

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra technických studií

Kiosková aplikace

Autor práce: Lukáš Láník

Vedoucí práce: Ing. Ladislav Prokop

Jihlava 2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	Lukáš Láník
Studijní program:	Elektrotechnika a informatika
Obor:	Aplikovaná informatika
Název práce:	Kiosková aplikace
Cíl práce:	Naprogramování desktopové aplikace pro infokiosky do firmy Bosch s.r.o., která bude primárně sloužit k prohlížení a tisku výplatní pásky, dále pak k zobrazování vnitrofiremních portálů a vyhledávání na intranetu. Aplikace bude programována v jazyce C# wpf a interface aplikace bude vytvořen pomocí jazyku XAML. Infokiosky jsou pc s dotykovou obrazovkou. Uživatel se bude do aplikace přihlašovat magnetickou kartou.

Ing. Ladislav Prokop
vedoucí bakalářské práce

doc. Ing. Zdeněk Horák, Ph.D.
vedoucí katedry
Katedra technických studií

Abstrakt

Tato práce pojednává o vytvoření desktopové aplikace na infokiosky pro BOSCH DIESEL s.r.o., která umožňuje prohlížení a tisk výplatních pásek, prohlížení vnitropodnikových portálů a přístup na intranet. Uživatel se do aplikace přihlašuje pomocí magnetické karty, odhlášení je pak možné také pomocí magnetické karty nebo jednoho z odhlašovacích tlačítek v aplikaci. Je zde i funkcionality automatického odhlášení.

Aplikace je vytvořena ve vývojovém prostředí Microsoft Visual Studio (Aplikace WPF, .NET Framework) prostřednictvím jazyka C# a jazyka XAML.

Klíčová slova

.NET; WPF; C#; XAML; infokiosky.

Abstract

The thesis deals with creating a desktop application for information kiosks for BOSCH DIESEL s.r.o., that allows users to view and print payslips, to view in-house portals and to access the intranet. The user can log into the application with a magnetic card, the logout is then possible also with a magnetic card or with one of the logout buttons in the application. There is also a feature of automatic logout.

The application is created in the Microsoft Visual Studio development environment (Application WPF, .NET Framework) using C# and XAML.

Key words

.NET; WPF; C#; XAML; information kiosks.

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ.

Byl/a jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje AZ, zejména
§ 60 (školní dílo).

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že souhlasím s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutí licence.

V Jihlavě dne

.....

Podpis

.....

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce Ing. Ladislavu Prokopovi za poskytnutí tématu a možnost vytvářet ho pod jeho vedením.

Obsah

Poděkování.....	5
Obsah	6
Úvod.....	8
1 Popis .NET frameworku, C# WPF a XAML	10
1.1 Vysvětlení použitých výrazů.....	11
2 Praktická realizace projektu.....	12
2.1 Náhled interface a kód v XAMLu.....	13
2.2 App.xaml.....	14
2.3 Hlavní přihlašovací obrazovka.....	15
2.4 Hlavní nabídka	17
2.5 Admin okno.....	18
2.6 Tisk výplatnice	19
2.7 Auto odhlášení	20
2.8 Ostatní okna	21
3 Kód aplikace v C#.....	22
3.1 Třída Login.....	23
3.2 Třída Load_card_number.....	25
3.3 Třída PayRoll	27
3.4 Třída Extension_methods.....	32
3.5 Třída MainViewModel_Auto_log_out	33
3.6 Třída Auto_log_out_manager	34
3.7 MainWindow kód C#.....	35
3.8 Main_menu_window kód C#.....	38
3.9 Admin_window kód C#.....	39
3.10 Payroll_printing_window kód C#	40
3.11 Auto_log_out kód C#	41
3.12 Ostatní okna kód C#.....	42
Závěr	43
Citace	44
Zdroje.....	Chyba! Záložka není definována.

Seznam obrázků

Obrázek 1: Hlavní přihlašovací obrazovka.....	15
Obrázek 2: Hlavní nabídka	17
Obrázek 3: Admin okno.....	18
Obrázek 4: Tisk výplatnice	19
Obrázek 5: Auto_odhlaseni	20
Obrázek 6: Ostatní okna	21

Seznam ukázek kódu

1 Ukázka kódu XALM	16
2 Ukázka kódu BackgroundWorker	24
3 Ukázka Kódu iTextSharp	31

Úvod

Ve firmě BOSCH DIESEL s.r.o. jsem byl požádán, abych vytvořil aplikaci pro infokiosky, kde si zaměstnanci mohou zobrazit svou výplatní pásku a jiné vnitropodnikové informace. Tuto nabídku jsem přijal s tím, že jí pojmu jako svojí bakalářskou práci.

Tato práce pojednává o tvorbě kioskové aplikace. Kiosková aplikace slouží primárně k zobrazení a vytisknutí výplatní pásky, dále pak k přístupu na různé vnitropodnikové portály.

Kiosková aplikace je vytvořena v jazyce C# WPF, interface pak v jazyce XAML, ve vývojovém prostředí Visual Studio. Vše bude programováno objektově a budu se snažit využívat aktuálních postupů a technologií.

Jelikož předchozí aplikace byla webová, tak si od mé aplikace slibuji hlavně vyšší rychlost.

1 Popis .NET frameworku, C# WPF a XAML

.NET je zdarma mezi platformový framework pro programování různorodých aplikací. Nabízí mnoho programovacích jazyků, knihoven a editorů pro webové stránky, mobilní aplikace, desktopové aplikace, hry a mnoho dalších. Jeden z podporovaných jazyků je C# WPF, kterým je naprogramován program jímž se zabývá tato bakalářská práce. (Microsoft)

Framework Windows Presentation Foundation, dále jen WPF, vzniknul jako nástupce zastaralejšího Windows Form, dále jen WF. Kvůli podpoře jazyka XAML pro tvorbu uživatelského prostředí, se stal WPF sice o něco komplexnější, ale mnohem kvalitnější prezentační vrstvy a navíc podporuje tzv. Bindingů, které dovolují propojit objekty a prvky formuláře. (David Čápka, 2020)

C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft. C# vzniknul na předloze jazyků C a Java, u kterých jsou vidět výrazné podobnosti s C#. (2012 Pro WPF 4.5 in C#)

C# WPF je součástí .Net frameworku od verze 3.0. Slouží k vytváření komplexních aplikací a disponuje velkou nabídkou formulářových prvků a širokou paletou stylizování. (David Čápka, 2020)

Mezi výhody C# WPF také patří nezávislost na rozlišení zobrazovacího zařízení, neboť WPF používá jednotky nezávislé na zařízení. Jednotka je definována jako 1/96 palce. (2012 Pro WPF 4.5 in C#)

XAML je Xtensible Application Markup Language vychází ze základů XML a stejně jako XML je to značkovací jazyk, mezi jehož široké využití patří tvorba grafického rozhraní pro WPF aplikace. XAML není na WPF závislý a tudíž velice dobře slouží k oddělení grafického rozhraní a logiky programu. (Sells Chris a Ian Griffiths 2005 Programming WPF (2nd Edition))

1.1 Vysvětlení použitých výrazů

Proměnná – uchovává data při běhu programu. Reprezentuje část paměti, do které můžeme vkládat, upravovat a mazat data.

Funkce – množina příkazů sjednocených tak, že je lze provést zavoláním jedné určité funkce. Funkce může, ale nemusí, mít návratovou hodnotu.

Třída – Objekt obsahující vlastní proměnné a funkce (zde se nazývají metody). Proměnné a metody mohou být tzv. *private* a *public*. *Private* proměnné a metody se používají vnitřně a nic kromě třídy, kde jsou vytvořené k nim nemůže přistupovat. *Public* proměnné a metody (proměnné většinou nebývají *public*) jsou dostupné z vnitřku i vnějšku třídy.

Objektově orientované programování – styl programování, které se snaží program uzpůsobit tak, aby kód byl bližší lidskému chápání a také napomáhá k větší přehlednosti kódu.

Vlákno – paralelně vytvořený proces který běží současně s hlavním procesem. Hlavní proces většinou bývá hlavní program a vedlejší proces je program který pracuje souběžně s hlavním procesem, ale nechceme, aby se na něj v hlavním programu čekalo.

2 Praktická realizace projektu

Tato část se zabývá praktickou realizací kioskové aplikace. Je zde popis kódu, náhled interface, vysvětlení použitých metod, problémy, na které jsem narazil a způsoby jak jsem je vyřešil.

Pro programování této aplikace jsem čerpal těchto zdrojů:

- Itnetwork: C# objektové programování (David Čápka)
- Itnetwork: Okenní aplikace v C# .NET WPF (David Čápka)
- Itnetwork: Práce se soubory v C# (David Čápka)
- Diskusní fórum Stack Overflow
- Dzone: c best practices (Faisal Pathan)
- Codeproject: Some Best Practices for C Application Development (Kunal Chowdhury)
- Microsoft: C# WPF documentation
- Tutorialspoint: WPF tutorial

2.1 Náhled interface a kód v XAMLu

Zde je popis kódu v XAMLu kterým je tvořena uživatelské prostředí aplikace a ukázka jednotlivých oken a kódu v XAMLu. XAML je značkovací jazyk podobný HTML.

V XAMLu se dají dělat rozličné prostředí za pomoci velké palety nástrojů, stylů objektů, bindingů a mnoho dalších. Já jsem použil hlavně nástroje pro tvorbu uživatelského prostředí a tvorbu stylů.

2.2 App.xaml

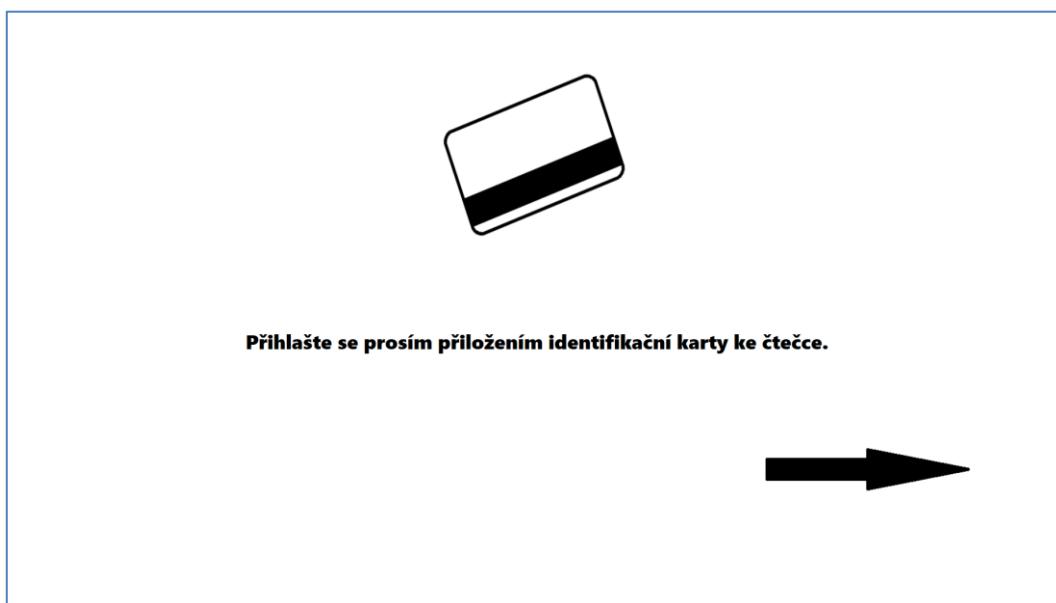
Součástí projektu, která uchovává mnou vytvořené styly, ne nepodobné kaskádovým stylům v HTML, pro objekty v grafickém návrhu aplikace. Styly jsou dobrý a jednoduchý způsob jak vylepšit vzhled své aplikace.

App.xaml obsahuje dva styly pro tlačítko a jeden styl pro groupbox. Styl „Zakl_button_style“ je pro všechna tlačítka kromě odhlašovacích, pro odhlašovací tlačítka se používá styl „Zakl_button_style_red“. Styl pro groupbox „Zakl_Groupbox_style“ upravuje groupboxy v Admin okně, které slouží pro sdružení labelů a progress baru zvlášť pro papír a zvlášť pro toner.

2.3 Hlavní přihlašovací obrazovka

Přihlašovací okno, slouží pouze jako vstupní okno do aplikace. Slouží pouze jako vstupní obrazovka pro nepřihlášeného uživatele. Po přiložení magnetické karty spojené s aktivním uživatelem, následuje přesměrování do hlavní nabídky.

Hlavní přihlašovací obrazovka se skládá z dvou obrázků a jednoho labelu.



Obrázek 1: Hlavní přihlašovací obrazovka

Ukázka kódu v XAMLu:

```
<Grid>

    <Grid.RowDefinitions>
        <RowDefinition Height="69*" />
        <RowDefinition Height="372*" />
        <RowDefinition Height="300*" />
        <RowDefinition Height="168*" />
        <RowDefinition Height="163*" />
    </Grid.RowDefinitions>

    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="367*" />
        <ColumnDefinition Width="225*" />
        <ColumnDefinition Width="368*" />
    </Grid.ColumnDefinitions>

    <Image x:Name="Obr_karta" Source="C:\obr\karta.bmp"
    Stretch="Fill" Grid.Column="1" HorizontalAlignment="Center"
    VerticalAlignment="Center" Grid.Row="1" />

    <Label x:Name="Prilozte_kartu_label" Content="Přihlašte se prosím
    přiložením identifikační karty ke čtečce." Margin="0"
    FontSize="36" FontFamily="Segoe UI Black" Grid.ColumnSpan="3" Grid.Row="2"
    HorizontalAlignment="Center" VerticalAlignment="Center" />

    <Image HorizontalAlignment="Center" Margin="0" VerticalAlignment="Center"
    Source="C:\obr\sipka.bmp" Grid.Column="2" Grid.Row="3" />

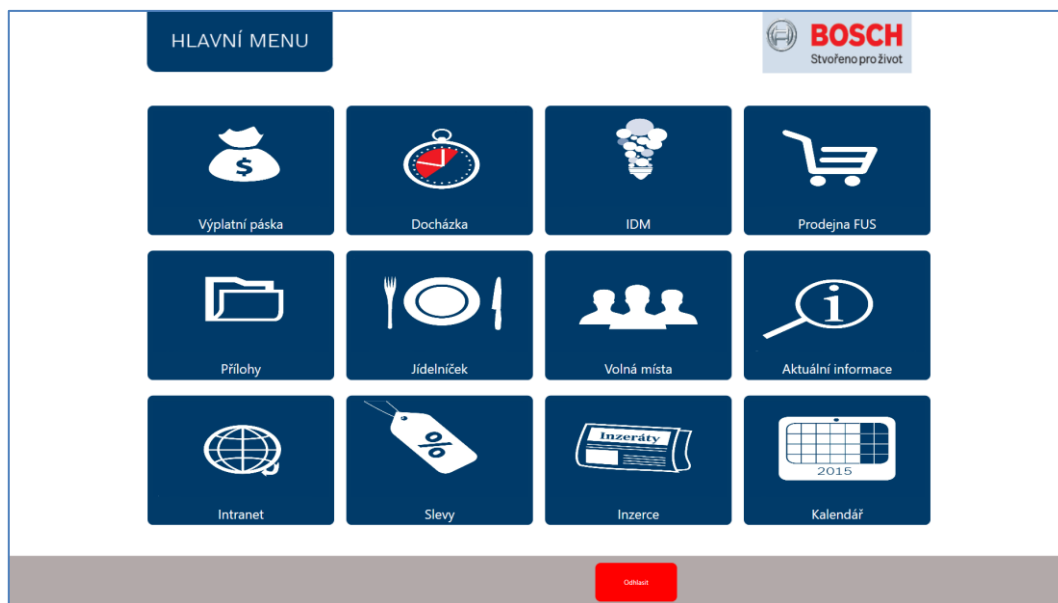
</Grid>
```

1 Ukázka kódu XALM

2.4 Hlavní nabídka

Matice tlačítek sloužící pro snadnou navigaci v nabídce aplikace. Po zmáčknutí každého tlačítka, je uživatel převeden na konkrétní službu, nebo v případě tlačítka pro odhlášení odhlášen.

Každé tlačítko v hlavní nabídce obsahuje pro přehlednější orientaci obrázek reprezentující službu spojenou s tlačítkem.



Obrázek 2: Hlavní nabídka

2.5 Admin okno

Admin okno, které se zobrazí pro uživatele, kteří mají admin práva. Slouží k doplnění počítačů pro papír a toner, tyto hodnoty jsou ukládány do XML souboru. Po zmáčknutí tlačítek doplnit se nastaví hodnota papír, nebo toner na maximum, při každém vytisknutí dokumentu se zase určitá hodnota zase odečte. Tlačítko „Pokračovat“ vede do hlavní nabídky.

The screenshot displays the 'Admin okno' interface. It features two main sections for resource management: 'Papír' (Paper) and 'Toner'. Each section includes a green progress bar representing the current level, with 'Stav:' (Status) and 'Max:' (Maximum) labels. To the right of each section is a 'Doplnit' (Refill) button. Below these sections is a 'Pokračovat' (Continue) button.

Resource	Current Status (Stav)	Maximum (Max)	Action Button
Papír	200	200	Doplnit
Toner	100	100%	Doplnit

Below the 'Doplnit' buttons is a 'Pokračovat' button.

Obrázek 3: Admin okno

2.6 Tisk výplatnice

Okno pro procházení výplatních pásek přihlášeného uživatele. Je zde možnost zobrazit si aktuální výplatnici, s možností zobrazení výplatnic i z minulého roku, a tisk výplatnice. Tlačítka „následující měsíc“ a „předchozí měsíc“ slouží pro přechod mezi výplatními páskami. Pokud je zobrazena aktuální, tedy nejnovější, výplatnice je tlačítko „následující měsíc“ neaktivní, pro poslední dostupnou výplatnici je neaktivní tlačítko „Předchozí měsíc“. Po zmáčknutí tlačítka „tisk“ se dokument zobrazený v objektu webbrowser vytiskne. Tlačítko „Home“ převede uživatele do hlavní nabídky a tlačítko „Odhlásit“ uživatele odhlásí.

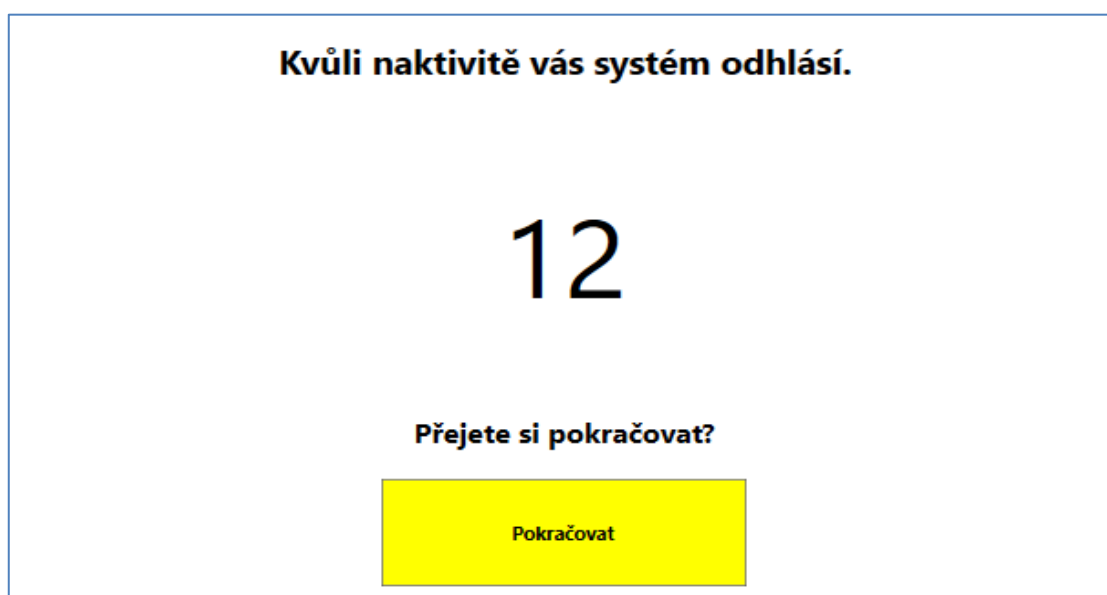


Obrázek 4: Tisk výplatnice

2.7 Auto odhlášení

Okno pro signalizaci automatického odhlášení. Pokud aplikace zaznamená neaktivitu trvající déle jak 30 sekund, spustí se okno s automatickým odhlášením.

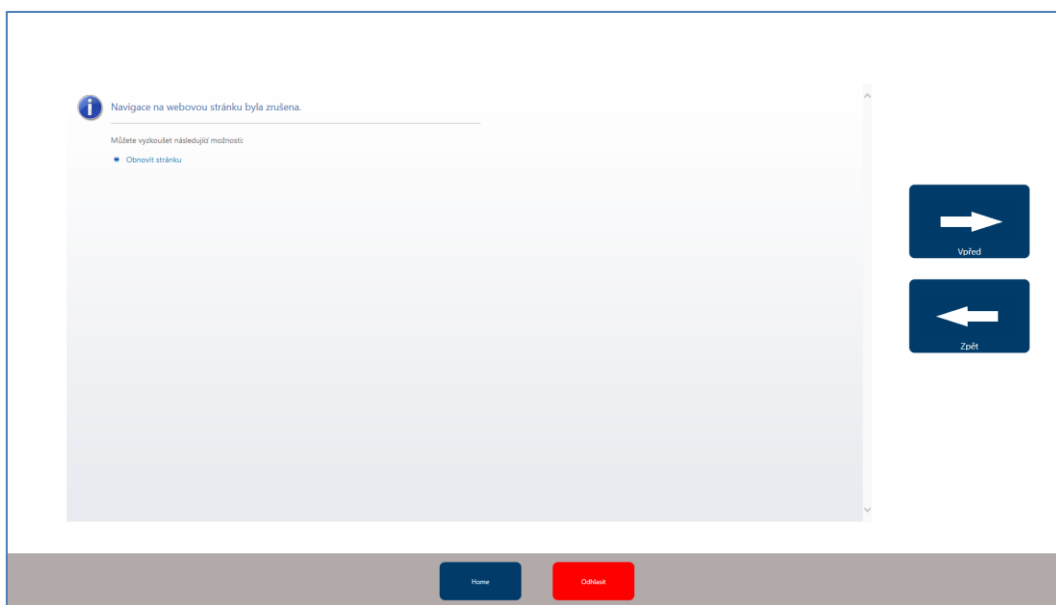
Po vypršení časového intervalu, zde 12 sekund, program uživatele odhlásí. Automatické odhlášení lze zrušit stisknutím tlačítka „Pokračovat“.



Obrázek 5: Auto_odhlaseni

2.8 Ostatní okna

Vzhled a kód v XAMLu ostatních oken je velice podobný, liší se pouze tím, na jakou adresu je odkazován prvek „WebBrowser“. Tlačítka „Zpět“ a „Vpřed“ slouží k pohybu na předchozí, nebo následující zobrazený obsah ve WebBrowseru, podobně jako ve webovém prohlížeči. Tlačítko „Home“ převede uživatele do hlavní nabídky a tlačítko „Odhlásit“ uživatele odhlásí.



Obrázek 6: Ostatní okna

3 Kód aplikace v C#

Tato část se zabývá popisem funkční vrstvy Kioskové aplikace. Popisem tříd, metod, funkcí, problémů, na které jsem v průběhu tvorby aplikace narazil a řešení, které jsem pro odstranění těchto problémů použil.

3.1 Třída Login

Tato třída se zabaluje celý proces přihlášení pomocí magnetické karty. Obsahuje možnosti začít číst ze sériového portu číslo karty, což vykonává třída NacteniCislaKarty, a podle tohoto čísla načíst osobní číslo osoby, vrátit číslo osoby, vrátit stav přihlášení (přihlášen/nepřihlášen), vymazat aktuální přihlášení a rozhodnout, zda se jedná o admin účet, nebo ne.

Proces načtení magnetické karty a následné přihlášení, či odhlášení, bylo nutné přiřadit do samostatného vlákna. Pro snadnější a logičtější manipulaci s vlákny jsem použil BackgroundWorker, což je třída pro práci s vlákny.

Popis proměnných, metod a funkcí:

Proměnná „card_number“ typu string uchovává číslo načtené karty, podle nějž se vyhledá osobní číslo uživatele.

Proměnná „logged_in“ typu bool slouží k uchování hodnoty, jestli někdo je, nebo není přihlášen.

Metoda „Admin_account“ slouží k rozhodnutí, zda přihlášený uživatel má admin práva, či nikoli.

Metoda „Load_card_number“ provádí proces načtení čísla karty, za použití metod z třídy „Load_card_number“.

Metoda „Load_personal_number“ se stará o načtení osobního čísla, podle čísla načtené karty.

Metoda „Clear“ nastaví proměnné „osobní_cislo“ a „číslo_karty“ na prázdný řetězec.

Ukázka BackgroundWorkeru

```
private BackgroundWorker bw;
bw = new BackgroundWorker { WorkerSupportsCancellation = true };
bw.DoWork += bw_DoWork;
bw.RunWorkerCompleted += bw_RunWorkerCompleted;
bw.RunWorkerAsync();

private void bw_DoWork(object sender, DoWorkEventArgs e)
{
    while (true)
    {
        login.Load_personal_number();
        if (login.Personal_number != "")
        {
            if (!login.Logged_in)
            {
                if (login.Admin_account())
                {
                    e.Result = false;
                    return;
                }
                else
                {
                    e.Result = true;
                    return;
                }
            }
        }
        login.Clear();
    }
}

private void bw_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
    if (!Boolean.Parse(e.Result.ToString())) Start_admin_win();
    else Start_Main_menu_win();
}
```

2 Ukázka kódu BackgroundWorker

3.2 Třída Load_card_number

Tato třída provádí práci se sériovým portem a načtení čísla karty. Po přiložení magnetické karty k čtečce se načte kód, který reprezentuje číslo načtené karty. Tento kód je načítán po jednotlivých bajtech.

S načtením magnetické karty pomocí čtečky jsem se potýkal s problémem, jak číst ze sériového portu. Přes mé snahy se mi stále nedařilo data ze sériového portu načíst. Po měsíci hledání a zkoušení různých způsobů jsem se stále nedomáhal úspěchu. Po konzultaci se svým vedoucím práce Ladislavem Prokopem jsme našli řešení. Jak se pak ukázalo, bylo velmi prosté, k čtečce karet bylo nutné nainstalovat příslušné ovladače, tím se vše vyřešilo.

Z informací které jsem dostal, bylo zřejmé, že kód karty je v hexadecimálním tvaru. Ze sériového portu se ale po jednotlivých bajtech načítaly dvouciferná čísla v desítkovém tvaru. Díky zkušenostem, které jsem nabyl ve škole, jsem dokázal přijít na to, že se jedná o ASCII kód hexadecimálního čísla.

Tento ASCII kód hexadecimálního čísla, je tedy nutné převést na hexadecimální řetězec a tento pak na decimální řetězec, protože k načtení osobního čísla je nutný právě tento tvar. Tento decimální řetězec je již číslo karty, které je hlavní a jedinou návratovou hodnotou této třídy.

Popis proměnných, metod a funkcí:

Poměnná txtScanner_hex je typu sting a obsahuje hexadecimální hodnotu čísla magnetické karty.

Poměnná txtScanner_dec je typu sting a obsahuje decimální hodnotu čísla magnetické karty.

Proměnná buff typu int obsahuje číselné vyjádření velikosti čísla karty v Bajtech.

Objekt SerialPort1 softwarově nasimulovaný sériový port (který je fyzicky USB portem, sériový port je simulován ovladači čtečky magnetické karty)

Metoda `Clear()` vymaže obsah proměnných `txtScanner_hex` a `txtScanner_dec`.

Metoda `Get_port_state()` vrací stav sériového portu, přesněji jestli je, nebo není otevřený.

Metoda `Get_txtScan_dec()` vrací decimální hodnotu načtené magnetické karty.

Metoda `ScannerRead()` zkontroluje stav portu, pokud je zavřený, tak ho otevře a pomocí metody „`Read`“ načte data na sériovém portu.

Metoda `Read` načítá data ze sériového portu v podobě ASCII kódu hexadecimální hodnoty čísla karty, pomocí metody „`Convert_Ascii_to_hex`“ převádí tyto data na hexadecimální hodnotu a pomocí metody „`Convert_hex_to_dec`“ pak tuto hodnotu převádí na decimální hodnotu, která je uložena do proměnné „`txtScanner_dec`“ a nakonec uzavře sériový port.

Metoda `Convert_Ascii_to_hex` převádí hodnotu ASCII znaku kódu hexadecimálního čísla, které se předává parametrem, na hexadecimální číslo.

Metoda `Convert_hex_to_dec` převádí hexadecimální číslo, které se předává parametrem, na decimální.

3.3 Třída PayRoll

Tato třída je nejobsáhlejší a stará se o sestavení cesty k výplatnici, zobrazení výplatnice, tisk výplatnice a zápis stavu papíru a toneru do souboru.

V konstruktoru, po nastavení předaných objektů, se hledá výplatnice z ledna minulého roku, pokud je nalezena, nastaví se jako nejstarší výplatnice, pokud taková výplatnice není nalezena, postupuje se po měsíci k aktuálnímu datu, dokud se nějaká výplatnice nenajde. Pokud se žádná výplatnice nenajde, vypíše se o tom při pokusu o zobrazení výplatnice chybová hláška.

Sestavení cesty k výplatnici je skládání základní cesty s aktuálním měsícem, rokem a osobním číslem osoby.

Zobrazení výplatnice se provádí při otevření okna Payroll_printing_window a zobrazuje se vždy nejaktuálnější nalezená výplatnice, pokud se žádná nenajde, vypíše se hláška o nenalezení.

Jelikož je výplatnice ve formátu PDF a je zaheslovaná, použil jsem pro snadnější práci a odheslování tohoto formátu externí knihovnu iTextSharp.

Popis proměnných, metod a funkcí:

Proměnná `personal_number` typu `string` uchovává osobní číslo právě přihlášeného uživatele.

Proměnná `path_temp_pdf_decrypted` typu `string` obsahuje cestu ke složce kde se nachází dočasně od heslované výplatnice.

Proměnná `path_pass` typu `string` obsahuje cestu k zakódovanému souboru s hesly k výplatnicím.

Proměnná `path_base` typu `string` obsahuje neměnnou část cesty ke konkrétní výplatnici.

Proměnná `path` typu `string` obsahuje již celou cestu ke konkrétní výplatnici.

Proměnná `month` typu `int` obsahuje číslo měsíce zobrazované výplatnice.

Proměnná `month_current` typu `int` obsahuje číslo aktuálního měsíce, získána za pomoci „`DateTime.Now.Month`“.

Proměnná `year_current` typu `int` obsahuje číslo aktuálního roku, získána za pomoci „`DateTime.Now.Year`“.

Proměnná `year` typu `int` obsahuje číslo roku zobrazované výplatnice.

Proměnná `amount_paper` typu `int` obsahuje počet papírů v tiskárně.

Proměnná `amount_toner` typu `int` obsahuje číslo, které reprezentuje stav toneru (200 plný, 0 prázdný).

Proměnná `min_month` typu `int` obsahuje číslo měsíce nejstarší zobrazitelné výplatnice.

Proměnná `min_year` typu `int` obsahuje číslo roku nejstarší zobrazitelné výplatnice.

Proměnná `counter` typu `int` je pomocná proměnná pro metodu „`provereni_cesty(bool zkuska)`“

Proměnná `next_month_picture` typu `Image` obsahuje obrázek, který je na tlačítku pro zobrazení výplatnice z následujícího měsíce.

Proměnná `prev_month_picture` typu `Image` obsahuje obrázek, který je na tlačítku pro zobrazení výplatnice z předchozího měsíce.

Proměnná `button_previous` typu `Button` obsahuje referenci na tlačítko v okně `Payroll_printing_window` pro zobrazení výplatnice z předchozího měsíce.

Proměnná `button_next` typu `Button` obsahuje referenci na tlačítko v okně `Payroll_printing_window` pro zobrazení výplatnice z následujícího měsíce.

Proměnná `pdf_viewer` typu `WebBrowser` obsahuje referenci na objekt `WebBrowser` v okně `Payroll_printing_window`, ve kterém se zobrazují výplatnice.

Objekt `login` třídy `Login` obsahuje aktuální přihlášení.

Konstruktor PayRoll nastaví všechny potřebné proměnné, zjistí rok a měsíc nejstarší zobrazitelné výplatnice a zablokuje tlačítko pro zobrazení výplatnice z následujícího měsíce.

Metoda Encode_pass provádí zakódování souboru s hesly k výplatnicím.

Metoda Decode_pass provádí odkódování souboru s hesly k výplatnicím.

Metoda Load_pass_to_pdf načte heslo k výplatnici ze souboru s hesly.

Metoda Show_pdf zobrazí výplatnici do objektu WebBrowser na okně Payroll_printing_window.

Metoda Verification_path prověřuje, zda existuje alespoň jedna zobrazitelná výplatnice u aktuálně přihlášeného uživatele, pokud ano vrátí se true, pokud ne vrátí se false a vypíše se hláška o nenalezení výplatnice.

Metoda Build_path sestaví kompletní cestu k výplatnici z konkrétního roku a měsíce. Pokud tato soubor z takto sestavené cesty neexistuje vrátí tato metoda false, v opačném případě true.

Metoda Previous_month zobrazí do objektu WebBrowser na okně Payroll_printing_window výplatnici z předchozího měsíce. Pokud neexistuje žádná starší zobrazitelná výplatnice zablokuje tlačítko „button_previous“ pro zobrazení výplatnice z předchozího měsíce.

Metoda Next_month zobrazí do objektu WebBrowser na okně Payroll_printing_window výplatnici z následujícího měsíce. Pokud neexistuje žádná novější výplatnice zablokuje tlačítko „button_next“ pro zobrazení výplatnice z následujícího měsíce.

Metoda Print_payroll vytiskne právě zobrazenou výplatnici.

Metoda Write_paper_toner zapíše stav papíru a toneru do souboru.

Metoda Load_paper_toner načte hodnotu papíru a toneru ze souboru.

Metoda Enable_disable_button zpřístupní/zablokuje určité tlačítko a změní obrázek na tomto tlačítku podle zadaných parametrů.

Metoda Fill:paper doplní hodnotu papíru na maximum.

Metoda Fill_toner doplní hodnotu toneru na maximum.

Metoda Delete_pdfViewer vymaže obsah objektu WebViewer.

Metoda Delete_temp_pdf vymaže obsah složky s dočasně od heslovanými výplatnicemi.

Metoda Decrypt_Pdf_ItextSharp odhesluje výplatnici pomocí rozšiřující knihovny iTextSharp a odheslovanou verzi uloží do složky s odheslovanými výplatnicemi.

Ukázka odheslování PDF

```
private void Decrypt_Pdf_ItextSharp(string input, string output)
{
    string password = Load_pass_to_pdf();
    try
    {
        PdfReader reader = new PdfReader(input, new
        ASCIIEncoding().GetBytes(password));

        using (MemoryStream memoryStream = new MemoryStream())
        {
            PdfStamper stamper = new PdfStamper(reader,
            memoryStream);
            stamper.Close();
            reader.Close();
            File.WriteAllBytes(output, memoryStream.ToArray());
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Chyba odheslování PDF:" + ex.Message, "Výplatnice",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

3 Ukázka Kódu iTextSharp

3.4 Třída `Extension_methods`

Malá třída s prázdným delegátem, pro smazání obsahu WebBrowseru.

3.5 Třída MainViewModel_Auto_log_out

Třída, která se stará o chod okna Auto_odhlaseni.

Tato třída dědí z INotifyPropertyChanged, to má za následek, že každá změna vyvolá reakci. V tomto případě při každém tiknutí DispatcherTimeru se změní hodnota počtu sekund v okně Auto_odhlaseni.

Popis proměnných, metod a funkcí:

Objekt timer typu DispatcherTimer se stará o časový odpočet.

Proměnná Zbyva obsahuje hodnotu odpočtu.

Konstruktor MainViewModel() inicializuje objekt timer, nastaví mu interval na jednu sekundu a odstartuje časovač.

Metoda Timer_Tick je spjatá s objektem timer a vykonává se každou sekundu.

Metoda stop_timer zastaví odpočet.

Metoda OnPropertyChanged vyvolá akci při každé změně proměnné Zbyva

3.6 Třída `Auto_log_out_manager`

Tato třída se stará o posouzení neaktivity uživatele a spuštění automatického odhlášení. Pokud po dobu, která je nastavena proměnnou `time`, neprovede nějaká akce, je spuštěno automatické odhlášení.

Popis proměnných, metod a funkcí:

Proměnná `time` typu `int` obsahuje hodnotu času, po jehož uplynutí, pokud se nepovede žádná akce, se zahájí automatické odhlášení.

Proměnná `timer` je objekt třídy `DisoatcherTimer`, tento objekt obsahuje odpočet automatického odhlášení.

3.7 Třída Load config

Tato třída se stará o načtení konfiguračních dat ze souboru config.con, většinou se jedná o cesty k souborům a nastavení čísla COMu sériového portu. V případě chybného načtení nebo neexistence souboru se stávající konfigurační soubor smaže a vytvoří se nový s výchozími hodnotami.

Popis proměnných, metod a funkcí:

Proměnná path_kiosks typu string obsahuje cestu ke složce kiosks.

Proměnná path_temp_pdf_decrypted typu string obsahuje cestu k složce kde se nachází odheslované výplatnice.

Proměnná path_pass typu string obsahuje cestu k souboru s hesly.

Proměnná path_base typu string obsahuje základ cesty k výplatnicím.

Proměnná path_card_number_persoal_number typu string obsahuje cestu k souboru s čísly karet a osobními čísli.

Proměnná path_admin_users typu string obsahuje cestu k souboru se seznamem uživatelů s administrativními právy.

Proměnná com typu string obsahuje název COM portu.

Load_config je konstruktor této třídy

Metoda Write_conf provede vytvoření konfiguračního souboru s výchozími daty, pokud konfigurační soubor neexistuje, nebo pokud se ho nepovedlo přepsat.

Metoda Load_conf načte data z konfiguračního souboru a zapíše je do proměnných.

Metody Get_+název proměnných vrací tu určitou proměnnou.

3.8 Třída Thread_manager

Třída pro správu vytvořených vláken. Všechna vytvořená vlákna se ukládají do listu typu BackgroundWorker nebo DispatcherTimer. Po odhlášení uživatele tato třída slouží k vypnutí všech dosud spuštěných vláken. Tato třída je statická a tudíž je dostupná odkudkoli.

Metoda Add_bw slouží k přidání objektu typu BackgroundWorker do listu tohoto typu.

Metoda Add_timer slouží k přidání objektu typu DispatcherTimer do listu tohoto typu.

Metoda Kill_all slouží k vypnutí všech dosud běžících vláken.

3.9 MainWindow kód C#

Hlavní okno zobrazené po spuštění aplikace, na pozadí, ve vlastním vlákne, zde běží cyklus pro načítání magnetické karty. Pro práci s vlákny jsem zvolil BackgroundWorker, díky kterému, podle mého názoru, je práce s vlákny jednodušší a logičtější.

Popis proměnných, metod a funkcí:

Proměnná pay_roll je objekt třídy PayRoll, která pracuje s výplatnicemi.

Proměnná login je objekt třídy Login, která se stará o přihlašování a odhlašování.

Proměnná bw je objekt třídy BackgroundWorker, která slouží k práci s vlákny.

MainWindow je bezparametrický konstruktor.

Metoda bw_DoWork obsahuje vykonávané akce na samostatném vlákne, v tomto případě je to čekání na přihlášení magnetickou kartou.

Metoda bw_RunWorkerCompleted se vykoná po skončení metody bw_DoWork, v tomto případě uživatele přesměruje na admin okno, nebo do hlavní nabídky, podle oprávnění uživatele.

Metoda Start_admin_win přesměruje na admin okno.

Metoda Start_Main_menu_win přesměruje na hlavní nabídku.

Metoda Test_user_button_Click a Test_admin_button_Click jsou pro testovací účely, slouží pro přístup k admin oknu, nebo k hlavní nabídce, bez nutnosti přihlášení magnetickou kartou.

3.10 Main_menu_window kód C#

Zde se nachází hlavně funkce pro spuštění určitého okna po kliknutí na dané tlačítko. Je zde možnost odhlášení pomocí magnetické karty, pokud je přiložena stejná karta, pomocí které bylo provedeno přihlášení.

Jsou zde využívány třídy Login a Auto_log_out_manager. Třída Login je zde využívána na odhlášení pomocí magnetické karty a třída Auto_log_out_manager na spuštění automatického odhlášení při neaktivitě.

Popis proměnných, metod a funkcí:

Proměnná login a login_logout je objekt třídy Prihlaseni, která se stará o přihlašování a odhlašování.

Proměnná bw je objekt třídy BackgroundWorker, která slouží k práci s vlákny.

Statická proměnná first_start typu bool slouží pro kontrolu, zda je toto okno otevřeno poprvé, protože při znovuotevření tohoto okna v rámci jednoho přihlášení se již znovu nespouští další vlákno pro odhlášení.

Statická proměnná Auto_odhlaseni_spusteno typu bool je pro zjištění, jestli již je, či není spuštěné autoodhlášení.

Metoda Nabídka_okno je konstruktor.

Metody x_button_Click slouží k přesměrování na „x“ okno.

Metoda bw_RunWorkerCompleted se vykoná po skončení metody bw_DoWork, v tomto případě uživatele odhlásí, zavře všechna okna a zobrazí přihlašovací obrazovku..

3.11 Admin_window kód C#

Do tohoto okna se dostanou jen uživatelé s admin právy. Je zde možnost zadat doplnění papíru, nebo toneru. Po zmáčknutí tlačítka pokračovat je uživatel přesměrován na hlavní nabídku.

Popis proměnných, metod a funkcí:

Proměnná `pay_roll` je objekt třídy `PayRoll`, která pracuje s výplatnicemi, v tomto případě se používá k zápisu stavu papíru a toneru.

Proměnná `login` je objekt třídy `Login`, která se stará o přihlašování a odhlašování.

Proměnná `paper_toner` je pole typu `double` a slouží k načtení hodnot papíru a toneru z XML souboru.

Proměnná `paper_progress.Value` reprezentuje hodnotu `ProgressBaru` pro papír.

Proměnná `toner_progress.Value` reprezentuje hodnotu `ProgressBaru` pro toner.

Proměnná `state_paper_label.Content` reprezentuje hodnotu `Labelu` pro papír.

Proměnná `state_toner_label.Content` reprezentuje hodnotu `Labelu` pro toner.

Funkce `Fill_paper_button_Click` se provede po zmáčknutí tlačítka „Doplnit_papir_button“ a doplní hodnotu `ProgressBaru` a `Labelu` pro papír na maximum.

Funkce `Fill_toner_button_Click` se provede po zmáčknutí tlačítka „Doplnit_toner_button“ a doplní hodnotu `ProgressBaru` a `Labelu` pro toner na maximum.

Funkce `Continue_button_Click` se provede po zmáčknutí tlačítka „Continue_button“ a převede uživatele na `Main_menu_window`

3.12 Payroll_printing_window kód C#

Zde se provádí zobrazené výplatní pásy, jsou zde funkce pro stisknutí tlačítek a v konstruktoru je zobrazena výplatnice do prvku WebBrowser (zde PDF_viewer_webbr). Je zde předáno aktuální přihlášení.

Popis proměnných, metod a funkcí:

Proměnná pay_roll je objekt třídy PayRoll, která pracuje s výplatnicemi.

Proměnná login je objekt třídy Login, která se stará o přihlašování a odhlašování.

Metoda Print_button_Click vytiskne výplatnici.

Metoda Next_month_button_Click zobrazí výplatnici z následujícího měsíce.

Metoda Previous_month_button_Click zobrazí výplatnici z předchozího měsíce.

Metoda Log_out_Button_Click odhlásí uživatele, zavře všechna okna a zobrazí přihlašovací okno.

Metoda Home_button_Click přesměruje uživatele do hlavní nabídky.

3.13 Auto_log_out kód C#

Okno s výstrahou o zahájení procesu automatického odhlášení. Uživatel má možnost tento proces zrušit a pokračovat v práci. Pokud časovač doběhne, dojde k odhlášení aktuálního uživatele.

Popis proměnných, metod a funkcí:

Proměnná timer je objekt třídy DispatcherTimer, tento objekt obsahuje odpočet automatického odhlášení.

Metoda Auto_log_out je konstruktor.

Metoda Continue_button_Click resetuje odpočet a zavře okno s auto odhlášením.

3.14 Ostatní okna kód C#

U ostatních oken je kód velice podobný, liší se pouze v odkazu na obsah zobrazovaný ve WebBrowseru. Tlačítko „Home“ přesměruje uživatele do hlavní nabídky, tlačítko „Odhlásit“ odhlásí uživatele a tlačítka zpět a vpřed, označená šipkami, slouží k navigaci ve WebBrowseru, podobně jako v internetových prohlížečích.

Popis proměnných, metod a funkcí:

Proměnná login je objekt třídy login, která se stará o přihlašování a odhlašování.

Funkce Home_button_Click se spustí po kliknutí na tlačítko „Home“ a přesměruje uživatele do hlavní nabídky.

Funkce Log_out_Click se spustí po kliknutí na tlačítko „Odhlásit“ a provede odhlášení uživatele.

Funkce update_button slouží k rozhodnutí, zda zpřístupnit, či zablokovat tlačítka zpět a vpřed, podle toho, je-li možné přistupovat k předešlé nebo následující stránce.

Funkce Forward_button_Click se spustí po kliknutí na tlačítko vpřed a přesměruje obsah WebBrowseru na následující obsah, pokud je to možné.

Funkce Back_button_Click se spustí po kliknutí na tlačítko zpět a přesměruje obsah WebBrowseru na předchozí obsah, pokud je to možné.

Závěr

Práce na kioskové aplikaci mě vesměs bavila a byla to zajímavá zkušenost. Byli doby kdy mě to nebavilo, ba naopak skoro štvalo, každý programátor mi jistě dá zapravdu, že není nic horšího, než když vám v programu něco nefunguje, nebo se program nechová tak, jak byste čekali a vy nevíte proč.

Kiosková aplikace, tak jak jsem jí vytvořil, má všechny funkcionality, které mít měla. Navíc oproti plánu jsem přidal ještě automatické odhlašování. Vše je naprogramováno dle principů objektového programování, kód je podle mého názoru přehledný a dobře čitelný. Určitě by se tam našli místa pro vylepšení, ale osobně si myslím, že kvalita, co se týče stylu naprogramování, je dosti slušná.

Kiosková aplikace byla otestována v reálném prostředí a byla tak potvrzena její funkčnost a je tedy možné ji nasadit.

Citace

ČÁPKA, David. <https://www.itnetwork.cz/csharp/formulare/wpf/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace/>. *ITnetwork* [online]. Praha [cit. 2020-01-03]. Dostupné z: <https://www.itnetwork.cz/csharp/formulare/wpf/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace/>

What is .NET? An open-source developer platform.. .NET | Free. Cross-platform. Open Source. [online]. Dostupné z: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>

MACDONALD, Matthew. Pro WPF 4.5 in C#, 4th Edition [online]. 4th ed. Apress, 2012 [cit. 2021-04-08]. ISBN 1430243651. Dostupné z: http://sd.blackball.lv/library/Pro_WPF_4.5_in_CSharp_4th_edition.pdf

SELLS, Chris a Ian GRIFFITHS. Programming WPF, 2nd Edition [online]. 2th ed. O'Reilly Media, 2007 [cit. 2021-04-08]. ISBN 9780596510374. Dostupné z: [http://index-of.es/Programming/O'Reilly%20Programming%20Series/O'Reilly%20Programming%20WPF%20\(2nd%20Edition\).pdf](http://index-of.es/Programming/O'Reilly%20Programming%20Series/O'Reilly%20Programming%20WPF%20(2nd%20Edition).pdf)

Objektově orientované programování v C# .NET. itnetwork.cz – David Čápka – Ajt'ácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Copyright © 2021 itnetwork.cz. Veškerý obsah webu [cit. 24.04.2021]. Dostupné z: <https://www.itnetwork.cz/csharp/oop>

WPF - Okenní aplikace v C# .NET. itnetwork.cz – David Čápka – Ajt'ácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Copyright © 2021 itnetwork.cz. Veškerý obsah webu [cit. 24.04.2021]. Dostupné z: <https://www.itnetwork.cz/csharp/wpf>

Soubory a síť v C# .NET. itnetwork.cz – David Čápka – Ajt'ácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Copyright © 2021 itnetwork.cz. Veškerý obsah webu [cit. 24.04.2021]. Dostupné z: <https://www.itnetwork.cz/csharp/soubory/>

Stack Overflow - Where Developers Learn, Share, & Build Careers. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/>

C# Best Practices – Faisal Pathan – 2019 – DZone Web Dev. DZone [online]. [cit. 24.04.2021] Dostupné z: <https://dzone.com/articles/c-best-practices>

Some Best Practices for C# Application Development (Explained) - CodeProject. CodeProject – Kunal Chowdhury – For those who code [online]. Copyright © [cit. 24.04.2021]. Dostupné z: <https://www.codeproject.com/Articles/118853/Some-Best-Practices-for-C-Application-Developmen>

Dokumentace k Windows Presentation Foundation pro .NET 5 | Microsoft Docs. [online]. Copyright © Microsoft 2021 [cit. 24.04.2021]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/desktop/wpf/?view=netdesktop-5.0>

WPF Tutorial - Tutorialspoint. Apache HTTP Server Test Page powered by CentOS [online]. Copyright © Copyright 2021. All Rights Reserved. [cit. 24.04.2021]. Dostupné z: <https://www.tutorialspoint.com/wpf/index.htm>