

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

**«Розробка інформаційно-аналітичної системи для
оптимізації процесу інвентаризації ІТ-активів в
доменній мережі підприємства»**

Виконав студент групи ЗПЗмд-23-1
ВЕЛИКОБОРЕЦЬ Олег Анатолійович

Керівник роботи:
ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:
БАКАЛЮК Тетяна Анатоліївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
 ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
 програмного забезпечення
 Тетяна ВАКАЛЮК
 «14» лютого 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу

Здобувач вищої освіти: **ВЕЛИКОБОРЕЦЬ Олег Анатолійович**

Керівник роботи: **ЛОБАНЧИКОВА Надія Миколаївна**

Тема роботи: **«Розробка інформаційно-аналітичної системи для процесу інвентаризації ІТ-активів в доменій мережі підприємства»**, затверджена
 Наказом закладу вищої освіти від «14» лютого 2025 р. №61/с

Вихідні дані для роботи:

•

Керівники / консультанти з магістерської кваліфікаційної роботи із
 зазначенням розділів, що їх стосуються:

Розділ	Керівник/ консультант	Завдання видав	Завдання прийняв
1	Лобанчикова Н.М	15.02.2025р.	15.02.2025р.
2	Лобанчикова Н.М.	15.02.2025р.	15.02.2025р.
3	Лобанчикова Н.М.	15.02.2025р.	15.02.2025р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена розробці інформаційно-аналітичної системи для здійснення процесів інвентаризації шляхом автоматизації збору даних про ІТ-активи в межах доменної мережі підприємства. Пояснювальна записка складається з вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 79 сторінок, 27 ілюстрацій, 1 таблицю, 1 додаток та 25 посилань на зовнішні джерела.

Метою роботи є розробка інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

Об'єктом дослідження є технології розробки інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

Предметом дослідження є методи, алгоритми та технології розробки інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

В роботі описано процес створення скрипту, що призначений для збереження переліку ключових даних, котрі можуть становити інтерес з точки зору ІТ підрозділу підприємства, а також бази даних та програми для подальшого відображення, пошуку та аналізу зібраних даних. Проведено тестування програми.

КЛЮЧОВІ СЛОВА: ІНВЕНТАРИЗАЦІЯ, АВТОМАТИЗАЦІЯ, АНАЛІЗ ДАНИХ.

ABSTRACT

Master's thesis is devoted to the development of informational-analytic system to assist the inventarization process by automating IT-assets data collection within enterprise's domain network. The explanatory note consists of an introduction, three chapters and a list of used literature. The thesis note contains 79 pages, 27 illustrations, 1 table, 1 appendix and 25 links to external sources.

The purpose of the work is researching optimization methods of the computer's hardware and software inventarisation process summing up to development of an information-analytic system for automated data collection and analysis within the enterprise's domain network.

The object of research is the development technology of informational-analytic system to optimize IT-assets inventarisation processes within the enterprise's domain network.

Subjects of research are methods, algorithms and development technologies of informational-analytic system to optimize IT-assets inventarisation processes within the enterprise's domain network.

The work describes the process of creating a script that is purposed to store the collection of key data, that is deemed to be useful from IT department's viewpoint. As well as both database and software sample for later viewing, searching and analyzing of collected data. The program has been tested.

KEY WORDS: INVENTARIZATION, AUTOMATING, DATA ANALYSIS.

ЗМІСТ

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Аналіз проблем і особливостей процесу інвентаризації	9
1.2.Аналіз наявних застосунків для здійснення інвентаризації.....	13
1.3.Постановка задачі дослідження та вибір засобів реалізації.....	18
1.4. Методи оптимізації	20
Висновки до першого розділу	24
РОЗДІЛ 2. РОЗРОБКА ТА ПРОЕКТУВАННЯ ЕЛЕМЕНТІВ СИСТЕМИ ЗБОРУ ДАНИХ.....	25
2.1 Розробка скрипту для збору даних	25
2.2 Розробка бази даних	40
2.3 Розробка додатку на VB.NET	43
РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ	50
3.1 Порядок запуску і роботи	50
3.2 Робота у вкладці «Search & Data»	52
3.3 Робота у вкладці «Analytics»	58
Висновки до третього розділу	66
ВИСНОВКИ.....	67
ДОДАТКИ	72

ВСТУП

Актуальність теми. Проблеми обліку та супроводу обчислювальної комп'ютерної техніки є актуальними для переважної більшості сучасних підприємств. «Вагоме значення у досягненні бізнес-цілей будь-якої організації в умовах цифровізації має ІТ-інфраструктура, яка здійснює підтримку і обслуговування споживачів, підвищує продуктивність праці та прибутковість, підтримує бізнес-процеси і сприяє їх гнучкості та результативності, забезпечує стабільну роботу в цілодобовому режимі інформаційних систем, дозволяє автоматизувати всі процеси, мінімізує витрати, забезпечує безпеку електронної пошти та інших даних. Разом з цим, ефективний розвиток організації чи підприємства та підвищення рівня їх конкурентоспроможності можливі за умови наявності розвинутої ІТ-інфраструктури» [1].

Досить часто, в умовах сьогодення, підприємства мають розгалужену мережу представництв, філій. Це, в свою чергу, призводить до необхідності відповідного програмно-апаратного забезпечення кожного підрозділу. Керування подібною кількістю обчислювальної техніки зазвичай здійснюється за рахунок об'єднання пристроїв в єдину доменну мережу підприємства з чітко визначеними правилами для подальшого контролю та супроводу. Супровід додатково ускладнюється тим фактом, що відокремлені підрозділи часто мають суттєву віддаленість одна від одної, а забезпечення кожного представництва працівником ІТ департаменту вбачається недоцільним, тож більшість завдань, по можливості, виконуються віддалено. В таких умовах навіть відносно просте завдання – інвентаризація комп'ютерної техніки може ускладнитись, так як віддалений доступ не завжди можливий, транспортування працівника, за умов відсутності скарг на роботу електронно-обчислювальної техніки, виключно для збору інформації є недоцільним, а користувачі не завжди можуть самостійно надати достовірну інформацію про наявну в них комп'ютерну техніку. За таких умов звести дані в масштабах усього підприємства виявляється непростим,

трудомістким заняттям, котре до того ж включає ризики помилок по причині людського фактору. Враховуючи додатково розтягнутість по часу даного процесу є висока ймовірність того, що результати інвентаризації частково втратять свою актуальність та виявляться застарілими ще до її завершення.

В таких випадках автоматизація вищезазначених процесів збору даних – це саме те рішення котре дозволяє першочергово якщо не усунути, то значно скоротити ризики неточностей викликані людським фактором, суттєво прискорюючи та спрощуючи процес інвентаризації. Одночасно з тим наявність «під рукою» актуальної інформації про обчислювальну техніку суттєво спрощує роботу ІТ-фахівців та дозволяє більш детально планувати закупівлі з метою модернізації обладнання там де це найбільш необхідно, відкриває можливості глибшого аналізу зібраних даних та прийняття ефективніших рішень на основі проведеної аналітики.

Мета роботи – розробка інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

- провести аналіз інструментів для здійснення інвентаризації вузлів мережі;
- дослідити способи збереження, обробки та аналізу отриманих даних;
- розробити методи оптимізації процесу інвентаризації в доменній мережі підприємства;
- створити додаток що реалізує запропоновані методи.

Об’єктом дослідження є технології розробка інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

Предметом дослідження є методи, алгоритми та технології розробки інформаційно-аналітичної системи для оптимізації процесу інвентаризації ІТ-активів в доменній мережі підприємства.

оптимізації процесів інвентаризації комп'ютерної техніки в доменній мережі підприємства, процесів автоматизації ідентифікації та подальший аналіз зібраних даних.

Наукова новизна роботи полягає у:

- проведеному аналізу та узагальнені інструментів для здійснення інвентаризації вузлів мережі, що визначило переваги та недоліки кожного;
- проведеному дослідженні способів збереження, обробки та аналізу отриманих даних, що дозволило визначити функціональність та потреби програмного продукту;
- розроблених методів оптимізації процесу інвентаризації в доменній мережі підприємства для автоматизації отримання актуальних даних про обчислювальну техніку без провокування зайвих ризиків з точки зору мережевої безпеки.

Практичне значення полягає у програмній реалізації методів оптимізації процесу інвентаризації та проведення аналітики ІТ-активів в доменній мережі підприємства для автоматизації процесів та спрощення роботи ІТ-фахівців.

Апробація. Результати проведеного дослідження за темою кваліфікаційної магістерської роботи було опубліковано тези у Збірнику наукових праць «Технічна творчість» [2].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз проблем і особливостей процесу інвентаризації

На сьогоднішній день використання сучасних технологій є необхідністю в усіх сферах діяльності, і зі збільшенням масштабів підприємства більш нагальною стає потреба використання передових технологій для успішної діяльності. «Для ефективного управління необхідно аналізувати не тільки внутрішню, але й зовнішню інформацію. У зв'язку із чим виникає потреба в оновленні облікової системи підприємств для адаптації до викликів цифрової трансформації. Операційна аналітика дозволяє приймати швидкі та обґрунтовані рішення для підвищення якості операційного управління та ефективності бізнес-процесів. Для вдосконалення аналітичного забезпечення операційного управління ІТ-підприємств необхідно вдосконалити облікову політику для підприємств ІТ-індустрії та забезпечити впровадження сучасних інструментів бізнес-аналітики та створення єдиного інформаційного простору підприємства для отримання інформації в реальному часі, впровадження системи підвищення рівня кваліфікації у зв'язку зі швидкою зміною технологій збору, накопичення, зберігання та аналізу інформації»[3]. «Аналітика – це наука аналізу, яка застосовується у різних сферах життя – в політиці, бізнесі, фінансах, на ринках товарів і послуг, у страховій справі, у соціальній сфері тощо, на різних рівнях управління: національному, регіональному, на рівні підприємств усіх форм власності. Особлива увага приділяється бізнесу, тобто суб'єктам господарювання, метою діяльності яких є отримання прибутку, й аналізу окремих бізнес-процесів. Бізнес-аналітика передбачає застосування під час аналізу явищ та процесів широкого кола аналітичних показників, статистичних і математичних методів, комп'ютерних технологій і програмного забезпечення для прийняття управлінських рішень» [4].

Розглянемо основні типи аналітики. Описова аналітика – визначає, що сталося в минулому - базовий рівень аналітики, який допомагає зрозуміти, що

відбулося, шляхом підсумовування та опису даних. Діагностична аналітика – визначає, чому це сталося. Цей тип аналітики аналізує дані, щоб зрозуміти причини подій чи проблем. Прогнозна аналітика – визначає, що може статися в майбутньому, використовує історичні дані та алгоритми машинного навчання для прогнозування можливих подій. Пропозиційна аналітика – рекомендує, що потрібно зробити, допомагає знайти оптимальні рішення, враховуючи майбутні умови та прогнозні моделі. Когнітивна аналітика – використовує технології штучного інтелекту (ШІ), щоб імітувати людське сприйняття, прийняття рішень та навчання, надаючи більш глибокі, інтелектуальні висновки. Загалом використання аналітичних методів дозволяє вирішити цілий ряд проблем, в тому числі й у сфері інвентаризації. «Проблеми і труднощі інвентаризації виникають при появі помилок під час самого процесу проведення інвентаризації майна. На більшості підприємств інвентаризація проводиться ручним способом на паперових носіях, які в свою чергу, при великій кількості основних засобів, що припадають на одну матеріально-відповідальну особу, створюють ряд складнощів і проблем при зборі даних, зведенні фактичних відомостей з обліковими даними і оформленні результатів. А також хочеться зазначити, що проведення інвентаризації ручним способом, є однією з причин значних трудовитрат і досить низького рівня контролю майна. Інвентаризація є досить трудомісткою процедурою, пов'язаною з ручним методом її проведення і великою кількістю інвентарних об'єктів» [5]. «Інвентаризація може бути покращена за рахунок використання сучасних технологій, таких як автоматизовані системи обліку та ідентифікації, які знижують ризик помилок, прискорюють процес і забезпечують точніші результати»[6].

В розрізі сфери ІТ інвентаризація стосується як обладнання так і програмного забезпечення без яких неможливим є використання сучасних технологій на переважній більшості підприємств. В процесі інвентаризації здійснюється контроль за збереженням майна, тобто перевірка фактичної наявності відповідних засобів, оцінюється технічний стан, виявляються

несправності. «З розвитком обліку та його технічної бази методика й порядок проведення інвентаризації вдосконалюються. Зокрема, використання сучасної комп'ютерної техніки та впровадження автоматизованого робочого місця в організацію обліку дають змогу підвищити оперативність обліку» [7]. «Головною метою інвентаризації є виявлення розбіжностей між обліковими даними, щодо майна та його фактичною наявністю і станом. Ключовими завданнями інвентаризації є контроль за наявністю і рухом майна, його використанням у відповідності до визначених норм. Виявлення майна, що втратило свої якісні характеристики і не відповідає технічним вимогам, майна, що не відображено в обліку. Засобів, котрі використовуються з порушенням способу застосування та інше»[8]. «Також до завдань можна віднести: пошук непридатних для використання цінностей та їх подальше вилучення; виявлення майна, котре знаходиться поза обліком»[9]. Тож найбільшу складність вбачаємо у процесі зіставлення фактичних даних з даними бухгалтерського обліку для виявлення розбіжностей. «За результатами інвентаризація запасів може мати 3 результати: надлишок, нестача та пересортиця»[10]. Якщо розбіжності виявлено нерідко причиною тому є незафіксоване в бухгалтерському обліку з тих чи інших причин переміщення техніки. Відслідкувати в такому випадку послідовність переміщень може бути непросто і навіть просто актуального списку осіб котрі використовують в даний момент часу аналогічну модель пристрою було б достатньо для значного прискорення процесу пошуку. Якщо з такого списку виключити осіб на котрих, відповідно до бухгалтерського обліку, дійсно рахуються такі пристрої дозволить миттєво звужити коло пошуку до декількох осіб. Однак зазвичай подібна інформація відсутня що призводить до значних затримок процесу інвентаризації, та навіть, в окремих випадках, ігнорування подібних помилок, що лише призводить до їх накопичення в подальшому. «Ефективність і результативність інвентаризаційного процесу неможливо забезпечити без наявності раціональної організації та чіткої методики проведення інвентаризації. Відсутність їх на практиці призводить до

формалізації інвентаризаційної роботи, недотримання правильності та послідовності виконання окремих інвентаризаційних процедур, порушень чинних законодавчо-нормативних актів. Внаслідок цього інвентаризація як методичний прийом економічного контролю втрачає своє значення, не виконуються взагалі або виконуються частково поставлені перед нею мета й завдання»[11].

Не менш важливим аспектом вважаємо контроль за встановленим програмним забезпеченням та, особливо, ліцензіями. Подібний контроль, залежно від методів впровадження, може в тій чи іншій мірі запобігти проблемі несанкціонованого використання небажаного/небезпечного/неліцензійного програмного забезпечення. Також належний контроль дозволить ефективніше розподіляти існуючі ліцензії та вчасно вживати заходи для закупівлі нових, або продовженні терміну дії існуючих ліцензій. Найбільшу загрозу в даному випадку вбачаємо у самовільному використанні неліцензійних програмних продуктів користувачами, що може потенційно спричинити суттєві збитки для підприємства у випадку виявлення подібних порушень та накладення відповідних штрафних санкцій по відношенню до підприємства. Хоча обмеження прав користувачів в сфері встановлення програмного забезпечення суттєво зменшує ризик, появи несанкціонованого програмного продукту на комп'ютері - все-ж таки лишається імовірність використання портативних версій програмного продукту, нерідко «зламаних». Подібне також може, в свою чергу, викликати додаткові ризики інформаційної безпеки, адже неконтрольоване розповсюдження зловмисного програмного коду може мати руйнівні наслідки, наприклад масову втрату даних. Запобігання виникненню подібних випадків є, в тому числі, одним з обов'язків ІТ-фахівців. Одним з методів що використовуються в даному випадку – сегментування мережі на окремі підмережі, комп'ютери в межах котрих зможуть напряду взаємодіяти лише з пристроями в межах своєї підмережі і не матимуть змоги безпосередньо контактувати з комп'ютерами з інших підмереж. Подібні методи дозволяють

ефективно локалізувати імовірні загрози запобігаючи їх неконтрольованому розповсюдженню в короткі строки, але, одночасно з тим, накладають певні обмеження на роботу в мережі. Такими обмеженнями можуть бути, наприклад, проблеми у спільному використанні друкувальної техніки або складнощі при віддаленому підключенні у випадку виникнення такої необхідності для вирішення поставлених задач. Подібні заходи безпеки ускладнюють й збір інформації про комп'ютерну техніку, однак, враховуючи пов'язані ризики, вважаємо безпеку мережі більш пріоритетною ніж зручність певного функціоналу.

1.2. Аналіз наявних застосунків для здійснення інвентаризації

«Бурхливий розвиток комп'ютерної техніки та сучасних інформаційних технологій зумовлює необхідність визначення шляхів створення ефективної методики аудиту на основі електронної обробки облікової інформації. Своєю чергою, реальний стан програмного забезпечення дає змогу значно розширити сферу застосування автоматизації в процесі аудіювання, залучивши комп'ютер до виконання логічних, аналітичних і розрахункових процедур» [12]. «Комп'ютеризація процедури здійснення інвентаризації цінностей потребує створення комп'ютерних програм блоку проведення інвентаризації цінностей і оформлення її результатів. Програма виконання процедур інвентаризації при комп'ютеризації здійснюється за заданим алгоритмом, за допомогою якого виконується програма послідовності процедур контролю під час інвентаризації цінностей, що забезпечує досягнення бажаного результату» [13]. Одночасно з тим варто зазначити що впровадження нового програмного продукту на підприємстві може викликати певні труднощі, адже кошти на придбання платних продуктів виділяються неохоче, а безкоштовні продукти неминуче викликають сумніви в питанні безпечності їх використання.

Пропонуються досить різноманітні варіанти використання спеціалізованого програмного забезпечення та обладнання (в тому числі сканери

штрих-кодів, QR-кодів, RFID-мітки) для автоматизації збору та обробки даних. «Використання сучасних технологій у процесі інвентаризації не змінює звичайну методику проведення, а доповнює її, допомагаючи прискорити процес, підвищити точність та мінімізувати людський фактор. Кожне підприємство має вирішити для себе, яке обладнання та програмне забезпечення буде найбільш ефективно справлятися з поставленою задачею, без необхідності призупиняти роботу підрозділу, що перевіряється» [14]. «Прерогативами проведення інвентаризації програмних продуктів компанії за допомогою професійних програм є: змога паралельної ревізії АЗ і ПЗ, і перевірки їх порізно; існування даних про користувачів, МВО, формування звітів тощо» [15].

Мабуть, найпершим та найпростішим програмним забезпеченням що широко використовується для потреб інвентаризації є таблиці Microsoft Excel (див. рис.1.1). Подібне рішення зазвичай не потребує додаткових витрат, адже нерідко підприємство вже використовує пакет програмного забезпечення Microsoft Office у повсякденній роботі, а табличний формат зазвичай є найбільш логічним та зручним способом збереження подібної структурованої інформації.

№ п/п	Відділення	Дільниця	Посада користувача	Процесор	ОС	НDD	Ємність НС	Операційна система
1	Бердичівська	Бердичівська	Майстер служби експлуатації і ПРГ	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
2	Бердичівська	Бердичівська	Технік 1-ої категорії служби експлуатації і ПРГ	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
3	Бердичівська	Бердичівська	Провідний бухгалтер відділу зарплатної плати	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	4 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
4	Бердичівська	Бердичівська	Майстер служби експлуатації ПРГ	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	4 GB	Geofram	240 GB	Майкрософт Windows 10 Pro
5	Бердичівська	Бердичівська	Оператор з уведення даних в ЕОМ	Intel(R) Pentium(R) CPU G645 @ 2.90GHz	2 GB	Hitachi HD57210500LE630	500 GB	Майкрософт Windows 7 Pro
6	Бердичівська	Бердичівська	Файлвець (з балансування) служба обліку природного газу	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
7	Бердичівська	Бердичівська	Секретар	Intel(R) Pentium(R) CPU G2020 @ 2.90GHz	2 GB	Hitachi HD57210500CLA	500 GB	Майкрософт Windows 7 Ultimate
8	Бердичівська	Бердичівська	Провідний інженер відділу інвестицій	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
9	Бердичівська	Бердичівська	Архіварус: технічної документації виробничо-технічний відділ	Pentium(R) Dual Core CPU E5800 @ 3.2GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
10	Бердичівська	Бердичівська	Майстер служби обліку природного газу	Intel(R) Celeron(R) CPU J1800 @ 2.41GHz	4 GB	ST5000312CS	500 GB	Майкрософт Windows 10 Pro
11	Бердичівська	Бердичівська	Оператор з уведення даних в ЕОМ	Intel(R) Pentium(R) CPU G630 @ 2.70GHz	4 GB	WDC WD2500AAJX	500 GB	Майкрософт Windows 7 Pro
12	Бердичівська	Бердичівська	Майстер служби експлуатації ВЕГО	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
13	Бердичівська	Бердичівська	Майстер АДС	Intel(R) Pentium(R) CPU G2030 @ 3.0GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
14	Бердичівська	Бердичівська	Майстер служби ЕКЗ, КПО	AMD E-300 1.3GHz	2 GB	ST9160821AS	160 GB	Windows 7 Ultimate
15	Бердичівська	Бердичівська	Майстер служби ЕКЗ, КПО	Intel(R) Celeron(R) CPU @ 2.00GHz	1 GB	Samsung SP0411N	40 GB	Майкрософт Windows XP
16	Бердичівська	Бердичівська	Провідний інженер з метрології	Intel(R) Pentium(R) G2010 CPU @ 3.00GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
17	Бердичівська	Бердичівська	АДС	Intel(R) Pentium(R) CPU G645 @ 2.90GHz	2 GB	Hitachi HD57210500LE630	500 GB	Windows 7p
18	Бердичівська	Бердичівська	Слосар з експлуатації та ремонту газового устаткування	Intel(R) Pentium(R) Dual CPU E2160 @ 1.80GHz	512MB	Samsung HD161H	160 GB	Windows XP
19	Бердичівська	Бердичівська	Провідний інженер з капітального будівництва	Intel(R) Pentium(R) CPU G2020 @ 2.90GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Pro
20	Бердичівська	Бердичівська	Старший механік	Intel(R) Celeron(R) CPU E3300 @ 2.50GHz	2 GB	WDC WD2500AAJS	250 GB	Windows XP
21	Бердичівська	Бердичівська	Диспетчер автомобільного транспорту	Intel(R) Pentium(R) G2030 CPU @ 3.00GHz	2 GB	ST5000M002	500 GB	Майкрософт Windows 7 Ultimate
22	Бердичівська	Бердичівська	Провідний інженер з охорони праці	Intel(R) Pentium(R) CPU E3300 @ 2.40GHz	1 GB	ST1160811AS	160 GB	Windows 7 Ultimate
23	Бердичівська	Бердичівська	Головний інженер	Intel(R) Celeron(R) CPU G2930 @ 2.90GHz	4 GB	Toshiba DT01ACA050	500 GB	Майкрософт Windows 10 Pro
24	Бердичівська	Бердичівська	Комірник	Intel(R) Pentium(R) CPU G2020 @ 2.90GHz	2 GB	Hitachi HD57210500CLA	500 GB	Майкрософт Windows 10 Pro
25	Бердичівська	Бердичівська (ЦОК)	Файлвець з обслуговування клієнтів	Intel(R) Celeron(R) CPU G3930 @ 2.90GHz	4 GB	Toshiba DT01ACA050	500 GB	Майкрософт Windows 10 Pro
26	Бердичівська	Бердичівська (ЦОК)	Файлвець з обслуговування клієнтів	Intel(R) Celeron(R) CPU G3930 @ 2.90GHz	4 GB	Toshiba DT01ACA050	500 GB	Майкрософт Windows 10 Pro
27	Бердичівська	Чуднів	Провідний файлвець служби обліку природного газу	Intel(R) Pentium(R) G2030 CPU @ 3.00GHz	2 GB	ST5000M002-1B0142	500 GB	Майкрософт Windows 7 Pro
28	Бердичівська	Чуднів	Архіварус	Intel(R) Pentium(R) E3400 CPU @ 2.60GHz	2 GB	ST1160811AS	160 GB	Майкрософт Windows 7 Pro
29	Бердичівська	Чуднів	Секретар	Intel(R) Pentium(R) G2020 CPU @ 2.90GHz	2 GB	ST5000M002-1B0142	500 GB	Майкрософт Windows 7 Pro
30	Бердичівська	Чуднів	Провідний інженер ВТВ	Intel(R) Pentium(R) G2020 CPU @ 2.90GHz	2 GB	Hitachi HD57210500CLA360	500 GB	Майкрософт Windows 7 Pro
31	Бердичівська	Чуднів	Повідний інженер ВТВ	Intel(R) Pentium(R) G2020 CPU @ 2.90GHz	2 GB	ST5000M002-1B0142	500 GB	Майкрософт Windows 7 Pro

Рис. 1.1. Інтерфейс додатку Microsoft Excel (web)

Крім того, додатковий функціонал у вигляді формул, фільтрів та зведених таблиць дозволяє проводити базовий аналіз наявних даних та їх відображення у зручному для використання вигляді.

До недоліків даного методу можна віднести суттєву вразливість до помилок при вводі даних працівником, особливо якщо таблиця є спільною і заповнюється одночасно кількома працівниками. Загалом, хоча універсальність даного програмного продукту й дозволяє його використовувати, в тому числі, й для потреб інвентаризації, це не є його основним призначенням тож про можливості автоматизації не йдеться. До переваг окрім відносної простоти й доступності можна віднести й особисту ознайомленість. Адже регулярно маючи справу в повсякденній роботі з даним програмним комплексом багато користувачів нададуть перевагу використанню вже добре знайомої програми аніж витратитимуть час на освоєння нової, до того ж функціонал котрої, потенційно, може й не задовольнятиме наявні потреби.

Snipe-IT – це система управління активами з відкритим кодом, яка розроблена на основі Laravel спеціально для управління та відстеження ІТ-активів у межах організації з можливістю платного хостингу на стороні

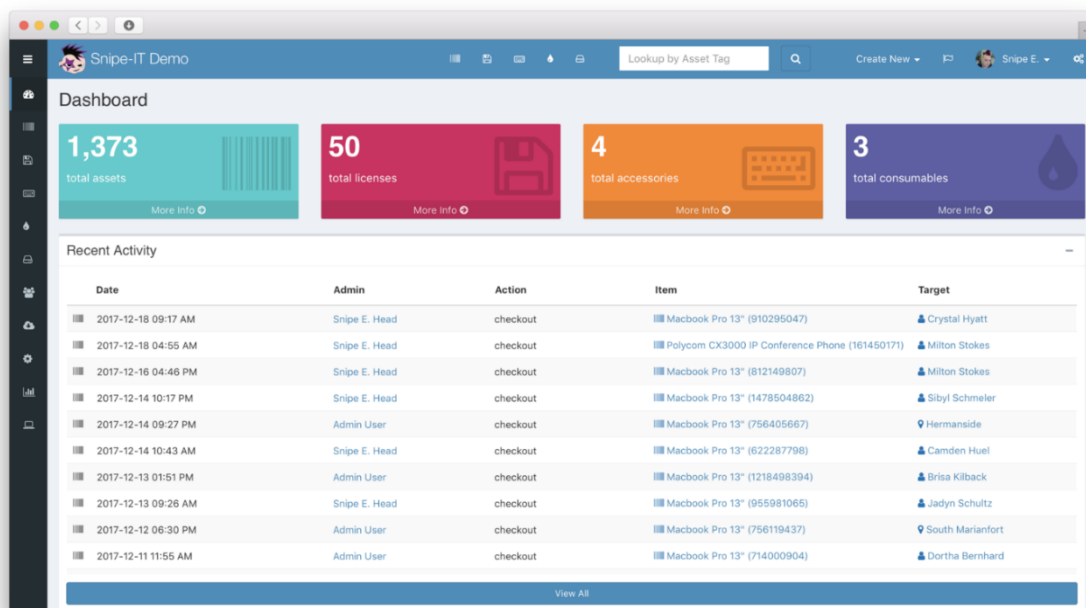


Рис. 1.2. Інтерфейс додатку Snipe-IT

розробника або безкоштовного використання у випадку розгортання сервісу на власних потужностях без втрати функціоналу (див. рис.1.2).

Snipe-IT дозволяє організаціям вести детальний облік ІТ-активів, таких як комп'ютери, монітори, мобільні пристрої, сервери тощо. Користувачі можуть записувати інформацію про кожен актив, включаючи модель, серійний номер, дату покупки та статус гарантії. Платформа дозволяє відстежувати, хто користується кожним активом, прив'язуючи його до конкретних користувачів. Ця функція допомагає в управлінні призначеннями активів і розумінні розподілу ресурсів усередині організації. Додаток доступний через веб-інтерфейс, що дозволяє користувачам управляти активами з будь-якого місця, за умов наявності інтернет-доступу. Однак з точки зору автоматизації даний продукт не має своїх засобів збору даних натомість пропонуючи REST API для підключення самостійно створених агентів збору даних.

Total Network Inventory 6 програма для обліку комп'ютерів. Ключові можливості даного додатку: віддалене сканування Windows та Unix-подібних систем, VMware, SNMP та інших пристроїв; сканування комп'ютерів резидентним агентом; інвентаризація та облік обладнання та програмного забезпечення; звіти будь-якої складності; планувальник завдань; повідомлення про проблеми з комп'ютерами та програмним забезпеченням; журнал змін обладнання та ПЗ; аудит програмного забезпечення (SAM); модуль обліку ліцензій ПЗ; обчислення статусу ліцензій та зберігання ліцензійних ключів ПЗ; моніторинг апаратних датчиків; створення детальної карти мережі.

Загалом програмний комплекс пропонує значний пакет можливостей для збору та обробки інформації як всередині доменної мережі, так і поза її межами – програма може сканувати віддалені комп'ютери через інтернет, за умови, що необхідні налаштування мережі та брандмауера дозволяють такий доступ (див. рис.1.3). Загалом подібний програмний комплекс є чудовим варіантом пропонуючи повну автоматизацію збору даних. Однак можемо виокремити 2 суттєвих недоліки що є спільними в переважній більшості аналогічних рішень, а

саме: вартість – даний програмний продукт є платним (ціна залежить від кількості підтримуваних робочих місць); спосіб збору даних – враховуючи те, що збір даних відбувається за допомогою мережевого підключення потрібно відкривати порти для безпосереднього доступу програми до усіх комп’ютерів підприємства, що в свою чергу може бути неприйнятним з точки зору політики інформаційної безпеки підприємства.

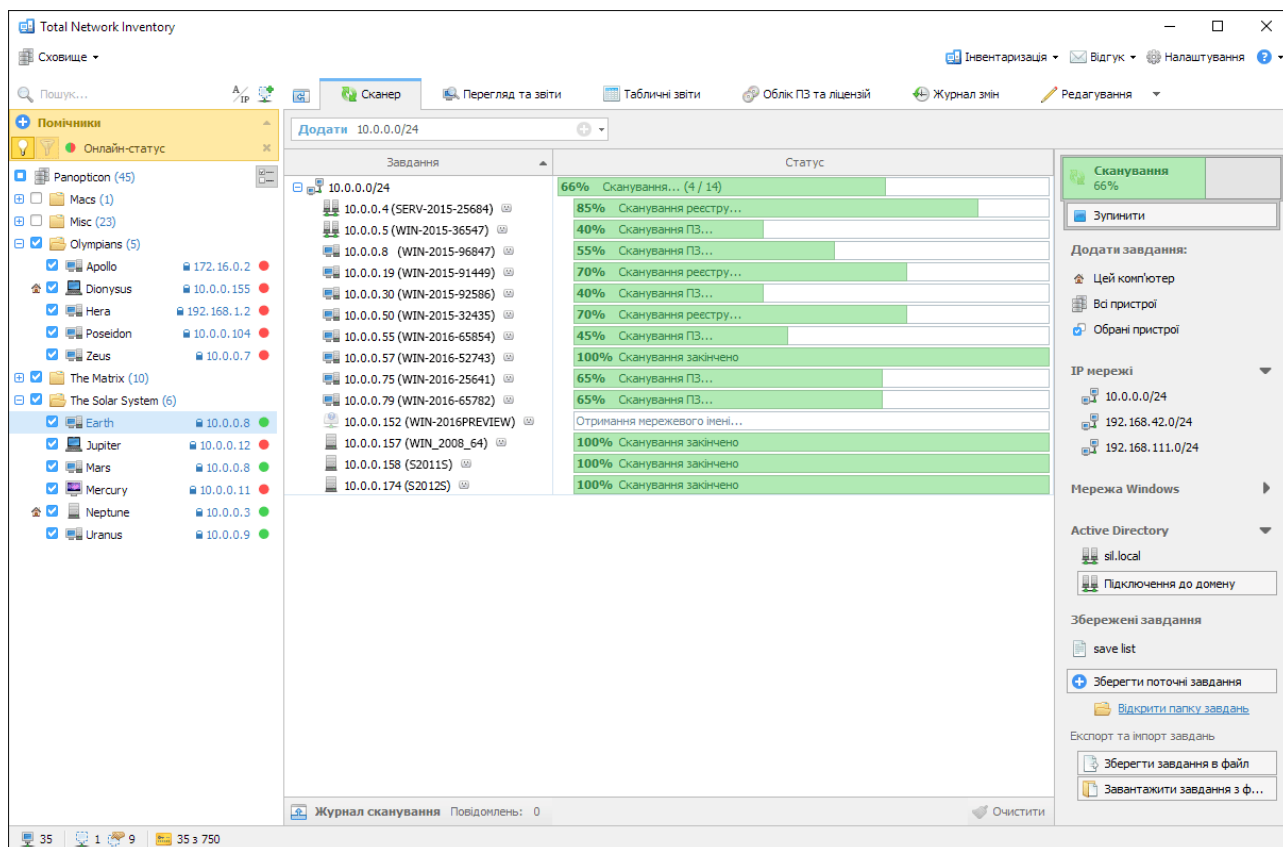


Рис. 1.3. Інтерфейс додатку Total Network Inventory

Порівняння інструментів для здійснення інвентаризації представлено в таблиці 1.1. Проаналізувавши усі переваги та недоліки можна виявити основні аспекти, які мають бути виправлені та покращені у майбутньому додатку. Найголовнішим недоліком є відсутність автоматизованого механізму збору даних, котрий не покладався б на безпосереднє підключення до віддаленого комп’ютера. Можемо сформулювати перше уявлення про те, як має функціонувати програмний додаток. Він має давати можливість автоматизованого збору

інформації про комп'ютери, пошуку комп'ютерів по заданому параметру, зведення даних.

Таблиця 1.1

Порівняння інструментів для здійснення інвентаризації

	Microsoft Excel	Snipe-IT	Total Network Inventory
Складність впровадження	Низька (скоріш за все додаток вже використовується на підприємстві)	Середня (необхідно створювати власний хостинг або купувати існуючий)	Середня (платна, для роботи потрібні додаткові мережеві налаштування)
Доступ	Внутрішній додаток / веб додаток	Веб - додаток	Внутрішній додаток
Автоматизація збору даних	Ні	Ні (доступне API для підключення власного агента)	Так

Крім зазначених слід відмітити Spiceworks, Open-AudIT, ManageEngine, OCS Inventory NG.

1.3. Постановка задачі дослідження та вибір засобів реалізації

Варто розділити створення програмного додатку на етапи, для спрощення проектування та подальшої реалізації. На першому етапі потрібно отримати дані про комп'ютер. На другому етапі потрібно підібрати ефективний спосіб зберігання зібраних даних. На третьому етапі потрібно розробити додаток для перегляду, пошуку та обробки зібраних даних.

В межах підприємства де проводилась розробка всі робочі місця мають операційні системи сімейства Windows, тобто збір даних має бути сумісним з різними версіями цієї системи а саме: Windows 7, Windows 8, Windows 10, Windows 11.

Крім того вкрай бажано щоб для збору даних не потрібно було вносити зміни в існуючі мережеві налаштування. Також бажано досягти максимальної

сумісності з вже встановленими системами, тобто, щоб для збору даних не потрібно було встановлювати додаткове програмне забезпечення, оновлення, фреймворки і.т.д.

Враховуючи все вищесказане було прийнято рішення здійснювати збір даних за допомогою скрипту написаного на VBScript (Visual Basic Scripting Edition) – це скриптова мова програмування, розроблена компанією Microsoft на основі Visual Basic. Вона інтерпретується Windows Script Host (WSH) і широко використовується для автоматизації завдань в операційній системі Windows. Ключовою перевагою VBScript є те, що він тісно інтегрований з операційною системою Windows та дозволяє використовувати її функціональність. Тестування на практиці показало, що завдяки простоті та універсальності обраного засобу скрипт успішно відпрацьовує збір даних на усіх наявних системах (навіть на безнадійно застарілих та виведених з домену комп'ютерах з Windows XP) без необхідності інсталяції будь-яких додаткових компонентів.

Другий етап виявився значно складнішим, спроби використання звичайних таблиць для збереження даних хоч і були частково вдалим, продемонстрували, що користуватись ними потім вкрай незручно і непрактично. Після чого було прийнято рішення про збереження даних в реляційній базі даних. Було розроблено діючий прототип на основі Microsoft Access, але незважаючи на працездатність продукту в цілому впродовж кількох місяців використання було виявлено безліч недоліків, котрі частково були спричинені в тому числі й невдало спроектованою структурою бази даних, після аналізу виявлених недоліків було прийняте рішення про використання натомість нереляційної бази даних – MongoDB.

Третій етап – додаток для зручного перегляду даних розроблено на основі VisualBasic.NET з підключенням до бази даних та можливістю пошуку по більшості тих параметрів пошук по котрих міг би бути корисним з нашого погляду.

1.4. Методи оптимізації

Методи оптимізації – це по суті методи прийняття ефективних рішень. «Оптимізація (англ. optimization, optimisation) — процес надання будь-чому найвигідніших характеристик, співвідношень (наприклад, оптимізація виробничих процесів і виробництва, енерговикористання, тощо). Задача оптимізації – задача знаходження точки (точок) екстремуму, або декількох екстремумів заданої функції. Оптимальний (від лат. optimus) — найкращий з можливих варіантів чогось, найвідповідніший певному завданню, умовам. Наприклад, максимальна продуктивність устаткування чи відповідний мінімум витрат по закінченому виробничому циклу. Оптимальні розрізи — значення розмірів чи конструкції технологічного елемента, зумовлені спеціальним розрахунком для досягнення максимального виробничого ефекту, мінімальних трудових чи грошових витрат та ін. критерію.

Критерій оптимальності — основний показник якості роботи системи, фундаментальне поняття системи оптимального функціонування об'єктів (машин, процесів, підприємства, галузі, економіки загалом). В економіці, наприклад, критерієм оптимальності може бути максимум прибутку, мінімум трудових затрат, мінімальний час досягнення мети тощо.

Математичною оптимізацією або математичним програмуванням в математиці, інформатиці та дослідженні операцій називають відбір найкращого елемента (за певним критерієм) з множини доступних альтернатив. У найпростішому випадку задача оптимізації полягає у знаходженні екстремуму (мінімуму або максимуму) дійсної функції шляхом систематичного вибору вхідних значень з дозволеного набору та обчислення значення функції.

Методи оптимізації класифікують відповідно до задач оптимізації. Локальні методи: сходяться до якого-небудь локального екстремуму цільової функції. У разі унімодальної цільової функції, цей екстремум єдиний, і буде глобальним максимумом/мінімумом.

Глобальні методи: мають справу з багатоекстремальними цільовими функціями. При глобальному пошуку основною задачею є виявлення тенденцій глобальної поведінки цільової функції. Існуючі в цей час методи пошуку можна розбити на три великі групи: детерміновані, випадкові (стохастичні) та комбіновані. За критерієм вимірності допустимої множини, методи оптимізації поділяють на методи одновірної оптимізації і методи багатомірної оптимізації. За видом цільової функції й допустимої множини, задачі оптимізації й методи їхнього розв'язання можна розділити на такі класи. Задачі оптимізації, у яких цільова функція $f(x)$ і обмеження є лінійними функціями, розв'язуються так званими методами лінійного програмування. В іншому разі мають справу із задачею нелінійного програмування і застосовують відповідні методи.

За вимогами до гладкості й наявності в цільової функції частинних похідних, їх також можна розділити на: прямі методи, що вимагають тільки обчислень цільової функції в точках наближень; методи першого порядку: вимагають обчислення перших частинних похідних функції; методи другого порядку: вимагають обчислення других частинних похідних.

Крім того, оптимізаційні методи поділяються на такі групи, а саме аналітичні методи, чисельні методи та графічні методи. Розділами математичного програмування є параметричне програмування, динамічне програмування і стохастичне програмування. Математичне програмування використовується при розв'язанні оптимізаційних задач дослідження операцій. Спосіб знаходження екстремуму повністю обумовлюється класом задачі. Але перед тим, як отримати математичну модель, потрібно виконати чотири етапи моделювання:

- визначення меж системи оптимізації (факторний простір);
- вибір змінних проєктування (керованих змінних);
- визначення обмежень на керовані змінні;
- вибір числового критерію оптимізації (цільової функції)»[18].

Для прийняття ефективних рішень потрібно зібрати точну, надійну і адекватну інформацію, базуючись на котрій потім можливо зробити вибір серед можливих варіантів. Можливою є наступна послідовність процесу прийняття рішень: визначення проблеми, вибір критеріїв та обмежень, збір даних, розробка варіантів рішень, моделювання, аналіз варіантів та їх оцінка, порівняння варіантів, прийняття рішення.

Загальну оптимізаційну задачу можна представити у вигляді (1.1) [25]:

$$f(x) \rightarrow \min, \quad x \in X, \quad (1.1)$$

де X – задана множина; $f(x)$ – функція, визначена на X . Потрібно знайти точки мінімуму функції f на множині X . При цьому f називають цільовою функцією, X – допустимою множиною, кожний елемент $x \in X$ – допустимою точкою задачі (1.1).

Розглядаємо лише задачу мінімізації, оскільки задача максимізації еквівалентна аналогічній задачі мінімізації (1). Слід відмітити, що множини розв'язків цих задач збігаються. Точка $x^* \in X$ називається точкою глобального мінімуму функції f на множині X , або глобальним розв'язком задачі мінімізації (1.1), якщо [25]:

$$f(x^*) \leq f(x) \text{ при всіх } x \in X \quad (1.2)$$

Точка $x^* \in X$ називається точкою локального мінімуму функції f на множині X , або локальним розв'язком задачі мінімізації (1.1), якщо існує $\epsilon > 0$, таке, що [25]:

$$f(x^*) \leq f(x) \text{ при всіх } x \in U_\epsilon(x^*), \quad (1.3)$$

де $U_\epsilon(x^*) = \{x \in X : \|x - x^*\| \leq \epsilon\}$ – куля радіуса ϵ з центром у x^* . Якщо нерівності (1.2) або (1.3) виконуються в строгому сенсі, точку x^* називають точкою строгого мінімуму відповідно в глобальному чи локальному сенсі.

Гradientний метод – один з класичних методів мінімізації першого порядку. Він базується на заміні цільової функції f в околі точки чергової точки x^k лінійною частиною розкладу її в ряді Тейлора. У методах спуску послідовність

наближень $x^0, x^1, x^2, \dots, x^k, \dots$ до точки мінімуму вибирається за правилом (1.4) [25]:

$$x^{k+1} = x^k + a_k h^k, a_k > 0, k = 0, 1, 2, \dots, \quad (1.4)$$

де h^k – напрям спадання функції f у точці x^k . У градієнтному методі h^k беруть рівним антиградієнту функції f у точці x^k , тобто $h^k = -f'(x^k)$. Отже, у градієнтному методі (1.5)[25]:

$$x^{k+1} = x^k - a_k f'(x^k), a_k > 0, k = 0, 1, 2, \dots \quad (1.5)$$

Якщо довжину кроку вибирати з умови (одновимірної) мінімізації функції уздовж напрямку антиградієнта, одержуємо варіант градієнтного методу, що називається градієнтним методом найшвидшого спуску. На практиці, зазвичай, доводиться задовольнятися наближеними методами пошуку оптимального значення довжини кроку, наприклад методом дроблення.

Метод Ньютона та його модифікації – один з найефективніших засобів числового розв’язання задач безумовної оптимізації. Припускається, що функція f строго опукла і двічі диференційована на R^n , причому матриця $f''(x)$ не вироджена на R^n . У методі Ньютона послідовність $x^1, x^2, \dots, x^k, \dots$ будують за правилом (1.6) та (1.7) [25]:

$$x^{k+1} = x^k + h^k, \quad (1.6)$$

$$h^k = -[f''(x^k)]^{-1} \cdot f'(x^k), k = 0, 1, \dots \quad (1.7)$$

Отже, метод Ньютона – це метод мінімізації другого порядку. Як видно з (1.6), довжина кроку $a_k = 1$; напрям, що визначається вектором h^k , є напрямом спадання функції f . Квадратична апроксимація заданої функції в малому околі деякої точки значно точніша за лінійну апроксимацію. Тому в методі Ньютона природно сподіватися на більш точне наближення до розв’язку, ніж у градієнтному методі.

У випадку інвентаризації проблемою можемо обрати нестачу інформації про поточне місцезнаходження та загальну реальну кількість активів, в якості критеріїв – точність, актуальність та швидкість отримання зазначеної інформації, обмеження – сегментованість мережі, фінансові обмеження (незважаючи на

актуальність проблематики ймовірність виділення коштів для вирішення даного питання невисока), зібравши дані про функціонування підприємства бачимо що отримання інформації про активи з дотриманням вищезазначених критеріїв неможливе з використанням поточного інструментарію, після чого потрібно знайти рішення, котре б забезпечило вирішення проблеми з дотриманням критеріїв і таким рішенням на наш погляд є автоматизація процесу збору, передачі та аналітики отриманих даних.

Після моделювання того який варіант міг би бути використаний в умовах наявних обмежень мережі підприємства, оцінивши та порівнявши існуючі програмні продукти було прийняте рішення, що найбільш оптимальним варіантом вирішення проблеми буде використання власного продукту, можливості котрого можуть в подальшому бути допрацьовані для вирішення додаткових завдань та ускладнень що неминуче виникатимуть в процесі експлуатації.

Висновки до першого розділу

В даному розділі були проаналізовані та детально розглянуті аналоги програмного забезпечення. Виявлені явні переваги та недоліки наявного функціоналу. Приділено увагу методам оптимізації як засобам прийняття ефективних рішень. Завдяки розглянутій інформації було сформовано орієнтовний план роботи програмного додатку. Було складене технічне завдання зі шляхами реалізації та вимогами до функціонального наповнення програми на кожному з етапів. На основі розглянутих аналогів було підкреслено позитивні аспекти, які варто допрацювати та реалізувати у даному програмному додатку. Головними елементами даного програмного додатку є скрипт для збору інформації, база даних для її збереження та застосунок для пошуку, аналізу та відображення.

РОЗДІЛ 2. РОЗРОБКА ТА ПРОЕКТУВАННЯ ЕЛЕМЕНТІВ СИСТЕМИ ЗБОРУ ДАНИХ

2.1 Розробка скрипту для збору даних

Основним критерієм для вибору засобу збору даних була універсальність та простота запуску, тож перевага віддавалась вбудованим засобам котрі не потребують установки додаткових компонентів і працюють на різних версіях ОС Windows. «Порівняно з пакетними командними файлами сценарії WSH мають ряд переваг. Перш за все, VBScript та JScript – це повноцінні алгоритмічні мови із вбудованими функціями та методами для обробки рядків, виконання математичних операцій, обробки виключень тощо; крім того, може бути використана будь-яка інша мова сценаріїв із встановленими відповідними модулями підтримки (наприклад, Perl, Python, Ruby) для написання сценаріїв WSH. По-друге, WSH підтримує декілька власних об'єктів. Властивості та методи цих об'єктів дозволяють вирішувати деякі типові щоденні завдання адміністраторів операційних систем: використовувати мережеві ресурси, змінні середовища, реєстр, ярлики та спеціальні папки Windows, запускати та керувати іншими програмами. По-третє, зі сценарію WSH можете отримати доступ до служб будь-якого автоматизованого серверного додатка (наприклад, програми Microsoft Office), об'єкти якого зареєстровані в операційній системі. Нарешті, сценарій WSH дозволяє використовувати об'єкти інформаційної моделі Windows Management Instrumentation (WMI), який надає програмний інтерфейс для управління всіма компонентами операційної системи, а також служби каталогів Active Directory Service Interface (ADSI).» [19] Першочергово скрипти Visual Basic Script використовувались для швидкого перегляду інформації про комп'ютер, однак після того як з'явилась ідея про централізований масовий збір даних постало питання – які саме дані нам необхідні, тобто що має записуватись у файл згенерований скриптом? В першій редакції крім мітки часу зберігалось лише доменне ім'я комп'ютера, логін користувача, MAC-адреса пристрою та список програм. Однак виявилось що цієї інформації вкрай недостатньо і

додатково необхідно принаймні модель процесору для ідентифікації пристрою. Тому поступово в результаті тестувань було сформовано список даних котрі можливо з відносним ступенем надійності отримати з більшості комп'ютерів, адже не всі значення завжди є заповненими і численна кількість змінних в класах WMI повертають порожнє або некоректне значення. Вдалося виокремити наступні дані про комп'ютер за допомогою скрипту:

Datetime – поле що містить дату (поточна системна дата й час на момент виконання скрипта). Дата у ньому представлена у форматі ISODATE з явним вказанням типу наступним чином: «"Datetime": { "\$date" : "2024-12-01T05:06:17+02:00"}».

Зроблено це для того, щоб при імпорті файлу в подальшому дату не було імпортовано як текст, оскільки це поле потім активно використовуватиметься у запитах, в тому числі й для виокремлення останніх документів з усього переліку, тож навідміну від решти значень важливо зберегти коректний формат для даного поля. VBScript не має вбудованого методу відображення дати у форматі ISODATE тож створено власну функцію, котра отримуючи на вхід аргумент-дату на його основі конструює й повертає текстовий рядок у необхідному форматі, тож в подальшому, при імпорті даних з файлу, типом даних для даного поля буде визначено «date» замість «string».

Username – поле що містить ім'я користувача під котрим здійснено вхід в систему (наприклад «oleh.velykoborets»). Значення для даного поля отримується за допомогою наступного фрагменту коду:

```
strUser = LCase(CreateObject("WScript.Network").UserName)
```

Тут досить важливим є використання LCase, адже практика показала, що іноді логіни можуть витягатись з великими буквами (наприклад «Oleh.Velykoborets») до того-ж траплялись й випадки коли з різних комп'ютерів логін одного й того-ж користувача зберігався по-різному, допускаємо що це пов'язано з тим яким чином логін було введено при аутентифікації під час входу в систему. Одночасно з тим з точки зору бази даних в операціях сортування та

групування логіни з різним написанням будуть вважатись різними значеннями, і, хоча цю проблему можливо вирішити, це ускладнить запити та сповільнить швидкодію при обробці великих об'ємів даних. Вважаємо що значно простіше привести дані до стандартного вигляду вже на даному етапі уникнувши таким чином подальших ускладнень в цьому питанні;

OS_Caption – поле що містить назву операційної системи (наприклад: «Майкрософт Windows 10 Enterprise»), отримується з класу WMI «Win32_OperatingSystem», властивість «Caption».

OS_Version – поле що містить версію операційної системи (наприклад: «10.0.19045»), отримується з класу WMI «Win32_OperatingSystem», властивість «Version».

OS_Bit – поле що містить розрядність операційної системи (x32/x64), отримується з класу WMI «Win32_OperatingSystem», шляхом зчитування перших двох символів властивості «OSArchitecture» (якщо вони рівні «64» то для поля призначається значення «x64» в усіх інших випадках «x32»).

CDN – поле що містить ім'я ПК, отримується з класу WMI «Win32_NetworkAdapterConfiguration», властивість «DNSHostName».

Мережеві адаптери (зберігаються всі про які є відомості їх поточного статусу, але першими завжди активні підключені адаптери (якщо такі є) потім решта), причому зберігаються у вигляді 4 окремих масивів: AdapterDesc – масив назв адаптерів отримується з класу WMI «Win32_NetworkAdapter», властивість «Description»; AdapterStatus – статус адаптера у текстовому форматі отримується з класу WMI «Win32_NetworkAdapter», властивість «NetConnectionStatus», однак дана властивість повертає лише умовне кодове позначення статусу адаптера у вигляді цифри котра потім, для зручності, конвертується в текстовий еквівалент за наступними правилами:

0 - "Disconnected"

1 - "Connecting"

2 - "Connected"

- 3 - "Disconnecting"
- 4 - "Hardware Not Present"
- 5 - "Hardware Disabled"
- 6 - "Hardware Malfunction"
- 7 - "Media Disconnected"

При тому в першу чергу в усі 4 масиви зберігаються дані з адаптерів з кодом 2 ("Connected"), потім решта. Дані з адаптерів значення статусу котрих відсутнє не зберігаються.

MAC – MAC-адреса адаптера, отримується з класу WMI «Win32_NetworkAdapter», властивість «MACAddress», може бути порожнім для неактивного адаптера, у такому випадку записується порожній рядок в якості значення відповідного елементу масиву.

IP – IP-адреса адаптера, отримується з класу WMI «Win32_NetworkAdapterConfiguration», властивість «IPAddress», і аналогічно MAC-адресі може бути порожнім для неактивного адаптера, у такому випадку записується порожній рядок в якості значення відповідного елементу масиву.

Domain – поле що містить домен або групу ПК (наприклад «WORKGROUP»), отримується з класу WMI «Win32_ComputerSystem», властивість «Domain».

CPUDesc – поле що містить назву центрального процесору (наприклад «Intel(R) Core(TM) i5-10500T CPU @ 2.30GHz»), отримується з класу WMI «Win32_Processor», властивість «Name».

TotalMemory – поле що містить загальний обсяг оперативної пам'яті (в гігабайтах), отримується з класу WMI «Win32_PhysicalMemory», властивість «Capacity» шляхом перебору елементів в циклі та додавання їх значень з вищезазначеної властивості з конвертацією в гігабайти.

RamType – поле що містить тип планок оперативної пам'яті (успішно протестовано визначення наступних типів: DDR2, DDR3, DDR4, DDR5), отримується з класу «MSSmBios_RawSMBiosTables», шляхом об'єднання даних

властивості «SMBiosData» в масив з подальшою його звіркою з заздалегідь визначеними для кожного типу пам'яті шаблонами у вигляді регулярних виразів.

UsedSlots – поле що містить кількість встановлених планок оперативної пам'яті, отримується шляхом підрахунку кількості елементів з класу WMI «Win32_PhysicalMemory».

EmptySlots – поле що містить кількість вільних слотів для планок оперативної пам'яті, отримується шляхом віднімання кількості зайнятих планок (UsedSlots) від загальної кількості слотів, котра в свою чергу отримується з класу WMI «Win32_PhysicalMemoryArray», властивість «MemoryDevices».

RamSlots – масив для наочної візуалізації планок оперативної пам'яті. Розмір масиву відповідає загальній кількості слотів. Дані отримуються з класу WMI «Win32_PhysicalMemory». Елементи містять розмір (властивість «Capacity») та частоту роботи (властивість «Speed») відповідної планки в кожному зі слотів. Кількість елементів в масиві визначається з класу WMI «Win32_PhysicalMemoryArray», властивість «MemoryDevices» тож після перебору зайнятих слотів та запису даних про них в перші елементи масиву решта вільних слотів (якщо такі лишились) заповнюються значенням «Empty» що є індикатором того що в даному слоті планка пам'яті відсутня.

Manufacturer – поле що містить назву виробника материнської плати, отримується з класу WMI «Win32_ComputerSystem», властивість «Manufacturer».

Model – поле що містить назву моделі материнської плати, отримується з класу WMI «Win32_ComputerSystem», властивість «Model».

GPU – поле що містить модель графічного процесору, отримується з класу WMI «Win32_VideoController», властивість «Caption».

GPU_RAM – поле що містить об'єм RAM графічного процесору, отримується з класу WMI «Win32_VideoController», властивість «AdapterRAM», як показує практика, значення може бути відсутнє, або суттєво спотворене, особливо на комп'ютерах з відносно потужними відеокартами де об'єм пам'яті відео адаптера перевищує 2GB однак у випадку офісної техніки (переважна

кількість котрої не обладнана виокремленими відеоадаптерами) параметри зазвичай знаходяться в межах цього ліміту, тож і значення в більшості випадків відображаються коректні. До того ж враховуючи що дане поле не є пріоритетним з точки зору ідентифікації пристрою і носить більше інформаційних характер вважаємо наявність поодиноких похибок у ньому прийнятним.

Printer – поле що містить назву принтера по замовчанню, отримується з класу WMI «Win32_Printer», властивість «Name».

PortName – поле що містить ім'я порту через котрий під'єднано принтер по замовчанню, отримується з класу WMI «Win32_Printer», властивість «PortName».

UpTime – поле що містить час від моменту останнього увімкнення (години), отримується як різниця годин між поточним часом на момент виконання скрипту та значенням властивості «LastBootUpTime» з класу WMI «Win32_OperatingSystem».

Інформація про наявні диски (окрім мережових та CD/DVD дисководів), інформація, аналогічно мережевим адаптерам, розбита по категоріях і зберігається в 7 масивах.

Drive_Models – масив моделей підключених накопичувачів, отримується з класу WMI «Win32_DiskDrive», властивість «Model». Приклад значення елемента масиву: «Samsung SSD 980 PRO 2TB».

Drive_Types – масив типів підключених накопичувачів, наприклад «Fixed hard disk media» або «Removable Media», отримується з класу WMI «Win32_DiskDrive», властивість «MediaType».

Drive_PNPDeviceID – містить унікальний ідентифікатор пристрою в системі, отримується з класу WMI «Win32_DiskDrive», властивість «PNPDeviceID», додатково у зв'язку з наявністю спец-символів що розцінюються в MongoDB як символ екранування (а саме «\») що в подальшому стає на заваді успішного імпорту файлу в базу даних. Для усунення цього використовується «Replace(txt, "\", "\\")» котра екранує даний символ. Приклад значення елемента масиву:

«SCSI\DISK&VEN_NVME&PROD_SAMSUNG_SSD_980\5&20A5412E&0&000000».

Drive_Letters – містить букву що позначає розділ диску, отримується з класу WMI «Win32_LogicalDisk», властивість «name», додатково видаляється останній символ («:») для уникнення конфліктів при імпорті чи створенні запитів в MongoDB, тож елемент просто містить букву, котра позначає розділ (без двокрапки).

Drive_Total – масив що містить загальну ємність розділів (в гігабайтах), отримується з класу WMI «Win32_LogicalDisk», властивість «Size» з подальшим конвертуванням в гігабайти.

Drive_Free – містить об'єм вільного місця в розділі (в гігабайтах), отримується з класу WMI «Win32_LogicalDisk», властивість «FreeSpace» з подальшим конвертуванням в гігабайти.

Drive_Status – масив елементів що позначають поточний стан диску», отримується з класу WMI «Win32_DiskDrive», властивість «status».

Інформація про наявні монітори – дані розбито на 4 категорії котрі зберігаються в масивах (на випадок наявності декількох моніторів у користувача). Monitor_Manufacturer – масив, елементи котрого позначають виробників моніторів, отримується з класу WMI «WmiMonitorID», властивість «ManufacturerName»; Monitor_ProductCode – масив, елементи котрого містять код продукту монітору, отримується з класу WMI «WmiMonitorID», властивість «ProductCodeID»; Monitor_Serial – масив, елементи котрого містять серійний номер, отримується з класу WMI «WmiMonitorID», властивість «SerialNumberID»; Monitor_Name – масив, елементи котрого містять назву моделі монітору, отримується з класу WMI «WmiMonitorID», властивість «UserFriendlyName», зазвичай саме в цьому полі містяться моделі моніторів, що використовуються, і загалом ним одним можливо було б і обмежитись, однак, враховуючи те, що для певних моделей це поле може бути не заповненим збираються й інші дані щоб мати хоч якусь інформацію (у багатьох випадках

монітори на підприємствах закупаються партіями тож в окремих випадках навіть просто інформації про назву фірми-виробника монітору може бути достатньо для орієнтовної ідентифікації пристрою).

Software – масив з назвами програм що є встановленими на комп'ютері відповідно до даних з реєстру, котрі отримуються з «Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall», також для x64 додатково з «Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall».

Links – масив з назвами ярликів (розширення *.LNK) котрі знаходяться на робочому столі або у папках на робочому столі (функція відбору заснована на порівнянні розширення файлу, працює рекурсивно тож охоплює як робочий стіл так і всі вкладені папки та підпапки).

Link_Paths – масив з шляхами ярликів, кожен елемент котрого відповідає елементу масиву Links і зберігає шлях куди веде кожен з ярликів. Основне призначення даної пари масивів – спроба відслідкувати наявність портативних програм на комп'ютері, отримується при пошуку ярликів за допомогою функції ExtractTargetPath().

URLs – масив з назвами закладок (розширення *.URL) котрі знаходяться на робочому столі або у папках на робочому столі (функція відбору заснована на порівнянні розширення файлу, працює рекурсивно тож охоплює як робочий стіл так і всі вкладені папки та підпапки).

URL_Paths – масив з шляхами закладок, кожен елемент котрого відповідає елементу масиву URLs і зберігає веб-посилання куди веде кожен з закладок, отримується при пошуку закладок за допомогою функції ExtractTargetPath().

EXEs – масив з назвами застосунків (розширення *.EXE) котрі знаходяться на робочому столі або у папках/підпапках на робочому столі (функція відбору заснована на порівнянні розширення файлу, працює рекурсивно тож охоплює як робочий стіл так і всі вкладені папки та підпапки).

Так як зберігання даних відбувається в базі даних MongoDB збереження даних відбувається в форматі UTF-8 в .json файл з відповідною структурою для того, щоб кожен згенерований файл з інформацією про комп'ютер міг без проміжкових редагувань та виправлень одразу бути імпортованим як повноцінний документ в базу даних. JSON (JavaScript Object Notation) – це легкий формат обміну даними, який використовується для зберігання та передачі структурованої інформації. Він є текстовим форматом, що легко читається людьми та зручно обробляється комп'ютерами. Основним його призначенням є зберігання і передачу даних між сервером та клієнтом в веб-розробці, а також між різними додатками.

JSON часто використовують у веб-програмуванні для комунікації між серверами та веб-браузерами, оскільки він є універсальним і підтримується більшістю мов програмування.

Основні характеристики JSON: текстовий формат — його можна прочитати та змінити вручну, адже це простий текст; легкий для парсингу — дані в JSON можна швидко обробляти як програмно, так і вручну; мова незалежна — JSON є мовно-нейтральним, тобто його можна використовувати з будь-якими мовами програмування, що підтримують роботу з текстовими даними (наприклад, Python, JavaScript, Java, PHP тощо).

Структура JSON. JSON складається з двох основних типів структур. Об'єкти – набір пар "ключ-значення". Об'єкт у JSON записується в фігурних дужках {}. Масиви – впорядковані набори значень, які записуються в квадратних дужках [].

Головними перевагами використання JSON є. Простота – зрозумілий і компактний формат, який легко зчитується і записується що спрощує процес розробки і дозволяє безпосередньо переглядати згенеровані скриптом файли. Універсальність – працює з багатьма мовами програмування. Легкість у використанні для обміну даними через мережу – JSON є популярним для передачі даних між серверами та клієнтами.

JSON став стандартом де-факто для обміну даними в сучасних веб-додатках завдяки своїй простоті, зручності та ефективності.

Після збору всіх даних відбувається запис файлу в папку відповідно до налаштувань заданих константою LocalCreate – у випадку якщо константа дорівнює «1» файл з даними буде створено в тому ж розташуванні звідки запущено файл скрипту, у випадку якщо ж її значення дорівнюватиме 0 файл з даними буде створено в папці «C:\Users\[ім'я поточного користувача]\AppData\Roaming\System» (у випадку відсутності папки «System» її там буде створено), після чого (за умови що константа isNet матиме значення False та операційна система не Windows XP) відбуватимуться спроби копіювання створеного файлу в мережеву папку до шляху вказаного в константі NetPath, до тих пір поки файл там не з'явиться з інтервалом 60 секунд.

Цей підхід переслідує дві цілі – по перше при тимчасовій втраті мережевого підключення файл буде записано після відновлення підключення, по-друге файл може не записуватись по причині відсутності вільного місця на диску. В такому випадку файл буде записано щойно звільниться місце. Таким чином комп'ютер на котрому буде запущено даний скрипт може лишити файл з інформацією про себе в спеціально для цього відведений папці на загальнодоступному мережевому ресурсі.

Для автоматизації процесу достатнім буде додати даний скрипт в автозапуск що дозволить створювати файл з інформацією про комп'ютер при кожному його включенні (назва файлу складається з дати (з видаленим часом, лише день, місяць, рік), імені користувача, імені комп'ютера), тож повторні запуски того-ж комп'ютера тим же користувачем впродовж дня перезаписуватимуть файл більш актуальним. Додання в автозапуск може бути здійснене як вручну – що є досить трудомістким та небажаним процесом так і за допомогою вказання цього скрипту як скрипту, що виконується при запуску комп'ютера за допомогою групових політик домену через Group Policy Management Console.

Group Policy Management Console (GPMC) – це інструмент адміністратора в операційних системах Windows, який використовується для створення, редагування та управління груповими політиками (Group Policies) в доменах Active Directory.

Active Directory (AD) – це служба каталогів, розроблена Microsoft, яка використовується для управління ресурсами мережі, такими як користувачі, комп'ютери, принтери, групи та інші пристрої в корпоративних або доменних мережах. Вона є основним компонентом операційних систем Windows Server, що дозволяє централізовано керувати всіма аспектами мережі в організації. Основні компоненти Active Directory, зокрема контролер домену (Domain Controller), каталог Active Directory, об'єкти Active Directory, групи Active Directory, політики груп.

Контролер домену (Domain Controller) – це сервер, який зберігає всі дані про користувачів, комп'ютери та інші об'єкти в Active Directory. Він відповідає за автентифікацію та авторизацію користувачів і комп'ютерів. Коли користувач намагається увійти в систему або отримати доступ до ресурсів, контролер домену перевіряє його права доступу.

Каталог Active Directory – це база даних, що містить інформацію про всі об'єкти в домені, включаючи користувачів, групи, комп'ютери, принтери, політики безпеки та інші ресурси. Зазвичай він зберігається у файлі NTDS.dit.

Об'єкти Active Directory це одиниці, які зберігаються в каталозі AD. Об'єкти можуть бути користувачами, комп'ютерами, групами, принтерами, політиками і т. д. Наприклад, користувач є об'єктом, і на кожного користувача зберігається інформація про його ім'я, пароль, налаштування доступу тощо.

Групи Active Directory групи використовуються для спрощення управління доступом до ресурсів. Замість того, щоб призначати дозволи кожному користувачеві окремо, адміністратор може призначити дозволи групі, а потім додавати користувачів до цієї групи.

Політики груп (Group Policy) політики груп дозволяють централізовано налаштовувати параметри безпеки, програм, доступу та інших налаштувань для користувачів і комп'ютерів в домені. Адміністратори можуть використовувати політики для автоматизації налаштувань і збереження безпеки в організації.

Реплікація у великих мережах можуть бути кілька контролерів домену, і всі вони повинні мати однакові дані для забезпечення коректної роботи. Реплікація — це процес синхронізації даних між контролерами домену, щоб всі вони мали актуальну інформацію.

Основні функції Active Directory. Централізоване управління користувачами та комп'ютерами. Усі користувачі і комп'ютери в домені мають єдину облікову запису, що спрощує адміністрування мережі. Наприклад, адміністратор може змінювати паролі або права доступу для всіх користувачів в організації з одного місця. Аутентифікація та авторизація. Active Directory відповідає за аутентифікацію (перевірку, чи є у користувача право на доступ) та авторизацію (визначення, що користувач може робити після входу в систему). Управління доступом до ресурсів за допомогою груп і політик безпеки адміністратори можуть визначати, хто має доступ до певних файлів, принтерів, мережевих ресурсів тощо.

Active Directory дозволяє створювати ієрархічну структуру, де домени можуть бути об'єднані в доменні дерева (tree) та ліси (forest) для організації мережі, яка складається з кількох доменів. Ліс (Forest) це найбільша одиниця в AD. Ліс може містити кілька доменів, які об'єднані спільною схемою та глобальним каталогом. Всі домени в лісі мають спільний каталог і можуть взаємодіяти один з одним. Дерево доменів (Domain Tree) це ієрархія доменів, де один домен є батьківським, а інші домени — дочірніми. У кожному дереві є принаймні один домен, але може бути кілька. Домен (Domain) основна логічна одиниця в Active Directory, яка містить набір комп'ютерів, користувачів та інших ресурсів, що належать до однієї організації або частини організації. Всі ресурси в межах домену мають спільну політику безпеки та централізоване управління

через контролер домену. Кожен домен має власну базу даних і контролер домену. Організаційні підрозділи (OU) використовуються для групування об'єктів в домені. Це дозволяє організувати користувачів і комп'ютери за різними критеріями (наприклад, по відділах, по географічним локаціям тощо). У кожному OU можна застосовувати окремі групові політики.

Принцип роботи Active Directory. Створення користувача - адміністратор створює обліковий запис користувача в домені через Active Directory Users and Computers. Користувач отримує доступ до ресурсів у межах домену після автентифікації. Автентифікація користувача коли користувач намагається увійти в комп'ютер, він вводить свої облікові дані. Контролер домену перевіряє ці дані і надає доступ, якщо інформація вірна. Це забезпечує єдину точку входу в систему для всіх користувачів у домені. Ієрархія доступу за допомогою груп адміністратори визначають, хто і до яких ресурсів має доступ. Наприклад, група "Адміністратори" може мати доступ до всіх ресурсів, а група "Користувачі" — лише до деяких.

Перевагами Active Directory є. Централізоване управління котре забезпечує ефективне адміністрування великої кількості користувачів і комп'ютерів. Забезпечення безпеки завдяки можливості налаштовувати доступ до ресурсів та редагувати права користувачів. Масштабованість – Active Directory може обслуговувати від невеликих до дуже великих організацій з кількома тисячами користувачів і пристроїв.

Групові політики – це набір налаштувань, які можуть бути застосовані до комп'ютерів і користувачів в доменній мережі для централізованого управління конфігурацією та поведінкою системи. За допомогою групових політик адміністратори можуть: налаштовувати безпеку, керувати програмами та додатками, контролювати доступ до певних функцій і ресурсів, автоматизувати виконання скриптів при вході в систему чи на старті.

GPMS є основним інструментом для адміністраторів для роботи з груповими політиками в Active Directory. Він дозволяє адмініструвати всі

аспекти групових політик у домені, включаючи: створення та редагування GPO (Group Policy Objects) – об'єктів групової політики, які містять налаштування для користувачів і комп'ютерів; використання та делегування прав – адміністратори можуть делегувати права на управління політиками іншим користувачам чи групам; перевірка застосування політик – можливість переглядати, чи застосовуються політики на певних комп'ютерах або користувачах, а також діагностувати проблеми; підключення до кількох доменів – GPMC дозволяє управляти політиками в межах кількох доменів у лісах Active Directory.

Основні функції GPMC. Управління груповими політиками: створення нових групових політик (GPO). Застосування політик до різних організаційних підрозділів (OU) в Active Directory. Копіювання і переміщення політик між різними доменами чи підрозділами.

Налаштування політик. Можливість редагувати налаштування політик для користувачів та комп'ютерів, такі як безпека, налаштування програмного забезпечення, профілі користувачів тощо.

Аналіз політик. Інструменти для перевірки, чи застосовуються групові політики на певних машинах або для конкретних користувачів. Можливість діагностики, якщо політики не застосовуються належним чином.

Діагностика та відстеження. Вбудовані інструменти для перевірки застосування групових політик (наприклад, Resultant Set of Policy або gpresult), що дозволяє адміністратору побачити, які політики діють на конкретному комп'ютері або для конкретного користувача.

Делегування прав. Можливість делегувати права управління політиками іншими адміністраторами або певними групами користувачів, щоб забезпечити більш гнучке управління.

Для додавання скрипту за допомогою групових політик (Group Policy) в автозапуск на доменних ПК, користуємось груповою політикою, щоб автоматично запускати скрипт при вході користувачів або при старті системи.

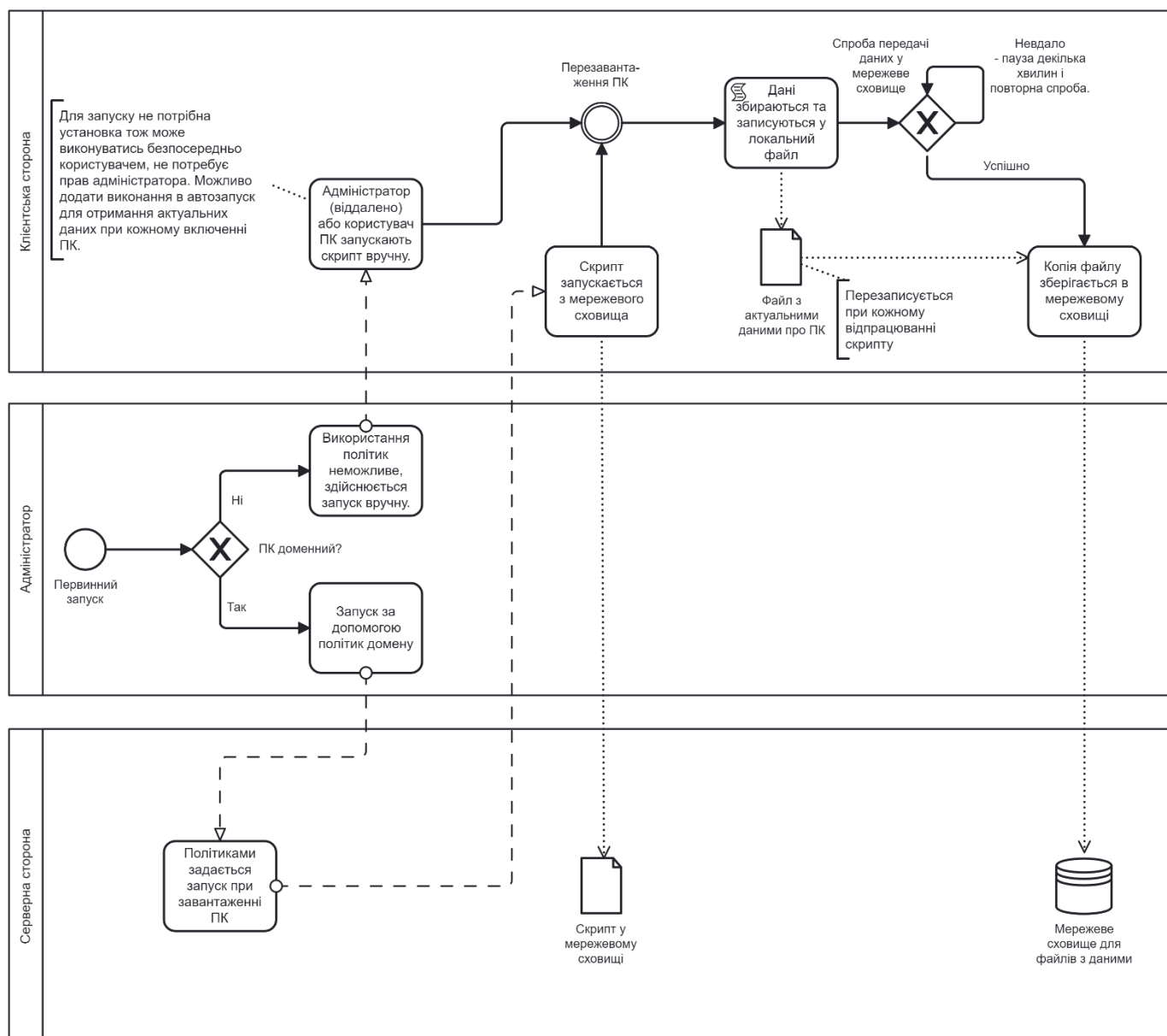


Рис. 2.1. BPMN діаграма послідовності роботи скрипту

Перше, що потрібно зробити – зберегти скрипт на сервері, доступному для всіх користувачів домену. Це може бути загальна папка. Далі на сервері або комп'ютері з адміністративними правами відкриваємо Редактор групових політик (Group Policy Management Console) «Пуск → Адміністративні інструменти → Управління груповими політиками (Group Policy Management)». Після того у Group Policy Management Console в контекстному меню домену або OU обираємо «Create a GPO in this domain, and Link it here». Після створення в контекстному меню групової політики обираємо «Edit», переходимо до User Configuration, Для

запуску при вході користувача обираємо User Configuration → Policies → Windows Settings → Scripts (Logon/Logoff). Нас цікавить спрацювання при вході в систему, обираємо Logon → Add... → Browse та вказуємо шлях до скрипту в мережевій папці. Для застосування нової політики може знадобитись певний час, зазвичай до двох годин.

Система була розроблена саме з розрахунку на використання можливостей запуску за допомогою групових політик, що дозволяє у відносно стислі строки розповсюдити завдання запуску скрипту на доменних комп'ютерах підприємства, ефективно автоматизуючи процес збору інформації про комп'ютери на підприємстві. Однак, якщо з певних причин частина комп'ютерів підприємства не знаходиться в домені або занадто застарілі для використання сучасних SMB протоколів з метою доступу до мережових дисків (Windows XP) то скрипт можливо додати до автозапуску вручну.

2.2 Розробка бази даних

«З розвитком IT індустрії зростає потреба в нових підходах до вирішення проблеми зберігання та управління великим об'ємом даних. Сьогодні найбільш популярними базами даних є реляційні SQL бази даних, які з'явилися ще у 70-х роках минулого століття. Однак зі стрімким зростанням обсягів зберігання даних, які є погано структурованими, висуваються підвищені вимоги до швидкості пошуку та продуктивності обробки запитів до бази даних. Якщо цілісність даних не відіграє великого значення, а першочерговими характеристиками є гнучкість та швидкодія, то доречніше використовувати нереляційні (NoSQL) бази даних».[21] Для збереження та обробки зібраних даних було обрано MongoDB. «MongoDB – це популярна NoSQL база даних, орієнтована на документи, яка дозволяє зберігати структуровані дані у форматі JSON-подібних документів. Ключові переваги використання віддаленої бази даних MongoDB полягають у наступному:

- масштабованість
- висока продуктивність
- гнучкість схеми даних
- підтримка реплікації та шардингу
- багатий набір інструментів для розробників.

Масштабованість MongoDB дозволяє легко розширювати базу даних відповідно до зростаючих вимог додатків та збільшення обсягів даних. Висока продуктивність досягається завдяки ефективній індексації та кешуванню, що забезпечує швидкий доступ до даних та обробку запитів. Гнучка схема даних надає свободу у структуруванні та зберіганні різнорідних даних без необхідності жорстко визначати схему заздалегідь. Підтримка реплікації та шардингу забезпечує відмовостійкість та розподіл навантаження між декількома вузлами, відповідно. Також, MongoDB пропонує багатий набір інструментів для розробників, включаючи драйвери для різних мов програмування, візуальні інструменти для адміністрування та моніторингу. Використання віддаленої MongoDB як інструменту для збереження даних на сервері дозволяє розробникам зосередитися на основній логіці додатків, не витрачаючи зусиль на управління власною базою даних».[22]

«MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій».[23] Документи можуть містити різні типи даних, включаючи вкладені документи та масиви, що забезпечує гнучкість у моделюванні даних. NoSQL: MongoDB не використовує SQL (Structured Query Language) для запитів. Замість цього вона використовує власний API запитів, що дозволяє ефективно працювати з документоорієнтованими даними.

Гнучка схема: На відміну від реляційних баз даних, MongoDB не вимагає попереднього визначення схеми таблиць. Це дозволяє легко змінювати структуру даних з часом без необхідності міграції даних. Масштабованість: MongoDB розроблена для горизонтального масштабування, що дозволяє розподіляти дані між кількома серверами (шардування) та обробляти великі обсяги даних та високе навантаження.

Висока продуктивність: Завдяки документоорієнтованій моделі та ефективним індексам, MongoDB забезпечує високу швидкість читання та запису даних. Підтримка індексів: MongoDB підтримує різні типи індексів, включаючи індекси за одним полем, складені індекси, текстові індекси та геопросторові індекси, що дозволяє оптимізувати виконання запитів. Агрегація: MongoDB надає потужні можливості для агрегації даних, що дозволяє виконувати складні аналітичні запити.

Вибір нереляційної бази даних ефективно спростив задачу її проектування до мінімуму: фактично те що ми маємо на виході скрипту – це повноцінний документ, тож більшою мірою вибір MongoDB повпливав саме на створення скрипту, адже для того, щоб документ можна було імпортувати в базу даних MongoDB дані всередині документу мають бути збережені з дотриманням структури .json документу. Крім того на практиці виявилось, що простого дотримання структури недостатньо, адже збережений за допомогою «object.CreateTextFile(filename[, overwrite[, unicode]])» .json документ не вдається імпортувати навіть незважаючи на те, що третій аргумент було виставлено «true». Доречі використання unicode виявилось, насправді, обов'язковим, адже деякі назви програмних продуктів можуть містити ієрогліфи, тож якщо під час запису файлу не використовувати кодування unicode, файл не буде успішно створеним, але «object.CreateTextFile(filename[, overwrite[, unicode]])» при вказанні аргументу «Unicode» як «true» зберігає файл у кодуванні UTF-16, що є несумісним з MongoDB. Рішенням цієї проблеми стало натомість використання

при збереженні файлу об'єкту «ADODB.Stream», що дозволяє явно вказати потрібний тип кодування (в нашому випадку це UTF-8).

Також використовується MongoDB Tools – це набір утиліт командного рядка, що надаються разом з MongoDB, для виконання різних адміністративних та операційних завдань з базами даних MongoDB. Вони дозволяють імпортувати та експортувати дані, створювати резервні копії, відновлювати дані, виконувати запити та багато іншого. Власне, було використано `mongoimport` - інструмент для імпорту даних у MongoDB з різних форматів, таких як JSON, CSV та TSV.

MongoDB Tools встановлюються разом з сервером MongoDB. Після встановлення MongoDB, шляхи до цих інструментів зазвичай додаються до системної змінної PATH, що дозволяє запускати їх з командного рядка з будь-якої директорії.

Для використання MongoDB Tools (`mongoimport`) необхідно відкрити командний рядок (термінал) та ввести команду з потрібними параметрами:

```
for %f in ("%path%\*.json") do mongoimport "%f" -d inv -c SoftInfo
```

Дана команда циклом перебирає усі файли з розширенням *.json у вказаній теці та імпортує кожен з них у задані базу даних та колекцію. Передбачається, що завантажені файли будуть міститись в одній колекції з якою ми далі й взаємодіємо за допомогою відповідного додатку.

2.3 Розробка додатку на VB.NET

VB.NET (Visual Basic .NET) – це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft на платформі .NET Framework. Вона є еволюцією класичної мови Visual Basic, але з суттєвими змінами, що зробили її повноцінною мовою .NET.

.NET Framework: VB.NET повністю інтегрована з платформою .NET Framework, що надає доступ до великої кількості бібліотек та функціональності. Об'єктно-орієнтована: VB.NET підтримує всі основні принципи об'єктно-орієнтованого програмування: інкапсуляцію, спадкування, поліморфізм. «Код

компілюється в збірку загальною мовою CIL (Common Intermediate Language), яка далі зберігається у файлі формату PE (Portable Executable). Ця збірка компілюється у байт-код та за допомогою технології JIT (Just-in-time compilation), транслюється у машинний код під час роботи програми. Саме через CIL стає можливою розробка різних модулів програмного забезпечення різними мовами.» [24] Visual Studio: VB.NET тісно інтегрована з інтегрованим середовищем розробки Visual Studio, що надає потужні інструменти для розробки, відлагодження та тестування.

VB.NET дозволяє створювати класичні Windows-додатки з графічним інтерфейсом користувача котрий і було використано в даній роботі. Загалом функціонал додатку більшою мірою зводиться до підключення до бази даних, генерування відповідних запитів на основі дій користувача та відображення результатів цих запитів. Однією з основних проблем, що постає на даному етапі – виокремлення тих комп'ютерів інформацію про які потрібно відобразити з загального переліку, уникаючи при тому відображення неактуальних, застарілих документів. На цьому етапі складно визначити єдине вірне рішення котре працювало б в усіх ситуаціях, адже основною проблемою є те, що насправді жоден з тих параметрів котрі збираються за допомогою скрипту VBScript не здатен гарантовано ідентифікувати комп'ютер.

Так користувач може працювати одразу за декількома комп'ютерами - наприклад користувач може експлуатувати стаціонарний комп'ютер і ноутбук, або замінити іншого працівника тимчасово виконуючи роботу за його робочим місцем, також диспетчери аварійно-диспетчерської служби котрі позмінно цілодобово чергують в очікуванні звернень мають один комп'ютер, але кожен заходить в нього під своїм обліковим записом. Так один обліковий може використовуватись на декількох комп'ютерах, і на кожному комп'ютері може здійснювати вхід декілька різних працівників.

Доменні імена комп'ютерів зазвичай рідко змінюються, однак це також трапляється тож цілком може статись ситуація коли в різні проміжки часу 2

різних комп'ютери носили одне й те саме ім'я. Більш постійним може бути ідентифікатор на кшталт MAC адреси – вона не гарантовано унікальна, однак її повторення малоймовірні, крім того, якщо не вийде з ладу мережевий адаптер адреса може лишатись незмінною роками, хоча також помилково буде стверджувати, що MAC адреса здатна однозначно ідентифікувати пристрій. Так якщо використовується ноутбук то MAC-адреса буде змінюватись вже принаймні від того чи є на даний момент активним Lan-адаптер чи Wi-Fi-адаптер. Відповідно в такому випадку якщо ми обиратимемо найновіший документ спираючись на MAC адресу то в результуючий звіт ноутбук може потрапити двічі під різними адресами.

Узагальнюючи вищесказане можемо підсумувати проблематику: залежно від того по якому принципу обираються найактуальніші документи ми ризикуємо або «задвоїти» пристрій якщо у нього є записи в базі даних з різними значеннями того параметру по котрому відбувається відбір, або помилково відкинути пристрій по тій причині, що параметр, по котрому проводився пошук, співпав з аналогічним параметром іншого пристрою тож всі інші з меншою датою були відкинуті як застарілі.

Нажаль, виокремити унікальний, незмінний ідентифікатор, котрий, до того ж, можливо було б зчитати за допомогою VBScript для, по суті, модульної системи, кожний компонент котрої, теоретично, може бути замінено, не вдалося. Тобто повністю уникнути задвоєнь практично неможливо, хоча, залежно від мети запиту можливо сформувані його таким чином, щоб мінімізувати їх вплив. Наприклад, допустимо що мережа має пристрої з декількох різних доменів і хочемо порахувати загальну кількість доменів, якщо відбір сформуємо на основі найновішого документу для кожного доменного імені комп'ютера, то ризикуємо витягнути застарілий запис про певний комп'ютер котрий було переведено в інший домен і відповідно перейменовано для відповідності правилам нового домену.

З точки зору доменного імені пристрій до перейменування та пристрій після перейменування це 2 різних пристрої, однак з точки зору MAC-адреси пристрій може бути ідентифіковано як один і той самий, що буде більш коректним. Але не завжди метою є виокремлення унікального комп'ютера, так з точки зору пошуку програмного продукту бажано бачити інформацію про кожен комп'ютер в розрізі кожного з користувачів. Тобто подібні «зادвоєння» можуть бути навіть бажаними залежно від контексту запиту.

З другої сторони проблема помилкового відкидання вирішується шляхом комбінування умов пошуку зі списку параметрів. Переважна більшість запитів оформлена у вигляді Aggregation Pipeline (конвеєр агрегації). В MongoDB – це фреймворк для обробки даних, що дозволяє перетворювати та агрегувати дані з колекції у послідовність етапів (stages). Кожен етап конвеєра приймає на вхід документи, обробляє їх і передає результат наступному етапу. Це дозволяє виконувати складні операції агрегації, такі як фільтрація, групування, сортування, проектування, перетворення та збагачення даних безпосередньо на сервері бази даних.

Основні концепції. Документи: Конвеєр агрегації працює з документами BSON (Binary JSON), які є основним форматом зберігання даних в MongoDB. Етапи (Stages): Етапи – це будівельні блоки конвеєра. Кожен етап виконує певну операцію над документами. Конвеєр (Pipeline): Конвеєр – це послідовність етапів, через яку проходять документи. Результат попереднього етапу передається на вхід наступному.

Основні етапи конвеєра агрегації: \$match: Фільтрує документи за заданими критеріями. Аналогічний оператору find(). \$project: Вибирає, додає або перейменовує поля в документах. Дозволяє створювати нові обчислювані поля. \$group: Групує документи за заданим полем та застосовує агрегатні функції (наприклад, \$sum, \$avg, \$min, \$max, \$count). \$sort: Сортує документи за одним або кількома полями. \$limit: Обмежує кількість документів, що повертаються. \$skip: Пропускає задану кількість документів. \$unwind: Розгортає масив у окремі

документи для кожного елемента масиву. \$lookup: Виконує операцію з'єднання (join) з іншою колекцією.

\$out: Записує результати агрегації в нову колекцію. \$merge: Записує результати агрегації в існуючу колекцію, виконуючи злиття (merge) з існуючими документами. \$addFields: Додає нові поля до документів. \$replaceRoot: Замінює корінь документа вказаним полем. \$count: Повертає кількість документів, що відповідають умовам.

Переваги конвеєра агрегації. Продуктивність: Агрегація виконується на сервері бази даних, що зменшує обсяг даних, що передаються по мережі. Гнучкість: Конвеєр дозволяє виконувати складні операції обробки даних за допомогою комбінації різних етапів. Виразність: Синтаксис конвеєра є досить зрозумілим та дозволяє легко описувати логіку обробки даних.

Конвеєр агрегації широко використовується для: аналітичних звітів; обробки великих обсягів даних; створення агрегованих представлень даних; реалізації складних бізнес-логік.

Конвеєр агрегації є потужним інструментом для обробки даних в MongoDB, що дозволяє виконувати складні аналітичні операції безпосередньо на сервері бази даних. Він забезпечує високу продуктивність, гнучкість та виразність, що робить його незамінним для багатьох завдань.

В загальному пошуку використовується конвеєр агрегації наступного вигляду (див. рис.2.2). Тобто спочатку колекція сортується за датою потім групується за комбінацією імені користувача, доменного імені комп'ютера та першого елемента з масиву MAC адрес і з кожної групи відбирається перший (він же найновіший) документ. Усі базові пошуки відбуваються саме в межах ось цієї групи відібраних найновіших документів.

На даний момент всі документи, окрім найновіших, котрі збережені в базі даних не виконують якихось функцій будучи відсіяними в переважній більшості запитів, однак це не означає що вони не потрібні.

:

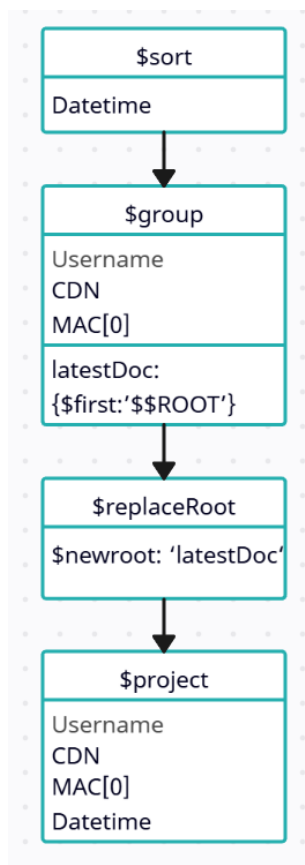


Рис. 2.2. Умовна схема основного конвеєру агрегації

В процесі подальшого доопрацювання додатку планується використовувати дані попередніх входів для конструювання запитів, котрі дозволять переглядати історію логінів користувача (на яких комп'ютерах і коли здійснювався вхід) потенційно дозволяючи виявити та звернути увагу на користувачів чи техніку які довгий час неактивні з тієї чи іншої причини, відслідковувати рух техніки за рахунок генерування наочної історії коли й хто проходив аутентифікацію на певному комп'ютері та коли й які зміни відбулись, в тому числі в програмному забезпеченні. Також це дозволить реалізувати пошук станом на певну дату.

Тож збереження детальних даних минулих періодів потенційно також може принести багато користі.

Висновки до другого розділу

Детально розглянуто принцип роботи кожного сегменту програмного комплексу, послідовність збору, збереження та обробки даних, перелічено ключові проблематику та труднощі, що виникали при розробці разом зі шляхами їх усунення, особливостями реалізації окремих компонентів. Означено перспективи подальшого розвитку програмного комплексу.

РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

3.1 Порядок запуску і роботи

Для роботи додатку необхідне підключення до бази MongoDB з наявними в ній базою даних «inv» та колекцією «SoftInfo». На даний момент застосунок розроблено з розрахунку на локальну роботу тож він за замовчанням підключається до локального серверу mongod.exe. При відкритті програма перевіряє чи запущено процес mongod. Відсутність даного процесу в списку активних процесів свідчить про те що сервер mongod.exe не запущено і для роботи програми його потрібно запустити. Тож якщо сервер mongod не запущений, при відкритті програми відбудеться спроба його запуску. В програмному коді зазначено наступні типові шляхи встановлення mongod з яких програма намагатиметься його запустити:

"C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe",

"C:\Program Files\MongoDB\Server\7.0\bin\mongod.exe",

"C:\Program Files\MongoDB\Server\6.0\bin\mongod.exe",

"C:\Program Files\MongoDB\Server\5.0\bin\mongod.exe",

"C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe",

"C:\Program Files\MongoDB\bin\mongod.exe",

"C:\mongodb\bin\mongod.exe",

"C:\Mongo\Mongod\bin\mongod.exe"

У випадку якщо mongod.exe не буде виявлено за жодним з цих розташувань користувачу буде запропоновано вказати шлях до mongod.exe самостійно з метою запуску серверу. Вікно вибору шляху до серверу зображено на рис. 3.1. Якщо сервер запущений відбудуться спроби підключитись до бази даних з налаштуваннями за замовчанням, а саме:

шлях підключення клієнта mongodb://localhost:27017;

база даних за замовчанням – inv;

назва колекції SoftInfo;

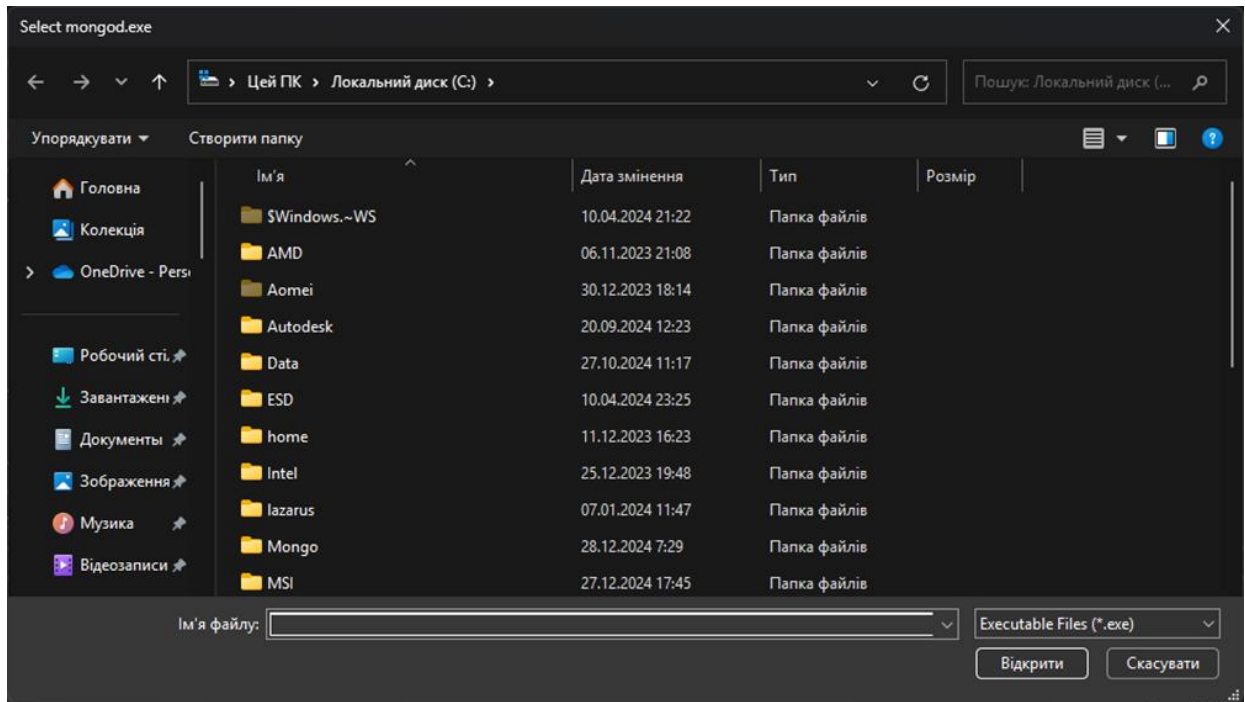


Рис. 3.1. Запит шляху до mongod.exe

За умови успішного підключення відкриється головне вікно програми в лівій частині якого буде сформовано список пристроїв (відбирається «найсвіжіший» документ (з найбільшим значенням Datetime) для кожної унікальної комбінації MAC адреси, імені користувача та імені комп'ютера, що практично мінімізує імовірність помилкового відкидання документу).

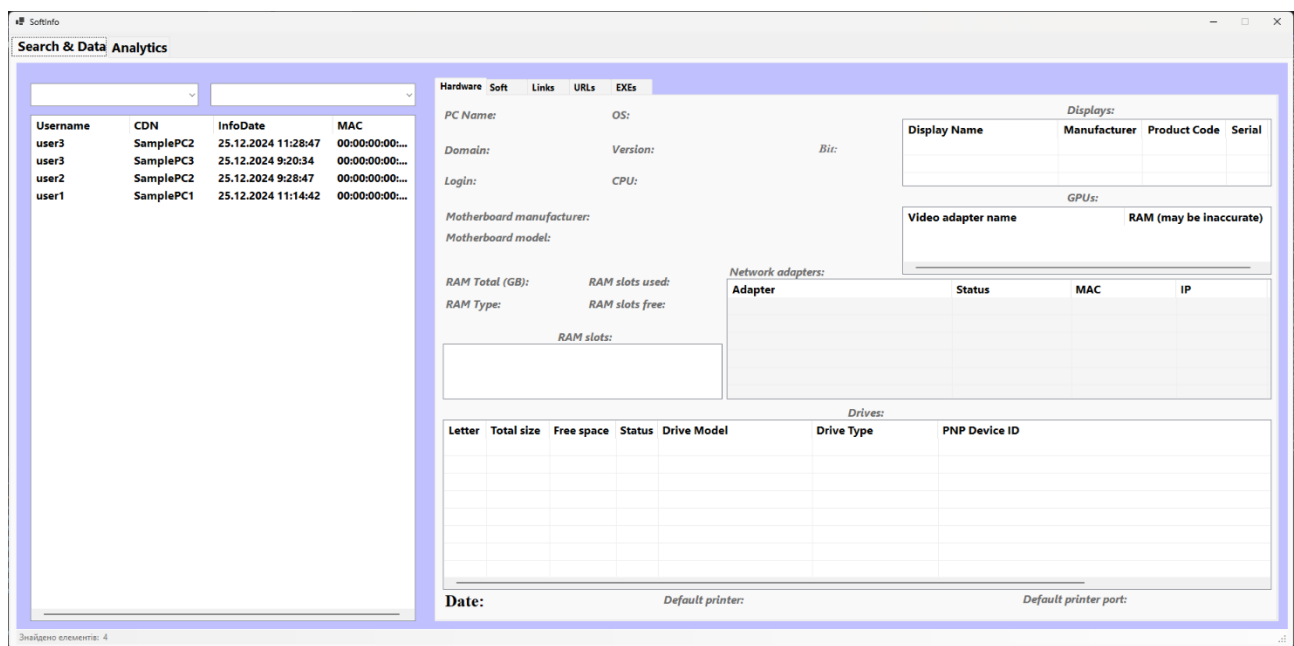


Рис. 3.2. Загальний вигляд застосунку

Для кожного елементу відображені ім'я користувача, ім'я ПК, дата збереження даних, MAC-адреса (для можливості розрізнення ПК на випадок якщо обидва імені співпадатимуть).

3.2 Робота у вкладці «Search & Data»

Вкладка «Search & Data» є активною при відкритті програми і призначена для пошуку та відображення даних по обраних параметрах. У верхній лівій частині міститься два поля з випадаючими списками, призначені для пошуку. В першому полі можливо обрати тип поля по котрому відбуватиметься пошук з наступного переліку (див. рис.3.3):

- Username – ім'я користувача;
- OS_Caption – назва операційної системи;
- OS_Version – версія операційної системи;
- OS_Bit – розрядність операційної системи;
- CDN – доменне ім'я комп'ютера;
- MAC – Mac-адреса комп'ютера;
- IP – IP-адреса комп'ютера;
- Domain – домен в котрому знаходиться комп'ютер;
- CPUDesc – назва процесора комп'ютера;
- TotalMemory – сумарний об'єм оперативної пам'яті (в гігабайтах);
- RamType – тип оперативної пам'яті;
- Manufacturer – виробник материнської плати;
- Model – модель материнської плати;
- Drive_Models – модель накопичувача;
- Drive_Status – статус накопичувача;
- Monitor_Manufacturer – виробник дисплея;
- Monitor_Name – назва моделі монітору;
- Monitor_ProductCode – код продукту монітору;
- Software – програмне забезпечення

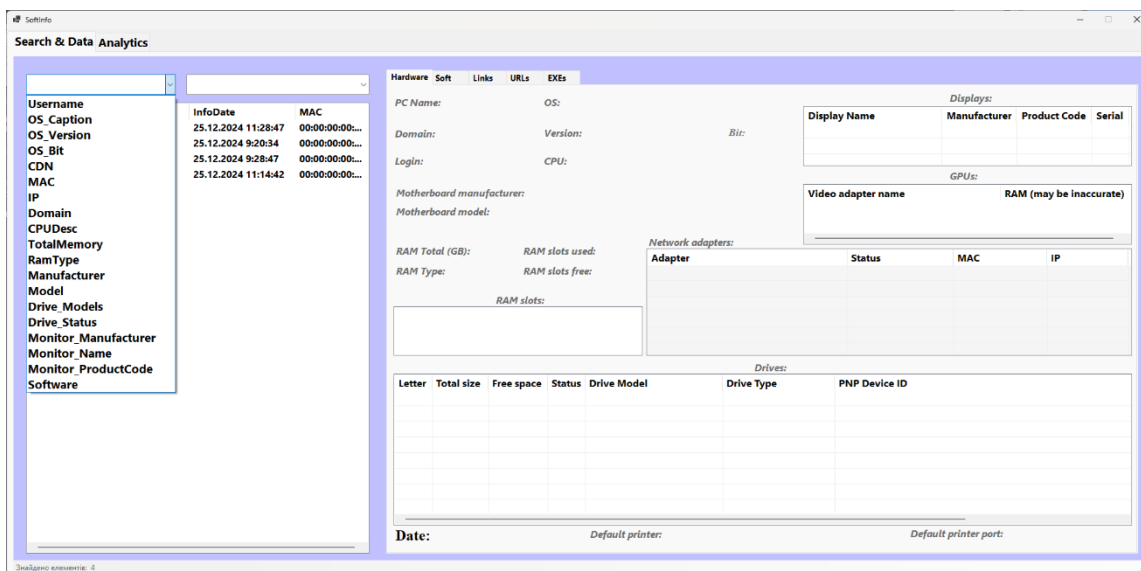


Рис. 3.3. Вибір поля по котрому здійснюватиметься пошук

Другий список – список значень – по замовчанню порожній. При здійсненні вибору типу поля в першому списку (по котрому відбуватиметься пошук), в другому полі список можливих значень буде заповнено списком значень того типу котрий обрано в першому списку, з усіх можливих значень наявних в документах. Так, якщо обрати в першому полі тип значення CPUDESC – (назва процесора комп’ютера), то в другому списку буде запропоновано для вибору перелік що сформовано на основі всіх наявних в базі даних процесорів (див. рис. 3.4).

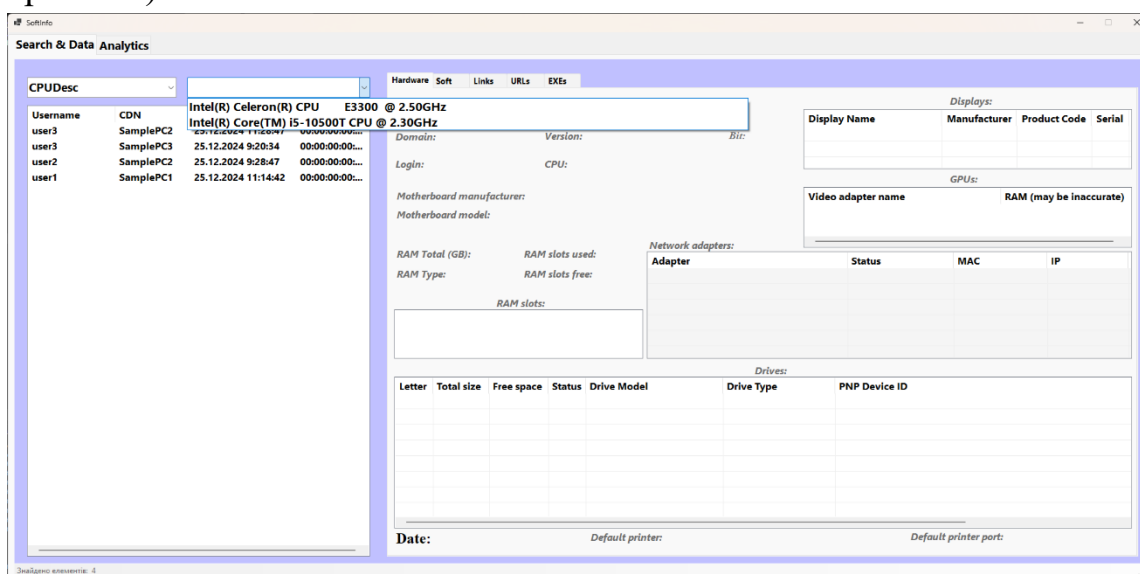


Рис. 3.4. Вибір варіантів з обраної категорії

При виборі значення в другому полі список пристроїв оновлюється з урахуванням обраного критерію пошуку. Тобто якщо обрати певну модель процесора в списку пристроїв відобразяться ті з них котрі обладнані обраним процесором, при тому в лівому нижньому кутку фіксується кількість елементів котрі відображаються у списку пристроїв (див. рис.3.5).

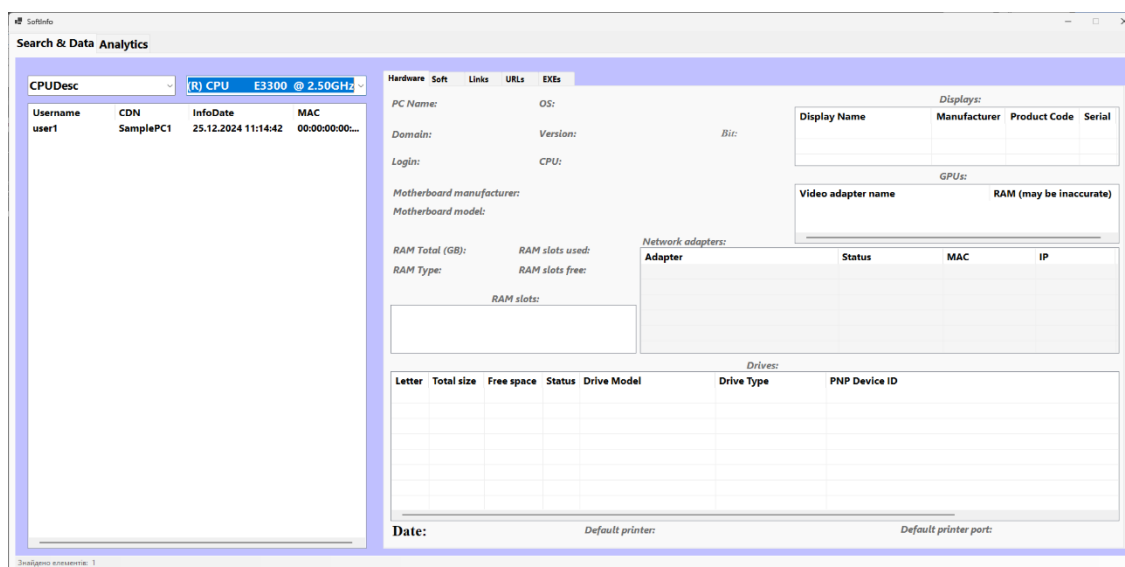


Рис. 3.5. Список пристроїв, що містять обраний процесор

При виборі пристрою зі списку права частина вікна заповнюється даними обраного пристрою (див. рис.3.6), а саме апаратними характеристиками, списком програмного забезпечення котре встановлене на даному пристрої і у даного

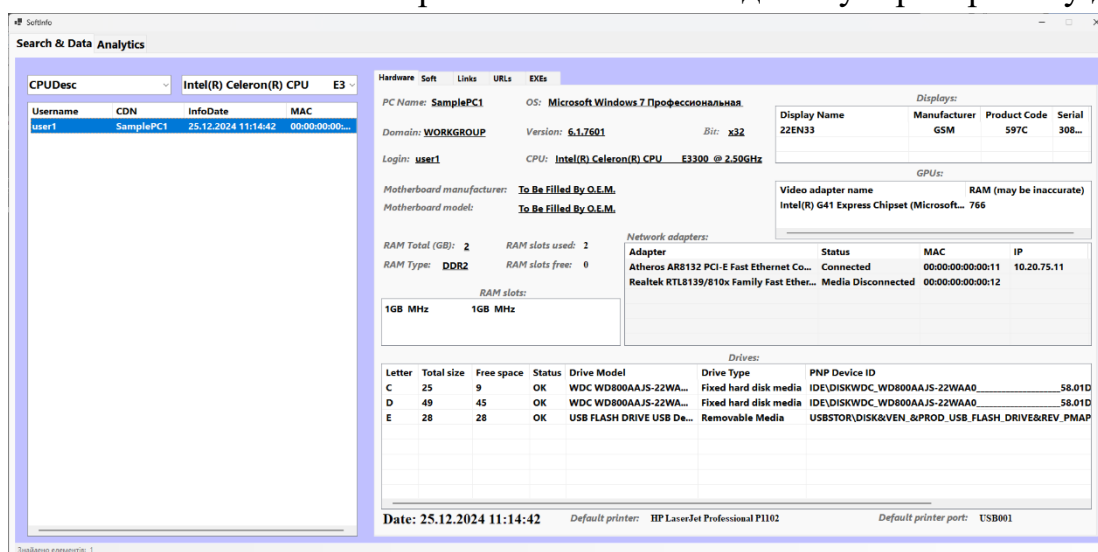


Рис. 3.6. Відображення параметрів обраного пристрою

користувача, ярликами та застосунками, що розміщені на робочому столі або у папках розміщених на робочому столі (як вже зазначалось в описі скрипту при зборі даних запускається рекурсивна функція котра зчитує усі папки та усі вкладені підпапки з робочого столу) (див. рис.3.7).

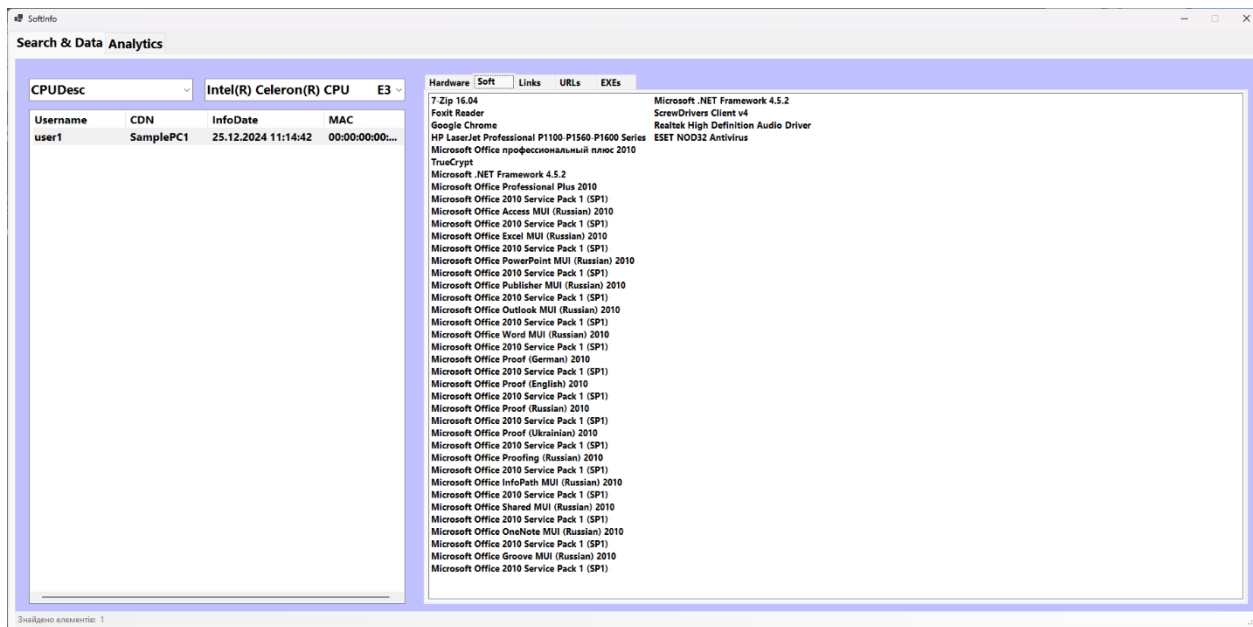


Рис. 3.7. Відображення списку програм для обраного пристрою

Одиначний клік по підкреслених параметрах або програмах копіює їх текст до буферу обміну (див. рис.3.8).

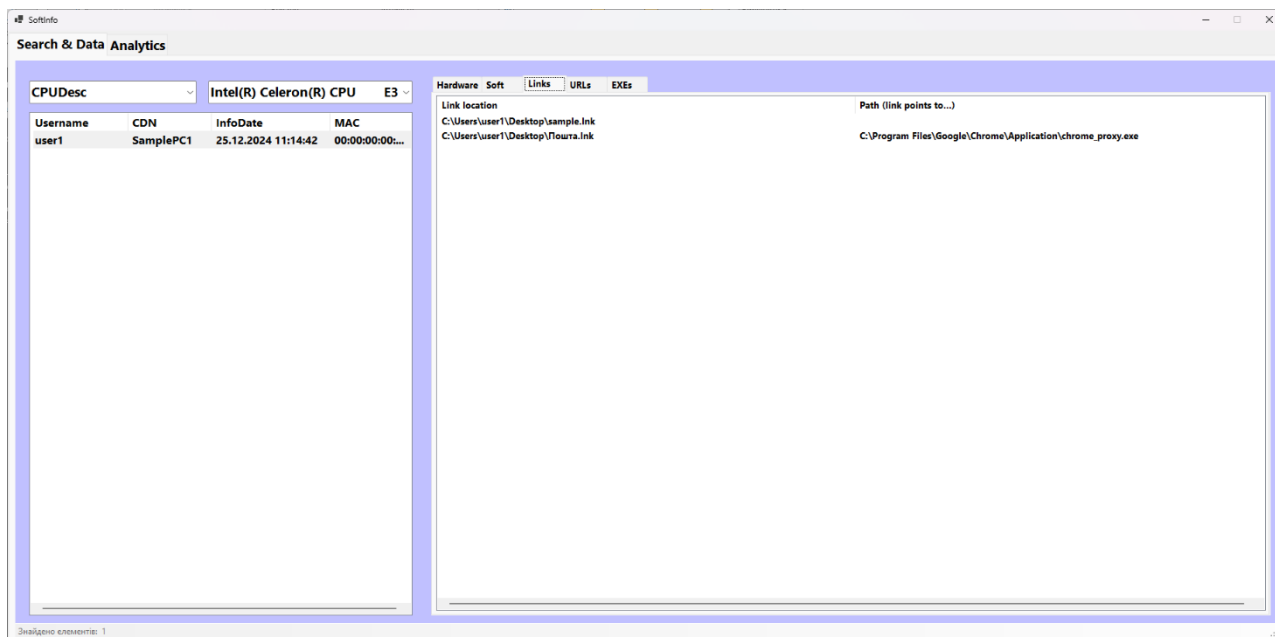


Рис. 3.8. Відображення списку ярликів для обраного пристрою

Подвійний клік по підкреслених елементах у вкладці параметрів, або по списку в розділі програм – виставляє обраний тип параметру та значення поточною умовою пошуку (див. рис.3.9), що дозволяє переглянути, які ще пристрої мають такий самий параметр (наприклад на яких ще комп’ютерах є такі-ж процесор, операційна система, або на які ще системи було здійснено вхід користувачем).

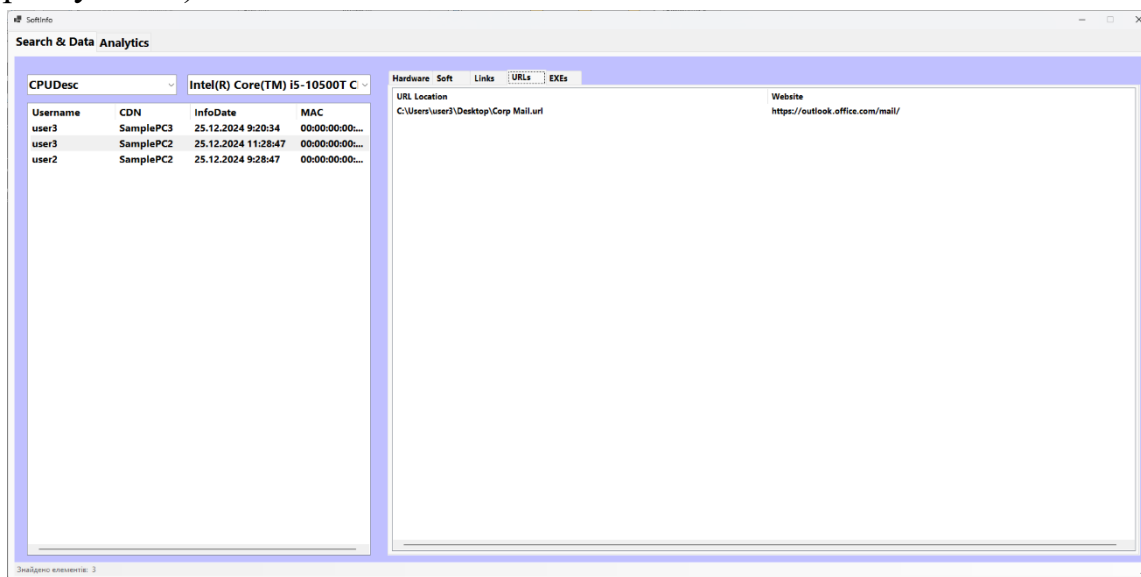


Рис. 3.9. Відображення списку закладок з робочого столу для обраного пристрою

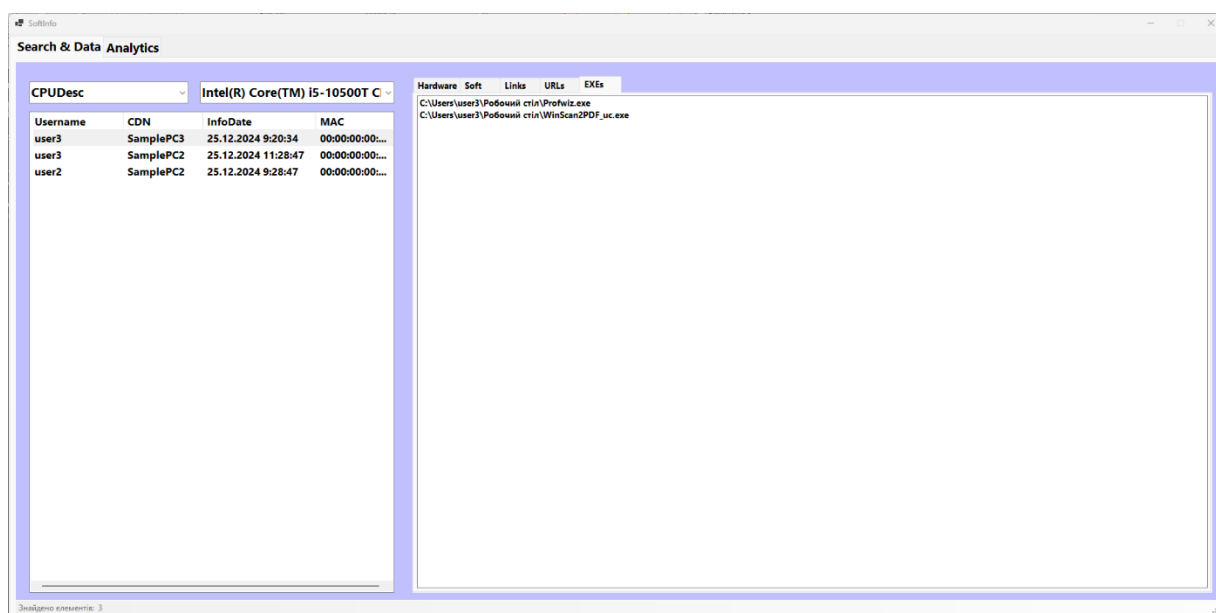


Рис. 3.10. Відображення списку застосунків з робочого столу для обраного пристрою

Таким чином можемо з легкістю обрати будь який параметр пошуку на основі даних обраного пристрою що пришвидшує роботу створюючи позитивний користувацький досвід.

Однак, незважаючи на всю користь вбудованих пошуків часто актуальною є необхідність додатково обробляти отримані дані, зберігати та надсилати отриману інформацію, звіряти знайдені дані з наявними – важко навіть заздалегідь передбачити всі потреби, котрі можуть виникнути в процесі використання, котрі неможливо буде здійснити в межах даного програмного комплексу. З метою вирішення цього питання для переважної більшості списків додане контекстне меню з опцією експорту цього списку в Microsoft Excel (див. рис. 3.11).

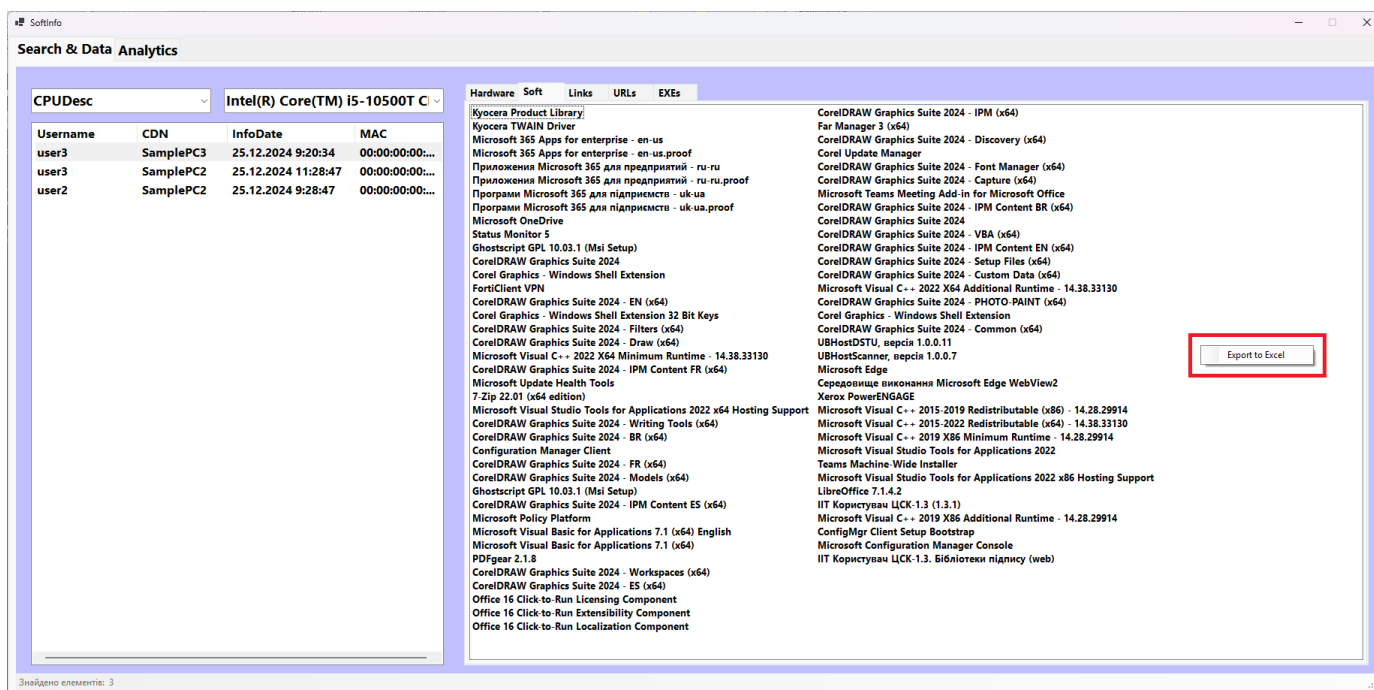


Рис. 3.11. Контекстне меню експорту

При експортуванні створюється (але не зберігається) новий документ Excel та до нього додаються дані з таблиці на основі котрої було викликане контекстне меню (див. рис.3.12).

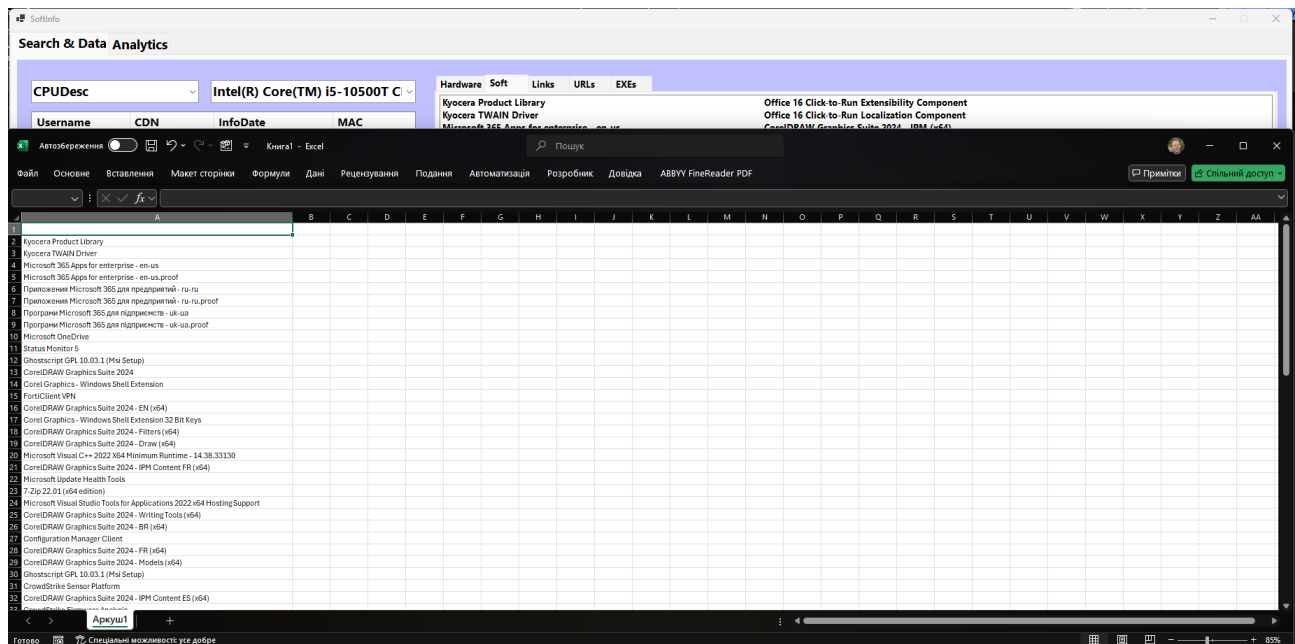


Рис. 3.12. Зразок експорту списку програм

Результуючу таблицю потім можливо зберегти за потреби, або закрити не зберігаючи.

3.3 Робота у вкладці «Analytics»

Вкладка «Analytics» задумана для відображення зведеної інформації по пристроях по обраному параметру (див. рис. 3.13).

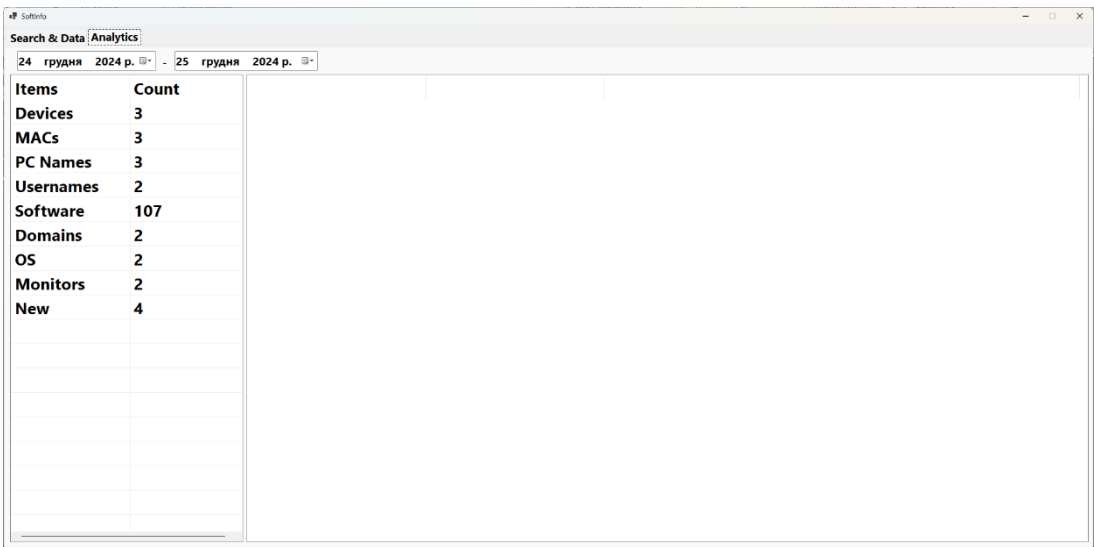
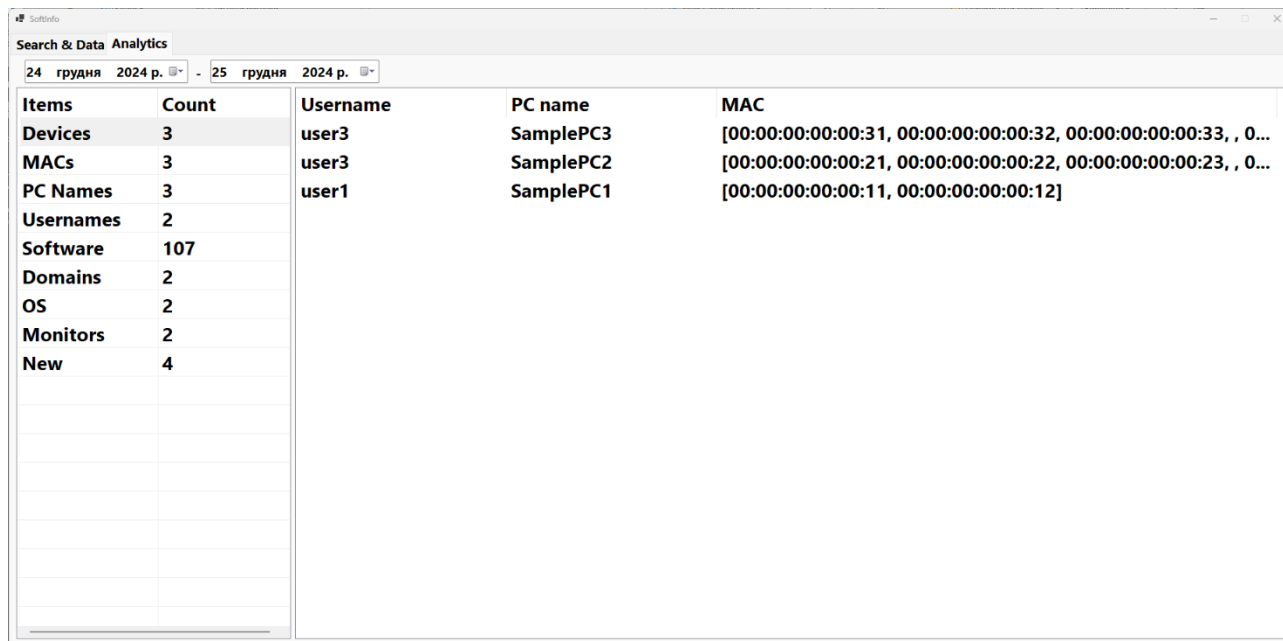


Рис. 3.13. Загальний вигляд вкладки «Analytics»

На даний момент передбачені наступні пункти в меню вибору: Devices – кількість пристроїв, при виборі виводить список пристроїв з логінами та Mac-адресами (див. рис.3.14).



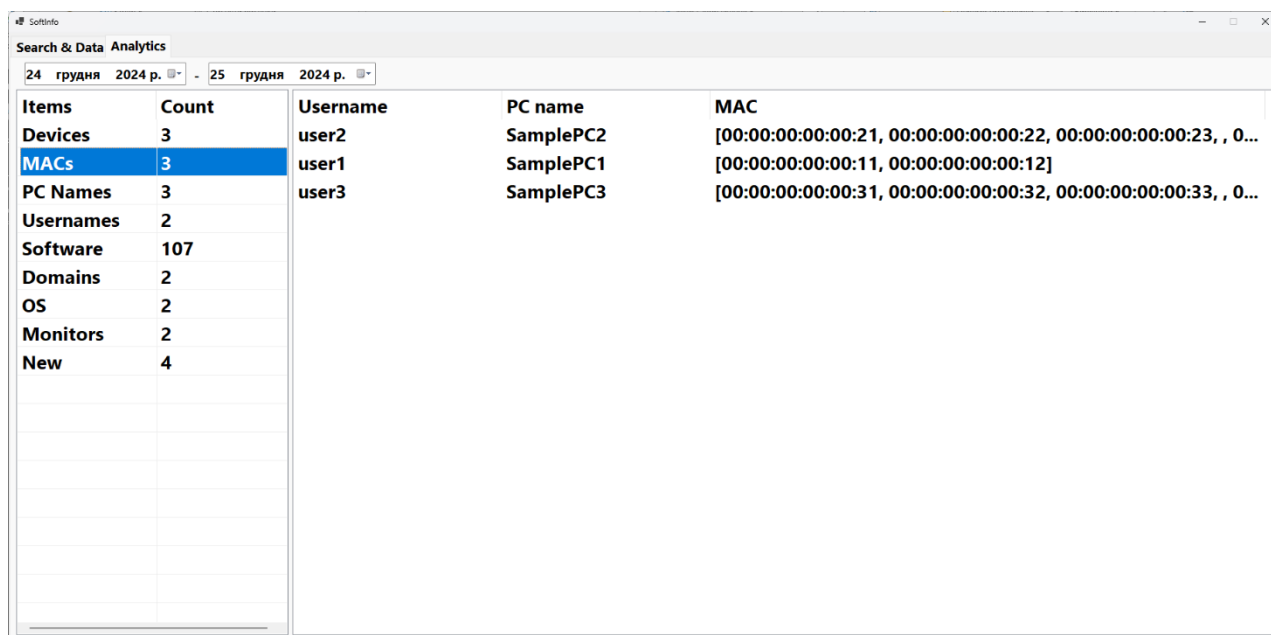
Items	Count	Username	PC name	MAC
Devices	3	user3	SamplePC3	[00:00:00:00:31, 00:00:00:00:32, 00:00:00:00:33, , 0...
MACs	3	user3	SamplePC2	[00:00:00:00:21, 00:00:00:00:22, 00:00:00:00:23, , 0...
PC Names	3	user1	SamplePC1	[00:00:00:00:11, 00:00:00:00:12]
Usernames	2			
Software	107			
Domains	2			
OS	2			
Monitors	2			
New	4			

Рис. 3.14. Відображення зведених даних при виборі елементу «Devices»

Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів: групування по полях «CDN» - ім'я комп'ютера та MAC-адреси одночасно (для кожного документу використовується перший елемент з масиву MAC-адрес) комбіноване групування дає можливість потенційно вивести більше документів ніж вивело б групування по якомусь одному з елементів окремо. Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу.

MACs – кількість унікальних MAC-адрес, при виборі виводить список пристроїв з логінами та Mac-адресами аналогічно попередньому пункту (див. рис.3.15). Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів: групування по полю MAC-адреси (для кожного документу використовується перший елемент з масиву MAC-адрес), можуть бути

ситуації коли кількість пристроїв у попередньому пункті буде більша (наприклад «здвоєння» після перейменування комп'ютера)



Items	Count	Username	PC name	MAC
Devices	3	user2	SamplePC2	[00:00:00:00:21, 00:00:00:00:22, 00:00:00:00:23, 0...
MACs	3	user1	SamplePC1	[00:00:00:00:11, 00:00:00:00:12]
PC Names	3	user3	SamplePC3	[00:00:00:00:31, 00:00:00:00:32, 00:00:00:00:33, 0...
Usernames	2			
Software	107			
Domains	2			
OS	2			
Monitors	2			
New	4			

Рис. 3.15. Відображення зведених даних при виборі елементу «MACs»

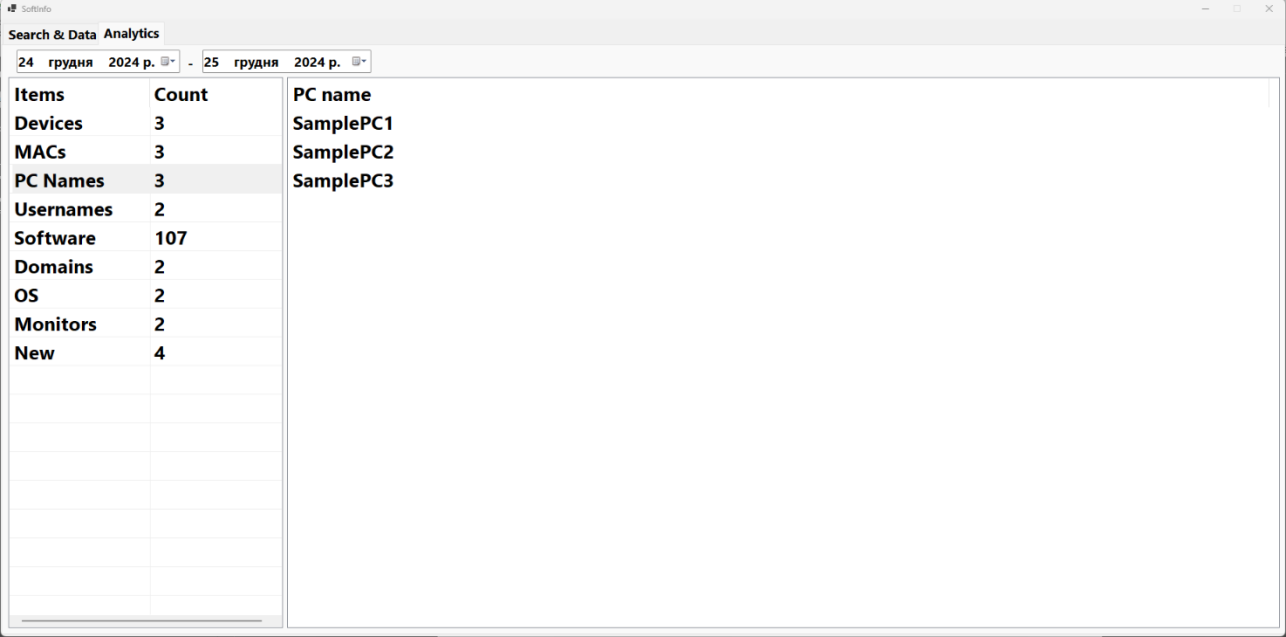
Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу.

PC Names – кількість унікальних доменних імен комп'ютера, при виборі виводить список доменних імен комп'ютерів.

Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів:

Групування по доменному імені дає змогу відсіяти комп'ютери котрі заходили під одним й тим самим іменем, але різними MAC-адресами (наприклад використання різних мережевих адаптерів чи заміна адаптера).

Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу (див. рис.3.16).



The screenshot shows the SiftInfo Search & Data Analytics window. The date range is set from 24 грудня 2024 р. to 25 грудня 2024 р. The left sidebar contains a table with the following data:

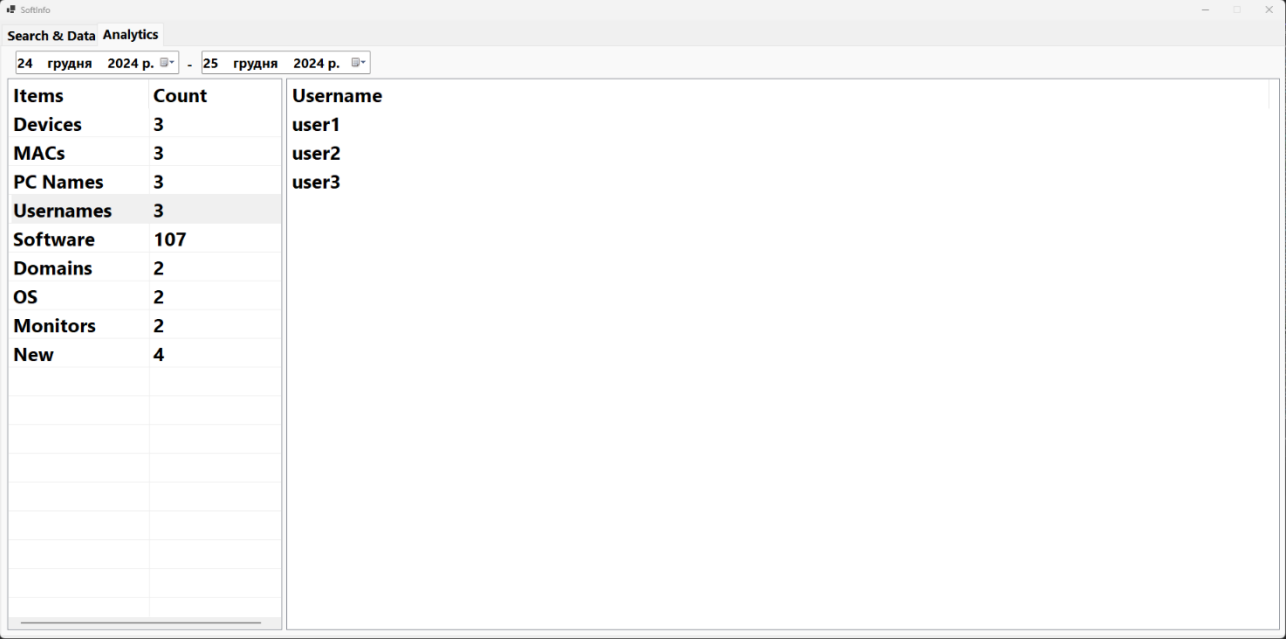
Items	Count
Devices	3
MACs	3
PC Names	3
Username	2
Software	107
Domains	2
OS	2
Monitors	2
New	4

The main panel displays the results for 'PC Names', showing a list of three sample PC names:

PC name
SamplePC1
SamplePC2
SamplePC3

Рис. 3.16. Відображення зведених даних при виборі елементу «PC Names»

Username – кількість унікальних імен користувачів (логіни), при виборі виводить список імен користувачів (див. рис.3.17).



The screenshot shows the SiftInfo Search & Data Analytics window. The date range is set from 24 грудня 2024 р. to 25 грудня 2024 р. The left sidebar contains a table with the following data:

Items	Count
Devices	3
MACs	3
PC Names	3
Username	3
Software	107
Domains	2
OS	2
Monitors	2
New	4

The main panel displays the results for 'Usernames', showing a list of three sample usernames:

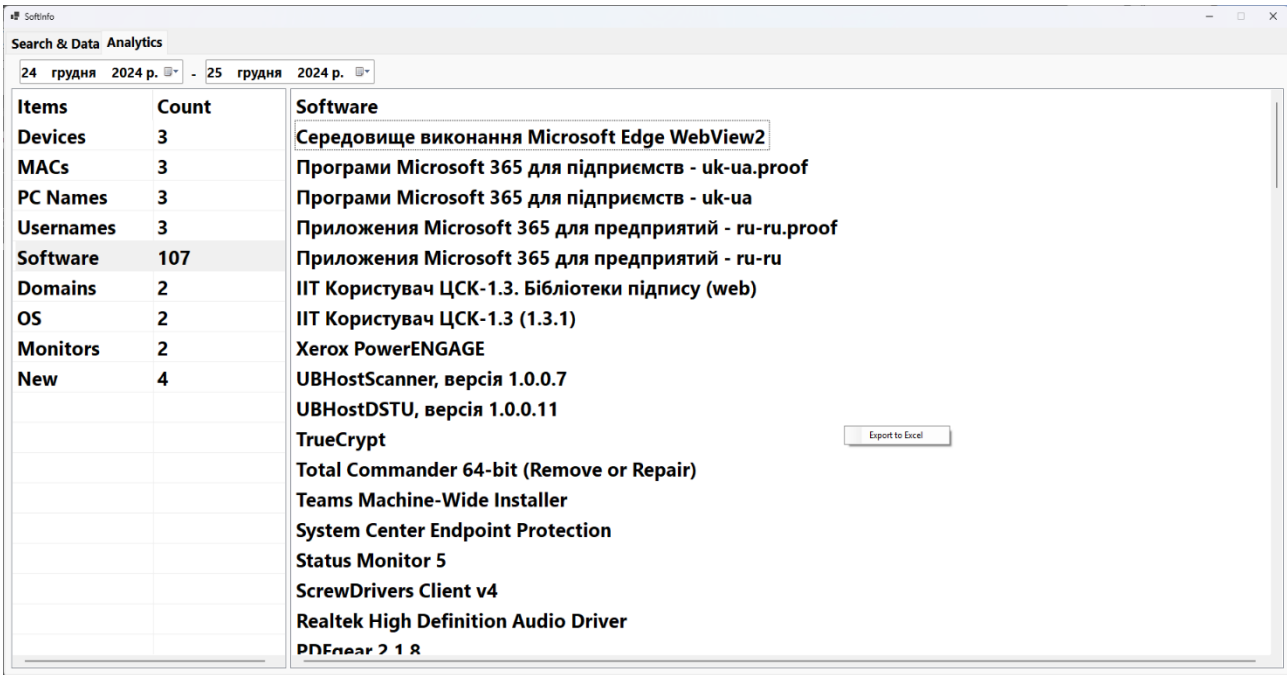
Username
user1
user2
user3

Рис. 3.17. Відображення зведених даних при виборі елементу «Usernames»

Software – кількість унікальних програмних продуктів відповідно до останніх отриманих даних. Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів (див. рис.3.18).

Сортування по даті в результаті чого найновіші документи будуть на початку списку. Групування по комбінації полів MAC-адреси, доменного імені комп'ютера, та логіну користувача одночасно з відбором останнього елементу (з найбільшим значенням дати завдяки попередньому сортуванню) в кожній групі.

Заміна кореневого елементу – службовий етап, підготовка даних для наступного етапу. \$unwind – розгортання масиву, підготовчий етап для групування масиву поелементно, якщо пропустити порівнюватиме масиви цілком а не їх елементи, що нам не підходить. Групування по програмному забезпеченню. Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу.



Items	Count	Software
Devices	3	Середовище виконання Microsoft Edge WebView2
MACs	3	Програми Microsoft 365 для підприємств - uk-ua.proof
PC Names	3	Програми Microsoft 365 для підприємств - uk-ua
Usernames	3	Приложения Microsoft 365 для предприятий - ru-ru.proof
Software	107	Приложения Microsoft 365 для предприятий - ru-ru
Domains	2	IIT Користувач ЦСК-1.3. Бібліотеки підпису (web)
OS	2	IIT Користувач ЦСК-1.3 (1.3.1)
Monitors	2	Xerox PowerENGAGE
New	4	UBHostScanner, версія 1.0.0.7
		UBHostDSTU, версія 1.0.0.11
		TrueCrypt
		Total Commander 64-bit (Remove or Repair)
		Teams Machine-Wide Installer
		System Center Endpoint Protection
		Status Monitor 5
		ScrewDrivers Client v4
		Realtek High Definition Audio Driver
		PDFgear 2.1.8

Рис. 3.18. Відображення зведених даних по програмах «Software», контекстне меню експорту

Domains – кількість наявних доменів. Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів. Сортування по

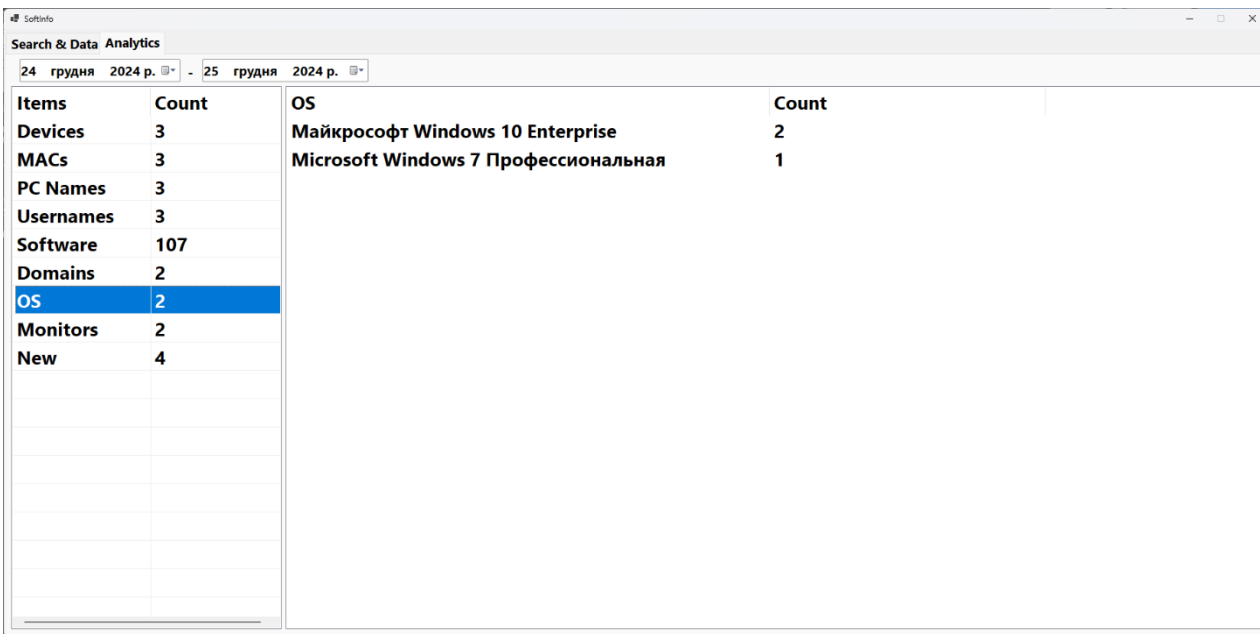
даті в результаті чого найновіші документи будуть на початку списку. Групування по МАС-адресі з відбором першого документа (з найбільшим значенням дати завдяки попередньому сортуванню) в кожній групі, тобто для кожної МАС-адреси (див. рис.3.19).

Таким чином маємо змогу відсіяти старі записи якщо наявні комп'ютери котрі були переведені в інший домен з домену або групи котрі більше не використовуються. Заміна кореневого елементу – службовий етап, підготовка даних для наступного етапу. Групування по домену. Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу.

[illegible]

Рис. 3.19. Відображення зведених даних по доменах «Domains»

S - кількість унікальних операційних систем, відповідно до останніх отриманих даних. Даний параметр розраховується за допомогою конвеєру агрегації з використанням наступних етапів (див. рис.3.20).



The screenshot shows the 'SoftInfo Search & Data Analytics' window. At the top, there are date filters for '24 грудня 2024 р.' and '25 грудня 2024 р.'. Below the filters, there are two tables. The left table lists various items and their counts, with 'OS' highlighted in blue. The right table shows the details for the selected 'OS' category.

Items	Count
Devices	3
MACs	3
PC Names	3
Username	3
Software	107
Domains	2
OS	2
Monitors	2
New	4

OS	Count
Майкрософт Windows 10 Enterprise	2
Microsoft Windows 7 Профессиональная	1

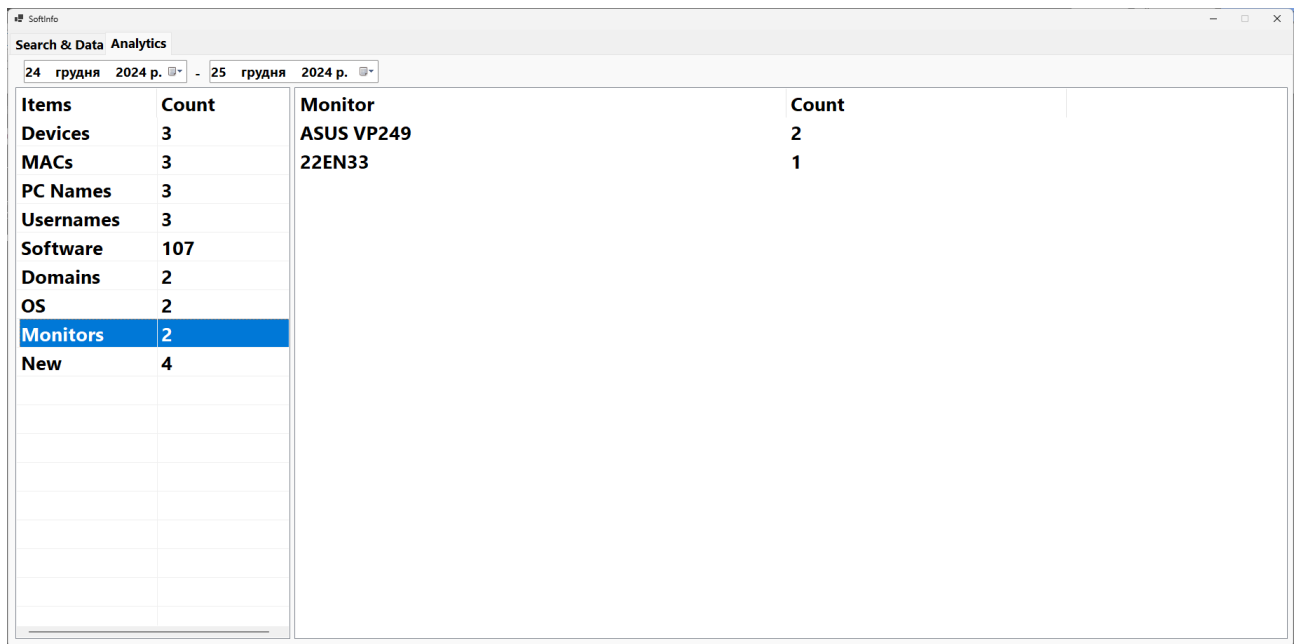
Рис. 3.20. Відображення зведених даних по обраному параметру «OS»

Сортування по даті в результаті чого найновіші документи будуть на початку списку. Групування по MAC-адресі з відбором першого документа (з найбільшим значенням дати завдяки попередньому сортуванню) в кожній групі, тобто для кожної MAC-адреси.

Таким чином маємо змогу відсіяти старі записи якщо наявні комп'ютери, на яких була перевстановлена операційна система (наприклад внаслідок модернізації комп'ютера змогли замінити Windows 7 на Windows 10).

Заміна кореневого елементу – службовий етап, підготовка даних для наступного етапу. Групування по операційних системах. Підрахунок кількості – етап котрий підраховує кількість елементів отриманих в результаті попереднього етапу.

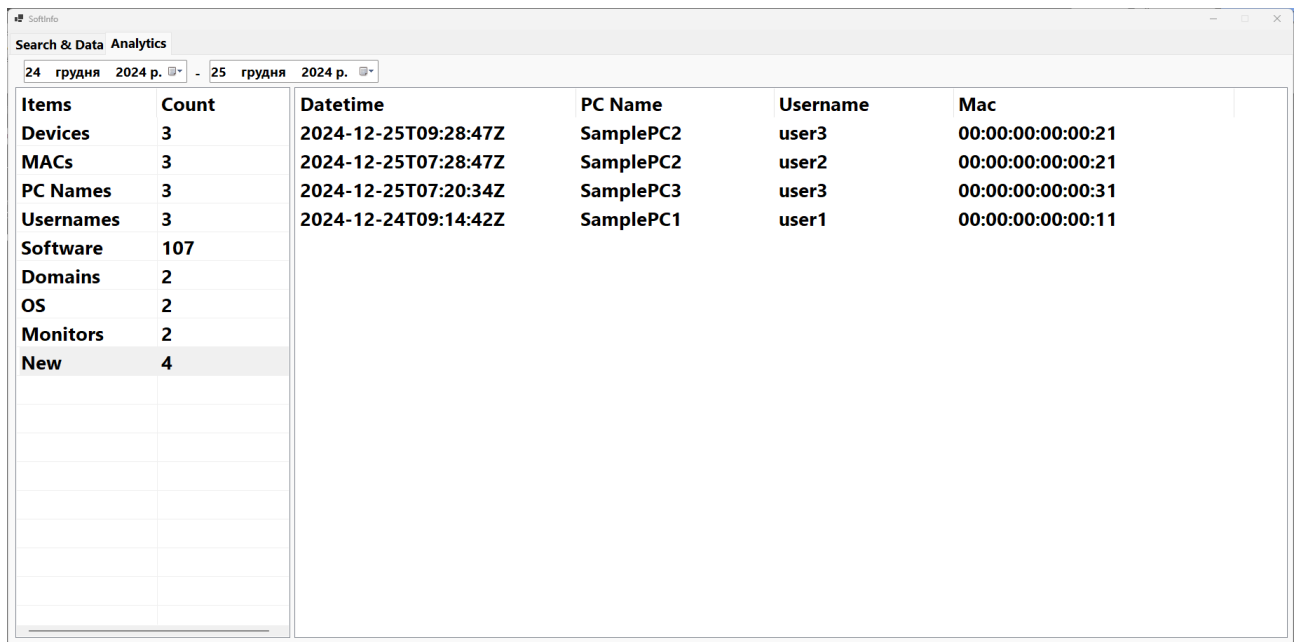
Monitors – кількість типів моніторів. (див. рис.3.21)



Items	Count	Monitor	Count
Devices	3	ASUS VP249	2
MACs	3	22EN33	1
PC Names	3		
Username	3		
Software	107		
Domains	2		
OS	2		
Monitors	2		
New	4		

Рис. 3.22. Відображення зведених даних по обраному параметру «Monitors»

New – відображає лише ті пристрої, записи про котрі з'явилися вперше в межах вказаного діапазону дат. (див. рис.3.22)



Items	Count	Datetime	PC Name	Username	Mac
Devices	3	2024-12-25T09:28:47Z	SamplePC2	user3	00:00:00:00:00:21
MACs	3	2024-12-25T07:28:47Z	SamplePC2	user2	00:00:00:00:00:21
PC Names	3	2024-12-25T07:20:34Z	SamplePC3	user3	00:00:00:00:00:31
Username	3	2024-12-24T09:14:42Z	SamplePC1	user1	00:00:00:00:00:11
Software	107				
Domains	2				
OS	2				
Monitors	2				
New	4				

Рис. 3.22. Відображення зведених даних по обраному параметру «New»

Діапазон дат визначається двома елементами для вибору дат, при запуску програми вони автоматично заповнюються відповідно найдавнішою датою (дата

збору інформації про, імовірно, перший доданий пристрій) і найсвіжішою датою (останній доданий пристрій) і не можуть їх перевищувати. При редагуванні діапазону дат всі показники перераховуватимуться і відображатимуть інформацію відповідно до обраного діапазону.

Висновки до третього розділу

В даному розділі наведений опис порядку запуску і роботи користувача з програмою. Розглянуто графічний інтерфейс програми. Описано можливості пошуку по довільних параметрах та перегляду детальної інформації про пристрої у закладці Search&Data, зведення/групування даних по різних критеріях в розділі Analytics та експорту даних в Microsoft Excel.

ВИСНОВКИ

В результаті проведених досліджень було отримані наступні основні висновки.

Проаналізовано та детально розглянуто аналоги програмного забезпечення. Виявлені явні переваги та недоліки наявного функціоналу. Завдяки розглянутій інформації було сформовано орієнтовний план роботи програмного додатку. На основі розглянутих аналогів було підкреслено позитивні аспекти, які варто допрацювати та реалізувати.

Визначено, що головними елементами програмного додатку для оптимізації процесів оптимізації процедури інвентаризації є скрипт для збору інформації, база даних для її збереження та застосунок для пошуку та відображення зібраної інформації.

В даній роботі було реалізовано інформаційно-аналітичний автоматизований програмний комплекс, котрий збирає, зберігає, обробляє та відображає в зручному для сприйняття вигляді дані про апаратне та програмне забезпечення робочих місць в межах доменної мережі підприємства. Розглянуто проблеми, що виникають в процесі реалізації та використання, та запропоновано їх рішення. Програмний комплекс пройшов випробування на практиці. Отримані результати відповідають поставленим до програми вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кустов В., Коваленко М. Теоретико-методологічні підходи до управління іт-інфраструктурою в умовах цифровізації. Herald of Khmelnytskyi National University. Economic sciences, Vol. 322, №. 5, pp. 50–56, Sep. 2023, DOI: [10.31891/2307-5740-2023-322-5-7](https://doi.org/10.31891/2307-5740-2023-322-5-7).
2. Великоборець О. А. Перспективи використання інформаційних технологій в газорозподільних підприємствах. Технічна творчість: Збірник наукових праць./ Укл.: Скиба М.Є., Поліщук О.С., Романець Т.П. Хмельницький: ХНУ, 2023. № 7. С.134-136. Режим доступу: https://maees.khmnpu.edu.ua/wp-content/uploads/2024/02/Технічна-творчість_Збірник_№7_2023.pdf.
3. Басін, А. Аналітичне забезпечення операційного управління іт-підприємств України: ключові аспекти та виклики», Економіка та суспільство, 2025. (71). DOI: 10.32782/2524-0072/2025-71-10.
4. Сидорова А. В., Біленко Д. В., Буркіна Н. В. Бізнес-аналітика: навчально-методичний посібник. Вінниця: ДонНУ імені Василя Стуса. 2020. 104 с.
5. Андрєєва М. В. Шляхи вдосконалення інвентаризаційного процесу в сучасних умовах. Актуальні аспекти соціально-економічного розвитку України: погляд молоді: тези IV Всеукр. студент. наук.-практ. конф., Одеса, 20-21 жовт. 2022 р. Одес. нац. технол. ун-т. Навч.-наук. ін-т прикладної економіки і менеджменту ім. Г. Е. Вейнштейна. Одеса, 2022. С. 102–104.
6. Фрундіна Л.І. Вдосконалення процесу інвентаризації шляхом застосування комп'ютерних технологій. International scientific e-journal ЛОГОΣ. ONLINE. 2020. Режим доступу: <https://www.ukrlogos.in.ua/10.11232-2663-4139.16.64.html>.
7. Крикунова М. О. Інвентаризація як форма контролю майна підприємства : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 071 - облік і оподаткування. Суми : Сумський державний університет, 2023. 48 с.

8. Костюнік О., Чайковська Т. Інвентаризація необоротних активів як один із елементів методу бухгалтерського обліку. Молодий вчений, 2020. № 9 (85). С. 87-90. DOI: 10.32839/2304-5809/2020-9-85-21.

9. Кузуб М., Пучок М. Інвентаризація в управлінні підприємства. Socially competent management of corporations in a behavioral economy: Collection of scientific papers / resp. ed Konstantin Pavlov, Elena Pavlova, Ludmila Shostak, Tatiana Bukoros, Josef Zyatko, 2021. European institute of further education, Podhájka, 2021. Chapter II. С.257–258.

10. Максименко І. Я. Максименко Ю. О. Інвентаризація запасів як метод контролю в управлінні діяльністю підприємства. In The 11th International scientific and practical conference “Eurasian scientific congress”(November 1-3, 2020) Barca Academy Publishing, Barcelona, Spain. 2020. P. 538.

11. Поплюйко К.В. Інвентаризація як елемент методу обліку. Актуальні проблеми та перспективи розвитку обліку, аналізу та контролю в соціально-орієнтованій системі управління підприємством: Матеріали II Всеукраїнської науково-практичної конференції. Частина 1. (м. Полтава, 23 квітня 2019 року). Полтава, 2019. С. 179-180.

12. Шевців Л.Ю., Дарчук П.О. Особливості проведення інвентаризації на підприємствах України: стан і перспективи. Proceedings of the 10th International Scientific and Practical Conference «Challenges in Science of Nowadays». (November 16-18, 2022). Washington, USA. №132. С.78-85.

13. Кузуб, М. В., Перківська М. Діджиталізація бухгалтерського обліку в Україні. Бухгалтерський облік, контроль та аналіз в умовах інституційних змін: Збірник наукових праць VII Всеукраїнської науково-практичної конференції.Полтава: ПДАУ, 2024. С. 595-597. Режим доступу: <https://drive.google.com/file/d/1D0af4IRnGzC4hrV5tDD6-TG-TiklPBR4/view>

14. Крилович І. О., Рижук Д. В. Інвентаризація як елемент бухгалтерського обліку: проблеми та шляхи удосконалення. сучасні проблеми і перспективи розвитку обліку, аналізу і контролю в умовах глобалізації економіки Тези

доповідей XIII Міжнародної науково-практичної конференції молодих науковців, аспірантів, здобувачів і студентів. 2021. Випуск 15. С. 241.

15. Городецька М. О. Програмне забезпечення: специфіка обліку та інвентаризації : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 071 - облік і оподаткування / наук. кер. Ю. С. Серпенінова. Суми : Сумський державний університет, 2021. с. 38.

16. Product Features - Snipe-IT Free open source IT asset management [Електронний ресурс] – Режим доступу до ресурсу: <https://snipeitapp.com/>

17. Network Inventory Tool: Computer Auditing and Inventory Management Software [Електронний ресурс] – Режим доступу до ресурсу: <https://www.total-network-inventory.com/>

18. Білецький В. С., Методологія наукових досліджень технічних об'єктів та їх оптимізація. Харків: ФОП Халіков Руслан Халікович, 2023. Accessed: Mar. 28, 2025. [Електронний ресурс] – Режим доступу до ресурсу: <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/04399d6a-36c7-4bd6-949e-9a756e6908ef/content>

19. Горбань Г. В. Засоби автоматизації завдань в операційній системі Windows : навч. посіб. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2021. – 168 с.

20. MongoDB: The Developer Data Platform | MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/>

21. Ісаков, А. В., Ліміна, В. Ю., Романюк, О. В. Сфери застосування нереляційних баз даних [Електронний ресурс] – Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/29409/9416.pdf>.

22. Белей М.-В. А., Ващишак, С. П. Використання віддаленої бази даних MongoDB як інструменту для збереження даних на сервері. Студентські наукові дискусії поза форматом: матеріали XI Міжнародної наукової конференції (м. Івано-Франківськ, 11 квітня 2024 року). Івано-Франківськ: Редакційно-видавничий відділ ЗВО «Університет Короля Данила». 2024. С.50-51.

23. Мальований Д. В., Богач І. В.. Комп'ютерна програма "Реалізація програми для збору і аналізу статистики сервера, використовуючи мову Java та фреймворк Spring Boot ("server-metric-processor")" (№110271). Свідоцтво про реєстрацію авторського права на твір №110271 UA. Комп'ютерна програма «Реалізація програми для збору і аналізу статистики сервера, використовуючи мову Java та фреймворк Spring Boot ("server-metric-processor")» [Текст] / Д. В. Мальований, І. В. Богач (Україна), Міністерство економічного розвитку і торгівлі України. Дата реєстрації від 13.12.2021 р..

24. Дацко І. А. Засоби .NET для розробки програмного забезпечення / І. А. Дацко // Кібербезпека в сучасному світі : матеріали III Всеукраїнської науково-практичної конференції (м. Одеса, 19 листопада 2021 р.) / за ред. О. В. Дикого ; уклад.: С. А. Горбаченко, Н. І. Логінова. Одеса, 2021. С. 124-127.

25. Методи оптимізації: Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 124 «Системний аналіз» / КПІ ім. Ігоря Сікорського; уклад.: А.П. Яковлева, І.Я. Спекторський. Електронні текстові дані (1 файл: 0,49 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2021. 80 с. Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/222e5ba9-a5a6-4f94-acae-77bef6b23df7/content>.

ДОДАТКИ

Додаток А.

ПУБЛІКАЦІЇ



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА АКАДЕМІЯ ПЕДАГОГІЧНИХ НАУК УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

ТЕХНІЧНА ТВОРЧІСТЬ

ЗБІРНИК НАУКОВИХ ПРАЦЬ

№7 грудень 2023 р.

№7 грудень 2023 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА АКАДЕМІЯ ПЕДАГОГІЧНИХ НАУК УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Збірник наукових праць «Технічна творчість»

Засновано в жовтні 2015р.

Виходить 2 рази на рік

Хмельницький 2023, №7

Технічна творчість: Збірник наукових праць/ Укл: Скиба М.Є., Поліщук О.С., Романець Т.П. - Хмельницький: ХНУ, 2023. – № 7 - 191 с.

Редакційна колегія:

Голова:	Скиба М. Є. - д.т.н., проф. член-кореспондент НАПНУ
Співголова:	Радкевич В.О. - д.п.н., проф., член-кореспондент НАПНУ
Заступник голови:	Поліщук О.С. – д.т.н., проф.
Відповідальний секретар:	Романець Т.П. – к.т.н., доц.
Технічні редактори:	Горященко С.Л. –к.т.н., доц. Лісевич С.П. – старший викладач

Члени редакційної колегії:

Бонек М. д.т.н. (Польща), Бромберек Ф. д.т.н. (Польща), Бурмістенков О.П. д.т.н., Гергель М. д.т.н. (Польща), Гордєєв А.І. д.т.н., Діха О.В. д.т.н., Драпак Г.М. к.т.н., Зашенкіна Н.М. д.т.н., Злотенко Б.М. д.т.н., Калда Г.С. д.т.н., Мазуркевич А. д.т.н. (Польща), Майдан П.С. к.т.н., Мартинюк А.В. к.т.н., Матушевський М. д.т.н. (Польща), Міних В.І. д.т.н., Місяць В.П. д.т.н., Мусял Я. д.т.н. (Польща), Неймак В.С. к.т.н., Олександренко В.П. д.т.н., Орловський Б.В. д.т.н., Панасюк І.В. д.т.н., Пуць В.С. к.т.н., Синюк О.М. д.т.н., Смутко С.В. к.т.н., Снядковський М. д.т.н. (Польща), Танські Т. д.т.н. (Польща), Ткачук В.П. к.т.н., Троціковський Т. д.т.н. Польща, Чорний О.П. д.т.н.

**Рекомендовано до друку Рішенням Науково-технічної Ради
Хмельницького національного університету
Протокол № 15 від 18.12.2023 р.**

Наукові праці друкуються у авторській редакції
За достовірність фактів, назв та посилань на літературні джерела
відповідальність несуть автори

Адреса редакції Збірника наукових праць «Технічна творчість»:
Хмельницький національний університет
вул. Інститутська, 5, м. Хмельницький, Україна, 29016
кафедра машин і апаратів, електромеханічних та енергетичних систем
<https://maees.khmnpu.edu.ua>
e-mail: maees1969@gmail.com

© Хмельницький національний університет 2023
© Редакція Збірника наукових праць «Технічна творчість» 2023

Технічна творчість №7 2023

43	М. Кашпуренко, А. Гордєєв. УДОСКОНАЛЕННЯ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИГОТОВЛЕННЯ ДЕТАЛІ «КОРПУС ПРИВОДУ 421-001» З РОЗРОБЛЕННЯМ ПРОГРАМ ДЛЯ РОЗРАХУНКІВ СИЛИ ЗАТИСКУ ТА МІЦНОСТІ ЕЛЕМЕНТІВ ПРИСТРОЇВ	97
44	В. Янчук, А. Сельський, Ю. Соколан. МОДЕЛЮВАННЯ ПРОЦЕСУ КЕРУВАННЯ АВТОМАТИЗОВАНИМ СКЛАДОМ ЗБЕРІГАННЯ ПРОДУКЦІЇ	100
45	Р. Тимчик, А. Сельський. ПРОБЛЕМАТИКА ОРГАНІЗАЦІЇ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ КАТАЛІТИЧНОГО РЕФОРМУВАННЯ	102
46	О. Євдокимов, І. Форкун. ПРОБЛЕМАТИКА ПЕРЕМІЩЕННЯ БУДІВЕЛЬНИХ МАТЕРІАЛІВ НА АВТОМАТИЗОВАНОМУ СКЛАДІ	104
47	О. Татаревська, А. Сельський. МОДЕЛЮВАННЯ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ ОТРИМАННЯ ТОВАРНОГО БЕНЗИНУ	106
48	Д. Ковбасюк, Г. Радельчук. ВИБІР СПОСОБУ РЕГУЛЮВАННЯ КУТОВОЇ ШВИДКОСТІ ДВИГУНІВ ПОСТІЙНОГО СТРУМУ	108
49	О. Кунцов, О. Синюк. ОСОБЛИВОСТІ ЛИТТЯ ПІД ТИСКОМ НИЗУ ВЗУТТЯ ...	109
50	О. Kravchenko. SELECTION AND JUSTIFICATION OF ARCHITECTURE FOR A WEB-BASED SYSTEM FOR MILITARY TRAINING COURSES	112
51	А. Поліщук, М. Скиба, О. Поліщук. МОДЕЛЮВАННЯ РОБОТИ ШНЕКОВОГО ЕКСТРУДЕРА 3D-ПРИНТЕРА В ПРОГРАМНОМУ СЕРЕДОВИЩІ SOLIDWORKS FLOW SIMULATION	114
52	V. Lavrenchuk. CUTTING-EDGE TECHNOLOGIES FOR ENHANCED EMPLOYMENT INTERACTIONS	116
53	В. Мица, А. Мовчанюк. ПЕРЕВАГИ 3D МОДЕЛЮВАННЯ У СТВОРЕННІ ПРОДУКТУ	117
54	A. Zhurbey, R. Savitsky. SECURITY OF INFORMATION SYSTEMS	119
55	Ж. Жлалі, М. Рубанка, В. Дворжак, О. Поліщук. СИСТЕМА АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ AUTODESK INVENTOR ТА ОСНОВНІ ЇЇ ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ	121
56	V. Lavrenchuk. IMPLEMENTATION OF A RECOMMENDATION SYSTEM FOR PROFESSIONAL DEVELOPMENT	123
57	A. Hlobina. SYMBIOSIS OF PROGRAMMING AND CYBERSECURITY: PATH TO A RESILIENT AND SECURE DIGITAL WORLD	125
58	R. Tsmokh. INTERACTIVE ELECTRONIC SYSTEMS FOR LEARNING THE FUNDAMENTALS OF PROGRAMMING IN THE CONTEXT OF VISUAL PROGRAMMING	127
59	D. Lyman, O. Vlasenko. THE NEED TO SPEED UP THE IDENTIFICATION OF DISEASES	129
60	D. Lyman, O. Vlasenko. MEDICAL DATA ANALYSIS SYSTEM	130
61	Е. Федоренко. СТВОРЕННЯ ВЕБ-САЙТУ ЕЛЕКТРОТЕХНІКИ З АНАЛІЗОМ РИНКОВОГО ПОШУКУ	131
62	Б. Тимченко, С. Горященко. МОДЕЛЮВАННЯ РОБОТИ КОНТАКІВ ЕЛЕКТРИЧНИХ РЕЛЕ В ПП SOLID WORKS	132
63	О. Великоборець. ПЕРСПЕКТИВИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ГАЗОРОЗПОДІЛЬНИХ ПІДПРИЄМСТВАХ	134
64	М. Колиско, В. Павленко, М. Поліщук. МОЖЛИВОСТІ УДОСКОНАЛЕННЯ ФУНКЦІОНАЛЬНОЇ СКЛАДОВОЇ ВІРТУАЛЬНИХ НАВЧАЛЬНИХ	

правильно. На рисунку 3 зображено переміщення контактної пружини внаслідок дії сили.

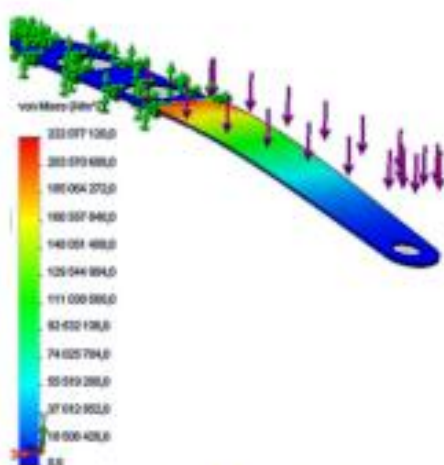


Рисунок 2 – Напруження в контактній пружині

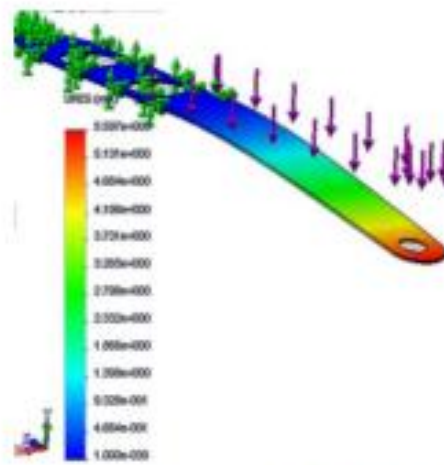


Рисунок 3 – Переміщення внаслідок дії сили на контактну пружину

Таким чином, моделювання контактів електричного реле визначається як важливий інструмент для оптимізації та покращення їхньої роботи.

Перелік посилань

1. Дудюк Д. Л., та ін.. Електричні вимірювання. Навчальний посібник – Львів: Афіша, 2003 – 272 с.
2. Altukhov, O., & Tarasov, O. (2022). Розробка програмних компонентів для моделювання інтенсивного пластичного деформування на основі параметричної моделі оснащення. Обробка матеріалів тиском, (1(51)), 214-223. [https://doi.org/10.37142/2076-2151/2022-1\(51\)214](https://doi.org/10.37142/2076-2151/2022-1(51)214)
3. Development of scientific foundations of resource-saving technologies of mineral mining and processing. Multiauthored monograph. - Sofia: Publishing House "St.Ivan Rilski", 2018. - 264 p.
4. Полішук О. С. Моделювання частотного перетворювача в системі керування універсальною швейною машиною / О. С. Полішук, С. Л. Горященко, Є. О. Голінка // Вісник Хмельницького національного університету. Технічні науки. - 2014. - № 2. - С. 138-142. - Режим доступу: http://nbuv.gov.ua/UJRN/Vchnu_tekh_2014_2_29.

УДК 004.9

Інформаційні технології

ПЕРСПЕКТИВИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ГАЗОРОЗПОДІЛЬНИХ ПІДПРИЄМСТВАХ

Великоборець О.А.

Державний університет «Житомирська Політехніка»

На сьогоднішній день розвиток технологій дозволяє більшості підприємств використовувати все більш сучасні методи та різноманітні інформаційні технології в своїй діяльності. Починаючи з відносно універсальних програмних комплексів що запроваджуються в більшості організацій на зразок електронного документообігу та системи ведення бухгалтерського обліку і закінчуючи більш спеціалізованими рішеннями використання котрих диктується особливостями діяльності підприємства.

Наприклад у випадку газорозподільної організації на практиці виявляється вкрай незручно вести облік об'єктів газорозподільної мережі (ГРМ) лише за допомогою класичної бази даних. Проблематика полягає в тому що значна кількість об'єктів знаходиться за межами населених пунктів тож чи не єдиний спосіб описати їх розташування – прямо вказати їх на мапі. В таких випадках на допомогу приходять геоінформаційні системи (ГІС) які дозволяють співставити запис в базі даних та розміщення об'єкту на мапі. Подібні системи дозволяють підвищити якість обліку об'єктів ГРМ та суттєво полегшити роботу більшості експлуатаційних підрозділів, робота котрих безпосередньо пов'язана з вищезазначеними об'єктами, спрощуючи планування робіт та формування звітності з обстеження та обслуговування. Впровадження подібної системи та подальші регулярні оновлення для підтримки актуальності даних в межах цієї системи потребують значних витрат часу внаслідок необхідності опрацювання великих обсягів даних для їх подальшої оцифровки. Однак практика використання подібних систем доводить їх зручність, перспективність та ефективність, що в свою чергу мотивує подальше впровадження електронних систем обробки даних. Зразком однієї з таких систем є програмно-апаратні комплекси телеметрії, використання котрих наразі вже практикується або активно впроваджується в низці газорозподільних підприємств України. Ці комплекси дозволяють віддалено за допомогою датчиків здійснювати контроль в реальному часі за показниками тиску, температури, загазованості а при необхідності й інших параметрів на критичних об'єктах ГРМ та зберігати по годинну, а в певних випадках й щохвилинну історію показників в базах даних програмного комплексу. В подальшому оператор може переглядати зведену інформацію з отриманих даних у вигляді таблиць та діаграм, що дозволяє відслідковувати тенденції та проводити аналіз даних на рівні недосяжному раніше – наприклад коливання тиску в системі впродовж доби. Подібні засоби вже широко застосовуються на тисячах об'єктів для дистанційного зняття показників лічильників газу та довели свою ефективність та практичність, дозволяючи в будь-який момент часу переглянути у веб-сервісі актуальну інформацію по обсягах споживання. Більш того все ширшого розвитку набуває телемеханіка – коли оператор додатково має змогу віддалено керувати обладнанням. Тобто система не лише інформує про виникнення нестандартної ситуації, а й надає оператору змогу негайно втрутитись і дистанційно відрегулювати тиск або закрити кран упереджуючи при цьому виникнення аварійної ситуації або мінімізуючи втрати при її виникненні. Особливо актуальним вбачається використання зазначених комплексів в сьогодишніх умовах війни, коли більшість підприємств відчувають нестачу людських ресурсів та у випадках розміщення об'єктів у зоні бойових дій коли дорога до об'єкту може бути небезпечною.

Нажаль незважаючи на привабливість цих засобів існують й певні обмеження що гальмують широке впровадження програмно-апаратних комплексів телеметрії та телемеханіки на об'єктах ГРМ в числі яких є і неминуча фінансова складова питання і обмеження що диктуються рівнем доступних технологій на сьогоднішній день. Так на віддалених об'єктах ГРМ якість покриття мобільних операторів залишає бажати кращого, що ставить під сумнів надійність спрацьовування з причини низької якості каналів зв'язку. Також у більшості випадків неможливо обрати в якості живлення централізоване електропостачання що призводить до необхідності використання акумуляторів для забезпечення живлення відповідних приладів. На практиці це викликає потребу в їх періодичній заміні що особливо відчутно в зимовий період часу коли під впливом низьких

температур акумулятор може повністю розрядитись впродовж тижня. При великій кількості об'єктів здійснення своєчасної заміни може бути проблематичним, що в свою чергу спричиняє відключення систем на окремих об'єктах внаслідок їх знеструмлення. Також глибокий розряд акумуляторів може призвести до їх виводу з ладу без можливості відновлення тож закупівля нових акумуляторів додатково збільшує витрати на утримання комплексу. Зазначені фактори негативним чином впливають на надійність роботи що є ключовою вимогою до програмно-апаратних комплексів телеметрії та телемеханіки.

Висновок: Незважаючи на присутні на даний момент недоліки, на мою думку, в майбутньому неминучим буде поєднання систем ГПС, телеметрії та телемеханіки в єдину систему управління та моніторингу за об'єктами ГРМ що суттєво підвищить ефективність роботи підприємств у цій галузі та дозволить зменшити експлуатаційні витрати, підвищити надійність та безпеку при експлуатації об'єктів ГРМ.

УДК 004.415:37

Інформаційні технології

МОЖЛИВОСТІ УДОСКОНАЛЕННЯ ФУНКЦІОНАЛЬНОЇ СКЛАДОВОЇ ВІРТУАЛЬНИХ НАВЧАЛЬНИХ СЕРЕДОВИЩ ЗА ДОПОМОГОЮ БЛОКЧЕЙНУ

Колиско М.І., Павленко В.М., Поліщук М.

Київський національний університет технологій та дизайну

Незважаючи на зростання кількості децентралізованих онлайн-освітніх платформ, якість курсів часто залишається непостійною. Крім того, відсутність уніфікованої системи сертифікації показує, що результати навчання не є загальновизнаними. Для подолання цих проблем використання блокчейну для збереження навчальних даних є ефективним рішенням.

Технологія блокчейн використовується для зберігання даних у розподіленій базі, що об'єднує кілька вузлів. Інформація систематизується хронологічно за допомогою часових позначок і є непридатною до вилучення після додавання блоку через використання криптографічних алгоритмів для запобігання шахрайській діяльності. З використанням блокчейн, дані студентів, такі як час навчання, файли курсів і результати тестів, можна точно систематизувати за допомогою міток часу та захистити від підробки чи видалення через метод криптографічного запису для збереження цілісності даних.

При всій популярності онлайн-освіти студентам часто не вистачає мотивації для завершення деяких курсів через відсутність визнання та офіційної сертифікації їхніх досягнень. Це пов'язано, в першу чергу, із запізненим розвитком протоколів сертифікації, оскільки не всі агентства ефективно сертифікують онлайн-навчання і не можуть відповідати вимогам швидкозмінюваної сфери онлайн-освіти. Якщо сертифікат втрачено, учень повинен пройти через доволі тривалий процес для отримання дублікату від платформи або навчального закладу.

Технологія блокчейн пропонує просте та ефективне рішення для сертифікації результатів навчання, зокрема академічної сертифікації. Навіть у випадку втрати студентських сертифікатів їх можна легко перевірити. В результаті практичне використання результатів онлайн-освіти може стати ефективнішим.

Зараз наявний великий вибір освітніх онлайн-платформ, що пропонують різноманітні курси, проте вони не завжди взаємодіють між собою через такі обмеження, як методи навчання та авторські права. Це становить серйозну проблему для студентів, які вивчають різні типи курсів, оскільки їм потрібно користуватися різними платформами. Також непросто студентам вищих навчальних закладів вивчати матеріали в інших закладах чи дисциплінах. Часто відбувається дублювання якісних ресурсів курсів що нерационально. У галузі освіти ефективний розподіл ресурсів та спільне їх використання