

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних  
технологій

**Кваліфікаційна робота бакалавра**

на тему: **Windows додаток для трекінгу часу використання  
програм**

Виконав студент групи К-19  
спеціальності 122 Комп'ютерні науки  
Халлієва Гульзада

Керівник доктор філософії, доцент  
Бучинська Ірина Вікторівна

Рецензент д.ф.-м.н., професор  
Ковальчук Володимир  
Володимирович

Одеса 2023

## ЗМІСТ

ВСТУП .....	6
1 КЛАСИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ПРОГРАМИ-ТРЕКЕРИ.....	8
1.1 Основні характеристики програмного забезпечення .....	8
1.2 Програми-трекери .....	12
1.2.1 Призначення програм стеження .....	13
1.2.2 Боротьба з прониклим трекером.....	14
1.3 Моніторинг часу роботи за комп'ютером.....	15
1.3.1 Спостереження за раціональним витрачанням часу співробітників для прийняття правильного рішення .....	16
1.3.2 Відстеження часу означає краще управління робочим часом.....	16
1.3.3 Відстеження часу допоможе розставити пріоритети. ....	17
1.3.4 Віддалений контроль персоналу можливий з додатком обліку часу .....	17
2 ОПИС МОВИ ПРОГРАМУВАННЯ ДЛЯ ВИКОРИСТАННЯ ДОДАТКУ ...	19
2.1 Обґрунтування вибору мови програмування .....	19
2.1.1 Опис мови Java .....	19
2.1.2 Опис мови C.....	21
2.1.3 Опис мови C++ .....	22
2.1.4 Опис мови C#.....	25
2.2 Вибір середовища розробки.....	27
3 АНАЛІЗ НАЙПОПУЛЯРНІШИХ ДОДАТКІВ ДЛЯ МОНІТОРИНГУ ЧАСУ	30
3.1 Аналіз найкращого додатку для відстеження вільного часу.....	32
3.2 Аналіз кайкращого додатку для відстеження часу для інтеграції з інструментом керування проектами вашої команди .....	34
3.3 Аналіз найкращого додатку для автоматичного відстеження часу .....	36
3.4 Аналіз найкращого додатку для відстеження часу для іншої візуалізації часу .....	38
3.5 Аналіз найкращого додатку для відстеження часу для зменшення відволікань .....	39

4 ОПИС ПРОГРАМНОГО ЗАСТОСУНКУ.....	41
4.1 Технологія WPF.....	42
4.2 Мова розмітки XAML.....	46
4.3 Опис роботи програми.....	49
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

## ВСТУП

Людина протягом усього свого життя постійно намагається полегшити свою працю. Для цього він або експлуатує інших людей, або придумує нові пристрої, на які перекладає частину своїх справ. У наше століття бурхливого розвитку комп'ютерних технологій на допомогу людині прийшов комп'ютер, здатний замінити його і навіть перевершити в деяких аспектах діяльності. Важко назвати ту область діяльності людства, де б сьогодні не використовувався комп'ютер.

Сучасна робота за комп'ютером крім своїх переваг має недолік у вигляді великої кількості відволікаючих чинників, так званих Тайм-кілерів. Необхідно знати куди подівся робочий день, а замовлення чомусь так і не виконано. Детальна статистика наочно покаже, скільки часу проводиться в неробство.

Оптимізації робочого процесу при правильному використанні допоможе програма обліку робочого часу. За допомогою різних програм роботодавець може не тільки фіксувати час, протягом якого кожен співробітник знаходився на робочому місці, а й бути в курсі відвідуваних сторінок, переміщень по офісу, кількості перекурів. На підставі всіх отриманих даних, в «ручному» або автоматизованому режимі виникає можливість оцінити ефективність співробітників, вжити заходів для її поліпшення або скорегувати підходи до управління персоналом в залежності від кожної конкретної ситуації, умови якої підтвержені і актуалізовані використанням спеціалізованого сервісу.

Темою бакалаврської кваліфікаційної роботи є розробка програми для контролю за робочим часом.

Мета і завдання додатку полягають у допомозі користувачам в ефективному управлінні своїм часом.

Основні цілі такого додатку включає:

– реєстрація часу: додаток повинен дозволяти користувачеві

реєструвати час, витрачений на різні завдання, проекти або діяльності. це може бути досягнуто за допомогою таймерів або вручну введенням часових значень.

- аналітика і статистика: додаток повинен надати користувачеві можливість перегляду статистики і аналізу витраченого часу, це може включати графіки, діаграми або звіти, які допомагають зрозуміти, як використовується час і де можливі покращення.
- налаштування цілей і обмежень: додаток може допомагати встановлювати цілі та обмеження для витрачення часу на конкретні завдання або діяльності. наприклад, користувач може встановити обмеження часу для виконання певного завдання або встановити ціль витратити певну кількість часу на певну діяльність.
- сповіщення і нагадування: додаток може надавати сповіщення та нагадування користувачу про часові обмеження, наближення до цілей або паузи для відпочинку. це допомагає покращити свідоме використання часу та дотримання графіку роботи.

Загалом, цілю додатку для трекінгу часу є надання користувачеві інструментів для ефективного управління часом, створення свідомості про використання часу і підвищення продуктивності. Завдання додатку полягають у забезпеченні точного відстеження часу, аналізу даних, сприянні установленню цілей та нагадуванні користувачу про оптимальне використання часу.

Розроблюване програмне забезпечення має забезпечити можливість вести статистику та контролювати робочий час проведений за комп'ютером у операційній системі Windows.

## **1 КЛАСИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ПРОГРАМИ-ТРЕКЕРИ**

### 1.1 Основні характеристики програмного забезпечення

Програмне забезпечення (ПЗ) являє собою сукупність програм, призначених для розв'язання завдань на комп'ютері. Програма – це впорядкований набір команд. Програмне та апаратне забезпечення працюють взаємопов'язано і в неперервній взаємодії. Будь-який апаратний пристрій управляється програмно [1].

Програмне забезпечення можна поділити на три класи: системне, прикладне та інструментальне. Наведена класифікація є досить умовною. Інтеграція програмного забезпечення призвела до того, що практично будь-яка програма має риси кожного класу.

Системне ПЗ призначено для управління роботою комп'ютера, розподілу його ресурсів, підтримки діалогу з користувачами, надання їм допомоги в обслуговуванні комп'ютера, а також для часткової автоматизації розробки нових програм.

Системне ПЗ – це комплекс програм, багато з яких постачаються разом з комп'ютером та документацією до неї. Системне ПЗ можна розділити на три основні частини: операційні системи (ОС), системи програмування та сервісні програми [1].

Основними компонентами загальносистемного програмного забезпечення являються: операційні системи, які вирішують задачі взаємозв'язаного функціонування окремих компонентів.

Існують 3 типи операційних систем [1]:

- операційні системи пакетної обробки: порівняно велика швидкість логічних і автоматичних операцій, але в свою чергу мала швидкість вводу і виводу завантаженість процесора на 20 – 30%.
- операційна система з розподілом задач по часу (організовується черга вводу і виходу задач, і обслуговується до 15 користувачів і

процесор завантажений на 80-90%).

- операційна система реального часу, використовується для керування різними процесами.

Системне програмне забезпечення здійснює управління роботою обчислювальної системи. Як правило, системні програми забезпечують взаємодію інших програм з апаратними складовими, організацію інтерфейсу користувача. Сюди відносять операційні системи, сервісні системи.

Прикладне програмне забезпечення призначене для розв'язання прикладних завдань фахової діяльності людини (тобто, прикладне до практики). Спектр таких програм надзвичайно широкий: від виробничих та наукових до навчальних та розважальних. Сюди відносять розрахункові, навчаючі, моделюючі програми, комп'ютерні ігри, тощо [1].

Інструментальне програмне забезпечення призначене для розробки всіх видів інформаційно-програмного забезпечення. При цьому під інформаційним забезпеченням розуміють сукупність попередньо підготовлених даних, необхідних для роботи програмного забезпечення. Наприклад, будь-яка сучасна програма має вбудовану довідку для роботи з цією програмою. Файл довідки являє собою інформаційне забезпечення. До інструментального програмного забезпечення відносять: редактори (текстові, графічні, музичні), системи табличної обробки даних (табличні процесори), системи управління базами даних, транслятори мов програмування, інтегровані системи діло виробництва, тощо.

Системи програмування призначені для полегшення та для часткової автоматизації процесу розробки та відлагодження програм. Основними компонентами цих систем є транслятори з мов високого рівня, наприклад, Паскаль, Сі, Бейсик та ін. Особлива роль належить Ассамблерам. Програму мовою Ассамблера називають машинно-орієнтованою. Мовою Ассамблера користуються, як правило, системні програмісти [1].

Транслятори здійснюють перетворення програм з мов високого рівня на машинну мову. Крім того, транслятори звичайно здійснюють

синтаксичний аналіз програми, яка транслюється. Вони можуть також відлагоджувати та оптимізувати програми, які одержують, видавати документацію на програму та виконувати ряд інших сервісних функцій. Ассамблери перетворюють програми, які представлені у машинноорієнтованих мовах, на машинну мову.

Сервісні програми розширюють можливості ОС. Їх, звичайно, називають утілітами. Утіліти дозволяють, наприклад, перевірити інформацію у шістнадцятковому коді, яка зберігається в окремих секторах магнітних дисків; організувати виведення на принтер текстових файлів у визначеному форматі, виконувати архівацію та розархівацію файлів та ін.

У структурі прикладного програмного забезпечення можна виділити: прикладні програми як загального, так і спеціального призначення.

Прикладне ПЗ загального призначення – це комплекс програм, який одержав широке використання серед різних категорій користувачів. Найбільш відомими серед них є: текстові редактори, графічні системи, електронні таблиці, системи управління базами даних та ін. [1].

Текстові редактори дозволяють готувати текстові документи: технічні описи, службові листи, статті та ін. Найбільш відомі такі текстові редактори:

- лексикон;
- write;
- word.

Графічні системи багаточисельні, а їх функції – різноманітні. Серед них можна виділити системи ділової графіки (Microsoft PowerPoint, Lotus Freelance Graphics), художньої графіки, які ще називають просто графічними редакторами (Paintbrush), інженерної графіки та автоматизованого проектування (Autodesk AutoCad), системи обробки фотографічних зображень (Adobe Photoshop), а також універсальні графічні системи (CorelDRAW).

Програми роботи з електронними таблицями (ЕТ) дозволяють розв'язувати широке коло задач, зв'язаних з числовими розрахунками.



Найширше використовують серед програм такого класу Supercalk, Microsoft Excel та Lotus 1-2-3.

Системи управління базами даних (СУБД) призначені для об'єднання наборів даних з метою створення єдиної інформаційної моделі об'єкта. Ці програми дозволяють накопичувати, обновляти, коригувати, вилучати, сортувати інформацію, організовану спеціальним засобом у вигляді банку даних [1]. Найпоширеніші СУБД: dBase III Plus, FoxBase+, Clipper, Oracle, Acces, FoxPro, Paradox.

Крім перерахованих систем до складу прикладного ПЗ загального призначення слід віднести й інтегровані системи. Ці системи об'єднують у собі можливості текстових редакторів, графічних систем, електронних таблиць та систем управління базами даних. Головна перевага інтегрованих систем перед окремими системами прикладного ПЗ загального призначення полягає у тому, що вони створюють єдині правила роботи для користувача, тобто вони мають єдиний інтерфейс як при роботі з текстом, так і при роботі з електронними таблицями та ін. Найвідоміші серед них: Microsoft Works, Microsoft Office, Lotus SmartSuite, Perfect Office.

Прикладні програми спеціального призначення використовують у специфічній діяльності користувачів.

Функції специфічних систем залежать від їх призначення. Наприклад, для систем навчального призначення це можуть бути інструментальні засоби для розробки комп'ютерних уроків (гіпермедійні та гіпертекстові системи, авторські та інші системи), імітаційне моделюючі програми навчального призначення, програми для розробки та підтримки шкільного розкладу, педагогічні програмні засоби різного призначення та ін.

До складу прикладних програм спеціального призначення можна також віднести пакети прикладних програм (ППП), які широко використовуються, наприклад, для статистичної обробки даних, бухгалтерського обліку, розрахунку будівельних конструкцій та ін. [1].

Наявність у комп'ютері різноманітних ППП дозволяє розв'язувати

значну частину простих прикладних задач, майже без програмування. В цьому випадку завдання на розв'язування тієї чи іншої задачі записується у вигляді директиви спеціальною проблемно-орієнтованою мовою та повідомляється комп'ютеру.

## 1.2 Програми-трекери

Трекери, або програми стеження (trackware), призначені для віддаленого стеження за діями користувачів на комп'ютері або інших типах персональних пристроїв; є різновидом шпигунських програм. Програми стеження застосовуються кіберзлочинцями для віддаленого збору інформації. Також їх встановлюють роботодавці, які бажають контролювати діяльність своїх співробітників, і батьки, що стежать за дітьми – на корпоративні і домашні комп'ютерні пристрої відповідно [2].

Технічно досконалий трекер може реєструвати: будь-які дії користувача за комп'ютером, що відкриваються сторінки сайтів, введення персональних даних, логінів і паролів; системну інформацію, дані про апаратне забезпечення; запускаються програми; дзвінки і відеодзвінки в Skype та інших подібних програмах; роздруківку на принтерах; текстовий інтернет-трафік (листування в месенджерах); GPS-координати мобільних пристроїв.

Багато трекерів поширюються вільно (у вигляді демонстраційних версій) або за невелику плату. Їх легко знайти і протестувати. Автори представляють свої розробки як корисне програмне забезпечення для спостереження за власним комп'ютером. Програма спостереження запам'ятовує натискання клавіш, визначає паролі від акаунтів електронної пошти та Skype, популярних соціальних мереж. Зібрані відомості зберігаються на місці, але висилаються на вимогу. Налаштовано гнучке формування звітів, що дозволяє не вникати в зайві подробиці. Збирається статистика, складаються наочні діаграми. Можна вести пряме візуальне спостереження, записувати скріншоти в вигляді послідовності слайдів.

Опціонально скрипт буде працювати разом з операційною системою. Можлива робота через веб-сайт, яка не залишає слідів.

Класифікація програм стеження (trackware). Трекери класифікуються за таких підстав [2]:

1) З технічних пристроїв для:

- смартфонів;
- планшетів;
- ноутбуків; настільних комп'ютерів;
- серверів.

2) За операційною системою для:

- Windows;
- Linux;
- iOS;
- Android;

Універсальних версій практично не буває, оскільки трекер інтенсивно використовує особливості операційної системи. По можливостях – збір системної інформації (якщо трекер є частиною функціональності шпигунської програми), відстеження натискань клавіш, прослуховування, візуальне спостереження. Існують комплексні рішення, що дозволяють вести широкомасштабне спостереження. По спрямованості – для протизаконної діяльності, на легальних підставах [2]. Як правило, один і той же трекер можна використовувати в різних цілях.

### 1.2.1 Призначення програм стеження

Об'єктами впливу є приватні особи та організації. Спостерігача можуть цікавити як загальні аспекти поведінки, так і конкретні відомості. Основним завданням програм стеження є збір профілів користувачів для того, щоб визначити найбільш підходящу їм рекламу. Подібні дані добре продаються і монетизуються. Програми стеження (трекери, trackware) виконують такий збір нелегально, в той час як корпорації на кшталт Facebook та Google

роблять те ж саме на законних підставах [2].

Комп'ютери заражаються трекером так само, як будь-яким іншим шпигунським програмним забезпеченням. Становлять небезпеку підключення чужою флешки, відвідування сумнівних сайтів, запуск неперевірених файлів з електронної пошти або інтернету.

Крім цього, деякі програми стеження можуть бути частиною операційної системи персонального комп'ютера або смартфона.

Існує ймовірність зараження без безпосередньої участі користувача. Уразливості в операційних системах і прикладних програмах, «таємні ходи», навмисне залишені розробниками, стають відомими кіберзлочинцям і активно ними експлуатуються.

Як виявити програми стеження? Помітити присутність трекера можна прямо і побічно. До прямих методів належать виявлення антивірусом, експертне дослідження операційної системи і встановлених програм вручну. Перевантаження і перегрів процесора, необгрунтоване уповільнення роботи системи – привід запідозрити ураження шпигунською програмою.

Непряме свідчення чужого спостереження – явний витік інформації (наприклад, показ цільової реклами про предмет, про який ніхто знати не повинен). В такому випадку потрібні більш широкоформатні заходи, так як джерелом витоку часто виявляється співробітник, а не техніка.

### 1.2.2 Боротьба з прониклим трекером

Боротьба з уже прониклим трекером включає видалення небажаних файлів або (в крайньому випадку) переустановку операційної системи з максимальним збереженням цінної інформації [2].

Використання антивіруса значно знижує ризик зараження, але краще попередити проникнення trackware [1]. Для цього бажано регулярно проводити профілактику:

– своєчасно оновлювати програмне забезпечення комп'ютера і мережевого обладнання;

- використовувати сучасні антивірусні засоби з розширеними базами для детектування небажаних програм;
- використовувати персональний мережевий екран і контролювати мережеву активність додатків;
- правильно організувати роботу вдома або на робочому місці, не допускати до свого комп'ютера сторонніх, не використовувати чужі знімні накопичувачі;
- в організаціях – налагодити роботу служби безпеки.

Однак програми трекери використовуються не тільки зловмисниками, часто на робочі комп'ютери роботодавець встановлює програмне забезпечення для контролю за робочим часом співробітника.

### 1.3 Моніторинг часу роботи за комп'ютером

Сьогодні практично всі завдання виконуються з використанням програм, файлів, документів і додатків. Що вже говорити про роботу з інтернет – ресурсами. Не дарма їх вважають основними пожирателями часу. Тому моніторинг часу роботи за комп'ютером так важливий [3].

Його здійснюють спеціальні програмами обліку. Вони відображають діяльність персоналу з моменту включення комп'ютера до його виключення: раціональність використання ресурсів, відпрацьовані години, тривалість і причини перерв, прихід і відхід.

Відстеження часу є спірною темою. У наші дні багато людей використовують свій персональний комп'ютер або ноутбук для виконання роботи, а також для розваги. Один із способів з'ясувати, чи використовуєте ваші співробітники певну програму або веб-браузер занадто часто, – перевірити, скільки часу використовувалося кожен додаток.

Відстеження часу – це шлях до рівноваги між роботою та особистим життям [3]. Контроль роботи співробітників за комп'ютером – це використання різних методів для збору інформації про комп'ютерну активності працівника і місцезнаходження співробітників. Співробітники не

боятися обліку часу, вони бояться мікроуправління. Ось чому важливо, щоб обране програмне забезпечення давало зведені дані і дозволяло бачити загальну картину, не вдаючись у подробиці.

Давайте розглянемо три причини, чому необхідно використовувати моніторинг роботи персоналу – не тільки для вигоди бізнесу, а й для вигоди співробітників:

### 1.3.1 Спостереження за раціональним витрачанням часу співробітників для прийняття правильного рішення

Спостереження за раціональним витрачанням часу співробітників допомагає приймати правильні рішення про витрати

Підприємства часто розглядають використання внутрішніх ресурсів як «безкоштовне», а використання зовнішніх ресурсів – як інвестиції. Але це не ефективно. Припустимо, є співробітник, який ніколи раніше не створював веб-сайт, а бізнесу він потрібен. Якщо не брати до уваги значення часу співробітника, можна легко призначити йому цю роботу, і тоді він витратить 100 годин на вивчення, настройку і, нарешті, доопрацювання проекту, коли сторонній фахівець міг би виконати цю роботу за 10 годин.

Уявіть, що потрібно провести кампанію контент-маркетингу. Співробітник повинен писати дві статті в тиждень протягом 10 тижнів. Щоб залучити таку ж кількість клієнтів, також можна запустити кампанію в Інстаграм. Тож ми дізнаємося, яке рішення краще про витрати, якщо відстежимо час.

### 1.3.2 Відстеження часу означає краще управління робочим часом.

Природним припущенням є те, що менеджери використовують автоматичний облік робочого часу, щоб побачити, чи продуктивно працює їхня команда.

Використовуючи статистику роботи персоналу її можуть бачити не тільки керівники, а й самі співробітники. Вони порівнюють з параметрами

попереднього звітнього періоду і зроблять висновки, щоб підвищити свою продуктивність [3].

Контроль роботи за комп'ютером може бути спірним, але статистика говорить сама за себе: компанії, які відстежують використання робочого часу працівниками, повідомляють про підвищення продуктивності праці на 20% протягом першого місяця.

### 1.3.3 Відстеження часу допоможе розставити пріоритети.

Здатність кожної людини управляти своїм використанням часу має бути в пріоритеті. Автоматизація обліку робочого часу дозволяє бути відповідальним за те, зосередилися працівники на своїй найважливішій роботі. Без моніторингу немає реального способу побачити це. Надання співробітникам можливості самим переконатися в цьому дає їм свободу дій протягом робочого дня, що, природно, підвищує продуктивність праці.

Не знаючи, куди витрачатися час, ми не можемо вчитися на помилках і ставати більш продуктивними в майбутньому. З великими даними приходить велика відповідальність. Якщо налаштувати відстеження часу, то потрібно переконатися, що час використовується на благо, а не на зло. Програма для автоматизації обліку часу за комп'ютером потрібна для поліпшення балансу між роботою та особистим життям членів команди. Стівен Кові писав: «Потрібно не витратити час, а інвестувати в нього» [3].

Вибір на користь програмного забезпечення для обліку часу полягає в тому, що воно допоможе дізнатися, як співробітники використовують час, що послужить підвищенню загальної ефективності компанії. Можна визначити слабкі сторони співробітників, щоб допомогти їм виправитися.

### 1.3.4 Віддалений контроль персоналу можливий з додатком обліку часу

Недавні дослідження показують, що до 2022 року до 60% людей будуть працювати віддалено [3]. Моніторинг робить управління віддаленими працівниками набагато більш ефективним, оскільки керівники можуть

стежити за ходом виконання завдання, не запрошуючи щотижневих або щоденних звітів.

Крім того, дослідження показують, що, коли спостереження за співробітниками не є для них таємницею, вони стають більш організованими і мотивованими, щоб залишатися на плаву. Це пояснює миттєве підвищення продуктивності на 30%, про який компанії повідомляють через тиждень після того, як вони почали використовувати програмне забезпечення для відстеження часу [3].



## **2 ОПИС МОВИ ПРОГРАМУВАННЯ ДЛЯ ВИКОРИСТАННЯ ДОДАТКУ**

### **2.1 Обґрунтування вибору мови програмування**

#### **2.1.1 Опис мови Java**

Мова Java розроблена компанією Sun Microsystems, творцем якої був Джеймс Гослінг, і випущений в 1995 році в якості основних компонентів компанії Sun Microsystems – Java платформ (Java 1.0 [J2SE]) [4].

Станом на 2018 рік останньою версією Java Standard Edition є 8 (J2SE). З розвитком Java, і її широкою популярністю, кілька конфігурацій були побудовані для різних типів платформ. Наприклад: J2EE – додатки для підприємств, J2ME – для мобільних додатків.

Sun Microsystems перейменувала колишню версію J2 і ввела нові: Java SE, Java EE і Java ME. Введення в програмування Java різних версій підтверджувало знаменитий слоган компанії «Напиши один раз, запускай скрізь».

Історія створення мови Java починається в червні 1991 року, коли Джеймс Гослінг створив проект для використання в одному зі своїх численних сет-топ проектів. Мова, яка росла поза офісом Гослінга, як дуб, Oak – первинна назва Java до 1995 року, після надалі історія Java тривала під ім'ям Green, а пізніше була перейменована як Java.

Але офіційною датою створення мови Java вважається 23 травня 1995 року, після випуску компанією Sun першої реалізації Java 1.0. Вона гарантувала «Напиши один раз, запускай скрізь», забезпечуючи недорогу вартість на популярних платформах [4].

13 листопада 2006 року, Sun випустила більшу частину як вільне і відкрите програмне забезпечення відповідно до умов GNU General Public License (GPL). Після 8 травня 2007 року доля Java склалася інакше. Компанія завершила процес, роблячи все щоб вихідний код був безкоштовним і відкритим, крім невеликої частини коду, на яку компанія не мала авторських прав [7].

Переваги Java як мови програмування:

1. Об'єктно-орієнтована: в Java все є об'єктом. Доповнення може бути легко розширено, тому що вона заснована на об'єктній моделі.

2. Платформонезалежність: на відміну від багатьох інших мов, включаючи C і C ++, Java, коли була створена, вона не компілювалась в платформі конкретної машини, а в незалежній від платформи байт-кодi. Цей байт код поширюється через інтернет і інтерпретується в Java Virtual Machine (JVM), на якій він в даний час працює.

3. Проста: процеси вивчення та введення в мову програмування Java залишаються простими. Якщо Ви розумієте основні концепції об'єктно-орієнтованого програмування, то вона буде проста в освоєнні.

4. Безпечна: методи перевірки автентичності засновані на шифруванні з відкритим ключем.

5. Архітектурно-нейтральна: компілятор генерує архітектурно-нейтральні об'єкти формату файлу, що робить скомпільований код виконуваним на багатьох процесорах, з наявністю системи Java Runtime.

6. Портативна: архітектурно-нейтральна і не має залежності від реалізації аспектів специфікацій – все це робить Java портативною. Компілятор в Java написаний на ANSI C з чистою переносимістю, який є підмножиною POSIX.

7. Міцна: виконує зусилля, щоб усунути помилки в різних ситуаціях, спираючись в основному на час компіляції, перевірку помилок і перевірку під час виконання.

8. Багатопотокова: функції багатопоточності, можна писати програми, які можуть виконувати безліч завдань одночасно. Введення в мову Java цієї конструктивної особливості дозволяє розробникам створювати налагоджені інтерактивні додатки.

9. Інтерпретована: Java байт-код переводиться на льоту в машинні інструкції та ніде не зберігається. Роблячи процес більш швидким і аналітичним, оскільки зв'язування відбувається як додаткове з невеликою

вагою процесу.

10. Високопродуктивна: введення Just-In-Time компілятора, дозволило отримати високу продуктивність.

11. Поширена: призначена для розподіленого середовища інтернету.

12. Динамічна: програмування на Java вважається більш динамічним, ніж на C або C ++, так як вона призначена для адаптації до мінливих умов. Програми можуть виконувати велику кількість під час обробки інформації, яка може бути використана для перевірки і дозволу доступу до об'єктів на час виконання.

### 2.1.2 Опис мови C

Мова програмування C є однією з найпопулярніших і поширених мов. Вона являє собою скомпільовані загального призначення мови програмування з статичними типізація, розроблений в 1969-1973 Bell Labs програміст Денніс Річі (Денніс Річі).

Мова C часто згадується як "середнього рівня" або навіть "низькорівнева" мова програмування, так як вона поєднує в собі мови високого рівня з їх функціональністю і продуктивністю, і працює поруч з устаткуванням комп'ютера. В результаті, ми можемо маніпулювати даними на низькому рівні і використовувати високорівневу структуру для контролю роботи програми [5].

Спочатку мова C була призначена для написання операційної системи UNIX. Згодом C стала однією з найпопулярніших мов, а її основна Сфера застосування-системне програмування, зокрема, створення операційних систем, драйверів, різних утиліт, антивірусів і т.д. до речі, Linux в основному написаний на C. Однак, тільки системне програмування не обмежує використання цієї мови. Ця мова може бути використана в програмах будь-якого рівня, де важливі швидкість і продуктивність. Таким чином, ми можемо написати з використанням C і додатки, і навіть веб-сайти (з використанням технології CGI-загальний шлюз інтерфейс). Але, звичайно,

для створення GUI і веб-додатків, є більш підходящі інструменти і технології, але тим не менше, діапазон використання С досить широкий.

Незважаючи на великі можливості мови С досить проста в той же час. Вона не містить багато конструкцій, бібліотек, її легко освоїти і вивчати. Тому її часто вибирають як мову для вивчення в загальному програмуванні.

С – це скомпільована мова, що означає, що компілятор переводить початковий код до виконуваного файлу, який містить набір інструкцій на комп'ютері. Але різні платформи мають свої власні риси, тому скомпільовані програми можуть не можуть переходити з однієї платформи на іншу. Однак, на рівні вихідного коду, С програми мають портативність, а також наявність компіляторів, бібліотек та інструментів розробки для майже всіх загальних платформ дозволяє компілювати той же вихідний код С в додатках на цих платформах .

Розвиток С має великий вплив на розвиток мов програмування. Зокрема, його синтаксис став основою для таких мов, як С++, С#, Java, PHP, JavaScript. Особливої згадки повинно бути зроблено з відносин з С. С++ безпосередньо походить від С. Але згодом їх розвиток був окремо одна від одною, і навіть не було несумісності між ними. Стандарт С++ додав ряд конфліктуючих функцій С++ на мову С. В результаті, зараз обидві мови насправді самодостатні і розвиваються самостійно.

Основні особливості С:

- універсальність-один і той же код можна зібрати практично на кожній платформі (якщо є компілятор для нього);
- висока швидкість виконання;
- компактний розмір, малий обсяг скомпільованих файлів.

### **2.1.3 Опис мови С++**

Мова програмування С++ – високорівнева скомпільована мова програмування загального призначення з статичною типізацією, яка

підходить для створення найрізноманітніших програм. Сьогодні С++ є однією з найпопулярніших і найпоширеніших мов.

Вона корениться в мові С, яка була розроблена в 1969-1973 від компанії Bell Labs програмістом Деннісом Річі. На початку 1980-х років данський програміст Бйорн Страуструп, який у той час працював на Bell Labs, розробив С++ як розширення мови С. Насправді, на початку, С++ просто доповнювало мову С з деякими об'єктно-орієнтованими можливостями програмування. Ось чому Страуструп сам назвав її "С з класами" в першу чергу.

Згодом нова мова почала набирати популярність. Вона додала нові функції, які зробили це не просто доповненням до С, а абсолютно нову мову програмування. В результаті, "С з класами" було перейменовано на С++. І від тих пір, обидві мови почали розвиватися незалежно одна від одної.

С++ – це потужна мова, успадкована від ресурсів С із багатою пам'яттю [5]. Таким чином, часто С++ знаходить своє застосування в системному програмуванні, зокрема, при створенні операційних систем, драйверів, різних утиліт, антивірусів і т.д. до речі, Windows в основному написана на С++.

Але тільки системне програмування не обмежує використання цієї мови. С++ можна використовувати в програмах будь-якого рівня, де важливі швидкість та продуктивність. Часто використовується для створення графічних додатків, різних прикладних програм.

Також особливо часто використовується для створення ігор з насиченою візуалізацією. Крім того, останнім часом мобільний напрям набирає обертів, де С++ також знайшла застосування.

І навіть у веб-розробці, ви також можете використовувати С++ для створення веб-додатків або деяких допоміжних послуг, які обслуговують веб-додатки. У загальному С++ є мовою широкого використання, де можна створити практично будь-які програми.

C++ – це скомпільована мова, яка означає, що компілятор переводить початковий код на C++ у виконуваний файл, який містить набір інструкцій на комп'ютері. Але різні платформи мають свої власні риси, тому скомпільовані програми можуть не тільки переходити з однієї платформи на іншу і там вже запущено.

Однак, на рівні вихідного коду, програми C++ є більш портативними, якщо ви не використовуєте ніяких специфічних функцій для поточної операційної системи. І наявність компілятора, бібліотек та інструментів розробки практично для всіх типових платформ дозволяє компілювати однаковий початковий код у застосунках C++ на цих платформах.

На відміну від цього, C++ дозволяє писати програми в об'єктно-орієнтованому стилі, представивши програму як колекцію взаємодіючих класів і об'єктів [5]. Що спрощує створення великих застосувань.

У 1979-80, Бйорн Страуструп розробив розширення мови C-класу. У 1983 мову було перейменовано на C++.

У 1985 була випущена перша комерційна версія мови C++, а також перше видання книги, "мова програмування C++", яка представила перший опис мови за відсутності офіційного стандарту.

У 1989 була випущена нова версія мови C++ 2,0, яка включала ряд нових можливостей. Після цього мова розвивається відносно повільно до 2011 року. Але в 1998 була зроблена перша спроба стандартизувати мову організацією ISO (Міжнародна організація стандартизації).

Перший стандарт був названий ISO/IEC 14882:1998 або скорочено C++98. Пізніше в 2003 вийшла нова версія стандарту C++03.

У 2011 було опубліковано новий стандарт C++11, в якому міститься багато доповнень і велика кількість нових функцій. Невелике доповнення до стандарту, також відоме як C++14, було випущено в 2014.

Для розробки C++ програм потрібен компілятор – він переводить початковий код з C++ у виконуваний файл, який потім може бути запущено. Але на даний момент є багато різних компіляторів.

Вони можуть відрізнятися в різних аспектах, зокрема щодо реалізації стандартів.. Рекомендується для розвитку, щоб вибрати тих компіляторів, які розробляють і реалізують всі останні стандарти.

#### **2.1.4 Опис мови C#**

На даний момент мова програмування C# є однією з найпотужніших, швидко зростаючих і популярних мов в IT-індустрії. В даний час на ній написані різні додатки: від невеликих настільних програм до великих веб-порталів і веб-сервісів, що обслуговують мільйони користувачів кожен день.

У порівнянні з іншими мовами C# досить молода, але в той же час вона вже пройшла довгий шлях. Перша версія мови прийшла з випуском Microsoft Visual Studio .Net в лютому 2002. Поточна версія мови C# 7.0, яка була випущена 7 березня 2017, разом з Visual Studio 2017.

C# – мова з C-подібним синтаксисом і схожа на C++ і Java. Тому, якщо ви знайомі з однією з цих мов, то буде простіше освоїти C# [6].

C# є об'єктно-орієнтованою і в цьому відношенні багато чого було перейнято у Java і C++. Наприклад, C# підтримує поліморфізм, успадкування, перевантаження оператора та статичний ввід. Об'єктно-орієнтований підхід дозволяє вирішувати завдання побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків. І C# продовжує активно розвиватися, і з кожною новою версією є все більше і більше цікавих функцій, таких як лямбда, динамічні посилання, асинхронні методи і т. д.

Мова C# є найбільш відомою новинкою в області створення мов програмування. З'явившись на світло в надрах Microsoft, вона з перших своїх кроків отримала потужну підтримку. Мова визнана міжнародним співтовариством.

Компілятори Microsoft будуються відповідно до міжнародних стандартів мови.

Керівником групи, що створює мову C#, є співробітник Microsoft

Андреас Хейлсберг. Як зазначав сам Андреас Хейлсберг, C# створювалась як мова компонентного програмування, і в цьому одне з головних достоїнств мови, спрямоване на можливість повторного використання створених компонентів. Створювані компілятором компоненти є саме документовані, крім коду містять метайнформацію, що описує компоненти, і тому можуть виконуватися на різних платформах [7].

В якості мови програмування для виконання дипломного проекту було обрано мову C#. Відзначимо наступні важливі фактори, які вплинули на вибір даної мови програмування:

- C# створювалась розвивається паралельно з каркасом Framework.Net і в повній мірі враховує всі його можливості;
- C# є повністю об'єктно-орієнтованою мовою;
- C# є потужною об'єктною мовою з можливостями успадкування та універсалізації;
- C# є спадкоємцем мови C++. Загальний синтаксис, загальні оператори мови полегшують перехід від мови C++ до C#. Зберігши основні риси свого батька, мова стала простіше і надійніше;

Завдяки каркасу Framework.Net, який став надбудовою над операційною системою, програмісти C# отримують переваги роботи з віртуальною машиною. Framework.Net представляє потужну платформу для створення додатків. Можна виділити такі основні можливості:

- багатомовна підтримка, платформа заснована на загальній мові виконання (CLR), яка дозволяє .net підтримувати декілька мов: разом з C#, це також VB.NET, C++, F#, а також різні діалекти інших мов, пов'язаних з .NET, такі як Delphi.NET. Під час компіляції код на будь-якій з цих мов складається в збірку на загальну проміжну мову (CIL), свого роду .NET асемблера. Таким чином, ми можемо зробити окремі модулі одного і того ж додатка в окремих мовах;
- кросплатформеність, .NET – це портативна платформа (з деякими обмеженнями). Наприклад, остання версія .NET Framework наразі



підтримується на більшості сучасних операційних систем Windows (Windows 10/8.1/8/7/Vista). І з моно проекту, ви можете створювати програми, які будуть працювати на інших операційних системах Linux, у тому числі Android і IOS мобільних платформах [6];

- потужна бібліотека класів, .NET – це бібліотека класів, яка є спільною для всіх підтримуваних мов. І будь-яка програма, яку ми збираємося писати в C# – текстовий редактор, чат, або складний веб-сайт в деякому роді, ми будемо працювати з бібліотеками .NET [8];

- різноманітні технології, Загальні мовні виконавчі файли та бібліотека базових класів є основою для цілого стеку технологій, розробники яких можуть використовувати їх для створення додатків. Наприклад, технологія ADO.NET призначена для роботи з базами даних у цьому стеку технологій. Створювати графічні програми з багатою інтерфейсною технологією WPF. Для створення Websites – ASP.NET і інші. [9];

Також примітна особливість C# та .NET Framework, автоматична колекція сміття [7]. Це означає, що в більшості випадків ми не повинні дбати про звільнення пам'яті, на відміну від C++. Вищезгадане мовне середовище буде викликати колектор сміття і очищати пам'ять.

## 2.2 Вибір середовища розробки

При розробці і реалізації програмного продукту було використано середовище візуальної розробки Visual Studio 2019.

Visual Studio – це інтегроване середовище розробки (IDE) від компанії Microsoft.

За допомогою Visual Studio можна розробляти:

- класичні додатки для комп'ютера під керуванням операційної системи Windows;
- мобільні додатки (Windows, iOS, Android);
- web-додатки;

- хмарні додатки;
- різні розширення для Office, SharePoint, а також створення власних розширень для Visual Studio;
- ігри;
- бази даних SQL Server і SQL Azure.

У Visual Studio можна використовувати наступні технології і мови програмування: .NET, Node.js, C, C #, C ++, Python, Visual Basic, F #, JavaScript.

Архітектура операційної системи захищає програми від ушкодження одна одної і самою операційною системою. При цьому використовується відмовостійка структурована обробка особливих ситуацій на всіх архітектурних рівнях, включаючи відновлювану файлову систему NTFS і забезпечує захист за допомогою вбудованої системи безпеки і вдосконалених методів управління пам'яттю.

Система надає можливості для роботи в багатьох країнах світу на національних мовах, досягається за допомогою застосування стандарту ISO Unicode.

Середу Visual Studio 2019 можна встановити і працювати в ній на наступних операційних системах (перераховані офіційно підтримувані версії):

- Windows 7 з Service Pack 1;
- Windows 8.1 (з оновленням 2919355);
- Windows 10 (1703 і вище);
- Windows Server 2012 R2 (з оновленням 2919355);
- Windows Server 2016 (Standard і Datacenter);
- Windows Server 2019 (Standard і Datacenter).

Мінімальні вимоги до обладнання:

- процесор з тактовою частотою не нижче 1,8 ГГц. Рекомендується використовувати як мінімум двоядерний процесор;
- 2 ГБ оперативної пам'яті, рекомендується 8 ГБ (якщо

встановлювати на віртуальну машину, то мінімум 2.5 ГБ);

– вільного місця на жорсткому диску від 800 мегабайт до 210 гігабайт, залежно від встановлених компонентів. Також Microsoft рекомендує встановлювати Visual Studio на SSD диск.

– відеоадаптер з мінімальним дозволом 1280 на 720 пікселів (для оптимальної роботи Visual Studio рекомендується дозвіл 1366 на 768 пікселів і вище).

Для розробки програмного продукту було вибрано програмне середовище Visual Studio 2019 з багатьох причин.

По-перше, Visual Studio 2019 є інтегрованим середовищем швидкої розробки програмного забезпечення для роботи під Microsoft Windows, на даний момент найпоширенішою операційною системою.

По-друге, середовище програмування Visual Studio 2019 року було обрано завдяки можливості безпроблемного зв'язку з офісним пакетом Microsoft Office. Досить проста установка зв'язку з цим пояснюється тим, як уже зазначалося вище, що навколишнє середовище Visual Studio 2019 – Це один з найвідоміших засобів створення Windows-додатків і зв'язку з продукцією Microsoft встановлюються досить легко.

По-третє, Visual Studio 2019 року було обрано через доступності цієї програми будь-якому користувачеві.

По-четверте, Visual Studio 2019 відзначається своєю стабільністю, швидкістю і низькими вимогами до апаратного забезпечення ЕОМ.

### **3 АНАЛІЗ НАЙПОПУЛЯРНІШИХ ДОДАТКІВ ДЛЯ МОНІТОРИНГУ ЧАСУ**

Люди, які працюють в бізнесі, повинні вести облік того, скільки часу займає кожен проект клієнта та бізнес-завдання, щоб мати змогу уникнути заниження плати за великі проекти або перевантаження роботою. Тому є багато причин відстежувати свій час: від запису оплачуваних годин до підвищення особистої продуктивності та оптимізації бізнес-процесів. А програми для відстеження часу допомагають зробити цю практику легшою, точнішою та набагато послідовнішою. Підключаючи додаток для відстеження часу до інших інструментів автоматизується відлік часу.

Найкращі трекери часу мають як вбудований таймер, так і можливість редагувати або додавати час вручну. Вони дозволяють приписувати записи часу різним завданням і проектам, а також надають детальні звіти та навіть функції виставлення рахунків для перетворення відстежуваного часу на дохід.

Розглянувши кілька десятків інструментів, було виділено п'ять найкращих програм для відстеження часу для окремих людей і невеликих команд. Найкращі програми для відстеження часу Toggl Track для програми для відстеження вільного часу HourStack для інтеграції з програмним забезпеченням для керування проектами команди. Для вчасного автоматизованого відстеження часу використовують TrackingTime, для іншої візуалізації часу RescueTime для зменшення відволікань.

Незалежно від того, працює людина самостійно чи в невеликій команді, програмне забезпечення для відстеження робочого часу може дати повний огляд щоденної, щотижневої, місячної та річної роботи. Мабуть, найважливіше те, що відстеження часу проекту може допомогти отримати гроші, дозволяючи бути впевненими щодо своїх рахунків, а не намагатися оцінити, скільки часу працювали після факту. Якщо людина є частиною команди, відстеження часу може допомогти відповісти на запитання: «Над

чим він працював цього місяця?» Оцінюючи найкращі програми для відстеження часу, буди враховані такі критерії:

- відстеження в реальному часі (майже всі програми для відстеження часу дозволяють відстежувати в режимі реального часу, тобто вони дають працюючий годинник, який запускається, коли починається завдання, і який можна призупинити або зупинити, коли закінчується);
- можливість редагувати відстежуваний час або вручну додавати часові блоки (найкращі програми дозволяють виправляти час, який відстежується після факту, наприклад, якщо випадково залишився таймер увімкненим, коли вирушаєте на обід, коли повинен мати можливість редагувати журнал часу, щоб відняти стільки хвилин, скільки не працювали);
- особливості звітності (найпотужніші програми для відстеження часу пропонують інформаційні панелі та звіти, які дають змогу розбивати щоденний, тижневий або місячний час, витрачений на проект, на людину чи на клієнта);
- можливість створити рахунок-фактуру або експортувати дані (для підсумовування всього часу, витраченого на роботу, та необхідність мати можливість вжити заходів, усі ці програми дозволяють або автоматично створювати рахунки-фактури на основі даних відстеження часу, що зберігаються в програмі, або експортувати цю інформацію у PDF-файли, CSV-файли чи Google Таблиці, щоб поділитися ними з потрібними людьми);
- кілька точок доступу (найкращі інструменти відстеження часу роблять запуск таймера легким і зручним; вони мають універсальність, щоб дозволити запускати таймер у будь-якому зручному місці: у веб-переглядачі, настільній програмі чи мобільному пристрої).

Аналіз програм зосереджено саме на пошуку засобу відстеження часу для фрілансерів, окремих людей і невеликих команд. В додаток включено два автоматичні інструменти відстеження, які відстежують дії користувачів, але дані доступні лише для очей кожного користувача.

### 3.1 Аналіз найкращого додатку для відстеження вільного часу

Інструмент відстеження часу Toggl Track має щедрий безкоштовний рівень обслуговування, який надає доступ до всіх програм (Android, iOS, Windows, macOS, Linux, Інтернет і розширення для браузера) і включає всі основні функції, які ви очікуєте від відстеження часу додаток (рис. 1).

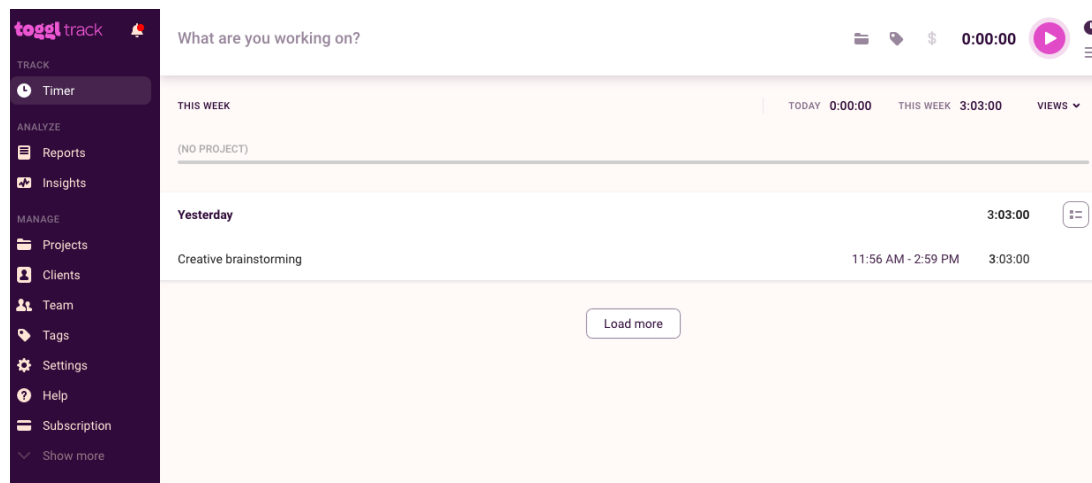


Рисунок 1 – Інструмент відстеження часу Toggl Track

Toggl Track працює так легко, перш за все тому, що він не вимагає багато вимог, коли починає вимірювати час виконання завдання. У багатьох програмах для відстеження часу потрібно ввести клієнта, проект і завдання у свій обліковий запис, перш ніж програма дозволить відстежувати час. Якщо ні, то доведеться на час відкласти завдання та ввести їх. За допомогою Toggl Track можна відразу відстежувати час, переглядати записи відстежуваного часу у форматі сітки чи календаря та піклуватися про деталі пізніше [10].

Крім того, при використанні Toggl Track через розширення браузера Chrome або Firefox, кнопка таймера Toggl з'являється практично в кожній веб-програмі, яку можна собі уявити, від Google Docs (та інших програм Google) до Help Scout. Оскільки завжди можна побачити кнопку Toggl Track під час роботи, то є постійне нагадування відстежувати свій час.

Toggl Track має одні з найбільш детальних параметрів, наприклад, у розширенні Toggl Track Chrome можна автоматично запускати та зупиняти відстеження часу залежно від того, відкритий браузер чи ні. Він також має відмінне виявлення простою, що допомагає підтримувати високу точність ваших записів. Коли програма виявляє, що комп'ютер був неактивний, але таймер все ще працює, вона пропонує способи виправити записаний блок часу. Таким чином, якщо перервана під час роботи 20-хвилинною розмовою, Toggl Track може вибити ці 20 хвилин із запису. У Toggl Track навіть є налаштування Pomodoro Technique, тож можна попрацювати певний час над проектом, який потребує зосередженості, а потім зробити коротку перерву з простими нагадуваннями, схожими на секундомір.

Для ще більшої гнучкості можна використовувати інтеграцію Toggl Track Zapier, щоб підключити Toggl Track до улюблених програм, для того щоб мати змогу починати нові записи часу в Toggl Track, коли починаються події календаря, додавати нові записи до електронної таблиці тощо.

Незважаючи на те, що безкоштовний план простий і легкий у використанні для базового відстеження робочого часу та звітування, якщо необхідно провести детальну статистику роботи, збережених звітів або оплачуваних годин, то за ці особливості потрібно перейти на Premium.

Toggl Track безкоштовний для п'яти користувачів; 9 дол. США за користувача/місяць при щорічній оплаті за план Starter, який включає округлення часу, тарифи, що підлягають оплаті, збережені звіти тощо.

Якщо необхіден простий додаток для відстеження часу без зайвих наворотів, можна розглянути Clockify. Хоча безкоштовний план не пропонує такий рівень функціональності, як Toggl Track, він включає необмежену

кількість відстеження, необмежену кількість користувачів, необмежену кількість проектів і необмежену кількість звітів – і все це безкоштовно [10].

### 3.2 Аналіз кайкращого додатку для відстеження часу для інтеграції з інструментом керування проектами вашої команди

Додаток HourStack – засіб відстеження часу проекту, який легко інтегрується з улюбленим інструментом керування проектами, щоб допомогти команді як планувати, так і відстежувати час (рис. 2).

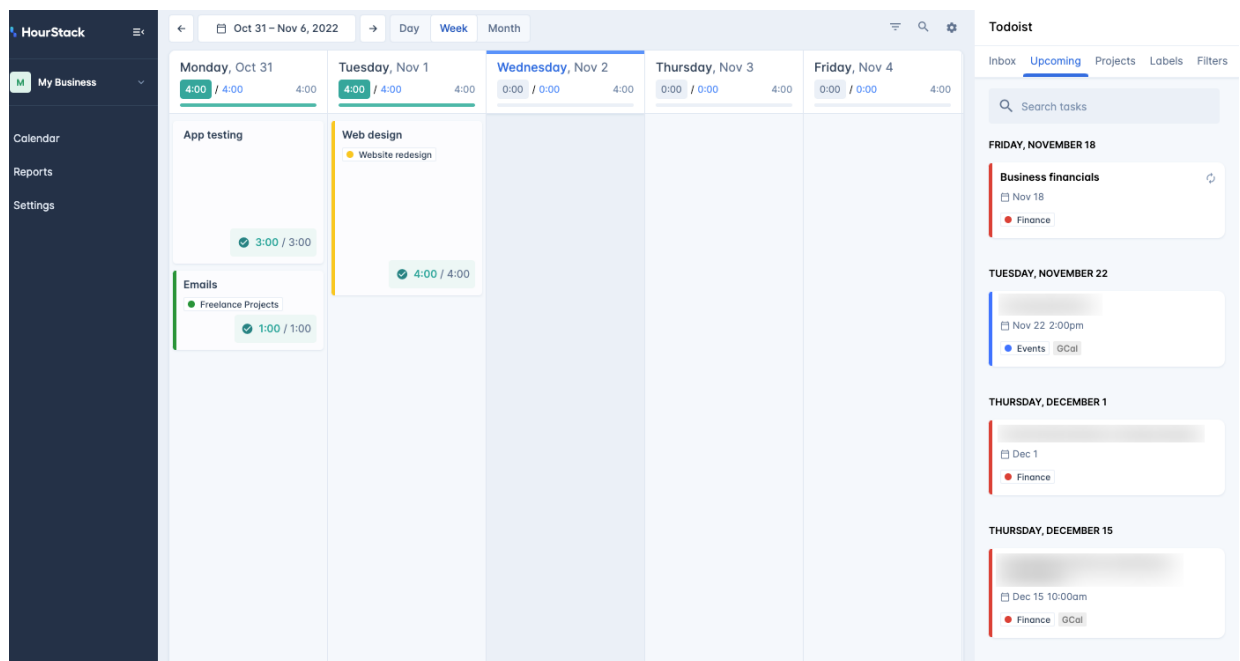


Рисунок 2 – Додаток HourStack

Напевно, найприємнішим сюрпризом під час тестування додатка, HourStack є елегантний та простий у використанні комбінацією календаря та відстеження часу. Додаток має вбудовану інтеграцію з такими популярними інструментами управління проектами, як Asana, Google Tasks, Trello, Todoist і навіть Google Calendar. Підключившись до однієї з цих програм, можна переглядати завдання на бічній панелі в HourStack – і перетягувати їх безпосередньо в календар HourStack, щоб планувати навантаження команди.



Інтеграція працює в обох напрямках, тож можна позначати виконані завдання з HourStack, і вони оновлюватимуться у менеджері проектів.

Сила HourStack полягає в його простоті. Немає ризику бути перевантаженим або відволікатися на складні налаштування чи інтерфейси. Сторінка «Календар» показує заплановані завдання та дозволяє планувати та записувати робочий час. Сторінка звітів, яку можна фільтрувати за клієнтами, проектами та мітками, відображає заплановані та зареєстровані години разом із кількістю завдань, які виконані. А за допомогою розширень браузера для Chrome, Firefox і Microsoft Edge можна швидко додати або почати відстежити час проекту, не вводячи завдання чи проект.

Якщо потрібно підключити HourStack до інструменту, який не підтримується нативно, можна створити власні інтеграції HourStack за допомогою Zapier. Підключивши HourStack до тисяч програм, можна координувати відлік часу, як би це було найбільш корисним.

HourStack не підходить: для тих, хто хоче програму для комп'ютерів і/або мобільних пристроїв. HourStack працює у веб-переглядачі та за допомогою розширень браузера, і хоча він дуже швидко реагує на мобільні пристрої, деякі користувачі можуть відчувати себе обмеженими через обмежені точки доступу.

Ціна HourStack 9 доларів США на місяць для одного користувача, 12 доларів США за користувача на місяць для команд.

Аналогічні програми для відстеження часу, оптимізовані для керування командними проектами – Harvest і Everhour. Обидві програми створено для надійної та легкої обробки робочих навантажень спільної роботи. Harvest працює в автономному режимі та має надійний безкоштовний план, і обидва інструменти мають кілька нативних інтеграцій, які дозволяють легко включити інструмент керування проектами вашої команди для планування [10].

### 3.3 Аналіз найкращого додатку для автоматичного відстеження часу

Найкращі програми для відстеження часу максимально полегшують точний запис того, як витрачається час. Timely – це відносно нова програма відстеження часу від Memory AS, яка створює «спогади» про кожне програмне забезпечення та веб-сайт, яким користуєтеся протягом дня. Потім у будь-який зручний час можна перетягнути ці Спогади у призначені їм завдання чи проекти, щоб створити таблиці.

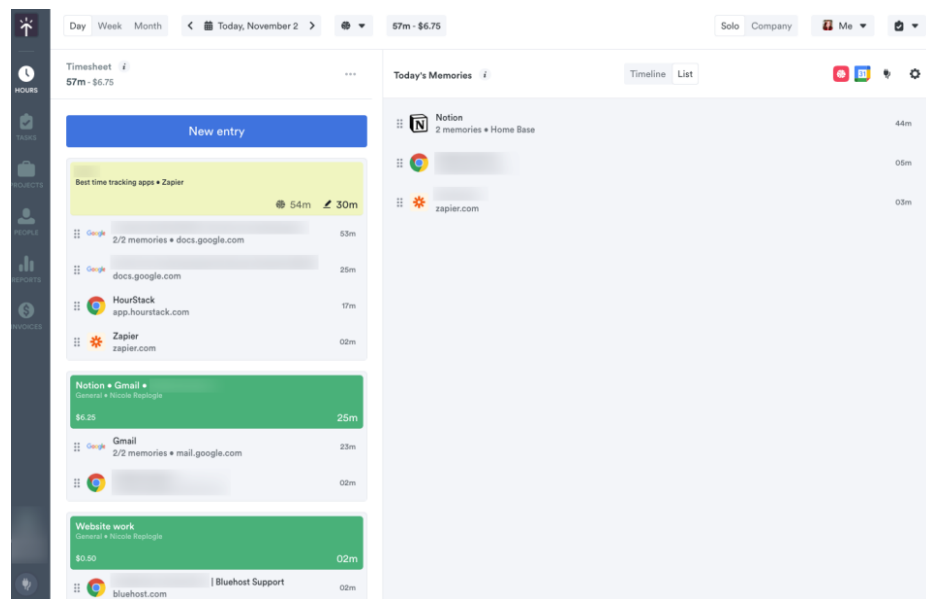


Рисунок 3 – Додаток HourStack

В організації, подібний інструмент може здатися трохи страшним, але Timely – це не додаток для стеження за працівниками. Усі дані, які він збирає, є конфіденційними, і лише користувач вирішує, скільки даних розміщувати у таблиці. І якщо не має бажання, щоб програма відстежувала діяльність у будь-який момент, то можна легко вимкнути програму на панелі меню.

Чудова особливість Timely полягає в тому, що нею легко почати користуватися. Після завантаження настільної програми буде запропоновано підключитися до календаря та/або інструментів для конференцій, щоб Timely

міг включити зустрічі у робочий розклад. Також можна легко додавати проекти, завдання та клієнтів і встановлювати погодинну ставку для людей і проектів.

Потім просто дозволити програмі працювати у фоновому режимі. Немає потреби запускати та зупиняти таймер – просто необхідно приділити кілька хвилин наприкінці дня, перетягуючи дії у відповідні проекти у своєму таблиці, і можна мати точний і детальний облік того, як витратили час.

Timely пропонує дуже докладні та індивідуальні звіти. Легко переглядайте неоплачені години та звіти для всього робочого простору або лише для окремого проекту. Також можна створювати шаблони звітів для індивідуальних переглядів, відфільтрованих за клієнтами, проектами, людьми, тегами або командами. Потім, коли приходить час виставляти рахунок, Timely підключається до QuickBooks, щоб створювати рахунки та надсилати їх у QuickBooks Online, коли вони будуть готові до надсилання.

А якщо ви любите планувати наперед, Timely також пропонує функцію планування завдань за додаткові 5 доларів на місяць. Ви можете планувати завдання за виконавцем або проектом. Потім щодня перетягуйте свої дії у відповідні завдання у своєму таблиці, щоб порівнювати запланований із відстеженим часом для кожного завдання.

Щоб отримати максимальну віддачу від програми, можна інтегрувати Timely із Zapier. Можна підключити Timely до решти свого технічного стеку, щоб оптимізувати такі процеси, як створення нових клієнтів або проектів у Timely.

Користувачі, які не хочуть, щоб інструмент відстежував кожен їхній рух на комп'ютері. Крім того, якщо багато працювати поза кадром, Timely може не підійти; він зосереджений головним чином на відстеженні та класифікації комп'ютерної роботи.

Своєчасна ціна: \$8/користувач/місяць для початкового плану (до 50 проектів і 3 команд). План Premium коштує 14 доларів США за користувача на місяць для необмеженої кількості проектів і команд.

### 3.4 Аналіз найкращого додатку для відстеження часу для іншої візуалізації часу

TrackingTime – це інструмент для спільного відстеження часу, який виглядає інакше, ніж багато інших програм для відстеження часу. Він представляє час у блоках або вікнах, як якщо б цей робочий час був подією у вашому календарі. Це не єдина програма, яка таким чином розподіляє ваш час, але вона пропонує найбільше можливостей для налаштування. Наприклад, ви можете вибрати перегляд дня, тижня, місяця або групи (рис.4).

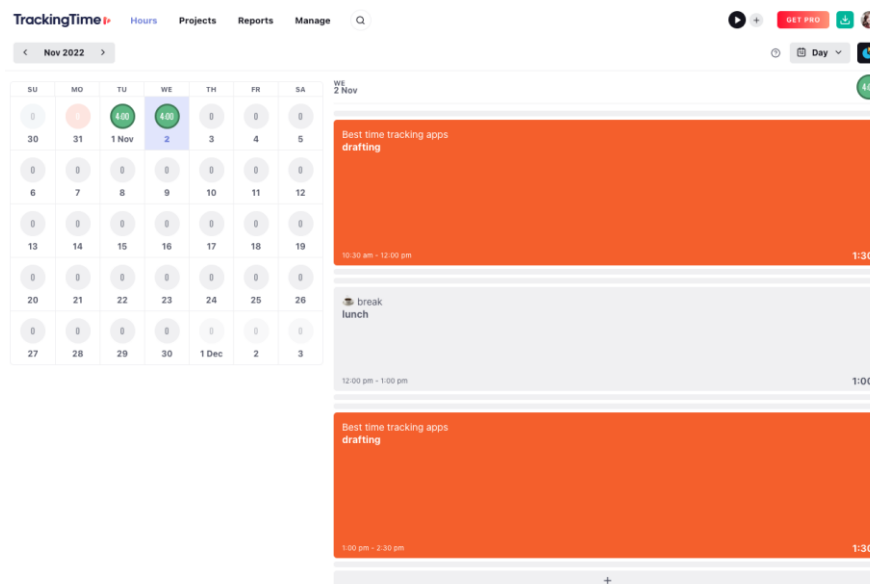


Рисунок 4 –TrackingTime

Щоденний перегляд має найкорисніший макет, який я бачив, показуючи повний місячний календар ліворуч (з кожним днем відображається піктограма, яка показує кількість відстежуваного часу), а день, який ви вибираєте, праворуч, позначений кольором і впорядкований за блоками відстежуваного часу.

TrackingTime також приділяє більше уваги плануванню вашого часу перед початком роботи порівняно з іншими програмами. Наприклад, ви

можете запланувати часові блоки у своєму календарі для завдань, над якими ви збираєтеся працювати пізніше протягом тижня, або для повторюваних подій, як-от зустрічей.

А розширення Chrome від TrackingTime інтегрується з декількома інструментами продуктивності, дозволяючи запускати таймер безпосередньо з таких програм, як Airtable, Asana та Notion. Або можна підключити TrackingTime до Zapier, щоб допомогти йому спілкуватися з усіма іншими програмами, якими ви користуєтеся.

TrackingTime не підходить для користувачів, яким не байдуже блокування часу чи планування певних завдань заздалегідь.

TrackingTime безкоштовний до 3 користувачів; 5 доларів США за користувача на місяць за план Pro для необмеженої співпраці, розширених звітів і функцій керування часом.

### **3.5 Аналіз найкращого додатку для відстеження часу для зменшення відволікань**

RescueTime – це програма для автоматичного відстеження часу, яка має версії практично для будь-якої платформи. Щоб почати використовувати RescueTime, спочатку необхідно встановити невелику програму моніторингу на свій комп'ютер. Це програмне забезпечення працює у фоновому режимі, спостерігаючи за використанням комп'ютера, і підтримує RescueTime Assistant, «тренера цифрового здоров'я».

Помічник доступний на панелі меню робочого столу, панелі завдань або системному лотку та допомагає керувати щоденною роботою, цілями та звітами. Наприклад, він може сповістити, коли робітник відволікається і виконує багатозадачність більше, ніж зазвичай.

RescueTime розроблено, щоб допомогти зосередитись і взятися за глибоку роботу – проекти з високою вартістю, складні розумово. RescueTime називає цей тип роботи «фокусованою роботою». Можна визначити дії, які

кваліфікуються як зосереджена робота, у своїх налаштуваннях і встановити цілі щодо того, скільки зосередженої роботи необхідно виконати за день.

Під час сеансу «Фокус-сесію» RescueTime блокує всі веб-сайти, які визначені як особиста діяльність (наприклад, сайти соціальних мереж або YouTube). Він відстежує, наскільки добре концентрується користувач, і наприкінці заняття надасть детальний звіт.

Користувач повинен ознайомитись з усіма термінами та з тим, як RescueTime визначає та класифікує певні дії. Але як тільки користувач відчує себе комфортно зі словниковим запасом і налаштуваннями, йому більше не доведеться возитися з чимось іншим і він зможе відразу зануритися у глибоку роботу.

Крім того, це єдина програма в цій підбірці, яка це робить не дозволяє експортувати години або автоматично створювати рахунок-фактуру. RescueTime призначений для людей, які хочуть відстежувати відпрацьовані години для власної вигоди, дізнаватися більше про свої робочі звички та як оптимізувати свою продуктивність.

Zapier може зв'язати RescueTime з усіма іншими програмами, якими користується користувач, тож є можливість витратити більше часу на цю роботу.

RescueTime не підходить для користувачів, які хочуть відстежувати час для окремих проектів і завдань. RescueTime зосереджено на зменшенні відволікань і збільшенні «часу зосередженості» без розмежування між конкретними видами діяльності, тому він розподіляє відстежувані дії лише на три категорії: зосереджена робота, інша робота та особиста діяльність.

RescueTime безкоштовний на план Lite для окремих осіб, який не включає блокування активності, розширені звіти, встановлення цілей або введення часу офлайн. RescueTime для команди починається 9 доларів США за користувача на місяць [10].

## 4 ОПИС ПРОГРАМНОГО ЗАСТОСУНКУ

Вибір додатку залежить від потреб користувачів і розвитку технологій. Він є корисним для керування часом, підвищення продуктивності та відстеження розподілу часу між різними завданнями.

Додаток повинен надавати різні функції, такі як реєстрація робочого часу, статистика витраченого часу на різні діяльності, сповіщення про перевищення встановлених обмежень часу тощо. Вони можуть бути корисними як для індивідуального користувача, так і для команд або організацій, які бажають відстежувати та аналізувати використання часу.

Однак, важливо враховувати приватність та безпеку при використанні таких додатків. Перед встановленням будь-якого додатку, ретельно перегляньте дозволи та політику конфіденційності, щоб переконатися, що дані будуть оброблятися відповідно до ваших вимог та нормативів.

Визначення завдань дозволяє встановити їх важливість. Люди без чітких цілей зазвичай витрачають свій час і енергію на непотрібні справи. Такі люди часто стикаються з невизначеністю та неточністю. Встановлення цілей допомагає нам відділитися від вимог, що накладаються ситуацією або іншими людьми, і досягти тих цілей, які дійсно важливі.

Цей етап є обов'язковим, і при виділенні задач і дій слід вказувати час, необхідний для їх виконання, а також пріоритети задач, щоб визначити, які є найважливішими для досягнення цілі. Структурований робочий день може включати наступні етапи:

- Робота над задачами.
- Відстеження використання часу.
- Планування задач на наступний день.

Самоаналіз поглиначів часу щотижня також є важливим кроком. Постійний аналіз поглиначів часу допомагає постійно підвищувати свій коефіцієнт росту.

Також слід проводити самоаналіз ефективності, коефіцієнту росту та показників в кінці виконання кожної цілі. Аналіз ефективності (рис. 4) є більш повним і обґрунтованим, оскільки в ньому аналізуються більша кількість показників та факторів, що дозволяє отримати ще більше результатів в роботі.

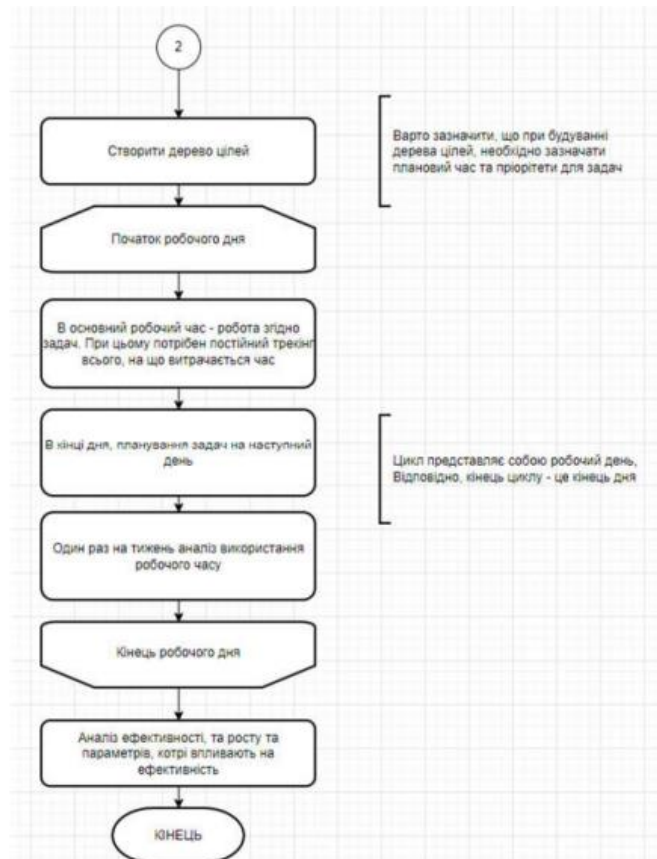


Рисунок 4 – Аналіз ефективності

#### 4.1 Технологія WPF

Windows Presentation Foundation (WPF), з кодовою назвою Avalon, є графічною (презентаційною) підсистемою, яка входить до складу .NET Framework 3.0 і тісно пов'язана з XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista і доступна для установки в Windows XP Service Pack 2 і Windows Server 2003.

Ця технологія є першим суттєвим оновленням середовища користувацького інтерфейсу з моменту випуску Windows 95. Вона включає



нове ядро, яке призначене замінити GDI і GDI+, що використовуються в поточній платформі Windows. WPF є високорівневим об'єктно-орієнтованим функціональним фреймворком, що дозволяє створювати двовимірні та тривимірні інтерфейси.

Протягом багатьох років розробники на платформі .NET створювали настільні програми за допомогою технології Windows Forms. Windows Forms є вищезазначеною, стабільною технологією, яка виконує основні завдання сучасних бізнес-додатків. Однак, ця технологія тісно пов'язана з архітектурою операційної системи, і деякі завдання, що вимагають швидкості та мінімального програмування, вимагають великих зусиль і витрат часу.

WPF змінила підхід до розробки настільних програм. Використовуючи базову технологію DirectX, Microsoft надала розробникам можливість швидко створювати складні елементи керування і повністю керувати процесом візуалізації. Тепер, для створення привабливої кнопки з анімаційними ефектами, можна обійтися без написання жодного рядка коду на C#.

Завдяки WPF, робота з мультимедійним вмістом стала набагато простішою, модель зв'язування даних, друку та роботи з документами розширилася. WPF поєднує найкращі аспекти традиційної розробки для Windows з безліччю нововведень, що дозволяє будувати інтерфейси користувача з багатою графікою. Крім того, ця технологія має всього дві прості вимоги: володіння мовою C# та вміння користуватися бібліотеками .NET Framework. Архітектура WPF наведена на схемі, представленій на рис. 5.

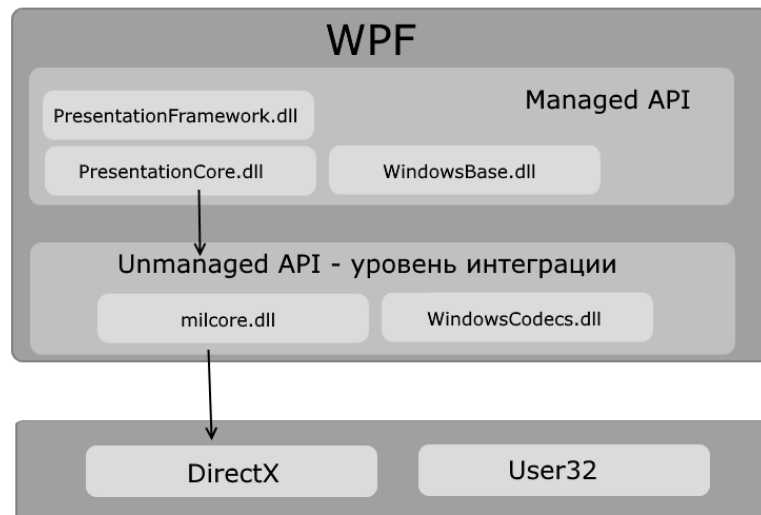


Рисунок 5 – Архітектура WPF

Як можна побачити на рис. 5, архітектура WPF складається з двох рівнів: управлінні рівень API та некерований рівень API (інтеграція з DirectX). Управлінні рівень API (Managed API) включає код, що виконується під керуванням середовища виконання .NET - Common Language Runtime [9]. Цей API визначає основний функціонал платформи WPF і складається з наступних компонентів:

- presentationframework.dll: містить реалізації компонентів і елементів управління, які використовуються при побудові графічного інтерфейсу;
- presentationcore.dll: містить базові типи для більшості класів з PresentationFramework.dll;
- windowsbase.dll: містить допоміжні класи, які використовуються в WPF і можуть бути використані поза платформою; Некерований рівень API (Unmanaged API) використовується для інтеграції з DirectX:
- milcore.dll: забезпечує інтеграцію компонентів WPF з DirectX. Цей компонент написаний на некерованій мові програмування (C / C++) для взаємодії з DirectX;

- `windowscodecs.dll`: бібліотека, яка надає низькорівневу підтримку для роботи з зображеннями в WPF.

Ще нижче власне знаходяться компоненти операційної системи і DirectX, які виконують візуалізацію компонентів програми, або виконують іншу низькорівневу обробку. Зокрема, за допомогою низькорівневого інтерфейсу Direct3D, який входить до складу DirectX, відбувається трансляція.

Тут також на одному рівні знаходиться бібліотека `user32.dll`. І хоча вище говорилося, що WPF не використовує цю бібліотеку для рендерингу і візуалізації, проте для ряду обчислювальних задач (що не включають візуалізацію) дана бібліотека продовжує використовуватися. WPF є частиною системи .NET і розвивається разом з фреймворком .NET і має ті ж версії. Перша версія WPF 3.0 вийшла разом з .NET 3.0 і операційною системою Windows Vista в 2006 році. З тих пір платформа послідовно розвивається. Остання версія WPF 4.6 вийшла паралельно з .NET 4.6 в липні 2015 року, ознаменувавши дев'ятиріччя даної платформи. Переваги, які надає WPF розробнику:

Використання традиційних мов .NET-платформи – C# і VB.NET для створення логіки додатка [8].

Можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, заснованому на xml і представляє альтернативу програмному створенню графіки та елементів управління, а також можливість комбінувати XAML і C# / VB.NET.

Незалежність від дозволу екрану: оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним дозволом.

Нові можливості, яких складно було досягти в WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін.

Хорошу взаємодію з WinForms, завдяки чому, наприклад, в додатках WPF можна використовувати традиційні елементи управління з WinForms.

Багаті можливості по створенню різних додатків: це і мультимедіа, і двомірна і тривимірна графіка, і багатий набір вбудованих елементів управління, а також можливість самим створювати нові елементи, створення анімацій, прив'язка даних, стилі, шаблони, теми і багато іншого.

Апаратне прискорення графіки – незалежно від того, чи працюєте ви з 2D або 3D, графікою або текстом, все компоненти програми транслюються в об'єкти, зрозумілі Direct3D, і потім візуалізуються за допомогою процесора на відеокарті, що підвищує продуктивність, робить графіком більш плавною.

Створення додатків під безліч операційних систем сімейства Windows – від Windows XP до Windows 10.

## **4.2 Мова 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 застосований у різних предметних областях, його основне призначення - конструювання інтерфейсів користувача для платформи WPF. Другими словами, документи XAML визначають розміщення панелей, кнопок та інших елементів керування, які складають вікна в WPF-додатках. Ймовірно, вам не потрібно буде писати код XAML вручну, оскільки використовується інструмент, який генерує необхідний код XAML.

За допомогою XAML в першу чергу описується інтерфейс. Вся логіка управління виконується у процедурному коді (наприклад, VB, JavaScript, C#, і т.д.). XAML може бути використаний як для веб-додатків, так і для звичайних десктопних додатків.

XAML включає основні чотири категорії елементів:

- панелі;
- елементи управління;

– елементи, пов'язані з документом і графічні фігури.

Додатки, оголошені в XAML, можуть включати безліч сторінок.

PageViewer – дозволяє розбивати зміст на сторінки і забезпечує навігацію по ним.

ContextMenu – допомагає в створенні навігаційних меню програми.

XAML широко використовується в Windows Presentation Foundation (WPF), Windows Workflow Foundation (WWF), .NETFramework3.0 і Silverlight. В WPF XAML використовується як мова розмітки для користувача інтерфейсу, для визначення елементів призначеного для користувача інтерфейсу, прив'язки даних, підтримки подій і ін. властивостей [9].

XAML файли можна створювати і редагувати за допомогою інструментів візуального конструювання, таких як: Microsoft Expression Blend, Microsoft Visual Studio, WPF visual designer. Також, їх можна створювати за допомогою стандартного текстового редактора, редактора коду такого як: XAMLPad, або графічного редактора, такого як Vestroпу.

Все створене або реалізоване в XAML може бути виражено за допомогою більш традиційних .NET мов, таких як: C# або Visual Basic.NET. Однак, ключовим аспектом технології є зменшення складності використовуваних для обробки XAML інструментів, так як XAML заснований на XML. В результаті цього з'являється безліч продуктів, що створюють засновані на XAML додатки. Оскільки XAML базується на XML, у розробників і дизайнерів існує можливість одночасно працювати над вмістом без необхідності компіляції [6].

Об'єктний елемент зазвичай оголошує екземпляр типу. Цей тип визначений в збірках, що надають резервні типи для технології, в якій XAML використовується як мова.

Синтаксис об'єктних елементів починається з дужки (<). Далі йде ім'я типу, для якого буде створюватися екземпляр. Після цього можна оголосити атрибути. Для завершення тега об'єктного елемента закриваємо дужку (>). Також можна встановити і самозакривну форму.

Властивості об'єкта в більшості випадків можна виразити у вигляді атрибутів об'єктного елемента. Синтаксис атрибутів називає властивість, що задається в синтаксисі атрибутів, після чого слідує оператор призначення (=). Значення атрибута завжди задається як рядок, взятий в лапки.

Синтаксис атрибутів це найбільш простий синтаксис завдання властивостей, інтуїтивно зрозумілий розробникам, які раніше використовували мови розмітки.

Для деяких властивостей об'єктного елемента неможливий синтаксис атрибутів, так як об'єкт або відомості, необхідні для задання значення властивості, не можуть бути адекватно представлені в рядку, про яку йдеться в лапках, з дотриманням обмежень синтаксису атрибутів. У цих випадках може використовуватися інший синтаксис – синтаксис елементів властивостей.

Для відкриваючого тега елемента властивості використовується наступний синтаксис: `<ім'я_типа.ім'я_властивості>`. У більшості випадків вміст тега є об'єктним елементом типу, до якого може належати значення цієї властивості. Вказавши вміст, необхідно завершити елемент властивості закриваючим тегом. Для закривання тега використовується синтаксис `</ім'я_типа.ім'я_властивості>`.

Якщо використовувати синтаксис атрибутів, це, як правило, зручніше і призводить до створення більш компактної розмітки. Однак, часто вибір між синтаксисом атрибутів і елементів обумовлений стилем програміста, а не обмеженнями технічної безпеки.

Мова XAML має деякі оптимізації, що полегшують сприйняття розмітки. Одна з таких оптимізацій полягає у тому, що якщо певна властивість приймає колекційний тип, елементи, вказані як дочірні елементи в розмітці всередині цієї властивості, стають частиною цієї колекції. Значенням цієї властивості є колекція дочірніх об'єктних елементів, визначених у розмітці.

XAML надає можливість відображати поточну функцію мови шляхом

призначення лише однієї властивості об'єктного елемента як властивості вмісту XAML. Дочірні елементи цього об'єктного елемента використовуються для задання значення цієї властивості вмісту. Іншими словами, для властивості вмісту (і тільки для неї) можна пропустити елемент властивості, вказавши цю властивість в XAML-розмітці, що дозволяє створити більш наочну метафору батьківсько-дочірніх елементів у розмітці.

### 4.3 Опис роботи програми

Після запуску програми на екрані з'являється головне вікно (рис. 6).

В головному вікні, у верхній частині, розташоване меню, в якому користувач може виконувати наступні дії:

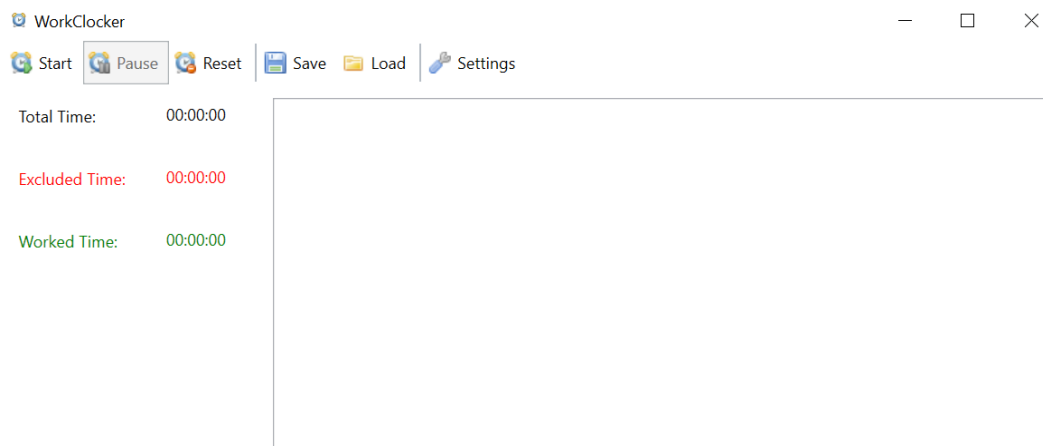


Рисунок 6 – Головне вікно програми

- запуск трекінгу;
- призупинення трекінгу;
- обнулення трекінгу;
- зберегти журнал дій у файл.
- завантажити журнал дій з файлу.
- перехід до налаштування (рис. 7).

У лівій частині розташована статистика, яка включає наступні показники:

- загальний час роботи трекера.
- кількість вручну виключеного часу з загального робочого часу.
- робочий час.

Білий прямокутник позначає вікно журналу дій. На рис. 7 журнал дій відсутній, оскільки трекер вимкнений.

У полі "AFK Delay" у вікні налаштувань можна встановити, через який проміжок часу програма почне враховувати, що користувач відлучився від комп'ютера. Відмітка "Autosort List" служить для автоматичного сортування програм у журналі за кількістю часу, витраченого на них.

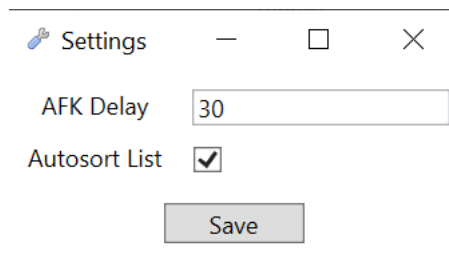


Рисунок 7 – Вікно налаштувань

Після запуску трекінгу можна переглядати витрачений час у вікні журналу (рис. 8).

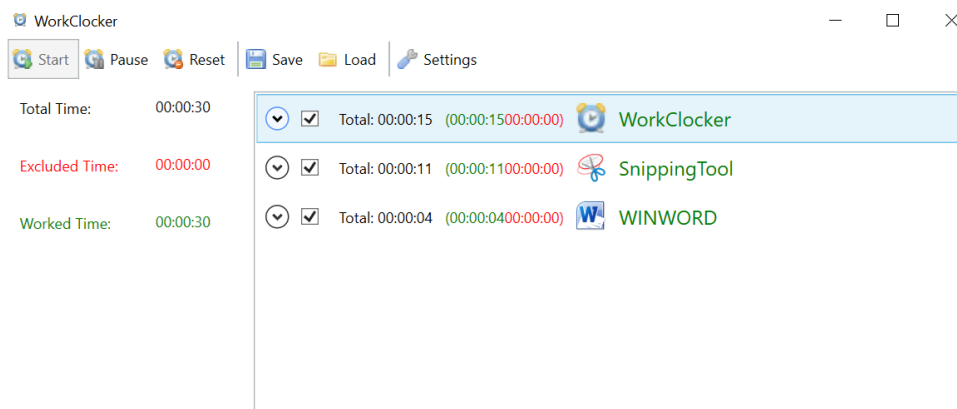


Рисунок 8 – Трекінг часу

Якщо через зазначений у налаштуваннях час користувач не виконував ніяких дій з клавіатурою або мишею, то у лозі з'являється процес «Away



From Keyboard» (рис. 9).

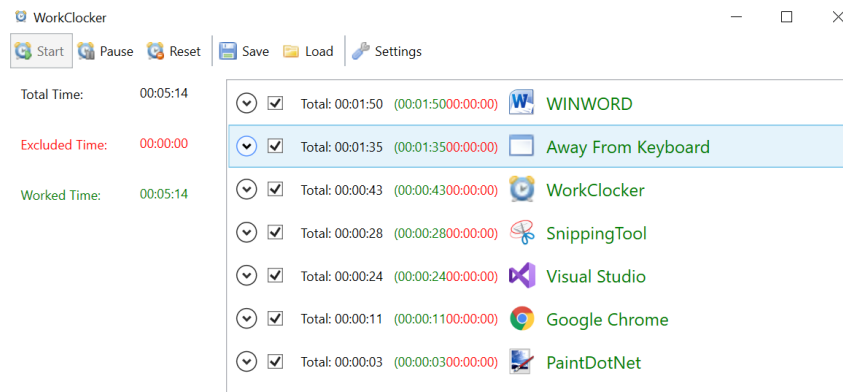


Рисунок 9 – Процес «Away From Keyboard»

Для детального контролю над вікном, кожний запис у лозі можна розгорнути, що представлено на рис. 10.

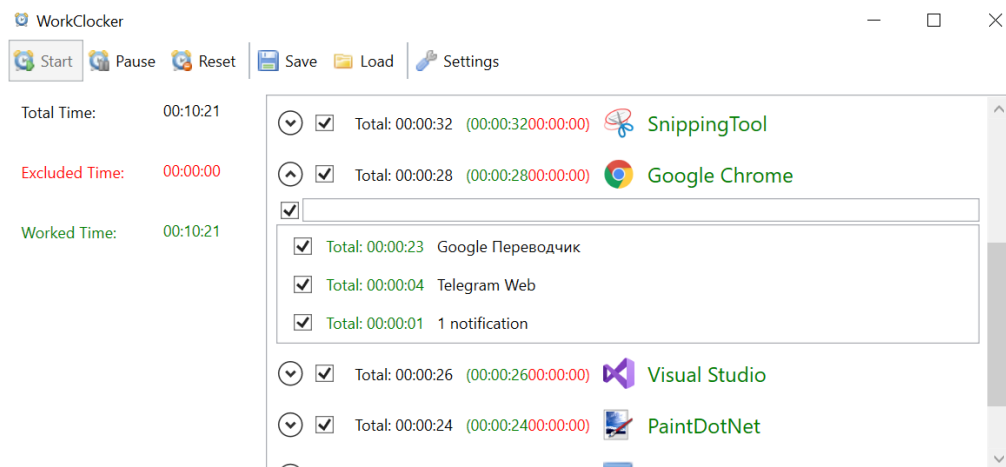


Рисунок 10 – Розгорнутий запис

Метод `SettingsButton_Click` викликається при натисканні кнопки "SettingsButton" або при виникненні відповідної події. У середині цього методу створюється нове вікно `SettingsWindow` і воно відображається за допомогою методу `Show()`.

```
private void SettingsButton_Click(object sender,
    RoutedEventArgs e)
{
```

```

        var window = new SettingsWindow();
        window.Show();
    }

```

Метод `SaveButton_Click` викликається при натисканні кнопки "SaveButton" або при виникненні відповідної події. Усередині цього методу викликається метод `_viewModel.SaveToDisk()`. Ймовірно, `_viewModel` – це об'єкт або екземпляр класу, який має метод `SaveToDisk()`, який здійснює збереження даних на диск.

```

private void SaveButton_Click(object sender, RoutedEventArgs e)
{
    _viewModel.SaveToDisk();
}

```

Метод `Window_Closing` викликається при закритті вікна додатку або при виникненні відповідної події. Усередині цього методу також викликається метод `_viewModel.SaveToDisk()`, який зберігає дані на диск.

```

private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
{
    _viewModel.SaveToDisk();
}

```

Загалом, цей код пов'язує певні дії з певними подіями. Він відкриває вікно налаштувань при натисканні кнопки "SettingsButton", зберігає дані на диск при натисканні кнопки "SaveButton" і при закритті вікна додатку.

За замовченням всі процеси у лозі вважаються робочими, необхідно вручну зняти мітки з неробочих процесів і це буде відображено у статистиці часу (рис. 11).

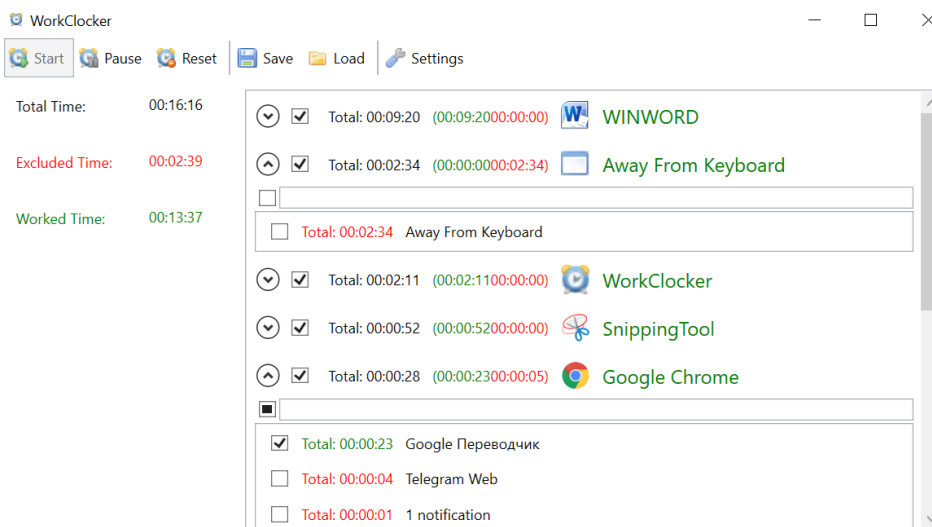


Рисунок 11 – Керування неробочим часом

При згортанні програмне вікно згортається у системний трей.

При завершенні роботи статистика автоматично зберігається у файл и доступна при наступному завантаженні.

## ВИСНОВКИ

Головна особливість сучасного етапу розвитку суспільства полягає в його інформатизації, мета якої у впровадженні комп'ютерів і засобів зв'язку в усі сфери діяльності людини.

Сьогодні безліч людей відкривають для себе необхідність і важливість використання програми для обліку робочого часу співробітників підприємства у виробництві і організації діяльності.

Функціонування сучасних підприємств в ринкових умовах передбачає розробку і впровадження економічного механізму, що забезпечує зростання ефективності на всіх рівнях управління та напрямки діяльності. Одним з основних елементів такого механізму є розробки програми для обліку робочого часу співробітників підприємства, а також бізнес процесів, які не тільки відображають функціональну та інформаційну діяльність, а й впливають на неї.

Моніторинг часу роботи за комп'ютером буденна річ сьогодні, адже більшість завдань офісних співробітників і фрілансерів виконується саме за ним. Якісний сервіс надасть звітність про повний робочий день співробітника з структурованої і простий для сприйняття інформацією.

В ході виконання бакалаврської кваліфікаційної роботи було отримано повнофункціональне програмне забезпечення для контролю над робочим часом на комп'ютері. Графічний інтерфейс виконано з використанням технології WPF. Інтерфейс створеної програми зручний, простий, відображає її можливості.

Розроблене програмне забезпечення задовольняє всім вимогам, поставленим на етапі постановки завдання.

Важливо знати, що актуальність додатку може змінюватися з часом, оскільки розробники постійно вдосконалюють функціональність та додають нові можливості.

Правильний додаток для відстеження часу позбавить користувача від здогадів щодо планування та обліку робочого дня. Легко запускати та зупиняти таймери, додавати або редагувати записи часу вручну, а також експортувати чи переглядайте дані для більш точного виставлення рахунків і планування. І можна заощадити ще більше свого дорогоцінного часу, автоматизувавши потік даних між програмами та вибраним пристроєм відстеження часу. Таким чином можна проводити свій час там, де це найважливіше.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Сэм Канер, Джек Фолк, Енг Кек Нгуен. Тестування програмного забезпечення. Print2print, 2001. 544 с.
- 2 Шпионские Программы и Приложения – ТОП-7 в 2023. URL: <https://www.business2community.com/ru/spy/shpionskie-programmy> (дата звернення 12.05.2023)
- 3 Автоматичний звіт часу ⇒ Контроль роботи за компютером URL: <https://timetracker.yaware.com.ua/blog/avtomaticheskij-uchet-vremeni-kontrol-raboty-za-kompyuterom/>. (дата звернення 10.05.2023)
- 4 Копитко М.Ф., Іванків К.С., Основи програмування мовою Java. Тексти лекцій Львів. Видавничий центр ЛНУ імені Івана Франка. 2002. 83 с.
- 5 Грицюк Ю.І., Рак Т.Є. Програмування мовою C++: навчальний посібник. Львів: Вид-во Львівського ДУ БЖД, 2011. 292 с. ISBN 978-966-3466-85-9.
- 6 Коноваленко І.В., Марущак П.О. Платформа .NET та мова програмування C# 8.0: навчальний посібник Тернопіль: ФОП Паляниця В. А., 2020. 320 с
- 7 Krypton.com.ua. URL: <https://krypton.com.ua/rozdil-1-vvedennya-c/> (дата звернення 12.05.2023)
- 8 Довідник з C#/ URL: <http://www.csharpcoderr.com> (дата звернення 15.05.2023)
- 9 Чарли Х., Бину Д. C# Performance / Х. Чарли,; McGraw-Hill Companies, 2010, 2008. 720с.
- 10 21 Best productivity and employee time tracking apps of 2023/ URL: <https://deskttime.com/best-employee-time-tracking-software/> (дата звернення 15.05.2023)