrov, vizualizátorov a bánk. V rámci projektu bude stačiť pridať jednu novú triedu na parsovač z novej banky.
- Vytvorenie užívateľsky-priateľského rozhrania. Projekt bude čerpať z aktuálnych UI trendov a pobeží na novej JavaFX platforme.
- Multiplatformovosť. Aplikáciu bude možné spustiť na rôznych operačných systémoch Windows, Linux a MAC OS s nainštalovaným JRE balíčkom verzií aspoň 8.0.

Analýza

V tejto kapitole sa čitateľ dozvie podrobnejšie o dôvodoch: prečo sa realizácia projektu zvolila formou standalone a nie typickou client-server formou, pozitívach a negatívach open source, využiteľnosti analyzátorov finančných transakcií, moderných technológiách zvolených pre tvorbu užívateľského rozhrania a databázy, možnostiach vizualizácií dát, formátoch dátových vstupov. Záver kapitoly priblíži čitateľovi štatistické možnosti, ktoré sa dajú vyfiltrovať z bankových transakcií.

2.1 Standalone vs client-server aplikácie

Hoci sú v dnešnej dobe veľmi populárne client-server aplikácie, prinášajú bezpečnostné riziko, ktoré pri nedôkladnom návrhu a implementácii môže spôsobiť problémy. Práve pre bezpečnostné riziko a prácu s citlivými dátami (finančné transakcie) sa finálne zvolila standalone forma.

2.1.1 Standalone[1]

- + Rýchlejší začiatkový vývoj aplikácie, hlavne GUI.
- + Bezpečnosť dát. Dáta nie sú uložené na verejnom serveri, ktorý pri nesprávnej manipulácii predstavuje riziko úniku informácií.
- + Flexibilita aplikácií.
- + Výkonnosť aplikácie nie je ovplyvnená inými faktormi (internetové pripojenie).
- Manuálne schválenie aktualizácií.
- Nie sú až tak ľahko portovateľné ako client-server aplikácie.

2.1.2 Client-server[1]

- + Portovateľnosť aplikácií v rámci rôznych operačných systémov.
- + Jednoduchá údržba softvéru.
- Bezpečnostné riziko.
- Pri značnom počte užívateľov je nutné navyšovanie výkonu serverov.

2.2 Open source

Open source softvér [2] je všeobecne bezplatný softvér, ktorý je možné používať v biznis sfére. Príklady open source produktov sú: internetový prehliadač Mozilla Firefox, Wikipedia, Open Office.

- + Každoročne ušetrí firmám a jednotlivcom viac ako 60 miliónov dolárov.
- + Open source softvér sa stále vyvíja a počet užívateľov rastie, čo znamená, že môže byť v lepšej kvalite s menšou náchylnosťou na chyby v programe ako platené verzie podobého typu.
- + Užívateľ sa neviaže na rovnakého predajcu pri využívaní open-source.
- + Možnosť softvér meniť a prispôbovať vlastným požiadavkám.
- Zložitejšie GUI ako pri platených aplikáciách, ktoré dbajú na prezentovateľnosti aplikácie.
- Vysoké riziko odhalenia zraniteľnosti programu.
- Podpora pre program je vo forme komunity užívateľov, ktorí nie sú časovo viazaní riešiť chyby v aplikácií.

2.3 Využitie analyzátorov

Veľká časť populácie využíva služby bankových subjektov formou osobného účtu. Postupom času sa na každom účte nahromadí veľké množstvo finančných pohybov a pre klienta banky sa stáva obtiažne orientovať sa vo všetkých finančných transakciách na bankovom účte. Softvérové aplikácie, ktoré ponúkajú banky, neposkytujú dostatočnú interpretáciu informácií získaných z finančných transakcií. Napríklad internet banking Fio banky poskytuje len základnú vizualizáciu zostatkov na účte. Práve pri týchto prípadoch sa analyzátoři stretávajú s veľkým úspechom. Pomáhajú užívateľom zorientovať sa v bankových výpisoch prostredníctvom grafov a iných vizuálnych nástrojov. Ponúkajú širokú škálu možností, na základe ktorých si užívateľ presne zvolí potrebné filtre a dokáže spracovávať výpisy z rôznych účtov rôznych bánk.

2.4 Aktuálne formy riešenia

V súčasnej dobe existuje len málo analyzátorov, ktoré dovoľia užívateľovi vizualizovať transakčné dáta do detailov bez nutnosti synchronizácie aplikácie a bankového účtu. Mnoho finančných analyzátorov je riešených formou manažérov a Spend Trackerov, ktorým často chýba možnosť vizualizovať dáta za určité obdobie formou grafického vizualizátora. V jednotlivých podkapitolách sa stručnejšie rozoberú najznámejšie transakčné analyzátory a manažére: Spending Tracker TM, KMyMoney, HomeBank, Money for Mac, Wallet - Budget Tracker, Peňaženka - Prima banka.

2.4.1 Peňaženka - Prima banka[3]

Síce sa jedná o aplikáciu, ktorá je prioritne určená na správu účtu (posielanie a prijímanie platieb, kontrola hypoték a úverov, aktivácia bankových produktov...), je možné s touto aplikáciou vizualizovať pohyby na účte.

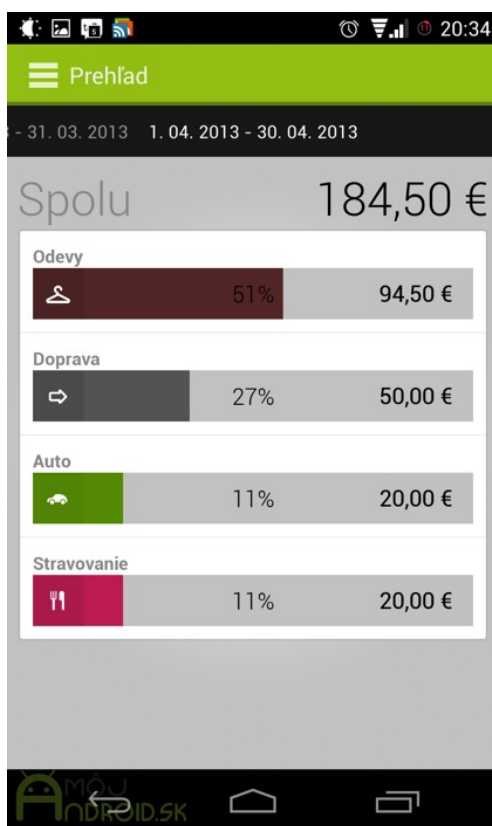
- + Ponúka veľa filtrov a možností, napríklad dokáže odfiltrovať výdaje spojené s obedom v reštauráciách, zobrazíť výdaje za určité obdobie a zobrazíť užívateľovi analyzované hodnoty formou tabuľky alebo grafu.
- + Jedná sa o bezplatnú aplikáciu.
- Plnú funkcionality využijú len klienti Prima banky.
- Čisto mobilné prevedenie.

2.4.2 Wallet - Budget Tracker[4]

Android aplikácia, ktorá je zameraná na sledovanie a správu financií na bankových účtoch.

- + Podporuje webovú aplikáciu (nejedná sa len o čisto mobilné prevedenie).
- + Podporuje export dát.
- + Jednoduché a prehľadné GUI.
- + Analýza dát a plánovanie výdavkov.
- Nutná synchronizácia s bankovým účtom.
- Spoplatnenie niektorých produktov v aplikácií.

2. ANALÝZA



Obr. 2.1: Ukážka obrazovky z aplikácie Prima banka - Peňaženka

2.4.3 Money for Mac[5]

Aplikácia využíva viac ako 1 500 000 užívateľov.

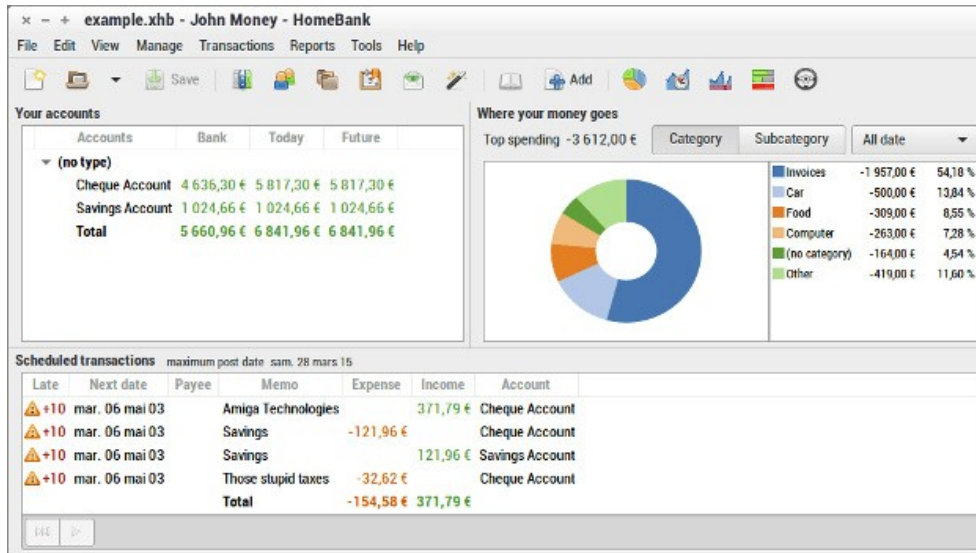
- + Veľká škála služieb od manažovania bankových účtov až po analýzu transakcií.
- + Preložená do 25 jazykov.
- Spoplatnenie aplikácie a určenie len na Mac OS.
- Nutnosť synchronizácie bankových účtov s aplikáciou.

2.4.4 HomeBank[6]

Aplikácia je kompatibilná s Windows, Mac OS a Linux.

- + Jednoduché užívateľské rozhranie.
- + Možnosti finančného manažéra (automatizácia transakcií).

- + Podpora importu a exportu súborov rôznych formátov.
- + Prevedenie formou open-source (aplikácia je dostupná bezplatne).
- Zložitejšie nastavenie bankových účtov.



Obr. 2.2: Ukážka obrazovky z aplikácie HomeBank

2.5 Výber technológií

Aplikácia má byť funkčná na Mac OS, OS Windows, OS Linux a má byť vo forme open-source s možnosťou budúcej nadstavby. Preto ako hlavný návrh na programovací jazyk prišiel do úvahy programovací jazyk Java. Java ako jeden z najpoužívanejších programovacích jazykov s neustálym vývojom ponúka všetky potrebné funkcionality pre túto prácu.

2.5.1 Frontendové technológie

2.5.2 Framework AngularJS[7]

Open source javascript framework.

- + Jednoduché testovanie.
- + Vhodný na klientsky orientované aplikácie.
- + Možnosť pridať interaktívne grafy pomocou knižnice AngularJS Directive.

- + Rýchly vývoj softvéru (efektívne vyjadrujúci sa jazyk).
- Nedostačujúca dokumentácia.
- Zložitosť niektorých jeho súčastí.
- Ťažký na profesionálne zvládnutie.

2.5.2.1 Framework Vaadin[8]

Ďalším zaujímavým frameworkom je Vaadin, ktorý je zameraný na tvorbu profesionálnych užívateľských rozhraní.

- + Jedná sa o open source framework.
- + Je to nový, rozvíjajúci sa framework, ktorý používa viac ako 150 000 programátorov celosvetovo s podrobnou dokumentáciou a mnohými internetovými návodmi.
- + Nie je potrebná znalosť javascriptu alebo značkovacích jazykov. Framework využíva programovací jazyk javu.
- Zložitejšie elementy ako napríklad grafy sú spoplatnené (nie sú súčasťou free verzie).
- Spoplatnenie Designera (jedná sa o drag-and-drop aplikáciu obsahujúcu elementy z frameworku).

2.5.2.2 Framework JavaFX[9][10]

Po závažných nedostatkoch v bezplatnej verzii predchádzajúceho frameworku sa pozornosť zamerala na framework Java FX, ktorý v posledných rokoch stúpil na popularite u mnohých programátorov využívajúcich UI framework.

- + Podporuje desktopové aplikácie na OS Windows, OS Linux a Mac OS.
- + Celý framework spolu s Scene Builderom je bezplatný (jedná sa o drag-and-drop aplikáciu s jednotlivými elementami a ich kompletnými vizuálnymi nastaveniami).
- + Oddeluje programátorskú (Controller triedy) a vizuálnu vrstvu (FXML súbory).
- + Veľká škála UI elementov, do ktorých patria aj grafy (zahrnuté od verzie 1.2).
- + Možnosť doladovať vizuálnu stránku elementov pomocou CSS.
- Stručnejšia dokumentácia.

2.5.3 Backendové technológie

2.5.3.1 SQLite-JDBC (Java)[11][12]

Jedná sa o knižnicu určenú na vytváranie a sprístupňovanie databázových súborov.

- + Nevyžaduje samostatný server, no napriek tomu dokáže podať veľmi dobré časové výsledky.
- + Jednoduchá inštalácia a použitie.
- + Je voľno dostupná.
- + Podporuje väčšinu SQL príkazov a má dobrú dokumentáciu (existuje množstvo internetových návodov a trikov z oblasti tejto knižnice).
- Nie je vhodná na veľkoplošné aplikácie, ktoré vyžadujú client - server model.
- Nepodporuje šifrovanie, ukladá dáta do súboru, ktorý je prístupný pre každého s právom čítania daného súboru. Vzniká tu bezpečnostné riziko, ak na počítač s uloženým databázovým súborom má prístup iná osoba okrem užívateľa.

2.5.3.2 PostgreSQL[13]

Jedna z najpokročilejších open source databáz. Vývoj PostgreSQL prebieha už desaťročia.

- + Dokumentácia pokrývajúca takmer všetky problémové oblasti.
- + Vysoko stabilná databáza s minimálnym počtom databázových spadnutí.
- + Navrhnutá pre prostredia s vysokou záťažou.
- Nepodporuje celý ANSI SQL 92' štandard.
- Zložitá inštalácia a použitie.

Pre túto prácu sa vo výslednej forme zvolila práve knižnica SQLite-JDBC Java. Keďže sa jedná o desktopovú aplikáciu, ktorá nevyužíva client-server štruktúru, jej databázová zložitosť (počet vzťahov a tabuliek) je nízka. Knižnica SQLite je postačujúcou a najvhodnejšou alternatívou pre tento projekt.

2.6 Grafy - vizualizácia dát

Kľúčovou vlastnosťou AFT je vizualizácia dát. Užívateľ AFT by mal byť schopný vizualizovať dáta podľa svojej potreby v rámci možnosti zvoleného filtra. Najpoužívanejšie základné grafové typy, ktoré sa použili do AFT sú: koláčový graf, stĺpcový graf, čiarový graf, bodový graf, plošný graf. Ďalšou zaujímavou formou vizualizácie dát s polohou je mapa. Pri tvorbe grafových výstupov treba myslieť na niekoľko základných pravidiel:

- Jednoduchá vizualizácia - užívateľ by sa mal vedieť rýchlo zorientovať v grafe.
- Používať vhodné grafy na dané datasety - väčšina užívateľov nepreferuje 3D grafy vzhľadom na zložitú orientáciu a neprehľadnosť dát.
- Ponechať možnosť užívateľovi zvoliť si svoj preferovaný grafový model.

2.6.1 Koláčový graf[14]

- + Vizualne jednoduchší ako ostatné typy grafov.
- + Potrebuje minimum dodatkových informácií.
- + Zobrazuje relatívne pomery viacerých tried z dát.
- Z prvotného zobrazenia nie je jednoduché určiť presné hodnoty.
- Je nevhodný na vizualizáciu časových zmien.

2.6.2 Stĺpcový graf[15]

- + Takmer presný odhad hodnôt z grafickej vizualizácie.
- + Sumarizuje veľké datasety do vizuálnej formy.
- + Jednoduchý na porozumenie.
- Vyžaduje dodatkové informácie.
- Nevhodný na veľké množstvo časových úsekov.

2.6.3 Čiarový graf[16]

- + Vhodný na vizualizáciu časových zmien.
- + Uprednostňuje sa pred bodovým grafom ak je dôležitejšia lokálna zmena pri bodoch.
- Nedokáže zobraziť veľké množstvo dátových tried (optimum sa pohybuje okolo hodnoty 4 tried na 1 graf).
- Nevhodný na vizualizáciu dát s veľkou škálou odlišnosti.

2.6.4 Bodový graf[17]

- + Vhodný na pozorovanie veľkého množstva dát.
- + Vizualizuje koreláciu a klasterové efekty medzi dátami.
- + Možnosť zakomponovať koláčový typ grafu pre obsiahlejšie pozorovania.
- Problém s diskretizovanými hodnotami.
- Zložitejší na porozumenie a prezentovanie oproti iným bežným grafovým variantám.

2.6.5 Plošný graf[18]

- + Zvýraznenie spojitých dát.
- Nevhodný na prekrývajúce sa typy dát.
- Prehľadne dokáže vizualizovať malé množstvo tried.

2.6.6 Mapa

- + Prezentovanie dát z iného pohľadu oproti bežným grafom.
- + Vhodný na pozorovanie korelácií a clusterových efektov medzi dátovými vzorkami.
- Nutný geocoding pre dáta bez presnej zemepisnej šírky a dĺžky.
- Vizualizuje len dáta s polohou.
- Pri využívaní online máp (primárne geocodingu) nutnosť internetového prístupu alebo vysoká pamäťová náročnosť.

2.7 Dátové formáty

Dátové formáty sú dôležitou súčasťou projektu. Pre projekt AFT sa zvolili formáty CSV a XLS. Pôvodne bol zvolený len formát CSV ale z analýzy formátov u niektorých bankových subjektov vyšla najavo skutočnosť chýbajúceho, chybné formátovaného CSV súboru.

2.7.1 CSV formát[19]

Súborový formát používaný na uskladňovanie dát. Súbor formátu CSV je podobný textovému súboru. V súčasnej dobe štandardom CSV formátu je RFC 4180, avšak masívne používanie CSV formátu vyústilo k odklonu od tohto štandardu. Za CSV súbor sa preto považuje akýkoľvek súbor dodržiajúci niekoľko základných pravidiel:

- Obsahuje obyčajný text kódovaný v ASCII, rôznych typoch Unicode, EBCDIC alebo Shift JIS.
- Pozostáva zo záznamov (väčšinou 1 záznam na 1 riadok).
- Záznamy sú rozdelené do polí pomocou oddeľovačov (čiarka, bodkočiarka, pomlčka...).
- Každý záznam má rovnakú sekvenciu polí.

Výhody a nevýhody CSV formátu[20]:

- + Je čitateľný a ručne editovateľný.
- + Využívaný veľkou časťou bankových subjektov.
- + Rýchlejší na spracovanie a menej náročný na pamäť.
- Absentuje univerzálny štandard, ktorý by všetky subjekty dodržiavali.
- Podporuje iba základné dátové typy.
- Slabá podpora špeciálnych znakov.

2.7.2 XLS(X) formát[21]

Jedná sa o súborový formát Microsoft Excel. XLS bol hlavným formátom pre Excel do roku 2003. Síce jeho nástupca XLSX ponúka širokú škálu lepších možností (menšia pamäťová náročnosť, viac riadkov a stĺpcov...), XLS je stále veľmi populárnym spreadsheet formátom. XLS ukladá informácie priamo do binárneho formátu, zatiaľ čo u nástupcu XLSX je informácia ukladaná do textového súboru, ktorý využíva XML na definovanie jej parametrov.

- + Má univerzálny štandard, vďaka ktorému je jednoduchšie parsovanie dát z jednotlivých výpisov.
- + Pri otvorení v spreadsheet editore je prehľadnejší ako formát CSV.
- Nutnosť mať nainštalovaný vhodný editor/viewer (súbory tohto formátu sa nedajú otvoriť bežnými textovými editormi).
- Nie je až tak rozšírený u bankových subjektov ako štandardný CSV formát.

2.8 Podporované bankové subjekty

Výsledný stav aplikácie bude podporovať možnosť nahrať súbory z dvoch bankových subjektov.

- Fio banka SK
 - Podporovaný vstupný formát z Fio banky SK je CSV.
- Prima banka
 - Podporovaný vstupný formát z Prima banky je XLS.

2.9 Štatistiky z transakcií

Táto sekcia sa podrobnejšie venuje jednotlivým štatistikám, ktoré je možné odsledovať a následne odfiltrovať z transakčných výpisov. V sekcii je spomenutých len pár hlavných a najzaujímavejších prípadov.

2.9.1 Bankomaty

Aplikácia ponúkne užívateľovi prehľad o tom, koľko v jednotlivých dňoch, mesiacoch, rokoch vybral z bankomatov a pri bankách, ktoré ponúkajú informáciu aj o mieste daného bankomatu, bude užívateľ schopný vizualizovať miesta bankomatov na mape.

2.9.2 Celkový príjem vs celkový výdaj

Získanie rýchleho a presného obrazu finančných príjmov a výdajov za určité obdobie.

2.9.3 Platby kartou v určitom subjekte

Užívateľ si bude môcť zistiť presnú sumu koľko minul za zvolené obdobie v určitom obchode. Ak použije zobrazenie formou mapy, tak zistí aj presný počet transakcií, ktoré za dané obdobie v subjekte urobil s presným popisom jednotlivých transakcií.

2.9.4 Sledovanie platieb na selektovaných účtoch

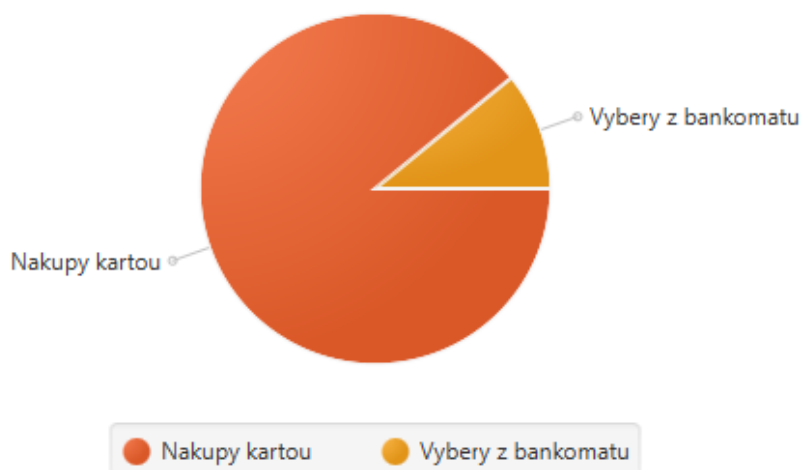
Sledovanie pohybov finančných transakcií na účtoch za zvolené časové obdobie.

2.9.5 Filtrovanie podľa typu transakcií

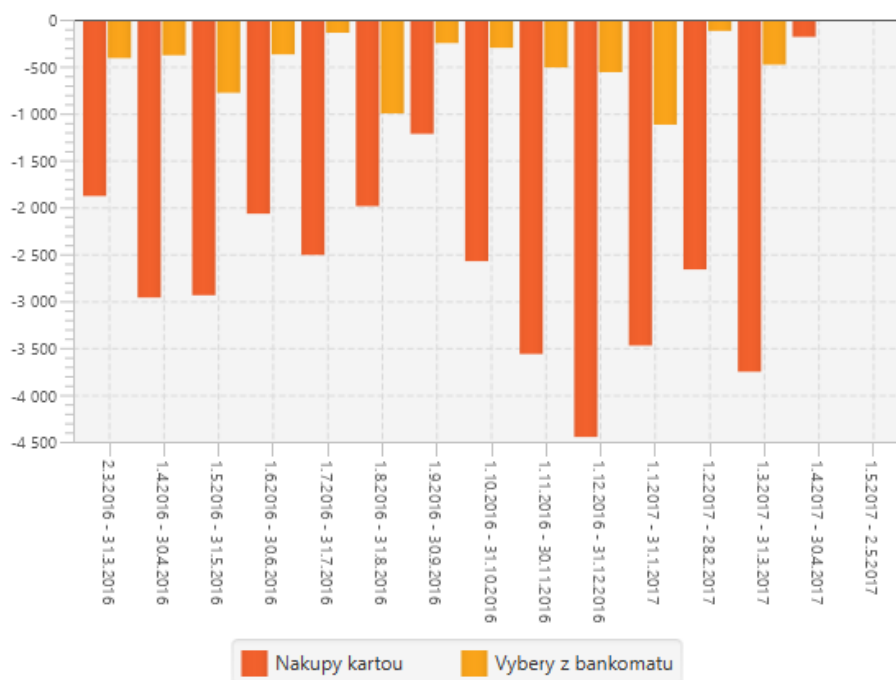
Užívateľovi je sprístupnené filtrovanie podľa typu transakcií aké na jeho účte prebiehali. Podľa tejto ponuky si môže zvoliť jednu ale aj viac typov transakcií a odsledovať ich priebeh vo zvolenej časovej jednotke.

2.9.6 Variabilné, konštantné a špecifické symboly

Jednoduchý výber a zobrazenie transakcií s vybranými symbolmi.



Obr. 2.3: Ukážka koláčového grafu z AFT



Obr. 2.4: Ukážka stĺpcového grafu z AFT

Návrh

3.1 Analýza požiadaviek

3.1.1 Funkčné požiadavky

3.1.1.1 F1 - Nahratie výpisu pohybov z účtu

Systém umožní užívateľovi nahrať výpisy pohybov z účtu. Užívateľ si vyberie banku a následne formulár. Aplikácia umožní nahrať viac výpisov z rovnakej banky súčasne.

3.1.1.2 F2 - Oznámenie o možných duplicitách pri nahrávaní výpisu

Systém upozorní užívateľa pri možnej zhode (rovnakých udalostiach z rôznych výpisov) a dovoľí mu zvoliť možnosť pre prácu s touto formou dát. Aplikácia pracuje s dátami ako s duplicitami a použije ich, až keď sa stanú opäť unikátne (pri zmazaní výpisu s duplicitnými dátami) alebo užívateľ zvolí, že sa jedná o unikátne udalosti a aplikácia s nimi pracuje ako s bežnými dátami.

3.1.1.3 F3 - Vymazanie výpisu pohybov z účtu

Aplikácia umožní užívateľovi vymazať zvolené výpisy pohybov z databázy programu.

3.1.1.4 F4 - Zobrazenie grafu

Aplikácia umožní vybrať časové obdobie, z ktorého bude čerpať dáta z databázy. Užívateľ si následne zvolí časový filter na osi X (deň, mesiac, rok, celý čas - pre koláčový a mapový graf). Podľa preferencie a zvoleného filtra si na záver užívateľ zvolí výsledný formát grafu z dostupných grafových možností pre daný filter.

3. NÁVRH

3.1.1.5 F5 - Stiahnutie grafu

Každý graf bude možné stiahnuť ako obrázok vo formáte png. Táto možnosť bude prístupná pri každej obrazovke s grafom.

3.1.2 Nefunkčné požiadavky

3.1.2.1 N1 - Dostupnosť na operačných systémoch Windows, Linux, Mac OS

Aplikácia bude dostupná a plne funkčná na operačných systémoch Windows, Linux a Mac OS. Jedinou požiadavkou je mať nainštalovaný JRE balíček vo verzií 8 a viac.

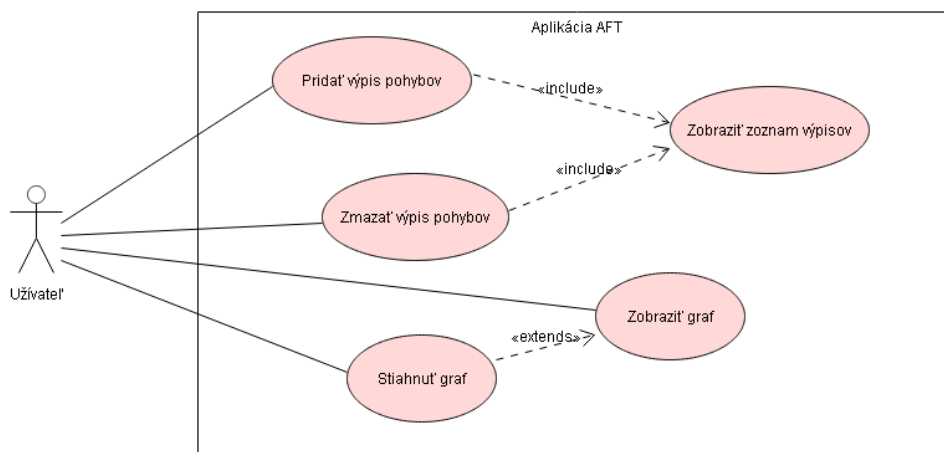
3.1.3 N2 - Funkčnosť v offline režime

Aplikácia bude plne funkčná v offline režime (okrem vykreslovania mapy).

3.1.4 N3 - Vlastná databáza funkčná v offline režime

Aplikácia bude disponovať databázou, ktorú si v prípade potreby vytvorí a ktorá bude ukladať a sprostredkovať dáta aplikácií aj v offline režime.

3.1.5 Use case diagram



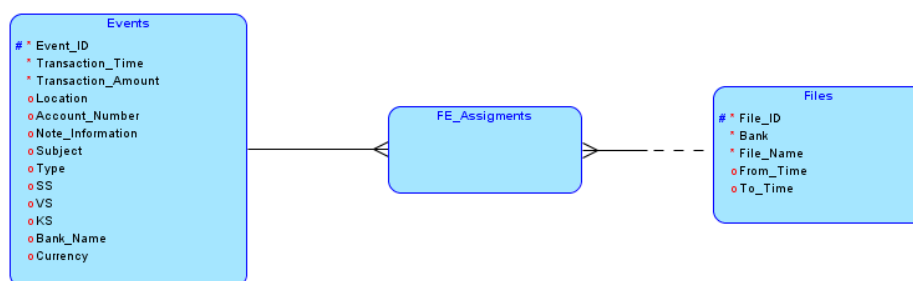
Obr. 3.1: Model prípadov využitia pre analyzátor finančných transakcií

Vyššie uvedený obrázok znázorňuje niektoré prípady využitia analyzátoru finančných transakcií. Pre dôkladnejšie pochopenie práca demonštruje UC - Stiahnuť graf. UC začína keď sa užívateľ rozhodne stiahnuť si grafickú vizualizáciu dát. Aplikácia zobrazí okno s možnosťami výberu filtru, grafu, časovej

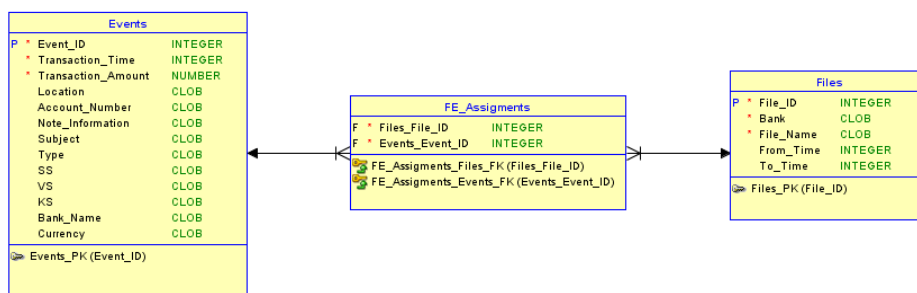
periódy pre os X a časového úseku, z ktorého má čerpať dáta. Užívateľ zvolí všetky potrebné informácie, na záver vyberie formu grafického zobrazenia, pre ktorú sa rozhodol. Aplikácia otvorí nové okno s grafickým výstupom a tlačítkom pre uloženie daného grafu. Užívateľ zvolí názov súboru a cestu, kde má byť súbor uložený. Následne sa uloží graf ako png súbor.

3.2 Návrh databázy

3.2.1 Modely databázy



Obr. 3.2: Logický model databázy



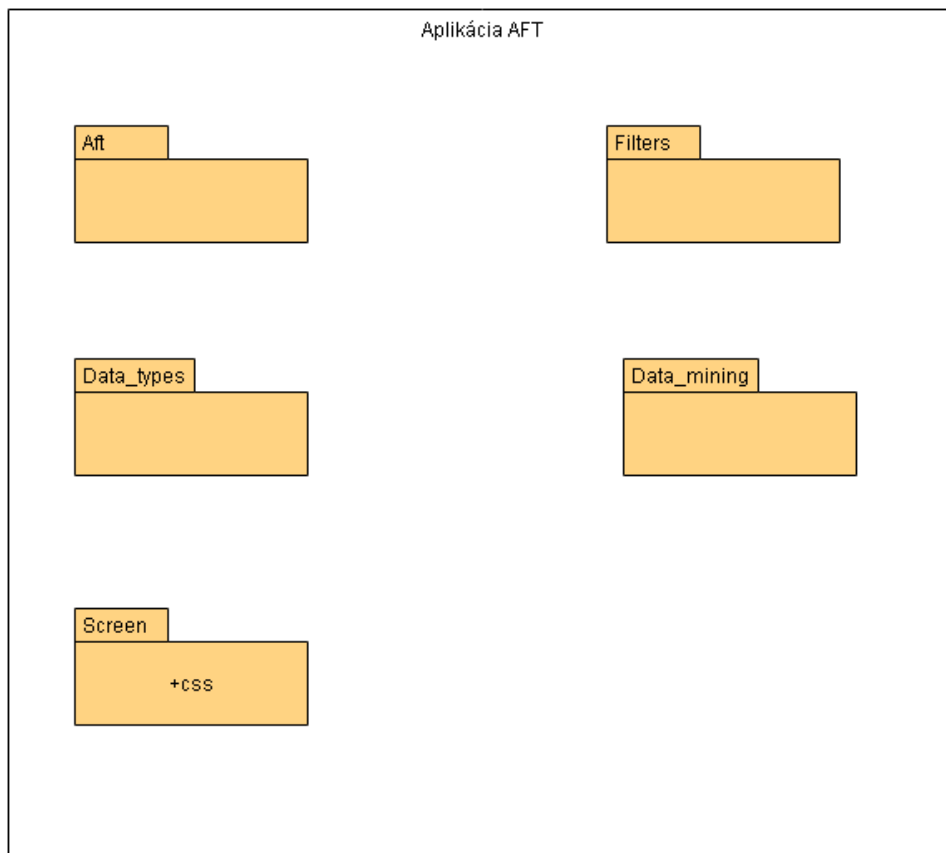
Obr. 3.3: Relačný model databázy

Z vyššie uvedených modelov je zrejmé, že sa nejedná o zložitú databázu. Tabuľka Files slúži na evidovanie nahratých súborov. Pri každom nahrávaní súboru sa eviduje názov banky, názov súboru a ak nie je súbor prázdny tak aj dátum prvej a poslednej transakcie. IDčko súboru je pridelené automaticky pomocou autoincrementu. Tabuľka Events obsahuje všetky udalosti zo súboru. Pri udalostiach evidujeme čas a výšku transakcie. Nepovinnými údajmi z výpisu, ktoré sa zaznamenávajú sú: mena, názov banky odkiaľ prichádzajú alebo

odchádzajú finančné prostriedky, konštantný symbol, variabilný symbol, špecifický symbol, typ transakcie, predmet (konateľ/spoločnosť), doplnková informácia, číslo protiúčtu, lokalita. IDčko pre udalosti je pridelované automatickou formou autoincrementu. Tabuľka FE_Assignments slúži na dekompozíciu vzťahu M:N, ktorý nastal pri situáciách kedy dochádzalo ku kolíziám (duplikátom) udalostí v rámci dvoch rôznych súborov. Väčšina údajov pri udalostiach je nepovinná z dôvodu rozmanitosti elektronických formátov u jednotlivých bankových subjektoch (banky nemajú vždy rovnaký CSV/xls formát) a tak tiež možnosť pre užívateľa vložiť neúplný bankový výpis (neobsahujúci všetky položky).

3.3 Diagram balíčkov

Kvôli lepšej prehľadnosti sa zdrojové súbory rozdelili do niekoľkých balíčkov. Do každého balíčka sa združujú triedy a súbory, ktoré spolu najviac súvisia. V balíčku Aft sa nachádza hlavná trieda AFT s main funkciou. Ďalej sa tu nachádzajú final classy (Database, Geocoder, ThreadPoolExe, Scenes) so statickými metódami, ku ktorým sa dá pristupovať v iných častiach programu. V balíčku Filters sa nachádza abstraktná trieda AbstractFilter spolu s jednotlivými podtriedami (filtrami). Balíček Data_types obsahuje vytvorené dátové typy, ktoré umožňujú jednoduchšiu manipuláciu s informáciami v programe. Balíček Data_mining zahŕňa interface a triedy zabezpečujúce načítanie a uloženie dát zo vstupných súborov do adekvátnych štruktúr. Balíček Screen pozostáva z Controllerov (JavaFX tried riadiacich programovú a vizuálnu stránku jednotlivých scén) a fxml súborov (súbory obsahujúce nastavenia jednotlivých UI component na scéne v jazyku fxml).



Obr. 3.4: Diagram balíčkov

Implementácia

4.1 Dátové typy

Z dôvodu lepšej synchronizácie a výmene informácií v programe sa implementovali nasledujúce dátové typy:

4.1.1 Enum dátové typy

Bankový enum obsahuje názvy bánk, ktoré aplikácia podporuje. Časový enum je využitý pri zadávaní časového filtra na osi X. Užívateľ si môže vybrať, či potrebuje filtrovať dáta (rozdeliť dáta do kategórií) na dni, mesiace, roky alebo len jednu kategóriu a to celý čas.

4.1.2 Tabuľkové dátové typy

Tieto dátové typy konkrétne `FileTable` a `EventTable` slúžia na reprezentáciu dát z databázy. `FileTable` obsahuje všetky premenné z databázovej tabuľky `Files` a `EventTable` z databázovej tabuľky `Events`. Okrem základného využitia (čerpania dát z databázy) sa využívajú aj v GUIcku či parsovaní dát z podporovaných formátov, napríklad `FileTable` sa využíva pri tabuľkách s výberom súborov. Prvotným riešením bolo využívať `ResultSet`, ktorý obsahuje výslednú množinu dát po spracovaní databázového dotazu, avšak nastal problém so zatváraním databázy. `ResultSet` totiž vyžaduje mať otvorený prístup do databázy, kým sa s ním pracuje. Toto riešenie sa muselo vylúčiť a nájsť spomínané riešenie s dátovými typmi `EventTable` a `FileTable`.

4.1.3 `Option_CSV`

Tento dátový typ sa využíva pri načítavaní dát z CSV súborov. Podľa hlavičky CSV súboru sa zistia jednotlivé položky a ich index. Následne sa vyplní dátový typ `Option_CSV`, ktorý slúži ako predloha pre následné parsovanie súboru a preto sa aplikácia nemusí obmedzovať na jeden formát CSV súboru,

ale poradí si aj s formátmi, ktoré nemajú úplný charakter (neobsahujú všetky položky). Tento dátový typ sa môže využiť aj pri iných formátoch ako je CSV. Dôležité je , aby daný súbor obsahoval hlavičku, podľa ktorej sú rozmiestnené informácie v súbore.

4.1.4 MapData

MapData slúži ako pomocný formát pre Mapu. Obsahuje informácie ako zemepisná dĺžka, zemepisná šírka, dodatkové informácie pre informačné okno v mape, celkový objem finančnej transakcie pre danú identifikovanú lokalitu, peňažnú menu a názvy všetkých subjektov z predmetnej lokality. Pôvodné riešenie umožnilo aplikácií vytvoriť viac markerov (označení na google mape) s rovnakou zemepisnou šírkou a dĺžkou. Toto riešenie sa stalo nevyhovujúcim. Optimálnym riešením je aktuálna dátová štruktúra MapData.

4.1.5 EList_Key_Bank_entity

Jedná sa o pomocnú štruktúru obsahujúcu EventTable List, vygenerovaný kľúč databázou pre daný súbor a názov banky. Táto štruktúra sa využíva pri načítavaní dát zo súboru a následným posielaním informácií ďalej v programe.

4.2 Backendové časti aplikácie

V nasledujúcich podkapitolách sa popíšu jednotlivé backendové časti programu. Tieto časti sa starajú o komunikáciu s databázou, parsovanie súborov, treadovanie, nastavenie programových premenných a tvorenie filtrov.

4.2.1 Komunikácia s databázou

Komunikácia s databázou prebieha formou verejnej konečnej triedy s verejnými statickými metódami. Na začiatku programu sa volá funkcia createDatabase(), ktorá vytvorí databázový súbor spolu s príslušnými tabuľkami, ak neexistujú. Keďže každá statická metóda si sama otvorí spojenie s databázou a aj si ho zavrie, bolo nutné realizovať nahrávanie veľkého množstva záznamov v rámci jedného volania metódy, inak by došlo k významnému spomaleniu aplikácie. Tento problém sa podarilo vyriešiť poslaním listu obsahujúceho záznamy do funkcie realizujúcej nahrávanie záznamov. Aby sa časová efektívnosť zvýšila na maximum bolo potrebné nastaviť autoCommit na hodnotu false a realizovať len jeden commit po všetkých akciách v databáze. Aby bol zdrojový kód prehľadnejší vytvorili sa tri statické funkcie na pridávanie dát do databázy, updateFiles() má na starosť pridávanie súborov. Na pridávanie udalostí slúžia dve funkcie updateFE_Assignments, ktorá slúži na pridávanie duplikátnych udalostí (nevytvára nové udalosti len pridá odkazy na už existujúce udalosti) a

`updateEventsAndFE_Assignments()`, ktorá pridá novovytvorené udalosti (definované ako unikátne) do tabuľky `Events` a následne pridá odkazy na tieto udalosti do tabuľky `FE_Assignments`. Zmazanie vybraného súboru je implementované formou metódy `deleteFile()`. Jej vstupným parametrom je IDčko súboru, ktorý sa ide zmazať. Celý proces prebieha v troch krokoch, najskôr sa vymažú všetky udalosti z tabuľky `Events`, ktoré sú pre daný súbor unikátne, následne sa zmažú všetky odkazy v tabuľke `FE_Assignments` týkajúce sa daného súboru a nakoniec aj samostatný súbor z tabuľky `Files`. Na vyčistenie celej databázy (`Files`, `FE_Assignments`, `Events`) je použitá funkcia `clearDatabase()`. Získavanie dát z databázy je realizované dvoma metódami: `selectFromFiles()` a `selectFromEvents()`. Obe funkcie využívajú už spomínané dátové typy `FileTable` a `EventTable`, ktoré vracajú formou listu, podľa obmedzení v sql dotaze. Metódy `getAmount()` a `getDistinctFromEvents()` boli implementované z dôvodu urýchlenia programu, keďže výpočty a následné poslanie už odfiltrovaných dát je omnoho rýchlejšie ako poslanie celého obsahu databázy s následným získaním informácií v programe.

4.2.2 Nahrávanie a parsovanie súborov

Na implementáciu nahrávania a parsovania súborov sa použil továrenský vzor. „Jedná sa o jeden z najpoužívanejších vzorov v jave, kedy pri vytváraní objektu sa neodhaľuje logika vytvárania objektu ale odkazuje sa na tento objekt pomocou známeho rozhrania.“[22] Metóda `addToDatabase()` je implementovaná u každej banky ináč, podľa typu a formátu súboru, ktorý je podporovaný bankou. Napríklad u `PrimaBanky` je typ podporovaného formátu `xls`, `FioBankaSk` uprednostňuje `csv`. Pre budúce pridávanie nových bánk je nutné vytvoriť novú triedu s názvom novej banky, implementovať rozhranie `Bank` a pridať podmienku v metóde `getBank()`, ktorá sa nachádza v triede `BankFactory`.

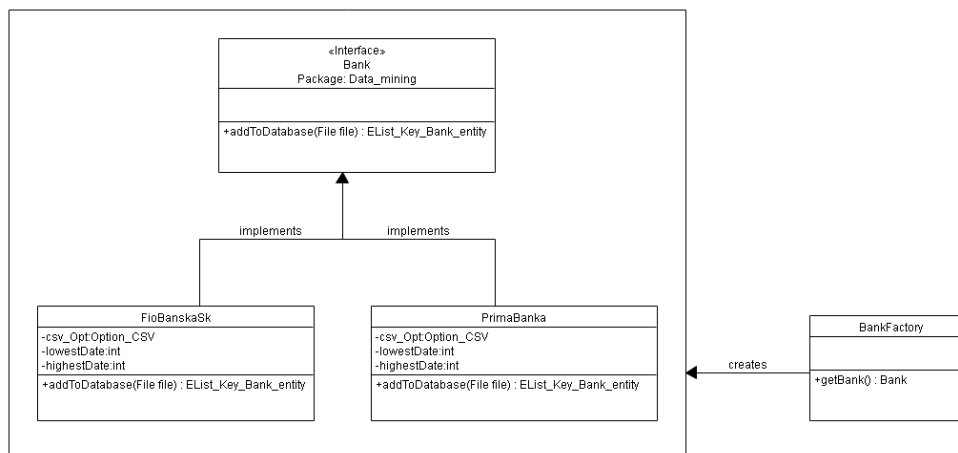
4.2.2.1 Apache POI

Jedná sa o Java API na Microsoft dokumenty. Táto knižnica bola použitá na čítanie súborov vo formáte `xls`, avšak zvláda čítanie aj novších verzií formátov napríklad `xlsx`. Použitá verzia tejto knižnice je 3.15. „Je už zabezpečená pred možnými DOS útokmi, ktoré mohli nastať pri otváraní špeciálne vytvoreného OOXML súboru, známeho tiež ako XML Entity Expansion (XEE) attack.“[23]

4.2.3 Threadovanie a programové premenné

Threadovanie je implementované formou `ThreadPoolExecutora`, do ktorého sú vkladané výpočtové úlohy. Následne si `ThreadPoolExecutor` vyberá úlohy a rieši ich nezatažujúc hlavný `Thread`, ktorý zabezpečuje grafický dizajn aplikácie. Programové premenné, konkrétne `Geocoder`, slúži na získanie polohy z

4. IMPLEMENTÁCIA



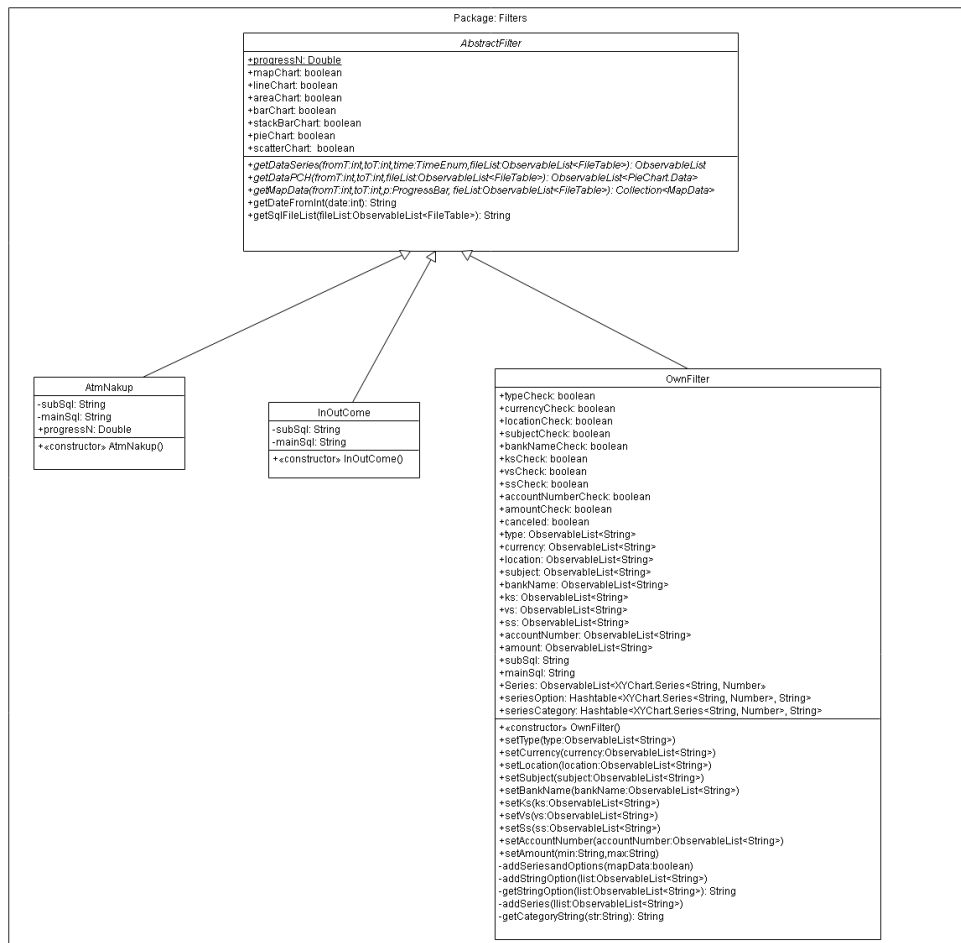
Obr. 4.1: Ukážka použitia továrenského vzoru pri nahrávaní a parsovaní súborov.

časti názvu v textovej forme. Aplikácia využíva Geocoder z knižnice google-maps-services-java[24], ktorá okrem Geocodera obsahuje aj iné Google Maps APIs. Nevýhoda využívania Google Maps Geocoding API je v množstve možných požiadaviek na server za 1 deň, tento limit je obmedzený na 2500 požiadaviek na 1 kľúč. Výhodou je, že si každý užívateľ môže tento kľúč vygenerovať sám. Ďalšou premennou, ktorú je potrebné nastaviť na začiatku programu, je Tooltip od javaFX. Dĺžka času ako dlho má byť Tooltip viditeľný a za ako dlho po nabehtnutí myšou na daný objekt sa zobrazí, sa v rámci instance Tooltipu dá nastaviť až vo verzii Java 9.0. Kvôli tomuto obmedzeniu bolo použité riešenie z GitHubu[25] od užívateľa darmbrust na nastavenie globálnych časových parametrov pre Tooltip v Java 8.

4.2.4 Filtre

Na implementáciu filtrov bola použitá abstraktná trieda `AbstractFilter`, ktorá obsahuje tri abstraktné metódy na získanie vyfiltrovaných dát v rôznych formátoch (pre rôzne typy grafov). Na získanie dát pre mapu sa používa metóda `getMapData()`, pre koláčový typ grafu sa používa metóda `getDataForPCH()` a pre všetky ostatné typy grafov sa používa metóda `getDataSeries()`. Bežné vstupné parametre pre tieto funkcie sú zoznam súborov, z ktorých majú byť informácie čerpané a časové rozmedzie vo forme `intu` (YYYYMMDD). Metóda `getDataSeries()` obsahuje navyše parameter `TimeEnum`, ktorý určuje v akých časových obdobiach chce dané hodnoty pozorovať. Formou troch tried, ktoré rozširujú abstraktnú triedu `AbstractFilter`, sú naprogramované tri filtre. Trieda `AtmNakup` produkuje filter na základe rozdelenia udalostí na: udalosti spojené s výberom hotovosti z bankomatu a udalosti spojené s kartovou trans-

akciou. Trieda `InOutCome` rozdeľuje dáta na základe informácií, či sa jedná o príjem alebo výdaj. Trieda `OwnFilter` implementuje užívateľom zvolený filter, môže si vybrať podmienky filtrovania a kategórie, podľa ktorých má program vyberať dáta z databázy. Dôležitou súčasťou tejto triedy je metóda `addSeriesandOptions()`, ktorá na základe vstupnej boolovskej hodnoty určí, či má vytvoriť sql dotaz pre mapu alebo klasický graf a následne dané filtrovacie podmienky v sql dotaze vytvorí. Tento dotaz sa neskôr použije pri získavaní dát už v spomýnaných metódach.



Obr. 4.2: Diagram tried pre balíček Filters.

4.3 Frontendové časti aplikácie

Medzi frontendové časti aplikácie patria jednotlivé obrazovky a controller triedy, ktoré sa starajú o správne zobrazovanie informácií, komunikáciu me-

dzi GUI prvkami a programom. V jednotlivých podkapitolách sa postupne rozoberú obrazovky s príslušnými GUI prvkami.

4.3.1 Scény

Na začiatku programu sa vytvoria všetky statické scény (obrazovky) a načítajú sa príslušné css súbory. Tento proces urýchli prechody medzi obrazovkami a program nemusí načítavať danú scénu opakovane. Táto trieda obsahuje verejné statické metódy, takže je možné k nim pristúpiť v ktorejkoľvek časti programu. Okrem sprostredkovania scén, táto trieda umožňuje aj prístup ku controllerom daných obrazoviek.

4.3.2 Hlavné menu

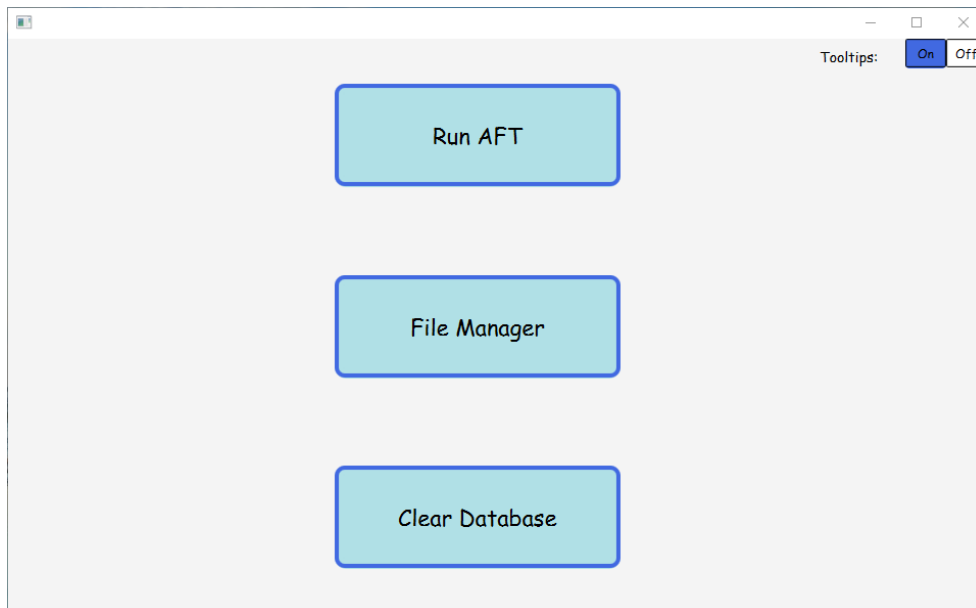
Hlavné menu pozostáva z:

- Tlačidla na spustenie analyzátoru, po kliknutí na toto tlačidlo sa otvorí nová obrazovka s možnosťami pre výber grafu.
- Tlačidla na spustenie file manažéra, po kliknutí na toto tlačidlo sa otvorí obrazovka s možnosťami nahrávania a mazania súborov v tejto aplikácii.
- Tlačidla na vyčistenie databázy, po kliknutí na toto tlačidlo sa otvorí vyskakovacie okno s upozornením na potvrdenie vyčistenia databázy. Po potvrdení užívateľom sa zmažú z databázy všetky údaje.
- Prepínacieho tlačidla slúžiaceho na zapnutie a vypnutie Tooltipov pre GUI elementy v aplikácii.

4.3.3 File manager

Okno file manažéra pozostáva z interaktívnej tabuľky, v ktorej sú zobrazené aktuálne nahraté súbory. Užívateľ si môže súbory zoradiť podľa názvu súborov, názvu bánk a časového rozmedzia súborov. Ďalšími dôležitými prvkami tejto obrazovky sú tlačidlá a textové polia:

- Späť na hlavné menu.
- Výber banky. Jedná sa o combo-box prvok, ktorý umožňuje vybrať užívateľovi banku, z ktorej pochádza vkladateľný súbor.
- Nahratie súborov. Po stlačení tohto tlačidla sa užívateľovi otvorí nové okno, kde si môže zvoliť súbory zo svojho zariadenia, ktoré chce nahráť do aplikácie.
- Zmazanie súborov. Po aktivácii tohto tlačidla sa zmažú všetky vybrané súbory z tabuľky.



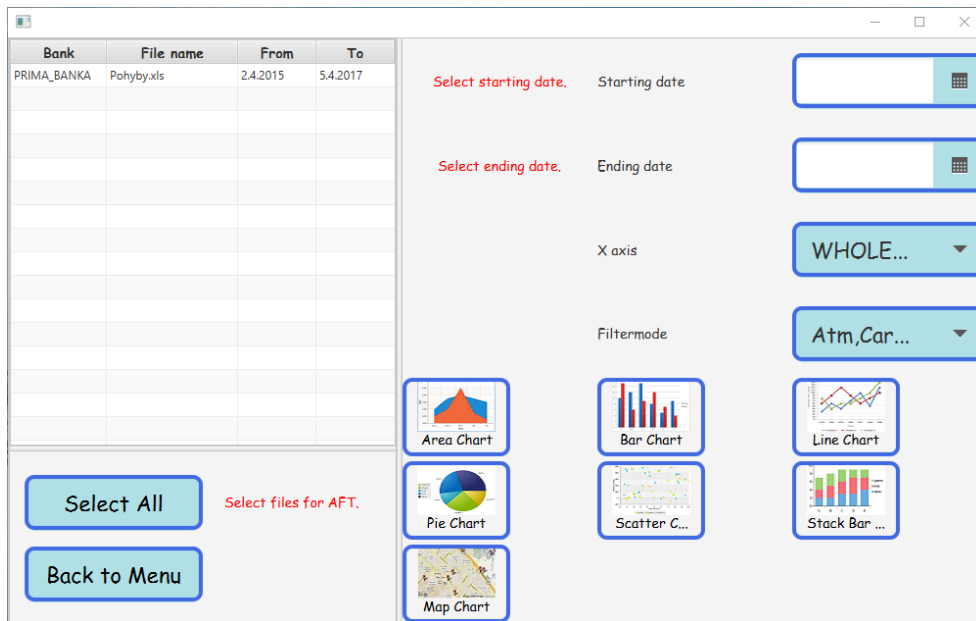
Obr. 4.3: Ukážka hlavného menu aplikácie AFT.

- Textové pole na zobrazenie upozornenia pri nevybratí banky pred zvolením nahratia súborov.
- Informačné textové polia na vizualizáciu počtu nahratých a zmazaných súborov.

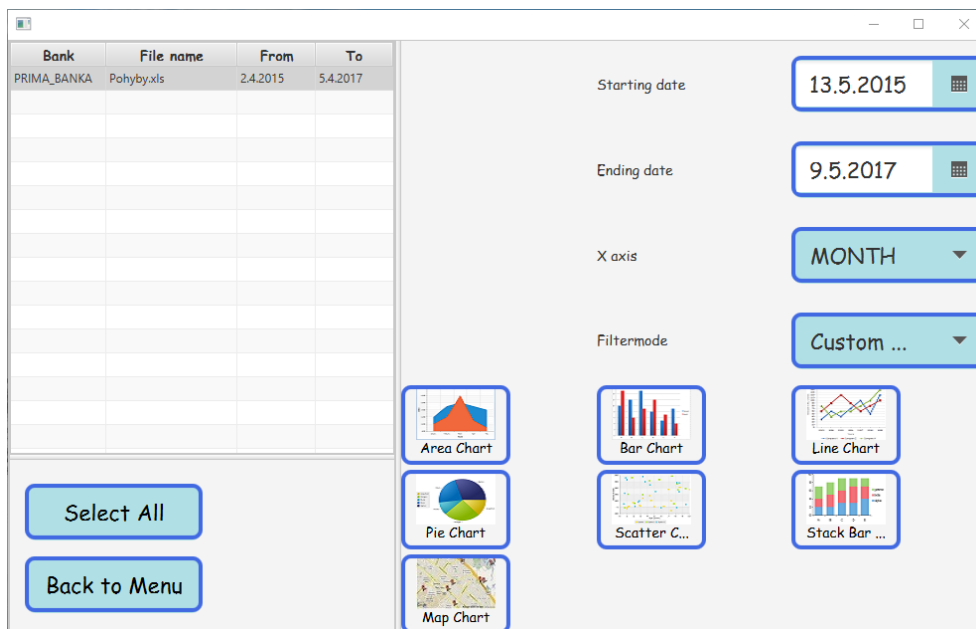
4.3.4 Grafové menu

V tomto okne si užívateľ volí najzákladnejšie možnosti pre analýzu dát ako napríklad : časové rozmedzie, použité súbory, časovú kategóriu na osi X, filter dát a grafový typ prevedenia. Na výber súborov, z ktorých sa dáta analyzujú, sa používa tabuľka. Užívateľ stlačením tlačidla "Zvoliť všetko", zvolí všetky súbory v tabuľke, ak nie je zvolený aspoň jeden súbor, aplikácia vypíše hlášku o chýbajúcom výbere súboru. Na zvolenie dátumu, od ktorého majú byť dáta filtrované a dátumu, do ktorého majú byť dáta filtrované sa používa DatePicker, pričom ak užívateľ zvolí koncový dátum pred začiatočným dátumom alebo niektorý z dátumov nevyplní, aplikácia zobrazí chybovú hlášku podľa problému, ktorý nastal pri konkrétnom DatePickeri. Na vybratie kategórie na osi X a filtra sa používa combo-box. Výber grafového typu sa realizuje formou tlačidiel s ikonou daného grafu a úzko súvisí s vybratým filtrom. Každý filter má prípustné iné grafové zobrazenia. Napríklad kým AtmNakup filter má možnosť zobraziť dáta do mapy, InOutcome filter túto možnosť nemá. Pri zobrazovaní dát do koláčového alebo mapového grafu sa automaticky nastaví hodnota na osi X do celého času.

4.3. Frontendové časti aplikácie



Obr. 4.5: Ukážka grafového menu s chybne vyplnenými vstupmi.



Obr. 4.6: Ukážka správne vyplneného grafového menu.

4. IMPLEMENTÁCIA

OwnFilter

Set as category Type: ATM, Tra... Set as category KS: [dropdown]

Currency: [dropdown] VS: [dropdown]

Location: [dropdown] SS: [dropdown]

Subject: [dropdown] Amount: [input] <= X <= 100

Bank name: [dropdown] Account num: [dropdown]

Import Filter Cancel

Obr. 4.7: Ukážka funkčne vyplneného vlastného filtra.



Obr. 4.8: Ukážka obrazovky so stĺpcovým grafom pre 4 konštanté symboly.

Testovanie

5.1 Testovanie na OS

Aplikácia bola vyvíjaná a zároveň aj testovaná na Windows 10. Okrem tohto operačného systému bola aplikácia spustená a otestovaná na operačných systémov Windows 8 a Linux Ubuntu 14.04. Aplikácia bežala a zobrazovala dáta správne.

5.2 Testovanie SQL dotazov

Počas vývoja aplikácie bolo nutné testovať korektnosť SQL dotazov. Syntaktická forma sa testovala online a praktické použitie dotazov sa testovalo formou výpisov databázy pred a po zmenách.

5.3 Testovanie na užívateľoch

Aplikácia nebola ešte masívne testovaná na užívateľoch z dôvodu podporovateľnosti dvoch bankových subjektov. U testovaných osobách neboli hlásené žiadne závažné nedostatky a pripomienky ku funkčnosti aplikácie AFT.

5.4 Testovanie funkčných a nefunkčných požiadaviek

Všetky funkčné a nefunkčné požiadavky boli otestované a splnené okrem požiadavky N1, ktorú sa nepodarilo plne otestovať z dôvodu chýbajúceho operačného systému Mac OS, hoci aplikácia bežala správne pod testovanými OS Windows a aj OS Linux.

Perspektíva

6.1 Perspektíva XML formátu – jednotný štandard pre elektronické výpisy

Usporiadanie informácií v bankových výpisoch z účtov ani ich formát nie sú koordinované. Na trhu existuje veľké množstvo rôznych formátov. Aktuálny stav prináša pre banky aj pre klientov navyše náklady z titulu automatizácie spracovania elektronických výpisov. Slovenská banková asociácia (SBA) spolu s Českou bankovou asociáciou (ČBA) sa podujali vyriešiť pretrvávajúci problém tým, že zdefinovali jednotný národný štandard pre výpisy z účtov poskytované bankami v elektronickej podobe.

„Zámerom štandardu bola snaha o priblíženie sa už k zaužívaným formátom v iných európskych krajinách a preto bolo dohodnuté využitie štandardu ISO 2000222) – camt 053.001.02 vo formáte XML. Vývoj národného štandardu kordinovala SBA v úzkej spolupráci s ČBA. Správu národného štandardu pre XML výpisy z účtu zastrešuje SBA. Popis XML výpisu z účtu je zverejnený na jej web stránke[27].

Elementy XML výpisu zodpovedajú príslušným položkám v norme ISO 20022 „Message Definition Report (MDR) for Bank-to-Customer Cash Management“. Všeobecný popis výpisu označovaný “camt.053.001.02“ – BankToCustomerStatementV02 možno nájsť na www.iso20022.org.“ [28]

Tuzemská legislatíva neukladá povinnosť bankám používanie jednotného štandardu pre elektronické výpisy z účtov. Avšak rozšírenie národného štandardu bude prinášať značné výhody hlavne v oblasti importu dát z účtov vedených rôznymi bankami do firemných ekonomických systémov, pri prechode klienta k inej banke ale aj pri ďalšom analytickom spracovávaní finančných transakcií na účtoch klienta. Jednotné dátové prostredie v podobe štandardizovaných XML výpisov sa stáva pre ďalšie využitie výsledkov bakalárskej práce nevyhnutnou perspektívou.

6.2 Perspektíva AFT

S masívnym nástupom mobilných zariadení bude vhodné, aby AFT existoval aj na mobilnej platforme. Táto myšlienka prináša veľké množstvo možností týkajúcich sa implementácie AFT ako aj jeho nových vylepšení. Jednou z novinek sa očakáva prepočet finančných transakcií do jednotne zvolenej meny cez online výmenný kurz. Otvorenou ostáva oblasť práce s grafmi ako napríklad približovanie a zaostrovanie alebo aj implementácia novej dátovej štruktúry, ktorá by dovoľovala užívateľovi pri kliknutí na daný bod v grafe zobrazíť všetky informácie o transakcií. Novou funkcionalitou AFT by mohla byť aj predikcia vývoja kriviek grafu v najbližom časovom období. Očakáva sa vylepšenie procedúr spojených s vytváraním, ukladaním a mazaním užívateľom zadaných filtrov.

Záver

Cieľom bakalárskej práce je navrhnutie a vytvorenie softvérovej aplikácie na analýzu finančných transakcií, ktoré sú zaznamenávané na bankových účtoch. Zámerom je vytvoriť aplikáciu s jednoduchým ovládaním a užívateľským komfortom. Po analýze problémových oblastí rôznych softvérových riešení, ktoré v súčasnosti ponúkajú banky a ostatné nebankové subjekty, nasleduje spracovanie požiadavky a návrh riešenia. Vytvoreniu aplikácie však predchádza analýza vhodnej technológie, dátovej základne a grafického zobrazenia štatistických informácií uskladnených v bankových výpisoch. V práci je zadaná požiadavka na aplikáciu, návrh relačného modelu databázy spolu s diagramom balíčkov. Softvérová aplikácia je vyriešená v prostredí java infraštruktúry, v ktorej je namodelované jednoduché užívateľské menu. Následne sú vytvárané užívateľské filtre a grafické výstupy zo selektovaných finančných transakcií. Navrhnuté softvérové riešenie je oproti existujúcim bežným aplikáciám efektívnejšie z pohľadu otvorenosti projektu. Vytvorenú aplikáciu je možné implementovať do rôznych operačných prostredí, s možnosťou pridávania bankových dát od ďalších bankových subjektov. Aplikácia bude slúžiť pre potreby klientov bánk ako ďalšia softvérová nadstavba k existujúcim riešeniam, ktoré ponúkajú komerčné banky v súčasnosti. Keďže od budúceho roka začne platiť nová európska smernica o platobných systémoch s orientáciou na voľný pohyb bankových služieb PSD2[29], otvára sa obrovský priestor pre tretie strany (technologické firmy zamerané na finančný sektor), ktorým banky musia poskytnúť prístup do svojich bankových systémov. Trend štandardizácie elektronických výpisov v štruktúre XML dáva preto silný predpoklad využitia bankových informácií v unifikovanom prostredí aj prostredníctvom analyzátorov finančných transakcií, o ktorých pojednáva táto bakalárska práca.

Literatúra

- [1] Pros and Cons to Web and Desktop Applications. Dostupné z: <https://blog.computerlagoon.com/2015/08/04/pros-and-cons-to-web-and-desktop-applications/>
- [2] Bridge, R.: Open source software - The advantages & disadvantages. *Entrepreneur Handbook [online]*, [cit. 2017-05-12]. Dostupné z: <http://entrepreneurhandbook.co.uk/open-source-software/>
- [3] Orgonáš, J.: Banka v mobile: Prima banka. [cit. 2017-03-05]. Dostupné z: <http://banky.sk/banka-v-mobile-prima-banka-9cast/>
- [4] Wallet - Budget Tracker. [online], [cit. 2017-03-12]. Dostupné z: <http://budgetbakers.com/>
- [5] Money for Mac. [online], [cit. 2017-02-05]. Dostupné z: <http://ibearmoney.com/money-mac.html>
- [6] HomeBank. *Gizmos Freeware - elektrotechnický review žurnál [online]*, [cit. 2017-02-02]. Dostupné z: <http://www.techsupportalert.com/content/homebank.htm-0>
- [7] Chemel, R.: Pros and Cons of AngularJS. [cit. 2017-01-05]. Dostupné z: <http://blog.backand.com/pros-and-cons-of-angularjs/>
- [8] vaadin. [cit. 2017-01-20]. Dostupné z: <https://vaadin.com/home>
- [9] ORACLE: *What Is JavaFX?* [cit. 2017-01-25]. Dostupné z: <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>
- [10] Morris, S.: *JavaFX in Action*. Manning Publications, 2010.
- [11] SQLite Reviews. [online], [cit. 2017-01-08]. Dostupné z: <https://www.g2crowd.com/products/sqlite/reviews>

- [12] SQLite Java. [online], [cit. 2017-01-10]. Dostupné z: <http://www.sqlitetutorial.net/sqlite-java/>
- [13] Dugam, A.: What are the advantages and disadvantages of PostgreSQL? *Quora*, [cit. 2017-02-04]. Dostupné z: <https://www.quora.com/What-are-the-advantages-and-disadvantages-of-PostgreSQL>
- [14] Data Presentation: Pie Charts. Technická zpráva, Barcelona Field Studies Centre S.L., [cit. 2017-05-01]. Dostupné z: <http://geographyfieldwork.com/DataPresentationPieCharts.htm>
- [15] Data Presentation: Bar Graphs. Technická zpráva, Barcelona Field Studies Centre S.L., [cit. 2017-05-01]. Dostupné z: <http://geographyfieldwork.com/DataPresentationBarCharts.htm>
- [16] Dostupné z: <https://infogr.am/blog/the-line-chart-how-and-when-to-use-it/>,note=
- [17] Skau, D.: The Pros and Cons of Scatterplots. [online], [cit. 2017-05-02]. Dostupné z: <http://www.scribblelive.com/blog/2012/05/30/the-pros-and-cons-of-scatterplots/>
- [18] Area Charts. [online], [cit. 2017-05-02]. Dostupné z: <http://www.fusioncharts.com/chart-primers/area-chart/>
- [19] Wikipedia: Comma-separated values. [online], [cit. 2017-04-01]. Dostupné z: https://en.wikipedia.org/wiki/Comma-separated_values
- [20] Dalik, T.: CSV - What, Why and How. [online], [cit. 2017-04-01]. Dostupné z: <https://www.shopping-cart-migration.com/blog/61-must-know-tips/5985-csv-what-why-and-how>
- [21] Joan, B.: Difference Between XLS and XLSX. *Difference Between*[online], [cit. 2017-04-03]. Dostupné z: <http://www.differencebetween.net/technology/difference-between-xls-and-xlsx/>
- [22] tutorialspoint.com: *Design Pattern - Factory Pattern*. [cit. 2017-02-06]. Dostupné z: https://www.tutorialspoint.com/design_pattern/factory_pattern.htm
- [23] Apache POI - the Java API for Microsoft Documents. [online], [cit. 2017-04-20]. Dostupné z: <https://poi.apache.org/>
- [24] Java client library for Google Maps API Web Services. [pouz. 2017-05-02]. Dostupné z: <https://github.com/googlemaps/google-maps-services-java>
- [25] JavaFX Tooltip Configuration. [pouz. 2017-04-28]. Dostupné z: <https://gist.github.com/darmbrust/9559744d1b1dada434a3>

- [26] GMapsFX. [pouz. 2017-05-01]. Dostupné z: <http://rterp.github.io/GMapsFX/>
- [27] [pouz. 2017-05-01]. Dostupné z: <http://www.sbaonline.sk/sk/>
- [28] xml standard. [online - doc dokument], [cit. 2017-05-01]. Dostupné z: http://www.sbaonline.sk/files/subory/SEPA/xml_standard_sk_v2_2.doc
- [29] Európska smernica PSD2. [cit. 2017-05-01]. Dostupné z: <https://www.etrend.sk/trend-archiv/rok-2016/cislo-37/bankovy-sezam-otvor-sa.html>
- [30] Tidwell, J.: *Designing Interfaces second edition*. O'Reilly, 2010.

Zoznam použitých skratiek

GUI Graphical user interface

XML Extensible markup language

AFT Analyzátor finančných transakcií

CSV Comma-separated values

XLS Spreadsheet (Microsoft Excel) file format

XLSX Microsoft Excel Open XML Format Spreadsheet file

SBA Slovenská banková asociácia

ČBA Česká banková asociácia

ASCII American Standard Code for Information Interchange

FXML XML-based user interface markup language

EBCDIC Extended Binary Coded Decimal Interchange Code

Shift JIS Shift Japanese Industrial Standards

UC Use Case

UI User interface

Obsah priloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe.....	adresár so spustiteľnou formou implementácie
	src	
	impl	zdrojové kódy implementácie
	latex.....	zdrojové súbory a obrázky pre prácu v L ^A T _E X
	text	text práce
	thesis.pdf	text práce vo formáte PDF
	thesis.tex.....	zdrojová forma práce vo formáte L ^A T _E X