

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

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

Кваліфікаційна робота бакалавра

на тему: Розробка системного застосунку для менеджменту
робочого часу користувача на базі WPF

Виконала студентка групи К-19і
спеціальності 122 Комп'ютерні науки
Савченко Вероніка Ігорівна

Керівник д.т.н. професор
Казакова Надія Феліксівна

Рецензент регіональний
координатор Програми EGAP
Копиченко Іван Юрійович

ЗМІСТ

Перелік скорочень.....	6
Вступ.....	7
1 Програми для відстеження комп'ютерів персоналу	9
1.1 Основні цілі контролю співробітників	10
1.2 Характеристика системи обліку робочого часу.....	10
1.3 Контроль доступу системи.....	11
1.4 Можливості системи СОУД.....	11
1.5 Переваги СУРВА і СОУД.....	13
2 Основні системи для моніторингу ефективності співробітників.....	14
2.1 Принцип роботи систем моніторингу ефективності роботи співробітників і обліку робочого часу	15
2.2 Світовий ринок систем моніторингу ефективності співробітників і обліку робочого часу	17
2.3 Системи контролю і обліку дій співробітників	17
2.4 Недоліки програмного забезпечення моніторингу працівників	19
3 Опис програмних середовищ.....	21
3.1 Опис програмного забезпечення Kickidler.....	21
3.2 Опис програмного середовища ActivTrak	22
3.3 Опис системи Hubstaff.....	24
3.4 Опис системи DeskTime Pro.....	25
3.5 Опис програми Toggl.....	26
3.6 Опис програми Screenshot monitor	28
3.7 Опис програмного середовища Tmetric	29
4 Вибір середовища розробки.....	32
4.1 Опис мови С#.....	322
4.2 Основні компоненти для розробки	34
4.3 Опис технології WPF	37

4.4	Опис мови розмітки XAML	40
5	Опис програмного середовища.....	44
	Перелік посилань.....	55

ПЕРЕЛІК СКОРОЧЕНЬ

СУРВА – система обліку робочого часу;

СОУД – система обліку та контролю доступу;

ПЗ – програмне забезпечення;

XAML – Extensible Application Markup Language;

WWF – Windows Workflow Foundation;

WPF – Windows Presentation Foundation.

ВСТУП

Однією з головних проблем для більшості роботодавців є нецільове використання співробітниками робочого часу.

Візуально контролювати, чим саме зайнятий співробітник протягом робочого дня, дуже важко виконати. Ґрунтуватися на звіти про зайнятість, що заповнюють самі працівники – безглуздо (можуть бути недостовірні: приховувати періоди прокрастинації і завищувати продуктивність). Тому єдиний спосіб достовірно відстежувати активність персоналу і робити на основі її відповідні висновки – це спеціалізована програма для контролю.

Це програми які повинні безперервно відстежувати будь-які дії користувача на робочому комп'ютері, це зазвичай агент, який встановлюється на ПК співробітника і починає вести прихований моніторинг, записуючи натискання клавіш, введені пошукові запити, відвідані сайти, періоди роботи з різним ПО (в залежності від своїх можливостей). Найчастіше агент не відображається в списку запущених процесів Windows (або інший операційної системи), так що користувач не тільки не може його видалити, відключити або припинити, але іноді і не підозрює про сам факт його роботи.

На ринку з'явилося безліч рішень для спостереження за комп'ютером, від безкоштовного шпигунського ПЗ до розвинених систем моніторингу та обліку робочого часу з функціями запобігання витоку даних. Але не всі з них володіють необхідним і достатнім функціоналом, відповідним конкретної організації.

Згідно з дослідженнями: 64% працівників компаній витрачають від 15 хвилин до однієї години в день на особисті цілі; 46% щодня читають особисту пошту або новини, 33% спілкуються з особистих питань в месенджерах, 26% витрачають робочий час на соціальні мережі.

Ще одна з найсерйозніших загроз для бізнесу сьогодні – це витоку конфіденційної інформації. Як правило, джерелами таких загроз є недобросовісні співробітники компаній – інсайдери. Неконтрольоване

використання інтернету (соціальні мережі, месенджери, особиста пошта) може привести до витоку конфіденційної інформації, що в свою чергу завдасть бізнесу істотну шкоду.

Крім того, проблемою будь-якого керівника є шахрайство співробітників. Різні відкати і сірі схеми здатні принести не тільки разовий економічних збитків, а й серйозно підмочити репутацію організації на довгий час, що призводить до ще більших сумарним фінансових втрат.

Облік робочого часу та контроль співробітників в подібній ситуації – це необхідний захід. Багато роботодавців дуже зацікавлені в контролі поведінки працівника на робочому місці. Тому сьогодні все більше компаній придивляються до систем моніторингу ефективності співробітників, які допомагають вирішувати комплекс завдань щодо запобігання витоку інформації, розслідування інцидентів, якщо такі відбулися, а також контролювати бізнес-процеси і то, як робочий час використовується співробітниками.

Мета роботи: розробка програмного забезпечення контролю робочого часу працівника за персональним комп'ютером.

Виходячи з поставленої мети, в роботі розглянуті сучасні технологічні платформи для реалізації подібних програм, наведено порівняльний аналіз можливих варіантів реалізації, проаналізовано його результати. Також розглянуто повний цикл розробки програми.

Диплом містить: вступ, 5 розділів, висновки, рисунків, 1 таблицю та перелік посилань.

1 ПРОГРАМИ ДЛЯ ВІДСТЕЖЕННЯ КОМП'ЮТЕРІВ ПЕРСОНАЛУ

Співробітник, який працює за ПК, може замість користі принести організації значні збитки, якщо замість вирішення службових завдань займається особистими справами (або просто відсиджує час, чекаючи закінчення робочого дня). Тому будь-яка велика компанія, зацікавлена в оптимізації своїх бізнес-процесів, та приходять до необхідності впровадження програм стеження за комп'ютерами співробітників, тим більше, що подібних систем на ринку чимало.

Навіть в малому штаті офісних співробітників керівник не завжди може бути впевнений, що працівник пунктуальний в питаннях приходу на роботу і її завершення, не зловживає перервами і не займається інтернет-серфінгом замість вирішення поставлених завдань.

У великих корпораціях таке питання виникає особливо гостро. Крім того, що візуальний контроль стає просто неможливим, виникає небезпека, що співробітник стане використовувати робочі інструменти і месенджери в особистих цілях, що спричинить витік конфіденційної інформації.

Безцільне просиджування робочих годин, захопленість соціальними мережами з боку співробітника можуть привести до зриву термінів проекту, невірно підготовленою документації і падіння репутації фірми. Виходить, що керівник оплачує робочий час, який не тільки не приносить користі його організації, але може завдати їй значної шкоди.

При цьому виявити таких працівників не завжди просто, а також складно застосувати до них аргументовані заходи впливу. Цю дилему може вирішити впровадження системи моніторингу, яка мотивує безвідповідальних співробітників зосередитися на роботі, а відповідальним дає впевненість, що їх праця буде помічений і оцінений по достоїнству.

За статистикою офісні співробітники мінімум третину робочого часу витрачають на спілкування, розваги та відпочинок, наприклад, комп'ютерні

ігри або соціальні мережі. Все це знижує ефективність і результативність праці. Облік робочого часу співробітників лежить в основі ефективного управління персоналом. Це необхідний захід для будь-якого підприємства або організації. Домогтися трудової дисципліни допоможуть сучасні технології, які контролюють робочий час і зайнятість персоналу.

Керівництво компанії завжди має приділяти увагу контролю роботи персоналу. На сьогоднішній день кількість способів контролю співробітників в організації досить багато. Так, підприємство вибирає ту методику, яка найбільш зручна і ефективна.

1.1 Основні цілі контролю співробітників

Основні цілі контролю співробітників полягають у наступному:

- оцінка ефективності (ризик недостатньої навантаження персоналу, так працівники можуть виконувати завдання, але більшу частину часу проводять за сторонніми справами).
- оптимізація обсягу завдань (зрозуміла формулювання і правильний розподіл завдань між співробітниками).
- моніторинг робочого часу (спостереження і фіксування запізнь або затримок).
- Відстеження порушень (дотримання професійної етики спілкування з клієнтами і контрагентами).

1.2 Характеристика системи обліку робочого часу

Система обліку робочого часу (СУРВА) – це комплекс апаратних і програмних заходів, які використовуються для контролю діяльності співробітників фірми в автоматичному режимі .

Перші дані про те, скільки співробітники витрачають годин на роботу і скільки – на сторонні завдання, доступні вже за кілька днів після впровадження автоматизованої системи обліку часу.

Облік робочого часу співробітників включає запис та аналіз:

- контролю доступ (час приходу на роботу та відходу додому);
- знаходження на робочому місці (дозволяє виявити співробітників, які без причини залишають робоче місце);
- відпрацьованих годин (відслідковуються лікарняні, відпускні дні, відгули);

Всі ці дані зіставляються з реально проведеними на робочому місці годинами.

1.3 Контроль доступу системи

Системи обліку робочого часу використовують СОУД – система обліку та контролю доступу. Крім функції контролю СОУД необхідним є заходи безпеки, тому, що дозволяє контролювати доступ сторонніх до об'єкту і забезпечує збереження корпоративного майна .

Завдання СОУД – зафіксувати точний час появи на робочому місці і час уходу. Ці системи дозволяють легко контролювати запізнення або момент завершення робочого дня, навіть коли начальства немає на місці. СОУД використовують не тільки на великих промислових об'єктах, але і на приватних підприємствах і окремих секретних приміщеннях.

1.4 Можливості системи СОУД

Робота всіх компонентів системи спрямована на облік, аналіз, зберігання, реєстрацію та обробку даних про діяльність підприємства. Основна можливість СОУД – це контроль приходу і відходу персоналу. Дана

функція реалізується за допомогою автоматизованої системи, що встановлюється на вхід і не дозволяє пройти без електронного ключа доступу.

Механізм обмеження доступу повинен включати в себе систему відеоспостереження, турнікети, посилені замки з електронним ключем і інші елементи, які забороняють вільний вхід / вихід в будівлю або офіс.

У кожного співробітника повинен бути свій ідентифікатор, який «вшивається» в електронний ключ. Ключем доступу можуть виступати документи, посвідчення або біометрія.

Дані ключів зчитуються та записуються в загальний електронний журнал обліку, який зберігається на хмарному чи локальному сервері.

Програмний модуль СУРВА відстежує активність кожного користувача. Залежно від версії ПЗ, типу підписки і вартості, програма документує дані про роботу з встановленим набором програм. Зустрічаються програми обліку де є можливість контролю телефонних дзвінків, перегляд робочого столу користувача в режимі реального часу, контроль поштової скриньки.

Процес установки апаратної частини системи контролю має включати прокладку кабелів для установки апаратних компонентів, встановлення та налаштування центрального ПК для обробки даних обліку. Далі необхідна робота з програмною оболонкою СУРВА, що передбачає впровадження системи контролю для центрального комп'ютера, установку необхідного для роботи СУРВА програмного забезпечення. Також необхідно зробити налаштування параметрів контролю і розробка схеми відображення звітів по роботі кожного співробітника. Фірма має можливість встановлювати системи контролю доступу та обліку робочого часу як разом, так і окремо.

1.5 Переваги СУРВА і СОУД

Впровадження СУРВА і СОУД допомагає:

- підвищити рівень контролю за рахунок повного виключення людського фактора, коли всю роботу з обліку виконує комп'ютер, і співробітники зможуть підмінити або «прикривати» один одного.
- налагодити безперервну роботу з можливістю запуску системи в режимі 24/7.
- поєднати підтримку обмеження доступу, обліку робочого часу, відеоспостереження та інших функцій.

Система обліку робочого часу – це зручний спосіб проконтролювати регламент роботи і продуктивність співробітників фірми. СУРВА служить інструментом для оцінки персоналу, який допомагає приймати зважені кадрові рішення.

2 ОСНОВНІ СИСТЕМИ ДЛЯ МОНІТОРИНГУ ЕФЕКТИВНОСТІ СПІВРОБІТНИКІВ

Моніторинг продуктивності або програмне забезпечення відстеження часу проекту є важливою частиною програми управління ризиками. Розуміння фактичного розподілу часу співробітників навколо конкретних завдань дозволяє прозоро вимірювати ризику до проекту та дотримуватися організаційних вимог. Програмне забезпечення для моніторингу працівників дозволяє вам дізнатися про слабкі місця в рамках розробленої тактики та бізнес-процесів або виявити неефективні методи управління персоналом.

Системи моніторингу ефективності співробітників застосовують для вирішення наступних завдань:

- облік робочого часу співробітників;
- виявлення нелояльних співробітників і шахрайських схем всередині підприємства;
- оцінка і контроль ефективності роботи персоналу;
- пошук можливих витоків інформації і захист від інсайдерів;
- оперативний контроль співробітників і визначення груп ризику;
- розслідування інцидентів інформаційної безпеки;

Типова архітектура таких систем передбачає наявність сервера, бази даних, агента і консолі адміністратора. При цьому можна виділити:

- системи на товстому клієнті, сервер і агенти обов'язково повинні перебувати в рамках локальної мережі;
- рішення, що дозволяють працювати з агентами, які перебувають в інших мережах;
- рішення на основі хмари – база даних знаходиться у вендора, а агенти не прив'язані до локальної мережі, управління і перегляд здійснюється через веб-консоль.

Якщо ж відштовхуватися від концепції рішень, то системи можна диференціювати таким чином:

- перша група систем дозволяє збирати великий обсяг даних і надавати їх у вигляді аналітичних звітів, а також вести статистику продуктивного і непродуктивного робочого часу;
- друга група надає керівнику можливість здійснювати моніторинг співробітників через перегляд онлайн-трансляцій або відеозаписи їхніх дій, а також здійснювати контроль порушень (рис. 1);

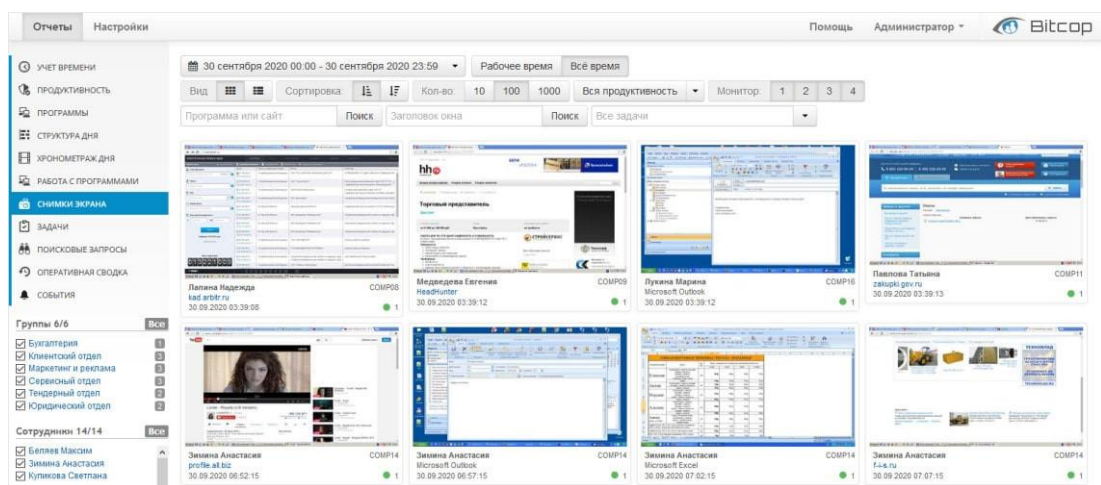


Рисунок 1 – Приклад моніторингу співробітників

- третя група систем має досить вузьким набором функцій, які зводяться тільки до обліку робочого часу персоналу.

2.1 Принцип роботи систем моніторингу ефективності роботи співробітників і обліку робочого часу

У загальному випадку схема роботи систем моніторингу ефективності роботи співробітників виглядає наступним чином: на комп'ютери співробітників встановлюється агент системи, який збирає інформацію про дії користувача на робочому місці. Агент передає інформацію на сервер для аналізу.

Адміністратор або керівник за допомогою консолі («товстий» клієнт або веб-консоль) знайомиться з результатами діяльності співробітників: це можуть бути звіти, перегляд онлайн-трансляцій з моніторів співробітників, перегляд скріншотів, відеозапис робочого столу або перехоплений в листуванні файл і т.д.

При цьому з функціональної точки зору системи можна поділити на три основні групи:

- моніторинг подій;
- моніторинг інформації;
- моніторинг системи.

Основним напрямком моніторингу подій є контроль факту включення / вимикання системи, натискання клавіш, використання (кліків) мишки, а також факти роботи з додатками і інтернетом (відвідування веб-сайтів). Факт включення, вимикання системи, натискання клавіш і мишки дає можливість контролювати активність користувача, яка відображає: прихід і відхід співробітників, наявність запізнень, переробок, загальний час роботи, а також час активності і простою в годиннику.

Наступний блок, який впливає з контролю активності, це факти роботи з додатком і відвідуванням веб-сайтів, а також розбивка отриманої інформації на продуктивну і непродуктивну діяльність. Активне час аналізується в процентному співвідношенні роботи з продуктивними, невизначеними і непродуктивними ресурсами. При цьому важливою складовою є можливість будувати звіти, топи, а також порівнювати відділи та користувачів.

Крім завдання, пов'язаної з контролем ефективності користувачів, системи моніторингу вирішують ряд проблем, пов'язаних з контролем каналів передачі даних і запобіганням витоків конфіденційної інформації – системи можуть успішно виконувати основну, найбільш затребувану частину функцій DLP і відстежувати витік конфіденційної інформації завдяки

наявності контролю всіляких каналів передачі даних і роботи з зовнішніми пристроями.

У частині моніторингу подій системи може контролюватися реєстр ПЗ, апаратне забезпечення, порти, використання програм і веб-сайтів

2.2 Світовий ринок систем моніторингу ефективності співробітників і обліку робочого часу

На світовому ринку представлено безліч різних систем моніторингу ефективності співробітників – як від відомих вендорів в області безпеки (Symantec і Trend Micro), так і надвеликих гравців, що займаються тільки цим класом продуктів. Згідно з дослідженням Hexa Reports Employee Monitoring Software Market Size, Share, Industry Trends, Analysis And Forecast, 2022, на світовому ринку представлені наступні основні гравці: 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.

Далі в роботі буде проаналізовано кілька програмних середовищ: Kickidler, ActivTrak, Hubstaff, DeskTime Pro, Toggl, Screenshot Monitor, Tmetric.

2.3 Системи контролю і обліку дій співробітників

Програма-шпигун (рис. 2) для прихованого спостереження за робочим комп'ютером співробітника – це оптимальне рішення для організацій, які націлені на підвищення власної конкурентоспроможності. У сучасних реаліях, коли частина співробітників може працювати віддалено, особливо гостро постає питання про контроль використання робочого часу та недопущення його безконтрольних втрат. Тільки комп'ютерний шпигун, який діє з відома співробітників, але ніяк не виявляє своєї присутності і не заважає

щоденній роботі, може дати роботодавцю чітке уявлення про витрачання оплачуваної їм робочого часу.

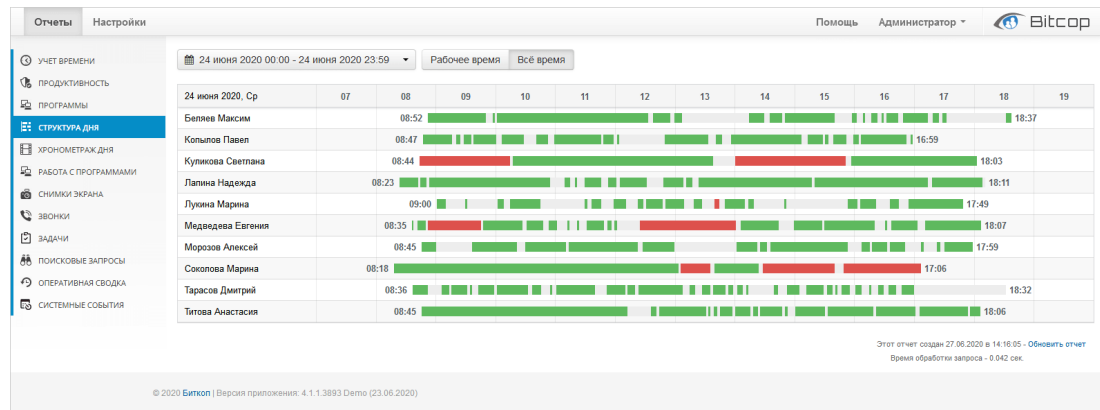


Рисунок 2 – Вікно програми шпигуна

Програма тотального контролю персонального комп'ютера, впроваджена в організації, вирішує відразу кілька проблем роботи і низьких показників:

- виявляє перспективних і непродуктивних співробітників;
- показує запізнення, переробки, часті відлучки і перерви в роботі;
- задає мотивацію на підвищення продуктивності;
- допомагає запобігти витоків інформації і обчислити неблагонадійних членів команди;
- показує програми і сайти, які поглинають робочий час співробітників і тому повинні бути обмежені або заборонені для використання на робочому місці.

Оплачуючи робочий час, роботодавець має право вимагати, щоб воно витрачалось на благо організації шляхом виконання прямих службових обов'язків, і використовувати для перевірки цього необхідні засоби контролю. Систематично аналізуючи дані, зібрані обліковою програмою, і застосовуючи відповідно до них гнучку управлінську політику, можна в найкоротші терміни істотно підвищити віддачу кожного співробітника і всього штату в цілому. На відміну від простих утиліт, які розповсюджуються безкоштовно, або урізані демо-версії великих систем, протягом пробного

періоду дають можливість швидко оцінити її переваги і зрозуміти, наскільки її використання підходить і чи має воно результат.

2.4 Недоліки програмного забезпечення моніторингу працівників

Незважаючи на те, що використання програмного забезпечення для моніторингу працівників має багато переваг, але і є кілька потенційних недоліків – найбільшим із них є зниження довіри та морального стану працівників. Співробітники можуть контролювати програмне забезпечення працівників як відсутність довіри роботодавців до них, що врешті-решт може породити невдоволення та знизити моральний дух компанії. Нікому не подобається відчувати, як за ними спостерігають, а програмне забезпечення для моніторингу працівників може розцінюватися як втрата конфіденційності.

Під час першого впровадження програм роботодавці повинні пройти тонку межу. Щоб залучити співробітників до бази програмного забезпечення для моніторингу та зменшити ризик плинності чи невдоволення працівників, причини використання програмного забезпечення та підкреслити корисність для компанії.

Іншим потенційним недоліком є кількість часу, яке може знадобитися для осмислення даних, зібраних із програмного забезпечення. Програмні забезпечення пропонують безліч інформації, яка може допомогти компаніям поліпшити безпеку та впорядкувати свої процеси, але для розшифровки найважливіших точок даних може знадобитися трохи додаткової роботи.

Впровадження програмного забезпечення для моніторингу працівників, може привести до потенційних ризиків конфіденційності та безпеки. Це може здатися неінтуїтивним, оскільки програмне забезпечення призначене для захисту вашої мережі, але цілком ймовірно, що можна натрапити на приватні дані співробітників, такі як медичні записи або дані банківського рахунку. Для того, щоб зменшити такий ризик, необхідно пояснити працівникам, до

яких вони повинні, а до яких не повинні мати доступ на корпоративних пристроях.

Моніторинг працівників дозволяє менеджерам відповідально відстежувати різні види діяльності, такі як:

- обсяг діяльності (клацання миші, прокрутка та натискання клавіш).
- екранні завдання (за допомогою скріншотів менеджери можуть переглядати, які завдання відкриті у співробітників, будь то веб-сторінки, програми чи інше програмне забезпечення).
- коли працюють співробітники (інформаційна панель в режимі реального часу дозволяє точно бачити, коли співробітники працюють в Інтернеті).
- усі відвідані веб-сайти та додатки (знімки екрана - не єдиний спосіб побачити, з якими веб-сторінками та програмами взаємодіє працівник, програмне забезпечення також надає вичерпний перелік усіх веб-сайтів та програм, які відвідують працівники).

3 ОПИС ПРОГРАМНИХ СЕРЕДОВИЩ

Абсолютно всі програмні забезпечення для моніторингу персоналу, як і будь-який інший B2B-продукт, створюється виключно в комерційних цілях. Тобто, в кінцевому підсумку все воно є платним.

Однак деякі розробники надають користувачам безкоштовні (Freemium) версії своїх програм з обмеженою функціональністю, якої цілком може вистачити для вирішення мінімальних завдань по автоматизації контролю співробітників.

3.1 Опис програмного забезпечення Kickidler

Kickidler – система для обліку робочого часу нового покоління (рис. 3). ПЗ призначено для автоматизації в організації функцій контролю співробітників.

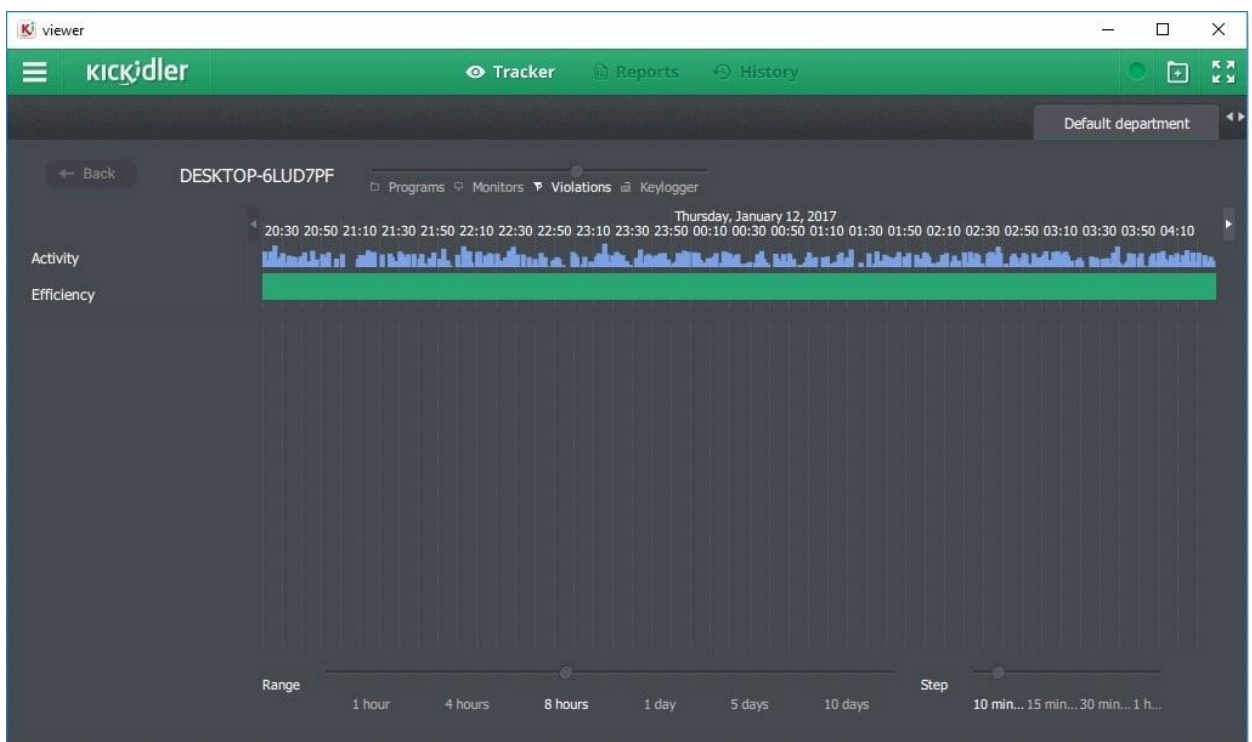


Рисунок 3 – Вікно системи Kickidler

Програма надає потужний набір інструментів для моніторингу робочих комп'ютерів і виявлення порушень робочого розпорядку.

Система обліку робочого часу Kickidler дає користувачам безкоштовні функції з можливістю моніторингу для великої кількості користувачів. Крім цього, дана програма надає до 2-х тижнів безкоштовного використання свого повного функціоналу для моніторингу необмеженої кількості ПК.

Доступні функції в безкоштовному режимі:

- онлайн-моніторинг комп'ютерів співробітників (перегляд екранів співробітників в реальному часі а також доступний багатокористувальницький режим);
- звіти робочого часу співробітників (облік робочого часу, аналіз ефективності, динаміка активності, рейтинг співробітників по продуктивності, рейтинг найбільш часто використовуваних додатків, деталі дня);
- кейлоггер (збереження історії натиснутих клавіш).

У безкоштовної версії Kickidler доступний моніторинг до 6 комп'ютерів співробітників.

У платній версії програми доступні наступні функції:

- відеозапис і відтворення історії дій користувача за комп'ютером;
- контроль порушень (виявлення і перегляд підозрілих дій користувача);
- автоматичні повідомлення про скоєних порушеннях, а також дистанційне управління ПК співробітників.

В Kickidler найбільш цікаві три функції: відеозапис з трансляцією на робочий стіл, автоматичне розпізнавання непродуктивних дій і віддалений перехоплення управління. Програма швидко розвивається, і обростає новими функціями.

3.2 Опис програмного середовища ActivTrak

ПЗ ActivTrak – це програма, яка призначена для моніторингу активності, аналітики продуктивності і поведінки співробітників за робочими комп'ютерами, а також для виявлення інсайдерських погроз. Програма (рис. 4) проста у використанні і установці рішення для хмарного моніторингу, яка має локальну версію.

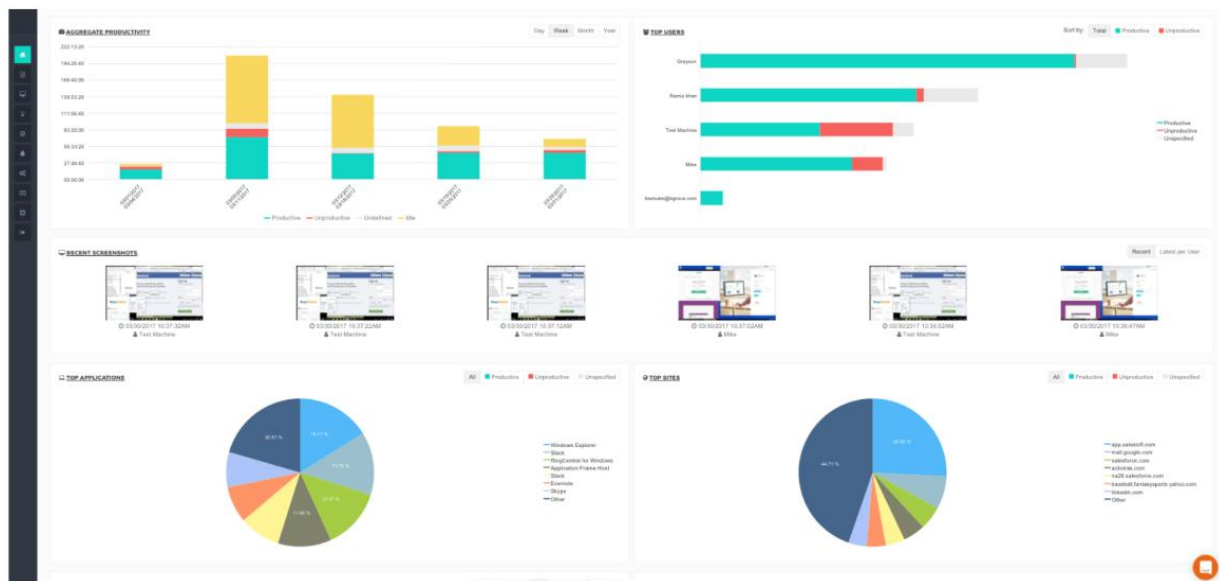


Рисунок 4 – Вікно програми ActivTrak

ActivTrak надає своїм користувачам безкоштовну версію програми для моніторингу до 3 робочих станцій з обмеженнями по функціональності і місцю на хмарному сховище до 3 Гб.

Доступні функції в безкоштовному режимі.

- онлайн-моніторинг комп'ютерів співробітників (доступ в реальному часі до перегляду екранів співробітників, не містить багатогористувальницького режиму);
- звіти про робочий час;
- блокування сайтів;

- кейлоггер;
- скріншоти екранів співробітників (не більше 1 знімка екрана для кожної активності).

Платна версія програми не має на увазі будь-яких додаткових функцій, але надає користувачеві необмежену місце на хмарі для зберігання логів програми, а також – можливість моніторингу практично необмеженого числа співробітників.

3.3 Опис системи Hubstaff

Hubstaff – це система відстеження часу з функціями захоплення екрану, моніторингу завдань, автоматичної оплати праці, розширеними звітами, відстеженням в режимі реального часу (рис. 5).

Organization: A Acme Corp

Invoices (2)

CLIENTS **TEAM**

OUTSTANDING: \$14,024.00 PAID: \$94,020.40

CLIENTS: Hubstaff STATUS: All Apply

Batch actions: Export CSV New Invoice

<input type="checkbox"/>	Invoice	Client name	Organization	Created on	Total	Paid	Status	Actions
<input type="checkbox"/>	#21	Hubstaff	E Epic Coders	8/4/2016	\$240.00	\$0.00	Sent	Actions
<input type="checkbox"/>	#22	Hubstaff	E Epic Coders	8/5/2016	\$160.00	\$0.00	Paid	Actions

Showing 1-2 of 2

Рисунок 5 – Головне вікно Hubstaff

Сервіс інтегрується з більш ніж 30 популярними інструментами. Hubstaff трохи відрізняється від попередніх програм і більше підходить для віддалених команд, невеликих компаній та фрілансерів.

Безкоштовна версія програми Hubstaff дозволяє вести моніторинг робочої станції 1 співробітника, а також має обмеження по функціональності.

Доступні функції в безкоштовному режимі:

- звіти про робочий час;
- відстеження рівня активності співробітника на основі використання миші та клавіатури;
- скріншоти екранів співробітників (обмеження за місцем в сховище).

Програма Hubstaff призначена для роботи з віддаленими співробітниками. У ній є унікальні в своєму роді інструменти для нарахування заробітної плати фрілансерам. Крім цього, у Hubstaff є функція GPS-відстеження працівників.

3.4 Опис системи DeskTime Pro

DeskTime – це онлайн-сервіс з мобільними і десктопними додатками для відстеження робочого часу, відстеження прогресу в проекті, аналізу різних показників ефективності, генерації докладних звітів. Як і Hubstaff, програма орієнтована на управління віртуальними командами (рис. 6).

Використовуючи DeskTime, можу досягти максимальної ефективності роботи колективу, відмовитися від ручного заповнення табелів обліку робочого часу і складати точні звіти для клієнтів або ради директорів компанії.

Безкоштовна версія програми DeskTime Pro надає моніторинг одного користувача з великими обмеженнями по функціональності.

Доступні функції в безкоштовному режимі:

- звіти по робочому часу (відстеження часу);
- звіти по використовуваних додатків;

– мобільний додаток.

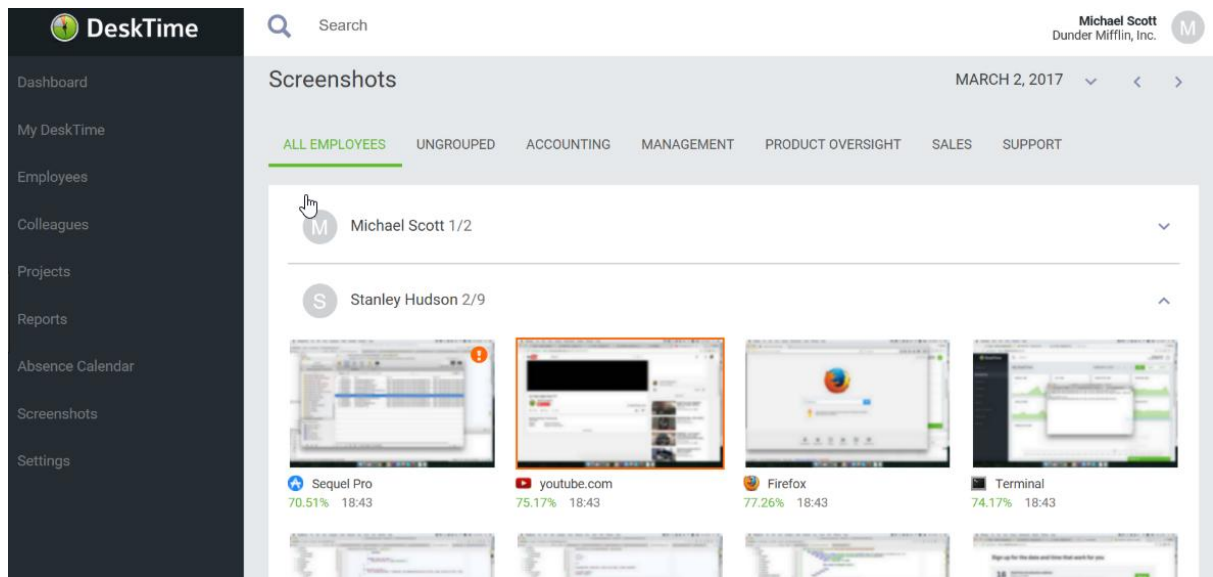


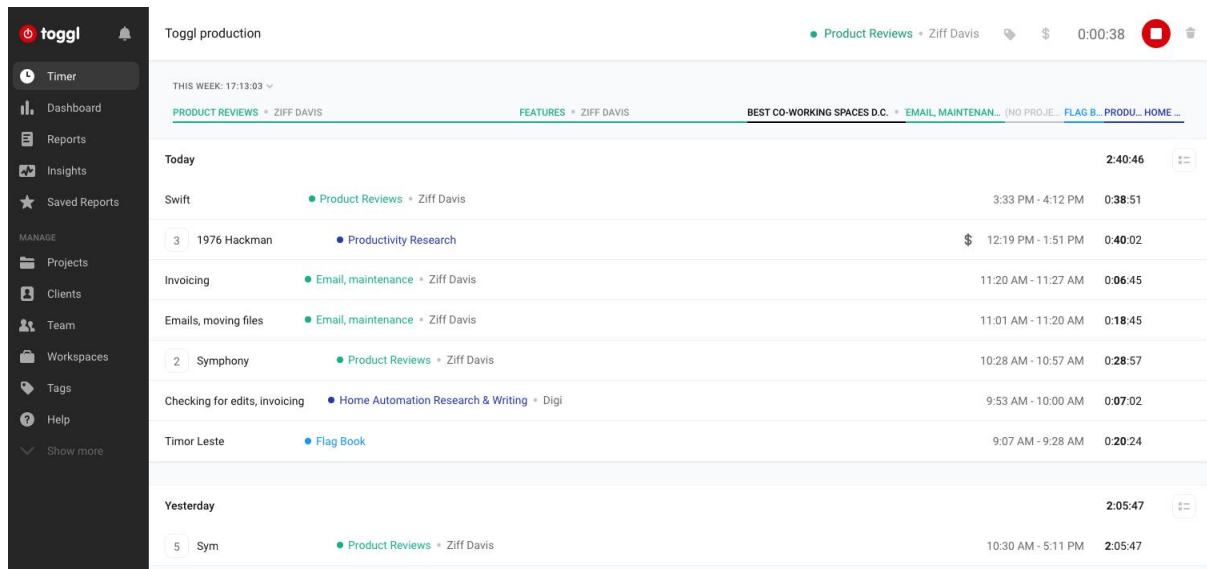
Рисунок 6 – Вікно програми DeskTime Pro

Програма моніторингу співробітників DeskTime Pro пропонує потужний інструментарій для контролю та управління віддаленими командами з можливістю тонкого налаштування під кожного користувача. Найцікавіші тут функції підрахунку заробітної плати, відстеження проектів і вбудованого Pomodoro-трекера.

3.5 Опис програми Toggl

Хмарна програма Toggl (рис. 7) відноситься до іншої категорії програм, ніж всі перераховані вище продукти. Даний софт, більш призначений для відстеження часу і самоконтролю, ніж для моніторингу співробітників з боку менеджерів. Працює програма як на десктопних, так і на мобільних пристроях. Toggl має можливість створювати звіти та графіки про використаний час, який, при бажанні, експортується в різні формати. На

відміну від інших продуктів, Toggl дає на безкоштовне тестування повного функціоналу цілих 30 днів.



Рисунко 7 – Вікно програми Toggl

У безкоштовної версії програми доступний моніторинг 5 робочих машин співробітників.

Доступні функції в безкоштовному режимі:

- звіти робочого часу;
- вбудований Pomodoro-трекер;
- складання графіків роботи.

Платна версія програми складається з двох пакетів: Стандарт і Преміум. У них відсутні обмеження по кількості користувачів та доступні розширені функції з управління проектами, складання звітів і управління командою.

3.6 Опис програми Screenshot monitor

Основним призначенням програми Screenshot Monitor (рис. 8), є періодичні знімки моніторів співробітників, які відправляються керівнику. Кількість скріншотів, вироблених програмою за годину, залежить від обраного тарифіку. У безкоштовній версії програми доступні 3 скріншота на годину. Крім скріншотів, в безкоштовній версії програми є функція обліку робочого часу. Програма досить демократична порівняно з перерахованими вище продуктами, наприклад, у співробітника є можливість видаляти скріншоти, на які потрапила особиста інформація.

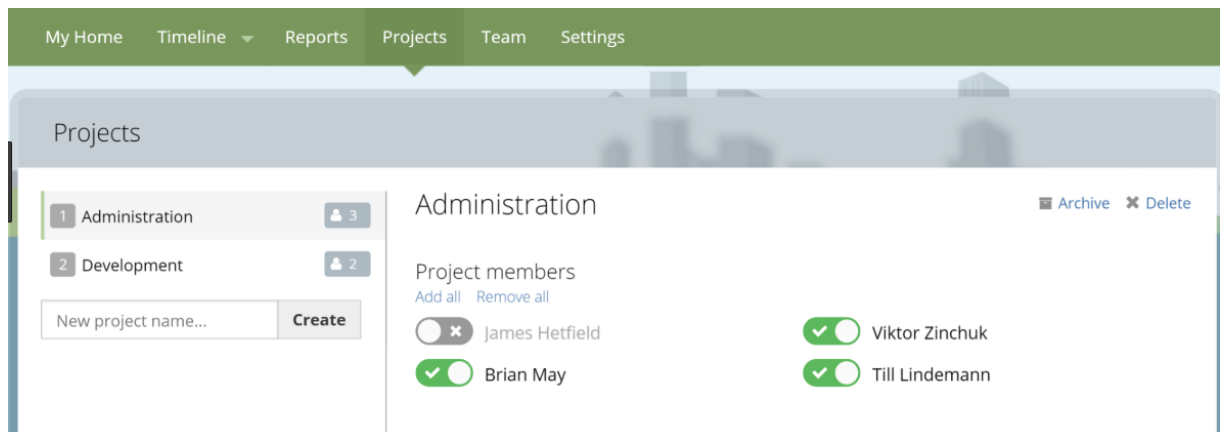


Рисунок 8 – Вікно програми Screenshot monitor

У безкоштовній версії програми доступний моніторинг до 3 комп'ютерів співробітників.

Доступні функції в безкоштовному режимі:

- звіти по робочому часу;
- скріншоти екранів комп'ютерів співробітників (до 3 скріншів на годину);
- час зберігання скріншотів – 2 тижні.

Платна версія програми дозволяє аналізувати рівні активності користувачів, додатки і використовуються url. Про-версія програми робить до 30 скріншотів на годину і зберігає їх протягом року.

3.7 Опис програмного середовища Tmetric

Tmetric – це сервіс відстеження часу (рис. 9). Ідеально підходить для фрілансерів і невеликих команд, які шукають просте рішення для управління часом і завданнями при роботі з віддаленими клієнтами.

TMetric має можливість:

- встановлювати бюджети для проектів;
- встановити ставки на будь-яке завдання, клієнта, проект або члена команди;

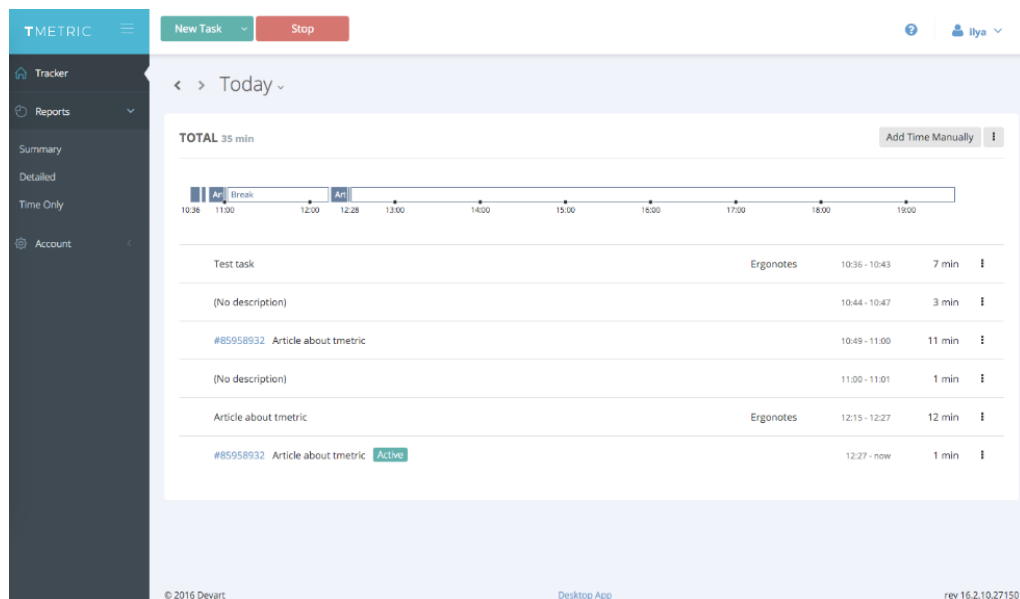


Рисунок 9 – Вікно програми Tmetric

- створювати теги і встановлювати їх для будь-якого завдання;
- бачити свій робочий день на часовій шкалі;
- отримати докладні звіти про час і гроші.

У безкоштовної версії програми доступний моніторинг 5 робочих машин співробітників.

Доступні функції в безкоштовному режимі:

- звіти по робочому часу;
- необмежені проекти і клієнти;
- інтеграція практично з усіма відомими веб-додатками для бізнесу.

Всі функції, пов'язані бюджетуванням проектів і виставленням рахунків, в програмі є платними. Звіти по додатках також доступні тільки в платній версії програми. Крім того, в безкоштовній версії недоступні синхронізація з JIRA і Redmine.

В заключенні можна представити основні параметри найкращих популярних та безкоштовних програмних забезпечень, показані у табл. 1.

За допомогою програмного забезпечення для моніторингу працівників можна відстежувати активність та продуктивність працівників. До найосновніших функцій можна віднести:

- веб-перегляд, що дозволяє контролювати веб-сайти, які відвідують співробітники;
- моніторинг додатків, який показує вам конкретні програми, якими користуються працівники;
- відстеження активності, яке контролює активність клавіатури та миші, щоб ви могли визначити, чи справді співробітники працюють у робочий час.

Таблиця 1 – Аналіз програмних середовищ

Програма	Пробний період (free)	Кількість користувачів	Хмарна або 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 i Linux
Screenshot monitor	30 днів	3	І те, і інше			Windows, Mac
Tmetric	30 днів	5	І те, і інше			macOS, Windows, Linux, iOS и Android

В даному розділі було проаналізовано програмні середовища такі як Kickidler, ActivTrak, Hubstaff, DeskTime Pro, Toggl, Screenshot Monitor, Tmetric. Для створення програмне забезпечення було вибрано за функціоналом програмне середовища Kickidler.

4 ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

4.1 Опис мови C#

На даний момент мова програмування C# є однією з найпотужніших, швидко зростаючих і популярних мов в ІТ-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.

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#.

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.

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;

- 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;
- 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.;

Також примітна особливість C# та .NET Framework, автоматична колекція сміття. 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.

4.2 Основні компоненти для розробки

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

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 з багатьох причин:

- по-перше, Visual Studio є інтегрованим середовищем швидкої розробки програмного забезпечення для роботи під Microsoft Windows, на даний момент найпоширенішою операційною системою.
- по-друге, середовище програмування Visual Studio 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 VS – це один з найвідоміших засобів створення Windows-додатків і зв'язку з продукцією Microsoft встановлюються досить легко.
- по-третє, Visual Studio було обрано через доступності цієї програми будь-якому користувачеві.
- по-четверте, Visual Studio відзначається своєю стабільністю, швидкістю і низькими вимогами до апаратного забезпечення ЕОМ.

4.3 Опис технології WPF

Windows Presentation Foundation (WPF, кодова назва – Avalon) – графічна (презентаційна) підсистема в складі .NET Framework 3.0, що має пряме відношення до XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista, а також доступна для установки в Windows XP Service Pack 2 і Windows Server 2003.

Це перше реальне оновлення технологічного середовища призначеного для користувача інтерфейсу з часу випуску Windows 95. Воно включає нове ядро, яке повинне замінити GDI і GDI+, використовувані на нинішній Windows-платформі. WPF є високорівневим об'єктно-орієнтованим функціональним шаром (англ. framework), що дозволяє створювати двовимірні та тривимірні інтерфейси.

Багато років. NET розробники створювали настільні прикладні програми, використовуючи технологію Windows Forms. Windows Forms – зріла і повнофункціональна технологія, здатна виконати основні завдання сучасних бізнес програм. Але робота технології жорстко пов'язана на архітектурі операційної системи і деякі завдання, які повинні виконуватися швидко і без зайвого програмування, вимагають великих витрат часу і зусиль.

WPF змінила світ програмування настільних прикладних програм. Поклавши в основу технологію DirectX, Microsoft надає можливість розробникам швидко створювати складні елементи управління і повністю керувати процесом візуалізації. Тепер створити красиву кнопку з анімаційними ефектами можна не написавши жодного рядка коду на C#.

Працювати з мультимедійним вмістом стало набагато простіше, розширилася модель зв'язування даних, друку і роботи з документами. WPF комбінує кращі аспекти традиційної розробки для Windows, з безліччю нововведень, дозволяючи будувати насичені графікою інтерфейси для користувача. Окрім цього, така технологія має всього дві прості вимоги:

знання мови C# та вміння користуватися бібліотеками .NET Framework. Схематично архітектура WPF представлена на рис. 10.

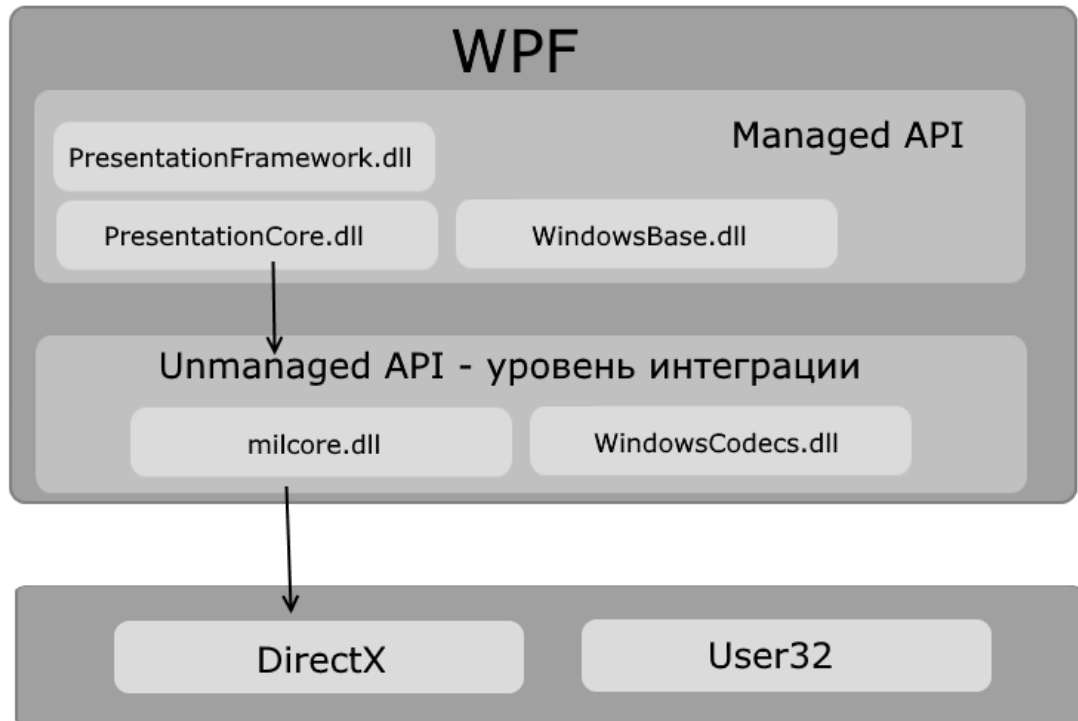


Рисунок 10 – Архітектура WPF

Як видно на рис. 12, WPF розбивається на два рівня: managed API і unmanaged API (рівень інтеграції з DirectX). Managed API (керований API-інтерфейс) містить код, що виконується під управлінням загальномого середовища виконання .NET – Common Language Runtime. Цей API описує основний функціонал платформи WPF і складається з наступних компонентів:

- PresentationFramework.dll: містить всі основні реалізації компонентів і елементів управління, які можна використовуватися при побудові графічного інтерфейсу;
- PresentationCore.dll: містить всі базові типи для більшості класів з PresentationFramework.dll;

- WindowsBase.dll: містить ряд допоміжних класів, які застосовуються в WPF, але можуть також використовуватися і поза даної платформи;
- Unmanaged API використовується для інтеграції вищого рівня з DirectX;
- milcore.dll: власне забезпечує інтеграцію компонентів WPF з DirectX. Даний компонент написаний на некерованому кодї (C / C++) для взаємодії з DirectX;
- WindowsCodecs.dll: бібліотека, яка надає низькорівневу підтримку для зображень 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.

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, всі 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.

4.4 Опис мови розмітки XAML

XAML (скорочення від Extensible Application Markup Language – розширювана мова розмітки застосунків) є мовою розмітки, яку використовують для створення екземплярів об'єктів .NET. Хоча мова XAML – це технологія, що може бути застосовна до багатьох різних предметних областей, її головне призначення – конструювання інтерфейсів користувачів WPF. Інакше кажучи, документи XAML визначають розташування панелей, кнопок та інших елементів керування, що становлять вікна в застосунку

WPF. Малоімовірно, що вам доведеться писати код XAML вручну. Замість цього ви використовуєте інструмент, що генерує необхідний код XAML.

За допомогою XAML насамперед описується інтерфейс. Вся логіка керується процедурним кодом (VB, JavaScript, C#, JavaScript і т. XAML може бути використаний як і для браузерних додатків, так і для звичайних додатків 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 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.

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.

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 допустимо синтаксис атрибутів, його використання, як правило, більш зручно і призводить до створення більш компактною розмітки, але часто вибір пов'язаний тільки зі стилем, а не з правилами техніки безпеки.

Мова XAML включає деякі оптимізації, які роблять розмітку зручніше для людського сприйняття. Одна з оптимізацій полягає в тому, що якщо певна властивість приймає тип колекції, елементи, оголошені в розмітці як дочірні в межах значення цієї властивості, стають частиною колекції. В цьому випадку колекція дочірніх об'єктних елементів є значенням, яке задається для властивості колекції.

XAML відображає поточну функцію мови, за рахунок чого клас може призначити тільки одне зі своїх властивостей як властивість вмісту XAML. Дочірні елементи даного об'єктного елемента використовуються для завдання значення цієї властивості вмісту. Іншими словами, для властивості вмісту (і тільки для нього) можна опустити елемент властивості, вказавши цю властивість в XAML-розмітці, і тим самим створити більш наочну метафору батьківського-дочірнього елементів в розмітці.

5 ОПИС ПРОГРАМНОГО СЕРЕДОВИЩА

Програмне забезпечення для відстеження персоналу збирає та аналізує інформацію про робочий час, повідомляючи про рівень продуктивності праці працівників. На рис.11 представлена структура основних елементів візуального інтерфейсу.

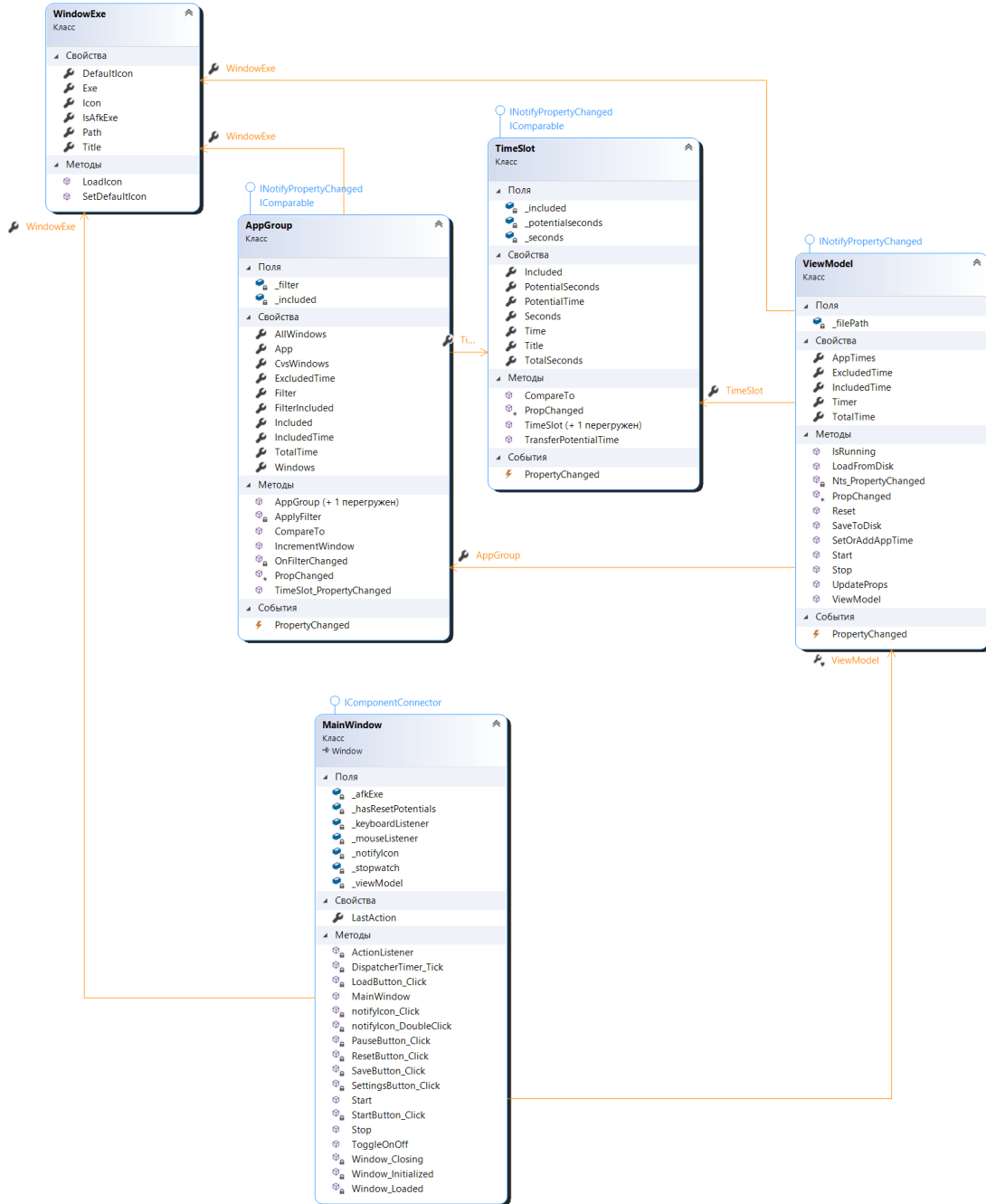


Рисунок 11 – Структура програми

MainWindow – основний клас, що описує дію компонентів візуального інтерфейсу. У конструкторі класу створюється об'єкт ViewModel. Одним з полів класу є об'єкт WindowExe, що відповідає за створення запису про відсутність користувача за комп'ютером:

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

У конструкторі класу подія класу ViewModel Timer.Tick підписується на метод DispatcherTimer_Tick. Також створюються слухачі для подій відсутності рухів мишею і натискання на клавіатуру.

```
public MainWindow()
{
    InitializeComponent();

    WindowExe.SetDefaultIcon(Properties.Resources.application);

    _viewModel = new ViewModel.ViewModel();
    _viewModel.Timer.Tick += DispatcherTimer_Tick;
    DataContext = _viewModel;

    _stopwatch = new Stopwatch();
    _stopwatch.Start();

    _mouseListener = new MouseHookListener(new GlobalHooker())
{ Enabled = true };
    _mouseListener.MouseDownExt += ActionListener;
    _mouseListener.MouseMove += ActionListener;
    _mouseListener.MouseWheel += ActionListener;

    _keyboardListener = new KeyboardHookListener(new GlobalHooker()) { Enabled = true };
    _keyboardListener.KeyDown += ActionListener;

    _viewModel.LoadFromDisk();

    Stop();
}
```

Обробник події DispatcherTimer_Tick встановлює або оновлює час використання запущених додатків, визи маючі методи класу ViewModel.

```
private void DispatcherTimer_Tick(object sender, EventArgs e)
{
    if (LastAction > Properties.Settings.Default.AFKDelay)
    {
        if (!_hasResetPotentials)
        {
            _hasResetPotentials = true;
        }
    }
}
```

```

        foreach (var window in
_viewsModel.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);
}
}

```

Клас `ViewModel` описує та зберігає значення таймеру, описує та зберігає список об'єктів `AppGroup` для котрих зберігаються часові параметри використання додатків. Методи `IncludedTime`, `ExcludedTime`, `TotalTime` повертають час активної роботи з додатком

```

public TimeSpan IncludedTime
{
    get { return new TimeSpan(0, 0, (int) AppTimes.Sum(o =>
o.IncludedTime.TotalSeconds)); }
}
public TimeSpan ExcludedTime
{
    get { return new TimeSpan(0, 0, (int) AppTimes.Sum(o =>
o.ExcludedTime.TotalSeconds)); }
}

public TimeSpan TotalTime
{
    get { return new TimeSpan(0, 0, (int) AppTimes.Sum(o =>
o.TotalTime.TotalSeconds)); }
}

```

Метод `SetOrAddAppTime` додає час використання до додатку, якщо такий додаток вже є у списку, та сортує час використання за спаданням для

внутрішніх вікон (наприклад, вкладки браузера), або додає новий запис у списку додатків AppTimes.

```

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 nts = new AppGroup(app);
        nts.PropertyChanged += Nts_PropertyChanged;
        nts.IncrementWindow(app.Title, lastAction, timeInc);
        AppTimes.Add(nts);
    }

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

```

Для збереження інформації після завершення роботи додатку створені методи LoadFromDisk та SaveToDisk

```

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);
}

```

Клас `AppGroup` зберіє час активної роботи, виключений час та загальний час роботи додатку.

Клас `TimeSlot` зберігає 2 поля з інформацією в секундах, та розраховує загальну кількість секунд.

Клас `WindowsExe` відповідає за збереження інформації про додаток, та отримання його іконки – метод `LoadIcon`.

```

public void LoadIcon()
{
    try
    {
        var icon =
System.Drawing.Icon.ExtractAssociatedIcon(Path);

        if (icon == null) return;
        using (var bmp = icon.ToBitmap())
        {
            var stream = new MemoryStream();
            bmp.Save(stream, System.Drawing.Imaging.ImageFormat.Png);
            Icon = BitmapFrame.Create(stream);
        }
    }
    catch (Exception)
    {
        Icon = DefaultIcon;
    }
}
}
}

```

Після запуску програми на екрані з'являється головне вікно (рис. 12).

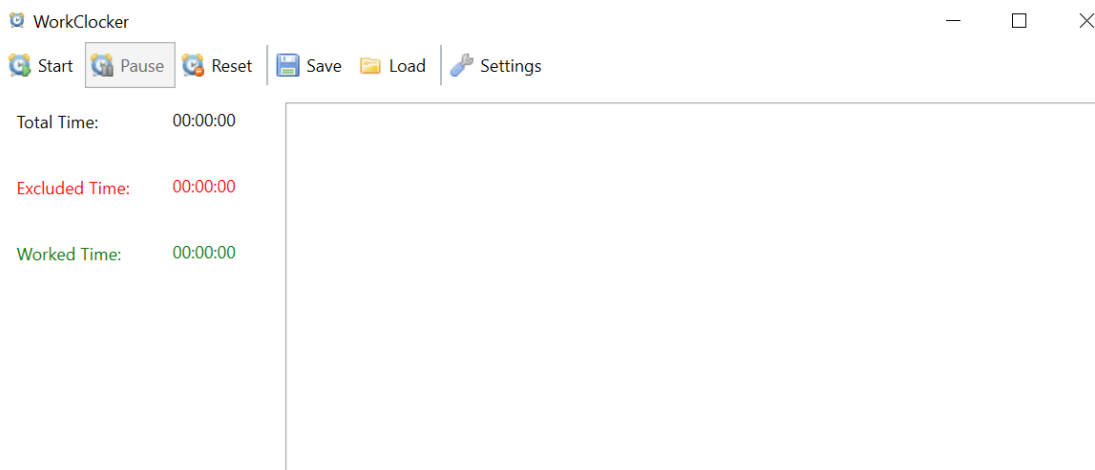


Рисунок 12 – Головне вікно програми

У верхній частині головного вікна знаходиться меню у якому користувач може:

- запустити трекінг;
- призупинити трекінг;
- обнулити трекінг;
- зберегти лог дій у файл;
- завантажити лог дій з файлу;
- перейти у налаштування (рис. 13).

У лівій частині знаходиться статистика:

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

Білий прямокутник – вікно логу дій. На рис. 14 лог дій відсутній, так як трекер вимкнений.

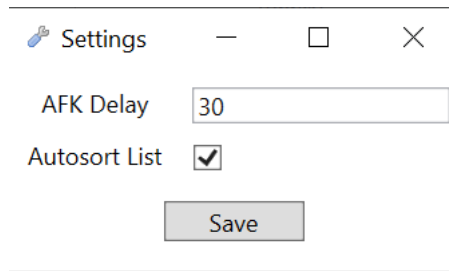


Рисунок 13 – Вікно налаштувань

У вікні налаштувань у полі «AFK Delay» можна вказати через яку кількість часу у секундах програма почне рахувати, що користувач відлучився від комп'ютера.

Відмітка «Autosort List» призначена для автоматичного сортування програм в логу за кількістю витраченого на них часу.

Після старту трекінгу можна відслідковувати витрачений час у вікні логу (рис. 14).

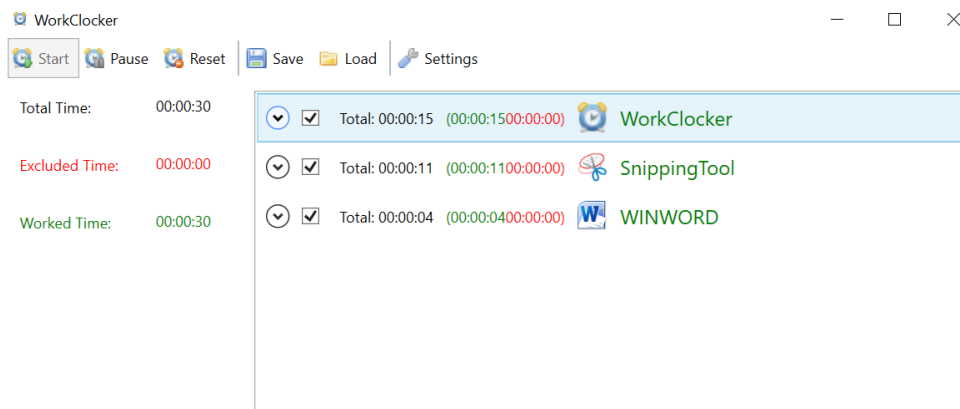


Рисунок 14 – Трекінг часу

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. 15).

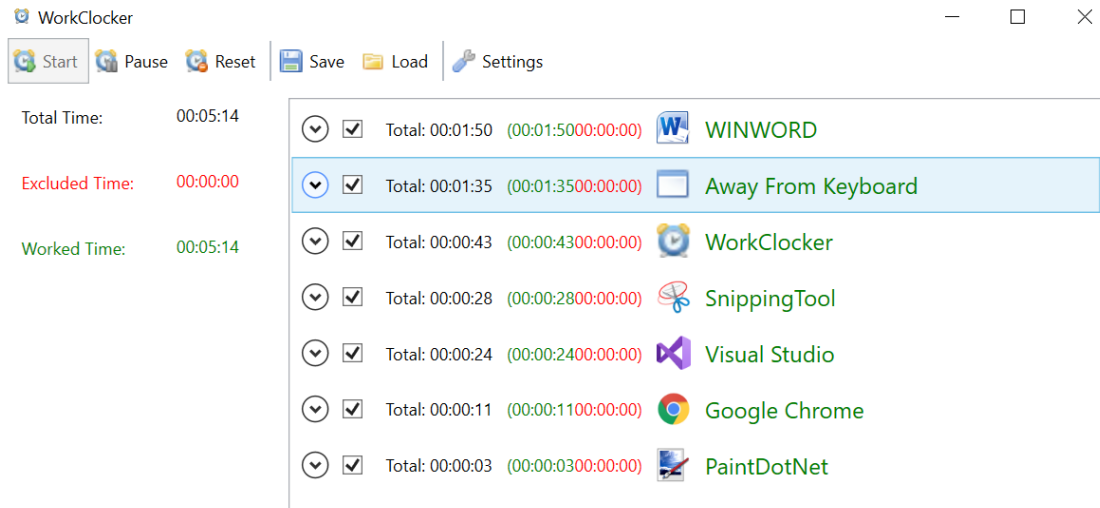


Рисунок 15 – Процес «Away From Keyboard»

Для детального контролю над вікном, кожний запис у лозі можна розгорнути, що представлено на рис. 16.

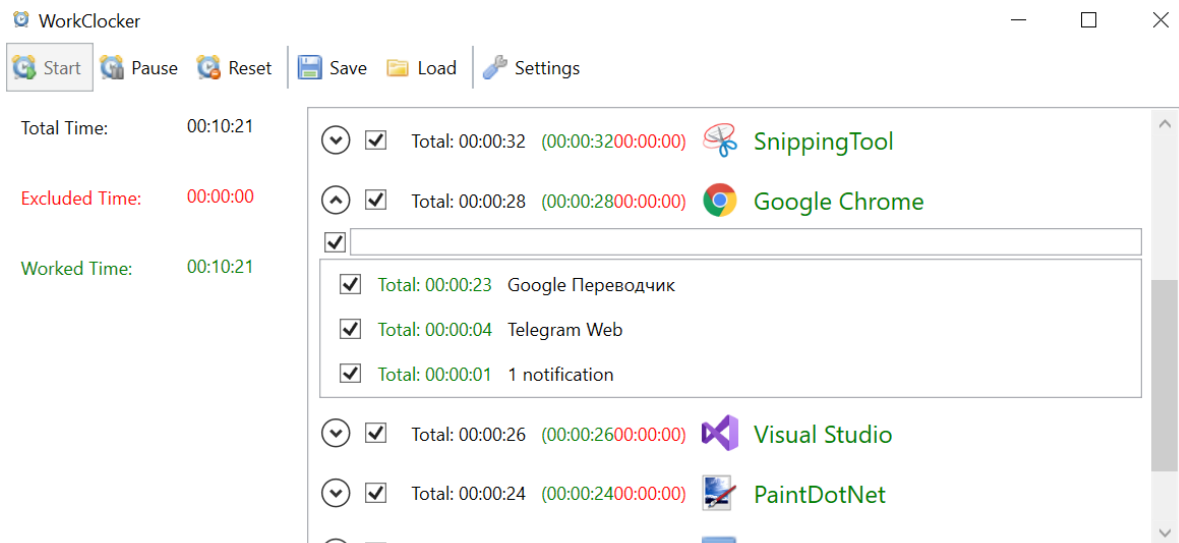


Рисунок 16 – Розгорнутий запис

За замовченням всі процеси у лозі вважаються робочими, необхідно вручну зняти мітки з неробочих процесів і це буде відображено у статистиці часу (рис. 17).

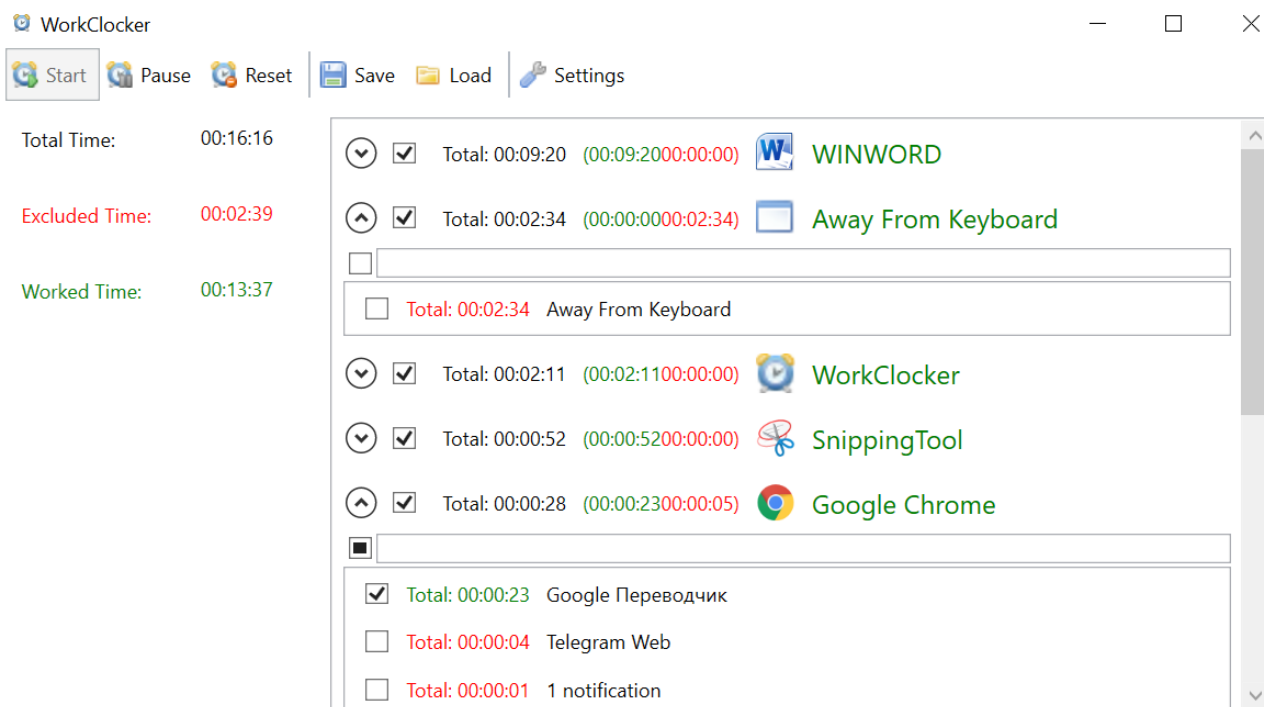


Рисунок 17 – Керування неробочим часом

При згортанні програмне вікно згортається у системний трей.

При завершенні роботи статистика автоматично зберігається у файлі та доступна при наступному завантаженні.

ВИСНОВКИ

Найкращий спосіб визначити, якими додатками користуються працівники, мінімізувати час перебування в Інтернеті та соціальних мережах, покращити відвідуваність або зменшити витрати на рідко використовувані ліцензії – це використання програмного забезпечення для моніторингу комп'ютера. Програмне забезпечення для моніторингу комп'ютера надає детальну інформацію, необхідну для аналізу всіх вищезгаданих комп'ютерних дій. За допомогою цього інструментів можна оцінити фактичну продуктивність праці співробітників, знати, хто наполегливо працює, відстає та працює у сферах, які потребують вдосконалення.

Моніторинг використання пристроїв співробітників може бути хорошим способом мотивувати розумно використовувати свій час. Деякі компанії рекомендують проводити наради щодо ефективності, щоб оцінити, як у всіх справи, та відзначити напрямки покращення. Відстеження та архівування діяльності співробітників також може бути корисним, якщо потрібні докази поганої поведінки незадоволеного працівника.

Програмне забезпечення для моніторингу працівників також може допомогти захистити бізнес та мережу. Наприклад, використовуючи функції моніторингу програмного забезпечення, можна бути впевненим, що працівники не переглядають шкідливі веб-сайти та не виконують шкідливих дій. Такі функції, як електронна пошта, USB та моніторинг друку, чудово допомагають переконатись, що конфіденційні дані не просочуються. Також можна використовувати моніторинг мережі для відстеження функціональності всіх ваших мережевих компонентів та зменшення простою мережі.

Програмне забезпечення для моніторингу співробітників включає широкий спектр функцій, щоб стежити за тим, як працівники проводять час на роботі. Найкраще програмне забезпечення для моніторингу дозволяє

адміністраторам встановлювати конкретні правила та політики для різних членів організації та отримувати сповіщення про порушення правил.

Також є можливість мати повний контроль над віддаленими працівниками за допомогою щоденних звітів про активний час, відстеження активності та скріншоти. Це також допомагає спростити процес нарахування заробітної плати, оскільки ви можете ввести погодинні ставки для кожного працівника в команді.

Результати дипломної роботи було створено програмне середовище, яке написано з використанням наступних технологій та мов програмування: .NET, Node.js, C, C #, C ++, Python, Visual Basic, F #, JavaScript, WPF, XAML, розглянуті сучасні технологічні платформи для реалізації подібних програм, наведено порівняльний аналіз можливих варіантів реалізації, проаналізовано його результати, реалізована аналітика, яка відстежує всі види взаємодій користувача зі сторінками додатка.

ПЕРЕЛІК ПОСИЛАНЬ

1. Топ-7 лучших бесплатных систем учета рабочего времени и контроля сотрудников 2021, рейтинг и сравнение URL: <https://www.kickidler.com/ru/for-it/methods-of-working/top-7-luchshix-besplatnyix-sistem-ucheta-rabochego-vremeni-i-kontrolya-sotrudnikov-2019.html> (дата звернення 09.03.2021)
2. Программы для отслеживания компьютеров персонала, что это и как с их помощью повысить эффективность работы. URL: <https://bitcop.ru/monitoring/slezhenie-za-kompyuterom-sotrudnikov> (дата звернення 15.03.2021)
3. Kickidler. URL: www.kickidler.com (дата звернення 16.03.2021)
4. Аналіз робочої сили та продуктивність праці. URL: <https://www.activtrak.com/> (дата звернення 16.03.2021)
5. Hubstaff. Time Tracking and Productivity Monitoring Tool. URL: hubstaff.com (дата звернення 16.03.2021)
6. DeskTime: A Time Tracking App For Businesses & Freelancers. URL: deskttime.com (дата звернення 19.03.2021)
7. Toggl. URL: <https://toggl.com/> (дата звернення 19.03.2021)
8. Screenshotmonitor. URL: <https://screenshotmonitor.com/> (дата звернення 20.03.2021)
9. Бесплатное приложение для учета времени – Tmetric. URL: tmetric.com (дата звернення 20.03.2021)
10. Краткий обзор языка/ URL: C#/ <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> (дата звернення 19.03.2021)
11. Visual Studio Лучшие в своем классе средства для разработчиков. URL: <https://visualstudio.microsoft.com> (дата звернення 20.03.2021)
12. Руководство по WPF. URL: <https://metanit.com/sharp/wpf/> (дата звернення 20.03.2021)

13. Обзор XAML (WPF.NET). URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/fundamentals/xaml?view=netdesktop-5.0> (дата звернення 20.03.2021)