|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Time Tracker** | | |
| Jan Beníšek | | |
| [místo pro vložení obrázku] | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2024/2025 | |

**Poděkování**

*Rád bych poděkoval rodině za podporu a pánům učitelům Ing. Petru Grussmannovi a Mgr. Marku Lučnému za jejich pomoc s projektem, jelikož mi pomohli se směřováním projektu.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2023

*podpis autora práce*

**ANOTACE**

Tato aplikace, nazvaná **TimeTracker**, je navržena pro monitorování uživatelských aktivit na počítači. Program sleduje aktivní okna, dobu jejich aktivity a zobrazuje jejich název, Id procesu – PID a využití systémových prostředků (CPU, RAM). Aplikace se zaměřuje na primárně uživatelsky přívětivé aplikace, jako jsou webové prohlížeče a běžné programy, a vylučuje systémové procesy, které nemají uživatelské rozhraní, lze však sledovat i ostatní procesy je potřeba vybrat jiný filtr.

TimeTracker umožňuje zobrazení aktuálně běžících aplikací v reálném čase s automatickým obnovenímv základu sekundu lze to však změnit. Poskytuje tabulkové zobrazení dat, které lze řadit podle různých kritérií, například podle času aktivace nebo využití zdrojů. Aplikace podporuje neomezený počet záznamů s možností scrollování a statickým oknem, aby byla data snadno přístupná.

Program se spouští při startu systému a je možné jej otevřít kliknutím na ikonu v systémové liště. TimeTracker je určen pro detailní analýzu a přehled využití počítače, a to jak pro individuální potřeby uživatele, tak pro účely správy času.

Celkově aplikace přispívá k lepšímu pochopení toho, jak jsou zdroje systému využívány a jaký čas je věnován konkrétním úkolům.

**ABSTRACT**

- anglický překlady

OBSAH

[Úvod 5](#_Toc185375102)

[1 Teoretická a metodická východiska 6](#_Toc185375103)

[1.1 Aplikace na sledování aktivity a zdrojů 6](#_Toc185375104)

[1.2 Princip fungování aplikace 6](#_Toc185375105)

[2 Využité technologie 7](#_Toc185375106)

[2.1 Programovací jazyk 7](#_Toc185375107)

[2.2 Knihovna pro sledování procesů 7](#_Toc185375108)

[2.3 Knihovna pro GUI 7](#_Toc185375109)

[2.4 Knihovny pro snímky obrazovky 7](#_Toc185375110)

[3 Způsoby řešení a použité postupy 8](#_Toc185375111)

[3.1 Sledování Aktivních Oken 8](#_Toc185375112)

[3.2 Sběr Systémových Informací 8](#_Toc185375113)

[3.3 Pořízení Screenshotů 9](#_Toc185375114)

[3.4 Grafické Uživatelské Rozhraní (GUI) 9](#_Toc185375115)

[3.5 Více vláknové Zpracování a Plánování Úkolů 9](#_Toc185375116)

[3.6 3.6 Problémy a Řešení 9](#_Toc185375117)

[4 Výsledky řešení, výstupy, uživatelský manuál 10](#_Toc185375118)

[Závěr 11](#_Toc185375119)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 12](#_Toc185375120)

[Seznam příloh 13](#_Toc185375121)

Úvod

Text úvodu

Problematika sledování a analýzy využití uživatelských aplikací je v současnosti klíčová pro efektivní správu času a optimalizaci pracovních procesů. Rostoucí počet aplikací a procesů běžících na osobních počítačích klade nároky na přesné monitorování systémových zdrojů a uživatelské aktivity. Tato práce se zaměřuje na vývoj a implementaci aplikace TimeTracker, která poskytuje podrobný přehled o aktivních uživatelských aplikacích, jejich čase používání a využití systémových prostředků.

Hlavním cílem práce je vytvoření nástroje, který umožní uživatelům nebo administrátorům sledovat aplikace v reálném čase, analyzovat využití CPU, RAM a pomocí automatických screenshotů spravovat jejich aktivitu efektivněji.

Práce si klade za cíl poskytnout užitečný nástroj pro sledování aktivity, který může být použit jak v pracovním, tak osobním prostředí, a zároveň inspirovat k dalšímu rozvoji podobných systémů.

# Teoretická a metodická východiska

## Aplikace na sledování aktivity a zdrojů

Existuje celá řada aplikací určených k monitorování systémových zdrojů a aktivit, které uživatelům umožňují sledovat využití procesoru (CPU), operační paměti (RAM) a jednotlivé běžící procesy. Mezi nejběžnější nástroje patří Správce úloh (Windows) a Monitor aktivity (macOS), které poskytují základní přehled o systému. Pokročilejší aplikace, jako například Process Explorer nebo htop, umožňují podrobnější analýzu výkonu a detailní monitoring jednotlivých procesů. Tyto nástroje obvykle nesledují čas strávený v konkrétních aplikacích a neobsahují funkce pro zaznamenávání aktivit uživatele.

## Princip fungování aplikace

Moje aplikace se zaměřuje na sledování času stráveného v jednotlivých aplikacích a využití systémových zdrojů (CPU, RAM) v reálném čase. Aplikace pravidelně (například každé 3 sekundy) načítá aktuální data o běžících procesech a jejich zatížení systémovými prostředky. Na základě těchto údajů identifikuje aktivní aplikaci, zaznamenává její dobu aktivace a dobu trvání používání. Systém automaticky filtruje procesy bez uživatelského rozhraní a systémové služby, což zajišťuje přehlednost dat, ale tuto funkci lze upravit. Uživatel může seřadit data podle různých kritérií, jako je čas aktivace, zatížení CPU nebo délka používání aplikace. Dále je k dispozici funkce pro pořizování screenshotů také spuštěna dle času který správce nebo uživatel určí, která umožňuje zachytit, co uživatel v daný čas na obrazovce dělá, což zvyšuje efektivitu při sledování a správě produktivity.

# Využité technologie

## Programovací jazyk

Pro vývoj aplikace byl zvolen programovací jazyk Python, který je ideální pro práci s ope-račními systémy a umožňuje rychlý vývoj aplikací s přehledným kódem. Python poskytuje širokou podporu pro práci s operačními prostředími Windows i macOS, což bude v budoucnu klíčové pro dosažení multiplatformní kompatibility aplikace.

## Knihovna pro sledování procesů

Pro monitorování systémových procesů a získávání informací o využití zdrojů byla využita knihovna psutil. Tato knihovna umožňuje snadné získání informací o běžících procesech, zatížení CPU, využití paměti RAM a dalších systémových zdrojích. Knihovna psutil je velmi efektivní a umožňuje načítat data o procesech v reálném čase, což je nezbytné pro správné fungování aplikace.

## Knihovna pro GUI

Pro vývoj grafického uživatelského rozhraní byla použita knihovna PyQt5, která je založena na frameworku Qt a poskytuje široké možnosti pro tvorbu moderních desktopových aplikací. V aplikaci byla využita především pro správu hlavního okna, systémové lišty a interaktivního menu, což umožňuje ovládat aplikaci přímo z ikony v liště. PyQt5 také umožňuje vytváření tabulek a ovládacích prvků, což bylo klíčové pro zobrazení informací o systémech a procesech, a umožnilo snadnou integraci s dalšími moduly, jako je psutil pro sledování aktivních oken a využití systémových prostředků.

## Knihovny pro snímky obrazovky

Funkce pro pořizování screenshotů byla implementována pomocí knihovny pyscreenshot, která je součástí Pythonu pro snadné pořizování snímků obrazovky na různých operačních systémech. Tato knihovna umožňuje jednoduchý přístup k funkcionalitě pro zachycení celého obrazovky nebo aktivních oken a následné uložení těchto snímků v různých formátech, včetně PNG. V aplikaci byla použita pro pořízení screenshotů každých 10 sekund a jejich uložení do specifikované složky na disku.

# Způsoby řešení a použité postupy

## Sledování Aktivních Oken

Pro sledování aktuálně aktivních oken v systému macOS je využita třída ActivityTracker. Tento modul využívá NSWorkspace a NSWorkspaceDidActivateApplicationNotificati-on, což jsou součásti Cocoa frameworku, který je dostupný pouze na macOS. Pomocí těch-to technologií aplikace monitoruje aktivitu aplikací na pozadí a zaznamenává změny mezi aktivními okny.

* Signál windowChanged: Třída ActivityTracker emituje signál, který informuje o změně aktivního okna. Tento signál je následně zachycen v hlavní třídě aplikace (Menu), kde se zaktualizují informace o aktivním okně.
* Sledování Změn Okna: Jakmile dojde ke změně aktivního okna, aplikace zazname-ná čas poslední aktivace a dobu trvání aktivity, což je klíčová funkcionalita pro zá-znam aktivních oken.

## Sběr Systémových Informací

V aplikaci se používá knihovna psutil k získávání informací o procesech běžících na systému, jako jsou:

* CPU a RAM: Monitorujeme využití CPU a paměti pro každý proces pomocí psutil.process\_iter(), která vrací informace o běžících procesech včetně jejich názvů, PID a využití systémových prostředků.
* Filtrace Uživatelských Procesů: Aplikace filtruje systémové procesy (např. kernel\_task, launchd), které nemají žádnou interakci s uživatelem, a zobrazuje pouze procesy spojené s běžnými aplikacemi. Tato filtrace se provádí v metodě is\_user\_process.
* Zobrazení v Tabulce: Informace o procesech jsou zobrazeny v tabulce, která je součástí GUI. Každý řádek zobrazuje název aplikace, její poslední aktivaci, dobu aktivace, CPU a RAM využití, a PID procesu.

## Pořízení Screenshotů

Pro pořízení screenshotů aplikace využívá knihovnu pyscreenshot. Pořízení screenshotu je naplánováno pomocí QTimer, který každých 300 sekund spustí metodu take\_screenshot.  
Čas za který je metoda spuštěna lze změnit

* Ukládání Screenshotů: Screenshoty jsou ukládány do výchozí složky (nebo do specifikované složky) s názvem souboru obsahujícím časovou značku.
* Zabezpečení Cesty: Před pořízením screenshotu je zajištěno, že složka pro uložení obrázků existuje. Pokud ne, je vytvořena pomocí metody os.makedirs.

## Grafické Uživatelské Rozhraní (GUI)

Pro GUI je použita knihovna **PyQt5**. Aplikace vytváří hlavní okno s tabulkou, která zobrazuje aktivní procesy a jejich systémové informace. Dále využívá **QSystemTrayIcon** pro zobrazení ikony v systémové liště, kde je k dispozici kontextové menu.

* Tray Ikona a Menu: Aplikace běží na pozadí, a to buď jako minimalizované okno, nebo s aktivním oknem. K dispozici je možnost otevřít hlavní okno nebo aplikaci ukončit.
* Aktualizace Tabulky: Tabulka se automaticky aktualizuje každou sekundu, aby zobrazovala aktuální data o procesech. Řazení dat je povoleno, což umožňuje uživatelskou interakci s tabulkou, jako je třídění podle jména procesu, CPU, RAM nebo času poslední aktivace.

## Více vláknové Zpracování a Plánování Úkolů

Aplikace používá QTimer k plánování pravidelných úkolů, jako je aktualizace informací o procesech nebo pořizování screenshotů. Tento přístup umožňuje, aby aplikace běžela hladce na pozadí, aniž by blokovala hlavní vlákno pro interakci s GUI.

## 3.6 Problémy a Řešení

# Výsledky řešení, výstupy, uživatelský manuál

Text čtvrté kapitoly

* výčet splněných a nesplněných cílů, obrázky (schémata, vzorce apod.) z finálního provedení, prokázání funkčnosti, výsledné parametry výrobku apod.
* podle zaměření a charakteru práce je třeba volit vhodný nadpis pro tuto kapitolu, je samozřejmě možné i rozdělení na více kapitol (např. Uživatelské rozhraní internetové aplikace; Administrace internetové aplikace…)

# Závěr

Text závěru

* povinná část,
* shrnuje výsledky, hodnotí splnění cíle práce, uvádí možnost uplatnění řešení v praxi a nastínění případných dalších budoucích vylepšení
* kapitola se nečísluje (stejné jako úvod)

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] BOHMAN, Ludvík. Zákon o pojistné smlouvě. Praha: Linde Praha a. s., 2004. 381 s. ISBN80-7201-504-4

[2] DUCHÁČKOVÁ, Eva. Principy pojištění a pojišťovnictví. 3. aktualizované vydání. Praha: Ekopress 2009. 224 s. ISBN 978-80-86929-51-4

[3] KUBALA, Petr. Planetární dvojcata - Věda a technika (Český rozhlas) [online].   
Č. 2000-2008, poslední revize 19. 3. 2008 [cit. 2008-03-20].  
<http://www.rozhlas.cz/veda/vesmir/\_zprava/435849>.

[4] KULDOVÁ, O., FLEISCHMANNOVÁ, E. Metodická příručka k technice administrativy a obchodní korespondence. 1.vyd. Praha: Fortuna 1998. 111 s.   
ISBN 80-7168-574-7. Kapitola 6, Metody nácviku psaní hmatovou metodou,   
s. 28-29.

[5] VLACH, J. JE Temelín a zásobování teplem. Energetika, 2001, roč. 51, č. 3, s. 84 -85. ISSN 0375-8842.

* musí zahrnovat všechny prameny, knihy, internetové odkazy a další studijní podklady, z nichž jsme čerpali;
* kapitola se nečísluje a zde končí číslování stránek práce;
* jednotlivé publikace se uvádějí v abecedním pořadí podle příjmení autorů a iniciál jeho jména, který se píše za čárkou;
* příjmení autora se píše velkými písmeny;
* název publikace se zvýrazňuje kurzívou;
* jestliže jsou uvedeni více než tři autoři, je možné vypsat hlavního autora s poznámkou „a kol.“(a kolektiv).

Seznam příloh

č. 1 Titulní list

č. 2 Čestné prohlášení

č. 3 Poděkování

Nepovinná část – pokud nemáte žádné přílohy ke své práci, tuto část odstraňte!

* Přílohy se zařazují na konec práce.
* Jsou to texty, obrázky, grafy, tabulky, které by přímo v textu byly zbytečně detailní, ale mají být po ruce k dokreslení východisek i výsledku řešení.
* Jsou číslovány a v textu se na ně může odkazovat.
* Před první přílohu se umisťuje seznam příloh.
* Každá příloha je označena číslem - např. Tabulka č.. 1, Schéma č. 2, Obrázek č. 3.
* Každá tabulka by měla mít i vlastní název, který stručně vystihuje její obsah.
* (Tabulka č. 1 Zakázky stavebních prací v roce 2009-2010).
* Pokud je z tabulky vytvořen graf, umístíme jej na stejné stránce jako tabulku.

**Příloha č. 1: Titulní list**