

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему: МОНІТОРИНГ ТА ОБЛІК ЗАВАНТАЖЕННЯ КОМП'ЮТЕРНИХ
РОБОЧИХ МІСЦЬ

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-20 -1,
спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Синька Микити
Максимовича

Науковий керівник


(підпис)

Погореловська І. Д.,
к.е.н., доцент

Керівник освітньо-професійної
програми


(підпис)

Омельчук А.А., к.т.н.

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем, протокол від «22» травня 2023 № 15

Завідувач кафедри


(підпис)

Філоненко М. М., к.ф.-м.н., доцент

Реєстрація № 30 27.05.2024
(дата)


(підпис відповідальної особи)

Результати перевірки до захисту
(до захисту, на доопрацювання)

22.05.2024 
(дата) (підпис)

Результат захисту 4 (добре)
(оцінка)

06.06.2024 
(дата) (підпис)

Голова комісії:


(підпис)

В.М. Франчук
(ініціали, прізвище)

Члени комісії:


(підпис)

М.М. Філоненко
(ініціали, прізвище)

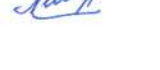

(підпис)

С.М. Грищенко


(підпис)

В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	21		06.06.2024 Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 
2. Дотримання вимог щодо оформлення	від 0 до 10	7		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	35		

Оцінювання кваліфікаційної роботи науковим керівником - 36 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
41	C	4 (добре)	06.06.2024	Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій

Кафедра комп'ютерних та інформаційних технологій і систем

Галузь знань: 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні управляючі системи та технології (за галузями)»

Затверджую

завідувач кафедри

 Філоненко М. М.

«20» лютого 2023

Завдання на кваліфікаційну роботу

здобувача вищої освіти

Синьку Микиті Максимовичу

1. Тема кваліфікаційної роботи: Моніторинг та облік завантаження комп'ютерних робочих місць

затверджена наказом ректора від 20.10. 2023 №1411

2. Строк здачі закінченої роботи: 20 травня 2024

3. Цільова установка: застосування набутих знань, умінь у забезпеченні моніторингу завантаженості комп'ютерних робочих місць

Мета роботи: розробка програмного забезпечення для відстеження часу роботи працівників.

Об'єкт дослідження: завантаженість комп'ютерних робочих місць, засоби моніторингу стану завантаження комп'ютерних робочих місць.

Предмет дослідження: функціональність систем моніторингу завантаження комп'ютерних робочих місць

4. Вихідні дані до роботи: методи, підходи та програмно-апаратні засоби здійснення моніторингу завантаження комп'ютерних робочих місць.

5. Перелік графічного матеріалу: структура програмного забезпечення, діаграмами стану завантаження комп'ютера, копії екранів, презентація роботи

6. Зміст кваліфікаційної роботи: Розділ 1. Теоретичні аспекти моніторингу стану завантаження комп'ютерних робочих місць. 1.1. Цілі і завдання систем моніторингу діяльності персоналу. 1.2. Види програмного забезпечення моніторингу стану завантаження комп'ютерних робочих місць. Розділ 2. Аналіз функціональності інструментів моніторингу комп'ютерних робочих місць. 2.1. Формування критеріїв вибору програмного забезпечення для моніторингу персоналу. 2.2 Порівняльна характеристика програмних засобів моніторингу. Розділ 3. Розробка системи контролю та обліку робочого часу 3.1. Вибір інструментів розробки. 3.2. Програмна реалізація задачі контролю та обліку робочого часу

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури ДР	до 15.10.2023	15.10.2023
2	1. Вступ та I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	01.01.2024	01.04.2024
3	II розділ (аналіз експериментальної частини і методики досліджень)	03.02.2024	10.04.2024
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	10.03.2024	30.04.2024
5	Висновки	01.04.2024	06.05.2024
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	01.05.2024	25.05.2024
7	Отримання відгуку керівника	15.05.2024	30.05.2024
8	Отримання зовнішньої рецензії	17.05.2024	31.05.2024

8. Дата видачі завдання 20.10.2023

9. Керівник кваліфікаційної роботи



Погореловська І

10. Гарант
освітньо-професійної програмиОмельчук А.А.,11. Завдання до виконання отримав
здобувач вищої освітиСинько М.М.

АНОТАЦІЯ

Моніторинг та облік завантаження комп'ютерних робочих місць

Мета роботи: розробка програмного забезпечення для відстеження часу роботи працівників.

Завдання, що були вирішені при виконанні роботи:

- визначено цілі та завдання систем моніторингу діяльності персоналу ;
- проаналізовано види програмного забезпечення моніторингу стану завантаження комп'ютерних робочих місць;
- сформовано критерії вибору програмного забезпечення для моніторингу персоналу;
- виконано порівняльний аналіз програмних засобів моніторингу;
- здійснено вибір інструментів для розробки системи контролю робочого часу;
- **розроблено програмне забезпечення для реалізації задачі контролю та обліку робочого часу.**

Об'єкт дослідження: завантаженість комп'ютерних робочих місць. засоби моніторингу стану завантаження комп'ютерних робочих місць.

Предмет дослідження: функціональність систем моніторингу стану завантаження комп'ютерних робочих місць

Інформаційна база дослідження: методи, підходи та програмно-апаратні засоби здійснення моніторингу завантаження комп'ютерних робочих місць.

За результатами дослідження було сформульовано висновки про необхідність постійного моніторингу завантаження комп'ютерних робочих місць, а також запропоновано програмні засоби, які розширяють функціональні можливості системи обліку завантаження комп'ютерних робочих місць.

Одержані результати можуть бути використані при формуванні систем моніторингу завантаження комп'ютерних робочих місць, в освітньому процесі при викладанні навчальної дисциплін “Інформаційні технології проєктного менеджменту”.

Дипломна робота містить 55 сторінок, 16 рисунків, 1 таблиця і список використаних джерел з 45 найменувань.

Ключові слова: системи моніторингу, завантаження комп'ютерних робочих місць, тайм-трекери, програмування

Рік виконання дипломної роботи: 2023-2024

Рік захисту роботи: 2024.

ABSTRACT

Monitoring and development of computer workspaces

Meta robots: development of software to support the work hours of healthcare workers.

Legends that were believed during the Viconn's work:

- the purpose of the system for monitoring the activities of personnel has been determined;
- analyzed the types of software monitoring that will become increasingly popular in computer workspaces;
- criteria for the selection of software for monitoring personnel have been formed;
- final analysis of software monitoring features;
- a selection of tools has been made for developing a working hour control system;
- a security program has been developed to implement the task of monitoring and working hours.

Object of investigation: the importance of computer workspaces. The methods of monitoring will become more important in computer workspaces.

Subject of research: the functionality of monitoring systems will become increasingly important for computer workspaces

Information base for research: methods, approaches and hardware and software for monitoring the use of computer workstations.

Based on the results of the investigation, conclusions were formulated about the need for continuous monitoring of computer activity in the workplace, as well as program options to expand the functional capabilities of the computer environment There are many workers among them.

These results can be obtained when forming systems for monitoring the use of computer work stations, in the process of completing the initial discipline "Information Technologies and Project Management".

The thesis contains 55 pages, 16 figures, 1 table and a list of short stories with 45 names.

Keywords: monitoring systems, computer monitoring systems, time trackers, programming

Rik Vikonanny diploma work: 2023-2024

Rick zakhistu roboti: 2024.

ЗМІСТ

Вступ.....	2
Розділ 1. Теоретичні аспекти моніторингу стану завантаження комп'ютерних робочих місць.	4
1.1 Цілі і завдання систем моніторингу діяльності персоналу.....	4
1.2. Види програмного забезпечення моніторингу стану завантаження комп'ютерних робочих місць	6
Розділ 2 . Аналіз функціональності інструментів моніторингу комп'ютерних робочих.....	16
2.1Формування критеріїв вибору програмного забезпечення для моніторингу персоналу.....	16
2.2Порівняльна характеристика програмних засобів моніторингу	13
Розділ 3 . Розробка системи контролю та обліку робочого часу	21
3.1. Вибір інструментів розробки.	21
3.2 Програмна реалізація задачі контролю та обліку робочого часу.....	29
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	37
Додаток.....	39
Додаток А.....	40

ВСТУП

Головною проблемою сучасних компаній є не раціональний розподіл часу співробітниками. Дуже складно протягом робочого тижня контролювати навантаження працівників. Спиратись на звіти заповнені співробітниками безглуздо. Оскільки вони самі їх заповняють вони можуть бути не достовірними, приховувати періоди прокрастинації та занадто завищувати їх продуктивність. Саме тому найкращий варіант для відстеження достовірного часу активної праці персоналу та аналізу їх роботи є застосування спеціального програмного забезпечення для моніторингу їх завантаженості та роботи.

Актуальність теми: на ринку програмного забезпечення є безліч різноманітних рішень для моніторингу діяльності працівників, від не дуже якісних безкоштовних програм з ризиками витоку конфіденційних даних до більш технологічних систем відстеження та обліку робочого часу з функціями запобігання витоків персональних даних. Однак не всі ці програми наявні на ринку є достатньо гнучкими до потреб клієнтів, у тому числі і в питаннях захисту конфіденційної інформації.

Мета кваліфікаційної роботи: розробка програмного забезпечення для відстеження часу роботи працівників.

Виходячи з поставленої мети, були визначені та вирішені наступні завдання:

- визначено цілі та завдання систем моніторингу діяльності персоналу ;
- проаналізовано види програмного забезпечення моніторингу стану завантаження комп'ютерних робочих місць;
- сформовано критерії вибору програмного забезпечення для моніторингу персоналу;
- виконано порівняльний аналіз програмних засобів моніторингу;
- здійснено вибір інструментів для розробки системи контролю робочого часу;

- розроблено програмне забезпечення для реалізації задачі контролю та обліку робочого часу.

Об'єктом дослідження є: завантаженість комп'ютерних робочих місць. засоби моніторингу стану завантаження комп'ютерних робочих місць.

Предмет дослідження: функціональність систем моніторингу стану завантаження комп'ютерних робочих місць.

Методи дослідження: системний аналіз, який застосовувався при роботі з інформаційними джерелами за темою дослідження; методи порівняння та групування були застосовані при дослідженні функціональних можливостей засобів моніторингу; методи оцінювання застосовувалися при виборі інструментів для розробки програмного забезпечення контролю часу роботи працівників; метод узагальнення застосовувався при формуванні висновків.

Інформаційна база дослідження: методи, підходи та програмно-апаратні засоби здійснення моніторингу завантаження комп'ютерних робочих місць.

Науково-практична значимість одержаних результатів: кваліфікаційна робота містить як теоретичні аспекти проблем моніторингу завантаженості комп'ютерних робочих місць, так і конкретні результати її виконання — розроблено систему контролю обліку робочого часу.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ МОНІТОРИНГУ СТАНУ ЗАВАНТАЖЕННЯ КОМП'ЮТЕРНИХ РОБОЧИХ МІСЦЬ.

1.1 Цілі і завдання систем моніторингу діяльності персоналу.

Програмне забезпечення для моніторингу співробітників - це система, яка може відстежувати та аналізувати діяльність співробітників на їхніх комп'ютерах. Його популярність зростає з початку 2000-х років завдяки розвитку комп'ютерних технологій, а пандемія Covid-19 у 2020 році призвела до стрімкого зростання популярності. Згідно з нещодавнім дослідженням Instant Office, 78% сучасних роботодавців відстежують своїх співробітників за допомогою різних інструментів [15].

Зазвичай програмне забезпечення для моніторингу працівників складається з агента, який працівники встановлюють на свої пристрої, та інформаційної панелі моніторингу, яка дозволяє менеджерам отримувати журнали та звіти.

Програмне забезпечення для моніторингу співробітників виступає в ролі невидимого наглядача. Від натискання клавіш до відвіданих веб-сайтів, ws системи збирають інформацію практично про все, що людина робить на комп'ютері в робочий час (рис.1.1).

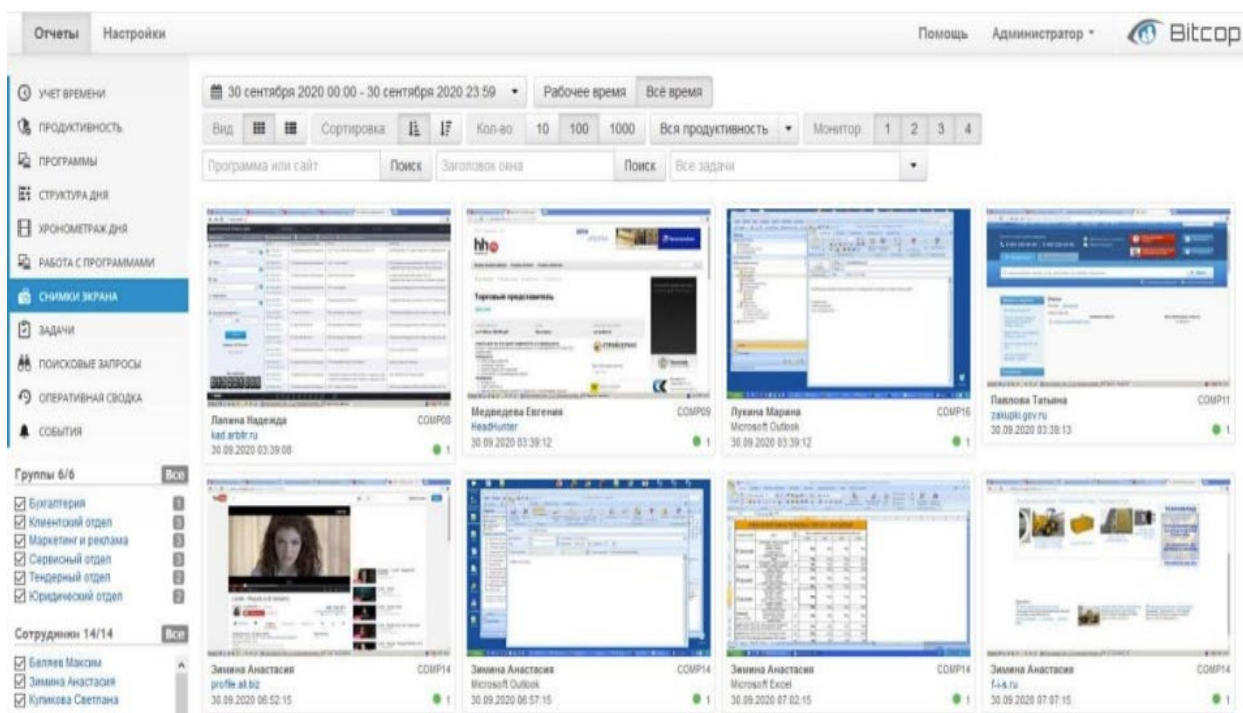


Рис. 1.1. Приклад роботи системи моніторингу

Більш досконале програмне забезпечення для моніторингу співробітників виходить далеко за рамки простого збору даних. Вони надають готові звіти про продуктивність кожного працівника та розрахунки заробітної плати, а також суттєво економлять час керівника.

Головне призначення такого роду програмного забезпечення - моніторинг продуктивності працівників. Скільки часу вони витрачають на проект, а скільки - на кумедні котячі відео на Youtube? Чи залучені вони в робочий процес, чи роблять перерви на каву кожні 10 хвилин? Чи гуглять вони інформацію, пов'язану з роботою, або дешеві квитки в Лас-Вегас на вихідні?

Навіть в невеликих фірмах з невеликим штатом співробітників керівник не завжди може бути впевнений, що працівник доцільно використовує свій час, не запізнюється на роботу чи не йде раніше, або не тратить цей час на інтернет. При великому штаті працівників це питання стає гострішим. І за великого обсягу працівників зникає можливість їх візуального контролю. З'являються ризики не доцільного використання майна компанії, що може призвести навіть до викрадення конфіденційної інформації. Додатки для моніторингу співробітників покликані відповісти саме на такі питання.

Крім виявлення прогульників, програмне забезпечення для моніторингу співробітників виконує ще кілька менш очевидних, але не менш важливих завдань.

Перш за все, ці системи можуть допомогти виявити працівників, які відчувають труднощі: непосильне навантаження (завдання), у виконанні якого вони не мають достатнього досвіду як молоді фахівці, вони можуть не наважуватися розповісти керівнику про труднощі, з якими стикаються. Виявлення цих проблем і ретельна робота над їх усуненням не тільки підвищить продуктивність, але й поліпшить клімат в офісі та ваш імідж як керівника.

Друга, не настільки очевидна причина спробувати програмне забезпечення для моніторингу співробітників, - це контроль ресурсів компанії. Чи зросли останнім часом витрати на папір для принтера? Чи користуються співробітники новим нещодавно придбаним програмним забезпеченням?

Дізнайтеся про це можна за допомогою звітів. Вони покажуть, як часто співробітники працюють в тій чи іншій програмі або використовують офісні принтери для особистих потреб.

Першокласне програмне забезпечення для моніторингу співробітників оснащено модулями захисту даних. Вони запобігають доступу, копіюванню або пересиланню конфіденційної інформації та попереджають адміністратора, якщо з'являється щось підозріле. Маючи під рукою такого всезнаючого охоронця, можна не турбуватися про витік інформації або несанкціонований доступ до корпоративних даних.

Це лише декілька проблем, у вирішенні яких можуть допомогти інструменти моніторингу.

Основними завданнями контролю діяльності співробітників є: “моніторинг ефективності, що допоможе зменшити навантаження на працівників з великою кількістю задач, та збільшити навантаження на працівників, які не задіюють себе повністю; покращення постановки завдань, що спиятиме кращому розумінню завдань, які необхідно виконати; відстеження часу перебування працівників на робочому місці, що допоможе виявити працівників, які зловживають запізненими та обідніми перервами” [9].

1.2. Види програмного забезпечення моніторингу стану завантаження комп'ютерних робочих місць

Програмне забезпечення для моніторингу працівників поділяється на три типи: “додатки відстеження часу; програмне забезпечення моніторингу активності; інструменти запобігання втрат даних” [3].

Додатки для відстеження часу забезпечують:

- автоматизований облік робочого часу та розрахунку заробітної плати;
- реєстрацію часу входу та виходу з системи, періодів бездіяльності працівника та час, витрачений на виконання кожного завдання;
- визначення кількості часу, яку співробітники витрачають на виконання конкретного завдання за проєктом; Точні дані відстеження також

життєво важливі для виставлення рахунків клієнтам, якщо ви працюєте за контрактом або на основі погодинної плати.

- генерування звітів у режимі реального часу автоматично, що виключає ризик обману або людської помилки.

Система обліку робочого часу – це один з оптимальніших способів контролю регламенту роботи та продуктивності підопічних. Ця система є чудовим інструментом для оцінювання працівників, та допомагає в вирішенні кадрових дилем, таких як скорочення та підвищення в посадах.

Ці тайм-трекери сприяють підвищенню продуктивності, мотивують співробітників до роботи та показують, куди витрачаються гроші [3].

Призначення програмного забезпечення моніторингу активності - фіксувати практично всі дії, що виконує працівник на офісному комп'ютері.

Ці системи забезпечують:

- реєстрацію натискання клавіш та події буферу обміну;
- виконання скріншотів;
- відстеження інтернет-активності працівника (включаючи соціальні мережі, чати);
- облік часу активної/пасивної роботи;
- стеження за зовнішніми пристроями (принтери, USB-накопичувачі);
- збирання даних про активність та генерування звітів про діяльність кожного працівника офісу у вигляді діаграм та графіків.

Програми моніторингу вищого рівня дозволяють спостерігати за працівниками у режимі реального часу, транслуючи відео з їхніх екранів і веб-камер (чи в записі)

Звіти дають цінну інформацію про продуктивність і рівень залученості, допомагають виявити неефективні схеми роботи і співробітників, які не справляються зі своїми обов'язками, заощадити ресурси компанії і багато іншого.

За умови ретельного та етичного використання, програмне забезпечення для відстеження активності може бути надійним партнером у розвитку бізнесу [7].

Інструменти запобігання втрати даних – системи DLP (data loss prevention software, DLP) призначені для визначення потенційних витоків даних та інших загроз шляхом моніторингу, виявлення та блокування даних, що використовуються, перебувають у русі чи у стані спокою [8].

Системи DLP:

- здійснюють пошук поширених типів конфіденційних даних, наприклад, банківські реквізити, номери соціального страхування тощо.
- виконують перевірку вмісту та контекстний аналіз даних, що надсилаються е-поштою чи месенджерами, використовуються працівником на своєму пристрої або зберігаються на сервері компанії.
- слугують для забезпечення дотримання компанією відповідних нормативних актів та захисту інформації, що ідентифікує особу.

Система DLP включає аналітичні методи, наприклад, точне або часткове зіставлення файлів, аналіз на основі правил і категорій, аналіз лексики, машинне навчання та статистичний аналіз.

Компанії використовують системи DLP для запобігання порушення даних, їх втрати, витоку конфіденційної інформації.

Отже, системи моніторингу та обліку витрат часу працівників має забезпечувати контроль перебування осіб на робочому місці, у т.ч. облік запізнення, а також здійснювати контроль кібербезпеки, тобто моніторити витoki інформації.:

РОЗДІЛ 2 . АНАЛІЗ ФУНКЦІОНАЛЬНОСТІ ІНСТРУМЕНТІВ МОНІТОРИНГУ КОМП'ЮТЕРНИХ РОБОЧИХ

2.1 Формування критеріїв вибору програмного забезпечення для моніторингу персоналу

При прийнятті рішення про придбання та використання того чи іншого інструменту для моніторингу персоналу необхідно враховувати: «мету здійснення моніторингу; функціонал системи; зручність користувацького інтерфейсу та його адаптивність; рівень безпеки; мультиплатформність; підтримка та технічна допомога; вартість системи» [5].

Кожне програмне забезпечення для моніторингу співробітників призначене для певних цілей або поєднує в собі кілька суміжних: захист від витоку даних, відстеження продуктивності, контроль відвідуваності тощо. Чітко визначена мета моніторингу дозволяє одразу відкинути рішення, які їй не відповідають.

Важливим кроком привиборі програмного забезпечення є більш детальний розгляд функцій, які пропонують ці інструменти. Деякі програми мають лише кілька базових опцій, таких як скріншоти, запис клавіатури та моніторинг активності в Інтернеті. Інші першокласні системи, здається, реєструють кожне клацання миші та кожен рух працівника за комп'ютером. Природно припустити, що чим більше - тим краще, але при цьому можна переплатити за функціонал, який ніколи не буде використаний. Замість того, щоб витратити гроші і час даремно, треба продумати, які саме функції принесуть найбільш цінну інформацію, і вибрати систему, яка забезпечить саме необхідні функції.

Важливим є також такий критерій як “зручність користувацького інтерфейсу та адаптивність”. Система моніторингу співробітників повинна легко розгортатися на всіх офісних комп'ютерах та мати інтуїтивно зрозумілий інтерфейс, забезпечувати чітку візуалізацію даних, наприклад, кольорові діаграми та графіки. Ви повинні надавати користувачу можливість швидко та ефективно орієнтуватися, не ламаючи голову над тим, що означає ця діаграма і де знайти звіт про продуктивність.

Моніторинг працівників повинен бути масштабованим та адаптованим. Додавання або видалення працівника з системи не повинно бути складним і трудомістким. Гнучкість налаштувань також є великою перевагою, цей процес має бути простим і зрозумілим.

Програмне забезпечення для стеження за працівниками збирає та зберігає великі обсяги інформації, у тому числі конфіденційні дані (паролі, фінансові дані). Тому важливим критерієм їх вибору має бути рівень безпеки. Перед тим, як використовувати будь-яке програмне забезпечення для моніторингу співробітників, необхідно переконатися, що зібрані дані безпечно зберігаються.

Важливим критерієм вибору системи для моніторингу є мультиплатформність. Оскільки співробітники можуть працювати на Mac, Windows чи iPhone, то щоб заощадити на ліцензіях, необхідно обирати програмне забезпечення, яке підтримує кілька платформ. Підтримка та технічний супровід компанією-розробником також мають бути враховані при виборі програмного забезпечення для моніторингу. Необхідно перевірити, чи передбачають ліцензійні умови безкоштовну технічну допомогу, чи будь-яка підтримка є взагалі відсутньою.

Моніторинг продуктивності співробітників має важливе значення для розвитку компанії. Тому розумно підібрана система моніторингу діяльності співробітників допоможе зробити цей процес простим та ефективним.

Вартість програмного забезпечення для моніторингу персоналу є також важливим критерієм вибору.

Врахування зазначених критеріїв допоможе обрати інструмент, який принесе найбільшу користь для компанії та її бізнесу.

Необхідно також зазначити, що використання програмного забезпечення для моніторингу працівників надає багато переваг керівництву, водночас воно також може створити і певні проблеми в компанії. Найбільшою проблемою є зниження довіри та погіршення морального стану працівників. Співробітники можуть сприймати таке програмне забезпечення як відсутність довіри з боку роботодавців, що може викликати невдоволення та знизити моральний дух компанії. Нікому не подобається відчувати постійний нагляд, і моніторингове ПЗ може розцінюватися як порушення конфіденційності.

Іншим потенційним недоліком є час, необхідний для аналізу даних, зібраних програмним забезпеченням. Хоча ці програми надають багато інформації, яка може допомогти компаніям покращити безпеку та впорядкувати процеси, для розшифровки та розуміння найважливіших даних може знадобитися додаткова робота.

Впровадження моніторингового ПЗ також може призвести до ризиків конфіденційності та безпеки. Хоча ці програми призначені для захисту мережі, є ймовірність зіткнутися з приватними даними співробітників, такими як медичні записи чи дані банківських рахунків. Щоб зменшити цей ризик, необхідно чітко пояснити працівникам, до яких даних вони можуть мати доступ на корпоративних пристроях, а до яких – ні [45].

Отже, при прийнятті рішення щодо придбання та використання того чи іншого інструменту для моніторингу персоналу необхідно враховувати:

- мету здійснення моніторингу;
- функціонал системи;
- зручність користувацького інтерфейсу та його адаптивність;
- рівень безпеки;
- мультиплатформність;
- підтримка та технічна допомога;
- вартість системи.

2.2 Порівняльна характеристика програмних засобів моніторингу

Ринок програмного забезпечення пропонує велику кількість різних систем моніторингу ефективності співробітників, як від відомих розробників у галузі безпеки (Symantec, Trend Micro), так і від великих компаній, які спеціалізуються лише на цьому класі продуктів. Згідно з дослідженням Hexa Reports "Employee Monitoring Software Market Size, Share, Industry Trends, Analysis And Forecast, 2022", основними гравцями на світовому ринку є: Teramind, Veriato 360 (SpectorSoft), SentryPC, NetVizor, InterGuard, Work Examiner, StaffCop, OsMonitor, iMonitor EAM, Pearl Echo.Suite, WorkTime, Symantec, Trend Micro, BetterWorks, Monitis, Quest Foglight, StackDriver [23].

Усі програмні забезпечення для моніторингу персоналу, як і будь-який інший B2B-продукт, створюються виключно з комерційною метою, тобто вони є платними.

Однак деякі розробники пропонують користувачам безкоштовні (Freemium) версії своїх програм з обмеженою функціональністю, яка може бути достатньою для вирішення базових завдань автоматизації контролю співробітників.

У подальшому в роботі будуть проаналізовані можливості програмних проєктів: Kickidler, ActivTrak, Hubstaff, DeskTime Pro, Toggl, Screenshot Monitor, Tmetric.

2.2.1 Опис програмного забезпечення Kickidler

Kickidler — система нового покоління для обліку робочого часу працівників (рис. 2.1). Ця програма зроблена для автоматизації на підприємстві управління персоналом.

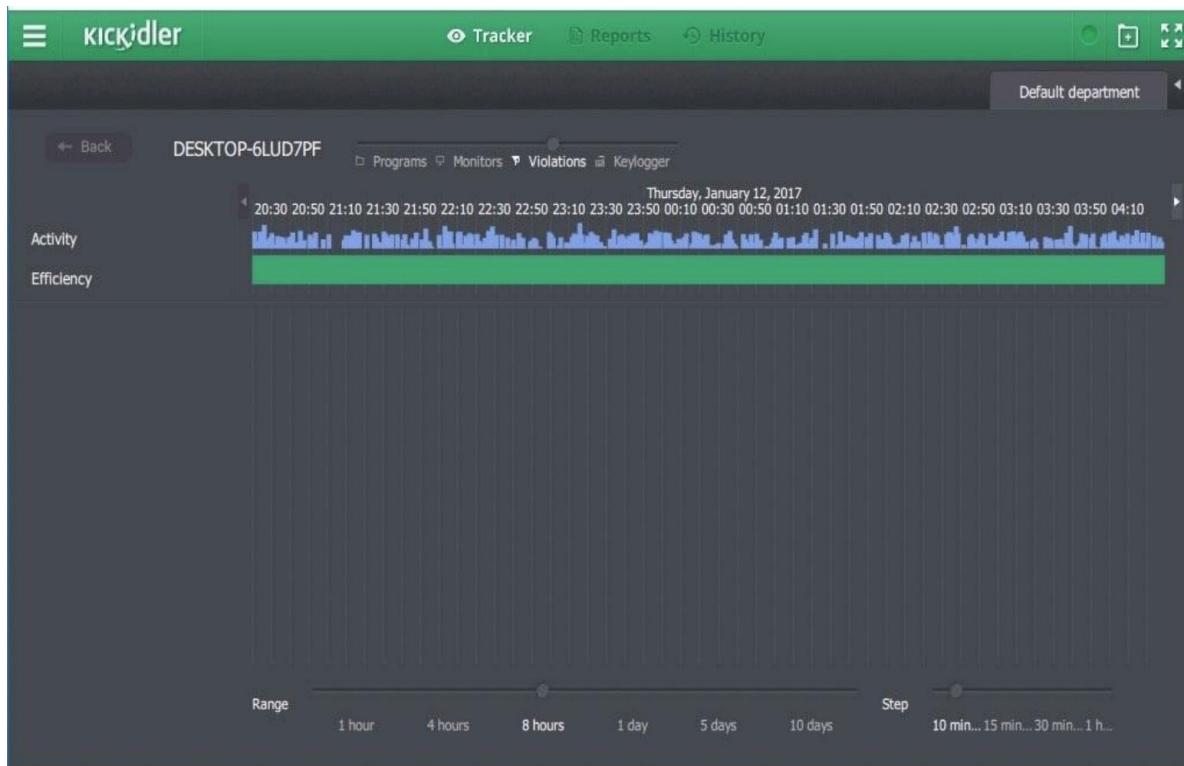


Рис.2.1. Вікно системи Kickidler

Ця програма має широкий спектор різноманітних інструментів для відстеження діяльності працівників та знаходження порушників робочого розпорядку

Система нагляду робочого часу Kickidler надає користувачам безкоштовні функції, яка може обробляти інформацію з великої кількості користувачів. Також компанія надає пробну двох тижневу підписку з повним асортиментом функцій для наглядом за працівниками

Функції безкоштовного тарифу:

Віддалений моніторинг робочих машин персоналу (перегляд екрану працівників у потоці, знімки екрану в випадковий відрізок часу);

Підрахунок робочого часу працівника (облік часу між початком та кінцем зміни, затрати працівника на обідню перерву, аналіз ефективності, відстеження піків активності, таблиця рейтингу співробітників по продуктивності, рейтинг най більш вживаних програм та сервісів);

Збереження взаємодії з клавіатурою.

Для безкоштовної версії доступний моніторинг до шести робочих систем. До платної версії входить:

Відеозапис дій працівника та его збереження з метою подальшого перегляду;

Контроль прущення правил компанії (знаходження та спостереження за підозрілими діями користувачів);

Програма сама знаходить та повідомлення про скоєння порушень;

дистанційне управління комп'ютером співробітника.

Це ПЗ має три головні переваги: запис та подальший перегляд дій на робочому столі комп'ютера, те що вона сама може виявляти не продуктивну діяльність персоналу і віддалене перехоплення комп'ютера працівника. Kickidler швидко набуває нові надбудови.

2.2.2 Опис програмного середовища ActivTrak

ActivTrak — це програмне забезпечення для моніторингу активності, аналізу продуктивності та поведінки співробітників на робочих комп'ютерах, а також для виявлення інсайдерських загроз. Це рішення для хмарного моніторингу (рис. 2.2) відзначається простотою у використанні та встановленні і має локальну версію.

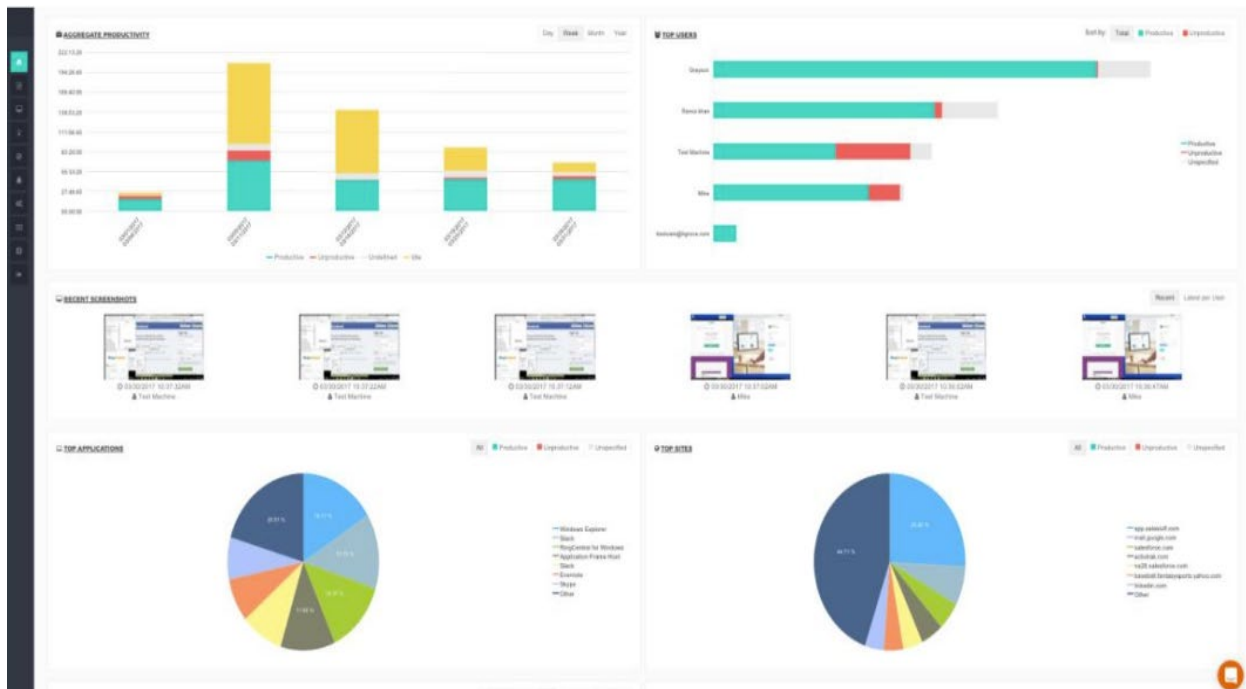


Рис. 2.2. Вікно програми ActivTrak

ActivTrak надає компаніям безкоштовну тестову версію програми для здатну відстежувати до трьох робочих станцій з обмеженнями функціональності та обсягом хмарного сховища до трьох ГБ.

До безкоштовної версії ActivTrak входять:

- онлайн-моніторинг комп'ютерів співробітників (реальний час доступу до перегляду екранів, без багатокористувацького режиму);
- звіти про робочий час;
- блокування сайтів;
- кейлоггер;
- скріншоти екранів співробітників (не більше одного знімка екрана для кожної активності).

Платна версія програми не додає нових функцій, але надає користувачеві необмежений обсяг хмарного сховища для зберігання логів програми та можливість моніторингу практично необмеженої кількості співробітників.

2.2.3 Опис системи Hubstaff

Hubstaff — це система відстеження часу, яка включає функції зйомки екрана, моніторингу завдань, автоматичної оплати праці, розширеного звітування та відстеження в режимі реального часу (рис.3.3.).

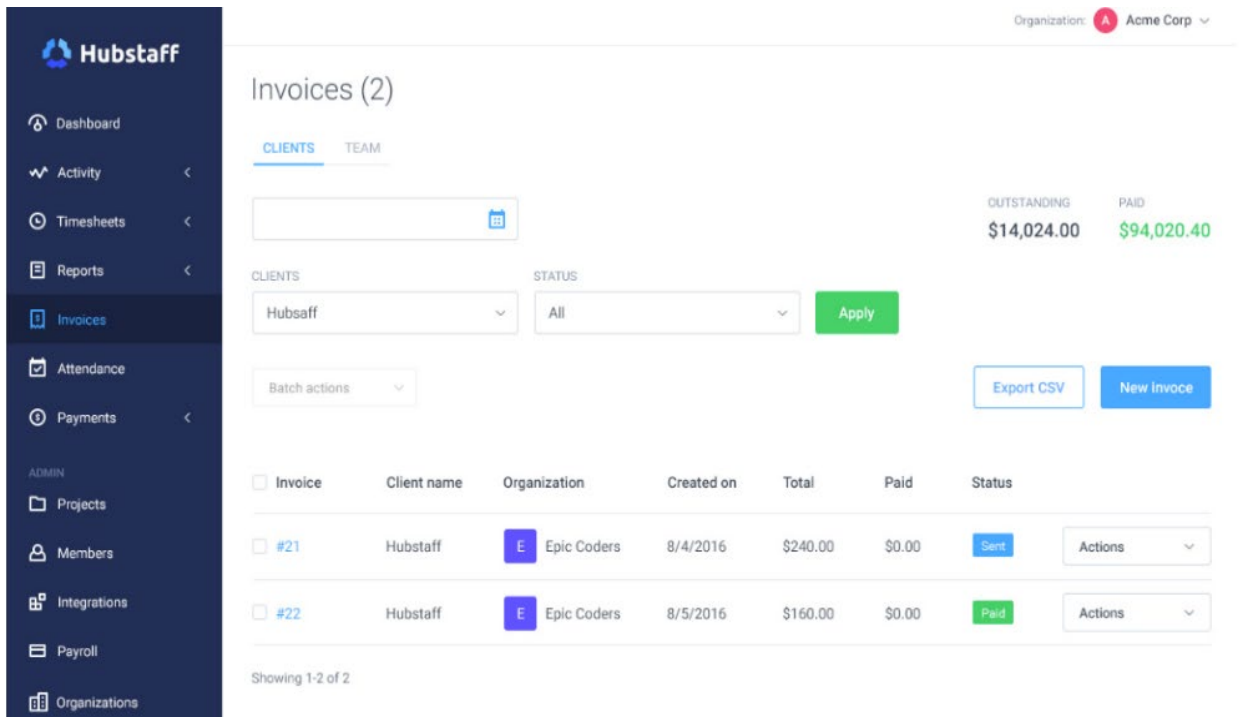


Рис.2.3. Головне вікно Hubstaff

Сервіс сумісний за більш ніж трьома десятками інструментів. Програма дещо відрзняється від інших сервісів, він більш заточений під оутсерс компаній чи невеликих компаній

До тестової версії входить моніторинг одного комп'ютера та має обмежений функціонал. До безкоштовної версії входить:

Інформація затрати часу на роботу;

Збір даних активу працівників клавіатурою та мишю;

Знімки робочого столу у випадковому відрізку часу.

Сервіс призначений для роботи з працівниками котрі знаходяться віддалено, також сервіс має унікальний інструмент для нарахування платні фрілансерам.

2.2.4 Опис системи DeskTime Pro

DeskTime — це онлайн-сервіс з мобільними та десктопними додатками для відстеження робочого часу, моніторингу прогресу в проектах, аналізу різних показників ефективності та створення докладних звітів. Як і Hubstaff, цей сервіс орієнтований на управління віртуальними командами (рис. 2.4.).

Використання DeskTime дозволяє досягти максимальної ефективності роботи колективу, позбавитися ручного заповнення табелів обліку робочого часу та створювати точні звіти для клієнтів або ради директорів компанії.

Безкоштовна версія DeskTime Pro дозволяє моніторити одного користувача, але має значні обмеження функціональності.

До безкоштовної версії DeskTime входять:

звіти про робочий час (відстеження часу);

звіти про використані додатки;

мобільний додаток.

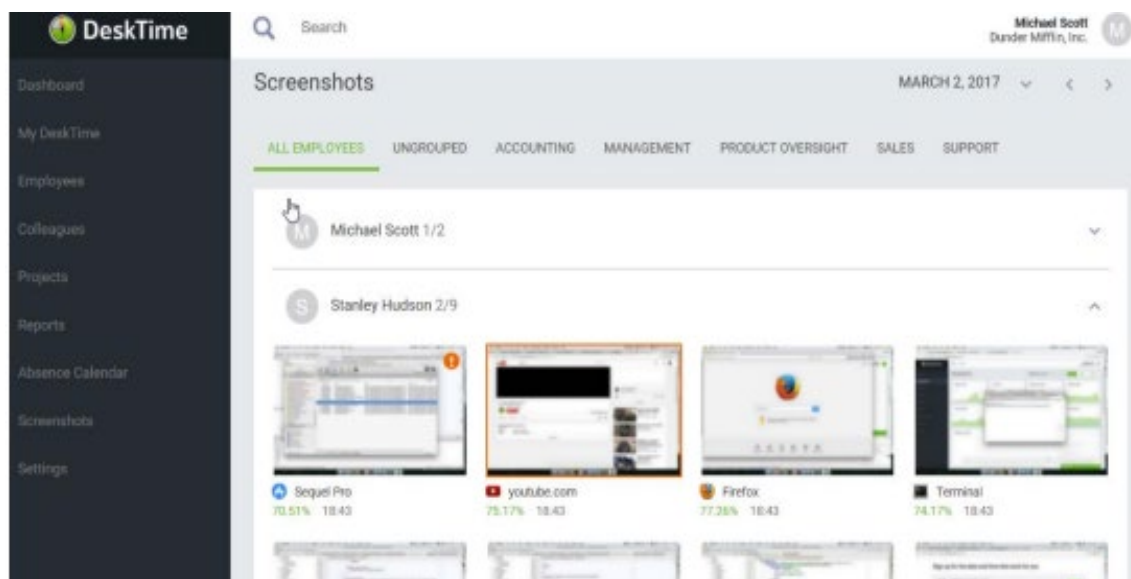


Рис. 2.4. Вікно програми DeskTime Pro

Програма для моніторингу співробітників DeskTime Pro має великий набір інструментів для контролю та управління віддаленими командами з можливістю тонкого налаштування для кожного користувача. Найбільш цікаві функції включають підрахунок заробітної плати, відстеження проектів та вбудований Pomodoro-трекер.

2.2.5 Опис програми Toggl

Програма Toggl (рис.2.5) належить до іншої категорії програм. Цей софт більше орієнтований на відстеження часу, а не на моніторинг співробітників з боку менеджерів. Програма працює як на десктопних, так і на мобільних пристроях. Toggl дозволяє створювати звіти та графіки про використаний час, які можна експортувати в різні формати. Відмінністю Toggl є можливість безкоштовного тестування всього функціоналу протягом тридцяти днів.

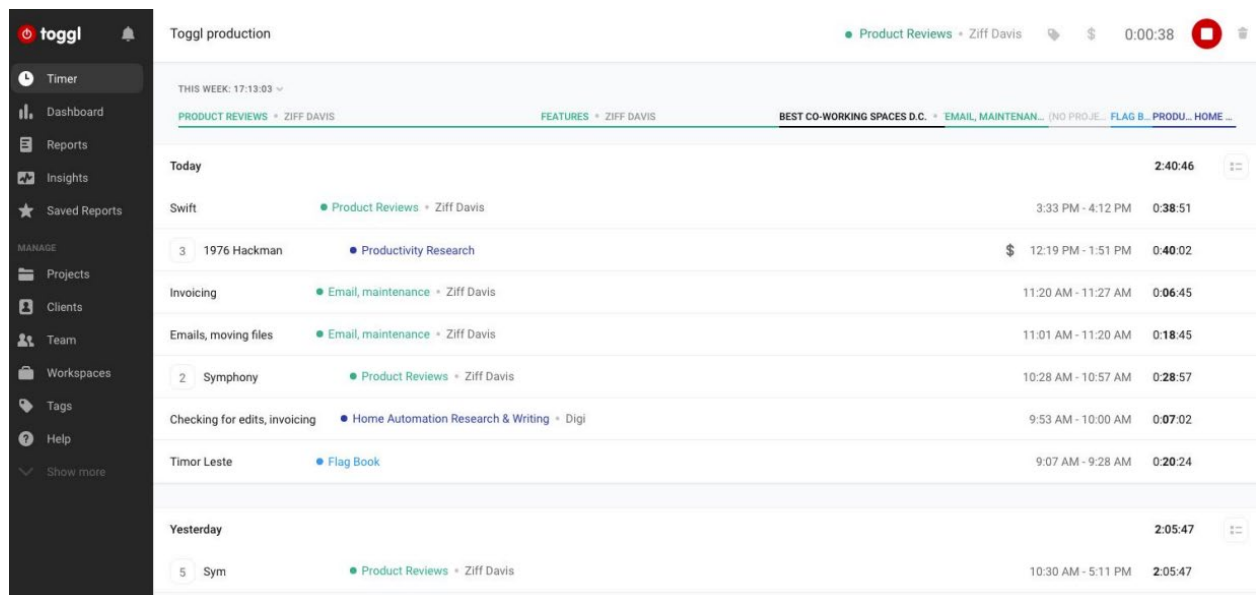


Рис 2.5. Вікно програми Toggl

До безкоштовної версії програми входить моніторинг до 5 робочих машин співробітників.

До безкоштовної версії Toggl входять:
звіти про робочий час;
вбудований Pomodoro-трекер;
складання графіків роботи.

У платній версії сервісу є два пакета: Стандарт і Преміум. Ці пакети не мають обмежень по кількості користувачів і включають розширені функції для управління проектами, складання звітів і управління командою.

2.2.6 Опис програми Screenshot monitor

Програма Screenshot Monitor (рис.2.6.) періодично робить скріншоти з моніторів співробітників і відправляє їх керівнику. Кількість скріншотів,

зроблених програмою за годину, залежить від вибраного тарифного плану. У безкоштовній версії доступно три скріншоти на годину. Крім цього, в безкоштовній версії є функція обліку робочого часу. Програма досить лояльна порівняно з іншими продуктами, наприклад, співробітник має можливість видаляти скріншоти, на яких зафіксована особиста інформація.

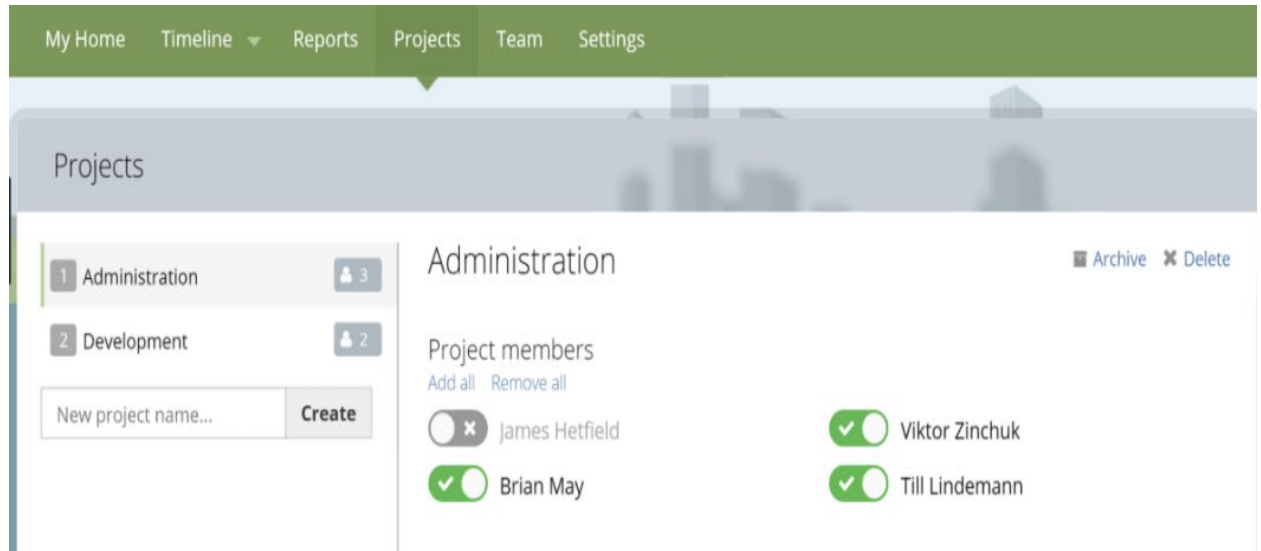


Рис 2.6. Вікно програми Screenshot monitor

У безкоштовній версії програми можливо відстежувати активність на трьох комп'ютерах. Функції, що доступні безкоштовно, включають:

- звіти про використаний робочий час;
- скріншоти з екранів комп'ютерів (до трьох скріншотів на годину);
- зберігання скріншотів протягом двох тижнів.

Платна версія програми надає можливість аналізу рівнів активності користувачів, використання додатків і веб-адрес. Про-версія здатна робити до тридцяти скріншотів на годину і зберігає їх протягом одного року.

2.2.7 Опис програмного середовища Tmetric

TMetric (рис.2.7.) це сервіс для відстеження часу, який ідеально відповідає потребам фрілансерів та малих команд, що працюють з віддаленими клієнтами і потребують простого інструменту для управління часом та завданнями.

Функціональні можливості TMetric включають:

- встановлення бюджетів на проекти;

визначення ставок для завдань, клієнтів, проектів або членів команди;
створення тегів та їх застосування до завдань;
візуалізація робочого дня на часовій шкалі;
генерація детальних звітів про витрачений час і зароблені кошти.

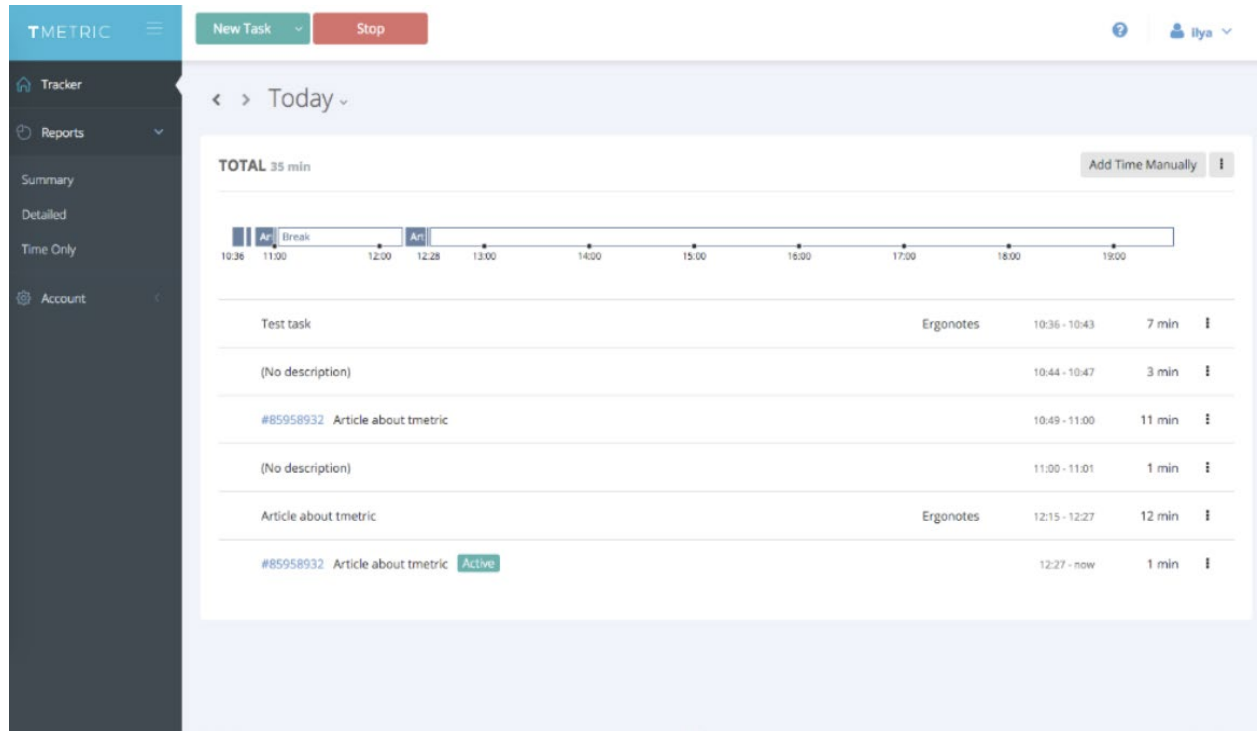


Рис. 2.7. Вікно програми Tmetric

Для безкоштовної версії програми є можливість моніторингу до 5 робочих комп'ютерів співробітників.

До безкоштовної версії TMetric входять:
звіти про робочий час;
необмежена кількість проектів і клієнтів;
-інтеграція з більшістю відомих веб-додатків для бізнесу.

Функції бюджетування проектів та виставлення рахунків доступні лише в платній версії, як і звіти за додатками. Також синхронізація з JIRA і Redmine відсутня в безкоштовній версії.

До платної версії TMetric входять:
веб-моніторинг для контролю відвідуваних співробітниками сайтів;
моніторинг використання програм, що дозволяє бачити, якими додатками користуються працівники;
відстеження активності за допомогою моніторингу клавіатури та миші, що допомагає визначити реальну робочу активність співробітників.

У табл.2.1 представлено основні характеристики найпопулярніших і доступних безкоштовних програм, за допомогою яких можна слідкувати за активністю та продуктивністю співробітників.

Таблиця 2.1 .

Основні характеристики систем моніторингу

Програма	Пробний період (free)	Кіл-сть користувачів	Хмарна/інсталяція	Мобільний додаток	Онлайн моніторинг	Сумісні ОС
ActivTrak	до 2-х тижнів	3	Хмара		+	Windows, Mac, Chrome, Android, iOS
Hubstaff	до 2-х тижнів	1	Хмара	+		Mac, Linux, Android, iOS, Chrome OS, браузер Chrome
Терамінд	7-14 днів	5	І те, і інше			Windows, Windows Server, Mac
Kickidler	до 2-х тижнів	6	І те, і інше		+	Windows, GNU/ Linux
DeskTime Pro	до 2-х тижнів	1	Хмара	+		Windows, Mac, Linux, Android, iOS.
Toggl	30 днів	5	І те, і інше			iPhone, Android, Windows, Mac, Linux
Screenshot monitor	30 днів	3	І те, і інше			Windows, Mac
Tmetric	30 днів	5	І те, і інше			macOS, Windows, Linux, iOS, Android

Проведений порівняльний аналіз програмних середовищ Kickidler, ActivTrak, Hubstaff, DeskTime Pro, Toggl, Screenshot Monitor та Tmetric свідчить, що серед них, за своїми функціональними можливостями позитивно вирізняється система Kickidler.

РОЗДІЛ 3 . РОЗРОБКА СИСТЕМИ КОНТРОЛЮ ТА ОБЛІКУ РОБОЧОГО ЧАСУ

3.1. Вибір інструментів розробки.

3.1.1 Опис мови C#

C# – це мова програмування, яка зосереджується на об'єктах і даних. Вона слідує принципам об'єктно-орієнтованого програмування (ООП), що робить її ідеальним інструментом для опису та моделювання складних систем.

У C# дані та функції, які з ними пов'язані, упаковуються в об'єкти, що сприяє більш логічному і структурованому коду.

Типобезпека – ще одна важлива характеристика C#. Це означає, що під час роботи з даними в C# ви зобов'язані явно вказувати їхній тип, що зменшує можливість помилок і робить ваш код надійним.

C# тісно інтегрований з .NET (Microsoft .NET Framework) – потужною платформою для розробки та виконання додатків. Важливою частиною .NET є CLR (Common Language Runtime), яка забезпечує середовище виконання для C# та інших мов .NET. CLR забезпечує керування пам'яттю, безпеку та інші важливі функції, що дає змогу C# розробникам зосереджуватися на створенні коду, а не на низько рівневих деталях.

Інтеграція з .NET також означає доступ до великої бібліотеки класів, яка містить готові рішення для безлічі завдань. Це скорочує час розробки і дозволяє створювати більш функціональні додатки.

Завдяки інструментам, таким як .NET Core і .NET 5 (та їхнім наступним версіям), C# розробники можуть створювати крос-платформні додатки, які не залежать від конкретної операційної системи. Це особливо важливо в сучасному світі, де різноманітність пристроїв і платформ перебуває на піку.

Ця мова програмування підтримує безліч різних платформ, включно з Windows, macOS і Linux. Це означає, що ви можете створювати додатки, які працюють практично скрізь.

Одна з основних галузей застосування C# є Створення додатків для Windows. C# має тісну інтеграцію з операційною системою Windows, що

робить його чудовим вибором для створення настільних додатків під цю платформу.

За допомогою технологій:

WPF або Windows Presentation Foundation — це графічна підсистема Microsoft для відтворення інтерфейсів користувача в програмах Windows. Це частина .NET Framework, що забезпечує уніфіковану модель програмування для створення бізнес-додатків для настільних ПК у Windows.

Windows Forms — це набір керованих бібліотек у .NET Framework, розроблених для розробки багатофункціональних клієнтських програм. Це графічний API для відображення даних і керування взаємодією користувачів зі спрощеним розгортанням і кращою безпекою в клієнтських програмах.

UWP є одним із багатьох способів створення клієнтських програм для Windows. Програми UWP використовують API WinRT для забезпечення потужного інтерфейсу користувача та розширених асинхронних функцій, які ідеально підходять для пристроїв, підключених до Інтернету.

Тому для розробки додатку для цієї кваліфікаційної роботи - с# більш ніж доцільний вибір.

3.1.2 Основні компоненти для розробки

Під час розробки та впровадження програмного продукту використовувалося середовище візуальної розробки Visual Studio, розроблене компанією Microsoft. Visual Studio є інтегрованим середовищем розробки (IDE), яке дозволяє створювати:

Традиційні настільні програми для операційної системи Windows;

Мобільні застосунки для Windows, iOS та Android;

Веб-додатки;

Хмарні сервіси;

Розширення для Microsoft Office та SharePoint, а також розробку власних доповнень до Visual Studio;

Комп'ютерні ігри;

Бази даних для SQL Server та SQL Azure.

В Visual Studio є можливість працювати з різними технологіями та мовами програмування, серед яких: .NET, Node.js, C, C#, C++, Python, Visual Basic, F#, та JavaScript.

Архітектура операційної системи розроблена таким чином, що захищає програми від взаємного пошкодження та втручання самої системи. Вона включає стійкі механізми обробки екстраординарних ситуацій на кожному архітектурному рівні, а також використовує відновлювану файлову систему NTFS і передові методи управління пам'яттю, що підсилюються вбудованою системою безпеки.

Система підтримує багато національних мов різних країн світу завдяки використанню міжнародного стандарту ISO.

Unicode. Visual Studio 2019 можна інсталиувати та використовувати на таких офіційно підтримуваних операційних системах:

Windows 7 з Service Pack 1;

Windows 8.1

Windows 10

Windows Server 2012 R2

Windows Server 2016

Windows Server 2019

Мінімальні технічні вимоги для пристроїв:

Процесор з частотою не менше 1,8 ГГц, бажано мінімум двоядерний;
2 ГБ оперативної пам'яті, рекомендовано 8 ГБ або мінімум 2.5 ГБ для віртуальних машин;

Від 800 мегабайт до 210 гігабайт вільного простору на жорсткому диску, залежно від обраної конфігурації компонентів. Рекомендується використання SSD диска для оптимальної продуктивності;

Відеоадаптер з мінімальним роздільною здатністю 1280 на 720 пікселів, хоча для найкращої продуктивності Visual Studio рекомендується роздільна здатність 1366 на 768 пікселів або вище.

Для створення програмного продукту було обрано середовище Visual Studio з кількох вагомих причин:

По-перше, Visual Studio є найпопулярнішим інтегрованим середовищем для швидкої розробки додатків під Microsoft Windows, що є найбільш поширеною операційною системою.

По-друге, програмне середовище Visual Studio було вибрано через його зручний інтерфейс для інтеграції з Microsoft Office, що значно спрощує роботу з офісними додатками.

По-третє, Visual Studio є доступним для широкого кола користувачів.

По-четверте, середовище відоме своєю стабільністю, швидкодією та мінімальними вимогами до комп'ютерних ресурсів.

3.1.3 Опис технології WPF

Windows Presentation Foundation (WPF, кодове ім'я Avalon) є графічною підсистемою, включеною до .NET Framework 3.0, і тісно пов'язаною з XAML. Ця технологія інтегрована в Windows Vista і доступна для інсталяції на Windows XP Service Pack 2 та Windows Server 2003. WPF стало першим значним оновленням в технологіях розробки користувацьких інтерфейсів з часів випуску Windows 95, замінюючи старіші технології GDI та GDI+.

WPF вважається високорівневим об'єктно-орієнтованим фреймворком, що дозволяє розробляти складні двовимірні та тривимірні інтерфейси. Протягом багатьох років, розробники на .NET використовували Windows Forms для створення настільних програм - це добре налагоджена технологія, що забезпечує вирішення стандартних завдань для бізнес-додатків. Однак, деякі завдання вимагають значних часових та трудових витрат через залежність від архітектури операційної системи.

З впровадженням WPF, яке базується на DirectX, розробники отримали можливість ефективно створювати складні елементи управління та керувати процесом візуалізації без потреби у програмуванні на C# для реалізації анімаційних ефектів. Технологія спростила роботу з мультимедійним контентом, поліпшила модель зв'язування даних, друку та обробки документів. WPF поєднує переваги традиційної розробки для Windows з новаторськими підходами, дозволяючи створювати візуально насичені інтерфейси. Для роботи з цією технологією достатньо володіти C# та бібліотеками .NET Framework. Архітектура WPF детально представлена на рис. 3.1.

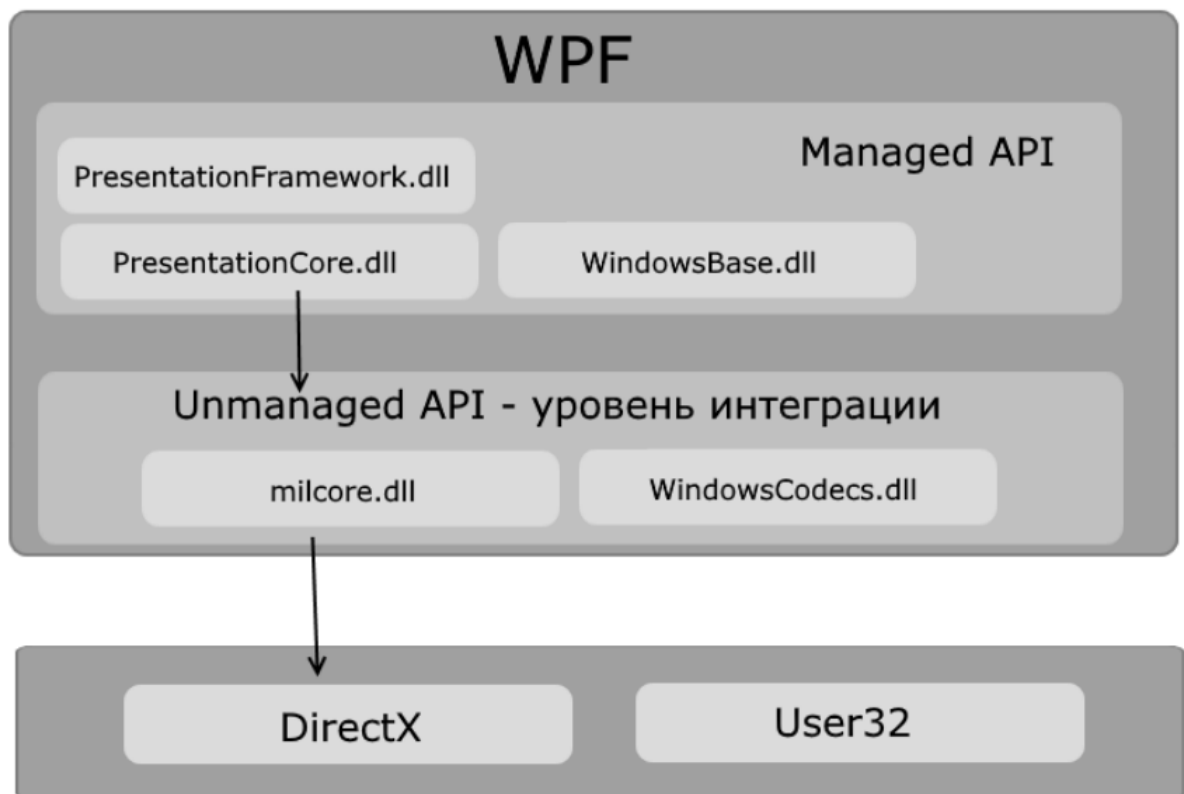


Рис. 3.1. Архітектура WPF

Як показано на рисунку 12, WPF розділяється на два рівня: керований API і некерований API (рівень інтеграції з DirectX). Керований API працює під управлінням загально використововуваного середовища виконання .NET - Common Language Runtime. Цей API визначає основний функціонал платформи.

WPF і включає наступні компоненти:

PresentationFramework.dll: містить реалізації компонентів і елементів управління, які використовуються для створення графічного інтерфейсу;

PresentationCore.dll: містить базові типи для більшості класів з

PresentationFramework.dll;

WindowsBase.dll: містить допоміжні класи, які використовуються в WPF, а також за межами цієї платформи;

Некерований API використовується для інтеграції WPF з DirectX і включає наступні компоненти:

milcore.dll: забезпечує інтеграцію компонентів WPF з DirectX, написаний на некерованому коді (C / C++);

WindowsCodecs.dll: бібліотека, яка забезпечує низькорівневу підтримку зображень в WPF.

Нижче розташовані компоненти операційної системи і DirectX, які відповідають за візуалізацію компонентів програми та проведення інших операцій на низькому рівні. Зокрема, за допомогою низькорівневого інтерфейсу Direct3D, що входить до складу DirectX, здійснюється трансляція.

На тому ж рівні розташована бібліотека user32.dll. Незважаючи на те, що WPF не використовує цю бібліотеку для рендерингу та візуалізації, вона залишається актуальною для деяких обчислювальних завдань, які не пов'язані з візуалізацією.

WPF є частиною платформи .NET і розвивається відповідно до розвитку фреймворку .NET, маючи ті ж версії. Перша версія WPF 3.0 вийшла разом з .NET 3.0 та Windows Vista в 2006 році. З тих пір платформа постійно розвивалася. Остання версія WPF 4.6 була випущена паралельно з .NET 4.6 в липні 2015 року, відзначивши дев'яти річчя даної платформи.

Переваги WPF для розробників включають:

Використання звичайних мов .NET, таких як C# і VB.NET, для реалізації логіки додатка.

Можливість декларативного визначення графічного інтерфейсу за допомогою мови розмітки XAML, що представляє альтернативу програмному створенню графічних елементів та елементів управління.

Незалежність від роздільної здатності екрану, оскільки всі елементи вимірюються в незалежних від пристрою одиницях.

Нові можливості, такі як створення тривимірних моделей, прив'язка даних, використання стилів, шаблонів, тем і багато іншого.

Гармонійна взаємодія з WinForms, дозволяючи використовувати традиційні елементи управління з WinForms у додатках WPF.

Широкі можливості для створення різноманітних додатків, включаючи мультимедіа, дво- і тривимірну графіку, анімації, прив'язку даних та інше.

Апаратне прискорення графіки для поліпшення продуктивності програм, незалежно від їх складності та графічних вимог.

Підтримка різних версій операційних систем сімейства Windows, від Windows XP до Windows 10.

3.1.4 Опис мови розмітки XAML

XAML (скорочення від Extensible Application Markup Language – розширювана мова розмітки застосунків) є мовою розмітки, яка використовується для створення екземплярів об'єктів .NET. Хоча XAML може бути застосована в різних сферах, її основне призначення – створення інтерфейсів користувача в WPF. Інакше кажучи, документи XAML визначають розташування панелей, кнопок та інших елементів керування, які складають вікна додатків WPF.

Зазвичай вам не доведеться писати код XAML вручну, оскільки для цього існують інструменти, що автоматично генерують необхідний код. XAML використовується головним чином для опису інтерфейсу, тоді як логіка додатка реалізується за допомогою процедурного коду (VB, C#, JavaScript тощо). XAML може бути використаний як для браузерних додатків, так і для звичайних настільних додатків.

Основні категорії елементів у XAML включають:

Панелі

Елементи керування

Елементи, пов'язані з документами, і графічні фігури

Додатки, оголошені в XAML, можуть включати кілька сторінок:

PageViewer – дозволяє розбивати зміст на сторінки і забезпечує навігацію між ними.

ContextMenu – допомагає створювати навігаційні меню для додатків.

XAML широко використовується в WPF, Windows Workflow Foundation (WWF), .NET Framework 3.0 і Silverlight. В WPF XAML служить мовою розмітки для користувацького інтерфейсу, дозволяючи визначати елементи інтерфейсу, прив'язку даних, підтримку подій та інші властивості.

Файли XAML можна створювати і редагувати за допомогою інструментів візуального дизайну, таких як Microsoft Expression Blend, Microsoft Visual Studio і WPF visual designer. Також можна використовувати стандартний текстовий редактор, редактор коду на кшталт XAMLPad або графічний редактор, такий як Vectropy.

Все, що створено або реалізовано в XAML, може бути виражено за допомогою традиційних мов .NET, таких як C# або Visual Basic .NET. Однак ключовим аспектом XAML є зменшення складності завдяки використанню

XML, що полегшує роботу з ним. Це призвело до появи багатьох продуктів, що створюють додатки на основі XAML. Оскільки XAML базується на XML, розробники і дизайнери можуть одночасно працювати над вмістом без необхідності компіляції.

Об'єктний елемент зазвичай оголошує екземпляр типу, визначеного в збірках, що підтримують відповідну технологію. Синтаксис об'єктних елементів починається з дужки (<), за якою слідує ім'я типу для створення екземпляра. Після цього можна оголосити атрибути, завершуючи тег закриваючою дужкою (>), або використовувати самозакривний тег.

Властивості об'єкта зазвичай виражаються у вигляді атрибутів об'єктного елемента. Синтаксис атрибутів включає назву властивості, оператор призначення (=) і значення атрибута в лапках. Синтаксис атрибутів є найпростішим і інтуїтивно зрозумілим для розробників, які раніше використовували мови розмітки.

Однак для деяких властивостей синтаксис атрибутів неможливий, оскільки об'єкт або відомості для значення властивості не можуть бути адекватно представлені у вигляді рядка. У таких випадках використовується синтаксис елементів властивостей. Відкриваючий тег для елемента властивості має вигляд <ім'я_типу.ім'я_властивості>. Зазвичай вміст тега є об'єктним елементом типу, який відповідає значенню цієї властивості, після чого тег закривається відповідним закриваючим тегом.

Якщо можливий синтаксис атрибутів, його використання зазвичай зручніше і призводить до більш компактної розмітки, хоча вибір часто залежить лише від стилю. XAML включає оптимізації для зручності читання, наприклад, якщо певна властивість приймає тип колекції, дочірні елементи в межах цієї властивості автоматично стають частиною колекції.

XAML підтримує концепцію властивості вмісту, що дозволяє класу призначити тільки одну властивість як властивість вмісту. Дочірні елементи цього об'єктного елемента використовуються для задання значення цієї властивості, що дозволяє створити наочну метафору батьківських-дочірніх елементів у розмітці.

3.2 Програмна реалізація задачі контролю та обліку робочого часу

Розроблене програмне забезпечення для спостереження за персоналом збирає та аналізує дані про робочий час, надаючи інформацію про рівень продуктивності працівників.

На рис. 3.2 показана структура основних елементів візуального інтерфейсу.

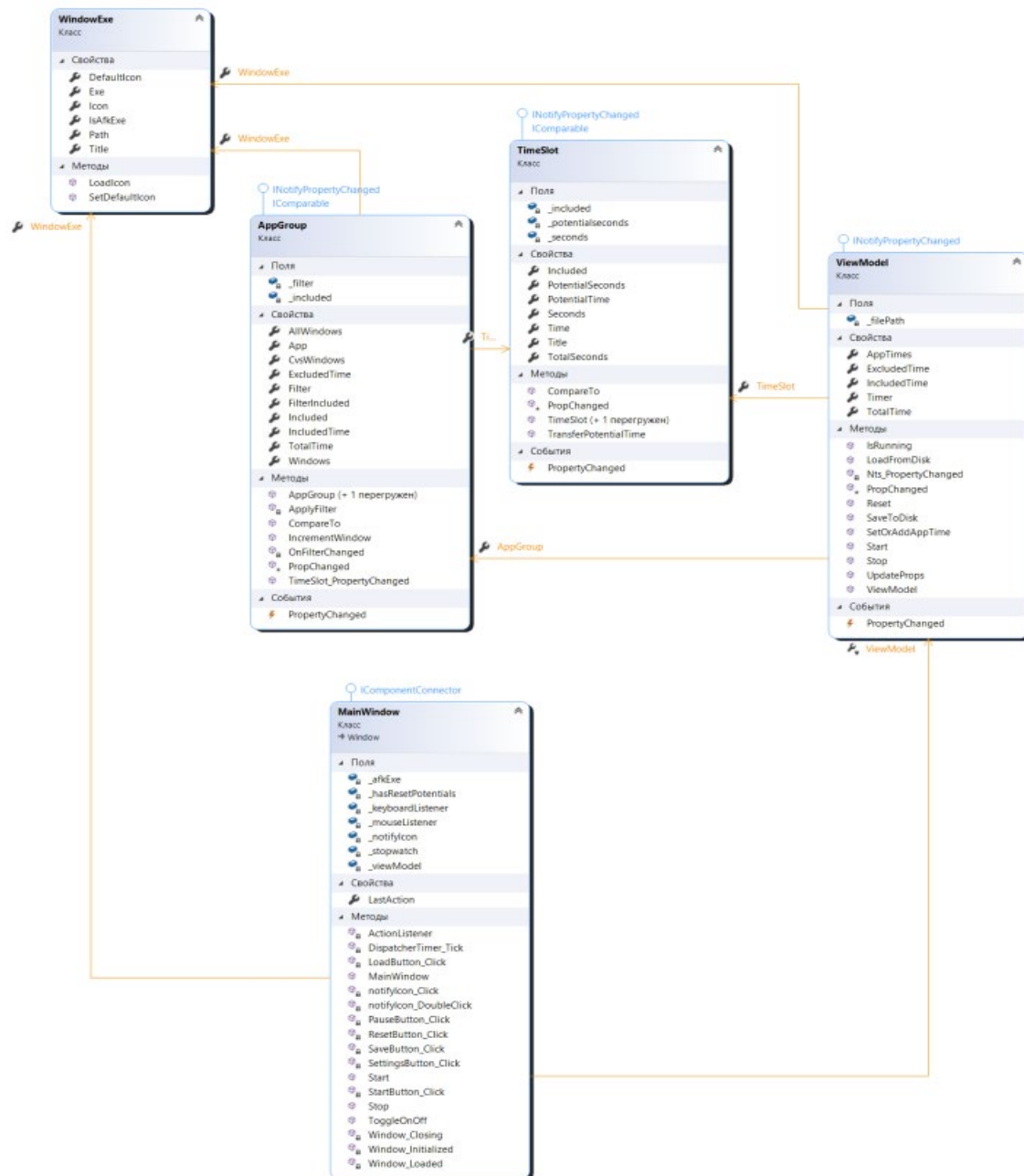


Рис.3.2. Структура програми

MainWindow – це основний клас, що описує поведінку компонентів візуального інтерфейсу. У конструкторі класу створюється об'єкт ViewModel. Одним з полів класу є об'єкт WindowsExec, який відповідає за реєстрацію відсутності користувача за комп'ютером.

Програмний код наведено у Додатку А

У конструкторі класу подія Timer.Tick з класу ViewModel підписується на метод DispatcherTimer_Tick. Крім того, створюються слухачі для подій, що відстежують відсутність рухів миші та натискань на клавіатуру.

Програмний код наведено у Додатку А

Обробник події DispatcherTimer_Tick перевіряє, чи час останньої дії перевищує встановлений час бездіяльності (AFKDelay). Якщо перевищує і потенціали не були скинуті, встановлюється прапорець _hasResetPotentials, і для кожного вікна в AppTimes обнуляються PotentialSeconds. Додається або оновлюється час для _afkExec.

Програмний код наведено у Додатку А

Клас ViewModel описує і зберігає значення таймера, а також список об'єктів AppGroup, для яких фіксуються часові параметри використання додатків. Методи IncludedTime, ExcludedTime та TotalTime повертають час активної роботи з додатком.

Програмний код наведено у Додатку А

Метод SetOrAddAppTime виконує наступні дії: якщо додаток вже присутній у списку, він додає час його використання; в разі внутрішніх вікон (наприклад, вкладок браузера) час використання сортується за спаданням. Якщо додаток відсутній у списку, то створюється новий запис у списку додатків AppTimes.

Програмний код наведено у Додатку А

Метод 'LoadFromDisk' перевіряє наявність файлу за вказаним шляхом. Якщо файл існує, він відкривається для читання. Вміст файлу, який є серіалізованим об'єктом 'ObservableCollection<AppGroup>', десеріалізується та додається до списку 'AppTimes'. Потім циклом обробляються елементи 'AppGroup', завантажуються їхні значки, встановлюються обробники подій для відстеження змін та оновлюються властивості.

Програмний код наведено у Додатку А

Клас AppGroup збирає дані щодо активного часу роботи, часу простою та загального часу використання додатку.

Клас TimeSlot містить два поля з інформацією у секундах та автоматично розраховує загальну кількість секунд.

Клас WindowsExe відповідає за збереження інформації про додаток і отримання його піктограми за допомогою методу LoadIcon.

Програмний код наведено у Додатку А

Після запуску програми на екрані з'являється головне вікно (рис.3.3).

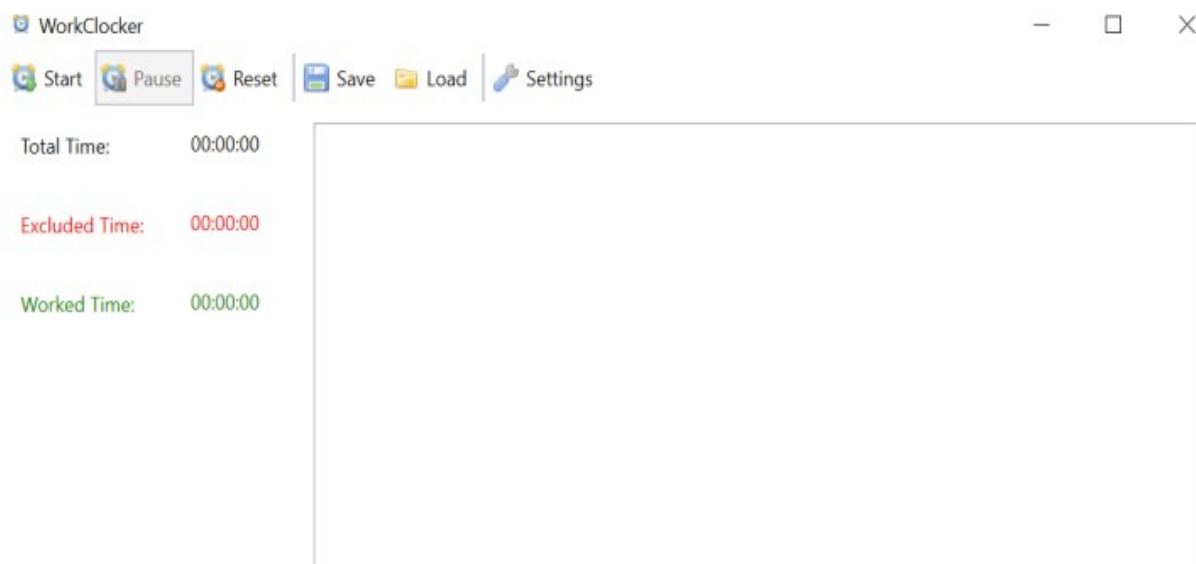


Рис. 3.3. Головне вікно програми

У верхній частині головного вікна розташовано меню, яке надає користувачеві такі опції:

- почати трекінг;
- призупинити трекінг;
- скинути дані трекінгу;
- зберегти журнал подій у файл;
- завантажити журнал подій з файлу;
- перейти до налаштувань (див. рис. 3.3).

У лівій частині вікна розміщена статистика, яка включає такі дані:

- загальний час роботи трекера;
- кількість вручну відключеного часу від загального робочого часу;
- робочий час.

Білий прямокутник – вікно журнагу дій. На рис. 3.4 журнал дій відсутній, так як трекер вимкнений.

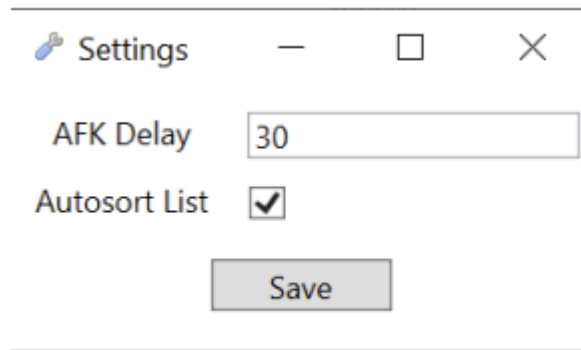


Рис 3.4. Вікно налаштувань

У вікні налаштувань у розділі «AFK Delay» можна встановити період часу в секундах, після якого програма почне реєструвати, що користувач неактивний і віддалено працює від комп'ютера.

Пункт «Autosort List» дозволяє автоматично сортувати програми в журналі за кількістю часу, витраченого на них.

Після початку трекінгу можна моніторити час, проведений у вікні журналу (див. рис. 3.5).

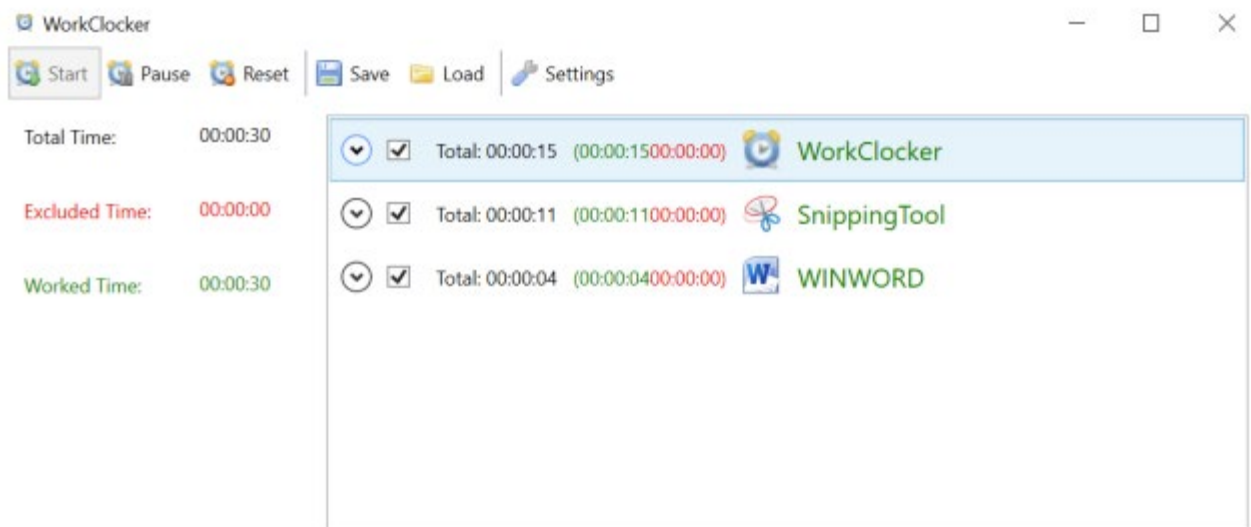


Рис. 3.5. Трекінг часу

Якщо протягом встановленого періоду часу користувач не взаємодівав ні з клавіатурою, ні з мишею, то у журналі відображатиметься процес «Away From Keyboard» (див. рис. 3.6).

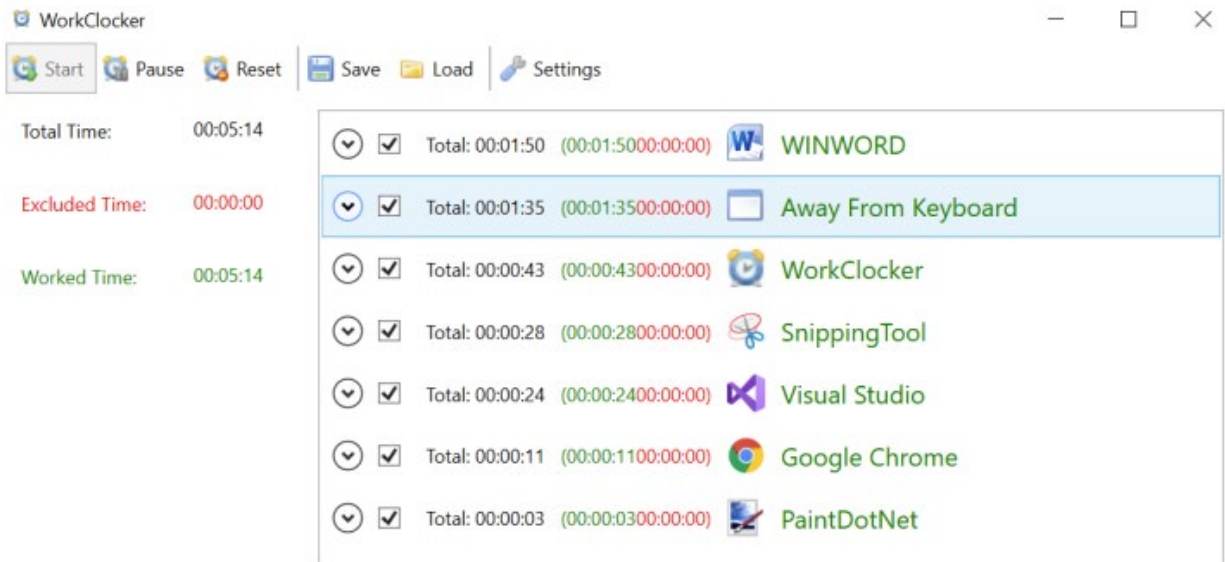


Рис.3.6. Процес «Away From Keyboard»

Для здійснення докладного контролю над вікном, кожен запис у журналі можна розгорнути, як показано на рис. 3.7.

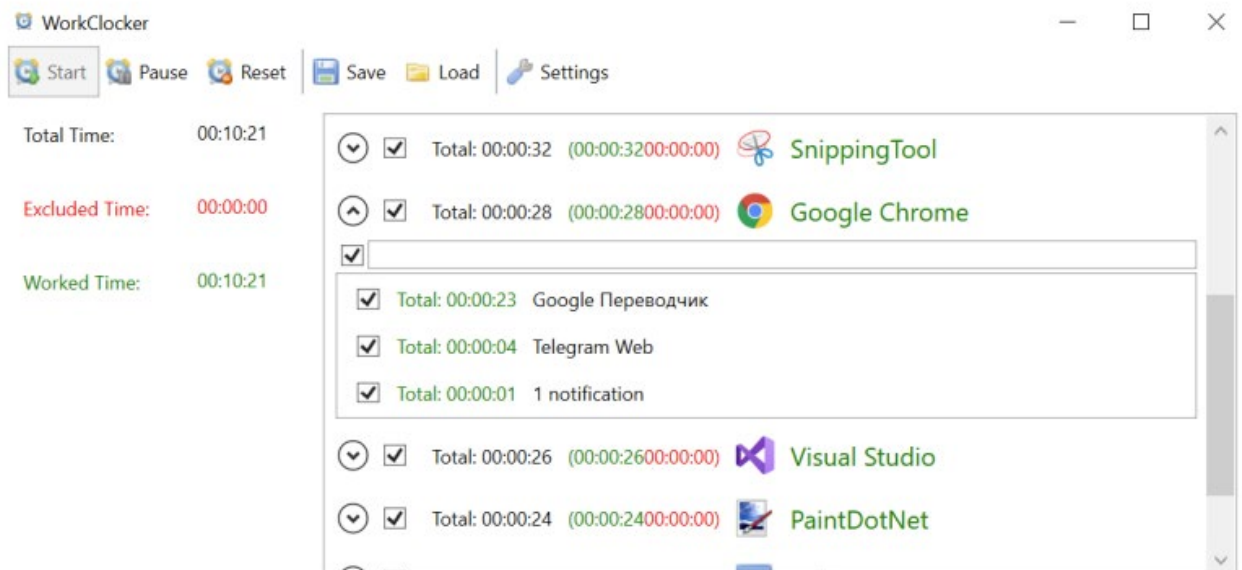


Рис.3.7. Розгорнутий запис

За замовчуванням всі процеси у журналі вважаються активними. Щоб позначити як неактивні, потрібно вручну зняти позначки з відповідних процесів. Ця дія буде відображена у статистиці часу (див. рис. 3.8).

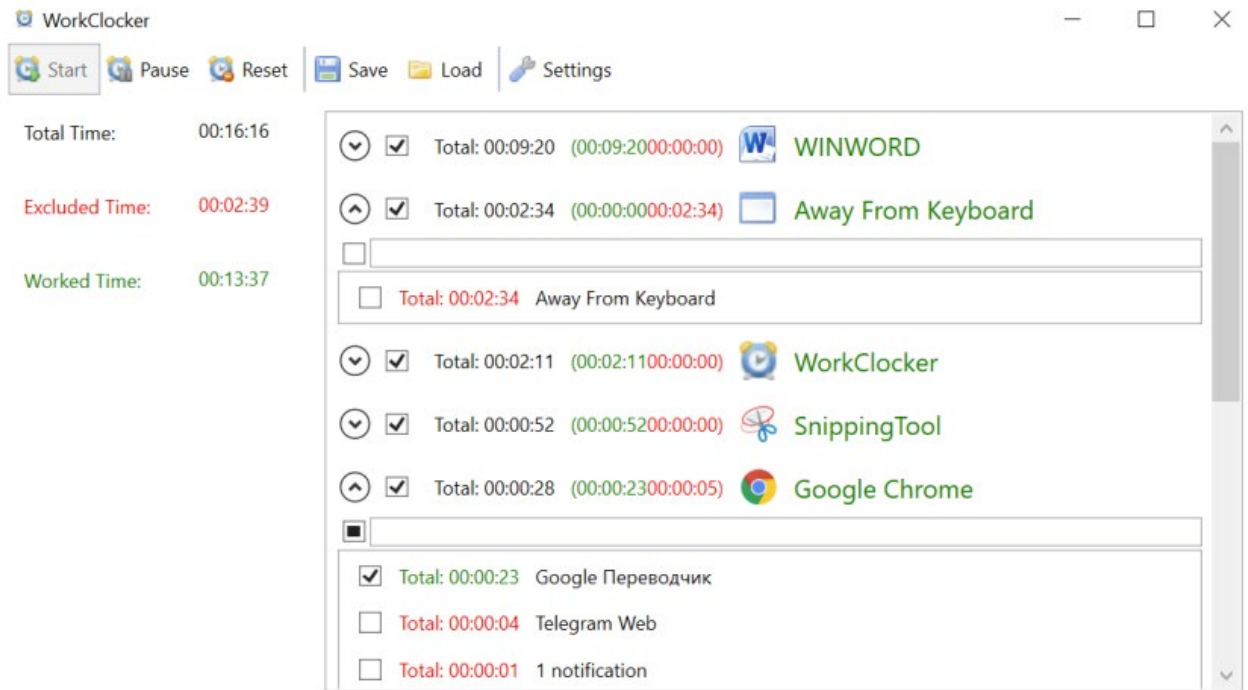


Рис 3.8. Керування неробочим часом

При зменшенні програмного вікна, воно автоматично згортається до системного трею. Коли робота завершується, статистика автоматично зберігається у файлі і стає доступною при наступному запуску програми.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було встановлено, що впровадження програмного забезпечення для моніторингу завантаженості комп'ютерного робочого місця — це один із найбільш ефективних способів не лише підвищення продуктивності праці співробітників, але й здійснення контролю за використанням ресурсів компанії та захисту від витоку конфіденційної інформації. Системи моніторингу надають детальну інформацію, необхідну для аналізу всіх вищезгаданих комп'ютерних дій. Завдяки цим інструментам можна оцінити реальну продуктивність співробітників, визначити, хто працює ефективно, хто відстає, і в яких сферах потрібне покращення.

Програмне забезпечення для моніторингу співробітників включає широкий спектр функцій для контролю за використанням робочого часу. Найкращі програми дозволяють адміністраторам встановлювати конкретні правила та політики для різних членів організації та отримувати сповіщення про порушення цих правил. Також є можливість контролювати віддалених працівників за допомогою щоденних звітів про активність, відстеження активності та скріншотів. Це також спрощує процес нарахування заробітної плати, оскільки можна ввести погодинні ставки для кожного працівника в команді.

Моніторинг використання пристроїв співробітників також може бути дієвим способом мотивувати їх раціонально використовувати свій час. Деякі компанії рекомендують проводити наради щодо ефективності для оцінки продуктивності та визначення напрямків для вдосконалення. Відстеження та збереження діяльності співробітників також може бути корисним для надання доказів у випадку неналежної поведінки.

Програмне забезпечення для моніторингу працівників також сприяє захисту бізнесу та мережі. Використовуючи функції моніторингу, можна бути впевненим, що працівники не відвідують небезпечні веб-сайти та не виконують шкідливих дій. Такі функції, як моніторинг електронної пошти, USB-накопичувачів і друку, допомагають переконатися, що конфіденційні дані не витікають. Моніторинг мережі дозволяє відстежувати функціональність всіх мережевих компонентів та зменшувати час простою.

Програмне забезпечення для моніторингу співробітників включає широкий спектр функцій для контролю за використанням робочого часу. Найкращі програми дозволяють адміністраторам встановлювати конкретні правила та політики для різних членів організації та отримувати сповіщення про порушення цих правил. Також є можливість контролювати віддалених працівників за допомогою щоденних звітів про активність, відстеження активності та скріншотів. Це також спрощує процес нарахування заробітної плати, оскільки можна ввести погодинні ставки для кожного працівника в команді.

Оскільки сучасний ринок пропонує достатньо велику кількість програмних продуктів, які забезпечують сервіси контролю часу, моніторингу трудової активності та захисту конфіденційних даних від втрат, важливим є визначення критеріїв вибору систем моніторингу. Саме від правильності їх формування залежить ефективність вибору програмного забезпечення.

За результатами виконання кваліфікаційної роботи було розроблено програмне середовище з використанням таких технологій та мов програмування, як .NET, Node.js, C, C#, C++, Python, Visual Basic, F#, JavaScript, WPF, XAML. Було розглянуто сучасні технологічні платформи для реалізації подібних програм, проведено порівняльний аналіз можливих варіантів реалізації, проаналізовано його результати, а також реалізовано аналітику, яка відстежує всі види взаємодій користувача зі сторінками додатка.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Time Doctor. Advanced Employee Monitoring Software. URL: <https://www.timedoctor.com/>
2. RescueTime. Personal Productivity and Time Management Software. URL: <https://www.rescuetime.com/> (дата звернення 16.03.2021)
3. Toggl Track. Time Tracking and Reporting. URL: <https://toggl.com/track/>
4. Clockify. Free Time Tracker for Teams. URL: <https://clockify.me/> (дата звернення 19.03.2021)
5. ActivTrak. Workforce Analytics and Productivity Monitoring. URL: <https://www.activtrak.com>
6. Workpuls. Employee Monitoring and Time Tracking. URL: <https://www.workpuls.com/>
7. DeskTime. A Time Tracking App For Businesses & Freelancers. URL: <https://desktime.com/>
8. Teramind. User Activity Monitoring and Data Loss Prevention. URL: <https://www.teramind.co/>
9. VeriClock. Employee Time Tracking Software. URL: <https://www.vericlock.com/>
10. Harvest. Time Tracking and Online Invoicing. URL: <https://www.getharvest.com/>
11. Everhour. Time Tracking Software for Teams. URL: <https://everhour.com/>
12. Timely. Automatic Time Tracking and Timesheet Software. URL: <https://timelyapp.com/>
13. ClickTime. Time and Expense Tracking for Businesses. URL: <https://www.clicktime.com/>
14. TSheets. Time Tracking and Scheduling Software by QuickBooks. URL: <https://www.tsheets.com/>
15. **Instant Office: Reports and Research.** URL: <https://www.instantoffices.com>]
16. Zenefits. HR and Payroll Software with Time Tracking. URL: <https://www.zenefits.com/>
17. TimeCamp. Time Tracking and Project Management. URL: <https://www.timecamp.com/>

18. ManicTime. Time Tracking with Automatic Work Hour Logging. URL: <https://www.manictime.com/>
19. OfficeTime. Time and Expense Tracking Software. URL: <https://www.officetime.net/>
20. actiTIME. Time Tracking Software for Businesses. URL: <https://www.actitime.com/>
21. Chrometa. Automatic Time Tracking Software. URL: <https://www.chrometa.com/>
22. Paymo. Work and Project Management Software. URL: <https://www.paymoapp.com/>
23. MyHours. Time Tracking for Freelancers and Teams. URL: <https://myhours.com/>
24. Hubstaff Tasks. Agile Project Management. URL: <https://hubstaff.com/tasks>
25. Insightful (formerly Workpuls). Employee Monitoring and Productivity Insights. URL: <https://www.insightful.io/>
26. Connecteam. Employee Management App with Time Tracking. URL: <https://www.connecteam.com/>
27. Jibble. Free Time and Attendance Tracking Software. URL: <https://www.jibble.io/>
28. OnTheClock. Employee Time Clock and Attendance Software. URL: <https://www.ontheclock.com/>
29. FunctionFox. Timesheets and Project Management Software. URL: <https://www.functionfox.com/>
30. Replicon. Time and Attendance Software. URL: <https://www.replicon.com/>
31. Scoro. Work Management Software. URL: <https://www.scoro.com/>
32. RescueTime for Business. Productivity and Time Management. URL: <https://www.rescuetime.com/business>
33. Qbserve. Automatic Time Tracking for Mac. URL: <https://qotoqot.com/qbserve/>
34. Timeular. 8-Sided Time Tracking Device and App. URL: <https://timeular.com/>
35. Monitask. Employee Monitoring Software. URL: <https://www.monitask.com/>
36. TrackingTime. Time Tracking and Project Management. URL: <https://trackingtime.co/>

37. Harvest Forecast. Team Scheduling and Planning Software. URL:
<https://www.getharvest.com/forecast>
38. Asana. Work Management Platform with Time Tracking Integration. URL:
<https://asana.com/>
39. Monday.com. Work OS with Time Tracking Features. URL:
<https://monday.com/>
40. Toggl Plan. Project Planning and Team Management Tool. URL:
<https://toggl.com/plan/>
41. Моніторингові програмні продукти. URL:
https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3%D0%BE%D0%B2%D1%96_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D1%96_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%B8
42. Найкраще програмне забезпечення для моніторингу співробітників та все, що вам потрібно знати про моніторинг співробітників URL:
<https://clevercontrol.com/uk/best-employee-monitoring-software/#pa>
43. Майстер Л.А. Моніторинг результатів діяльності персоналу в механізмі підвищення його конкурентоспроможності // Демографія, зайнятість населення і соціально-економічна політика. 2014. 4. - с. 225-229
44. 12 найкращих інструментів моніторингу персоналу. URL:
<https://clevercontrol.com/uk/12-best-employee-monitoring-tools/>
45. Галіцин В. К. Системи моніторингу: монографія / В. К. Галіцин. – К.: КНЕУ, 2000. – 231 с.

ДОДАТОК

ДОДАТОК А

```
private readonly WindowExe _afkExe = new WindowExe {Title = "Away
From Keyboard", Exe = "Away From Keyboard", IsAfkExe = true};
```

```
private void DispatcherTimer_Tick(object sender, EventArgs e)
{
    if (LastAction > Properties.Settings.Default.AFKDelay)
    {
        if (!_hasResetPotentials)
        {
            _hasResetPotentials = true;
            foreach (var window in _viewModel.AppTimes.SelectMany(appGroup =>
appGroup.Windows))
            {
                window.PotentialSeconds = 0;
            }
            _viewModel.SetOrAddAppTime(_afkExe, 0,
Properties.Settings.Default.AFKDelay + 1);
        }
        else
        {
            _viewModel.SetOrAddAppTime(_afkExe, 0);
        }
        _viewModel.UpdateProps();
        return;
    }
}
```

```
_hasResetPotentials = false;
var curApp = Natives.GetFocusWindow();
```

```
private void DispatcherTimer_Tick(object sender, EventArgs e)
{
    if (LastAction > Properties.Settings.Default.AFKDelay)
```

```

{
    if (!_hasResetPotentials)
    {
        _hasResetPotentials = true;
        foreach (var window in _viewModel.AppTimes.SelectMany(appGroup =>
appGroup.Windows))
        {
            window.PotentialSeconds = 0;
        }
        _viewModel.SetOrAddAppTime(_afkExe, 0,
Properties.Settings.Default.AFKDelay + 1);
    }
    else
    {
        _viewModel.SetOrAddAppTime(_afkExe, 0);
    }
    _viewModel.UpdateProps();
    return;
}

_hasResetPotentials = false;
var curApp = Natives.GetFocusWindow();

if (curApp?.Title != null)
{
    TitleManipulator.CleanTitle(ref curApp);
    _viewModel.SetOrAddAppTime(curApp, LastAction);
}
}

```

```

public TimeSpan IncludedTime
{
    get
    {
        var totalSeconds = AppTimes.Sum(o => o.IncludedTime.TotalSeconds);
        return TimeSpan.FromSeconds(totalSeconds);
    }
}

```

```

    }
}

public TimeSpan ExcludedTime
{
    get
    {
        var totalSeconds = AppTimes.Sum(o => o.ExcludedTime.TotalSeconds);
        return TimeSpan.FromSeconds(totalSeconds);
    }
}

public TimeSpan TotalTime
{
    get
    {
        var totalSeconds = AppTimes.Sum(o => o.TotalTime.TotalSeconds);
        return TimeSpan.FromSeconds(totalSeconds);
    }
}

public void SetOrAddAppTime(WindowExe app, int lastAction, int timeInc = 1)
{
    if (AppTimes.Any(o => o.App.Exe == app.Exe))
    {
        AppTimes.First(o => o.App.Exe == app.Exe).IncrementWindow(app.Title,
lastAction, timeInc);

        if (Properties.Settings.Default.Autosort)
            AppTimes.BubbleSort();
    }
    else
    {
        app.LoadIcon();
        var newAppGroup = new AppGroup(app);
        newAppGroup.PropertyChanged += Nts_PropertyChanged;
    }
}

```

```

        newAppGroup.IncrementWindow(app.Title, lastAction, timeInc);
        AppTimes.Add(newAppGroup);
    }

    PropertyChanged("IncludedTime");
    PropertyChanged("ExcludedTime");
    PropertyChanged("TotalTime");
}

```

```

public void LoadFromDisk()
{
    if (!File.Exists(_filePath))
        return;

    var serializer = new XmlSerializer(typeof(ObservableCollection<AppGroup>));

    using (var reader = new StreamReader(_filePath))
    {
        AppTimes.Clear();
        foreach (var app in
(ObservableCollection<AppGroup>)serializer.Deserialize(reader))
        {
            AppTimes.Add(app);
        }
        reader.Close();
    }

    foreach (var app in AppTimes)
    {
        app.App.LoadIcon();
        app.PropertyChanged += Nts_PropertyChanged;
        foreach (var timeslot in app.Windows)
        {
            timeslot.PropertyChanged += app.TimeSlot_PropertyChanged;
        }
    }
}

```

```

    UpdateProps();
}

```

```

public void SaveToDisk()
{
    var serializer = new XmlSerializer(typeof(ObservableCollection<AppGroup>));

    using (var writer = new StreamWriter(_filePath))
    {
        serializer.Serialize(writer, AppTimes);
    }
}

public void LoadIcon()
{
    try
    {
        var icon = System.Drawing.Icon.ExtractAssociatedIcon(Path);
        if (icon == null)
        {
            Icon = DefaultIcon;
            return;
        }

        using (icon)
        {
            using (var bmp = icon.ToBitmap())
            {
                using (var stream = new MemoryStream())
                {
                    bmp.Save(stream, System.Drawing.Imaging.ImageFormat.Png);
                    stream.Position = 0; // Reset stream position
                    Icon = BitmapFrame.Create(stream);
                }
            }
        }
    }
}

```

```
    }  
}  
catch (Exception ex)  
{  
    // Handle specific exceptions here, or log them for debugging  
    Icon = DefaultIcon;  
    Console.WriteLine("Error loading icon: " + ex.Message);  
}  
}
```