

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Aplikace pro hlídání změn obsahu internetových stránek



2020

Vedoucí práce: doc. RNDr. Jan  
Konečný, Ph.D.

Tomáš Veselský

Studijní obor: Aplikovaná informatika,  
kombinovaná forma

## **Bibliografické údaje**

Autor: Tomáš Veselský  
Název práce: Aplikace pro hlídání změn obsahu internetových stránek  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2020  
Studijní obor: Aplikovaná informatika, kombinovaná forma  
Vedoucí práce: doc. RNDr. Jan Konečný, Ph.D.  
Počet stran: 33  
Přílohy: 1 DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Tomáš Veselský  
Title: Application for watching to changes in the content of the websites  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2020  
Study field: Applied Computer Science, combined form  
Supervisor: doc. RNDr. Jan Konečný, Ph.D.  
Page count: 33  
Supplements: 1 DVD  
Thesis language: Czech

## Anotace

*Aplikace „WatchCo“ hlídá obsah internetových stránek a upozorňuje uživatele při splnění zadaných podmínek. V úvodu práce představím motivaci k tvorbě aplikace, přehled stávajících řešení s popisem jejich funkcí a návrh vlastního řešení a rozšíření funkcí. Text dále obsahuje použité technologie pro tvorbu aplikace, která je určena pro operační systém Microsoft Windows, se zaměřením na CefSharp a Chromium Embedded Framework. Následuje uživatelská příručka aplikace „WatchCo“ a poznatky z testování a řešení problémů při vývoji. V závěru je uvedena rekapitulace tvorby práce a celkové zhodnocení výsledku.*

## Synopsis

*The application „WatchCo“ watches the content of the web pages and notifies the user when the specified conditions are met. In the introduction of the thesis, I will introduce the motivation to create an application, an overview of existing solutions with a description of their functions and a proposal of my own solution and feature expansion. The text also contains the technologies used to create an application that is designed for the Microsoft Windows operating system, focusing on CefSharp and Chromium Embedded Framework. The following is the user manual of the „WatchCo“ application and knowledge of testing and solving development problems. In the conclusion is listed recapitulation of work creation and overall evaluation of the result.*

**Klíčová slova:** hlídání změn obsahu; internetová stránka; podmínky; upozornění; desktopová portable aplikace; Microsoft Windows; C#; WPF; XML; CEF; CefSharp

**Keywords:** watching of content changes; web page; conditions; notice; desktop portable application; Microsoft Windows; C#; WPF; XML; CEF; CefSharp

Děkuji vedoucímu mé práce doc. RNDr. Janu Konečnému, Ph.D. za jeho ochotu a pomoc při výběru tématu a za konzultace a cenné rady v průběhu psaní práce. Také moc děkuji mé manželce Kláře a synovi Antonínovi, kteří mi pomáhali nemyslet jen na školu a zároveň mi poskytli dostatek času, abych se jí mohl intenzivně věnovat. Na závěr ještě děkuji mým kolegům z práce za jejich přínosnou zpětnou vazbu z testování aplikace.

datum odevzdání práce

podpis autora

# Obsah

Seznam použitých zkratk	8
<b>1 Motivace k tvorbě hlídače změn obsahu internetových stránek</b>	<b>9</b>
<b>2 Existující řešení</b>	<b>10</b>
2.1 HZOIS dle metody vyhodnocování změny . . . . .	10
2.1.1 Zástupce pracující s XPath . . . . .	11
2.1.2 Zástupce pracující se změnou vzhledu . . . . .	11
2.2 HZOIS dle typu aplikace . . . . .	12
2.2.1 Desktopová aplikace . . . . .	12
2.2.2 Webová aplikace . . . . .	13
2.2.3 Rozšíření internetového prohlížeče . . . . .	14
2.2.4 Mobilní aplikace . . . . .	15
<b>3 Návrh vlastního řešení a přidání funkcí</b>	<b>17</b>
3.1 Kombinace podmínek . . . . .	17
3.2 Jednoduché zadání hlídání . . . . .	18
3.3 Přehledné zobrazení stavu . . . . .	19
3.4 Desktopová portable aplikace . . . . .	19
<b>4 Použité technologie</b>	<b>20</b>
4.1 C#, WPF a Microsoft Visual Studio . . . . .	20
4.2 XML . . . . .	20
4.3 Chromium Embedded Framework . . . . .	20
4.4 CefSharp . . . . .	21
<b>5 Uživatelská příručka aplikace WatchCo</b>	<b>23</b>
5.1 Požadavky aplikace . . . . .	23
5.2 Umístění a spuštění aplikace . . . . .	23
5.3 Uživatelské rozhraní . . . . .	23
5.3.1 Hlavní okno . . . . .	23
5.3.2 Okno pro nastavení hlídání . . . . .	24
5.3.3 Okno pro nastavení podmínky . . . . .	26
5.3.4 Dialogová okna . . . . .	27
<b>6 Testování a řešení problémů při vývoji aplikace WatchCo</b>	<b>28</b>
6.1 Vylepšení uživatelského rozhraní . . . . .	28
6.2 Problém se zobrazením oken se seznamy . . . . .	28
6.3 Problém s prvkem internetového prohlížeče . . . . .	28
6.4 Problém s načtením dynamicky generovaných prvků internetových stránek . . . . .	29
6.5 Problém s využitím paměti . . . . .	29
<b>Závěr</b>	<b>31</b>

<b>Literatura</b>	<b>32</b>
<b>A Obsah přiloženého DVD</b>	<b>33</b>

## Seznam obrázků

1	Výběr hlídané oblasti na internetové stránce přes XPath . . . . .	11
2	Výběr hlídané oblasti na internetové stránce přes nastavení hranic . . . . .	12
3	Hlavní okno desktopové aplikace NotiPage . . . . .	13
4	Úvodní stránka webové aplikace Wachete . . . . .	14
5	Okno rozšíření internetového prohlížeče od Visualping . . . . .	15
6	Snímek obrazovky s mobilní aplikací od Wachete . . . . .	16
7	Případy užití pro práci s hlídáními a podmínkami . . . . .	18
8	Případy užití systému hlídání . . . . .	19
9	Hlavní okno aplikace WatchCo . . . . .	24
10	Okno pro nastavení hlídání v aplikaci WatchCo . . . . .	26
11	Okno pro nastavení podmínky v aplikaci WatchCo . . . . .	27
12	Zobrazení využití paměti v nástroji <i>Sledování prostředků</i> při problému se zabráním celé dostupné paměti . . . . .	30
13	Zobrazení nárůstu a následného uvolnění využití paměti v diagnostickém nástroji ve Visual Studiu při problému se zabráním celé dostupné paměti . . . . .	30

## Seznam tabulek

1	Seznam prozkoumaných hlídačů změn obsahu internetových stránek z počátku roku 2019 a jejich klíčové vlastnosti . . . . .	10
---	--	----

## Seznam zdrojových kódů

1	Ukázka cesty k prvku v XPath . . . . .	11
2	Přidání funkce do prohlížeče pro získání cesty prvku v XPath . . . . .	22

## Seznam použitých zkratek

BSD	Berkeley Software Distribution
CEF	Chromium Embedded Framework
CLI	Command Line Interface
CSS	Cascading Style Sheets
DVD	Digital Versatile Disc
GB	gigabyte
HTML	Hypertext Markup Language
HZOIS	hlídač změn obsahu internetových stránek
MB	megabyte
RSS	Rich Site Summary
URL	Uniform Resource Locator
USB	Universal Serial Bus
WPF	Windows Presentation Foundation
W3C	World Wide Web Consortium
XML	Extensible Markup Language



# 1 Motivace k tvorbě hlídače změn obsahu internetových stránek

Obsah internetových stránek, stejně jako celé odvětví informačních technologií, prochází vývojem, díky kterému se neustále rozšiřují možnosti využití. Od statických stránek s textem a jednoduchou grafikou jsme dospěli ke stránkám s dynamicky se měnícím obsahem, který přináší uživatelům aktuální informace ve všech možných formách. Internetové stránky jsou tak dnes spíš webové aplikace než dokumenty psané pouze v jazyce HTML. Díky tomu mají sice uživatelé přístup k požadovaným datům, ale většinou je musí hlídat, získávat a zpracovávat sami.

Tuto situaci částečně řeší služby jako například informační e-maily o novinkách, čtečky RSS kanálů nebo různé srovnávače. Jejich použití je ale příliš specifické, a nelze je použít na obsah internetových stránek, které tyto služby samy nenabízí nebo s poskytovatelem služeb nespolupracují.

Řešení problému nabízí hlídače změn obsahu internetových stránek (dále jen HZOIS). Lze je použít mnohem obecněji než zmíněné informační e-maily, čtečky RSS kanálů a srovnávače. Uživatel má navíc možnost nastavovat parametry hlídání jako např. periodu kontroly, URL internetové stránky, výběr hlídané hodnoty na internetové stránce a další. V této práci popisuji některé zástupce existujících HZOIS a následně i tvorbu nového řešení včetně použitých technologií.

V další kapitole dostupná řešení proberu blíže, protože z nich bude vycházet můj návrh rozšíření o další funkce, které navíc uživatelům pomohou se zpracováním dat a s usnadněním používání programu. Jde především o možnost kombinace podmínek, díky níž dochází k vyhodnocení dílčích výsledků automaticky a nabízí větší škálu využití. Dalšími klíčovými vlastnostmi jsou přehledné zpracování a jednoduché ovládání uživatelského rozhraní desktopové portable aplikace WatchCo, která je určena pro operační systém Microsoft Windows. Vytvoření aplikace je součástí bakalářské práce.

## 2 Existující řešení

Na začátku roku 2019 jsem provedl průzkum existujících HZOIS. Zjistil jsem, že řešení bylo více, než jsem předpokládal. Často si ale byla podobná a po seznámení se s čtrnácti z nich jsem se rozhodl pouze určit nejpodstatnější rozdíly a vybrat zástupce, kteří by řešení v mé práci reprezentovali (tab. 1).

Nakonec jsem zvolil jen dvě důležité vlastnosti, podle kterých jsem řešení rozřídil. Jedná se o metodu vyhodnocování změny obsahu internetové stránky a o typ aplikace. Ostatní vlastnosti jako například dostupnost podpory, způsob upozornění, požadavky aplikace, cena atd. nebyly rozhodující pro následný návrh vlastní aplikace s rozšířením funkcí.

Tabulka 1: Seznam prozkoumaných hlídačů změn obsahu internetových stránek z počátku roku 2019 a jejich klíčové vlastnosti

Název HZOIS	Metoda vyhodnocování změny	Typ aplikace
ContentKing	XPath	webová
Distill	XPath	mobilní, rozšíření internetového prohlížeče
ChangeTower	XPath, změna vzhledu	webová
NotiPage	XPath	desktopová
OnWebChange	XPath	webová
PageCrawl	XPath	webová
Pagescreen	změna vzhledu	webová
Sken	XPath, změna vzhledu	webová, mobilní, rozšíření internetového prohlížeče
Trackly	XPath	webová
Visualping	XPath, změna vzhledu	webová, mobilní, rozšíření internetového prohlížeče
Wachete	XPath	webová, mobilní, rozšíření internetového prohlížeče
Web pages scanner	XPath	rozšíření internetového prohlížeče
WebChangeMonitor	XPath	desktopová
WebSite-Watcher	XPath	desktopová, mobilní

### 2.1 HZOIS dle metody vyhodnocování změny

Metodu vyhodnocování určuje způsob, jakým se kontroluje změna obsahu. Většina řešení používá metodu získání textového obsahu HTML prvku dle dříve nastavené cesty k tomuto prvku, která je definována standardem W3C [1] v ja-

zyce XPath (zdr. kód 1). Ostatní řešení pak nepracují s textem, ale s grafickým vzhledem internetové stránky, kde se hlídá předem zvolená oblast.

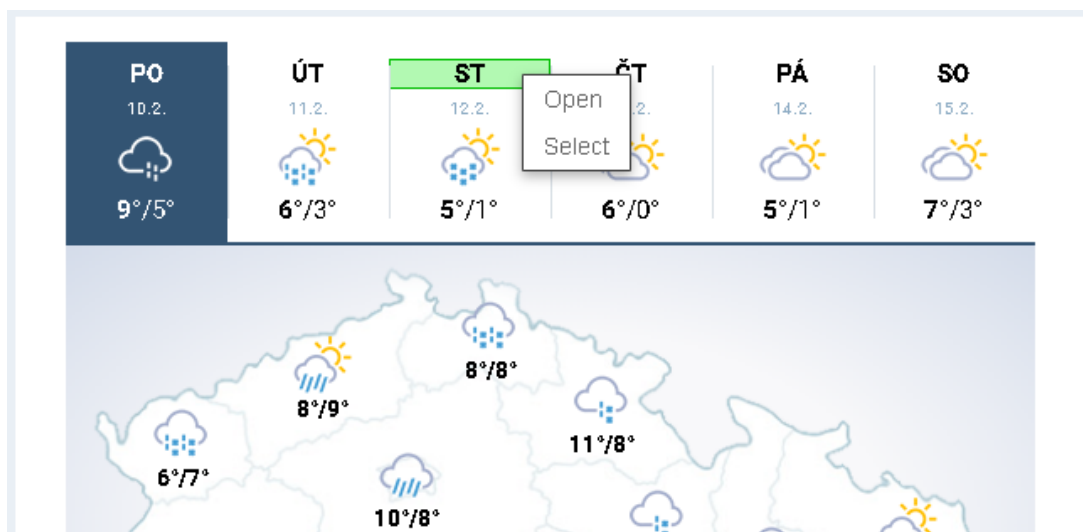
```
1 /html[1]/body[1]/div[3]/main[1]/article[1]/section[1]/div[1]
  ↪ /menu[1]/li[3]/a[1]/span[1]
```

Zdrojový kód 1: Ukázka cesty k prvku v XPath

### 2.1.1 Zástupce pracující s XPath

Vzorovým příkladem v této kategorii, a z mého pohledu celkově nejpracovanějším řešením, je webová aplikace Wachete, která se nachází na adrese [www.wachete.com](http://www.wachete.com). Po zadání URL adresy do Wachete se zobrazí internetová stránka ve vestavěném nástroji, který umožňuje procházení stránek nebo označení prvku s jehož obsahem chceme dále pracovat (obr. 1). Umožňuje i přepnutí do zjednodušeného zobrazení, které ignoruje většinu grafických prvků.

Po výběru prvku lze nastavit mnoho parametrů hlídání, včetně zadání podmínky pro upozornění. Těmi se budu zabývat v kapitole 3.1, která je součástí návrhu rozšíření funkcí. Možností této aplikace je opravdu hodně, ale velká část z nich je zpoplatněna. Nejspíš i díky tomu za asi jeden rok přibylo mnoho nových služeb a rozšíření. Aplikaci provozuje firma Wachete s.r.o. – Prague, která sídlí v České republice.[2]



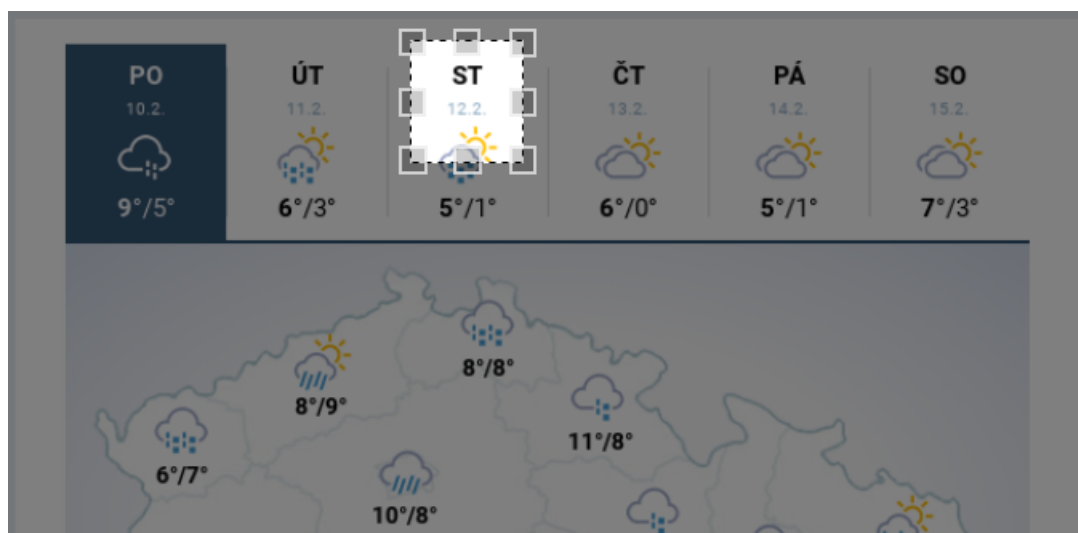
Obrázek 1: Výběr hlídání oblasti na internetové stránce přes XPath

### 2.1.2 Zástupce pracující se změnou vzhledu

Jedním z kvalitně zpracovaných řešení, pracujících se vzhledem internetových stránek, je webová aplikace Visualping, provozovaná na adrese [visualping.io](http://visualping.io). Po

zadání URL adresy do Visualping se stránka načte do vestavěného nástroje. Primární způsob výběru je označení libovolné oblasti na stránce (obr. 2). Oblast je pak při kontrole graficky porovnávána s aktuálním vzhledem, a pokud dojde ke změně zadaného rozsahu, je vyvoláno upozornění.

Dále jsou dostupné i možnosti pro hlídání veškerého textu na stránce nebo podle vybraného prvku, který nejspíš také využívá získání cesty k prvku dle XPath. Tyto možnosti však nejsou na takové úrovni jako u Wachete.



Obrázek 2: Výběr hlídané oblasti na internetové stránce přes nastavení hranic

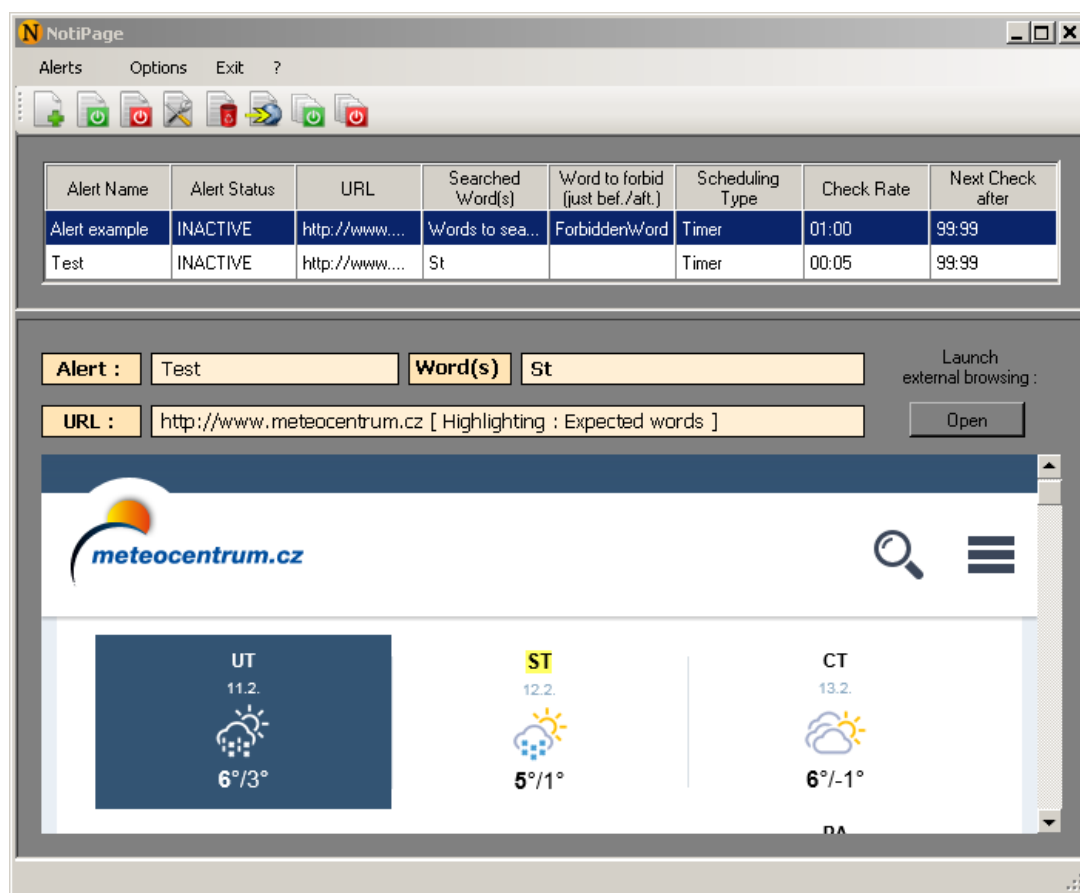
## 2.2 HZOIS dle typu aplikace

Trend posledních let je nahrazovat desktopové aplikace webovými a mobilními aplikacemi a HZOIS nejsou výjimkou. Tomuto vývoji odpovídají i počty aktuálně dostupných řešení HZOIS podle typu aplikace. Pokud je však potřeba aby byla aplikace dostupná pro různé internetové stránky a aby používala funkce internetového prohlížeče, tak je další možností vytvoření rozšíření internetového prohlížeče.

### 2.2.1 Desktopová aplikace

Desktopové verze HZOIS jsou dnes již spíše historické projekty bez podpory a aktualizací, kvůli čemuž nelze na správnou funkčnost spoléhat. Proto v současnosti v nabídce softwaru prakticky chybí. Nejaktuálnější desktopová aplikace, kterou jsem našel, se jmenuje NotiPage a poslední úpravy jsou dle informací o programu z roku 2012 (obr. 3). Aplikaci je nutné nainstalovat, a kvůli možnosti nastavení jen anglické nebo francouzské jazykové verze je už tento úkon uživatelsky nepřívětivý. Nabídka možností je i pro základní využití nedostačující, protože nelze pracovat s jednotlivými prvky internetové stránky, a tak je i nastavení podmínek velmi omezené. Během testování se mi často stávalo, že aplikace přestala

odpovídat a bylo třeba ji ukončit. K dalším nedostatkům patří i neintuitivní ovládání a chybějící nápověda. Přes to všechno bych řekl, že aplikace byla ve své době jedna z nejrozšířenějších, protože i dnes ji lze stáhnout z desítek adres na internetu.

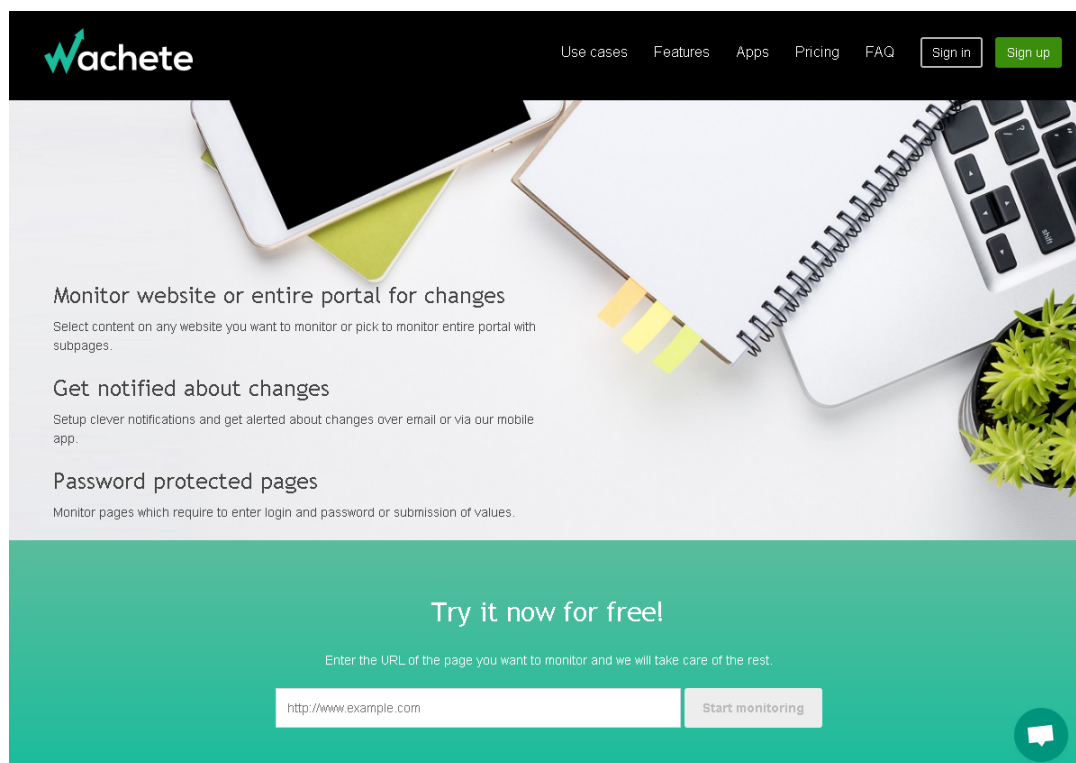


Obrázek 3: Hlavní okno desktopové aplikace NotiPage

### 2.2.2 Webová aplikace

Webové aplikace mají oproti desktopovým mnoho výhod. Aktualizace SW probíhá jen na straně serveru. Díky tomu je dostupná vždy jen jedna verze, což je pro vývoj a správu aplikace velký přínos. Uživatel navíc nemusí nic instalovat a stačí mu znát URL adresu, kde je aplikace umístěna. Výpočetní výkon je potřeba na straně serveru, takže na klienta jsou kladeny minimální požadavky, a může tak použít jakékoli zařízení s libovolným operačním systémem a internetovým prohlížečem. Poskytovatel i uživatel aplikace si tak přechodem na webovou aplikaci téměř ve všech ohledech polepší. Jednou z mála nevýhod je závislost uživatele na dostupnosti webové aplikace, která může být omezena z důvodu údržby nebo poruchy. Webové aplikace upozorňují uživatele zasláním e-mailu na zadanou adresu. I toto je prostor pro výskyt problémů, kdy uživatel nebude upozorněn například

z důvodu plné poštovní schránky. Zástupce této skupiny je již zmíněná aplikace Wachete (obr. 4).



Obrázek 4: Úvodní stránka webové aplikace Wachete

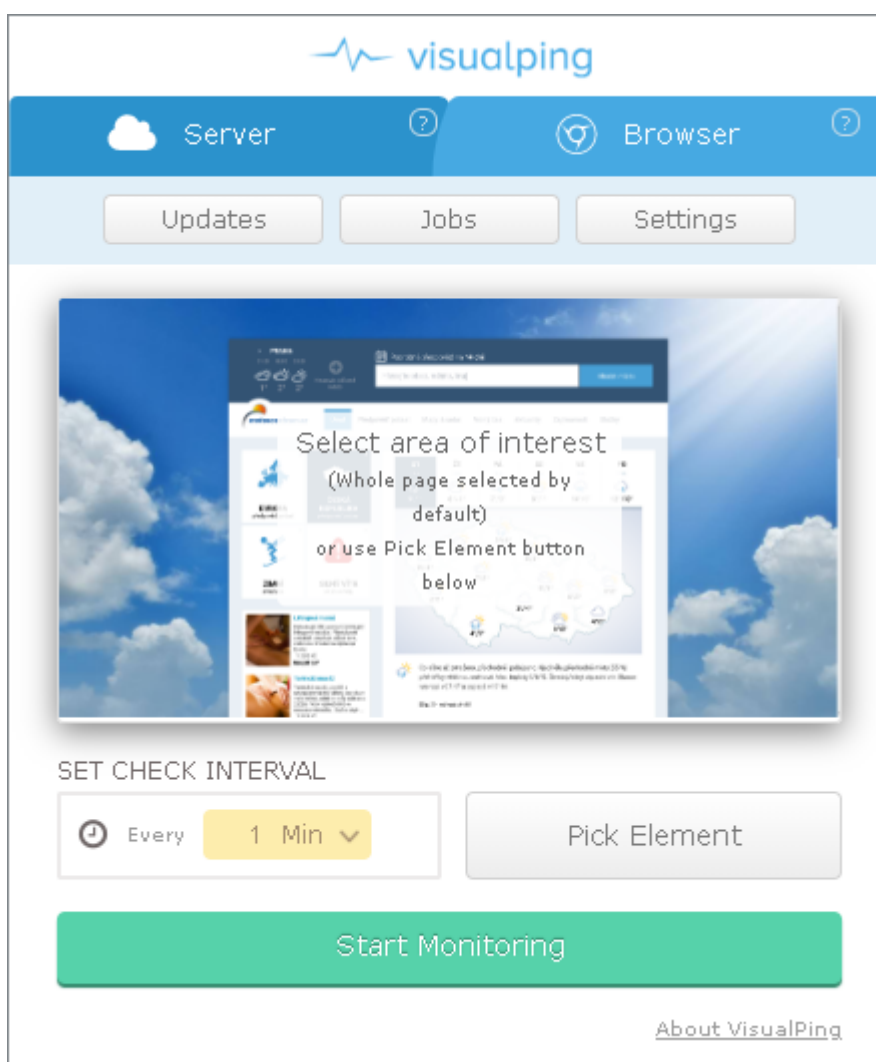
### 2.2.3 Rozšíření internetového prohlížeče

Dalším krokem, jak uživatelům zprostředkovat ještě snadnější používání HZOIS, je možnost rozšíření internetového prohlížeče. Tuto funkci umožňují všechny hojně používané prohlížeče. Toho využili i zmíněné aplikace Wachete a Visualping, které nabízí na svých stránkách kromě webových aplikací také instalaci rozšíření pro několik internetových prohlížečů.

Rozšíření od Wachete však pouze přejde na vlastní internetovou stránku s webovou aplikací, které předá původní URL adresu. Je to příjemné zjednodušení, ale jinou přidanou hodnotu toto řešení nemá.

Oproti tomu rozšíření od Visualping přináší možnosti, které nemá ani jejich webová aplikace (obr. 5). Například pro výběr oblasti lze použít i označení HTML prvku, které na rozdíl od webové aplikace funguje dle očekávání, anebo, že při práci v režimu „Browser“ namísto „Server“ funguje rozšíření nezávisle na serveru, takže je služba zajištěna i při výpadku webové aplikace.

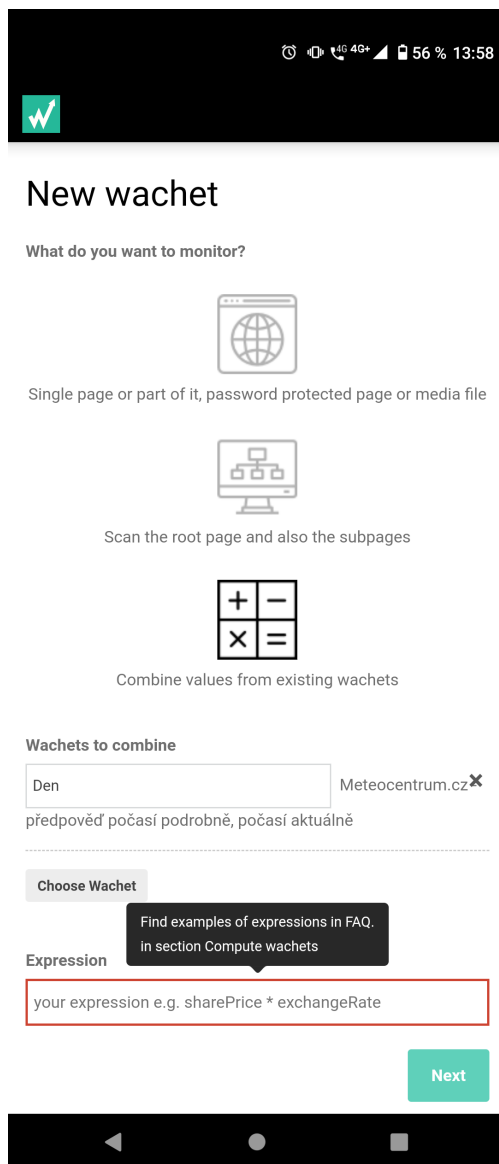
Rozšíření tak může být zajímavou alternativou i k desktopové aplikaci. Podmínkou však je stále spuštěný prohlížeč, aby hlídání dávalo smysl.



Obrázek 5: Okno rozšíření internetového prohlížeče od Visualping

#### 2.2.4 Mobilní aplikace

Mobilních aplikací, které nabízejí HZOIS, je minimálně pro operační systém Android dostatek. Podle uživatelských recenzí je ale těžké najít aplikaci, která funguje spolehlivě pro různá zařízení a verze Androidu. Bezplatné používání je u většiny aplikací podmíněno zobrazováním reklamních sdělení. Mobilní aplikace jsou také závislé na dostupnosti serveru, který zajišťuje samotné hlídání. Mají ale jednu velikou výhodu. Poskytují uživateli nejrychlejší a nejspolehlivější formu upozornění díky téměř nepřetržitému provozu mobilních zařízení, která mají většinou uživatelé vždy při ruce. Z uvedených řešení zmíním mobilní aplikaci od Wachete, neboť opět nabízí nejvíce možností, které jsou však někdy bez nápovědy jen těžko použitelné (obr. 6).



Obrázek 6: Snímek obrazovky s mobilní aplikací od Wachete



## 3 Návrh vlastního řešení a přidání funkcí

Mým záměrem ještě před studiem existujících řešení bylo vytvořit aplikaci, která bude uživateli pomáhat s rutinními úkony při získávání a zpracovávání dat z internetových stránek a bude co nejpraktičtější. Existující moderní řešení nabízejí přívětivé uživatelské rozhraní, získávání dat, nastavení podmínky a upozornění uživatele při jejím splnění.

Získávání dat funguje velice dobře, ale většina HZOIS má nejkratší periodu aktualizace dat pět minut, a to i u placených verzí. To je pro některé případy příliš dlouhá doba.

Nastavení podmínky pro upozornění poskytuje většinou spoustu možností, od určení formátu dat přes výběr operace pro vyhodnocení podmínky až k případnému zadání vlastního logického výrazu. Žádný HZOIS však nenabízí kombinaci podmínek obsahujících data z různých internetových stránek, což je například případ srovnávače cen. Výjimkou je sice mobilní aplikace od Wachtete, ale způsob zadání kombinace podmínek ve formě výrazu bez použitelné nápovědy je uživatelsky nepřívětivý a také má velmi omezené možnosti použití, protože výrazy mohou obsahovat jen základní matematické operace a spojování textových řetězců.

K upozornění uživatele při splnění zadaných podmínek dochází nejčastěji přes zaslání e-mailu. Pro mnoho uživatelů, kteří mají e-mailovou schránku a jsou neustále informováni o nových zprávách díky mobilnímu internetu, je to praktická forma. Problém však může nastat, pokud uživateli chodí velké množství e-mailů, které nestíhá kontrolovat a zpracovávat, a proto na upozornění zareaguje až s velkým zpožděním anebo vůbec.

Tyto poznatky mě vedly k určení vlastností a návrhu mého řešení HZOIS tak, aby nabídlo uživatelům širší a snadnější využití. Výslednou aplikaci, která toto řešení zahrnuje jsem pojmenoval WatchCo, což je složenina z anglických slov „watch“ a „conditions“. Tedy volně přeloženo „hlídej podmínky“. Shoda výslovnosti s českým nespisovným slovem „vočko“ má také svůj význam, protože mě inspiroval při tvorbě loga, které v aplikaci používám. Klíčové vlastnosti návrhu řešení jsou popsány ve zbytku této kapitoly.

### 3.1 Kombinace podmínek

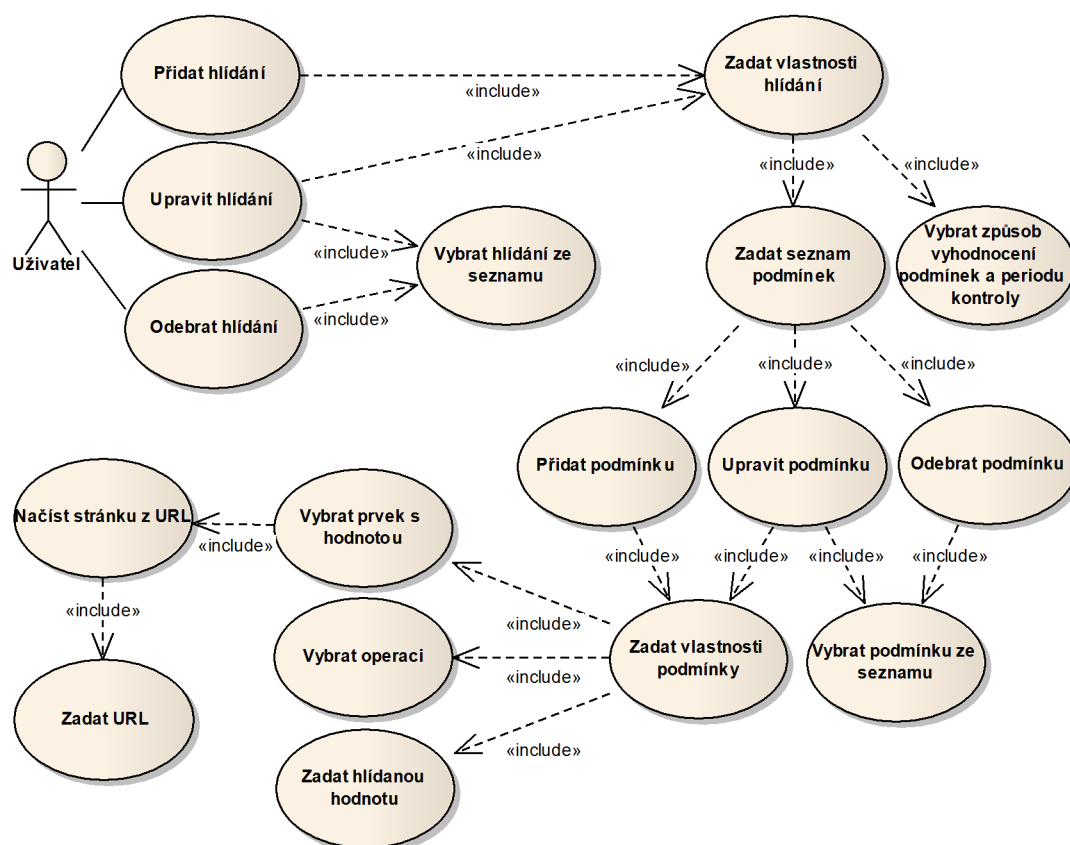
Nejdůležitější vlastností, která do problematiky HZOIS přináší zcela nové možnosti, je kombinace podmínek pro více hlídaných hodnot z několika internetových stránek. Záměrem bylo vytvořit jednoduchý nástroj pro zadávání podmínek, kterým půjde zadat podmínky pro jednotlivě hlídané hodnoty a následně určit, jak dílčí výsledky vyhodnotit.

Základní případ užití je například hlídání ceny na několika stránkách, kdy při poklesu ceny na jedné ze stránek pod stanovenou mez bude uživatel upozorněn. Další z příkladů je hlídání počtu volných míst v autobuse a v navazujícím přípoji, ve kterých uživatel potřebuje rezervovat místa pro celou skupinu. Při potřebném počtu volných míst v obou spojích zároveň, bude uživatel upozorněn.

Z technického hlediska se tedy nejedná o žádnou složitou funkci, ale z pohledu uživatele přináší zásadní vylepšení, které za něj vyhodnocuje data bez potřeby osobně hlídat dílčí výsledky.

### 3.2 Jednoduché zadání hlídání

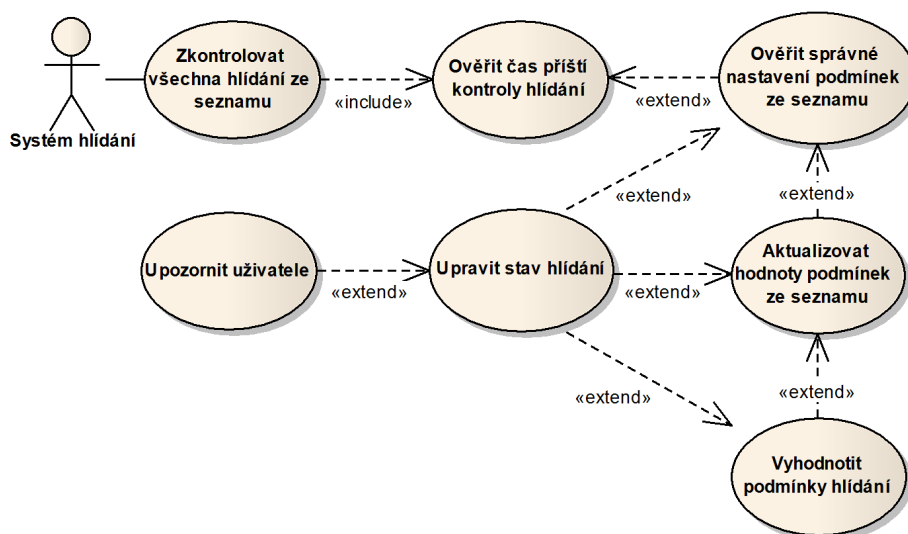
V aplikaci WatchCo jsem se rozhodl upřednostnit jednoduchost zadání před velkým množstvím možností, které nabízí např. aplikace Wachete. Je to z důvodu, že u některých nastavení pak musí uživatel dohledat, přečíst a pochopit, jak je má vlastně použít. Jde například o zadávání výrazů nebo práci s různými formáty dat. Díky tomu sice dojde k mírnému omezení využití, ale pro uživatele by mělo být jednodušší, intuitivnější a rychlejší nastavit hlídání, protože se setká jen se základními a známými ovládacími prvky, které navíc obsahují nápovědu. Celé zadání hlídání (obr. 7) je navrženo tak, aby uživatel musel co nejméně psát hodnoty ručně a mohl použít rozevírací nabídky a další prvky, které může ovládat jen myší.



Obrázek 7: Případy užití pro práci s hlídáními a podmínkami

### 3.3 Přehledné zobrazení stavu

Protože při hlídání obsahu internetové stránky může dojít k mnoha různým problémům, které znemožní načtení požadovaného obsahu, je třeba uživatele o výsledku kontroly informovat. V aplikaci WatchCo se uživatel v hlavním okně se seznamem hlídání dozví, jak dopadla poslední kontrola každého hlídání (obr. 8). V případě chyby bude stav obsahovat souhrn všech chyb, které pak může uživatel vyřešit v nastavení hlídání, kde budou chyby popsány u každé podmínky. Mělo by tak docházet k co nejkratším výpadkům v hlídání, způsobených například změnou struktury internetové stránky nebo neúplným nastavením hlídání.



Obrázek 8: Případy užití systému hlídání

### 3.4 Desktopová portable aplikace

O výhodách a nevýhodách desktopových aplikací jsem se již zmiňoval v kapitole 2.2.2. Přes současný trend webových a mobilních aplikací jsem se ale rozhodl vytvořit aplikaci WatchCo jako desktopovou portable verzi pro operační systém Microsoft Windows. Jedním z důvodů je nezávislost na dostupnosti serveru s webovou aplikací a serverových služeb, které používají mobilní aplikace. To se týká i odesílání upozornění do e-mailové schránky uživatele. Díky tomu půjde mnou navrženou aplikaci použít i pro intranet a uživatel nemusí před prvním spuštěním kromě minimálních požadavků na systém nic řešit. Dle mého předpokladu by tak desktopové řešení mělo mít vyšší spolehlivost i výkon, který umožní četnější kontrolu než obvyklých každých pět minut. Uživatel může spustit aplikaci přímo na svém počítači, z USB flash disku nebo třeba z cloudového úložiště. Veškeré nastavení bude uloženo v adresáři se spustitelným souborem. Oproti webovým a mobilním aplikacím odpadají náklady na provoz a údržbu serverové části, a vzhledem k nedostupnosti desktopových řešení, nemá mé řešení ve své kategorii přímou konkurenci.

## 4 Použité technologie

V této kapitole jsou popsány technologie, které byly použity při tvorbě aplikace WatchCo.

### 4.1 C#, WPF a Microsoft Visual Studio

Pro vývoj WatchCo jsem zvolil programovací jazyk C#, který je jednoduchý, moderní, objektově orientovaný a typově bezpečný programovací jazyk. C# má své kořeny v rodině jazyků C. Vývojářům umožňuje vytvářet řadu bezpečných a robustních aplikací, které fungují na .NET Frameworku. Pomocí C# lze vytvářet klientské aplikace pro operační systém Microsoft Windows, webové služby, distribuované komponenty, aplikace klient-server, databázové aplikace a mnoho dalších.[3]

Uživatelské rozhraní aplikace jsem vytvořil ve Windows Presentation Foundation (dále jen WPF), které umožňuje vytvářet klientské aplikace pro operační systém Microsoft Windows s vizuálně poutavými uživatelskými rozhraními. Jádrem WPF je modul vykreslování založený na rozlišení a vektorový vykreslovací modul, který je sestaven tak, aby využíval výhody moderního grafického hardwaru. WPF rozšiřuje jádro o komplexní sadu funkcí pro vývoj aplikací, které zahrnují jazyk Extensible Application Markup Language (dále jen XAML), ovládací prvky, datové vazby, rozložení, 2D a 3D grafiku, animace, styly, šablony, dokumenty, multimédia, text a typografie. WPF je součástí .NET.[4]

Pro psaní kódu jsem použil Microsoft Visual Studio Community 2017, které obsahuje editor kódu, návrháře uživatelského rozhraní, integrovaný debugger a mnoho dalších nástrojů, které usnadňují vývoj aplikací založených na jazyce C# a .NET Framework.[5]

### 4.2 XML

Nastavení WatchCo se zadanými i posledními načtenými hodnotami se automaticky ukládá do souboru XML. Pro ukládání a načítání je použita serializace a deserializace objektů, které provádí třída XmlSerializer.

### 4.3 Chromium Embedded Framework

Chromium Embedded Framework (dále jen CEF) je BSD licencovaný open source projekt založený Marshalllem Greenblattem v roce 2008. Je postaven na projektu Google Chromium. Na rozdíl od samotného projektu Chromium, který se zaměřuje hlavně na vývoj aplikací Google Chrome, se CEF zaměřuje na usnadnění případů použití vestavěného prohlížeče v aplikacích třetích stran. CEF izoluje uživatele od základní složitosti kódu Chromium a Blink tím, že nabízí stabilní produkční rozhraní API. Většina funkcí v CEF má implicitní implementace, které

poskytují bohatou funkčnost a zároveň vyžadují jen malou nebo žádnou integrační práci od uživatele. V současné době je na celém světě použito více než 100 milionů instancí CEF.

CEF podporuje širokou škálu programovacích jazyků a operačních systémů a lze jej snadno integrovat do nových i stávajících aplikací. Byl navržen od základů s ohledem na výkon a snadné použití. Základní rámec zahrnuje programovací rozhraní C a C++ vystavená prostřednictvím nativních knihoven, které izolují hostitelskou aplikaci od podrobností implementace Chromium a Blink. Poskytuje úzkou integraci mezi prohlížečem a hostitelskou aplikací, včetně podpory vlastních pluginů, protokolů, objektů JavaScriptu a rozšíření JavaScriptu. Hostitelská aplikace může volitelně řídit načítání prostředků, navigaci, kontextová menu, tisk a další, přičemž využívá stejný výkon a technologie HTML5 dostupné ve webovém prohlížeči Google Chrome. V mém řešení jsem použil CEF s číslem verze 75.1.14.[6]

## 4.4 CefSharp

CefSharp je .NET obal kolem projektu CEF. Nabízí možnost, jak do aplikace C# nebo VB.NET vložit plnohodnotný internetový prohlížeč kompatibilní s moderními internetovými standardy. Podporuje audio a video prvky HTML5, JavaScript, CSS3 a HTML5. CefSharp se inicializuje za méně než 10 ms a internetové stránky se vykreslují progresivně. CefSharp poskytuje ovládací prvky internetového prohlížeče pro aplikace navržené ve WinForms a WPF. Navíc dává možnost použít i verzi pro automatizační procesy, která nevykresluje prohlížeč na obrazovku.

K obsahu internetové stránky lze přistupovat a manipulovat s ním pomocí můstku mezi jazyky .NET a JavaScript. CefSharp může spouštět a vkládat JavaScript na stránky a při spouštění událostí JavaScriptu přijímat zpětná volání. Jde použít k zobrazení integrovaného uživatelského rozhraní vytvořeného pomocí HTML5 nebo k zobrazení vzdáleného webového obsahu a webových aplikací.

CefSharp je držitelem licence BSD, což znamená, že může být použit pro proprietární a free / open source aplikace. Vývojáři poskytují úplný zdrojový kód v C# a C++ / CLI. CefSharp nemá žádné externí závislosti a úplné sestavení CefSharpu má velikost asi 80 MB. CefSharp vyžaduje minimálně .NET 4.5.2. Vydání CefSharpu do verze 45 vyžadují Visual C++ Runtime 2012 a novější vydání vyžadují Visual C++ Runtime 2013 na uživatelském počítači.[7]

V aplikaci WatchCo jsem použil CefSharp s číslem verze 75.1.143. Ovládací prvek internetového prohlížeče, který CefSharp poskytuje, jsem použil k vytvoření nástroje pro výběr prvku z internetové stránky a k získání jeho cesty v XPath. Funkci získání cesty prvku v XPath však CefSharp neobsahuje. Do prohlížeče je přidána vykonáním JavaScriptu na hlavním rámci okna prohlížeče (zdr. kód 2). JavaScriptem je také upraveno chování prohlížeče tak, aby se zvýrazňoval označený prvek, a aby označení prvku nespustilo navigaci na jinou stránku. Skrytým prohlížečem pro automatizační procesy jsem řešil načítání aktuálních

hodnot z internetových stránek při kontrole podmínek. Tento proces tak probíhá na pozadí, a kvůli časové náročnosti je vykonáván ve vlastním vlákně.

```
1 browser.GetMainFrame().ExecuteJavaScriptAsync("" +
2 @"function getXPath(element)" +
3   @"{" +
4     @"var xpath = '';" +
5     @"for (; element; element = element.parentElement)" +
6       @"{" +
7         @"var id = $(element.parentElement).
           ↳ children(element.tagName).index(element) + 1;" +
8         @"id > 1 ? (id = '[' + id + ']') : (id = '');" +
9         @"xpath = '/' + element.tagName.toLowerCase() + id +
           ↳ xpath;" +
10      @"}" +
11     @"return xpath;" +
12   @"}");
```

Zdrojový kód 2: Přidání funkce do prohlížeče pro získání cesty prvku v XPath

## 5 Uživatelská příručka aplikace WatchCo

Následující odstavce obsahují informace o tom, jaké má WatchCo nároky na softwarové a hardwarové vybavení, a jak aplikaci zprovoznit a používat.

### 5.1 Požadavky aplikace

- 64bitová verze operačního systému Microsoft Windows 7 a novější
- Microsoft .NET Framework 4.5.2 a vyšší verze
- Microsoft Visual C++ 2015 Redistributable a novější
- minimálně 200 MB volného místa pro umístění aplikace
- možnost zapisovat do složky s aplikací
- minimálně 1 GB dostupné operační paměti

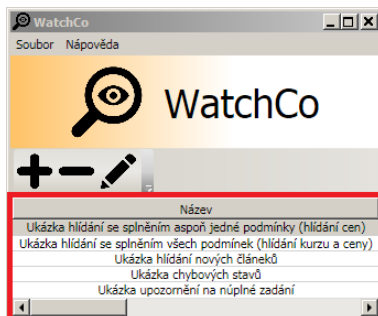
### 5.2 Umístění a spuštění aplikace

Složka s aplikací je na přiloženém DVD v adresáři bin. Protože si aplikace ukládá nastavení do složky, ve které je umístěn spustitelný soubor, je třeba celou složku zkopírovat do umístění s možností zápisu. K tomu lze využít například USB flash disk, cloudové úložiště nebo některý z disků počítače. Poté stačí ve složce spustit soubor `WatchCo.exe`.

### 5.3 Uživatelské rozhraní

#### 5.3.1 Hlavní okno

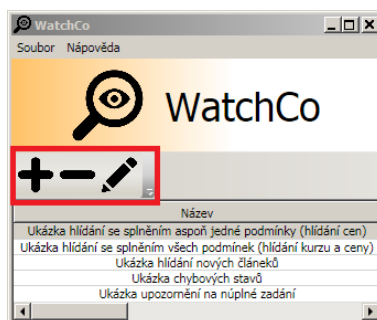
Po spuštění souboru `WatchCo.exe` se otevře hlavní okno aplikace WatchCo (obr. 9) a ze souboru `nastaveni.xml` se načte nastavení dříve vytvořených hlídání.



Pokud soubor není k dispozici, tak bude po spuštění aplikace seznam hlídání prázdný. Na DVD je ve složce s aplikací uloženo nastavení ukázkových hlídání, která představují příklady případů použití a zobrazení různých stavů hlídání.

Přes možnosti v menu lze pouze ukončit aplikaci nebo zobrazit okno s nápovědou. Při požadavku na ukončení aplikace si může uživatel vybrat, jestli chce nechat aplikaci spuštěnou na pozadí. Před ukončením aplikace dochází automaticky k uložení nastavení všech hlídání. Pokud při ukončení soubor s nastavením neexistuje, tak je před uložením vytvořen.

Panel nástrojů, který je nad seznamem hlídání, obsahuje tři tlačítka. *Přidat hlídání*, *Odebrat vybrané hlídání* a *Upravit vybrané hlídání*.



Poslední dvě tlačítka jsou aktivní jen pokud je některé hlídání vybrané. Při volbě *Přidat hlídání* nebo *Upravit vybrané hlídání* se zobrazí okno pro nastavení hlídání s nevyplněnými údaji nebo s údaji upravovaného hlídání. Při volbě *Odebrat vybrané hlídání* je vybrané hlídání trvale odstraněno ze seznamu hlídání.

V seznamu hlídání jsou zobrazeny vlastnosti jednotlivých hlídání a jejich stav při poslední kontrole. Pokud v hlídání nenastanou problémy, tak stav obsahuje pouze informaci o výsledku vyhodnocení podmínek. V případě výskytu problémů jsou v poli se stavem uvedeny unikátní názvy všech detekovaných chyb. Ty je možné detailně zobrazit v okně pro nastavení hlídání. Dvojklikem na některé hlídání v seznamu dojde k otevření okna pro nastavení hlídání s jeho aktuálním nastavením a hodnotami. Seznam hlídání jde seřadit podle kteréhokoli sloupce kliknutím na jeho název.



Obrázek 9: Hlavní okno aplikace WatchCo

### 5.3.2 Okno pro nastavení hlídání

Okno pro nastavení hlídání (obr. 10) se otevře po volbě *Přidat hlídání*, *Upravit vybrané hlídání* nebo po dvojkliku na některou položku v seznamu hlídání. Pokud nedojde k zavření okna pro nastavení hlídání, bude hlavní okno neaktivní z důvodu zajištění konzistence dat.



V hlídání je možné nastavit název, který slouží především pro přehled uživatele. Je však také použit při oznámení, že došlo ke splnění zadaných podmínek. Pole s názvem není povinné vyplnit.

Dále je třeba zadat rozsah časové platnosti nástroji pro výběr data, hodiny a minuty. Ke kontrole hlídání, která je složena z načtení aktuálních hodnot a kontroly podmínek, dochází jen v případě, že je hlídání platné.

Perioda kontroly určuje, po jaké době má dojít k další kontrole hlídání od dokončení poslední kontroly. Hodnota lze nastavit v rozmezí 1-120 minut buď vepsáním hodnoty do textového pole nebo tažením posuvníku.

Pro kontrolu hlídání musí být nastaveno jakým způsobem budou vyhodnocovány dílčí výsledky ze seznamu podmínek. Zda mají být splněny všechny, aby došlo k upozornění, nebo stačí splnit minimálně jednu z nich.

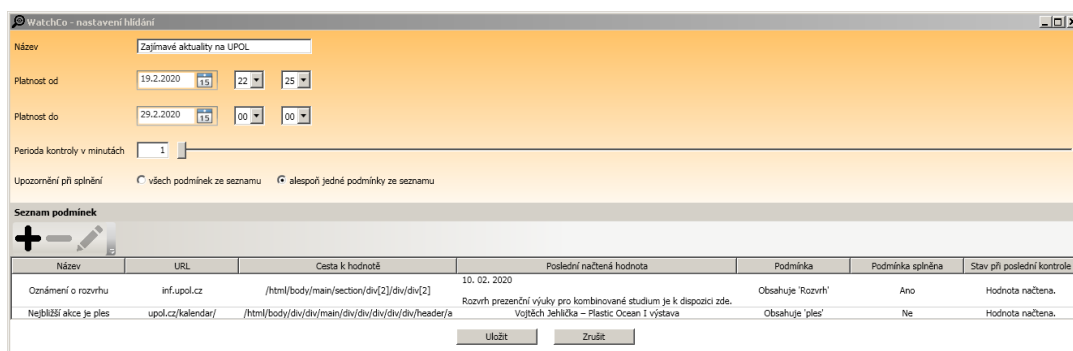
The screenshot shows a form with the following elements:

- Název:** Text input field containing "Ukázka hlídání nových článků".
- Platnost od:** Date and time selection. Date: 19.1.2020, Time: 22:57.
- Platnost do:** Date and time selection. Date: 31.1.2021, Time: 00:00.
- Perioda kontroly v minutách:** Slider control set to 30.
- Upozornění při splnění:** Radio buttons for "všech podmínek ze seznamu" (selected) and "alespoň jedné podmínky ze seznamu".

Seznam podmínek má stejné ovládací prvky jako seznam hlídání. Obsahuje panel nástrojů, ve kterém jsou tři tlačítka. *Přidat podmínku*, *Odebrat vybranou podmínku* a *Upravit vybranou podmínku*. Poslední dvě tlačítka jsou aktivní jen pokud je některá podmínka vybraná. Při volbě *Přidat podmínku* nebo *Upravit vybranou podmínku* se zobrazí okno pro nastavení podmínky s nevyplněnými údaji nebo s údaji upravované podmínky. Při volbě *Odebrat vybranou podmínku* je vybraná podmínka odstraněna ze seznamu hlídání.

V seznamu podmínek jsou zobrazeny vlastnosti jednotlivých podmínek, jejich poslední načtená hodnota a jejich stav při poslední kontrole. Pokud je vše v pořádku, tak stav obsahuje pouze informaci o načtení hodnoty. V případě výskytu problémů jsou v poli se stavem uvedeny popisy všech detekovaných chyb. Dvojklikem na některou podmínku v seznamu dojde k otevření okna pro nastavení podmínky s jejím aktuálním nastavením a hodnotami. Seznam podmínek jde seřadit podle kteréhokoli sloupce kliknutím na jeho název.

Po nastavení hlídání je třeba hlídání uložit kliknutím na tlačítko *Uložit* pod seznamem podmínek. Při uložení hlídání je vždy nastaven čas příští kontroly na čas začátku platnosti, kvůli co nejdříve vyvolání kontroly. V případě kliknutí na tlačítko *Zrušit* zůstane upravované hlídání beze změn a vytvářené hlídání bude zahazeno.



Obrázek 10: Okno pro nastavení hlídání v aplikaci WatchCo

### 5.3.3 Okno pro nastavení podmínky

Okno pro nastavení podmínky se otevře po volbě *Přidat podmínku*, *Upravit vybranou podmínku* nebo po dvojkliku na některou položku v seznamu podmínek (obr. 11). Okno je ve výchozím stavu maximalizované a lze ho pouze minimalizovat. Je to z důvodu potlačení responzivního designu, který by mohl způsobit odlišné zobrazení stránky, než jaké bude použito pro kontrolu. Dokud nedojde k zavření okna pro nastavení podmínky, bude hlavní okno i okno pro nastavení hlídání neaktivní z důvodu zajištění konzistence dat.

U podmínky je možné nastavit název, který slouží především pro přehled uživatele. Pole s názvem není povinné vyplnit.

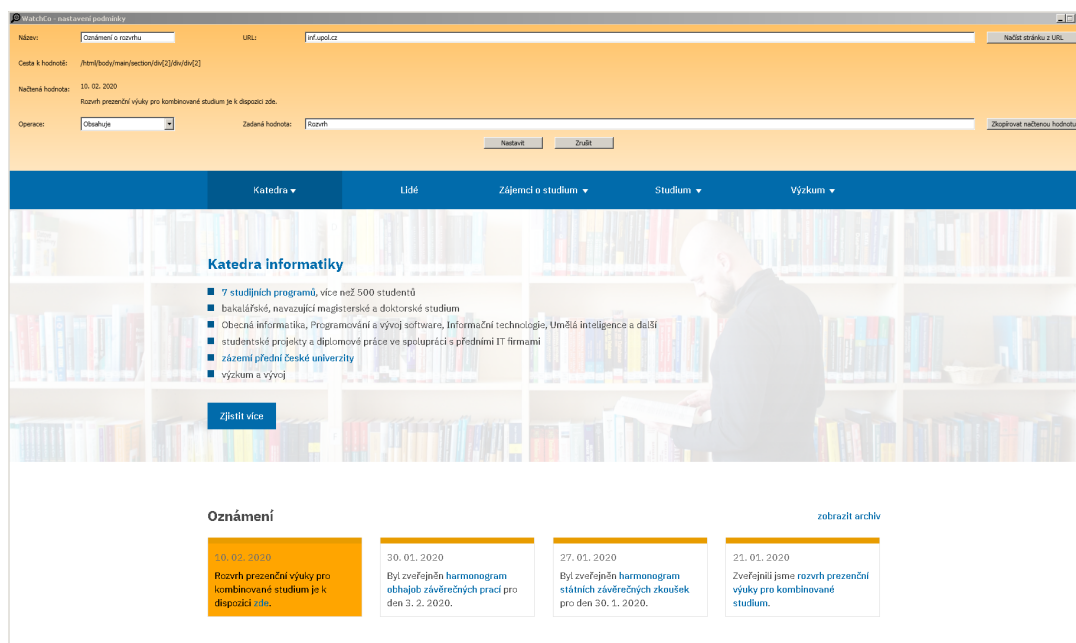
Do pole pro URL se zadává URL adresa internetové stránky, na které se nachází hodnota, kterou chceme hlídat. Po zadání URL adresy se zobrazí internetová stránka kliknutím na tlačítko *Načíst stránku z URL*. Pokud již dříve byla nastavena cesta k hodnotě, tak se po načtení stránky zvýrazní prvek, který odpovídá zadané cestě a načte se aktuální hodnota.

Pro nastavení nebo změnu cesty k hlídané hodnotě stačí kliknout na zobrazené stránce do místa s hodnotou. Po kliknutí se prvek s hodnotou zvýrazní a vyplní se pole s cestou k hodnotě a s načtenou hodnotou. Přesnost výběru hodnoty závisí na členění internetové stránky.

Dále je třeba určit operaci, která bude použita pro vyhodnocení podmínky mezi načtenou a zadanou hodnotou. Operace pracují pouze s textem, takže se i při porovnávání číselných hodnot používá lexikální řazení. Proto nemusí správně fungovat porovnávání čísel s řádově rozdílnými hodnotami.

Zadanou hodnotu uživatel může vyplnit ručně nebo může použít tlačítko *Zkopírovat načtenou hodnotu*, pokud si bude chtít načtenou hodnotu přizpůsobit.

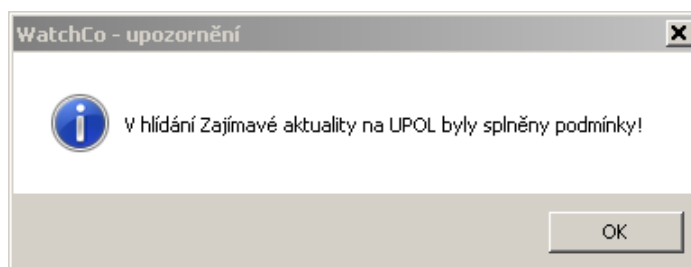
Po zadání nebo úpravě nastavení podmínky je třeba kliknout na tlačítko *Nastavit*. Tím se projeví změna do seznamu podmínek v nastavení hlídání, ale pokud nedojde v nastavení hlídání k uložení, tak bude změna zahozena. V případě kliknutí na tlačítko *Zrušit* zůstane upravovaná podmínka beze změn a vytvářená podmínka bude zahozena.



Obrázek 11: Okno pro nastavení podmínky v aplikaci WatchCo

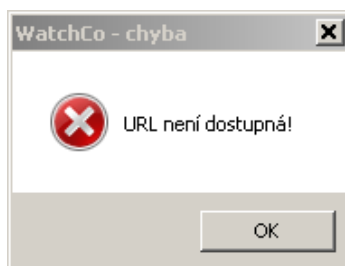
### 5.3.4 Dialogová okna

WatchCo používá pro informování uživatele o splnění podmínek hlídání dialogové okno, v jehož textu je uveden název dotyčného hlídání.



I po splnění podmínek dochází dál ke kontrole, ale okno s upozorněním se už znovu nezobrazí, pokud nedojde ke změně stavu a následně opět ke splnění podmínek.

V případě výskytu chyby při nastavování hlídání je uživatel informován chybovým dialogovým oknem, které obsahuje informaci o chybě.



## 6 Testování a řešení problémů při vývoji aplikace WatchCo

Kromě testování na mém počítači s Microsoft Windows 7 v průběhu tvorby aplikace, jsem také v poslední fázi vývoje provedl testování za pomoci tří testerů, kterými byli moji kolegové z práce. Mají sice bohaté zkušenosti s různými aplikacemi, ale zároveň již nemají pohled nezkušeného uživatele, což by bylo také přínosné. Jejich zpětnou vazbu jsem i tak velmi ocenil. Navrhli několik zásadních vylepšení uživatelského rozhraní pro snadnější ovládání a také odhalili několik problémů. V následujících odstavcích vylepšení a problémy proberu podrobně.

### 6.1 Vylepšení uživatelského rozhraní

Po testování aplikace byla doplněna nápověda ve formě tooltipů k většině ovládacích prvků. Díky tomu je okno s nápovědou již prakticky nepotřebné, až na prvotní seznámení s aplikací po prvním spuštění.

Dále byla přidána možnost otevřít okno pro nastavení hlídání a podmínky při dvojkliku na řádek v seznamu s požadovanou položkou. Je to jednodušší alternativa k označení položky v seznamu a kliknutí na tlačítko v panelu nástrojů pro úpravu hlídání nebo podmínky.

Poslední provedené vylepšení je změna grafického vzhledu hlavního okna a ostatních oken tak, aby nebyla snadno zaměnitelná. Vzhledem k tomu, že mohou být zároveň spuštěna až tři různá okna, která se postupně blokují, bylo někdy pro uživatele obtížné se zorientovat, které okno je aktivní. Do pozadí oken byly proto doplněny barevné přechody a do hlavního okna i logo a název aplikace.

### 6.2 Problém se zobrazením oken se seznamy

Díky testování se projevil problém, kdy u operačního systému Microsoft Windows 10 nebyly správně nastaveny minimální rozměry hlavního okna a okna pro nastavení hlídání tak, aby vždy byly vidět všechny prvky v okně, i pokud dojde k přidání nebo odebrání položek v seznamu. Stávalo se tak, že při zmenšení oken nebyly vidět všechny položky v sezonech.

Původně jsem ve výpočtu minimálních rozměrů okna používal konstanty pro rozměry prvků s neměnnou velikostí. Microsoft Windows 10 má ale pravděpodobně jiné rozměry prvků, podle kterých lze určit velikost vnitřku okna, než Microsoft Windows 7. Problém byl vyřešen vypočítáváním minimálních rozměrů oken z aktuálních rozměrů prvků namísto použití konstant.

### 6.3 Problém s prvkem internetového prohlížeče

Při tvorbě aplikace jsem nejdříve použil vestavěný ovládací prvek pro internetový prohlížeč z WPF, který se jmenuje WebBrowser. Řešení bylo jednodušší a nevyžadovalo použití dalších technologií. Prohlížeč však mnoho stránek nevykresloval

správně, v závislosti na nainstalované verzi Internet Exploreru, který k tomu účelu používá. Proto jsem nakonec zvolil řešení implementovat internetový prohlížeč přímo do vlastní aplikace, aby jeho chování bylo na všech zařízeních stejné. K tomu jsem využil ovládací prvek z CefSharp, o němž jsem psal v kapitole 4.4.

## 6.4 Problém s načtením dynamicky generovaných prvků internetových stránek

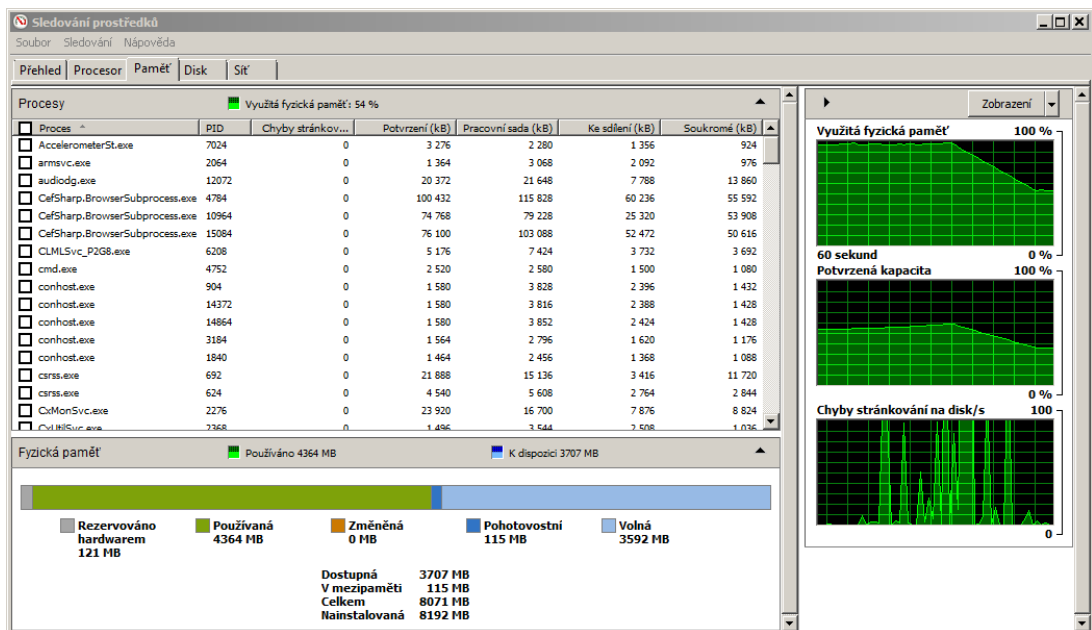
Už při tvorbě aplikace jsem zjistil, že internetové stránky mnohdy obsahují dynamicky generovaný obsah, který se mění podle uživatelských akcí. Většinou je k tomu použit kód napsaný v JavaScriptu, který například při rolování v seznamu článků přidává na konec nové a nové položky. Pokud uživatel vybere pro hlídání hodnoty takto vygenerovaný prvek, tak při kontrole buď dojde k chybě, protože nastavená cesta nevede k žádnému prvku, nebo bude načtena hodnota jiného prvku, který má sice odpovídající cestu, ale protože po načtení stránky v internetovém prohlížeči pro automatizační procesy již nedošlo k žádné akci, tak neproběhla změna obsahu jako před výběrem prvku.

Jednou z možností, jak tyto případy omezit, je úprava nebo doplnění zdrojového kódu načítaných stránek. Zasahovat do něj hlouběji jsem se však neodvážil, protože bych tím mohl nepříjemně ovlivnit funkčnost, konstrukci i vzhled stránek, což by nejspíš způsobilo další nečekané problémy. Nyní tedy může uživatel vybírat prvky v různých rozbalovacích částech stránky, ve vnořených oknech, v přepínatelném obsahu atd., přestože jejich další použití nemá smysl. Problém bude řešen až při případném pokračování ve vývoji aplikace.

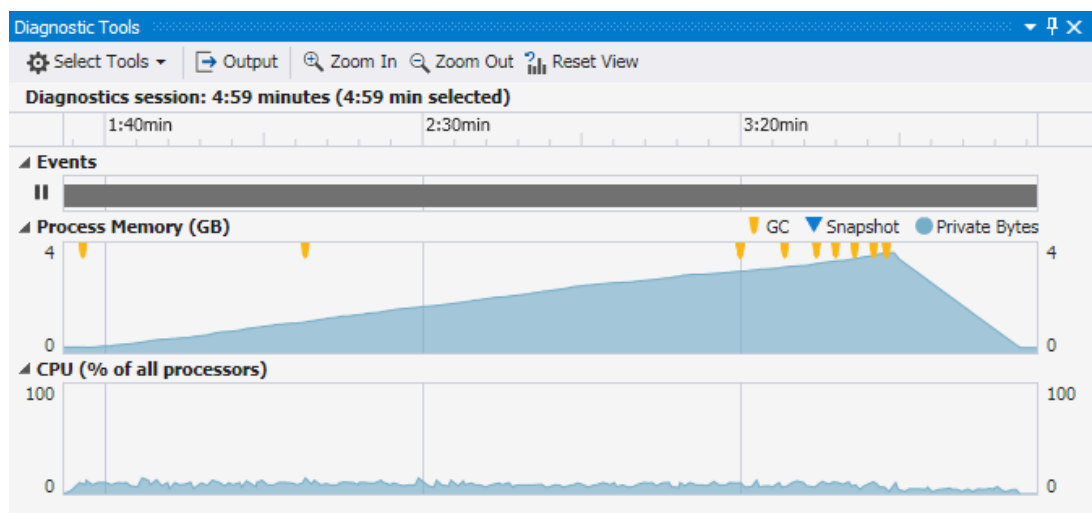
## 6.5 Problém s využitím paměti

Během testování občas docházelo k zabrání veškeré dostupné operační paměti bez ohledu na typ operačního systému (obr. 12). Pokud nebyly vykonávány další náročné požadavky na vestavěný internetový prohlížeč, tak se paměť sama v řádech desítek sekund uvolňovala, takže se aplikace i přes obtíže neočekávaně neukončovala (obr. 13). V důsledku toho měl někdy nástroj pro výběr prvku s hodnotou na internetové stránce velké zpoždění při vykonání skriptu, který po kliknutí zvýrazňuje prvek.

Při snaze nasimulovat problém jsem zjistil, že toto chování nastane, jen když probíhá abnormálně dlouhé načítání stránky v internetovém prohlížeči, který je použit jako nástroj pro výběr prvku v okně s nastavením podmínky, a během toho se spustí kontrola, při níž se internetový prohlížeč pro automatizační procesy snaží načíst stejnou stránku. I při splnění těchto podmínek je však zaplnění paměti jen málo časté. Přesnou příčinu se mi nepodařilo zjistit, a tak zatím zůstal tento problém nevyřešen. Na funkci a spolehlivost aplikace to však nemá negativní dopad.



Obrázek 12: Zobrazení využití paměti v nástroji *Sledování prostředků* při problému se zabráním celé dostupné paměti



Obrázek 13: Zobrazení nárůstu a následného uvolnění vyžití paměti v diagnostickém nástroji ve Visual Studiu při problému se zabráním celé dostupné paměti

## Závěr

Aplikace WatchCo i přes drobné nedostatky splňuje požadované funkce a vlastnosti, kterých jsem chtěl dosáhnout, a které byly uvedeny v zadání práce. Především nabízí novou funkci kombinace podmínek, snadno použitelné uživatelské rozhraní, praktické řešení portable aplikace, přehledné informování o stavu a spolehlivý chod aplikace.

Při návrhu aplikace jsem vycházel z dostupných řešení, na základě nichž jsem si určil klíčové vlastnosti mého řešení. Tyto vlastnosti pak vedly k výběru vhodných technologií pro tvorbu aplikace. Ta byla před svým dokončením testována, a některé poznatky byly do výsledného řešení zahrnuty. Zpracování ostatních poznatků bude až součástí dalšího vývoje.

S výsledkem práce jsem spokojen, a kromě něj si odnáším i mnoho cenných zkušeností a poznatků z oblasti zpracování obsahu internetových stránek. Této oblasti se chci i dále věnovat.

## Literatura

- [1] W3C. XML Path Language (XPath). [online]. 1999-11-16. [cit. 2020-03-03].  
Dostupné z: <https://www.w3.org/TR/1999/REC-xpath-19991116/>
- [2] Wachete - Monitor web changes. [online]. 2020-03-03. [cit. 2020-03-03].  
Dostupné z: <https://www.wachete.com/>
- [3] MICROSOFT. Introduction to the C# Language and the .NET Framework. [online]. 2015-07-20. [cit. 2020-03-03].  
Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [4] MICROSOFT. Introduction to WPF. [online]. 2016-11-04. [cit. 2020-03-03].  
Dostupné z: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/introduction-to-wpf>
- [5] MICROSOFT. Overview of Visual Studio. [online]. 2019-03-19. [cit. 2020-03-03].  
Dostupné z: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2017>
- [6] GREENBLATT, Marshall. README.md. [online]. 2020-03-03. [cit. 2020-03-03].  
Dostupné z: <https://bitbucket.org/chromiumembedded/cef/src/master/>
- [7] MAITLAND, Alex. CefSharp - Fast web browser for WinForms and WPF Apps. [online]. 2020-02-27. [cit. 2020-03-03].  
Dostupné z: <https://cefsharp.github.io/>



## A Obsah příloženého DVD

### **bin/**

Aplikace WATCHCO, spustitelná z umístění s povolením zápisu. Adresář obsahuje i všechny knihovny a další soubory potřebné pro bezproblémový běh aplikace.

### **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 kód textu, vložené obrázky, apod.

### **src/**

Kompletní zdrojové kódy aplikace WATCHCO se všemi potřebnými (případně převzatými) zdrojovými kódy, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí aplikace.

### **readme.txt**

Instrukce pro spuštění aplikace WATCHCO, včetně všech požadavků pro její bezproblémový běh.