

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Aplikace pro sledování a analýzu Cash Flow



2020

Vedoucí práce: Mgr. Jiří Zecpal,
Ph.D.

Petr Bobek

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Petr Bobek
Název práce: Aplikace pro sledování a analýzu Cash Flow
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2020
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Jiří Zaccpal, Ph.D.
Počet stran: 36
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Petr Bobek
Title: Application for Cash Flow monitoring and analysis
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2020
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Jiří Zaccpal, Ph.D.
Page count: 36
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

V této bakalářské práci představuji návrh a implementaci desktopové aplikace pro sledování a analýzu Cash Flow, která byla vytvořena pomocí vývojového prostředí Visual Studio v jazyce C#. Aplikace umožňuje uživateli zadávat v aplikaci budoucí příjmy, výdaje a na jejich základě předpovídat vývoj Cash Flow v následujícím období.

Synopsis

The desktop application for monitoring and analysis Cash Flow represent in this bachelor thesis was created with development environment Visual Studio, C#. This application enables it's users to enter future incomes, expenses and predict development of Cash Flow based on the embedded data.

Klíčová slova: Cash Flow, C#, Visual Studio

Keywords: Cash Flow, C#, Visual Studio

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce Mgr. Jiřímu Zaccpalovi, Ph.D. za cenné připomínky při zpracování této práce. Dále bych chtěl poděkovat rodině a přítelkyni za podporu v průběhu studia.

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

datum odevzdání práce

podpis autora

Obsah

1	Úvod	7
1.1	Požadavky	7
2	Průzkum existujících řešení	8
2.1	Caflou	8
2.2	Bankonaut Flow	9
2.3	Cash Flow Patrol	9
2.4	Shrnutí	10
3	Použité technologie	10
3.1	C#	10
3.2	Window Presentation Foundation	11
3.3	XML	11
3.4	XAML	12
3.5	Knihovny	12
4	Programátorská příručka	13
4.1	Logická část	13
4.1.1	Třídy	13
4.1.2	Algoritmy	18
4.2	Grafická část	23
5	Uživatelská příručka	25
5.1	Instalace	25
5.2	Hlavní okno	25
5.3	Hlavní menu	25
5.4	Horní menu	25
5.5	Graf	26
5.6	Tlačítka Příjem a Výdaj	27
5.7	Záznamy	28
5.8	Faktury	29
5.9	Kategorie	29
5.10	Import / Export dat	30
5.11	Posílání emailu	31
5.12	Nápověda	31
	Závěr	32
	Conclusions	33
	A Obsah přiloženého CD/DVD	34
	Literatura	35

Seznam obrázků

1	Caflou	8
2	Bankonaut	9
3	Hlavní menu	26
4	Graf	27
5	Tlačítko Příjem	27
6	Položka Záznamy	28
7	Kategorie	30
8	Specifikace nahrávaného souboru typu .xlsx	30

Seznam tabulek

Seznam vět

Seznam zdrojových kódů

1	Část zdrojového kódu pro při přidání záznamu	19
2	Přidání fakturovaného a současně opakovaného záznamu	20
3	Výpočet celkových příjmů a výdajů	22
4	Výpočet očekávaných peněz	22

1 Úvod

Cash flow, peněžní toky, kontrola nad financemi, dostatek příjmů, pokrytí veškerých výdajů. To jsou základní myšlenky a obavy, které provází většinu podnikatelů, běžných domácností, studentů či sportovních klubů. Základem úspěšného provozu téměř čehokoli je mít finance dostatečně pod kontrolou. S tím souvisí snaha předpovídat a dívat se do budoucnosti – kolik můžu utratit či investovat.

Ze začátku si jistě lze vystačit pouze s tužkou a papírem jednoduchými tabulkami a zápisky kolik vydělám a utratím. Časem ovšem s nárůstem starostí, výdajů a příjmů je potřeba použít složitější systémy, například tabulky Excelu a různé podpůrné programy. Pro firmy, podniky či sportovní kluby už je potřeba použít komplexnější aplikaci, která bude schopna kontrolovat a předpovídat finanční toky s menší námahou, a nebo si najmout finančního poradce, což by byla ovšem další investice.

Úkolem aplikací zaměřené na cash flow je schopnost vkládat finanční hodnoty, které byly provedeny jednorázově nebo opakovaně, a na jejich základě vyhodnotit budoucí situaci. Pokud bychom se zaměřili na již zmíněné sportovní kluby, pak mezi jejich příjmy patří především členské příspěvky, sponzorské dary, peníze za vstupenky. Co se týká výdajů, tam se jedná především o pronájmy hal, dopravních prostředků, vybavení pro členy a klub, plat zaměstnanců. Všechny tyto data je potřeba zohlednit při určování Cash flow a finančního stavu klubu.

Cílem práce je tedy vytvořit aplikaci, která by umožňovala sledovat a vyhodnocovat analýzu cash flow.

1.1 Požadavky

Aplikace by měla podle zadání splňovat následující požadavky:

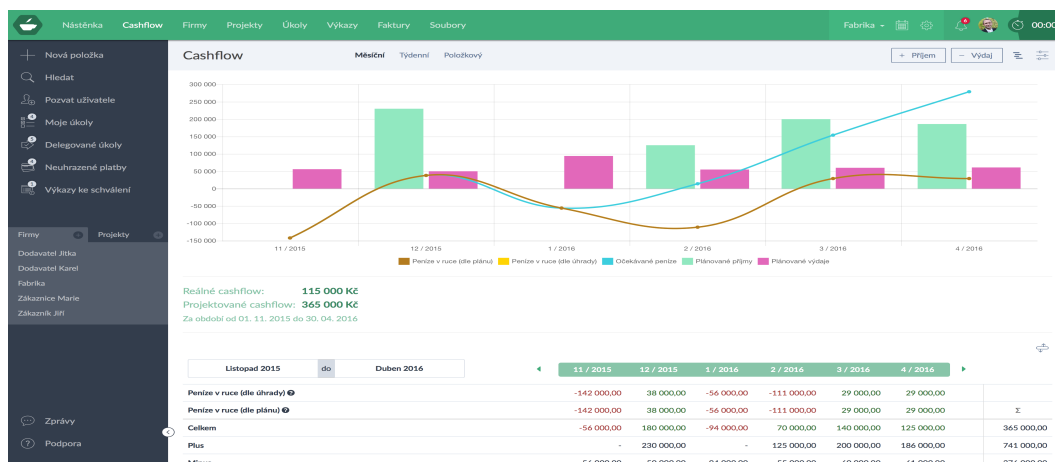
- Aplikace by měla umět předpovídat vývoj Cash Flow v následujícím období.
- Uživatel by měl možnost přímo v aplikaci zadávat budoucí příjmy a výdaje.
- Aplikace musí umět načítat vstupní data ze souboru a naopak exportovat informace o Cash Flow za dané časové období.

2 Průzkum existujících řešení

2.1 Caflou

Caflou (Obr. 1) je aplikace pro malé firmy a freelancery. Název aplikace plyne ze slov Cash a Flow a cílem je sledovat tok peněz a příjmů a následná predikce peněžních toků. Jedná se o celistvou aplikaci na organizaci práce, řízení Cash-Flow, fakturaci a jiné podnikatelské potřeby týkajících se financí. To vše bylo vytvořeno na základě potřeb z praxe a v podstatě se jedná o sadu nástrojů potřebnou k podnikání a organizaci své práce, přičemž vše podstatné nalezneme na jednom místě, online a v cloudu. Caflou je jednak ve formě aplikace webové, ale nabízí také aplikaci i pro mobilní platformu (Android a iOS). Je nabízena ve verzi placené, ale i verze zdarma, která je však omezena datovým prostorem. Mezi hlavní funkcionality patří: [1]

- správa zákazníků a dodavatelů
- řízení projektů a úkolů
- výkazy práce, času, materiálu, ...
- faktury a nabídky
- správa dokumentů
- kalendář



Obrázek 1: Caflou

Aplikace nabízí zobrazení aktuálních nebo plánovaných příjmů a výdajů v grafech a tabulkách (obr. 1) a po ruce je vždy přehled Vašich klientů nebo obchodních partnerů. Dále lze výkazy, projekty, úkoly, příjmy nebo výdaje snadno exportovat do pdf, xls nebo csv. To vše umožňuje dosáhnout vyšší úrovně kontroly nad podnikáním a vede k vyšší prosperitě. [1]

2.2 Bankonaut Flow

V roce 2018 představila společnost Bankonaut s.r.o. aplikaci s názvem Bankonaut Flow (Obr. 2), která se na rozdíl od konkurenčních aplikací podobného typu zabývá především budoucí finanční situací svých zákazníků. Aplikace je dostupná jak ve formě webové aplikace, tak i ve formě mobilní platformy. Účelem je evidovat nejen uskutečněné transakce, ale především ty, které zákazníci teprve čekají, a očekávané transakce si lze zaznamenat předem do finančního kalendáře, například nájem nebo výplata. Existuje ve dvou tarifech, a to bezplatném a premium. V premium tarifu lze opakované transakce vkládat s různou volbou intervalu, díky čemuž můžeme většinu transakcí vložit pouze jednou. [2]

LEDEN 2018			ÚNOR 2018			BŘEZEN 2018		
		ZŮSTATEK			ZŮSTATEK			ZŮSTATEK
01	0	0	01	0	7 000	01	0	25 050
02	0	0	02	0	7 000	02	0	25 050
03	0	0	03	0	7 000	03	0	25 050
04	+ 50 000	50 000	04	+ 50 050	57 050	04	+ 48 800	73 850
05	- 15 000	35 000	05	- 15 000	42 050	05	- 15 000	58 850
06	0	35 000	06	0	42 050	06	0	58 850
07	0	35 000	07	0	42 050	07	- 80 000	-21 150
08	- 10 000	25 000	08	0	42 050	08	0	-21 150
09	0	25 000	09	0	42 050	09	0	-21 150
10	0	25 000	10	- 17 000	25 050	10	0	-21 150
11	0	25 000	11	0	25 050	11	0	-21 150

Obrázek 2: Bankonaut

Aplikace je napsána tak, aby se použitá finanční data nedala nijak zneužít a potenciálnímu útočníkovi byla k ničemu. Jednou z bezpečnostních pojistek je to, že se zůstatky vůbec neukládají. Dále u transakcí nejde odhadnout řád, ve kterém se uživatel pohybuje a nelze tedy zjistit konkrétní měnu dané transakce, pouze číselnou hodnotu. A v neposlední řadě nelze u transakcí poznat, zda jsou pravé nebo ne, čímž se dostáváme k dalšímu využití aplikace. Tím jsou tzv. testovací transakce, což znamená, že si lze zadat chystanou platbu, například dovolenou či koupi automobilu, a zjistit, v jaké finanční situaci se bude uživatel poté nacházet. [3]

2.3 Cash Flow Patrol

Jedná se o mobilní aplikaci a slouží k hlídání výdajů uživatele, spravování osobních financí nebo rodinného rozpočtu. Pomocí této aplikace lze již po první měsíci evidence příjmů a výdajů zjistit, kde můžeme ušetřit a kde máme finanční rezervy.

Aplikace CFpatrol má 4 základní funkce

- Rozdělení do položek pomocí různých kategorií (různá časová období – týdenní, měsíční)

- Zadávání do aplikace (příjmy a výdaje, zadávání v cizích měnách, šablony pro opakující se transakce, přidávání fotografií k transakcím)
- Sledování příjmů a výdajů (filtrování transakcí, detailní výsledky peněžních toků, grafy kategorií)
- Zálohování dat [4]

2.4 Shrnutí

Aplikací, které se zabývají problematikou CashFlow je celá řada. Zde byli uvedeny jen základní, jejichž hlavní funkcí je analýza cash flow. Existují ovšem i daleko komplexnější aplikace, kde CashFlow tvoří pouze zlomek využití a jsou tedy mnohem složitější. Všechny uvedené aplikace mají velmi podobný základ – uživatel vloží do systému data příjmů a výdajů a zjistí své finanční možnosti a jistoty. Aplikace Caflou je vhodná spíše pro použití firem a podniků, zatímco Bankonaut Flow či Cash Flow Patrol jsou vhodné i pro běžné občany a domácnosti. Výhodou Bankonaut Flow oproti Patrolu je to, že kromě mobilní verze poskytuje i webovou.

3 Použité technologie

Kapitola obsahuje stručný popis technologií použitých k vývoji aplikace. Jako vývojové prostředí bylo použito Visual Studio 2019 a programovacím jazykem je C#.

3.1 C#

Jazyk C# byl vyvinut firmou Microsoft a jak vyplývá z názvu, tento jazyk z velké části vychází z programovacího jazyka C a C++, ovšem v mnoha ohledech je bližší spíše programovacímu jazyku Java. Byl představen s vývojovým prostředím .NET. [5] Mezi základní charakteristiky jazyka patří:

- Jedná se o čistě objektově orientovaný jazyk
- Obsahuje nativní podporu komponentového programování
- Podobně jako Java obsahuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní
- Kromě členských dat a metod přidává vlastnosti a události
- Podporuje zpracování chyb pomocí výjimek
- Zajišťuje zpětnou kompatibilitu se stávajícím kódem jak na binární tak na zdrojové úrovni

- Automatická správa paměti – o korektní uvolňování zdrojů aplikace se stará garbage collector

Značná část z uvedených vlastností vychází přímo z funkcionality vývojového rámce .NET a jazyk C# je integrován do vývojového prostředí Visual Studio.NET.

Jazyk C# se řadí mezi tzv. jazyky s virtuálním strojem, ve kterých je zdrojový kód přeložen do tzv. mezikódu, což je strojový (binární) kód přímo podporující objektové programování. Tento mezikód je poté relativně rychle interpretovatelný tzv. virtuálním strojem. Výsledkem je strojový kód pro náš procesor.

Co se týče překladačů jazyka C#, tak ty jsou case sensitive a rozlišují velká a malá písmena. V jazyce C# je zavedeno několik konvencí a jednou z nich je, že jména balíků, tříd, rozhraní a většiny dalších položek začínají velkými písmeny. Malými písmeny začínají privátní a chráněné atributy, lokální proměnné a parametry. [6]

3.2 Window Presentation Foundation

Window Presentation Foundation (WPF) je framework pro komplexní tvorbu bohatých formulářových aplikací a je součástí .NET frameworku. Jeho výhodou je, že umožňuje bohaté stylování vzhledu aplikace. Jeho starší verzí je Windows Forms, ten ale u graficky bohatších aplikací selhává. [7]

Vývojová platforma WPF podporuje rozsáhlou sadu funkcí sloužících pro vývoj aplikací, zahrnující aplikační model, prostředky, ovládací prvky, grafiky, rozložení, dokumentů a zabezpečení.

WPF je novější přístup pro vytváření GUI (Graphical User Interface) aplikací, tzv. grafické uživatelské rozhraní. Uživatel v této aplikaci pracuje skrze grafické prvky.

Framework nabízí značné množství již hotových komponent, které lze použít pro tvorbu grafického rozhraní a výslednou aplikaci si z nich lze jednoduše poskládat. Jedná se především o různá tlačítka, pole, posuvníky či další komponenty, které Microsoft nazývá controls, tzv. ovládací prvky. Je možné si vytvořit vlastní ovládací prvky v případě, že již hotové nevyhovují. [8]

3.3 XML

XML byl vyvinut v roce 1996 organizací World Wide Web Consortium (W3C) a je odvozen od SGML (Standard Generalized Markup Language). XML vznikl díky tomu, že v praxi se z SGML používala jen jeho část, která byla později použita pro tvorbu XML. [9]

XML dokument je složen z elementů a struktura dokumentu je stromová což znamená, že jednotlivé elementy v sobě mohou obsahovat jakékoliv množství dalších elementů, ty zase další elementy atd. Platí ale, že každý XML dokument obsahuje právě jeden kořenový element, a v tom jsou umístěny další podelementy. Elementy se zapisují do lomených závorek v nichž je obsažen název elementu.

Elementy do sebe mohou být tedy vnořeny, ale nemohou se překrývat. XML tagy jsou case sensitive. [10]

Hlavní výhodou souborů xml je jeho použitelnost. Většina aplikací totiž tento způsob předávání dat podporuje a díky strukturovaným datům dokáže rozlišit i jejich důležitost.

V dnešní době se XML používá především pro snadno výměnu informací a komunikaci, jenž je nezávislá na konkrétní aplikaci či platformě. Hlavní výhodou XML oproti jiným formátům používaným pro přenos informací je jeho nezávislost, standardizace, poměrně malá velikost, podpora národních kódování a jednoduchý převod na jiné formáty.

3.4 XAML

WPF využívá jazyk XAML (Extensible Application Markup Language), což je značkovací jazyk pro design formulářů aplikace. Jedná se o variaci jazyka XML (Extensible Markup Language), tzv. rozšiřitelný značkovací jazyk. XAML byl vytvořen pro tvorbu uživatelského rozhraní aplikací. XAML není vázán pouze na WPF, ale v této formě je aktuálně nejpoužívanější a slouží především k usnadnění zápisu. [11]

Obvykle se využívá ke tvorbě oken, dialogových oken, stránek a uživatelských ovládacích prvků a k jejich vyplnění ovládacími prvky, obrázky a grafikou. [12]

3.5 Knihovny

Zde jsou stručně popsány knihovny, které jsou nutné k běhu hlavním funkcí aplikace. Všechny uvedené knihovny jsou NuGet balíčky a Open Source.

Material Design WPF

Patří mezi nejpoblárnější knihovny pro GUI. Mezi její funkce lze zařadit stylování základních ovládacích prvků WPF (např. tlačítka), samostatné fungování či kompatibilita s dalšími knihovnami. [13]

LiveChart

Knihovna umožňující vizualizaci dat pro .NET ve formě grafů. Podporuje WPF a WinForms. [14]

SharpPDF

Pomocí této knihovny lze tvořit PDF soubory. Knihoven s touto funkcí je vícero, ovšem pro potřeby tvořené aplikace byla zvolena knihovna SharpPDF, díky své jednoduchosti a přehlednosti. Snadno vytváří PDF dokumenty z jakéhokoliv jazyka .NET. [15]

4 Programátorská příručka

Programátorská příručka popisuje hlavní třídy aplikace, které se podílí na výpočtu a analýze CashFlow. Jsou zde popsány i algoritmy výpočtů a třídění záznamů.

4.1 Logická část

První částí programátorské příručky je logická část, ve které jsou vypsány klíčové třídy aplikace. U jednotlivých tříd budou popsány jejich vlastnosti a metody.

4.1.1 Třídy

Třída **Record**

Obsahuje veškeré informace o záznamech a vlastnosti této třídy jsou následující:

- `DueDate` - datum splatnosti záznamu, podle něhož je záznam přiřazen do patřičného měsíce
- `PayDate` - datum úhrady záznamu, nabývá hodnoty `null` v případě, že vlastnost `Paid` je nastavena na `false`
- `Name` - název záznamu
- `Money` - částka
- `Type` - může nabývat dvou hodnot: `Income` (Příjem) a `Issue` (Výdaj)
- `Category` - kategorie záznamu
- `Paid` - je datového typu `bool` a určuje, zda byl záznam uhrazen
- `Invoiced` - je datového typu `bool` a určuje, zda byl záznam fakturován
- `Repeat` - udává zda je záznam opakovaný

Třída **Repetition**

Uchovává informaci o tom, zda je záznam opakovaný a kolikrát. Vlastnosti:

- `ItRepeats` - určuje, zda je záznam opakovaný či ne, nabývá dvou hodnot: `True` / `False`
- `NumberOfRepeats` - říká, kolikrát se daný záznam bude opakovat. V případě, že záznam je opakovaný a hodnota `NumberOfRepeats` je záporná, tak opakování nikdy neskončí.

Metoda:

- `ShrinkNumberOfRepeats` - snižuje počet opakování o zadanou hodnotu (v našem případě o 1). Uplatní se u záznamů s limitovaným počtem opakování.

Třída `ManagerRecords`

Třída `ManagerRecords` spravuje všechny záznamy a má na starosti jejich ukládání a načítání ze souboru. Nejdůležitější vlastnost je `Records`, což je seznam všech záznamů včetně faktur. Dále je zde vlastnost `Invoices`, což je seznam pouze fakturovaných záznamů.

Mezi hlavní metody patří:

- `Add` - přidá záznam do seznamu `Records`
- `Remove` - odstraní záznam ze seznamu `Records`
- `DeleteRecords` - odstraní všechny záznamy ze seznamu `Records`
- `Load` - načte všechny záznamy ze souboru a vloží je do seznamu `Records`
- `Save` - uloží postupně všechny záznamy ze seznamu `Records` do souboru

Třída `Month`

Třída `Month` reprezentuje jednotlivé měsíce. Záznamy jsou do příslušných měsíců přiřazovány podle vlastnosti `DueDate` a jsou v nich rozděleny do seznamů.

V této třídě je celkem 10 klíčových vlastností. 6 z nich jsou seznamy, do nichž se záznamy rozdělují. Na základě těchto seznamů následně dochází k výpočtu dalších 4 vlastností, které jsou viditelné v grafu `CashFlow`.

- `OrdinaryRecords` - seznam běžných (nefakturovaných) záznamů
- `InvoiceRecords` - seznam fakturovaných záznamů
- `PaidRecords` - seznam uhrazených záznamů
- `RepetitionRecords` - seznam záznamů s opakováním
- `RepetitionListOfCopy` - seznam kopií záznamů s opakováním (kopie záznamu se vytvoří v případě, že daný měsíc je posledním zobrazeným v grafu a záznam se má opakovat v následujících měsících)
- `ListRecords` - seznam všech záznamů, ze kterého jsou vypočteny následující vlastnosti vyobrazeny v grafu
- `TotalIncomes` - celkové příjmy za daný měsíc, tato hodnota je v grafu znázorněna zeleným sloupcem

- `TotalIssues` - celkové výdaje za daný měsíc, tato hodnota je v grafu znázorněna červeným sloupcem
- `ExpectedMoney` - jedná se o očekávané peníze a v grafu jsou znázorněny pomocí modré křivky
- `RealMoney` - vyjadřuje peníze na ruce (uhrazené) a v grafu je znázorněna pomocí oranžové křivky

Hlavními metodami této třídy jsou:

- `Add` - přidá záznam do příslušného seznamu daného měsíce
- `Remove` - odstraní záznam z příslušného seznamu daného měsíce
- `NullRecords` - nastaví vlastnosti, které se počítají, na nulovou hodnotu a vyčistí seznam `ListRecords`

Třída `ManagerMonths`

Třída `ManagerMonths` v sobě obsahuje třídu `ManagerRecords`. Její hlavní vlastností je `MonthsList`, která si udržuje seznam měsíců.

Mezi významné metody lze zařadit:

- `Add` - přidá záznam do patřičného měsíce na základě vlastnosti `DueDate`
- `Remove` - odstraní záznam z patřičného měsíce
- `DeleteLists` - odstraní veškeré měsíce ze seznamu `MonthsList`
- `Edit` - slouží k úpravě záznamu
- `Calculate` - jedná se nejspíše o nejvýznamnější výpočetní metodu aplikace. Její součástí jsou další metody pracující se seznamy záznamů. Dochází zde k přesunům záznamů do seznamu `ListRecords` ve třídě `Month`. Ze zmíněného seznamu jsou prováděny veškeré výpočty pomocí dalších metod. Konkrétně zde dochází k vypočtení vlastností `ExpectedMoney`, `RealMoney`, `TotalIncomes`, `TotalIssues` umístěných ve třídě `Month`. Příklady metod aplikovaných v metodě `Calculate`:
 - `MoveRepeatRecords`
 - `MoveOrdinaryRecords`
 - `MoveInvoiceRecords`
 - `CalculateExceptedMoney`
 - `CalculateRealMoney`
- `Load` - načte všechny záznamy ze seznamu `Records` a roztrídí si je do patřičných měsíců

- LoadCopyRecords - načte kopie záznamů ze souboru a uloží si je to seznamu RepetitonListOfCopy
- SaveCopyRecords - uloží kopie záznamů do souboru

Třída ManagerMain

Třída ManagerMain v sobě obsahuje třídu ManagerMonths a v grafické části popsanou třídu ManagerGraph. Jsou v ní popsány cesty k souborům. Strukturně se jedná o nejvýše postavenou třídu, která v sobě obsahuje veškeré klíčové metody pro výpočet CashFlow a zobrazení.

Hlavními vlastnostmi této třídy jsou:

- Categories - seznam všech kategorií
- ListMonths - seznam měsíců, které si uživatel zobrazí v grafu
- AllListRecords - seznam nefakturovaných záznamů, které jsou v aplikaci viditelné v položce *Záznamy*
- AllInvoiceRecords - seznam fakturovaných záznamů, které jsou v aplikaci viditelné v položce *Faktury*
- DateFrom - udržuje informaci o tom, od jakého data si chce uživatel zobrazit záznamy
- DateTo - udržuje informaci o tom, do jakého data si chce uživatel zobrazit záznamy
- FilterListRecords - seznam vyfiltrovaných záznamů (bez faktur)
- FilterListInvoice - seznam vyfiltrovaných faktur

Metody patřící do této třídy:

- Add,Edit, Remove - volají příslušné metody z třídy ManagerMonths a zároveň metody Add a Edit aktualizují seznam Categories
- CalculationView - stará se o celkový výpočet, skládá se ze tří metod:
 - Calculation - volá metodu Calculate ze třídy ManagerMonths, která se stará o logický výpočet
 - View - na základě metody Calculate zprostředkovává vypočtené vlastnosti. Ty jsou použity v grafu i tabulkách napříč aplikací. Podrobněji je popsána v grafické části.
 - Save - zavolá metodu Save ze třídy ManagerMonths a uloží veškeré záznamy do souboru

- `LoadDates` - načte datумы ze souboru, které uživatel naposledy zadal. V případě, že žádné datумы zatím nebyly zadány, načtou se data od ledna 2020 do aktuálního data
- `SaveDates` - uloží naposledy zadané datумы do souboru
- `FilterRecords` - sestaví `FilterListRecords` na základě předaných parametrů z filtrace, případně `FilterListInvoice`, jedná-li se o filtraci faktur

Třída `DateMonthYear`

Vzhledem k tomu, že v rámci aplikace se pracuje s formátem: měsíc, rok (bez dne), bylo vhodné vytvořit si třídu reprezentující daný formát. K tomu slouží třída `DateMonthYear`. Její vlastnosti jsou `Year` a `Month`.

Třída `Excel`

Má na starosti správu jednotlivých buněk v souboru `Excel` a uložení / načtení celého souboru. Metody:

- `CreateNewFile` - vytvoří nový `Excel` soubor s jedním listem
- `ReadCell` - přečte buňku na zadaných souřadnicích
- `WriteToCell` - zapíše do buňky na určených souřadnicích
- `SaveAs` - uloží soubor na cestě, kterou si zvolí uživatel
- `Close` - uzavře `Excel` soubor

Třída `ManagerExcel`

Obsahuje v sobě třídu `Excel` a stará se o veškeré listy a soubory typu `Excel`. Třída `Excel` je schopna zapisovat a číst z a do buněk, ale rozsah těchto buněk je určen třídou `ManagerExcel`.

Její součástí jsou pouze tři metody:

- `Load` - projde buňku po buňce a za pomoci metody `ReadCell` vytvoří nový záznam
- `SaveAs` - pomocí metody `WriteToCell` zapíše záznamy do `Excel` souboru a uloží jej na uživatelem zvolené místo
- `Close` - uzavře `Excel` soubor

Třída Pdf

U této třídy je podobný princip jako u třídy Excel. Opět zde dochází k vyplňování jednotlivých řádků a sloupců výsledného Pdf souboru.

Hlavní metody:

- `CreateDocument` - vytvoří nový dokument typu Pdf včetně titulku
- `SaveDocument` - uloží dokument na místo zvolené uživatelem
- `SetTablesToPages` - přidá na stránku tabulku o definovaných rozměrech
- `CreatePages` - vytvoří počet stran na základě předaného parametru
- `CreateTables` - vytvoří počet tabulek na základě předaného parametru
- `FillRow` - zapíše záznam do patřičného řádku v tabulce (záznam a řádek jsou předávanými parametry)

Třída ManagerPdf

Zde dochází k výpočtu počtu stran a tabulek, které budou vytvořeny. Výpočet je prováděn na základě počtu záznamů, přičemž je nastaveno, že na jedné stránce bude 30 záznamů. Metody:

- `AddRangeRecords` - přijímá seznam záznamů, ze kterého se poté tvoří soubor Pdf
- `CreatePdf` - dochází zde k výpočtu množství tabulek, jejíž řádky jsou naplněny metodou `FillRow`. Následně je vypočten počet stran pro daný dokument (podle počtu záznamů) a příslušné stránky jsou vyplněny vytvořenými tabulkami. Nakonec se soubor uloží.

4.1.2 Algoritmy

Algoritmus přidání záznamu do příslušného měsíce

Přidělení záznamů do příslušného měsíce má na starosti metoda `AddRecord`, která se nachází ve třídě `ManagerMonths`.

Metoda je volána ve dvou případech. Prvním je, když uživatel zadá do aplikace záznam přes tlačítko *Příjem*. V tomhle případě se jedná o kaskádovitý proces, ve kterém vše začíná ve třídě `ManagerMain`. Ta vyšle pokyn do třídy `ManagerMonths`, že si má přidat nový záznam a ta stejný pokyn zašle do třídy `ManagerRecords`, kde se záznam přidá do všech záznamů metodou `Add`. Ve třídě `ManagerMonths` se pak provede metoda `AddRecord`, která přidělí záznam do patřičného měsíce v závislosti na vlastnosti `DueDate`.

Druhým případem, kdy je volána metoda `AddRecord` je načítání záznamů ze souboru, ke kterému dochází při spuštění aplikace.

Na začátku jsou dvě proměnné:

- `createNewMonth` - proměnná určující tvorbu nového měsíce podle vlastnosti `DueDate`
- `createMonthForPaid` - proměnná určující tvorbu nového měsíce podle vlastnosti `PayDate` a vztahuje se tedy pouze k uhrazeným záznamům

Obě jsou v počátku nastaveny na hodnotu `True`. Obě proměnné dávají informaci o tom, zda je potřeba tvořit objekt s novým měsícem, nebo už se v systému takový měsíc vyskytuje a k jeho tvorbě nedojde. Při zařazování záznamů do adekvátního měsíce se prochází seznam `MonthsList` a je srovnávána vlastnost `DueDate` a datum záznamu ve formátu `DateMonthYear`. V případě, že se data shodují, dojde k přidání záznamů do nalezeného měsíce a proměnná `createNewMonth` je nastavena na `False`. Poté se stejným způsobem zjistí, zda je záznam uhrazen ve stejném měsíci jako je jeho datum splatnosti, a při shodě je přidán (Zdrojový kód 1).

Jestliže se srovnávaná data nebudou shodovat, proměnná `createNewMonth` zůstane nastavena na `True` a dojde ke tvorbě nového objektu měsíce odpovídající datu splatnosti nového záznamu. U proměnné `createMonthForPaid` je tomu stejně - v případě, že je datum úhrady odlišné od data splatnosti, dojde k vytvoření nového měsíce a nebo dohledání již existujícího.

```
1 private void AddRecord(Record record)
2 {
3     bool createNewMonth = true;
4     bool createMonthForPaid = true;
5     foreach(Month month in MonthsList)
6     {
7         if (month.EqualsWithDueDate(record))
8         {
9             month.Add(record);
10            createNewMonth = false;
11
12            if (record.PayDate == null)
13                createMonthForPaid = false;
14        }
15
16        if (month.EqualsWithPayDate(record))
17        {
18            month.PaidAdd(record);
19            createMonthForPaid = false;
20        }
21    }
```

Zdrojový kód 1: Část zdrojového kódu pro při přidání záznamu

Algoritmus přidání záznamu do seznamu

Přidání záznamu do příslušného seznamu je řízeno metodou `Add`, která náleží do třídy `Month`. Aplikuje se v případě, že je přidán nový záznam. Nejprve je přidělen do správného měsíce a poté je v daném měsíci rozdělen do správnému seznamu. Nejsložitějším mechanismem celém aplikace je práce s opakovanými záznamy. Ty je totiž nutno uchovávat, aby mohli být posílány dalším měsícům a z toho důvodu mají vlastní seznam, který s nimi pracuje. Fakturované záznamy a běžné záznamy jsou také odděleny. Celkem se jedná tedy o 3 seznamy (vlastnosti):

- `OrdinaryRecords` - seznam běžných (nefakturovaných) záznamů
- `InvoiceRecords` - seznam fakturovaných záznamů
- `RepetitionRecords` - seznam záznamů s opakováním

Na základě algoritmu (Zdrojový kód 2) dochází k roztrídění záznamů do těchto seznamů. Třídění probíhá podle několika podmínek, které následují po sobě. Nejprve se zjistí, zda je záznam fakturovaný a současně opakovaný. V případě že ano, tak je automaticky zařazen do fakturovaných záznamů. Potom je nutné zjistit počet opakování tohoto záznamu, a podle toho je přidělen do ostatních seznamů. Pokud má vlastnost `NumberOfRepeats` rovnou nule, přidá se k záznamům běžným. Pokud je hodnota rozdílná od nuly, přidá se k opakovaným záznamům.

```
1 if(record.Invoiced && record.Repeat.ItRepeats)
2 {
3     _invoiceRecords.Add(record);
4
5     if(record.Repeat.NumberOfRepeats == 0)
6         _ordinaryRecords.Add(record);
7
8     else
9         _repetitionRecords.Add(record);
10 }
```

Zdrojový kód 2: Přidání fakturovaného a současně opakovaného záznamu

Jestliže je záznam pouze fakturovaný, je zařazen do fakturovaných.

Další podmínkou je dotaz, zda je záznam pouze opakovaný a opět dochází ke zjištění jeho `NumberOfRepeats`. V případě nulové hodnoty je zařazen do běžných záznamů, u nenulové hodnoty se přidá k záznamům opakovaným. V případě, že neplatí žádná z výše uvedených podmínek, tak je záznam přidán k záznamům běžným.

Algoritmus výpočtu

Při výpočtu se pracuje se dvěma druhy seznamů - buď pouze s těmi měsíci, které si uživatel chce zobrazit v aplikaci (`MonthsFromUser`), nebo s kompletním seznamem všech měsíců, které kdy byli do aplikace zadány (`MonthsList`).

Výpočet probíhá ve třídě `ManagerMonths` pomocí metody `Calculate`, která je volána ze třídy `ManagerMain`. Metoda přijímá 2 parametry, a sice datum od a do. Jedná se o údaje, které uživatel zadal v aplikaci a určují zobrazovaný a počítaný časový rozsah (počet měsíců). Součástí výpočtu je několik metod, které se na procesu podílí:

- `FillMonths`

Pracuje s parametry od do a vlastností `MonthsFromUser`. Ta je na začátku vyprázdněna a následně jsou do ní vloženy měsíce z časového rozsahu zadaného od uživatele. Účelem této metody je doplnit měsíce, které se v daném intervalu nachází, ale nejsou v nich žádné záznamy a není pro ně vytvořen objekt, který by se do `MonthsFromUser` vložil.

- `NullAllRecords`

Volá metodu `NullRecords` ze třídy `Month`, která vynuluje veškeré číselné vlastnosti a seznam `ListRecords`. Z tohoto seznamu dochází k výpočtu ostatních hodnot.

- `MoveRepeatRecords`

Základním principem je přesun opakovaných záznamů do `ListRecords` a do dalších měsíců. Podrobněji je popsán v sekci Algoritmus opakování níže.

- `MoveOrdinaryRecords`

Přidává všechny běžné záznamy do seznamu `ListRecords`.

- `MoveInvoiceRecords`

Přidává všechny fakturované záznamy do seznamu `ListRecords`.

- `MoveMoney`

V této metodě dochází k výpočtu celkových příjmů, výdajů a uhrazených peněz. Pro každý měsíc je zavolána metoda `SetTotalMoney` a `SetTotalPaidMoney`. V těchto metodách jsou procházeny záznamy v seznamu `ListRecords` a podle typu jsou připočteny buď k vlastnosti `TotalIncomes` v případě příjmu, a nebo k vlastnosti `TotalIssues` v případě výdajů (Zdrojový kód 3).

Metoda `SetTotalPaidMoney` probíhá stejně s tím rozdílem, že je procházen seznam `PaidRecords`.

```

1 public void SetTotalMoney()
2 {
3     if (ListRecords != null)
4     {
5         foreach (Record record in ListRecords)
6         {
7             if (record.Type.Equals(Type.Income))
8                 TotalIncomes += record.Money;
9             else if (record.Type.Equals(Type.Issue))
10                TotalIssues += record.Money;
11        }
12    }
13 }

```

Zdrojový kód 3: Výpočet celkových příjmů a výdajů

- CalculateExceptedMoney

Na začátku metody je proměnná `actual` nastavena na 0 a poté prochází seznam měsíců v `MonthsList`. U každého měsíce je vypočtena vlastnost `FinalMoney`, což je rozdíl příjmů a výdajů. Tato vlastnost je poté postupně od prvního měsíce připočítána k proměnné `actual`, která ve výsledku udává očekávané peníze jednotlivých měsíců (Zdrojový kód 4)

```

1 private void CalculateExceptedMoney()
2 {
3     double actual = 0.0;
4     foreach (Month month in MonthsList)
5     {
6         actual += month.FinalMoney;
7         month.ExpectedMoney = actual;
8     }
9 }

```

Zdrojový kód 4: Výpočet očekávaných peněz

- CalculateRealMoney Tato metoda pracuje na stejném principu jako předchozí, liší se pouze v tom, že místo očekávaných peněz se zde počítají peníze v ruce, která je zastoupena vlastností `RealMoney`.

Algoritmus opakování

Má na starosti manipulaci a správu opakovaných záznamů. Základní mechanismus opakovaných záznamů je ten, že jim je na začátku nastavena hodnota `NumberOfRepeats` a ta se postupně snižuje. Tím je zajištěno, že záznamy s

opakováním na určitý počet měsíců se nebudou opakovat neustále, ale skončí kdy mají. Celý proces je rozdělen na 4 fáze.

V první fázi dojde k tomu, že je procházen seznam `RepetitionRecords` a u každého opakovaného záznamu je zjištěna hodnota `NumberOfRepeats` udávající, kolikrát se má záznam ještě opakovat. Pokud je tato hodnota rovna nule, je záznam ze seznamu smazán. V opačném případě je hodnota `NumberOfRepeats` snížena o jedna. Záznamy s opakováním do nekonečna jsou nastaveny na zápornou hodnotu, čímž je zabráněno jejich smazání.

V druhé fázi se vytvoří kopie všech opakovaných záznamů. Vytvořená kopie je uložena do seznamu `RepetitionListOfCopy`.

Ve třetí fázi dojde k tomu, že u záznamů v seznamu `RepetitionRecords` je hodnota `NumberOfRepeats` nastavena na nulu a jsou přesunuty do běžných záznamů a ze seznamu `RepetitionRecords` jsou smazány.

V poslední fázi dochází k přesunu kopií opakovaných záznamů do následujících měsíců. Nejprve se zjistí, zda aktuální měsíc je v seznamu `MonthsList` poslední. Pokud ne, tak dojde k tomu, že jednotlivým záznamům kopií je změněno datum splatnosti na datum následujícího měsíce, a poté je celý seznam `RepetitionListOfCopy` předán dalšímu měsíci a z původního smazán. Pokud nastane druhá varianta, že aktuální měsíc je v seznamu poslední, tak si seznam kopií uloží do souboru. Odtud je při dalším spuštění znovu načten, a pokud v seznamu `MonthsList` přibude další měsíc, tak je mu předán.

4.2 Grafická část

V této části jsou popsány třídy, které se podílejí na grafickém zobrazení záznamů a výpočtů. Součástí grafické části jsou samozřejmě i samotná okna aplikace, ale ty zde nebudou popsány. Bude zde pouze popsána cesta, jakou se do příslušných oken dostávají zobrazované hodnoty.

Třída `ManagerGraph`

Obsahuje několik vlastností, na jejichž základě je potom vykreslen graf. Vlastností jsou následující:

- `Graph` - symbolizuje celý graf
- `Labels` - udává hodnoty na ose x (má v sobě seznam názvů zobrazených měsíců)
- `Formatter` - udává hodnoty na ose y, což jsou peníze s měnou
- `Incomes` - seznam, který má v sobě hodnotu celkových příjmů pro jednotlivé měsíce
- `Issues` - seznam, který má v sobě hodnotu celkových výdajů pro jednotlivé měsíce

- `ExceptedMoney` - seznam, který má v sobě hodnotu očekávaných peněz jednotlivých měsíců
- `RealMoney` - seznam, který má v sobě hodnotu peněz v ruce jednotlivých měsíců

Dále obsahuje metody pro přidání hodnot do výše zmíněných vlastností.

Třída `ManagerMain`

Třída `ManagerMain` byla již zmíněna v logické části této příručky, ale obsahuje metodu `View`, která patří do části grafické, proto je zde zmíněna znovu.

Metoda `View` slouží k předání zobrazovaných hodnot, které byli ve třídě `ManagerMain` vypočteny metodou `Calculate`, do třídy `ManagerGraph`. Jejím dalším úkolem je práce s následujícími seznamy:

- `ListMonths` - do tohoto seznamu je nahrán seznam měsíců, který je obsažen v `MonthsFromUser` a obsahuje tedy pouze měsíce zobrazené uživatelem. Na základě tohoto seznamu jsou vyplněny vlastnosti (seznamy) v `ManagerGraph`
- `AllListRecords` - seznam všech záznamů bez faktur
- `AllInvoiceRecords` - seznam všech fakturovaných záznamů
- `ListRecordsAndInvoices` - seznam kategorií s výdaji a příjmy za jednotlivé měsíce
- `ListGrid` - jedná se o seznam, který v sobě udržuje dalších 5 seznamů, které se zobrazují v tabulce pod grafem (celkové výdaje, celkové příjmy, zůstatek, očekávané peníze, peníze v ruce). Každý z těchto jednotlivých seznamů má v sobě seznam hodnot příslušících k jednotlivým zobrazeným měsícům.

Nejprve jsou tyto seznamy vyprázdněny od předchozích hodnot a poté jsou do nich nahrány aktuální a nově vypočtené hodnoty (většinou po zadání nových záznamů). Takto naplněné seznamy jsou poté v aplikaci zobrazeny v tabulkové formě.

5 Uživatelská příručka

Uživatelská příručka slouží pro popis hlavních funkcí aplikace a uživatel má možnost se zde zorientovat v programu. Může zde vyčíst základní funkce a jak s programem pracovat. Jednou z funkcí aplikace je práce se soubory Excel a pro plnohodnotné využití této aplikace je tedy nutné mít nainstalovaný program Excel.

5.1 Instalace

Před samotným spuštěním aplikace je nutné ji nejprve nainstalovat pomocí instalačního souboru Setup. Poté se na ploše objeví ikona se zástupcem. Jako první se otevře hlavní okno aplikace.

5.2 Hlavní okno

Po spuštění se zobrazí hlavní okno aplikace, ve kterém bude graf a tabulka znázorňující CashFlow. V případě prvního spuštění bude graf prázdný, při dalších spuštěních zůstane uložena poslední pozice včetně posledního zadaného časového rozsahu a uložených dat. Na levé straně okna se nachází hlavní navigační menu.

V pravém spodním rohu je kurzor pro manipulaci s oknem – zvětšení či zmenšení.

5.3 Hlavní menu

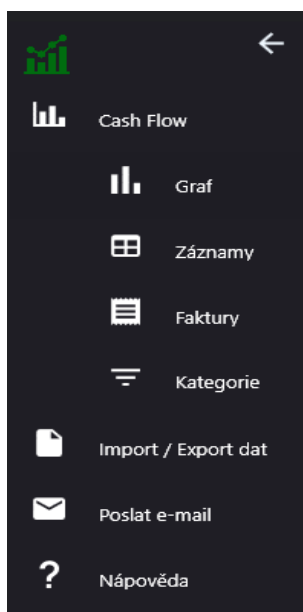
Na levé straně aplikace se vyskytuje hlavní navigační menu, kde lze přepínat mezi funkcemi aplikace (Obr. 3).

Při kliknutí na položku *CashFlow* dojde k otevření podnabídky, kde lze zvolit mezi grafickým nebo tabulkovým vyobrazením dat (*Záznamy* a *Faktury*). Nachází se tam také položka *Kategorie*. Další navigací je *Import / Export dat*, kde je možné pracovat se soubory typu excel a pdf. Následující tlačítko pak slouží pro posílání emailu a na konci se nachází *Nápověda*.

Při zmenšení okna aplikace dojde ke svinutí menu a zobrazeny budou pouze ikony položek.

5.4 Horní menu

V horní části okna se nachází název aplikace a po zvolení některé z položek menu je nahrazena jejím názvem pro snadnější orientaci uživatele. V pravé části se nachází klasické ikony pro ukončení aplikace, její minimalizaci či zmenšení/zvětšení.



Obrázek 3: Hlavní menu

5.5 Graf

Samostatný graf je zobrazen v hlavním okně aplikace po spuštění. Při zvolení položky *Graf* (Obr. 4) v hlavním menu se uživateli zobrazí nejen graf, ale také tabulky s vybranými záznamy.

V levém horním rohu se nachází kolonky s daty, kde uživatel zvolí časový rozsah, ve kterém si chce CashFlow zobrazit, např. 15.1.2020 – 20.5.2020. Vzhledem k tomu, že aplikace je vytvořena pro zobrazování měsíců, datумы se přepíše na první a poslední den daného měsíce. V pravém horním rohu se nachází tlačítka *Příjem* a *Výdaj*, s jejichž pomocí uživatel zadává nové záznamy. Tlačítka jsou popsána dále v textu.

Tabulky pod grafem zobrazují aktuální CashFlow v tabulkově-číselné formě v závislosti na zvoleném datovém rozsahu. V tabulce jsou vypsány pro jednotlivé měsíce celkové příjmy, celkové výdaje, zůstatek (tedy rozdíl příjmů a výdajů), očekávané peníze a peníze v ruce. Hodnoty nižší než nula jsou zobrazeny červeně, aby uživatel ihned viděl, v jakém měsíci má záporné finanční hodnoty.

Na obrázku je vidět, že součástí grafu jsou dvě křivky a sloupce. Zelené sloupce znázorňují příjmy za daný měsíc, červené výdeje. Na základě těchto hodnot jsou vypočteny očekávané peníze a peníze v ruce. Modrá křivka znázorňuje množství peněz podle data splatnosti, tzv. očekávané peníze, které by uživatel měl v danou dobu mít. Oranžová křivka je pro záznamy uhrazené a jedná se o peníze, které má uživatel reálně v ruce a které jsou již zaplacené.

Na ose x v grafu jsou vypsány zobrazené měsíce a na ose y jsou peníze pro příslušný měsíc.



Obrázek 4: Graf

5.6 Tlačítka Příjem a Výdaj

Tlačítka pro zadání nového příjmu (Obr. 5) či výdaje jsou umístěny v horní části aplikace v položkách *CashFlow*. Po kliknutí na tato tlačítka se uživateli zobrazí nové okno, kam uživatel vyplní veškeré potřebné informace.

Okna jsou rozdělena na 3 podokna – základní, popis a opakování. V horní části okna je kolonka pro název záznamu.

Příjem

Název*

Základní	Popis	Opakování
*Částka	09.05.2020	
<input type="checkbox"/> Uhrazeno		
<input type="checkbox"/> Fakturovano		
*Typ		

Storno
Uložit

Obrázek 5: Tlačítko Příjem

Základní podokno obsahuje datum splatnosti, které je implicitně nastaveno na aktuální datum. Kolonka s kalendářem umožňuje toto datum měnit. Vedle

je kolonka pro vyplnění částky, jedná se o povinnou položku. Součástí jsou také dva checkboxy, kde uživatel volí, zda je záznam uhrazen a fakturován. Pokud záznam uhrazen je, objeví se kolonka pro vyplnění data úhrady stejně jako tomu je u data splatnosti. Poslední komponentou je volba kategorie. Tu lze vybrat z již existujících, případně napsat novou, která se uloží ke stávajícím.

Podokno s popisem je značně jednodušší a slouží uživateli k tomu, aby si k danému záznamu mohl napsat různé poznámky či komentáře.

Poslední podokno je s opakováním. Tam se nachází jeden checkbox, kde uživatel určí, zda daný záznam bude opakován. Pokud opakováný bude, pak se objeví kolonky s volbou, kdy má opakování skončit. Uživatel má tři možnosti ukončení – nikdy, po počtu opakování a po datu. Při nastavení na nikdy bude záznam opakován navždy. Při volbě po počtu opakování je nutno zvolit počet měsíců, ve kterých bude záznam opakován a při možnosti po datu je nutno zvolit datum ukončení opakování.

Po kliknutí na tlačítko *Uložit* se záznam vytvoří, tlačítko *Storno* okno zavře bez uložení záznamu.

5.7 Záznamy

V této položce (Obr. 6) jsou vypsány všechny záznamy, které byli do aplikace zadány.

Datum Splatnosti	Datum Úhrady	Název	Částka	Typ	Kategorie	Opakovaný	Fakturováno	Uhrazeno
10. února 2020	01. března 2020	Členské poplatky	1 000,00	Příjem	členské	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. března 2020		Členské poplatky	1 000,00	Příjem	členské	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20. března 2020	20. května 2020	Sponzorský dar	3 000,00	Příjem	sponzor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
01. dubna 2020	01. dubna 2020	Rozhodčí	1 500,00	Výdaj	rozhodčí	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. dubna 2020		Členské poplatky	1 000,00	Příjem	členské	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
09. května 2020		Vstupenky - předprodej	1 000,00	Příjem	vstupné	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. května 2020		Členské poplatky	1 000,00	Příjem	členské	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. května 2020		dřghxdfcgfgh	200,22	Příjem		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. června 2020		Členské poplatky	1 000,00	Příjem	členské	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Obrázek 6: Položka Záznamy

V levém horním rohu se nachází tlačítko *Odstranit vše*, které uživateli umožňuje vymazat veškeré záznamy najednou, včetně faktur (ty jsou v položce Faktury, viz. dále). Po kliknutí na toto tlačítko se zobrazí varovné okno, zda si uživatel opravdu přeje tuto akci provést, aby nedošlo k nechtěnému smazání všech

údajů. Po potvrzení smazání vyskočí informační hláška s úspěšným odstraněním.

Vedle se nachází tlačítka *Upravit*, *Odstranit* a *Zobrazit*. Ty uživateli umožňují pracovat se záznamy. V případě tlačítka *Zobrazit* a *Upravit* se uživateli zobrazí stejné okno jako u Příjmu (Obr. 5), kde je možné daný záznam upravit, nebo si jej pouze zobrazit. U zobrazení není možné měnit jakkoli zadané hodnoty, jedná se pouze o informační funkci. V případě tlačítka *Upravit* záznamy měnit lze. Odlišnosti se vyskytují pouze u podokna s opakováním, kde nelze měnit již opakované záznamy a uživateli se zobrazí informační upozornění. V případě neopakovaného záznamu je možné změny v opakování provádět. Úpravu je možné provést také po dvojkliku na daný záznam, kdy uživateli opět vyskočí okno s úpravou.

Tlačítko *Odstranit* slouží k odstranění daného záznamu.

Před jakoukoli manipulací se záznamem je nutné jej označit v tabulce.

Neméně významnou funkcí okna je filtrace záznamů. Filtrovat lze podle několika parametrů: od data splatnosti, po datum splatnosti, od určité částky či po určitou částku, dále podle názvu, typu, kategorie a zda je záznam uhrazen či ne. Po kliknutí na tlačítko *Filtrovat* se provede vytřídění záznamů podle zadaných kritérií. Tlačítko *Vyčistit* vrátí vše do původního stavu - smaže zadané parametry ve filtru a vrátí do tabulky veškeré záznamy.

Účelem okna je mít přehledně vypsány veškeré záznamy a také možnost odebrat či upravovat záznamy.

5.8 Faktury

Položka *Faktury* je identická s položkou *Záznamy* s tím rozdílem, že obsahuje pouze fakturované záznamy. V případě, že je fakturovaný záznam uhrazen, objeví se i v položce *Záznamy* a bude v tabulkách obsažen dvakrát.

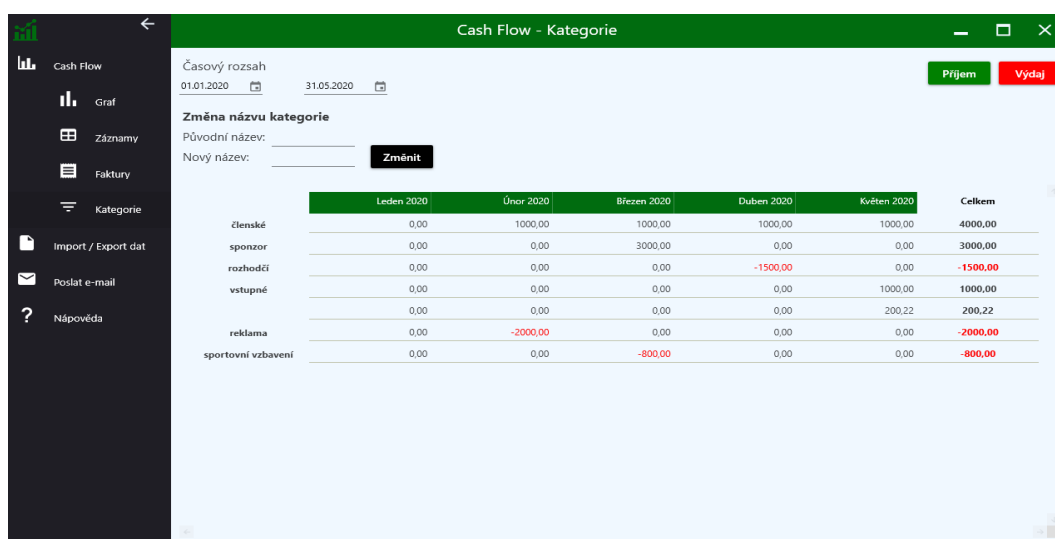
Další odlišností je, že v této položce se nenachází tlačítko *Odstranit vše*, ostatní tlačítka jsou ovšem identická s těmi u záznamů.

5.9 Kategorie

V této položce (Obr. 7) je v levém horním rohu vyobrazen časový rozsah, který je spojen s rozsahem u položky *Graf* a je měnitelný. Účelem této položky je uživateli zanalyzovat a zobrazit kategorie, za které daný měsíc či měsíce nejvíce utratil v přehledné tabulce.

Pod časovým rozsahem se nachází kolonky, které uživateli umožňují změnit název kategorie. Tato změna se projeví ve všech záznamech obsahující původní název a nahradí jej novým.

V tabulce níže jsou vypsány jednotlivé kategorie a ve sloupcích se nachází částky, které uživatel za tyto kategorie v daném měsíci utratil. V posledním sloupci je celková suma za všechny zobrazené měsíce. Záporné hodnoty jsou vypsány alarmující červenou.



Obrázek 7: Kategorie

5.10 Import / Export dat

Aplikace umožňuje pracovat s daty v souborech. Jedná se konkrétně o soubory typu Excel či PDF.

Pro nahrání dat ze souboru typu excel do aplikace slouží tlačítko *Nahrát*. Po jeho kliknutí se uživateli zobrazí okno pro výběr souboru. Nahrávaný soubor typu .xlsx by měl být přesně definován a je určeno, ve kterém sloupci se má jaký parametr nacházet (Obr. 8).

	A	B	C	D	E	F	G	H	I	J	K	L
1	Splatnost rok	Splatnost měsíc	Úhrada rok	Úhrada měsíc	Název	Částka	Typ	Kategorie	Popis	Uhrazeno	Fakturovano	Opakování
2	2020	2	2020	3	Členské poplatky	1000	Příjem	členské		Ano	Ne	0
3	2020	2			Reprezentační ples	2000	Výdaj	reklama		Ne	Ano	Ne
4	2020	3			Členské poplatky	1000	Příjem	členské		Ne	Ne	0
5	2020	3			Nákup materiálu	800	Výdaj	sportovní vybavení		Ne	Ano	Ne
6	2020	3	2020	5	Sponzorský dar	3000	Příjem	sponzor		Ano	Ano	Ne
7	2020	4	2020	4	Rozhodčí	1500	Výdaj	rozhodčí		Ano	Ne	Ne
8	2020	4			Členské poplatky	1000	Příjem	členské		Ne	Ne	0
9	2020	5			Vstupenky - předprodej	1000	Příjem	vstupné		Ne	Ne	Ne
10	2020	5			Členské poplatky	1000	Příjem	členské		Ne	Ne	0
11	2020	5			Turnaj	200,22	Příjem			Ne	Ne	Ne
12	2020	6			Členské poplatky	1000	Příjem	členské		Ne	Ne	0

Obrázek 8: Specifikace nahrávaného souboru typu .xlsx

Speciálním parametrem je opakování, kde je nutno zadat jednu ze 3 možností. V případě, že se záznam neopakuje, je nutno zadat Ne a nebo číslo 0 (v tomto případě se v aplikaci zaškrtně kolonka s opakováním, ale v dalších měsících se kvůli nulové hodnotě neprojeví). V případě, že záznam opakovaný je, uživatel musí vyplnit počet opakování číslovkou -1 pro nikdy nekončící opakování nebo číslovkou větší než 0 pro určitý počet opakování.

Tlačítko *Vytvořit* umožňuje uživateli naopak soubor z dat uložených v aplikaci vytvořit. Takový soubor může být ve formátu .pdf či .xlsx a opět se zobrazí okno, ve kterém si uživatel zvolí, kam chce tvořený soubor uložit.

V průběhu nahrávání či tvoření souboru je uživateli zobrazen červený text s informací, že nahrávání probíhá. Po dokončení procesu se objeví zelený nápis s úspěšným ohlášením, případně s informací že soubor se nahrát / vytvořit nepodařilo.

5.11 Posílání emailu

Jedna z posledních funkcí aplikace je odesílání emailu. Po kliknutí na příslušnou položku se otevře okno, ve kterém uživatel vyplní svoji emailovou adresu a heslo, e-mailovou adresu příjemce, text a je zde možnost vložení přílohy a poslat tak data z aplikace. Omezením funkce je, že e-mailová adresa odesílatele je limitována na @seznam.cz a @gmail.com. Před odesláním e-mailu z @gmail.com se uživateli zobrazí výzva s odkazem, kde je nutno povolit odeslání emailu z aplikací třetích stran.

5.12 Nápověda

V položce nápověda je stručněji shrnuta výše popsaná příručka a slouží k rychlé orientaci uživatele v programu v případě, že by si nebyl jistý kam zadat nové záznamy. Přehledně popisuje funkci jednotlivých položek.

Závěr

Cílem práce bylo vytvořit aplikaci pro analýzu a sledování CashFlow. Aplikace by měla splňovat 3 základní požadavky - možnost zadávat příjmy a výdaje, možnost nahrát data ze souboru, případně do souboru, a v neposlední řadě předpovídat vývoj CashFlow v budoucích měsících.

Výsledkem práce je aplikace splňující zadané požadavky. Hlavní sekce CashFlow je rozdělena na více částí, kde je možné si CashFlow za zvolené časové období zobrazit graficky či tabulkovou formou. Z obou možností uživatel zřetelně vidí, jaká je jeho aktuální finanční situace a při rozšíření časového období uvidí, jak na tom bude v následujících měsících při stejných příjmech a výdajích. Aplikace také umožňuje uživateli dohledat, za jaké kategorie nejvíce utratil a ve kterých měsících. Jednou z hlavních funkcí aplikace je práce se soubory, kde je možné nahrát do aplikace data ze souboru Excel, případně data z aplikace nahrát do souboru Pdf či Excel. Bonusovou funkcí je posílání emailu, kde je možné vygenerované soubory ihned zaslat dalším osobám.

Vzhledem k tomu, že aplikace je cílena pro sportovní kluby, je zde řada doplňujících funkcí, které by mohli rozšířit využití aplikace. Například kalendáře pro přehledné plánování akcí, zápasů či soustředění, a nebo seznam členů klubu s informacemi o poplatecích a jiné. Dalším rozšířením by mohlo být zapojení dalších měn do systému aplikace, jelikož aktuální zpracování je cíleno na českou klientelu.

Conclusions

The aim of this thesis was to create an application for analysis and monitoring CashFlow. The application should fulfil three requirements: to input incomes and issues, to load and save data from file and to forecast the trend in upcoming months.

The work resulted in application satisfying the mentioned requirements. The main section of CashFlow is divided into several parts where it is possible to visualize graphs and tables based on selected time period. From these two options is clearly visible what is user's actual financial situation. With extension of the time period, the user can see the potential development in the upcoming months with the same incomes and issues. The application also allows the user to find categories or months with most spending. One of the main functions of the application is to work with files. It is possible to load data from Excel files into the application and to export data into Pdf or Excel format. The additional function of the application is sending emails with generated files immediately to other people.

The application is designed mainly for sport clubs. It offers many additional functions which could extend the utilization of the application. For example, calendars for upcoming events, matches or sports camps, or even the list of club members and their financial commitments and others. Another expansion could be the inclusion of other currencies into the system which is currently aimed for Czech clients.

A Obsah příloženého CD/DVD

Na samotném konci textu práce je uveden stručný popis obsahu příloženého CD/DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

bin/

Obsahuje instalátor `SETUP.MSI` a adresář `Application`, jehož součástí je program `CASHSPORT`, spustitelný přímo z CD/DVD. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový běh programu z CD/DVD.

doc/

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

src/

Kompletní zdrojové texty programu `CASHSPORT` se všemi potřebnými zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu.

readme.txt

Instrukce pro instalaci a spuštění programu `PROGRAM`, včetně všech požadavků pro jeho bezproblémový provoz.

Navíc CD/DVD obsahuje:

data/

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

Literatura

- [1] Caflou, inteligentní podnikatelská aplikace pro malé firmy a živnostníky | Caflou. *Caflou, inteligentní podnikatelská aplikace pro malé firmy a živnostníky / Caflou* [online]. Copyright © 2019 Caflou [cit. 28.06.2019]. Dostupné z: <https://www.caflou.cz/>
- [2] Magazín o české fintech scéně. [online]. Copyright © 2019 FintechCowboys [cit. 28.06.2019]. Dostupné z: <https://fintechcowboys.cz/nova-aplikace-bankonaut-flow/>
- [3] Affilák.cz - Blog o affiliate marketingu a blogování [online]. Copyright © 2019 [cit. 29.06.2019]. Dostupné z: <https://affilak.cz/clanek-o-bankonaut-flow>
- [4] CashFlow Patrol – Aplikace na Google Play. [online]. Copyright © 2019 Google [cit. 30.06.2019]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.q.cfp\hl=cs>
- [5] Introduction to the C# Language and the .NET Framework | Microsoft Docs. [online]. Copyright © Microsoft 2020 [cit. 09.05.2020]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [6] Programovací jazyk C#. 301 Moved Permanently [online]. Dostupné z: <http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/index.html>
- [7] Co je to WPF? - The complete WPF tutorial. Welcome - The complete WPF tutorial [online]. Copyright © wpf [cit. 09.05.2020]. Dostupné z: <https://www.wpf-tutorial.com/cs/1/o-wpf/co-je-to-wpf/>
- [8] Úvod do WPF | Microsoft Docs. [online]. Copyright © Microsoft 2020 [cit. 09.05.2020]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/framework/wpf/introduction-to-wpf>
- [9] KOSEK, Jiří. *XML pro každého*. Praha: Grada Publishing, 2000, 164 s. ISBN 8071698601.
- [10] XML Syntax. W3Schools Online Web Tutorials [online]. Dostupné z: <https://www.w3schools.com/xml>
- [11] Jazyk XAML. Největší český web zaměřený na .NET framework [online]. Copyright © 2019 [cit. 30.06.2019]. Dostupné z: <https://www.dotnetportal.cz/clanek/198/Jazyk-XAML>
- [12] Přehled XAML - WPF | Microsoft Docs. [online]. Copyright © Microsoft 2020 [cit. 09.05.2020]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/desktop-wpf/fundamentals/xaml>

- [13] Material Design In XAML. Material Design In XAML [online]. Dostupné z: <http://materialdesigninxaml.net/>
- [14] Live Charts. Live Charts [online]. Dostupné z: <https://lvcharts.net/>
- [15] PDFsharp and MigraDoc - Home. PDFsharp [online]. Copyright © 2009 empira Software GmbH [cit. 27.05.2020]. Dostupné z: <http://www.pdfsharp.com/PDFsharp/>